

МЕТОД ПІДБОРУ ТА РОЗПОДІЛУ РЕКОМЕНДОВАНИХ НАВЧАЛЬНИХ КУРСІВ ДЛЯ ПРАЦІВНИКІВ ІТ-КОМПАНІЇ

Колесник Л. В.¹, Батраченко В. О.², Колесник О. Б.¹

¹Харківський національний університет радіоелектроніки

²ТОВ «ЕПАМ СИСТЕМЗ», м. Київ

Анотація. У статті розглядається проблема підбору та розподілу рекомендованих навчальних курсів для працівників ІТ-компанії. Метою роботи є розроблення ефективного методу, який забезпечує автоматизований вибір релевантних навчальних матеріалів та їх оптимальний розподіл між співробітниками. У межах дослідження здійснено аналіз освітніх потреб працівників, визначено ключові критерії відбору курсів, а також запропоновано метод підбору на основі аналізу вмісту та рекомендаційний підхід, що враховує індивідуальні компетенції, мету професійного розвитку та інтереси працівників. Розроблений метод дозволяє формувати персоналізовані освітні траєкторії, підвищувати ефективність навчання та сприяти більш раціональному розподілу ресурсів компанії. Впровадження запропонованого підходу забезпечить покращення професійних навичок персоналу та підвищить якість вирішення проєктних завдань у сфері інформаційних технологій.

Ключові слова: рекомендаційна система, метод на основі вмісту, персоналізоване навчання, розвиток навичок, оптимізаційна модель, метод ідеальної точки, максимізація корисності.

Вступ

Сучасний етап розвитку суспільства визначається стрімким зростанням функції диджиталізації, яка охоплює практично всі сфери життя та бізнесу. Цифрові технології стають ключовим фактором конкурентоспроможності організацій, а їх упровадження – стратегічним пріоритетом для більшості компаній.

Водночас розвиток технологій зумовлює підвищення вимог до компетенцій ІТ-працівників. Для утримання позицій на ринку та розширення проєктної діяльності компанії мають забезпечувати безперервний професійний розвиток своїх співробітників. Хоча більшість організацій створюють власні освітні ресурси, рівень складності вибору релевантних навчальних курсів збільшується через надлишок інформації та швидку зміну технологічних тенденцій. Невдалий вибір напрямів навчання призводить до дефіциту необхідних навичок, що може зменшити конкурентоспроможність компанії та спричинити втрату прибутку.

Отже, актуальним є завдання розроблення методу підбору та розподілу лише тих навчальних курсів, які найбільш відповідають професійним потребам працівників і вимогам поточних або перспективних проєктів [1]. Для цього пропонується використання рекомендаційного підходу, який враховує багатокритеріальність, а саме: аналіз ринку, профілі

працівників та характеристики доступних курсів.

Реалізація такого підходу дозволить підвищити ефективність процесу навчання персоналу, оптимізувати розподіл ресурсів і зміцнити позиції ІТ-компаній у конкурентному середовищі.

Для більш детального розуміння можливостей і перспектив застосування рекомендаційних підходів у сфері корпоративного навчання доцільно проаналізувати наявні наукові напрацювання та практичні рішення в цій галузі.

Аналіз публікацій

Рекомендаційні системи розглядають як інформаційні технології, що на основі аналізу даних та застосування алгоритмічних підходів формують персоналізовані рекомендації навчальних курсів для працівників з метою підвищення їхніх професійних компетентностей. Використання таких систем дає змогу ІТ-компаніям забезпечити належний рівень кваліфікації персоналу, підвищити ефективність проєктної діяльності та зміцнити позиції на ринку.

Процес надання персоналізованих рекомендацій полягає в доборі контенту, товарів чи послуг відповідно до індивідуальних потреб, інтересів і вподобань користувачів. Виокремлюють три основні підходи до формування таких рекомендацій [2].

Перший підхід – на підставі ознакових описів (content-based filtering) – використовує інформацію про властивості об'єктів і ґрунтується на ідеї рекомендувати користувачам ті об'єкти, що подібні до раніше вподобаних, з урахуванням їхніх характеристик і параметрів [3]. Перевагами такого підходу є простота реалізації та здатність до роботи з невеликою кількістю інформації. Однак підходи на основі ознакових описів мають обмежену потужність, що проявляється у їхній неспроможності ефективно моделювати складні залежності між користувачами та об'єктами, а також враховувати латентні вподобання та контекстні фактори. Крім того, такі підходи значною мірою залежать від якості та повноти опису об'єктів, що обмежує їх застосування у випадках неповної або слабо структурованої інформації.

Другий підхід – колаборативна фільтрація (collaborative filtering) – ґрунтується на аналізі взаємодії користувачів з об'єктами та передбачає формування персоналізованих рекомендацій через виявлення схожих користувачів або об'єктів на основі історії їхньої активності (оцінок, переглядів, лайків, додань до кошика чи покупок) [4].

Третій підхід – гібридний метод (hybrid filtering) – поєднує принципи колаборативної фільтрації та контент-базованих підходів, що дає змогу одночасно враховувати властивості об'єктів і подібність користувачів. Завдяки цьому гібридні рекомендаційні системи забезпечують вищу точність і різноманітність рекомендацій, підвищуючи рівень задоволення користувачів [5].

Однак застосування традиційних рекомендаційних підходів не завжди забезпечує достатню точність результатів. Це пов'язано з низкою факторів, зокрема з проблемою холодного старту (відсутність інформації про нових користувачів або об'єкти), високою розрідженістю матриці взаємодій, динамічною зміною вподобань користувачів, а також обмеженою здатністю враховувати багатовимірний контекст прийняття рішень.

Тому в сучасних рекомендаційних системах дедалі більшого поширення набувають методи багатокритеріального прийняття рішень, які дозволяють враховувати одночасно декілька суперечливих критеріїв, зокрема ступінь узгодженості освітніх потреб ІТ-компаній із професійними інтересами їхніх працівників.

До таких методів належать метод максимізації функції корисності (зваженої суми)

[6, 7], метод мінімізації відстані до ідеального рішення (TOPSIS) [8, 9], метод аналізу ієрархій (АНР) [9], а також підходи на основі нечіткої логіки [10].

Водночас методи аналізу ієрархій та нечіткої логіки визначаються підвищеною обчислювальною складністю та потребують залучення експертного аналізу, що ускладнює їх практичне застосування в умовах великої кількості альтернатив. У зв'язку з цим у нашому дослідженні запропоновано підхід, що поєднує переваги методів агрегування критеріїв і методу ідеальної точки, зокрема через застосування методу мінімізації зваженої відстані до ідеального рішення. Такий підхід дозволяє враховувати важливість критеріїв та ефективно знаходити компромісні рішення серед суперечливих.

Мета та постановка завдання

Метою дослідження є розроблення методу, що забезпечує ефективне узгодження освітніх потреб ІТ-компаній із професійними інтересами їхніх працівників під час формування рекомендаційних курсів.

Нехай R – резюме працівника, що містить основну інформацію, зокрема сферу розроблення, рівень та навички з їхніми рівнями; $U = \{u_1, u_2, \dots, u_n\}$ – множина наявних проєктів у системі, які можуть використовуватися для аналізу вимог (n – кількість проєктів); $V = \{v_1, v_2, \dots, v_m\}$ – множина наявних курсів у системі, що можуть бути запропоновані співробітнику (m – кількість курсів).

Кожен i -й проєкт u_i поданий як вектор текстових характеристик: $u_i = \{u_i^1, u_i^2, \dots, u_i^t\}$ (t – кількість характеристик проєкту).

Кожен j -й курс v_j поданий як вектор текстових характеристик: $v_j = \{v_j^1, v_j^2, \dots, v_j^s\}$, (s – кількість характеристик курсу).

Нехай

$$h : R \times U \times V \rightarrow E,$$

де h – функція, яка відображає корисність проєктів $u \in U$ та курсів $v \in V$ для працівника на основі його резюме R ; E – впорядкована множина (наприклад, цілі невід'ємні чи дійсні числа в певному діапазоні).

Тоді, для користувача R необхідно знайти такі курси $v \in V$, які дозволили б отримати максимальну корисність під час їх вивчення для покращення резюме працівника R .

Математично задачу пошуку найбільш релевантних курсів можна описати як

$$v^*(u) = \underset{v \in V}{\operatorname{arg\,max}} h(R, u, v),$$

де $h(R, u, v)$ – функція, що відображає корисність курсу v для працівника з резюме R у контексті проекту u .

У простих системах рекомендацій, що використовують явний зворотний зв'язок, функція корисності h виражається як оцінка, яку користувач ставить певному продукту за фіксованою шкалою. Проте значення функції h у нашому випадку залежить від трьох параметрів та має враховувати предметну галузь й обмеження, що впливають на її основі, наприклад вимоги проектів.

Тому завданням дослідження є виявлення такої функції корисності, яка б виявилася більш ефективною за наявні аналоги та задовільнила б вимоги.

1 Метод підбору та розподілу рекомендованих курсів працівникам ІТ-компанії

Визначимо основні етапи методу (рис. 1) та зазначимо вимоги до кожного з них [11]:

– *аналіз резюме працівника*. На цьому етапі здійснюється збір та оброблення інформації про працівника. Основними параметрами є його сфера діяльності (наприклад, веброзроблення), кваліфікаційний рівень, що є одним із ключових критеріїв під час формування рекомендацій навчальних курсів, а також наявні навички та ступінь їх опанування (за шкалою від 1 до 4). Відповідно до рівня кваліфікації працівники поділяються на чотири категорії: Junior, Middle, Senior та Lead;

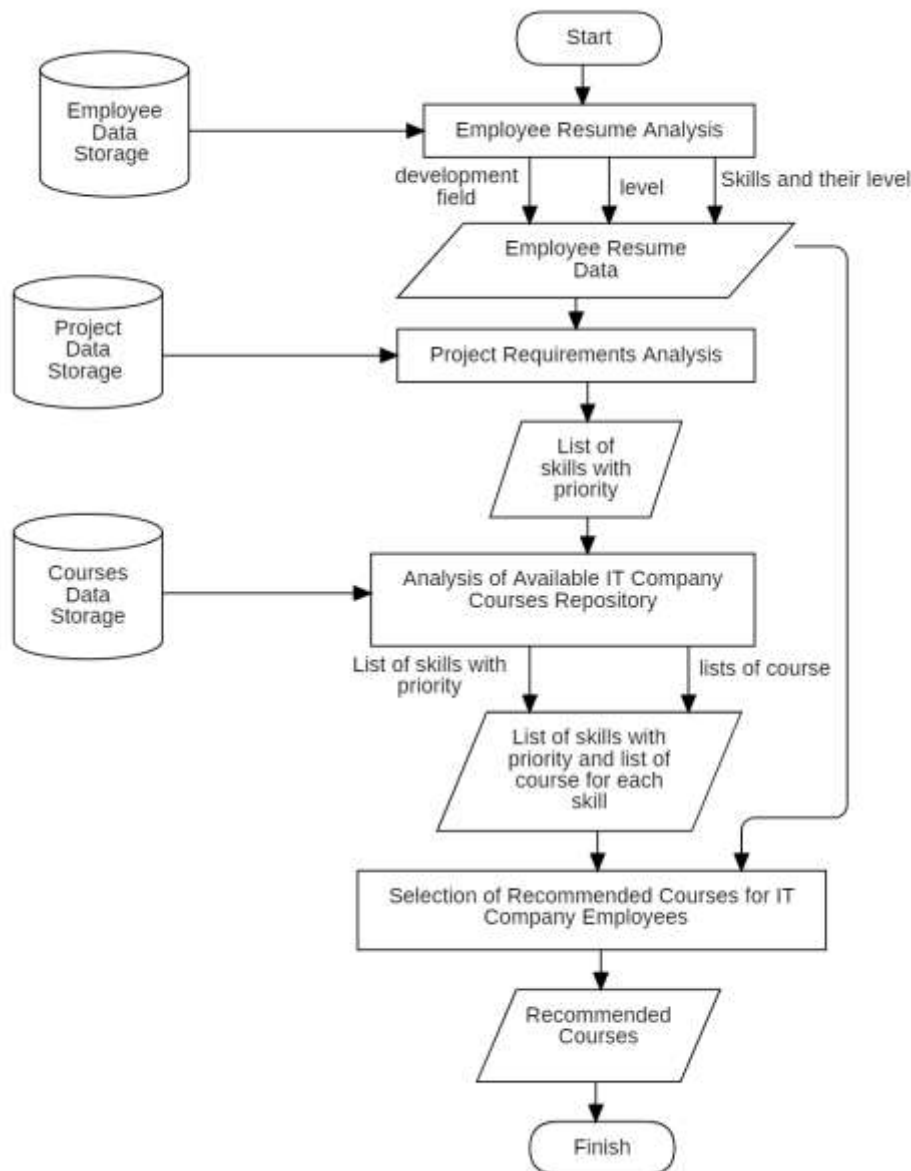


Рис. 1. Схема алгоритму методу підбору та розподілу рекомендованих курсів для працівників ІТ-компанії

– *аналіз вимог проєктів*. На цьому етапі здійснюється збір інформації про проєкти, зокрема визначаються їхня сфера реалізації, рівень працівника, а також необхідні та бажані вимоги до кандидатів. Отримана інформація дає змогу визначити навички, які потребують підвищення або опанування (наприклад, удосконалення рівня володіння технологією Java чи вивчення нового інструменту, такого як Spring Framework);

– *аналіз доступних курсів ІТ-компанії*. На цьому етапі здійснюється аналіз навчальних курсів, що пропонуються ІТ-компанією, з метою оцінювання їхньої доступності та формування інформації для подальшого коректного розподілу. Для проведення цього етапу використовуються результати попереднього – «List of skills with priority», який містить перелік навичок із визначеними пріоритетами. Додатково залучаються дані зі сховища «Courses Data Storage», що містять такі параметри: назву курсу, тривалість навчання, максимальну кількість учасників, поточну кількість зареєстрованих студентів, кількість працівників, яким курс уже був рекомендований, дату початку навчання, сферу розроблення та навички. Під час цього процесу формуються вихідні дані «List of skills with priority and list of courses for each skill», які містять перелік навичок із визначеними пріоритетами та відповідними курсами для їх розвитку;

– *підбір рекомендованих курсів для працівників ІТ-компанії*. На цьому етапі, використовуючи результати попередніх, здійснюється підбір навчальних курсів для працівників за допомогою алгоритму розподілу. Для цього на вхід подаються дані з «List of skills with priority and list of courses for each skill» та «Employee Resume Data» і формується вихідний блок «Recommended Courses», який містить перелік рекомендованих курсів для кожного працівника із зазначенням пріоритетності їх проходження.

Розглянемо більш детально кожен етап алгоритму.

1.1 Етап 1: аналіз резюме працівника

Необхідною умовою для реалізації цього етапу є наявність сховища даних «Employee Resume Data», з якого виокремлюються параметри «Area», «Level» та список «Skill» із відповідними рівнями володіння «Skill level». Приклад вихідних даних цього етапу наведено на рис. 2. Цю інформацію необхідно передати на наступний етап.

```
{
  "area": "Web development",
  "level": "Middle",
  "skills": [
    {
      "name": "Java",
      "level": 4
    },
    {
      "name": "Spring Framework",
      "level": 3
    }
  ]
}
```

Рис. 2. Приклад вихідних даних етапу 1

1.2 Етап 2: аналіз вимог проєктів

Необхідною умовою для реалізації цього етапу є наявність на вході даних у форматі, наведеному на рис. 2, та доступного сховища «Project Data Storage», з якого потрібно отримати список сфер розроблення проєкту, вимоги до рівня працівника, необхідні та бажані навички. Приклад вихідних даних цього етапу наведено на рис. 3.

```
{
  "area": [
    "Web development",
    "Mobile development"
  ],
  "requiredLevel": "Middle",
  "requiredSkills": [
    {
      "name": "Java",
      "level": 3
    },
    {
      "name": "Spring Framework",
      "level": 3
    }
  ],
  "desiredSkills": [
    {
      "name": "SpringBoot",
      "level": 3
    },
    {
      "name": "Angular",
      "level": 2
    }
  ]
}
```

Рис. 3. Приклад вихідних даних етапу 2

Після отримання списку проєктів (рис. 3) необхідно його відфільтрувати, залишивши лише ті, які мають збіг за сферою розроблен-

ня з працівником. Наступним етапом є фільтрація за необхідним рівнем працівника для проекту: якщо різниця між рівнями перевищує один, такий проект необхідно викреслити зі списку. Наприклад, якщо для проекту потрібен рівень Senior, а працівник має рівень Junior, цей проект не враховується.

Після відбору інформація потребує подальшого оброблення. Однією з ключових умов є забезпечення того, щоб загальна відповідність навичок працівника вимогам проекту становила щонайменше 50 %. Це необхідно для того, щоб не використовувати проекти, де більшість технологій незнайома співробітнику. Через значну кількість технологій можливі випадки, коли нерелевантні проекти можуть потрапляти до алгоритму, тому застосування такого критерію забезпечує більш коректний відбір.

Для оцінювання загального задоволення вимогам проекту X використовується така формула:

$$X = \frac{(C_{r1} + C_{r2} + \dots + C_{rk})}{k} \cdot a_r + \frac{(C_{d1} + C_{d2} + \dots + C_{dl})}{l} \cdot a_d,$$

$$a_r + a_d = 1,$$

де X – загальне задоволення навичками працівника вимогам проекту; k – кількість необхідних навичок; l – кількість бажаних навичок; a_r – коефіцієнт важливості для необхідних навичок ($0 < a_r < 1$); a_d – коефіцієнт для бажаних навичок ($0 < a_d < 1$); C_{ri} – i -те задоволення необхідних навичок; C_{dj} – j -те задоволення бажаних навичок.

Рівень задоволення i -тої необхідної навички працівника C_{ri} визначається за таким правилом:

$$C_{ri} = \begin{cases} 100\%, & \text{якщо рівень навички відповідає або перевищує вимогу;} \\ 50\%, & \text{якщо рівень навички нижчий за вимогу на один рівень;} \\ 0\%, & \text{якщо рівень навички нижчий на два та більше рівнів.} \end{cases}$$

Такий підхід дозволяє кількісно оцінити, наскільки наявні навички співробітника відповідають вимогам проекту та сформувані релевантний список відповідно до вимог.

Наступним етапом є формування списку навичок та присвоєння їм оцінок, які визначають пріоритет вивчення. Для цього ство-

рюється словник, де ключем є назва навички, а значенням – її оцінка. Словник є колекцією унікальних пар «ключ-значення», що дозволяє швидко знаходити оцінку за назвою навички [12].

Далі відбувається ітерація за списком проектів та аналіз їхніх вимог до навичок. Якщо рівень навички в проекті дорівнює рівню або менший за рівень працівника, вона пропускається.

Якщо ж рівень більший, перевіряється наявність цієї навички в словнику; якщо вона наявна, до оцінки додається певне значення, розраховане за формулою

$$Y = P \cdot K,$$

де P – коефіцієнт проекту, що залежить від співвідношення рівня навички працівника та вимоги проекту (якщо рівні однакові, то $P = 1$; в іншому випадку $P = 0,75$); K – коефіцієнт типу навички (для необхідних навичок $K = 1$; для бажаних – $K = 0,5$).

Після оброблення всіх проектів оцінки навичок у словнику нормуються: кожен оцінку ділять на кількість проектів і множать на 100, після чого результат округлюється до цілого числа. Таким чином, значення оцінки навички варіюється від 0 до 100. На виході отримуємо словник із навичками та оцінками (рис. 4).

```
[
  {
    "name": "Java",
    "priority": 56
  },
  {
    "name": "Spring Framework",
    "priority": 73
  }
]
```

Рис. 4. Приклад вихідних даних словника проекту

1.3 Етап 3: аналіз доступних курсів ІТ-компанії

Необхідною умовою для реалізації цього етапу є наявність вхідних даних у форматі, наведеному на рис. 3, а також доступ до сховища «Courses Data Storage», з якого потрібно отримати основні характеристики курсів (рис. 5).

До таких характеристик належать назва курсу, тривалість його вивчення, максимальна кількість студентів, що можуть записатися, кількість студентів, які вже записані, кі-

лькість працівників, яким курс був рекомендований, дата початку навчання, сфери розроблення та відповідні навички.

```
{
  "title": "Java Course for Web development",
  "area": [
    "Web development",
    "Mobile development"
  ],
  "skills": [
    {
      "name": "Java",
      "requiredLevel": 2,
      "enhancementLevel": 3
    },
    {
      "name": "REST API",
      "enhancementLevel": 1
    },
    {
      "name": "Unit Testing",
      "requiredLevel": 1
    }
  ],
  "duration": 80,
  "capacity": 100,
  "enrolled": 30,
  "recommended": 10,
  "start": "10.04.2024"
}
```

Рис. 5. Приклад вхідних даних курсу

Як видно з прикладу структури даних (рис. 5), параметр «requiredLevel» задає мінімальний рівень навички, необхідний для проходження курсу, і використовується для попередньої фільтрації навчальних програм. Параметр «enhancementLevel» визначає цільовий рівень компетентності, якого працівник може досягти після завершення курсу, що дозволяє оцінити корисність курсу в контексті підвищення кваліфікації.

Після формування списку курсів із вхідними даними необхідно здійснити їх первинну фільтрацію: залишити лише ті, що відповідають щонайменше одній сфері розроблення працівника, мають дату початку не раніше поточної, а також доступні за кількістю місць (тобто кількість вже зареєстрованих студентів є меншою за максимально допустиму).

Після цього відібрані дані підлягають подальшому обробленню з метою врахування реальної доступності курсів для кожної з навичок. Це є ключовим аспектом, який дозволяє уникнути ситуацій, коли всі працівники потрапляють на один і той самий курс із обмеженою кількістю місць або коли курс для певного навичка взагалі недоступний.

На першому етапі оброблення словник навичок впорядковується за спаданням їхнього пріоритету. Далі для кожного навичка зі словника визначається перелік курсів відповідно до вимог для його опанування. Водночас необхідно врахувати низку додаткових обмежень: по-перше, навички, для яких не знайдено жодного відповідного курсу, потрібно вилучити з подальшого розгляду, щоб уникнути оброблення нерелевантних даних; по-друге, для кожного курсу слід визначити його пріоритет з урахуванням важливості навичок і характеристик самого курсу, зокрема тривалості, кількості доступних місць, кількості вже зареєстрованих слухачів і кількість попередніх рекомендацій.

На другому етапі оброблення здійснюється аналіз за словником навичок, що впорядковані за спаданням відповідно до їхнього пріоритету. Для кожної визначається перелік курсів згідно з умовами відбору, а саме:

- назва навички має бути однаковою з відповідною навичкою курсу;

- якщо навичка курсу містить атрибут «requiredLevel», необхідно перевірити, чи відповідає рівень працівника мінімальному рівню, потрібному для опанування курсу; за відсутності цього атрибута попередні знання не є обов'язковими;

- якщо навичка курсу має атрибут «enhancementLevel», потрібно переконатися, що він перевищує поточний рівень працівника, тобто курс дійсно забезпечує розвиток хоча б однієї навички; у протилежному випадку курс відсіюється.

Якщо курс відповідає зазначеним умовам, він додається до списку можливих для опанування відповідної навички. Після цього для кожного такого курсу необхідно визначити його пріоритет. На цьому етапі виникає певна складність, оскільки курс може покращувати не один, а кілька навичок. Отже, під час розрахунку пріоритету потрібно врахувати низку критеріїв, зокрема такі:

- сумарні оцінки всіх навичок, які покращує курс, з урахуванням їхнього пріоритету та рівня підвищення;

- кількість вільних місць на курсі;

- тривалість курсу.

Однак у випадках, коли курс підвищує кілька навичок одночасно, більш ефективним підходом є опрацювання не словника навичок, а списку курсів. Для цього формується оновлений словник, у якому ключем є навичка, а значенням – список курсів із розрахованими пріоритетами.

Після цього опрацьовується весь список курсів: для кожного з них обчислюється пріоритет за формулою

$$P_{skills} = \sum_{i=1}^f O_i \cdot (e_i - c_i),$$

де f – кількість навичок, що покращуються курсом (мають атрибут «enhancementLevel»); O_i – оцінка відповідної навички, отримана на другому етапі; e_i – рівень, до якого підвищується навичка після проходження курсу; c_i – поточний рівень навички працівника (якщо навичка відсутня, він дорівнює 0).

У результаті формується структурований перелік навичок і відповідних курсів, упорядкованих за їхнім пріоритетом для подальшого вивчення.

1.4 Етап 4: підбір рекомендованих курсів працівникам ІТ-компанії

Після отримання проміжних результатів попередніх етапів фінальний етап передбачає використання цієї інформації для формування персональних рекомендацій щодо розподілу курсів. Основним завданням є визначення пріоритетів для співробітників щодо конкретних курсів з метою максимізації ефективності навчання всього персоналу. У цьому випадку враховуються такі фактори: наявні навички співробітника, їхній аналіз, тривалість курсу, кількість осіб, яким цей курс уже рекомендовано, а також максимальна допустима кількість слухачів.

Для формування рекомендацій застосуємо алгоритм контент-орієнтованої рекомендаційної системи [2–5] у поєднанні з підходом багатокритеріальної оптимізації, зокрема методом «зваженої» ідеальної точки [6–9].

Визначимо набір показників, які використовуються як критерії оптимізації:

- максимальна кількість студентів на курсі;
- кількість студентів, які вже записалися;
- кількість працівників, що отримали рекомендацію саме на цей курс;
- тривалість курсу (у годинах);
- пріоритет курсу, розрахований на основі відповідності навичкам.

Проаналізуємо вплив кожного показника на загальний критерій оптимізації.

Окремо показник «максимальна кількість студентів на курсі» не є інформативним без урахування фактичної кількості записаних. Водночас важливим стає співвідношення між кількістю зайнятих і доступних місць, яке

відображає відсоток вільних місць. Саме тому вибираємо критерієм «кількість вільних місць».

Аналогічно, поле «кількість працівників, що отримали рекомендацію саме на цей курс», також має значення лише в поєднанні з кількістю доступних місць. Зокрема важливим є співвідношення кількості рекомендацій до кількості вільних місць: чим більшою є кількість рекомендацій на обмежену кількість місць, тим нижчим має бути пріоритет призначення. Тому другим критерієм вибираємо «співвідношення рекомендацій до вільних місць».

Показники «тривалість курсу» та «пріоритет курсу, визначений на основі навичок», також є критеріями оптимізації і застосовуються у подальших обчисленнях.

Для формування узагальненого критерію оптимізації необхідно здійснити «нормалізацію» вищезгаданих критеріїв, а саме:

- «кількість вільних місць»:

$$k_1^n = \frac{C_{enrolled}}{C_{max}},$$

де C_{max} – максимальна місткість курсу; $C_{enrolled}$ – кількість працівників, що вже записалися на курс;

- «співвідношення рекомендацій до вільних місць»:

$$k_2 = \frac{C_{recommended}}{C_{max} - C_{enrolled}},$$

де $C_{recommended}$ – кількість працівників, яким рекомендовано цей курс.

Через те, що k_2 є критерієм витрат («чим більше, тим гірше»), застосовується функція інверсії та масштабування на інтервал $[0; 1]$ за допомогою експоненціальної функції зворотної експоненти [13]:

$$k_2^n = e^{-k_2/6}.$$

Ця функція вибрана через здатність стискування великих значень до ближчих до нуля, а також плавне спадання на інтервалі k_2 ;

– «тривалість курсу». Через те, що k_3 є критерієм витрат («чим більше, тим гірше»), застосовується обернена нормалізація:

$$k_3^n = \frac{H_{max} - H}{H_{max}},$$

де H – тривалість курсу; H_{max} – тривалість найбільш тривалого курсу;

– «пріоритет курсу, визначений на основі навичок»:

$$k_4^n = \frac{P_{skills}}{P_{skills_max}}$$

де P_{skills} – пріоритет поточного курсу; P_{skills_max} – максимальний пріоритет курсу в списку.

Для формування узагальненої оцінки курсу на основі чотирьох вищезазначених критеріїв застосовується квадратична функція витрат [6]. Оптимальне рішення x^0 визначається як

$$x^0 = \arg \min_{x \in X} \sum_{i=1}^4 w_i \cdot (k_i^* - k_i^n)^2;$$

$$\sum_{i=1}^4 w_i = 1; \quad 0 \leq w_i \leq 1,$$

де w_i – вага i -го критерію; k_i^* – значення ідеальної точки за i -м критерієм; k_i^n – нормалізоване значення i -го критерію для курсу.

Ідеальна точка $k^* = (k_1^*, k_2^*, k_3^*, k_4^*)$ вибирається як така, що відповідає найкращому значенню локальної функції корисності за кожним критерієм:

$$k_i^* = \arg \max_{x \in X} k_i^n(x); \quad i = \overline{1, 4}.$$

Вибір вагових коефіцієнтів w_i здійснюється на основі аналізу значущості кожного критерію для досягнення мети – максимального підвищення професійних навичок працівників та ефективного розподілу курсів.

Таким чином, загальна оцінка курсу визначається як зважена сума квадратів відхилень від ідеальної точки, що дозволяє ефективно ранжувати курси за їхньою придатністю для навчання співробітників.

Висновки

Під час проведеного дослідження запропоновано метод підбору та розподілу рекомендованих навчальних курсів для працівників ІТ-компанії, який дозволяє автоматизувати процес персоналізованого навчання та підвищення професійних навичок співробітників. Метод враховує індивідуальні характеристики працівника, вимоги конкретних проектів і доступність курсів у системі, що забезпечує високий рівень релевантності рекомендацій.

Розроблений підхід передбачає чотири ключові етапи: аналіз резюме працівника, аналіз вимог проектів, аналіз доступних курсів та безпосередній підбір рекомендованих програм.

Кожен етап містить кількісний аналіз відповідності навичок працівника вимогам проекту та визначення пріоритетності розвитку окремих компетенцій.

Застосування словників навичок та їхніх оцінок дозволяє систематизувати інформацію та забезпечує коректне формування списків рекомендованих курсів для кожного співробітника.

На відміну від традиційних рекомендаційних підходів (content-based та collaborative filtering), запропонований метод враховує багатокритеріальність задачі прийняття рішень і дозволяє інтегрувати як індивідуальні характеристики працівника, так і обмеження, пов'язані з ресурсами курсів. Це забезпечує зменшення впливу проблем інформації і холодного старту, а також підвищує якість персоналізації рекомендацій.

Це досягається завдяки використанню додаткових джерел інформації, зокрема інформації про резюме працівників, вимоги щодо проектів і характеристики навчальних курсів. На відміну від класичних підходів колаборативної фільтрації, запропонований метод не залежить лише від історії взаємодії користувачів із системою, що дозволяє формувати рекомендації навіть за умов обмеженої кількості такої інформації.

Крім того, поєднання контент-орієнтованого підходу з багатокритеріальною оптимізацією забезпечує можливість побудови релевантних рекомендацій для нових користувачів і курсів.

Використання методу ідеальної точки дозволяє формалізувати процес вибору оптимального курсу та забезпечити процес пошуку компромісного рішення між суперечливими критеріями, що є визначальною особливістю багатокритеріальних завдань прийняття рішень.

Таким чином, запропонований метод підвищує точність персоналізованих рекомендацій, оптимізує процес навчання та сприяє більш ефективному розвитку професійних компетентностей працівників. Впровадження цього підходу в ІТ-компаніях сприятиме покращенню якості проектної діяльності, збільшенню рівня конкурентоспроможності організації та раціональному використанню освітніх ресурсів.

Література

1. Батраченко В. О., Колесник Л. В. Розробка алгоритму підбору та розподілу рекомендаційних курсів працівникам ІТ-компанії. *Комп'ютерно-інтегровані технології автоматизації технологічних процесів на транспорті та у виробництві*: тези доповіді. Харків. 2023. С. 280–284.
2. Tarnowska K., Ras Z. W., Daniel L. Recommender System for Improving Customer Loyalty. Cham: *Springer International Publishing*, 2020. DOI: 10.1007/978-3-030-13438-9.
3. Recommender System Applications. *Recommender Systems*. 2020. P. 39–62. DOI: 10.1142/9789811224638_0003.
4. Saleh I., Chartron G., Kembellec G. Recommender Systems. Wiley & Sons, Incorporated, John, 2021. 252 p.
5. Agarwal D. K., Chen B. C. Statistical Methods for Recommender Systems. Cambridge University Press, 2015. 284 p. DOI: 10.1017/CBO9781139565868.
6. Adomavicius G., Manouselis N., Kwon Y. Multi-criteria recommender systems. In: *Ricci F., Rokach L., Shapira B., Kantor P.B. (Eds.). Recommender systems handbook*. Boston, Springer, 2011. P. 769–803. DOI: 10.1007/978-0-387-85820-3_24.
7. Determining preferences in recommender systems based on comparator identification technology / V. Beskorovainyi et al. *Innovative Technologies and Scientific Solutions for Industries*. 2022. No. 2 (20). P. 14–21. DOI: 10.30837/ITSSI.2022.20.014.
8. Mathebula J., Mbuli N. Application of TOPSIS for Multi-Criteria Decision Analysis (MCDA). *A Systematic Literature Review. Energies*. 2025. Vol. 18 (13). No. 3478. DOI: 10.3390/en18133478.
9. TOPSIS and AHP-Based Multi-Criteria Decision-Making Approach for Evaluating Redevelopment in Old Residential Projects / C. Park et al. *Sustainability*. 2025. Vol. 17. No. 7072. DOI: 10.3390/su17157072
10. Recommendation System Based on a Compact Hybrid User Model Using Fuzzy Logic Algorithms / N. Khairova et al. *Ceur Workshop Proceedings*. 2025. Vol. 4015. P. 22–31.
11. Hansen S., Narayanan N. H., Hegarty M. Designing Educationally Effective Algorithm Visualizations. *Journal of Visual Languages & Computing*. 2002. Vol. 13. No. 3. P. 291–317. URL: DOI: 10.1006/jvlc.2002.0236.
12. Wengrow J. A. Common-Sense Guide to Data Structures and Algorithms. Pragmatic Bookshelf. 2020. 508 p.
13. Beebe N. H. F. Exponential and logarithm. *The Mathematical-Function Computation Handbook*. Cham, 2017. P. 267–298. DOI: 10.1007/978-3-319-64110-2_10.

References

1. Batrachenko, V.O., Kolesnyk, L.V. (2023). Development of an algorithm for the selection and distribution of recommended courses for employees of IT companies. *Computer-integrated technologies for automation of technological processes in transport and production: theses of reports*. Kharkiv. P. 280–284.
2. Tarnowska, K., Ras, Z. W., Daniel, L. (2020). Recommender System for Improving Customer Loyalty. Cham: *Springer International Publishing*. DOI: 10.1007/978-3-030-13438-9.
3. Recommender System Applications. *Recommender Systems*. 2020, P. 39–62. DOI: 10.1142/9789811224638_0003.
4. Saleh, I., Chartron, G., Kembellec, G. (2021). Recommender Systems. Wiley & Sons, Incorporated, John, 252 p.
5. Agarwal, D. K., Chen, B. C. (2015). Statistical Methods for Recommender Systems. Cambridge University Press, 284 p. DOI: 10.1017/CBO9781139565868.
6. Adomavicius, G., Manouselis, N., Kwon, Y. (2011). Multi-criteria recommender systems. In: *Ricci F., Rokach L., Shapira B., Kantor P.B. (Eds.). Recommender systems handbook*. Boston, Springer, P. 769–803. DOI: 10.1007/978-0-387-85820-3_24.
7. Beskorovainyi, V., Kolesnyk, L., Alokhina, M., Kosenko, V. (2022). Determining preferences in recommender systems based on comparator identification technology. *Innovative Technologies and Scientific Solutions for Industries*, No. 2 (20), P. 14–21. DOI: 10.30837/ITSSI.2022.20.014.
8. Mathebula, J., Mbuli, N. (2025). Application of TOPSIS for Multi-Criteria Decision Analysis (MCDA). *A Systematic Literature Review. Energies*, Vol. 18 (13), No. 3478. DOI: 10.3390/en18133478.
9. Park, C., Son, M., Kim, J., Kim, B., Ahn, Y., Kwon, N. (2025). TOPSIS and AHP-Based Multi-Criteria Decision-Making Approach for Evaluating Redevelopment in Old Residential Projects. *Sustainability*, Vol. 17, No. 7072. DOI: 10.3390/su17157072
10. Khairova, N., Sytnikov, D., Hrebenuk, M., Sytnikova, P. (2025). Recommendation System Based on a Compact Hybrid User Model Using Fuzzy Logic Algorithms. *Ceur Workshop Proceedings*, Vol. 4015, P. 22–31.
11. Hansen, S., Narayanan, N. H., Hegarty, M. (2002). Designing Educationally Effective Algorithm Visualizations. *Journal of Visual Languages & Computing*, Vol. 13, No. 3, P. 291–317. URL: DOI: 10.1006/jvlc.2002.0236.
12. Wengrow, J. A. (2020). Common-Sense Guide to Data Structures and Algorithms. Pragmatic Bookshelf. 508 p.
13. Beebe, N. H. F. (2017). Exponential and logarithm. *The Mathematical-Function Computation Handbook*. Cham, P. 267–298. DOI: 10.1007/978-3-319-64110-2_10.

Колесник Людмила Володимирівна¹, к.т.н., доц., професор кафедри комп'ютерного моделювання та інтелектуальних технологій, тел.: +38 095-122-74-75, liudmyla.kolesnyk@nure.ua, ORCID: 0000-0003-4417-7759,

Батраченко Владислав Олександрович², Senior Java Backend developer, тел.: +38 095-065-16-54, vezyach11@gmail.com, ORCID: 0009-0000-3406-9071,

Колесник Олексій Броніславович¹, к.т.н., старший викладач кафедри комп'ютерного моделювання та інтелектуальних технологій, тел.: +38 095-908-98-76, oleksii.kolesnyk2@nure.ua, ORCID: 0009-0008-5055-8535

¹Харківський національний університет радіоелектроніки, пр. Науки, 14, 61166, м. Харків, Україна,

²ТОВ «ЕПАМ СИСТЕМЗ», вул. Сім'ї Прахових, 54, 01033, м. Київ, Україна.

Development of a method for selecting and distributing recommended courses to IT company employees

Annotation. Modern IT companies face the challenge of efficiently selecting and distributing educational courses that align with the diverse professional needs of their employees. The rapid expansion of digital competencies, combined with the necessity for continuous professional development, makes manual course assignment ineffective and resource-intensive. As a result, there is a growing need for automated and personalized mechanisms capable of identifying relevant training materials that match the employees' current skill levels, career objectives, and professional interests. The primary **goal** of the study is to develop an effective and automated method for selecting and distributing recommended learning courses among employees of an IT company. The proposed approach aims to ensure the relevance, personalization, and optimal allocation of educational resources by taking into account both organizational priorities and individual development trajectories. **Methodology.** The research includes a comprehensive analysis of employees' educational needs and the identification of key criteria for course evaluation. A content-based selection method is proposed, relying on course characteristics, competency requirements, and semantic similarity measures. The methodology integrates a recommendation approach enhanced by optimization techniques, including the ideal point method and utility maximization, which allows the system to automatically rank available courses and assign them to employees based on individualized learning requirements. **Results.** The de-

veloped method enables the automated identification of the most relevant training courses and provides an optimal strategy for distributing them among employees. Experimental evaluation demonstrates that the approach improves the accuracy and relevance of course recommendations, reduces redundancy in training assignments, and enhances the efficiency of the company's educational workflows. **Originality.** The scientific novelty of the research lies in the integration of a content-based recommendation mechanism with an optimization-oriented decision model for course distribution. Unlike traditional recommendation systems, the proposed method not only selects relevant learning materials but also optimizes their allocation by considering both individual employee profiles and global organizational needs. The **practical significance** of the proposed method is manifested in its ability to enhance employee skill development, improve training efficiency, and support strategic workforce planning in IT companies. By automating course selection and distribution, the system reduces the administrative burden on HR and training departments, ensures better use of educational resources, and promotes continuous professional growth.

Keywords: recommendation system, content-based method, personalized learning, skill development, optimization model, ideal point method, utility maximization.

Liudmyla Kolesnyk¹, Phd, Associate Professor, Professor,

tel.: +38 095-122-74-75, liudmyla.kolesnyk@nure.ua, ORCID: 0000-0003-4417-7759,

Vladyslav Batrachenko², Senior Java Backend developer,

tel.: +38 095-065-16-54, vezyach11@gmail.com, ORCID: 0009-0000-3406-9071,

Oleksii Kolesnyk¹, Phd, Senior Lecturer, tel.: +38 095-908-98-76, oleksii.kolesnyk2@nure.ua, ORCID: 0009-0008-5055-8535.

¹Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine.

²«ЕПАМ СИСТЕМЗ», Ltd, Prakhovykh Family Street. 54, Kyiv, 01033, Ukraine.

Стаття надійшла до редакції / Received: 10.01.2026.

Прийнята до друку після рецензування / Revised and Accepted: 21.01.2026.

Дата публікації статті / Published: 11.05.2026.