

*Л. В. БОДНАР, К. С. ШУЛАКОВА, Л. Е. ГРИЗУН*

### **АЛГОРИТМІЧНЕ ЗАБЕЗПЕЧЕННЯ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ВЕБСЕРВІСУ ДЛЯ ВИВЧЕННЯ ІНОЗЕМНИХ МОВ**

Дана робота присвячена аналізу алгоритмічного забезпечення рекомендаційних систем мультимедійного контенту та розробці вебсервісу для підвищення ефективності вивчення іноземних мов за допомогою рекомендаційної системи, яка персоналізовано підбирає навчальний контент для користувача. Для формування переліку необхідного мультимедійного контенту були обрані основні критерії рекомендаційної системи, виявлені основні потреби користувачів, які повинна вирішувати система, оскільки підвищення ефективності вивчення іноземної мови досягається не тільки шляхом вибору методів навчання, але й переглядом мультимедійного контенту, а саме новин, фільмів, навчальних роликів, кліпів та ін. З боку методу реалізації алгоритмічного забезпечення були розглянуті різні види фільтрації даних від сучасних технічних методів до бібліотек для забезпечення функціональності системи та обрано алгоритм на основі гібридної фільтрації, при якому використовуються відомі оцінки користувачів для прогнозування уподобань іншого користувача. Розроблені функціональні вимоги та запропоновано вебсервіс, який уможливило комплексний вплив на навчання користувача при вивченні іноземної мови, програмна реалізація якого виконана за допомогою мов Java Script, Python та додаткових бібліотек. Ця реалізація дозволяє побудувати процес відстеження змін стосовно вимог користувача та передавати інформацію в базу даних (БД) і після аналізу вхідних даних змінювати запропонований мультимедійний контент користувачу. Під час подальших досліджень заплановано проведення практичних експериментів з урахуванням специфіки тих чи інших методів навчання іноземних мов і застосування статистичних даних для оцінки ефективності роботи алгоритму запропонованої рекомендаційної системи.

**Ключові слова:** алгоритм, рекомендаційна система, гібридна фільтрація, БД, вебсервіс, Python, фільтрація за змістом, колаборативна фільтрація.

*Л. В. БОДНАР, Е. С. ШУЛАКОВА, Л. Э. ГРИЗУН*

### **АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СИСТЕМЫ РЕКОМЕНДАЦИИ ВЕБ-СЕРВИСА ДЛЯ ИЗУЧЕНИЯ ИНОСТРАННЫХ ЯЗЫКОВ**

Данная работа посвящена анализу алгоритмического обеспечения рекомендательных систем мультимедийного контента и разработке веб-сервиса для повышения эффективности изучения иностранных языков с помощью рекомендательной системы, которая персонализированно подбирает обучающий контент для пользователя. Для формирования перечня необходимого мультимедийного контента были выбраны основные критерии рекомендательной системы, выявлены основные потребности пользователей, которые должна решать система, поскольку повышение эффективности изучения иностранного языка достигается не только путем выбора методов обучения, но и просмотром мультимедийного контента, а именно новостей, фильмов, обучающих роликов, клипов и т.д. Со стороны метода реализации алгоритмического обеспечения были рассмотрены различные виды фильтрации данных от современных технических методов до библиотек для обеспечения функциональности системы и выбран алгоритм на основе гибридной фильтрации, при котором используются известные оценки пользователей для прогнозирования предпочтений другого пользователя. Разработаны функциональные требования и предложен веб-сервис, позволяющий комплексно воздействовать на обучение пользователя при изучении иностранного языка, программная реализация которого выполнена с помощью языков Java Script, Python и дополнительных библиотек. Эта реализация позволяет построить процесс отслеживания изменений в пользовательских требованиях и передавать информацию в базу данных (БД) и после анализа входных данных изменять предлагаемый мультимедийный контент пользователю. В ходе дальнейших исследований запланировано проведение практических экспериментов с учетом специфики тех или иных методов обучения иностранных языков и применения статистических данных для оценки эффективности работы алгоритма предлагаемой рекомендательной системы.

**Ключевые слова:** алгоритм, рекомендательная система, гибридная фильтрация, БД, веб-сервис, Python, фильтрация по содержанию, колаборативная фильтрация.

*L. V. BODNAR, K. S. SHULAKOVA, L. E. GRYZUN*

### **ALGORITHMIC SUPPORT OF THE WEB SERVICE RECOMMENDATION SYSTEM FOR LEARNING FOREIGN LANGUAGES**

This work is devoted to the analysis of algorithmic support of multimedia content recommender systems and the development of a web service to increase the efficiency of learning foreign languages using a recommender system that personalized the selection of educational content for the user. To form a list of necessary multimedia content, the main criteria of the recommender system were selected, the basic needs of users were identified, which the system should solve, since increasing the efficiency of learning a foreign language is achieved not only by choosing teaching methods, but also by watching multimedia content, namely news, films, educational videos, clips, etc. Therefore, in order to form a list of the necessary multimedia content, the main criteria of the recommender system were formed, the main needs of users were identified, which the system must solve. From the side of the method for implementing algorithmic support, various types of data filtering were considered, from modern technical methods to libraries to ensure the functionality of the system, and the algorithm based on hybrid filtering was chosen, in which known user ratings are used to predict the preferences of another user. Functional requirements have been developed and a web service has been proposed that allows a comprehensive impact on user learning when learning a foreign language, software implementation of which is made using Java Script, Python and additional libraries. This implementation allows you to build a process for tracking changes in user requirements and transfer information to the database (DB) and, after analyzing the input data, change the proposed multimedia content to the user. In the course of further research, it is planned to conduct practical experiments, taking into account the specifics of certain methods of teaching foreign languages and the use of statistical data to assess the effectiveness of the algorithm of the proposed recommender system.

**Keywords:** algorithm, recommender system, hybrid filtering, DB, web service, Python, content-based filtering, collaborative filtering.

**Вступ.** Багато найбільших комерційних вебсайтів вже тривалий час використовують рекомендаційні системи, щоб допомогти своїм користувачам знайти необхідний контент. Спільна фільтрація, яка зазвичай

використовується в сценаріях електронної комерції, виявляє взаємодії між користувачами та елементами і являє собою алгоритм рекомендаційної системи. На основі цієї фільтрації передбачаються уподобання

© Л. В. Боднар, К. С. Шулакова, Л. Е. Гризун, 2021

користувачів, аналізуючи їх поведінку в минулому, згідно з якою рекомендаційна система перенавчає математичну модель та пропонує нові списки контенту [1], допомагаючи тим самим користувачеві знайти необхідну інформацію.

**Постановка проблеми.** Сьогодні налічують близько 6000 іноземних мов [2]. За статистикою, у світі 1,5 млрд людей вивчають англійську мову [3], але далеко не у всіх є можливість займатися з викладачем офлайн, тому вони використовують різноманітні вебсервіси. Крім того, існують дослідження [4], які показують, що використання вебсервісів дозволяє підвищити ефективність вивчення іноземної мови.

Оскільки кількість різних інструментів досить велика, і скласти свій стек освітніх технологій може бути нелегко, тому мета даної роботи - розробка вебсервісу з алгоритмічним забезпеченням рекомендаційної системи, яка формує списки необхідного навчального контенту індивідуально під потреби кожного користувача.

Для досягнення мети необхідно було розв'язати наступні завдання:

- розробити принцип роботи вебсервісу;
- визначити рівні доступу користувачів вебсервісу;
- обрати алгоритмічне забезпечення рекомендаційної системи вебсервісу;
- обрати моделі даних для реалізації бази даних;
- виконати проектування та розробку бази даних;
- розробити сервіс регулювання навантаження;
- інтегрувати кешування даних і результатів запитів;
- розробити вебсервіс та налаштувати серверну частину.

Змістова постановка задачі представлена у вигляді IDEF0-діаграми (рис. 1).

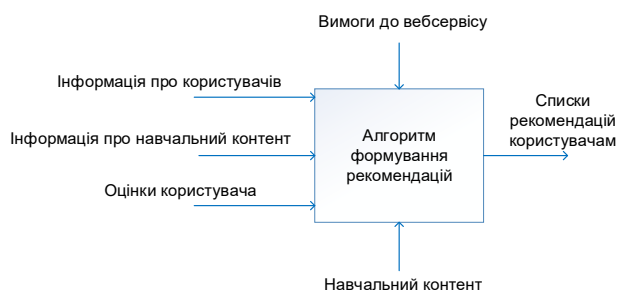


Рис. 1. IDEF0-діаграма

**Аналіз стану проблеми.** Досить часто плани створення рекомендаційної системи та БД готуються і оцінюються на основі думок замовників і розробників про необхідні функції та можливі якості програмного забезпечення. У зв'язку з цим, помилки при визначенні необхідних показників прогнозування, оцінки трудомісткості, вартості та тривалості створення рекомендаційної системи є поширеним явищем.

Часто рекомендаційні системи не відповідають вимогам до швидкого та точного підбору контенту,

оскільки реалізація цього типу алгоритмів дуже складна, її важко підтримувати, так як навіть незначні коригування в роботі призводять до змін алгоритму. Тому при розробці вебсервісу перш за все необхідно продумати логіку роботи, визначити чітке алгоритмічне забезпечення рекомендаційної системи мультимедійного контенту та обрати моделі даних для реалізації бази даних, що підходить саме цьому вебсервісу.

**Аналіз існуючих алгоритмів.** Зазвичай математичні алгоритми рекомендаційних систем сформовані за чотирма критеріями [5]:

- кількість рекомендованого контенту;
- кількість інформації про рекомендований контент для уточнення потреб користувача;
- додаткова інформація про користувача для конкретизації підбору рекомендацій;
- ергономічність сервісу для зручності відвідувача (чим комфортніше, тим більше інформації можна отримати про його реакції на поради рекомендаційної системи).

На сьогоднішній день найбільш поширені три алгоритми рекомендаційних систем:

- на основі спільної (колаборативної) фільтрації [6], де прогнози складаються індивідуально для кожного користувача, хоча використовується інформація зібрана від багатьох учасників внаслідок чого алгоритм погано масштабується;
- на основі фільтрації змісту (content-based) [7], в якому ключові слова використовуються для опису елементів, а профіль користувача створюється для визначення типу елемента, який йому подобається;
- на основі фільтрації знань [8], які засновані на явних знаннях, користувацьких перевагах і критеріях рекомендацій.

Аналізуючи роботу алгоритмів, колаборативна фільтрація надає рекомендації тільки досвідченим користувачам, які вже взаємодіяли з системою, так як вона рекомендує на основі їх уподобань, що робить неможливим генерувати рекомендації для нових користувачів. Алгоритм фільтрації за змістом надає рекомендації новим та досвідченим користувачам на основі ключових слів, які користувач визначив для опису елементів. Оскільки цей підхід залежить від якості опису метаданих елементів та користувачів, це приносить багато обмежень системі. Алгоритм фільтрації знань потребує додаткового джерела інформації, а також явних знань про персональні вимоги користувачів, але вирішує проблеми холодного старту.

**Аналіз існуючих рекомендаційних систем.** На сьогоднішній день існують дуже багато онлайн ресурсів з вивчення іноземних мов від вебсайтів до вебдодатків на мобільні дивайси, але відсутні рекомендаційні системи, які можуть допомогти підібрати контент не тільки для вивчення іноземної мови під персональні потреби, але при навчанні в цілому, що робить неможливим зробити аналіз існуючих рекомендаційних систем навчального контенту. Проте існують подібні розважальні вебсервіси з рекомендаційними системами мультимедійного контенту, які

використовуються в основному у вигляді відеохостингу: YouTube [9], Netflix [10], MEGOGO [11], Ivi [12] та інші.

Для спрощення збору інформації щодо відгуків та запитів Netflix використовує колаборативну фільтрацію, проте велика кількість даних приводить до зміщення результатів рекомендаційної системи і зниження її ефективності. Рекомендаційні системи MEGOGO та Ivi, що працюють на основі неявних призначених для користувача оцінок, тягнуть за собою проблему дотримання конфіденційності користувачів, захисту персональних даних. Крім цього, слід виділити проблему холодного старту по відношенню до користувача (коли невідомо що рекомендувати щойно зареєстрованій людині) та по відношенню до контенту (у випадку невизначеності щодо того, кому можна порекомендувати щойно створений контент) [13]. Система рекомендацій YouTube використовує кілька цільових функцій для ранжування і враховує особисті переваги користувача за рахунок алгоритму SVD [14], який базується на колаборативній фільтрації та фільтрації за змістом, тому алгоритм має менше недоліків, ніж вище вказані алгоритми окремо. Саме такий гібридний алгоритм буде застосовано для вирішення завдань рекомендаційної системи навчального контенту, який обрано при розробці даного вебсервісу. Основною проблемою залишається холодний старт по відношенню до контенту, оскільки не можна навчити нейронну мережу коректно передбачати оцінки, не маючи тестової вибірки, але інші завдання даного алгоритму вирішує.

**Алгоритмічне забезпечення рекомендаційної системи вебсервісу.** Гібридний алгоритм складається з випадкового і  $kNN$  алгоритмів. Випадковий алгоритм використовується для вирішення проблеми холодного старту по відношенню до користувача і збереження різноманітності рекомендаційної системи.  $kNN$  алгоритм знаходить  $k$  аналогічних користувачів за певною кількістю переглянутих елементів. Нехай перегляд елементів ( $TR$ ) буде порогом рекомендацій  $kNN$ , який визначає чи використовувати алгоритм  $kNN$ , тоді кількість переглянутих користувачем  $u_i$  елементів становить  $|br(u_i)|$  та є перемикачем  $kNN$ , який задається

$$kNN\ switch = \begin{cases} true, & \text{if } |br(u_i)| \geq TR, \\ false, & \text{otherwise.} \end{cases}$$

На стабільній стадії кількість переглянутих елементів  $TR$  достатня для виконання алгоритму  $kNN$ , щоб рекомендувати необхідну кількість елементів. Пункти списку навчального контенту формуються в основному на базі алгоритму  $kNN$ , але для різноманітності рекомендацій використовується випадковий алгоритм, щоб рекомендувати певну частину елементів.

Інкrementний гібридний алгоритм представлений алгоритмом 1 (рис. 2). Вхідні дані включають ідентифікатор користувача ( $u_{id}$ ) і три заданих користувачем параметра. Це загальна рекомендована кількість ( $N$ ), співвідношення випадкових рекомендацій ( $RT$ ) і  $TR$ . Фактично, він складається з чотирьох кроків.

**Input:**  $u_{id}, N, RT, TR$

**Output:** *recommender items*( $T_r$ )

**Method:** *hybridAlgorithm*

- (1)  $T_b$  = the browsed items by  $u_{id}$ ;
- (2)  $bm$  = the browser matrix of all users;
- (3) **if** ( $bm = \emptyset$  or  $T_b = \emptyset$ ) **then**
- (4)  $T_r$  = recommended  $N$  items by random algorithm;
- (5) **return**  $T_r$ ;
- (6) **end if**
- (7)  $U = \{u_1, u_2, \dots, u_n\}$ ;
- (8)  $Count = 0$ ;
- (9) **for each** ( $u_i$  in  $U$ ) **do**
- (10)  $N_b$  = the number of browsed items by  $u_i$ ;
- (11) **if** ( $N_b \geq TR$ ) **then**
- (12)  $Count = Count + 1$ ;
- (13) **end if**
- (14) **end for**
- (15) **if** ( $Count \leq 0$ ) **then**
- (16)  $T_r$  = recommended  $N$  items by random algorithm;
- (17) **return**  $T_r$ ;
- (18) **end if**
- (19)  $N_k = N * (1 - RT)$ ;
- (20)  $T_k$  = recommended  $N_k$  items by  $kNN$  algorithm;
- (21)  $N_{rd} = N - |T_k|$ ;
- (22)  $T_{rd}$  = recommended  $N_{rd}$  items by random algorithm;
- (23)  $T_r = T_k \cup T_{rd}$ ;
- (24) **return**  $T_r$ ;

Рис. 2. Алгоритм 1

Крок 1. На етапі ініціалізації немає ніяких даних історії матриці перегляду. Застосовується випадковий алгоритм для вирішення проблеми холодного старту. Цей крок відповідає рядкам 3-6 алгоритму.

Крок 2. Використовується, щоб визначити необхідність застосування алгоритму  $kNN$ . Спочатку підраховується кількість переглянутих елементів по  $u$ . Якщо  $N_b > TR$ , використовується алгоритм  $kNN$ . Цей крок відповідає рядкам 8-14 алгоритму.

Крок 3. Коли рекомендації працюють без помилок, використовується випадковий алгоритм, щоб рекомендувати деякі нові елементи для вивчення нових матеріалів. Цей крок відповідає рядкам 15-18 алгоритму.

Крок 4. Алгоритм  $kNN$  використовується, щоб рекомендувати деякі пункти списку. Інші пункти списку рекомендуються на основі випадкового алгоритму на перехідній і стабільній стадіях. Цей крок відповідає рядкам 17-23 алгоритму.

Вихідні дані алгоритму зберігаються в пам'яті для алгоритму 2, за результатами роботи якого, користувачу надається інтерактивна рекомендація (рис. 3).

Input:  $N, RT, TR$

Output: Recall and diversity

Method: interactiveRecommender

- (1)  $R_f = true$  //  $R_f$  is the flag of successfully recommender;
- (2)  $U = \{u_1, u_2, \dots, u_n\}$ ;
- (3)  $U'$  = an randomized array of  $U$ ;
- (4)  $T = \{t_1, t_2, \dots, t_m\}$ ;
- (5) while ( $R_f$ ) do
- (6)  $R_f = false$ ;
- (7) for each ( $u_{id}$  in  $U'$ ) do
- (8)  $T_r = hybridAlgorithm(u_{id}, N, RT, TR)$ ;
- (9)  $T_i = the\ interestarray\ of\ u_{id}$ ;
- (10) if ( $T_i \cap T_r = \emptyset$ ) then
- (11) the user quits to the RS;
- (12) else
- (13) the browsed item is recorded into  $bm(u_{id})$ ;
- (14)  $R_f = true$ ;
- (15) end if
- (16) end for
- (17) end while
- (18)  $recall = \frac{\sum_{i=1}^n \sum_{k=1}^m bm_{i,k}}{\sum_{i=1}^n \sum_{k=1}^m um_{i,k}}$ ;
- (19)  $BR(u, T) \subseteq T$ ;
- (20)  $diversity = \frac{|\cup_{u \in U} BR(u, T)|}{|T|}$ ;
- (21) return recall and diversity;

Рис. 3. Алгоритм 2

Програмне забезпечення рекомендаційної системи вебсервісу. Для коректної роботи вебсервісу

було виділено три рівні доступу до вебсервісу:

- гість – незареєстрований та неавторизований користувач системи, який не має доступу до функцій вебсервісу крім перегляду;
- користувач - зареєстрований та авторизований користувач системи, який має доступ до функцій вебсервісу, які відповідають за відображення рекомендаційного медіа-контенту;
- адміністратор – авторизований користувач системи, який має доступ до всіх функцій вебсервісу, налаштування системи та перегляду статистичних даних.

Після визначення рівнів доступу та задач, які виконує кожен користувач (табл. 1) була побудована БД, яка була розмежована згідно прав доступу, що надається користувачам системи в рамках вирішуваних ними задач (рис. 4). Виходячи з цього, розроблена система для формування рекомендацій була реалізована з сервісом регулювання навантаження на систему, кешуванням даних і запитів.

Для розробки вебсервісу використовувалася багатопарова архітектура [15], а саме 4-рівнева архітектура, яка надала сервісу гнучкість, можливість внесення змін в певний шар, замість того, щоб переробляти весь додаток цілком.

В якості програмної реалізації вебсервісу були застосовані такі технології, як:

- Python – об'єктно-орієнтована мова програмування високого рівня зі строгою динамічною типізацією [16];

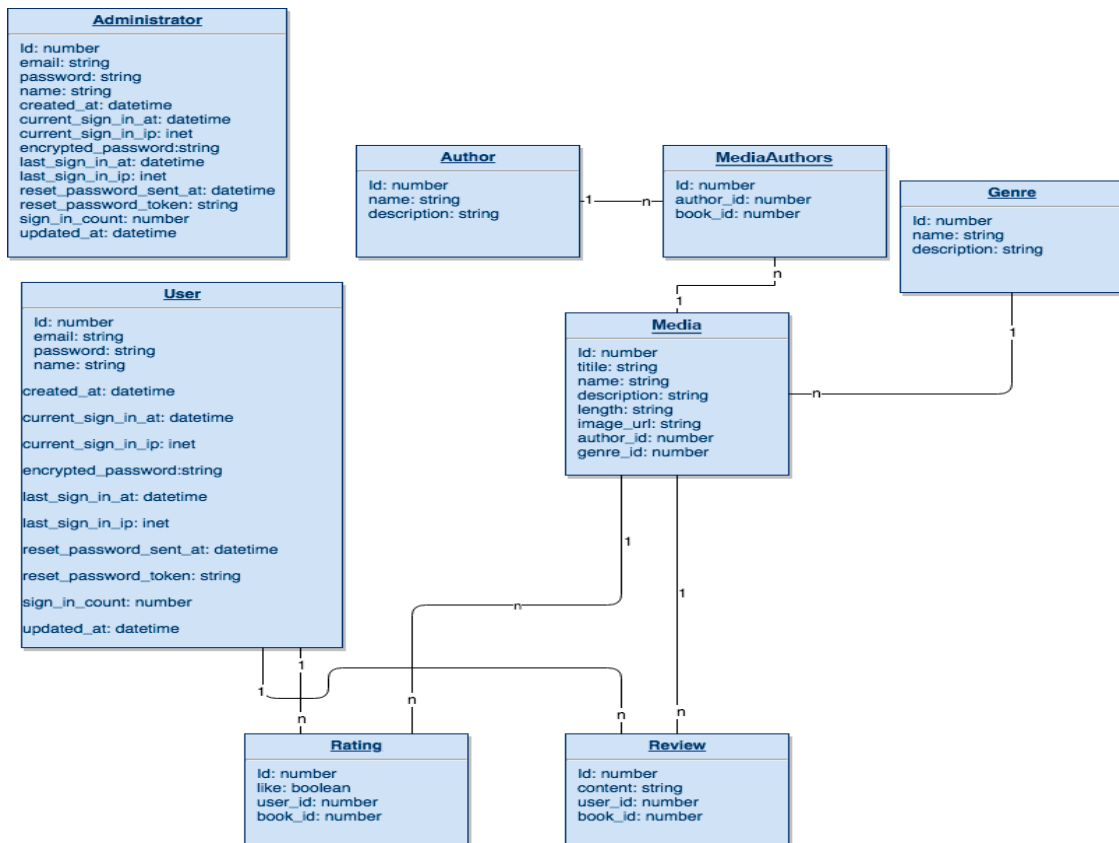


Рис. 4. Схема БД

Таблиця 1 – Перелік задач користувачів

Рівні доступу	Перелік задач	Вхідна інформація	Вихідна інформація
Гість	1) додавати, видаляти, змінювати дані у БД; 2) пошук інформації за будь-яким з можливих критеріїв; 3) змінювати БД відповідно до нових вимог предметної області	Інформація, що зазначена в документах та опис нових об'єктів предметної області	Підтвердження успішності запиту в разі додавання, видалення або зміни інформації в БД чи зміни самої БД; у разі пошуку інформації надається матеріал або повідомлення про його відсутність
Користувач	1) реєстрація та авторизація; 2) пошук та перегляд мультимедійного контенту; 3) перегляд інформації щодо акаунту	Логін та пароль, назва матеріалу	Підтвердження успішного запиту щодо реєстрації/авторизації користувача; надається матеріал, що потрібно знайти, або повідомлення про відсутність інформації; інформація акаунту
Адміністратор	1) перегляд статистики користувачів; 2) налаштування системи	Проміжок часу, критерії вибору користувачів, критерії вибору мультимедійного контенту	Відображення статистики з урахуванням пошукових критеріїв; підтвердження запиту щодо змін в системі

- JavaScript – динамічна, об'єктно-орієнтована прототипна мова програмування для створення сценаріїв вебсторінок для взаємодії з користувачем та обміну даними з сервером [17];

- NodeJS – платформа для виконання на серверній частині мережових Java script – скриптів за допомогою програмного рушія Java Script Google V8 [18];

- Fastify – швидкий, асинхронний вебфреймворк для Node.js [19];

- MongoDB – документо-орієнтована система керування базами даних, яка підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів [20];

- Mongoose – засіб моделювання об'єктів бази даних MongoDB, призначений для асинхронної роботи [21];

- Redis – розподілене сховище пар ключ-значення, які зберігаються в оперативній пам'яті з підтримкою структурованих даних, таких як: списки, хеші і множини [22].

**Висновки.** В роботі представлено розроблений вебсервіс з алгоритмічним забезпеченням рекомендаційної системи, що відповідає за надання навчального контенту при вивченні іноземної мови. На разі планується впровадження в декілька етапів: на першому етапі, серед студентів першокурсників Державного університету інтелектуальних технологій і зв'язку; на другому, серед студентів з поглибленим вивченням англійської мови; на третьому, серед всіх студентів університету. Після впровадження робота алгоритму може бути покращена на кожному з етапів, шляхом перенавчання під більш персоналізовані рекомендації за рахунок більшої кількості критеріїв формування пунктів списку.

Підсумовуючи роботу, слід зазначити, що рекомендаційні системи мультимедійного контенту мають значні перспективи застосування в індивідуальному онлайн навчанні, адже кожен користувач має

індивідуальні потреби, що спонукає його обирати вебсервіси, які зберігатимуть його час та надаватимуть необхідні мультимедійні матеріали.

#### Список літератури

1. Falk K. Recommender systems in practice. *Manning Publications*, 2017. 49 p. URL: <https://dmkpress.com/files/PDF/978-5-97060-774-9-1.pdf> (дата звернення: 13.01.2020).
2. *Статистичні дані ЮНЕСКО*. URL: <https://www.ukrinform.ua/rubric-world/3194622-u-sviti-vidznacaut-den-ridnoi-movi.html> (дата звернення: 30.03.2018).
3. Uzunboylu H. The Effectiveness of Web Assisted English Language Instruction on the Achievement and Attitude of Students. *ERIC Abstract*, 2005.
4. Tanyeli N. The Efficiency of Online English Language Instruction on Students' Reading Skills. *International Technology, Education and Development Conference (INTEd)*, Valencia, Spain, Mar 3–5, 2008. URL: <https://files.eric.ed.gov/fulltext/ED504676.pdf> (дата звернення: 06.05.2021).
5. Adomavicius G., Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 2005, 17 (6), pp.734–749.
6. Candillier L., Jack K., Fessant F., Meyer F. State-of-the-art recommender systems. *Collaborative and Social Information Retrieval and Access Techniques for Improved User Modeling*, 2009.
7. Pazzani M. J., Billsus D. Content-based recommendation systems. *The adaptive web*. Springer, Berlin, Heidelberg, 2007, pp. 325–341.
8. Ricci F., Rokach L., Shapira B. Introduction to recommender systems handbook. *Recommender systems handbook*. Springer US, 2011, pp. 1–35.
9. *Система рекомендацій YouTube*. URL: <https://neurohive.io/ru/novosti/iz-chego-sostoit-rekomendatelnaya-sistema-youtube> (дата звернення: 24.04.2019).
10. *Система рекомендацій Netflix*. URL: <https://uxplanet.org/netflix-binging-on-the-algorithm-a3a74a6c1f59> (дата звернення: 30.04.2020).
11. *Система рекомендацій Megogo*. URL: <https://www.contentarmor.net/megogo-selects-contentarmor-to-protect-its-pvod-service/> (дата звернення: 12.04.2021).
12. *Система рекомендацій Ivi*. URL: [https://discourse.world/h/2019/12/10/As-we-recommend-the-latest-catalogs-in-the-ivi-online-cinema\(Python-code\)](https://discourse.world/h/2019/12/10/As-we-recommend-the-latest-catalogs-in-the-ivi-online-cinema(Python-code)) (дата звернення: 10.12.2019).
13. Thai-Nghe N., Drumond L., Krohn-Grimberghe A., Schmidt-Thieme L. Recommender system for predicting student performance. *Procedia Computer Science*, 2010, 1 (2), pp. 2811–2819.

14. Moonen M., De Moor B. *SVD and Signal Processing, III: Algorithms, Architectures and Applications*. Elsevier Science, 2011. 504 p.
15. Bass L., Clements P., Kazman R. *Software Architecture in Practice*. 2-nd edition, Addison-Wesley, 2003.
16. Jones E., Oliphant Tr., Peterson P. et al. *SciPy: Open source scientific tools for Python, 2001*. URL: <http://www.scipy.org/> (дата звернення: 06.12.2015).
17. Zheng Y., Mobasher B. CARSKit: A Java-Based Context-aware Recommendation Engine. *Proceedings of the 15th IEEE International Conference on Data Mining*. NJ, USA, 2015.
18. *NodeJS*. URL: <https://metanit.com/web/nodejs/1.1.php> (дата звернення: 02.10.2019).
19. *Fastify framework*. URL: <https://www.fastify.io> (дата звернення: 06.11.2020).
20. *MongoDB documentation*. URL: <https://docs.mongodb.com/manual> (дата звернення: 26.08.2021).
21. *Mongoose Docs*. URL: <https://docs.npmjs.com/> (дата звернення: 09.09.2021) (дата звернення: 06.12.2019).
22. *Redis documentation*. URL: <https://redis.io/documentation> (дата звернення: 17.11.2020).

#### References (transliterated)

1. Falk K. Recommender systems in practice. *Manning Publications*, 2017. 49 p. URI: <https://dmkpress.com/files/PDF/978-5-97060-774-9-1.pdf> (accessed 13.01.2020).
2. *UNESCO statistics*. Available at: <https://www.ukrinform.ua/rubric-world/3194622-u-sviti-vidznacaut-den-ridnoi-movi.html> (accessed 30.03.2018).
3. Uzunboylu H. The Effectiveness of Web Assisted English Language Instruction on the Achievement and Attitude of Students. *ERIC Abstract*, 2005.
4. Tanyeli N. The Efficiency of Online English Language Instruction on Students' Reading Skills. *International Technology, Education and Development Conference (INTED)*, Valencia, Spain, Mar 3-5, 2008. Available at: <https://files.eric.ed.gov/fulltext/ED504676.pdf> (accessed 06.05.2021).
5. Adomavicius G., Tuzhilin A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, 2005, 17 (6), pp.734–749.
6. Candillier L., Jack K., Fessant, F., Meyer F. State-of-the-art recommender systems. *Collaborative and Social Information Retrieval and Access Techniques for Improved User Modeling*, 2009.
7. Pazzani M. J., Billsus D. Content-based recommendation systems. *The adaptive web*. Springer, Berlin, Heidelberg, 2007, pp. 325-341.
8. Ricci F., Rokach L., Shapira B. Introduction to recommender systems handbook. *Recommender systems handbook*. Springer US, 2011, pp. 1–35.
9. *YouTube documentation*. Available at: <https://neurohive.io/ru/novosti/iz-chego-sostoit-rekomendatelnaya-sistema-youtube> (accessed 24.04.2019).
10. *Netflix documentation*. Available at: <https://uxplanet.org/netflix-binging-on-the-algorithm-a3a74a6c1f59> (accessed 30.04.2020).
11. *Megogo documentation*. Available at: <https://www.contentarmor.net/megogo-selects-contentarmor-to-protect-its-pvod-service/> (accessed 12.04.2021).
12. *Ivi documentation*. Available at: [https://discourse.world/h/2019/12/10/As-we-recommend-the-latest-catalogs-in-the-ivi-online-cinema\(Pyton-code\)](https://discourse.world/h/2019/12/10/As-we-recommend-the-latest-catalogs-in-the-ivi-online-cinema(Pyton-code)) (accessed 10.12.2019).
13. Thai-Nghe N., Drumond L., Krohn-Grimberghe A., Schmidt-Thieme L. Recommender system for predicting student performance. *Procedia Computer Science*. 2010, 1 (2), pp.2811–2819.
14. Moonen M., De Moor B. *SVD and Signal Processing, III: Algorithms, Architectures and Applications*. Elsevier Science, 2011. 504 p.
15. Bass L., Clements P., Kazman R. *Software Architecture in Practice*. 2-nd edition, Addison-Wesley, 2003.
16. Jones E., Oliphant Tr., Peterson P. et al. *SciPy: Open source scientific tools for Python, 2001*. Available at: <http://www.scipy.org/> (accessed 06.12.2015).
17. Zheng Y., Mobasher B. CARSKit: A Java-Based Context-aware Recommendation Engine. *Proceedings of the 15th IEEE International Conference on Data Mining*. NJ, USA, 2015.
18. *NodeJS*. Available at: <https://metanit.com/web/nodejs/1.1.php> (accessed 02/10/2019).
19. *Fastify framework*. Available at: <https://www.fastify.io> (accessed 06.11.2020).
20. *MongoDB documentation*. Available at: <https://docs.mongodb.com/manual> (accessed 26.08.2021).
21. *Mongoose Docs*. Available at: <https://docs.npmjs.com/> (accessed 09.09.2021) (accessed 06.12.2019).
22. *Redis documentation*. Available at: <https://redis.io/documentation> (accessed 17.11.2020).

Надійшла (received) 01.11.2021

#### Відомості про авторів / Сведения об авторах / About the Authors

**Боднар Лілія Василівна** – кандидат педагогічних наук, доцент, Південноукраїнський національний педагогічний університет імені К. Д. Ушинського, доцент кафедри інноваційних технологій та методики навчання природничих дисциплін; м. Одеса, Україна; ORCID: <https://orcid.org/0000-0003-0893-496X>; e-mail: [bodnarl79@gmail.com](mailto:bodnarl79@gmail.com)

**Шулакова Катерина Сергіївна** – Державний університет інтелектуальних технологій і зв'язку, старший викладач кафедри комп'ютерної інженерії та інформаційних систем; м. Одеса, Україна; ORCID: <https://orcid.org/0000-0002-0035-6184>; e-mail: [katejojo29@gmail.com](mailto:katejojo29@gmail.com)

**Гризун Людмила Едуардівна** – доктор педагогічних наук, професор, Харківський національний економічний університет імені С. Кузнеця, професор кафедри інформаційних систем; м. Харків, Україна; ORCID: <https://orcid.org/0000-0002-5274-5624>; e-mail: [lgr2007@ukr.net](mailto:lgr2007@ukr.net)

**Боднар Лилия Васильевна** – кандидат педагогических наук, доцент, Южноукраинский национальный педагогический университет имени К. Д. Ушинского, доцент кафедры инновационных технологий и методики обучения естественных дисциплин; г. Одесса, Украина; ORCID: <https://orcid.org/0000-0003-0893-496X>; e-mail: [bodnarl79@gmail.com](mailto:bodnarl79@gmail.com)

**Шулакова Екатерина Сергеевна** – Государственный университет интеллектуальных технологий и связи, старший преподаватель кафедры компьютерной инженерии и информационных систем; г. Одесса, Украина; ORCID: <https://orcid.org/0000-0002-0035-6184>; e-mail: [katejojo29@gmail.com](mailto:katejojo29@gmail.com)

**Гризун Людмила Эдуардовна** – доктор педагогических наук, профессор, Харьковский национальный экономический университет имени С. Кузнеця, профессор кафедры информационных систем; г. Харьков, Украина; ORCID: <https://orcid.org/0000-0002-5274-5624>; e-mail: [lgr2007@ukr.net](mailto:lgr2007@ukr.net)



**Bodnar Liliia Vasylivna** – candidate of pedagogical sciences, docent, South Ukrainian National Pedagogical University named after K.D. Ushynsky, associate professor of the department of Innovative Technologies and Methods of Teaching Natural Sciences; Odessa, Ukraine; ORCID: <https://orcid.org/0000-0003-0893-496X>; e-mail: bodnarl79@gmail.com

**Shulakova Kateryna Serhiivna** – State University of Intellectual Technologies and Telecommunications, senior lecturer of the department of Computer Engineering and Information Systems; Odessa, Ukraine; ORCID: <https://orcid.org/0000-0002-0035-6184>; e-mail: katejojo29@gmail.com

**Gryzun Liudmyla Eduardivna** – doctor of pedagogical sciences, professor, S. Kuznets Kharkiv National University of Economics; Kharkiv professor of the department of Information Systems; Kharkiv, Ukraine; ORCID: <https://orcid.org/0000-0002-5274-5624>; e-mail: lgr2007@ukr.net

### ЗМІСТ

СИСТЕМНИЙ АНАЛІЗ І ТЕОРІЯ ПРИЙНЯТТЯ РІШЕНЬ.....	3
<i>Pavlov A. A.</i> Estimating with a given accuracy of the coefficients at nonlinear terms of univariate polynomial regression using a small number of tests in an arbitrary limited active experiment.....	3
<i>Shepeliev O. V., Bilova M. O.</i> Software testing results analysis for the requirements conformity using neural networks.....	8
УПРАВЛІННЯ В ТЕХНІЧНИХ СИСТЕМАХ.....	15
<i>Бабуджан Р. А., Ісаснков К. О., Красій Д. М., Водка О. О., Задорожний І. В., Ющук М. В.</i> Використання методів машинного навчання для бінарної класифікації робочого стану підшипників за сигналами їх віброприскорення.....	15
УПРАВЛІННЯ В ОРГАНІЗАЦІЙНИХ СИСТЕМАХ.....	23
<i>Orehov S. V., Malyhon H. V.</i> Metrics of virtual promotion of a product.....	23
МАТЕМАТИЧНЕ І КОМП'ЮТЕРНЕ МОДЕЛЮВАННЯ.....	27
<i>Мазманишвили А. С., Решетняк Н. Г., Сидоренко А. Ю.</i> Динамика электронного пучка формируемого магнетронной пушкой с вторичноэмиссионным катодом, в спадающем магнитном поле соленоида: эксперимент и моделирование.....	27
<i>Litvinov B. R., Bilova M. O.</i> Human body modeling technologies for e-commerce systems.....	35
<i>Чалий С. Ф., Лециньський В. О., Лециньська І. О.</i> Контрфактуальна темпоральна модель причинно-наслідкових зв'язків для побудови пояснень в інтелектуальних системах.....	41
<i>Нікуліна О. М., Северин В. П., Надуєва М. О., Бубнов А. І.</i> Моделювання розвитку епідемії на основі інформаційної технології оптимізації.....	47
<i>Гамаюн І. П., Шапков С. В.</i> Математична модель адаптивного управління функціонуванням світлофора на перехресті міської транспортної мережі.....	53
ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ.....	59
<i>Sokol V. Y., Krykun V. O., Bilova M. O., Perepelytsya I. D., Pustovarov V. V.</i> Topic segmentation methods comparison on computer science texts.....	59
<i>Kopp A. M., Orlovskiy D. L., Ersoyleyev D.</i> An approach to analysis of ArchiMate application architecture models using the software coupling metric.....	67
<i>Kozulia M. M., Sushko V. V.</i> Determine recommendation systems to search for books by preferences of web users.....	73
<i>Marchenko N. A., Sydorenko G. Yu., Rudenko R. O.</i> Using of multilayer neural networks for the solving systems of differential equations.....	81
<i>Godlevskiy I. M., Godlevskiy M. D., Stativka I. V.</i> Information technology of forming option for logistics distribution channel configuration resistant to emergencies.....	89
<i>Погребняк С. В., Водка О. О.</i> Застосування методів штучного інтелекту для апроксимації механічної поведінки гумоподібних матеріалів.....	95
<i>Боднар Л. В., Шулакова К. С., Гризун Л. Е.</i> Алгоритмічне забезпечення рекомендаційної системи вебсервісу для вивчення іноземних мов.....	100