

¹Артамонов Є.Б.,

orcid.org/0000-0002-9875-7372,

e-mail: eart@ukr.net,

²Головач Ю.Ю.,

orcid.org/0000-0001-9872-8144,

e-mail: iurii.golovach@gmail.com,

²Крант Д.В.,

orcid.org/0000-0003-4601-2827,

e-mail: danbiil988@gmail.com,

²Радченко К.М.,

orcid.org/0009-0009-8056-6293,

e-mail: radchenko.kn@gmail.com

ВИКОРИСТАННЯ АЛГОРИТМУ ЛЕВЕНШТЕЙНА ДЛЯ КАТЕГОРИЗАЦІЇ КОРИСТУВАЧІВ ІНФОРМАЦІЙНИХ СИСТЕМ

¹Національна академія СБУ²Національний авіаційний університет

Вступ

У сучасних інформаційних системах, особливо зі збільшенням обсягу даних і розмаїттям користувачів, ефективно розпізнавання та класифікація користувачів на основі їх поведінки стало критичним завданням. Класифікація користувачів забезпечує персоналізовану доставку контенту, адаптивні інтерфейси та підвищує безпеку шляхом виявлення аномальних дій або потенційних загроз. У системах з тисячами користувачів розуміння індивідуальних і групових моделей поведінки значно покращує роботу користувачів і загальну ефективність системи.

Однак проблема полягає в тому, що моделі поведінки можуть сильно відрізнятися навіть серед схожих груп користувачів. Існує потреба в алгоритмах, здатних ідентифікувати та вимірювати схожість між послідовностями взаємодії користувача в системі. Наприклад, різні користувачі можуть виконувати подібні дії в різній послідовності або з різною частотою, що ускладнює завдання визначення типових сценаріїв використання.

Для вирішення цієї проблеми можна використати алгоритм відстані Левенштейна, що оцінює подібності між різними послідовностями. Відстань Левенштейна, також відома як «відстань редагування»,

вимірює мінімальну кількість редагувань (вставок, видалень або замінів), необхідних для перетворення однієї послідовності в іншу.

Мета

Метою статті є дослідження застосування алгоритму відстані Левенштейна для класифікації користувачів інформаційних систем на основі їх поведінки. Цей підхід автоматизує групування користувачів за схожістю в їхніх шляхах використання, що може значно підвищити точність персоналізації, підвищити зручність використання інтерфейсу та забезпечити вищий рівень безпеки.

Вхідні дані для розрахунку

Експеримент проводився для *e-learning* систем, а дані категоризації користувачів використовувались для подальшого переключення сценаріїв представлення навчальних матеріалів.

Подібний підхід розглядався в роботах [1] і [2], де обговорюють організацію навчальних курсів, адаптованих до потреб і запитів користувачів. У статті [3] представлено метод побудови багатоагентної системи навчання з визначенням категорій користувачів.

Інтеграція динамічних інтерфейсів у системи електронного навчання являє собою трансформаційний підхід до сучасної

освіти [4]. Ці інтерфейси, які налаштовуються в режимі реального часу на основі поведінки та прогресу учня, пропонують персоналізований досвід навчання, який відповідає унікальним потребам і вподобанням кожного учня. Динамічний характер цих інтерфейсів підвищує взаємодію, оптимізує ефективність навчання та гарантує, що навчальний контент є актуальним і доступним [5].

Окрім того категоризація користувачів дозволяє переключати інтерфейси, реагуючи на взаємодію користувача з системою [6]. У контексті електронного навчання це означає, що система може змінювати представлення вмісту, шляхи навігації та інтерактивні елементи на основі того, як учень взаємодіє з матеріалом.

Модель представлення даних системи навчання можна представити наступним чином:

$$U = \{OP, SP\},$$

де,

$$OP = \{op_1, op_2, \dots, op_m\},$$

$$SP = \{sp_1, sp_2, \dots, sp_n\},$$

де, OP – сукупність об'єктивних параметрів, SP – сукупність суб'єктивних параметрів.

Систему електронного навчання можна принципово розділити на два основні набори параметрів:

- параметри інтерфейсу (IP): параметри, що охоплюють усі елементи, пов'язані з представленням і взаємодією навчальних ресурсів (дизайн макета, навігаційні структури, функції доступності та будь-які інші аспекти, які впливають на безпосередню взаємодію користувача з навчальним вмістом);

- функціональні параметри (FP): параметри, що пов'язані з основними функціями системи електронного навчання (механізми доставки контенту, відстеження продуктивності, системи зворотного зв'язку та будь-які інші операційні можливості, які підтримують процес навчання).

$$ESE = \{IP, FP\}$$

де,

$$IP = \{ip_1, ip_2, \dots, ip_k\},$$

$$FP = \{fp_1, fp_2, \dots, fp_l\},$$

де, IP – набір параметрів інтерфейсу, FP – сукупність функціональних параметрів.

В межах описаної моделі всі переходи між об'єктами системи можна подавати у вигляді рядку переходів (послідовностей), який включає об'єкти, які було задіяно при відображенні.

В інформаційній системі навчання взаємодії користувача можна представляти, як відвідування сторінок, натискання кнопок або виконані дії. Ці взаємодії можна перетворити на послідовності, які відображають шлях користувача через систему. Кожна унікальна взаємодія представлена у вигляді символу або маркера, створюючи «рядок» поведінки користувача. Наприклад, відвідування сторінки може бути записано як $P1$, натискання кнопки як $B1$, а надсилання форми як $F1$. Тоді сеанс користувача може бути представлений у вигляді послідовності $P1-B1-P2-F1$.

Алгоритм відстані Левенштейна, який часто називають «відстанню редагування», визначає подібність між двома послідовностями шляхом обчислення мінімальної кількості редагувань одного символу, необхідних для перетворення однієї послідовності в іншу [7]. Ці редагування включають три основні операції: вставку (додавання символу), видалення (видалення символу) і заміну (заміна одного символу іншим).

Рекурсивна структура алгоритму відстані Левенштейна робить його обчислювально-інтенсивним для довгих послідовностей, але його можна ефективно реалізувати за допомогою динамічного програмування, щоб уникнути надлишкових обчислень, зберігаючи проміжні результати в матриці.

Порівнюючи ці послідовності з відстанню Левенштейна, ми можемо оцінити, наскільки схожі шляхи користувачів. Це дозволяє віднести користувачів до тієї або

іншої категорії. Наприклад, якщо два користувачі мають невелику відстань Левенштейна між своїми послідовностями, це означає, що їхні шляхи взаємодії подібні, вказуючи на спільні цілі, рівні навичок або вподобання. Це групування на основі подібності забезпечує цільову персоналізацію та прогнозне моделювання в системі [8].

Відстань Левенштейна між двома рядками – це мінімальна кількість редагувань одного символу (вставка, видалення, заміна), необхідних для перетворення одного рядка в інший [9]. Формально він визначається рекурсивно наступним чином:

$$D(i, j) = \begin{cases} \max(i, j), & \text{if } \min(i, j) = 0, \\ \min \begin{cases} D(i-1, j) + 1 \\ D(i, j-1) + 1 \\ D(i-1, j-1) + [s[i-1] \neq t[i-1]] \end{cases}, & \text{otherwise,} \end{cases}$$

де, $D(i, j)$ – це відстань Левенштейна між першими i символами рядка s і першим j символом рядка t .

Щоб ефективно застосувати відстань Левенштейна, спочатку потрібно зібрати дані про взаємодію з користувачем і перетворити їх на послідовності, придатні для аналізу.

Дані про взаємодію користувача збираються за допомогою механізмів журналювання в системі. Кожна взаємодія позначається часом і зберігається. Ці дані фіксують порядок і частоту дій, забезпечуючи повне уявлення про поведінку користувача.

Необроблені дані взаємодії часто містять шум (випадкові клацання чи повторювані дії, які не сприяють суттєвому розумінню поведінки користувача). Попередня обробка може включати фільтрування цих нерелевантних взаємодій, консолідацію повторюваних дій або кодування складних дій як окремих маркерів.

Після попередньої обробки кожен сеанс користувача перетворюється на послідовність токенів, які представляють виконані дії. Наприклад, $P1-B1-P2-F1$ може

позначати серію кроків, під час яких користувач відвідав сторінку 1, натиснув кнопку 1, відвідав сторінку 2 і надіслав форму.

Кожному унікальному типу взаємодії присвоюється символ, що забезпечує ефективне порівняння між різними послідовностями користувачів.

Коли сеанси користувача кодуються як послідовності, відстань Левенштейна застосовується до кожної пари послідовностей, щоб виміряти їхню подібність. Використовуючи матрицю динамічного програмування, можна розрахувати відстань між послідовністю кожного користувача та іншими. Користувачі зі схожими послідовностями ідентифікуються як належні до однієї поведінкової групи.

На основі цих відстаней можна застосувати методи кластеризації для формування окремих груп.

Приклад розрахунку відстані Левенштейна для різних шляхів використання системи

Розглянемо неорієнтований зв'язний граф з 10 вершинами, представлений матрицею суміжності. Графік має таку структуру:

	0	1	2	3	4	5	6	7	8	9
0	0	1	1	1	0	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0	0
2	1	0	0	0	0	0	1	0	0	0
3	1	0	0	0	0	0	0	1	1	0
4	0	1	0	0	0	0	0	0	0	1
5	0	1	0	0	0	0	0	0	0	1
6	0	0	1	0	0	0	0	0	0	1
7	0	0	0	1	0	0	0	0	0	1
8	0	0	0	1	0	0	0	0	0	1
9	0	0	0	0	1	1	1	1	1	0

Для прикладу розглянемо шляхи «0149», «0379», «159», «269», «0269», «389».

Приклад розрахунку відстані Левенштейна між шляхами 1 і 2 («0149» і «0379») мають кілька кроків.

Крок 1. Ініціалізація: ми починаємо з ініціалізації матриці динамічного програмування D з розмірами $(n + 1) \times (m + 1)$,

де n – довжина першого рядка і t це довжина другого рядка. У цьому випадку обидва рядки мають довжину 4, тому матриця буде 5×5 .

$$\begin{pmatrix} & 0 & 3 & 7 & 9 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 0 & 0 \\ 4 & 3 & 0 & 0 & 0 \\ 9 & 4 & 0 & 0 & 0 \end{pmatrix}$$

Крок 2. Заповнення матриці: заповнюємо матрицю D шляхом розрахунку мінімальної вартості операцій редагування (вставлення, видалення, заміни), необхідних для перетворення підрядка першого рядка в підрядок другого рядка. Проходимо клітинку за клітинкою:

- $D(1, 1)$ порівнює "0" з "0", які однакові, тому вартість дорівнює 0:

$$D(1, 1) = \min(D(0, 1) + 1, D(1, 0) + 1, D(0, 0) + 0) = \min(2, 2, 0) = 0;$$

- $D(1, 2)$ порівнює «0» з «03», вартість дорівнює 1:

$$D(1, 2) = \min(D(0, 2) + 1, D(1, 1) + 1, D(0, 1) + 1) = \min(3, 1, 2) = 1;$$

- $D(1, 3)$ порівнює «0» з «037», вартість дорівнює 2:

$$D(1, 3) = \min(D(0, 3) + 1, D(1, 2) + 1, D(0, 2) + 1) = \min(4, 2, 3) = 2;$$

- $D(1, 4)$ порівнює «0» з «0379», вартість дорівнює 3.

$$D(1, 4) = \min(D(0, 4) + 1, D(1, 3) + 1, D(0, 3) + 1) = \min(5, 3, 4) = 3;$$

- $D(2, 1)$ порівнює "01" з "0", вартість дорівнює 1:

$$D(2, 1) = \min(D(1, 1) + 1, D(2, 0) + 1, D(1, 1) + 1) = \min(1, 3, 2) = 1;$$

- $D(2, 2)$ порівнює "01" з "03", вартість дорівнює 1:

$$D(2, 2) = \min(D(1, 2) + 1, D(2, 1) + 1, D(1, 1) + 1) = \min(2, 2, 1) = 1;$$

- $D(2, 3)$ порівнює "01" з "037", вартість дорівнює 2:

$$D(2, 3) = \min(D(1, 3) + 1, D(2, 2) + 1, D(1, 2) + 1) = \min(3, 2, 2) = 2;$$

- $D(2, 4)$ порівнює "01" з "0379", вартість дорівнює 3:

$$D(2, 4) = \min(D(1, 4) + 1, D(2, 3) + 1, D(1, 3) + 1) = \min(4, 3, 3) = 3;$$

- $D(3, 1)$ порівнює «014» з «0», вартість дорівнює 2:

$$D(3, 1) = \min(D(2, 1) + 1, D(3, 0) + 1, D(2, 0) + 1) = \min(2, 4, 3) = 2;$$

- $D(3, 2)$ порівнює "014" з "03", вартість дорівнює 2:

$$D(3, 2) = \min(D(2, 2) + 1, D(3, 1) + 1, D(2, 1) + 1) = \min(2, 3, 2) = 2;$$

- $D(3, 3)$ порівнює "014" з "037", вартість дорівнює 2:

$$D(3, 3) = \min(D(2, 3) + 1, D(3, 2) + 1, D(2, 2) + 1) = \min(3, 3, 2) = 2;$$

- $D(3, 4)$ порівнює "014" з "0379", вартість дорівнює 3:

$$D(3, 4) = \min(D(2, 4) + 1, D(3, 3) + 1, D(2, 3) + 1) = \min(4, 3, 3) = 3;$$

- $D(4, 1)$ порівнює "0149" з "0", вартість дорівнює 3:

$$D(4, 1) = \min(D(3, 1) + 1, D(4, 0) + 1, D(3, 0) + 1) = \min(3, 5, 4) = 3;$$

- $D(4, 2)$ порівнює "0149" з "03", вартість дорівнює 3:

$$D(4, 2) = \min(D(3, 2) + 1, D(4, 1) + 1, D(3, 1) + 1) = \min(3, 4, 3) = 3;$$

- $D(4, 3)$ порівнює "0149" з "037", вартість дорівнює 3:

$$D(4, 3) = \min(D(3, 3) + 1, D(4, 2) + 1, D(3, 2) + 1) = \min(3, 4, 3) = 3;$$

- $D(4, 4)$ порівнює "0149" з "0379", вартість дорівнює 2:

$$D(4, 4) = \min(D(3, 4) + 1, D(4, 3) + 1, D(3, 3) + 0) = \min(4, 4, 2) = 2.$$

Остаточна матриця має наступний вигляд:

$$\begin{pmatrix} & 0 & 3 & 7 & 9 \\ & 0 & 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 1 & 2 & 3 \\ 1 & 2 & 1 & 1 & 2 & 3 \\ 4 & 3 & 2 & 2 & 2 & 3 \\ 9 & 4 & 3 & 3 & 3 & 2 \end{pmatrix}$$

Кінцева клітинка $D(4,4) = 2$ вказує на те, що відстань Левенштейна між «0149» і «0379» дорівнює 2.

Усереднені розраховані відстані між послідовностями користувача і шаблоном користувачем категорії дозволяють

віднести користувача до тієї або іншої категорії.

Результати категоризації користувачів інформаційних систем

Дане дослідження є продовженням досліджень роботи [10], де досліджувались адаптивні інтерфейси систем і методи переключення між ними на основі аналізу взаємодії користувачів з системою. Використовувалися дані про автоматичне перемикання категорій в системі онлайн-навчання *MS SQL Server* і в апаратно-програмному комплексі навчання Брайля. Дані експерименту у порівнянні з минулою

роботою розширено до 145 слухачів курсу з вивчення *MS SQL Server* (навчання проходило 13 днів (табл. 1), число змін категорій на кінець дня представлено на рис. 1, число зафіксованих категорій на кінець кожного дня представлено на рис. 2) та до 170 користувачів апаратно-програмного комплексу для вивчення шрифту Брайля (навчання проходило протягом 23 дня (табл. 2), число змін категорій на кінець дня представлено на рис. 3, число зафіксованих категорій на кінець кожного дня представлено на рис. 4).

Таблиця 1. Категорії слухачів в системі електронного навчання для вивчення *MS SQL Server*

День	Категорія 1 на початку дня	Зміна категорії (кінець дня)	Категорія 2 на початку дня	Зміна категорії (кінець дня)
1	85	9	15	5
2	81	8	19	4
3	77	5	23	3
4	75	8	25	2
5	69	7	31	3
6	65	4	35	1
7	62	3	38	1
8	60	7	40	1
9	54	8	46	1
10	47	7	53	2
11	42	6	58	1
12	37	5	63	0
13	32	4	68	0
Результат	28		72	

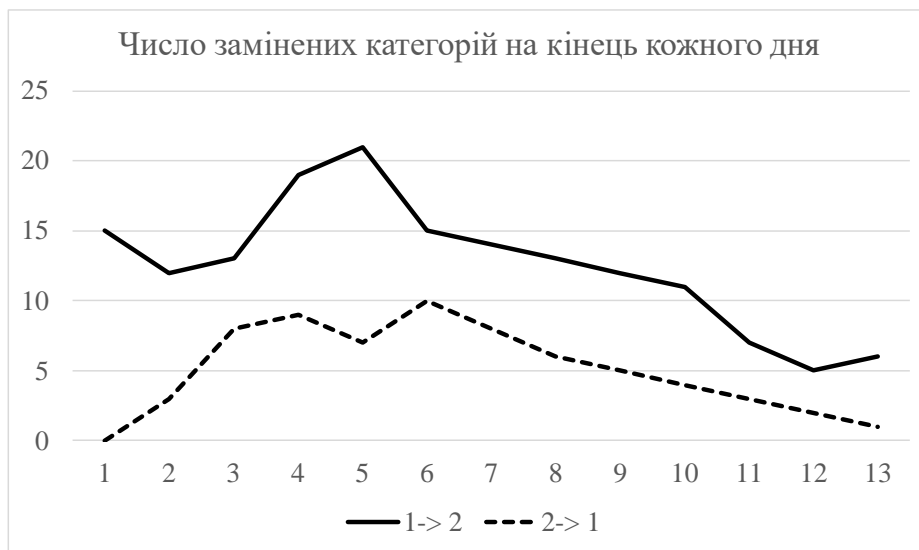


Рис. 1. Графіки змін категорій слухачів для курсу з вивчення *MS SQL Server*

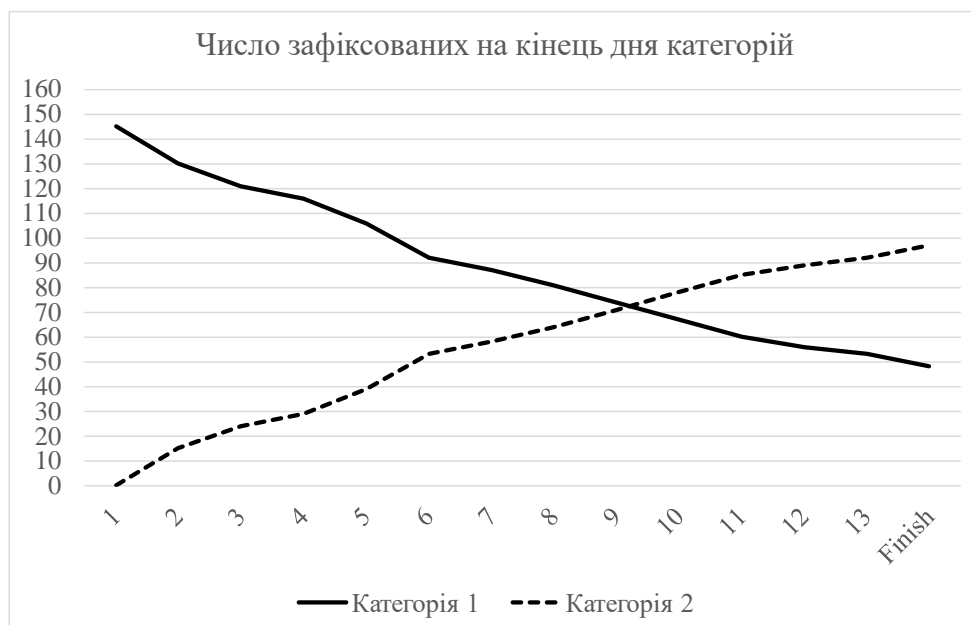


Рис. 2. Зафіксовані категорії слухачів для курсу з вивчення *MS SQL Server* на кінець кожного дня

Таблиця 2. Категорії користувачів в апаратно-програмному комплексі навчання Брайля

Урок	Категорія 1 (початок уроку)	Зміна категорії (кінець уроку)		Категорія 2 (початок уроку)	Зміна категорії (кінець уроку)		Категорія 3 (початок уроку)	Зміна категорії (кінець уроку)	
		1 до 2	1 до 3		2 до 1	2 до 3		3 до 1	3 до 2
1	170	8	6	0	0	0	0	0	0
2	156	5	4	8	3	2	6	2	1
3	152	6	3	9	4	2	9	4	3
4	151	7	2	12	3	3	7	4	3
5	149	5	3	16	3	5	5	5	2
6	149	5	2	15	2	3	6	3	4
7	147	7	1	19	4	5	4	5	2
8	148	8	3	19	5	4	3	3	3
9	145	4	3	21	5	3	4	2	3
10	145	5	2	20	3	3	5	3	1
11	144	5	4	20	3	4	6	2	1
12	140	6	3	19	2	5	11	2	2
13	135	8	2	20	2	2	15	2	1
14	129	9	3	25	2	3	16	2	1
15	121	8	4	30	1	2	19	2	1
16	112	8	4	36	1	3	22	2	1
17	103	7	3	41	3	4	26	1	2
18	97	9	2	43	1	4	30	0	2
19	87	9	2	49	0	5	34	1	3
20	77	8	4	56	2	4	37	0	1
21	67	7	2	59	1	2	44	1	1
22	60	8	2	64	2	3	46	1	1
23	53	8	2	68	2	2	49	0	1
Результат	45			73			52		



Рис. 3. Графіки змін категорій користувачів апаратно-програмного комплексу для вивчення шрифту Брайля

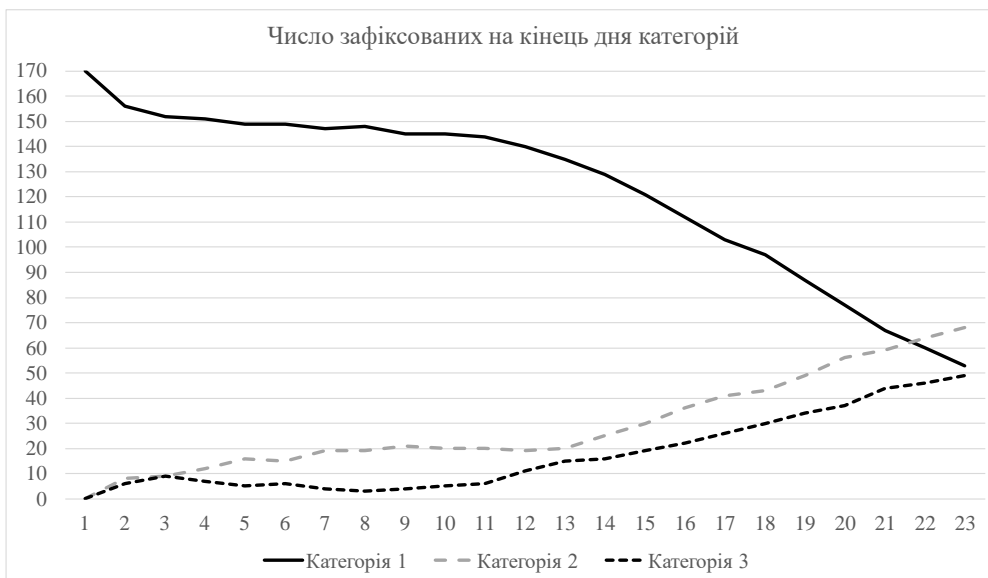


Рис. 4. Зафіксовані категорії на кінець кожного дня користувачів апаратно-програмного комплексу для вивчення шрифту Брайля

Аналіз змін категорій для слухачів для курсу з вивчення *MS SQL Server* показав наступне:

- кількість учасників з Категорією 1 з часом зменшується, тоді як Категорія 2 збільшується, що вказує на те, що з просуванням курсу все більше студентів переходить від початкового сценарію до більш складного. Це свідчить про те, що слухачі успішно просуваються за матеріалами курсу та залучаються до дедалі складнішого матеріалу;
- більша кількість переходів до Категорії 2 вказує на те, що слухачі

загалом прогресують, а не регресують, що відображає ефективність навчання та адаптацію до більш складного змісту, що демонструє успішну криву навчання, коли студенти добре адаптуються до матеріалу курсу та рухаються вперед, як очікувалося (рис. 2).

Аналіз змін категорій користувачів апаратно-програмного комплексу для вивчення шрифту Брайля:

- в системі навчання шрифтом Брайля кількість користувачів категорії 1 постійно зменшується, тоді як кількість користувачів категорій 2 і 3 зростає. Ця

тенденція вказує на те, що користувачі просуються навчальним матеріалом з різною швидкістю;

- часті зміни між категоріями (наприклад, 1->2, 2->1, 1->3) ілюструють чутливість системи до проходження користувачами занять;
- дані показують чіткий шлях прогресу, де більше користувачів з часом переходять до вищих категорій. Це відображає ефективність адаптивної системи у сприянні постійному навчанню та розвитку навичок Брайля.
- остаточні результати демонструють збалансований розподіл між категоріями, що свідчить про те, що система ефективно керує прогресом користувачів і гарантує, що всі отримують відповідні завдання (рис. 4).

Висновки

Застосування алгоритму відстані Левенштейна для категоризації користувачів в адаптивних системах електронного навчання доводить ефективність у покращенні персоналізованого досвіду шляхом аналізу моделей взаємодії користувача з системою.

Алгоритм відстані Левенштейна виявився достатньо ефективним у класифікації користувачів у рамках адаптивних інформаційних систем за рахунок вимірювання подібності поведінкових шляхів. Його унікальна здатність фіксувати як послідовність, так і вміст взаємодії користувача забезпечує точнішу класифікацію та персоналізацію, що підтримує динамічну адаптацію вмісту. Розраховуючи мінімальну кількість редагувань, необхідних для узгодження однієї послідовності з іншою, алгоритм дозволяє точно розрізнити поведінку користувачів, навіть якщо існують незначні варіації в їхніх шляхах взаємодії. Ця можливість робить його ефективним інструментом для класифікації користувачів на основі тонких поведінкових нюансів, сприяючи адаптивній реакції в системі. Крім того, ефективність відстані Левенштейна в обробці даних на основі послідовності дозволяє застосовувати в реальному часі середовища з великими

наборами даних, наприклад системи електронного навчання.

У курсі *MS SQL Server* перехід від категорій для початківців до категорій для просунутих продемонстрував успішну адаптацію до прогресивно складного матеріалу з чітким узгодженням між прогресом користувача та розробленим шляхом навчання. Подібним чином у системі навчання шрифтом Брайля користувачі постійно переходили від нижчих до вищих категорій із більш частими коригуваннями, що відображає взаємодію користувачів з матеріалом і різною швидкістю навчання.

Отримані дані підкреслюють можливість застосування різних методів поведінкової категоризації в адаптивних системах, закладаючи основу для подальшого розвитку персоналізованого електронного навчання.

Література

1. Chatzopoulou D. I., Economides A. A. Adaptive assessment of student's knowledge in programming courses. *Journal of Computer Assisted Learning*. 2010. Vol. 26, no. 4. P. 258–269. DOI: 10.1111/j.1365-2729.2010.00363.x.
2. Melesko J, Kurilovas E. Personalised intelligent multi-agent learning system for engineering courses. *2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE) : proceedings, Vilnius, Lithuania, 10–12 November 2016 / IEEE*. 2016. P. 1–6. DOI: 10.1109/AIEEE.2016.7821821.
3. Kristensen T, Dyngeland M. Design and Development of a Multi-Agent E-Learning System. *International Journal of Agent Technologies and Systems*. 2015. Vol. 7(2). P. 19–74. DOI: 10.4018/IJATS.2015040102
4. Artamonov E. B, Zholdakov O. O. Concept of creating a software environment for automated text manipulation. *Scientific journal "Proceedings of the National Aviation University"*. 2010. Vol. 3(44). P. 111–115.
5. Jimaa S. The impact of assessment on students learning. *Procedia-Social and Behavioral Sciences*. 2011. Vol. 28. P. 718–721.

6. Abid Masood Khan, Hifsa Batool, Saleha Ashraf. A comparative study to evaluate assessment facilities at government special education schools. *Pakistan Journal of Humanities and Social Sciences Research*. 2020. Vol. 02(01). URL: <https://www.researchgate.net/publication/344287807>.
7. Backurs A, Indyk P. Edit Distance Cannot Be Computed in Strongly Subquadratic Time (unless SETH is false). *SIAM J Comput*. 2018. Vol. 47(3). P. 1087–97.
8. Zorita E., Cusc P., Fillion G.J. Starcode: Sequence clustering based on all-pairs search. *Bioinformatics*. 2015. Vol. 31(12). P. 1913–1919.
9. Yujian L, Bo L. A normalized Levenshtein distance metric. *IEEE Trans Pattern Anal Mach Intell*. 2007. Vol. 29(6). P. 1091–1095.
10. Artamonov Y. et al. Dynamic Content Generation Methods Based on User Behavioral Ranking. *2022 IEEE 4th International Conference on Advanced Trends in Information Theory (ATIT)* : proceedings, Kyiv, Ukraine, 15–17 December 2022 / IEEE. 2022. P. 313–318. DOI: 10.1109/ATIT58178.2022.10024196.

Артамонов Є.Б., Головач Ю.Ю., Крант Д.В., Радченко К.М.

ВИКОРИСТАННЯ АЛГОРИТМУ ЛЕВЕНШТЕЙНА ДЛЯ КАТЕГОРИЗАЦІЇ КОРИСТУВАЧІВ ІНФОРМАЦІЙНИХ СИСТЕМ

Дослідження присвячено застосуванню алгоритму відстані Левенштейна для категоризації користувачів на основі поведінки в інформаційних системах, з акцентом на динамічну адаптацію в середовищах електронного навчання. В експерименті використовувалися дані онлайн-навчання курсу MS SQL Server і апаратно-програмного комплексу навчання шрифту Брайля, що дозволяє перемикає вміст на основі сценаріїв, які відповідають категорії користувачів. Результати показують ефективність використання алгоритму відстані Левенштейна для категоризації користувачів та навчання зберегло позитивну тенденцію просування за заздалегідь визначеними сценаріями навчання. Дослідження також підкреслює потенціал адаптивних систем електронного навчання для покращення результатів навчання шляхом адаптації контенту до індивідуальних потреб користувачів.

Ключові слова: відстань Левенштейна; категоризація користувачів; адаптивні системи; поведінка користувачів; схожість послідовностей.

Artamonov Y.B., Golovach I.I., Krant D.V., Radchenko K.M.

USING THE LEVENSHTAIN ALGORITHM FOR THE CATEGORIZATION OF INFORMATION SYSTEMS USERS

The study is devoted to the application of the Levenshtein distance algorithm to categorize users based on behavior in information systems, with an emphasis on dynamic adaptation in e-learning environments. The experiment used data from the MS SQL Server online course and the hardware and software complex for teaching the Braille font, which allows you to switch the content based on scenarios that correspond to the category of users. The results show the effectiveness of using the Levenshtein distance algorithm for user categorization and training has maintained a positive trend of advancement according to predefined training scenarios. The study also highlights the potential of adaptive e-learning systems to improve learning outcomes by adapting content to the individual needs of users.

Keywords: Levenshtein distance; user categorization; adaptive systems; user behavior; sequence similarity.