УДК 004.8:004.942:37.018.43

Н.Г. АКСАК, А.О. ТАТАРНИКОВ, М.В. КУШНАРЬОВ
Харківський національний університет радіоелектроніки

АГЕНТНА МОДЕЛЬ ПЕРСОНАЛІЗОВАНОГО НАВЧАННЯ В NETLOGO З ВИКОРИСТАННЯМ Q-LEARNING

У роботі досліджено ефективність застосування алгоритму підсиленого навчання Q-learning у контексті персоналізованого навчання студентів шляхом створення агентної моделі в середовищі моделювання NetLogo. Запропонована модель передбачає існування множини агентів-студентів, кожен із яких має індивідуальні характеристики, зокрема стиль навчання (візуальний, аудіальний або кінестетичний). Навчальне середовище представлено як сітка патчів, кожен із яких містить навчальний контент визначеного типу (текстовий, відео- або аудіоматеріал) і рівня складності.

Агенти взаємодіють із середовищем, ухвалюють рішення про залишення на поточному місці або перехід до іншої ділянки на основі ε-жадібної стратегії вибору дій. Алгоритм Q-learning дозволяє кожному агенту формувати й оновлювати власну Q-таблицю, що зберігає оцінки корисності дій у різних станах. Винагорода визначається на основі відповідності між стилем навчання агента та типом контенту. Якщо стиль збігається з форматом навчального матеріалу (наприклад, аудіальний стиль і аудіоконтент), агент отримує позитивну винагороду; інакше – негативну.

Результати серії симуляцій засвідчили, що агенти поступово навчаються уникати невідповідного контенту та віддають перевагу ділянкам, що відповідають їхньому стилю сприйняття. Спостерігалося зростання середньої винагороди, збільшення частки агентів із позитивним результатом, а також зменшення частоти випадкових переміщень із часом. Це свідчить про формування стабільної політики навчання та ефективну адаптацію до умов середовища. Окрім того, було проаналізовано вплив параметрів Q-learning (ε, α, γ) на успішність навчання, що дозволило визначити оптимальні конфігурації для досягнення найкращих результатів.

Отримані результати підтверджують доцільність використання підходу підсиленого навчання для моделювання персоналізованих освітніх стратегій. Запропонована модель може бути використана як основа для розроблення інтелектуальних систем підтримки навчання, здатних адаптувати освітній процес до індивідуальних потреб і особливостей учнів.

Ключові слова: *агентне моделювання, підсилене навчання, Q-learning, персоналізоване навчання, стиль навчання, інтелектуальна навчальна система, NetLogo.*

N.H. AKSAK, A.O. TATARNIKOV, M.V. KUSHNARYOV
Kharkiv National University of Radio Electronics

AN AGENT-BASED MODEL FOR PERSONALIZED LEARNING IN NETLOGO USING Q-LEARNING

This study investigates the effectiveness of using the Q-learning reinforcement learning algorithm in the context of personalized student learning by constructing an agent-based model in the NetLogo simulation environment. The proposed model assumes the existence of a population of student agents, each possessing individual characteristics, especially a distinct learning style (visual, auditory, or kinesthetic). The learning environment is modeled as a grid of patches, each containing educational content of a particular type (textual, video, or audio) and a specified difficulty level.

Agents interact with the environment by making decisions to either stay on their current location or move to another patch, based on an ε-greedy action selection strategy. The Q-learning algorithm enables each agent to form and continuously update its own Q-table, which stores the utility estimates of various actions in different states. The reward is determined by the degree of alignment between the agent's learning style and the content type. If the style matches the content format (e.g., auditory style and audio material), the agent receives a positive reward; otherwise, a negative one.

The results of a series of simulations showed that agents gradually learn to avoid mismatched content and begin to prefer patches that correspond to their perception style. An increase in average reward, a growing proportion of agents with positive outcomes, and a decrease in the frequency of random movements over time were observed. This indicates the formation of a stable learning policy and effective adaptation to the environmental conditions. Moreover, the influence of Q-learning parameters (ε, α, γ) on learning success was analyzed, allowing for the identification of optimal configurations for achieving the best performance.

The obtained results confirm the relevance of using reinforcement learning approaches to model personalized educational strategies. The proposed model can be used as a foundation for the development of intelligent learning support systems capable of adapting the educational process to individual learners' needs and traits.

Key words: *agent-based modeling, reinforcement learning, Q-learning, personalized learning, learning style, intelligent tutoring system, NetLogo.*

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

Постановка проблеми

Сучасні освітні системи стикаються з викликом необхідності адаптації навчального процесу до індивідуальних особливостей студентів, зокрема і стилю навчання, рівня знань і емоційного стану. Традиційні підходи до персоналізації мають обмежені можливості щодо динамічного оновлення стратегії викладання в режимі реального часу. Підкріплювальне навчання (Reinforcement Learning, RL), зокрема метод Q-learning, демонструє потенціал у створенні адаптивних моделей, здатних самостійно формувати ефективну освітню політику на основі досвіду взаємодії з учнем.

Попри наявність окремих досліджень у цій сфері, бракує робіт, що поєднують можливості агентного моделювання та Q-learning для реалізації персоналізованого навчання в симульованому середовищі з різноманітним навчальним контентом. Відсутні також моделі, які водночас ураховують стиль навчання, емоційний стан і рівень знань студента як частину простору станів агента.

Отже, постає науково-практична проблема: розробити агентну модель персоналізованого навчання з використанням Q-learning, яка адаптує дії навчальної системи до індивідуальних характеристик студента, забезпечує ефективне підвищення рівня знань і мотивації в умовах змішаного навчального середовища.

Аналіз останніх досліджень і публікацій

Підкріплювальне навчання (далі – RL) привернуло значну увагу в освіті завдяки здатності створювати адаптивні, персоналізовані освітні досвіди. За останні п’ять років дослідження, присвячені RL – зокрема Q-learning – у сфері освітніх технологій, активно розвивалися. RL дозволяє системам навчання динамічно змінювати складність матеріалу та зворотний зв’язок залежно від результатів і поведінки студента, що сприяє підвищенню залучення, мотивації та якості засвоєння матеріалу.

Персоналізовані системи навчання намагаються підлаштуватися під індивідуальні особливості студента – його стиль навчання, рівень знань і емоційний стан. Агентно орієнтоване моделювання в NetLogo дозволяє змоделювати таку взаємодію, створити агентів (наприклад, «студент» і «навчальна система»), які взаємодіють у середовищі. Для адаптації системи до студента можна використати алгоритм *Q-learning* – метод підкріплювального навчання (Reinforcement Learning), у якому агент поступово навчається оптимальної стратегії, пробує різні дії в різних ситуаціях і отримує винагороди за свої дії. У цій роботі розглядається, як реалізувати Q-learning у NetLogo для моделі «студент – персоналізована навчальна система», як основні компоненти Q-learning відповідають парадигмі агентів у NetLogo.

NetLogo отримав розширення, які дозволяють реалізовувати Q-learning безпосередньо в середовищі моделювання. Е. Баццанелла і Ф. Сантуш оцінили нове розширення NetLogo, яке додає команди для Q-learning [1]. Вони показали, що використання розширення дозволяє скоротити обсяг коду (на 14–47%) і спрощує створення агентних моделей. У роботі [2] автори оновили розширення, інтегрували його із Java-бібліотекою BURLAP, що розширило набір RL-алгоритмів (додано SARSA(λ), Actor-Critic). Ці розширення спрощують розробку моделей підкріплювального навчання для освітніх симуляцій у NetLogo.

І. Амзіл і співавтори представили адаптивну освітню систему, яка використовує Q-learning для вибору індивідуального шляху навчання студента [3]. Агент-система відстежує рівень знань студента та динамічно вибирає відповідні дії (наприклад, вправи визначеної складності). Це дозволяє адаптувати темп і методику викладання відповідно до потреб і стилю навчання кожного студента.

Подібно до цього, А. Даган і колеги розробили фреймворк RL, орієнтований на принципи універсального дизайну навчання, що враховує темп, мотивацію та увагу студентів [4]. Агент навчається уподобань і здібностей кожного студента, зокрема й ураховує темп, увагу, мотивацію та когнітивні особливості (наприклад, труднощі з концентрацією). Система адаптує вміст і структуру навчання, щоб підтримувати залучення студента.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

Ширший огляд таких систем представлено в роботі Б.Ф. Мон, де систематизовано приклади впровадження RL в освітніх середовищах [5]. Автори узагальнили використання різних RL-алгоритмів (від марківських процесів до глибокого RL) в адаптивних освітніх системах. Вони виділили найкращі практики, переваги та виклики впровадження RL у цифрові освітні середовища, підкресливши, що такі системи здатні підвищити залучення та успішність студентів.

К. Альхарбі й А.І. Крістеа створили агентну модель класу з реалістичними сценаріями поведінки учнів, що дозволяє оцінити ефективність різних стратегій управління класом [6]. Вони моделювали реальну шкільну динаміку за допомогою даних із британських початкових шкіл. Хоча RL не було використано безпосередньо, дослідження заклало основу для подальшої інтеграції навчальних агентів.

У пізнішій роботі автори поєднали агентне моделювання з машинним навчанням, зокрема з регресійним прогнозуванням результатів навчання [7]. Вони використовували лінійну регресію для прогнозування результатів навчання й інтегрували її в симуляцію класу. Такий гібридний підхід дозволив моделі враховувати як статистичні залежності, так і динамічну взаємодію агентів у віртуальному середовищі.

Х. Говеа зі співавторами реалізували модель глибокого підкріплювального навчання для виявлення емоцій студентів у гібридному навчальному середовищі (з використанням камер, мікрофонів і біометричних даних) [8]. RL-агент розпізнає емоції (фрустрація, нудьга, зацікавленість) і адаптує навчальні стратегії в реальному часі. У результаті точність розпізнавання емоцій зросла із 72 до 89%, а персоналізація навчання – із 70 до понад 90%. Це показує потенціал RL для створення емоційно чутливих адаптивних систем навчання.

І. Осакве та співавтори розглянули саморегульоване навчання (далі – SRL) як послідовний процес ухвалення рішень і застосували Q-learning для виявлення ефективних послідовностей дій (наприклад, самотестування, пошук допомоги, повторення матеріалу) [9]. RL-модель змогла виявити оптимальні навчальні стратегії, які забезпечили кращі результати, ніж нейронні мережі та генетичні алгоритми. Цей підхід відкриває нові можливості для інтелектуального підтримання студентів у виборі ефективної стратегії навчання.

У межах досліджень підвищення ефективності е-навчання на основі агентного моделювання варто також згадати роботу N. Axak, A. Tatarnykov, M. Kushnaryov [10], у якій запропоновано агентно орієнтований підхід до індивідуалізації освітніх траєкторій. Автори розробили метод, що дозволяє адаптувати навчальний процес відповідно до особливостей користувача, з урахуванням параметрів середовища та результатів попередніх дій. Запропонована модель забезпечує зростання ефективності завдяки аналізу динаміки навчання і ухвалення рішень на основі даних, що узгоджується з концепціями підкріплювального навчання.

Отже, упродовж 2020–2025 рр. значно зріс інтерес до поєднання RL, агентного моделювання та освітніх технологій. Було створено інструменти для інтеграції Q-learning із NetLogo, що спростило розроблення адаптивних агентів. У статтях розглядаються різні приклади: від персоналізованих репетиторських систем до симуляцій класів і моделей, що враховують емоційний стан.

Виділяються кілька тенденцій:

1. Поглиблення персоналізації – RL-системи навчаються реагувати на індивідуальні особливості студентів.

2. Урахування емоцій і стратегій навчання – моделі стають більш «людяними», зважають на настрій, мотивацію, стиль навчання.

3. Гібридні підходи – поєднання ABM із ML забезпечує точніше прогнозування навчальних результатів.

4. Розроблення кращих практик – формуються рекомендації з вибору алгоритмів, винагород і способів представлення станів студента.

Отже, RL у NetLogo стає ефективним інструментом для створення інтелектуальних, адаптивних і емпатичних освітніх систем, що реагують на потреби кожного студента.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

Виклад основного матеріалу дослідження

Компоненти Q-learning в агентній моделі NetLogo. Метод Q-learning оперує п'ятьма основними компонентами, як-от: ***агент, стан середовища, дії агента, винагорода*** та ***Q-таблиця*** (таблиця значень функції Q).

1. У підкріплювальному навчанні ***агент*** – це той, хто ухвалює рішення і навчається на основі взаємодії із середовищем. У нашій моделі можна виділити два типи агентів, як-от: студент і навчальна система (або «учитель»). Логіку Q-learning доцільно покласти на агента – навчальну систему, оскільки саме система повинна адаптувати свої дії під студента. Отже, RL-агентом у моделі буде персоналізована навчальна система, реалізована як окремий агент NetLogo (наприклад, черепашка породи tutor). Агент-студент (черепашка породи student) виступатиме частиною середовища, на яке впливають дії системи, водночас надаватиме зворотний зв'язок (винагороду) для навчання системи.

У NetLogo кожен агент може мати власні властивості (змінні). Зокрема, студент матиме такі змінні, як рівень знань, тип навчального стилю, емоційний стан тощо – вони визначають поточний стан, у якому перебуває студент. Агент-система може мати свою пам'ять (таблицю Q) та поточне уявлення про стан студента. Завдяки *агентній парадигмі* NetLogo кожен агент діє незалежно, що узгоджується з концепцією RL-агента, який самостійно ухвалює рішення і навчається.

2. ***Стан*** у Q-learning – це опис поточної ситуації, на основі якого агент вибирає дію. У нашій задачі стан повинен відображати характеристики студента, важливі для вибору навчальної стратегії. До таких характеристик належать:

– поточний рівень знань студента, представлений числовою змінною knowledge-state (рівень засвоєння матеріалу, % виконаних правильно завдань тощо) або категорією (низький/середній/високий рівень знань);

– стиль навчання – змінна learning-style, що може набувати значень на кшталт «візуальний», «аудіальний», «кінетичний» тощо (визначає, як студент краще сприймає інформацію);

– емоційний стан – змінна emotion для настрою чи залученості студента (наприклад, шкала від −1 до 1, де −1 – сильне розчарування, 0 – нейтральний стан, +1 – позитивне збудження, або дискретні стани: спокійний, зацікавлений, фрустрований тощо).

Комбінація цих чинників визначає повний стан середовища з погляду навчальної системи. У NetLogo ці властивості можна реалізувати як власні змінні черепашки студента.

Це дозволить кожному агенту-студенту зберігати свій поточний стан. Навчальна система (агент-репетитор) у виборі дії буде звертатися до цих змінних студента, щоб отримати поточний стан.

Для застосування Q-learning стан має бути вибраний із деякого кінцевого набору. Тому безперервні показники (рівень знань як відсоток, або емоція як дійсне число) доцільно розбити на категорії. Наприклад, knowledge-state можна дискретизувати: <40% = «низький», 40–70% = «середній», >70% = «високий» рівень знань. Емоцію можна квантувати у значення −1, 0, +1 (негативний, нейтральний, позитивний). Стиль навчання зазвичай фіксований для студента й задається під час ініціалізації. Отже, стан агента – навчальної системи можна подати як кортеж або комбінацію значень: наприклад, <learning-style, knowledge-state-category, emotion-category>. Зрештою, будь-якої миті симуляції студент перебуває в якомусь стані із цього простору, навіть якщо для деяких станів немає явної нагороди чи вони рідкісні.

У NetLogo можна реалізувати стан як єдину змінну (наприклад, код або список). Для простоти можна зберігати поточний стан в агента-репетитора, формуючи його із властивостей студента, наприклад: set current-state (list [learning-style] of student0 [knowledge-state-cat] of student0 [emotion-cat] of student0) – список, що кодує стан конкретного студента (де student0 – звернення до конкретної черепашки-студента). Інший підхід – зіставити кожну комбінацію станів із координатою патча на сітці NetLogo. Наприклад, можна штучно створити сітку, за віссю *x* якої відкладено

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

рівень знань, за віссю *y* – емоційний стан, а різні типи стилів навчання представляти окремими «шарами» або різними студентами. Кожному патчу можна призначити значення Q.

3. Дії – це можливі кроки, які агент – навчальна система може виконати, коли перебуває у визначеному стані (тобто працюючи з конкретним студентом в його поточному стані). Набір можливих дій повинен відображати різні стратегії навчання або педагогічні втручання.

Умовно дії агента – навчальної системи можна поділити на три типи:

1. Навчальні дії:

– пояснити матеріал – надати теоретичне пояснення або лекцію;

– показати приклад/ілюстрацію – візуалізувати концепцію, навести кейс;

– дати практичне завдання – вправу або задачу для вирішення студентом.

2. Оцінювальні дії:

– поставити тестове питання – перевірити знання, опитування за щойно вивченим.

3. Підтримувальні дії:

– надати підказку чи консультацію – якщо студент застряг або виглядає розгубленим;

– змінити підхід подачі – наприклад, перейти до більш візуального пояснення, якщо стиль студента візуальний, або навпаки.

Цей список дій буде зашитий у модель як набір дискретних опцій. У NetLogo їх можна реалізувати як умовні процедури чи номіновані константи. Наприклад, можна визначити константи або просто використовувати індекси дій: 0 – пояснення, 1 – приклад, 2 – вправа, 3 – тест, 4 – підказка. Агент-репетитор міститиме логіку вибору дії (на основі Q-таблиці та поточного стану) і виконання цієї дії.

Виконання дії в моделі означає зміну деяких параметрів студента. Наприклад, якщо дія – дати практичне завдання, то код моделі може викликати процедуру, яка з імовірністю, залежною від рівня знань, визначає, чи студент успішно виконав завдання. Від цього залежатиме зростання рівня знань (обчислення приросту знань (learning-gain) у разі успіху) та зміна емоційного стану (наприклад, підвищення emotion, якщо успішно – радість від успіху, або зниження в разі невдачі – фрустрація). Таким чином, дія агента зумовлює перехід стану студента: були значення (старий стан), після дії деякі змінні змінилися (новий стан). Важливо, що кожна дія може мати різний ефект залежно від стилю навчання – наприклад, візуальному учню демонстрація прикладу може дати більший приріст знань і мотивації, ніж довге текстове пояснення. Ці залежності теж можна закласти в модель під час оновлення стану.

4. *Винагорода* – це числовий зворотний зв'язок, який агент отримує після виконання дії в певному стані. Мета агента – максимізувати очікувану сумарну винагороду, тому правильне визначення винагороди критично впливає на те, чого навчиться система. У навчальній моделі винагородою може слугувати успішність навчання та задоволеність студента. Зазвичай винагороду формують на основі зміни стану студента:

– прогрес знань: якщо після дії рівень знань студента зріс (наприклад, студент правильно виконав завдання, зрозумів тему), то агент отримує позитивну винагороду. Чим більший приріст знань, тим більша винагорода;

– емоційна реакція: якщо дія покращила настрій студента або підтримала його мотивацію (зріс показник emotion), це теж бажано й можна додати до винагороди. Навпаки, якщо студент засмучений або нудьгує (emotion знижується), можна вважати це негативною винагородою (штрафом) для системи;

– досягнення цілей навчання: додатково можна давати велику позитивну винагороду, коли студент досягає якогось ключового результату (наприклад, склав підсумковий тест, закрив прогалину у знаннях). Негативну – якщо студент відмовляється продовжувати навчання (украй негативний емоційний стан).

Формально, можна визначити функцію винагороди *R (state, action, newState)*, що обчислює скаляр на основі старого стану, вибраної дії та нового стану.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

Наприклад: +10 балів за одиницю прогресу у знаннях (тобто learning-gain), +5 за покращення настрою. Якщо знання чи настрій погіршилися, винагорода буде від'ємна. Звісно, параметри можна підбирати. Важливо, що винагорода визначається таким чином, щоб стимулювати бажану поведінку системи – тобто ефективне навчання з урахуванням комфорту студента.

Агент-репетитор після виконання дії обчислює винагороду на основі того, як змінився стан студента, і використовує її для оновлення своєї Q-таблиці. У NetLogo значення винагороди може бути локальною змінною у процедурі, яка потім використовується у формулі Q-оновлення.

5. *Q-таблиця* – це структура даних, де агент зберігає оцінки корисності (значення Q) для пар «стан – дія». Кожен запис Q (s, a) відображає, наскільки цінним для агента є виконання дії *a* у стані *s* із погляду очікуваного отримання винагороди в майбутньому. Під час навчання ці оцінки поступово наближаються до оптимальних. Алгоритм Q-learning оновлює таблицю за правилом:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma max_a Q(s',a') - Q(s,a)\right],$$

де α – коефіцієнт навчання (*learning rate*), γ – фактор дисконтування майбутніх нагород (*discount factor*), r – отримана винагорода, s' – новий стан після дії.

Параметр α визначає, наскільки швидко агент ураховує новий досвід (вищі значення – швидше, але потенційно нестабільніше навчання), а γ $(0 \le \gamma \le 1)$ задає, наскільки агент цінує майбутні винагороди порівняно з негайними (вищий γ – більше уваги до довгострокової вигоди).

У таблиці 1 подано приклади взаємозв'язку між поточним станом студента, дією агента – навчальної системи й очікуваною винагородою, що використовується в алгоритмі Q-learning. Такий формат дозволяє формалізувати адаптацію навчального контенту до індивідуальних характеристик учня.

Таблиця 1

Приклад відповідності: *Стан* → Дія → *Очікувана винагорода*

Стан	**Дія**	**Очікувана винагорода**
Середній рівень знань, аудіальний стиль, нейтральний настрій.	Пояснити матеріал (вербально).	+10 до знань, +3 до емоційного стану.
Низький рівень знань, візуальний стиль, фрустрований.	Показати приклад (візуальний контент).	+15 до знань, +5 до настрою.
Високий рівень знань, кінестетичний стиль, позитивний настрій.	Дати практичне завдання.	+5 до знань, +2 до мотивації.

Реалізація Q-таблиці в NetLogo

NetLogo не має вбудованого типу для матриці, але є кілька способів зберігати Q-значення. Один із них як структура даних table: NetLogo надає розширення table (хеш-таблиці/словники). Можна створити для агента-репетитора змінну q-table типу таблиця, де ключами будуть пари (стан, дія), а значеннями – числові Q. Наприклад, ключем може бути список [state action].

Параметри та змінні моделі:

1. Стан агента-студента кодується як комбінація трьох характеристик state = [learning-style, knowledge-state, emotion] (табл. 2).

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

Таблиця 2

Learning-style, knowledge-state, emotion

Назва змінної	**Позначення**	**Значення**	**Призначення**
Стиль навчання	learning-style	візуальний / аудіальний / ін.	Визначає перевагу у форматі подачі
Рівень знань	knowledge-state	низький/середній/високий	Поточне засвоєння матеріалу
Емоційний стан	emotion	−1 … +1 або категорії	Мотиваційний і настроєвий фон

2. Приріст знань (learning-gain):

– визначає динаміку змін знань як різницю між новим і попереднім рівнем знань: learning-gain = knowledge-state(нове) – knowledge-state (попереднє);

– використовується для обчислення винагороди.

3. Q-таблиця (q-table):

– виглядає як словник (хеш-таблиця), де ключем є [state, action], а значенням – числова оцінка Q.

Параметри алгоритму Q-learning наведені в таблиці 3.

Таблиця 3

Параметри алгоритму Q-learning

Позначення	**Назва**	**Типове значення**	**Призначення**
α	learning rate	0.1	Швидкість оновлення знань
γ	discount factor	0.9	Вага майбутніх винагород
ε	epsilon (explore)	0.1	Імовірність випадкової дії

Представлення прогресу навчання та емоційного стану

У моделі NetLogo прогрес навчання (knowledge-state) і емоційний стан (emotion) студента відображають зміну стану агента в результаті дії. Значення знань збільшується в результаті успішних дій (наприклад, розв'язання вправи), а емоції – за позитивного досвіду (успіх, відповідність стилю). Ці змінні використовуються для обчислення винагороди та входять до простору станів агента. Для візуалізації можна відображати колір студента залежно від емоційного стану або виводити рівень знань у вигляді числового лейбла над агентом. Це спрощує аналіз поведінки агентів під час симуляції.

Рівень знань (knowledge-state) студента можна інтерпретувати як частку засвоєного матеріалу або кількість набраних балів. Початково ця змінна встановлюється на певний рівень (наприклад, 0 або якась оцінка попередніх знань). Кожна навчальна дія потенційно змінює рівень знань: успішне вирішення завдання, отримання зрозумілого пояснення збільшує цей рівень, тоді як пропуск або невдача можуть залишити без змін або навіть знизити (хоча знання зазвичай не втрачаються, можна вводити штраф за забуття в разі довгого нерозуміння). Оновлення knowledge-state здійснюється у процедурах дій (як показано раніше в perform-action). Для спостереження за прогресом у моделі можна будувати графік знань студента по часових кроках або виводити цю змінну на моніторі.

Емоційний стан (emotion) – складніша величина для моделювання, але її теж можна спростити до одного показника. Як зазначалося, можна задати шкалу від −1 до +1 або індекс настрою. Важливо оновлювати цей стан на основі подій: якщо студенту стало нудно (наприклад, багато раз повторюється вже відомий матеріал), emotion може зменшуватися. Якщо студент фрустрований (кілька невдалих спроб вирішити завдання), показник також знизиться. Навпаки, зацікавленість росте, коли дії системи відповідають стилю студента та рівню виклику: наприклад, успішне виконання складнішого завдання може підняти настроєність. У соціальному моделюванні емоцій часто використовують підхід, коли емоції представлені динамічними змінними агента.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

Тобто кожен агент має змінну, що описує інтенсивність якоїсь емоції, яка змінюється із часом за визначеними правилами. У нашому випадку emotion є саме такою динамічною змінною: її значення змінюється за кожної взаємодії залежно від результату. Можна навіть моделювати поступове згасання емоцій (наприклад, якщо студент довгий час не діє, його emotion повільно повертається до нейтрального стану – реалізується в циклі go без дій або як частина tick).

Стиль навчання (learning-style) студента, на відміну від перших двох, можна вважати фіксованою властивістю (або рідко змінюваною). Її задають під час створення агента (випадково або відповідно до сценарію). Ця змінна впливає на те, як навчальна система інтерпретує стан і вибирає дії. Наприклад, система може навчитися того, що для студента з візуальним стилем дії, пов'язані з ілюстраціями, приносять вищу винагороду (бо і знання, і емоції покращуються більше), тоді як для аудіального студента – ліпше спрацьовують вербальні пояснення. Отже, learning-style включено до простору станів, і Q-learning автоматично врахує його у формуванні різних значень Q для дій залежно від стилю (фактично, агент навчиться різної політики для різних стилів, якщо вони приведуть до різних винагород).

Моніторинг і візуалізація. NetLogo дозволяє легко спостерігати за змінними агентів під час симуляції. Наприклад, можна відображати емоційний стан студента через колір черепашки: зробити правило, що за високого emotion (>0,5) студент-патч змінює колір на зелений (щасливий), за низького (<−0,5) – на червоний (фрустрований), а між ними – на нейтральний сірий. Це забезпечить візуальний зворотний зв'язок у моделі. Також можна відображати рівень знань студента над агентом (наприклад, set label knowledge-state) або графіком, що показує прогрес. Отже, під час моделювання видно, як дії системи впливають на студента як у кількісному вимірі (знання), так і в якісному (настрій).

Методика проведення експериментів

Метою експериментів було дослідження ефективності використання алгоритму Q-learning для персоналізованого навчання студентів у середовищі NetLogo. Агентами виступають студенти з різними стилями навчання (візуальний, аудіальний, кінестетичний), які взаємодіють із навчальним контентом різного типу (текст, відео, аудіо) та складності.

Середовище реалізовано у вигляді сітки патчів із випадковим розподілом типів контенту. Кожен агент використовує ε-жадібну політику для вибору дій (залишитися або рухатись випадково) і оновлює Q-таблицю відповідно до отриманої винагороди.

Дані про середню винагороду, відсоток позитивних винагород і кількість агентів, які переміщуються, автоматично зберігалися в CSV-файл. Для аналізу даних використовувався Python із бібліотеками pandas і matplotlib.

Було проведено серію запусків моделі з різними параметрами навчання: ε, α, γ. Зокрема, розглянуто конфігурацію з $\varepsilon = 0{,}1$, $\alpha = 0{,}5$, $\gamma = 0{,}9$.

У моделі було реалізовано симуляцію персоналізованого навчального середовища, у якому агенти-студенти з різними стилями навчання взаємодіють із навчальним контентом (відео, текст, аудіо) різної складності. Агент – навчальна система реалізує алгоритм Q-learning, адаптуючи стратегії викладання на основі спостережень за реакціями студентів.

Для моделювання навчального процесу було створено 50 агентів-студентів, кожному з яких випадково присвоюється один із трьох стилів навчання: візуальний, аудіальний або кінестетичний (learning-style). Агенти розміщуються у випадкових координатах середовища, що імітує навчальний простір.

Кількість агентів зафіксовано у процедурі setup-students, яка є частиною початкової ініціалізації моделі (setup). Це дозволяє забезпечити повторюваність експерименту зі збереженням водночас випадковості в параметрах кожного агента (місцерозташування, стиль).

Усі агенти мають власну таблицю дій q-table, яка формується під час симуляції у процесі навчання на основі взаємодії зі змістовними елементами середовища й обчисленої винагороди.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

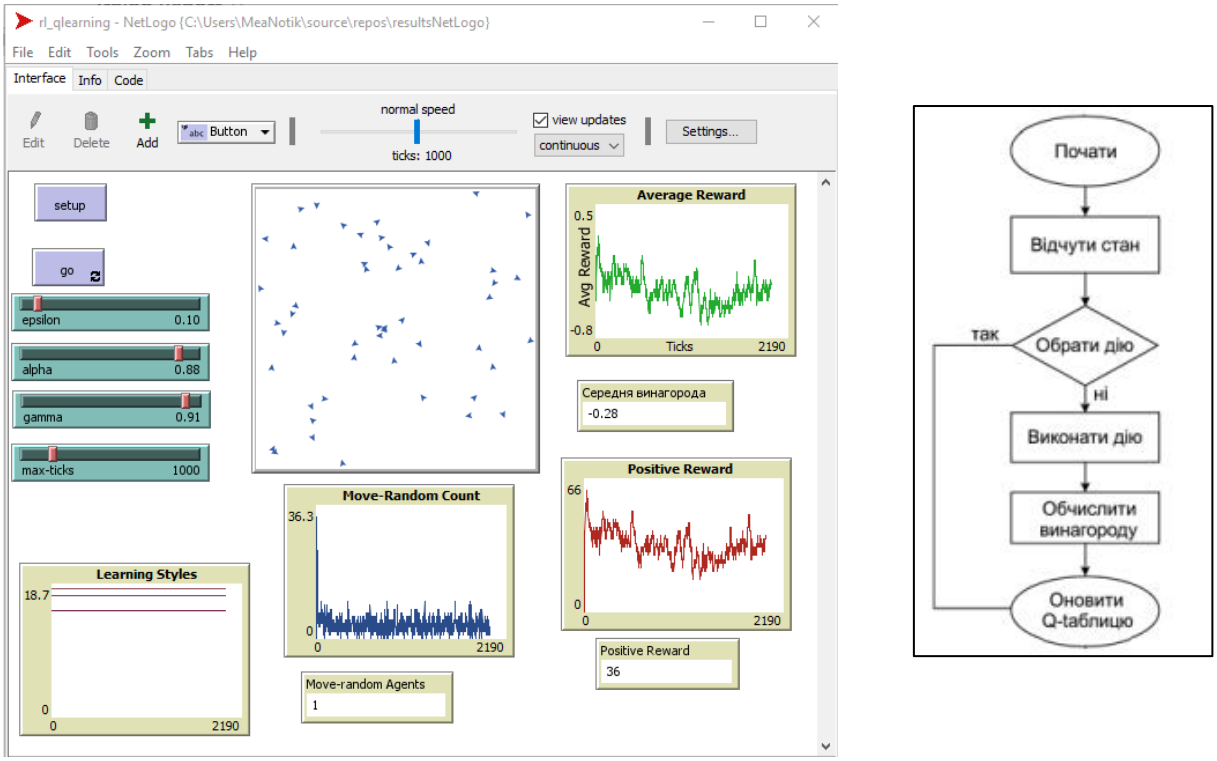
У вікні моделі видно приблизно 50 трикутників – це черепашки-агенти, які виконують дії згідно з Q-learning (рис. 1-а). Початковий стан симуляції в NetLogo: агенти-студенти розміщені випадково на сітці патчів, кожен із яких містить контент певного типу; зміна кольору агентів відображає емоційний стан, числові мітки показують рівень знань. Це зображення ілюструє, як виглядає просторове середовище та візуалізація змінних студентів у процесі навчання. Рисунок 1-б ілюструє як візуальний інтерфейс моделі, так і внутрішню логіку функціонування навчального агента.

У моделі NetLogo реалізовано підкріплювальне навчання агентів у середовищі з різними типами контенту. Параметри ε (epsilon), α (alpha) та γ (gamma) були автоматично варійовані в заданих діапазонах за допомогою скрипту на Python із використанням pyNetLogo. Для кожного запуску моделі зберігались середнє значення винагороди та відсоток агентів із позитивною винагородою.

З метою оцінювання впливу параметрів навчання було реалізовано автоматизований random search із Python-інтерфейсом pyNetLogo. Для кожного набору параметрів (ε, α, γ) NetLogo-сценарій запускався на 1 000 кроків. Було проведено 10 запусків із випадковими параметрами, кожен із яких фіксував:

– середню винагороду агентів (avg_reward);

– відсоток агентів із позитивною винагородою (positive_reward_percent).



а б

Рис. 1: а) – початковий вигляд моделі NetLogo з агентами-студентами, розміщеними у випадкових позиціях на сітці патчів; б) – блок-схема логіки агента: послідовність дій з ухвалення рішень на основі поточного стану, вибору дії, отримання винагороди й оновлення Q-таблиці

У моделі агенти-студенти пересуваються простором навчального середовища; кожен агент має власний стиль навчання (візуальний, аудіальний, кінестетичний); кожна ділянка (patch) має тип навчального контенту (text, video, audio) і рівень складності; агенти використовують алгоритм Q-learning для вибору дій: залишитися або рухатися; винагорода залежить від відповідності стилю навчання агента типу контенту.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

Поведінка агентів і ефективність навчання значною мірою залежать від параметрів Q-learning. Нижче наведено конфігурацію, яка продемонструвала найкращі результати під час серії симуляцій:

– Сценарій симуляції: запуск № 3.

– Параметри навчання:

$\varepsilon = 0{,}2$ (імовірність випадкової дії);

$\alpha = 0{,}52$ (швидкість навчання);

$\gamma = 0{,}81$ (дисконт майбутніх винагород).

– Середня винагорода: $-0{,}24$.

– Частка агентів із позитивною винагородою – 38%.

Результати свідчать про часткову адаптацію агентів до відповідного контенту, проте значна частина дій ще залишається неефективною.

На рисунку 2 показана динаміка змін у поведінці агентів за часом. Видно, як із ростом досвіду зменшується випадковість дій і зростає ефективність.

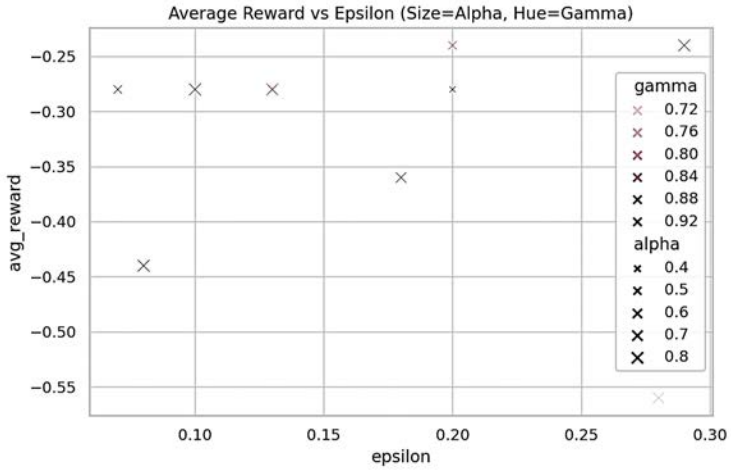


Рис. 2. Графіки середньої винагороди, частки агентів із позитивною винагородою та кількість переміщень

Рисунок 3 демонструє, як модель накопичує знання про ефективність дій у різних станах.

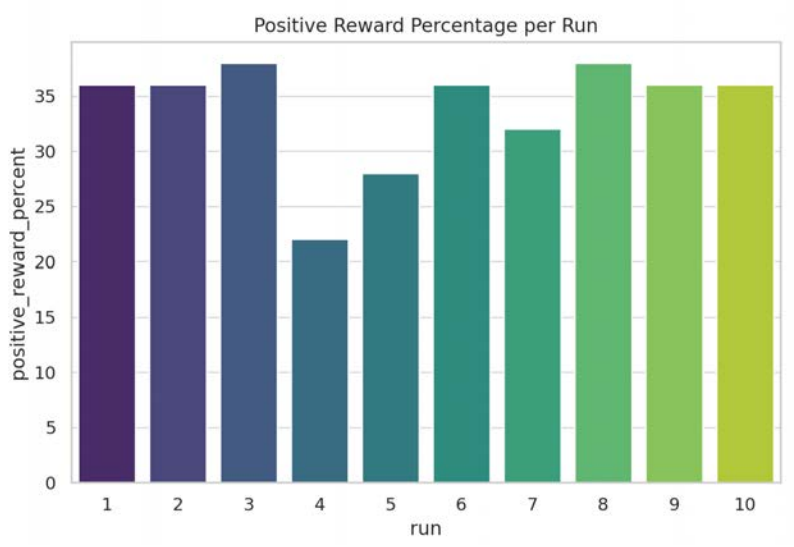


Рис. 3. Візуалізація Q-таблиці агента під час симуляції

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

Нижче наведено графіки залежностей середньої винагороди від параметрів. На рисунку 4 видно, що менші значення ε (менше дослідження) приводять до вищої ефективності агентів.

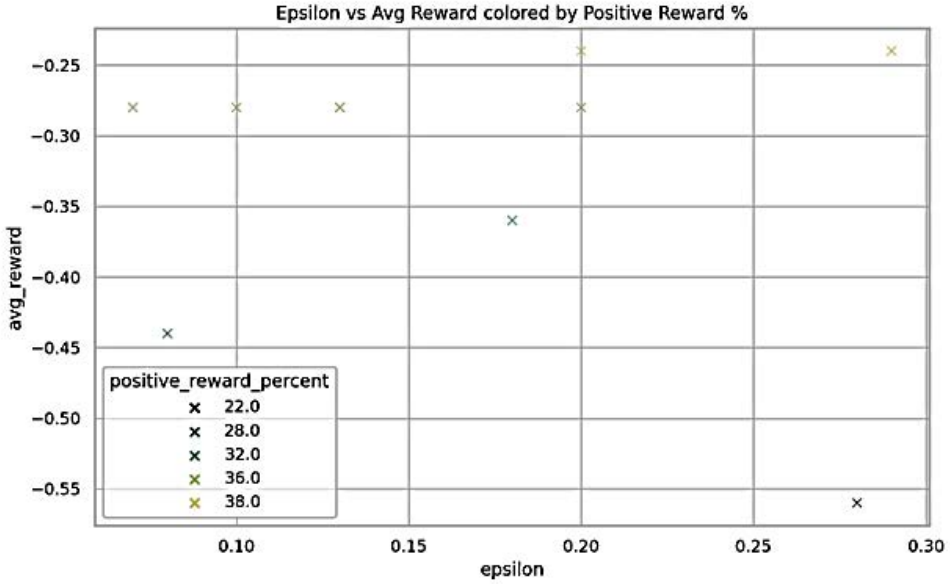


Рис. 4. Залежність середньої винагороди від ε (epsilon) – коефіцієнта експлорації

Оптимальні значення (рис. 5) забезпечують баланс між стабільністю та адаптивністю політики навчання.

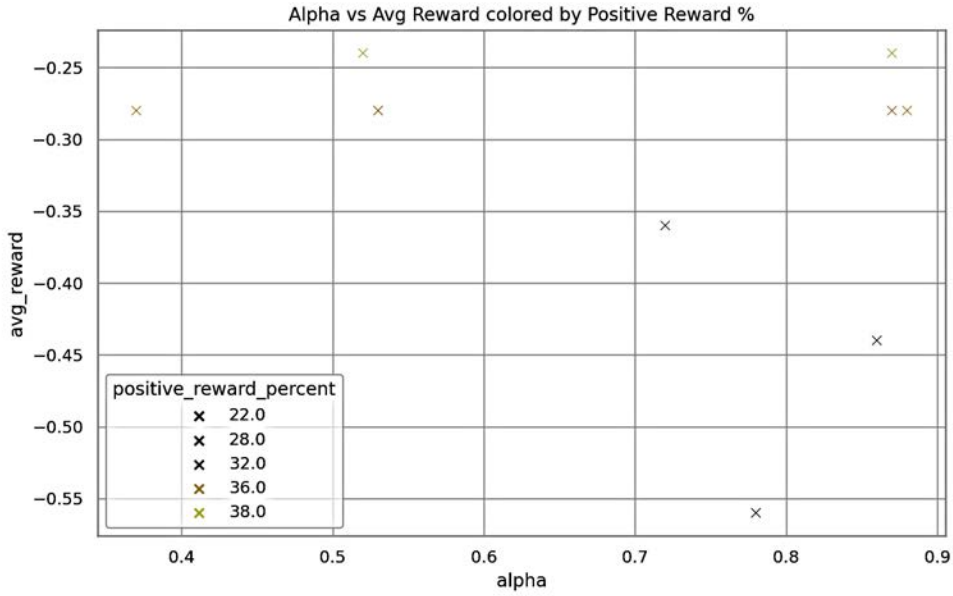


Рис. 5. Вплив α (alpha) – коефіцієнта навчання на результати симуляції

Високі значення γ (рис. 6) стимулюють стратегічне мислення агентів, але можуть знижувати негайну винагороду.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

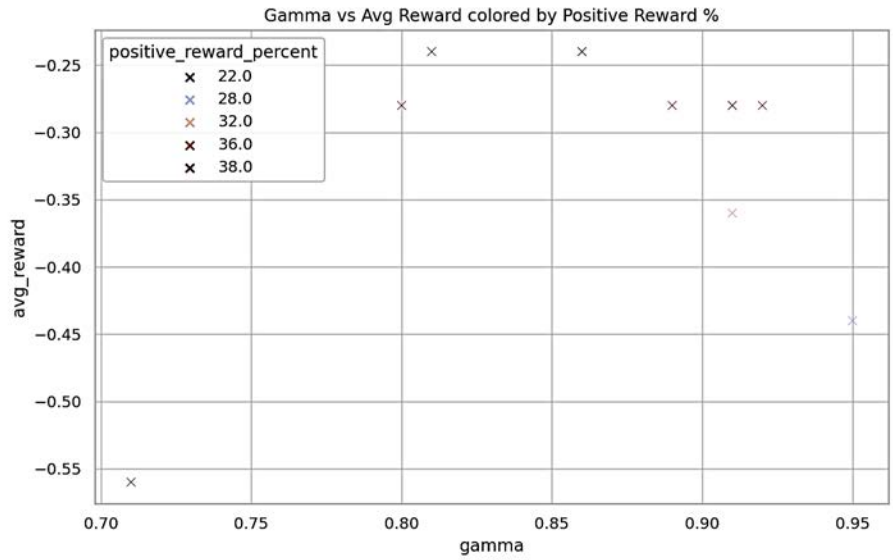


Рис. 6. Залежність результатів симуляції від γ (gamma) – параметра дисконтування майбутніх винагород

Аналіз параметрів та результати симуляцій

У процесі дослідження було проаналізовано вплив параметрів алгоритму Q-learning на ефективність навчання агентів у симуляційному середовищі. Оптимальні значення параметрів ε, α та γ визначалися за допомогою методу випадкового пошуку (*random search*).

Найкращі результати спостерігалися за:

ε ≈ 0,1–0,2 (коефіцієнт експлорації);

α ≈ 0,5–0,7 (швидкість навчання);

γ ≈ 0,9–0,99 (дисконт майбутніх винагород).

Однак значна варіативність у результатах свідчить про потенційну складність середовища та доцільність подальшого дослідження з використанням методів GridSearch або Bayesian Optimization.

У межах серії з 10 запусків, виконаних із рандомізованими комбінаціями параметрів, були отримані такі результати:

– середня винагорода (avg_reward) варіювалася від 0,56 до 0,20. Що ближче значення до нуля, то кращою була адаптація агентів до контенту;

– відсоток позитивної винагороди досягав 40% і вище, тобто до 40% агентів ефективно навчалися в конкретних умовах.

Інтерпретація параметрів:

– коефіцієнт експлорації ε (0,07–0,27) – найвищі результати досягались за низьких значень ε (0,07–0,18), що свідчить про ефективність експлуатації перевірених стратегій, тобто студенти, які дотримуються перевірених методів навчання, мають вищу ефективність;

– швидкість навчання α (0,35–0,74) – оптимальними виявились значення в межах 0,4–0,6, які забезпечували баланс між новим досвідом і вже накопиченими знаннями; занадто високе або низьке α спричиняє нестабільність у виборі дій;

– орієнтація на майбутнє γ (0,72–0,94) – завищене γ (>0,9) часто знижувало ефективність, отже, надмірна орієнтація на далекі винагороди може бути недоцільною в умовах навчання, це означає, що агенти мають бути орієнтовані на короткострокову вигоду, тобто швидкий результат у навчанні краще мотивує.

Узагальнені висновки: найкращі результати (вища середня винагорода та відсоток позитивного навчання) спостерігались за:

ε ≈ 0,1–0,18 (менше дослідження);

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

$\alpha \approx 0{,}4–0{,}6$ (помірне навчання);

$\gamma \approx 0{,}72–0{,}84$ (орієнтація на поточні винагороди).

Це узгоджується з освітньою практикою: швидкий зворотний зв'язок, помірна адаптація до нових підходів і фокус на успішному досвіді дають найкращі результати в персоналізованому навчанні.

Практичні рекомендації:

– інтелектуальні навчальні системи можуть адаптувати поведінку агентів відповідно до стилю учня, динамічно підлаштовувати контент;

– Q-learning – потужний інструмент для формування гнучких навчальних траєкторій;

– модель може змінювати вагу нових даних залежно від поточної успішності студента, забезпечувати адаптивну стратегію викладання.

Візуальний аналіз і підтвердження.

Графіки показали, що найкращі результати досягалися за низьких значень ε (0,07–0,18), помірного α (0,4–0,6) та високого γ (0,84–0,94). Це узгоджується з гіпотезою про те, що успішне персоналізоване навчання потребує фокусування на вже відомих ефективних стратегіях з обережним дослідженням нових підходів:

– ε: менші значення сприяють експлуатації перевірених стратегій. У контексті освіти це означає, що агенти, які менше експериментують, а частіше повторюють ефективні стратегії, досягають кращих результатів;

– α: помірне значення дозволяє балансувати між новими і старими знаннями. Це важливо для гнучкого оновлення політик агента без втрати накопиченого досвіду;

– γ: високі значення зумовлюють орієнтацію на майбутні вигоди, що добре узгоджується з освітніми завданнями, де мета – довготривалий прогрес студента.

Висновки та подальші напрями дослідження

У статті представлено агентну модель персоналізованого навчання на основі алгоритму Q-learning, реалізовану в середовищі NetLogo. Агенти-студенти взаємодіють із навчальним контентом, що вирізняється за типом і складністю, і адаптують свою поведінку відповідно до стилю навчання. Результати моделювання демонструють ефективність підходу до адаптивного вибору дій на основі винагороди, що залежить від відповідності контенту індивідуальним особливостям агента.

Проведений аналіз впливу параметрів Q-learning (ε, α, γ) показав, що:

– низький рівень експлорації ($\varepsilon \approx 0{,}1–0{,}2$) сприяє ефективному використанню вже відомих стратегій;

– помірні значення швидкості навчання ($\alpha \approx 0{,}4–0{,}6$) забезпечують баланс між стабільністю і адаптивністю;

– значення коефіцієнта дисконтування ($\gamma \approx 0{,}72–0{,}84$) дозволяють агентам зосередитись на коротко- та середньострокових цілях.

Отримані результати узгоджуються із принципами персоналізованого навчання, де важливі швидкий зворотний зв'язок, гнучке оновлення знань і фокус на індивідуальний прогрес.

У подальших дослідженнях передбачається:

– розширення підходу шляхом використання GridSearch або Bayesian Optimization для точнішого налаштування параметрів;

– моделювання взаємодії між агентами та її впливу на колективну динаміку навчання;

– інтеграція когнітивних і емоційних факторів у функцію винагороди для підвищення реалістичності моделі.

Список використаної літератури

1. Bazzanella E., Santos F. Does a Q-learning NetLogo Extension Simplify the Development of Agent-based Simulations*? Proceedings of the 15th Workshop-School on Agents, Environments,*

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

and Applications (WESAAC 2021), Porto Alegre, 2021. P. 1–12. https://doi.org/10.5753/wesaac.2021.33403.
2. Bazzanella E., Barros M.M., Santos F. Refactoring the NetLogo Reinforcement Learning Extension for Integration with the BURLAP Library. *Proceedings of the 18th Workshop-School on Agents, Environments, and Applications* (WESAAC 2024). Porto Alegre, 2024. P. 51–62. https://doi.org/10.5753/wesaac.2024.33455.
3. Amzil I., Aammou S., Tagdimi Z., Erradi H. Personalizing Learning Experiences with Q-Learning in Adaptive Educational Systems *E-Learning and Smart Engineering Systems (ELSES 2023).* Atlantis Press, 2023. P. 70–77. DOI: 10.2991/978-94-6463-360-3_9.
4. Dahan A., Roth N., Pelosi A.D., Reiner M. A Reinforcement Learning Framework for Personalized Adaptive E-Learning. *Advanced Technologies and the University of the Future*. Lecture Notes in Networks and Systems. Springer, Cham. 2024. Vol. 1140. P. 141–162. https://doi.org/10.1007/978-3-031-71530-3_10.
5. Mon B.F., Wasfi A., Hayajneh M., Slim A., Abu Ali N. Reinforcement Learning in Education: A Literature Review. *Informatics*. 2023. Vol. 10 (3). P. 74. https://doi.org/10.3390/informatics10030074.
6. Alharbi K., Cristea A.I., Shi L., Tymms P., Brown C. Agent-Based Simulation of the Classroom Environment to Gauge the Effect of Inattentive or Disruptive Students. *Intelligent Tutoring Systems (ITS 2021)*. Lecture Notes in Computer Science, Vol. 12677. Springer, Cham. 2021. P. 211–223. https://doi.org/10.1007/978-3-030-80421-3_23.
7. Alharbi K., Cristea A.I. Hybrid Agent-Based Machine Learning Simulation of a Classroom Disruption Model *Proceeding of the 38th ECMS conference, editors: Daniel Grzonka, Natalia Rylko, Grazyna Suchacka, Vladimir Mityushev.* 2024. Vol. 38. № 1. P. 125–132.
8. Govea J., Maldonado Navarro A., Sánchez-Viteri S., Villegas-Ch. W. Implementation of Deep Reinforcement Learning Models for Emotion Detection and Personalization of Learning in Hybrid Educational Environments. *Front. Artif. Intell*. 2024. Vol. 7. P. 1458230. https://doi.org/10.3389/frai.2024.1458230.
9. Osakwe I., Chen G., Fan Y., Rakovic M., Li X., Singh S., Molenaar I., Bannert M., Gašević D. Reinforcement Learning for Automatic Detection of Effective Strategies for Self-Regulated Learning. *Computers and Education: Artificial Intelligence*. 2023. Vol. 5. P. 100181. https://doi.org/10.1016/j.caeai.2023.100181.
10. Axak N., Tatarnykov A., Kushnaryov M. Agent-based method of improving the efficiency of the e-learning. *Proceedings of the 12th International Scientific and Practical Conference Information Control Systems and Technologies*, ICST 2024. Vol. 3790. P. 63–75. URL: http://www.scopus.com/inward/record.url?eid=2-s2.0-85207827452&partnerID=MN8TOARS.

References

1. Bazzanella, E., & Santos, F. (2021). Does a Q-learning NetLogo extension simplify the development of agent-based simulations? In *Proceedings of the 15th Workshop-School on Agents, Environments, and Applications (WESAAC 2021),* 1–12. Porto Alegre. https://doi.org/10.5753/wesaac.2021.33403 [in English].
2. Bazzanella, E., Barros, M.M., & Santos, F. (2024). Refactoring the NetLogo reinforcement learning extension for integration with the BURLAP library. In *Proceedings of the 18th Workshop-School on Agents, Environments, and Applications (WESAAC 2024),* 51–62. Porto Alegre. https://doi.org/10.5753/wesaac.2024.33455 [in English].
3. Amzil, I., Aammou, S., Tagdimi, Z., & Erradi, H. (2023). Personalizing learning experiences with Q-learning in adaptive educational systems. In M. Khaldi (Ed.), *E-Learning and Smart Engineering Systems (ELSES 2023),* 70–77. Atlantis Press. https://doi.org/10.2991/978-94-6463-360-3_9 [in English].

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1

4. Dahan, A., Roth, N., Pelosi, A.D., & Reiner, M. (2024). A reinforcement learning framework for personalized adaptive e-learning. In *Advanced Technologies and the University of the Future,* 1140, 141–162. Lecture Notes in Networks and Systems. Springer. https://doi.org/10.1007/978-3-031-71530-3_10 [in English].
5. Mon, B.F., Wasfi, A., Hayajneh, M., Slim, A., & Abu Ali, N. (2023). Reinforcement learning in education: A literature review. *Informatics,* 10 (3), 74. https://doi.org/10.3390/informatics10030074 [in English].
6. Alharbi, K., Cristea, A.I., Shi, L., Tymms, P., & Brown, C. (2021). Agent-based simulation of the classroom environment to gauge the effect of inattentive or disruptive students. In I. I. Bittencourt et al. (Eds.), *Intelligent Tutoring Systems (ITS 2021),* 12677, 211–223. Lecture Notes in Computer Science. Springer. https://doi.org/10.1007/978-3-030-80421-3_23 [in English].
7. Alharbi, K., & Cristea, A.I. (2024). Hybrid agent-based machine learning simulation of a classroom disruption model. In D. Grzonka, N. Rylko, G. Suchacka, & V. Mityushev (Eds.), *Proceedings of the 38th ECMS International Conference on Modelling and Simulation,* 38 (1), 125–132 [in English].
8. Govea, J., Maldonado Navarro, A., Sánchez-Viteri, S., & Villegas-Ch., W. (2024). Implementation of deep reinforcement learning models for emotion detection and personalization of learning in hybrid educational environments. *Frontiers in Artificial Intelligence,* **7**, 1458230. https://doi.org/10.3389/frai.2024.1458230 [in English].
9. Osakwe, I., Chen, G., Fan, Y., Rakovic, M., Li, X., Singh, S., Molenaar, I., Bannert, M., & Gašević, D. (2023). Reinforcement learning for automatic detection of effective strategies for self-regulated learning. *Computers and Education: Artificial Intelligence,* 5, 100181. https://doi.org/10.1016/j.caeai.2023.100181 [in English].
10. Axak, N., Tatarnykov, A., & Kushnaryov, M. (2024). Agent-based method of improving the efficiency of the e-learning. In *Proceedings of the 12th International Scientific and Practical Conference "Information Control Systems and Technologies" (ICST 2024),* 3790, 63–75. http://www.scopus.com/inward/record.url?eid=2-s2.0-85207827452&partnerID=MN8TOARS [in English].

Аксак Наталія Георгіївна – д.т.н., професор, професор кафедри комп'ютерних інтелектуальних технологій та систем Харківського національного університету радіоелектроніки. E-mail: nataliia.axak@nure.ua, ORCID: 0000-0001-8372-8432.

Татарников Андрій Олександрович – аспірант кафедри комп'ютерних інтелектуальних технологій та систем Харківського національного університету радіоелектроніки. E-mail: andrii.tatarnykov@nure.ua, ORCID: 0000-0002-1632-8188.

Кушнарьов Максим Володимирович – к.т.н., старший викладач кафедри комп'ютерних інтелектуальних технологій та систем Харківського національного університету радіоелектроніки. E-mail: maksym.kushnarov@nure.ua, ORCID: 0000-0002-3772-3195.

Aksak Natalia Heorhiivna – Doctor of Technical Sciences, Professor, Professor at the Department of Computer Intellectual Technologies and Systems of the Kharkiv National University of Radio Electronics. E-mail: nataliia.axak@nure.ua, ORCID: 0000-0001-8372-8432.

Tatarnikov Andriy Oleksandrovych – Postgraduate Student at the Department of Computer Intellectual Technologies and Systems of the Kharkiv National University of Radio Electronics. E-mail: andrii.tatarnykov@nure.ua, ORCID: 0000-0002-1632-8188.

Kushnaryov Maxim Volodymyrovych – Candidate of Technical Sciences, Senior Lecturer at the Department of Computer Intellectual Technologies and Systems of the Kharkiv National University of Radio Electronics. E-mail: maksym.kushnarov@nure.ua, ORCID: 0000-0002-3772-3195.

https://doi.org/10.32782/mathematical-modelling/2025-8-1-1