

DOI: 10.35681/1560-9189.2025.27.3.354582

УДК 004.93

Є. М. Федорченко, А. О. Олійник, М. С. Михайлова, Т. А. Зайко,  
О. О. Степаненко, Ю. В. Федорченко, Т. В. Федорончак

Національний університет «Запорізька політехніка»

вул. Жуковського, 64, 69063 Запоріжжя, Україна

## Інтелектуальні методи керування індивідуальним раціоном харчування

*Розробка персоналізованих дієт набуває дедалі більшої актуальності через зростання інтересу до здорового способу життя та необхідність урахування індивідуальних особливостей організму. Представлено підхід до створення індивідуальних дієт шляхом порівняння чотирьох алгоритмів оптимізації і трьох алгоритмів прогнозування. Запропонована система визначає оптимальний розподіл харчових компонентів на основі індивідуальної добової потреби в калоріях. Роботу алгоритмів оцінено за трьома критеріями: точність результату, швидкість обчислень і кількість можливих варіантів дієти. Результати показали переваги комбінованого підходу, який поєднує методи оптимізації і машинного навчання для формування збалансованої та індивідуально адаптованої дієти.*

**Ключові слова:** жадібний алгоритм, лінійне програмування, метод Монте-Карло, генетичний алгоритм, Random Forest, XGBoost, щільна нейронна мережа, індивідуальна дієта, оптимізація, прогнозування, ІМТ, базальний метаболізм, точність, швидкість, варіативність.

### Вступ

Харчування суттєво впливає на здоров'я, працездатність і якість життя. Балансування раціону є складним завданням, оскільки потребує врахування калорійності, обмежень і уподобань. Автоматизовані системи на основі штучного інтелекту (ШІ) спрощують цей процес, забезпечуючи персоналізовані рекомендації. У даній роботі використано модифікований генетичний алгоритм для оптимізації щоденного раціону з урахуванням уподобаних продуктів. Порівняльний аналіз показав, що поєднання методів оптимізації і машинного навчання забезпечує точне, швидке та різноманітне планування дієт.

© Є. М. Федорченко, А. О. Олійник, М. С. Михайлова, Т. А. Зайко,  
О. О. Степаненко, Ю. В. Федорченко, Т. В. Федорончак

## Аналіз предметної області

Розробка персоналізованого помічника з планування дієти потребує точних розрахунків індивідуальних потреб у харчуванні. Математичні формули на основі ключових фізіологічних показників є основою надійного алгоритму. Система має точно обчислювати добову калорійну потребу, використовуючи стандартизовані показники — індекс маси тіла (ІМТ) і базальний рівень метаболізму (BMR). ІМТ визначається за формулою [1]:

$$BMI = \frac{Weight}{Height^2}, \quad (1)$$

де BMI — індекс маси тіла; Weight — вага людини в кілограмах; Height — зріст у метрах.

Для більш точної оцінки добових енергетичних потреб використовується BMR, який відображає індивідуальні фізіологічні параметри. Серед різних формул BMR [2] найбільш точною є формула Міфліна-Сен Жеора. Для чоловіків визначається формулою [2]:

$$BMR = 88.36 + (13.4 \times Weight) + (4.8 \times (Height \times 100)) - (5.7 \times Age). \quad (2)$$

Для жінок визначається такою формулою [2]:

$$BMR = 447.6 + (9.2 \times Weight) + (3.1 \times (Height \times 100)) - (4.3 \times Age). \quad (3)$$

Дослідження використовує стандартизовані антропометричні показники (ІМТ і BMR). На їхній основі визначається потреба в корекції маси тіла та індивідуальна добова калорійність відповідно до даних користувача.

## Аналіз літературних даних і постановка проблеми

Сучасні дослідження у сфері персоналізованого планування раціону харчування демонструють широкий спектр методів оптимізації і прогнозування. Існуючі системи використовують як класичні алгоритми оптимізації, так і сучасні підходи на основі машинного навчання.

У дослідженні [3] було проведено scoring-огляд застосування методів лінійного програмування для розробки локально адаптованих дієтичних рекомендацій для дітей віком до п'яти років, використовуючи програму Optifood.

Лінійне програмування (LP) — це математичний метод оптимізації, що дозволяє отримати оптимальне рішення (наприклад, мінімальну вартість раціону), дотримуючись низки лінійних обмежень (на кількість макро- та мікронутрієнтів), як зазначено в дієтичних рекомендаціях [4].

Модель передбачала мінімізацію відхилень від рекомендованих норм нутрієнтів з обмеженням енергетичності, вартості та локальної доступності продуктів. Результати показали, що переважна більшість макро- і мікронутрієнтів може бути забезпечена, проте системними «проблемними» залишаються залізо, цинк, кальцій, а також інші мікроелементи залежно від віку дитини (наприклад, фолієва кислота, вітаміни групи B та C). Це підкреслює необхідність комбінації LP-застосувань із додатковими стратегіями.

У роботі Corne van Dooren [4] було проведено ґрунтовний огляд застосування також методу LP для задач оптимізації харчових раціонів. Автор проаналізував понад 50 робіт, у яких класична задача «diet problem» формулювалася у вигляді мінімізації витрат при задоволенні добових норм макро- та мікронутрієнтів. Загальну постановку задачі можна описати за допомогою такого виразу:

$$Z = \sum_{i=1}^n c_i x_i \rightarrow \min, \quad (4)$$

де  $c_i$  — вартість одиниці продукту  $i$ ;  $x_i$  — кількість продукту, що включається в раціон. Проте мінімізація витрат можлива лише за дотримання певних умов. Їх можна виразити системою обмежень:

$$\sum_{i=1}^n a_{ij} x_i \geq b_j, j = 1, 2, \dots, m,$$

$$x_i \geq 0, i = 1, 2, \dots, m, \quad (5)$$

де  $a_{ij}$  — вміст нутрієнта  $j$  в продукті  $i$ ;  $b_j$  — рекомендована добова норма нутрієнта  $j$ . У результаті класичне LP дозволяє отримати математично оптимальні рішення з точки зору економічності та поживності, проте в практичному застосуванні виникають труднощі. Зокрема, сформовані моделі раціонів часто не враховують культурних харчових уподобань і можуть мати низьку прийнятність для користувача. Для подолання цих проблем у ряді досліджень запропоновано вводити додаткові обмеження: екологічний вплив (наприклад, вуглецевий слід), різноманітність продуктів чи соціальні фактори. Автор робить висновок, що лінійне програмування є ефективним і прозорим методом, але для практичного використання воно має поєднуватися із багатокритеріальними підходами.

У роботах [5–7] метод Монте-Карло представлено як стохастичний алгоритм оцінки ймовірнісних характеристик складних систем шляхом повторних випадкових вибірок. Основна ідея полягає в апроксимації математичного очікування функції  $f(x)$  за допомогою  $N$  незалежних симуляційних реалізацій, що можна виразити формулою

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N f(x_i), \quad (6)$$

де  $x_i$  — випадковий вектор вхідних даних, згенерований відповідно до заданого розподілу ймовірностей;  $\hat{\mu}$  — оцінка математичного очікування;  $N$  — кількість симуляційних експериментів;  $f(x_i)$  — значення цільової функції при конкретному наборі вхідних параметрів. У [5] застосування методу дозволило оцінити вплив випадкових і шумних параметрів на результати моделювання, що продемонструвало високу точність при достатньому числі симуляцій, але потребувало значних обчислювальних ресурсів. У роботі [6] описано підхід квазі-Монте-Карло, який підвищує збіжність за рахунок використання детермінованих послідовностей точок, однак його реалізація є більш складною і не завжди універсальною. У [7] метод Монте-Карло було застосовано для моделювання медичних процесів із великою варіативністю параметрів, що дозволило детально оцінити ймовірнісні характеристики, проте збереглася проблема високої обчислювальної вартості та складності валідації моделей.

Таким чином, метод Монте-Карло є ефективним інструментом для моделювання систем із невизначеними параметрами та може бути застосований для оптимізації харчових раціонів у стохастичному середовищі, забезпечуючи прозорість і гнучкість, але потребує оптимізації обчислювальної складності.

У роботі [8] був застосований жадібний алгоритм для формування здорових щоденних меню з урахуванням макро- та мікронутрієнтних обмежень. Алгоритм послідовно обирає страви, які максимізували відповідність заданим нутріційним нормам, забезпечуючи базові вимоги щодо калорійності та харчової цінності. Результати показали, що жадібний підхід є ефективним у швидкому формуванні допустимих варіантів меню, проте відзначено обмеження щодо різноманітності раціонів і необхідність подальшого удосконалення для зменшення одноманітності.

Також у дослідженні [9] жадібний алгоритм використовувався для ініціалізації початкових рішень при плануванні дієт із врахуванням харчових обмежень. Хоча основний фокус був на методах машинного навчання, жадібний підхід дозволив швидко сформувати допустимий стартовий набір меню.

Отже, жадібний алгоритм демонструє ефективність у швидкому формуванні допустимих рішень і початкових наборів меню з урахуванням харчових обмежень. Проте його застосування обмежене у забезпеченні різноманітності та повної персоналізації раціонів, що потребує подальших модифікацій або комбінування з іншими методами оптимізації.

У роботах [10, 11] генетичний алгоритм (ГА) представлено як еволюційний стохастичний метод глобальної оптимізації, що оперує популяцією рішень, генами та операторами селекції, кросоверу та мутації. Основна ідея полягає у послідовному покращенні набору можливих рішень через обчислення значення цільової функції для кожного індивіда та відбору найкращих особин для формування наступного процесу, що можна виразити формулою.

$$x_{t+1} = \text{Mutation} \left( \text{Crossover}(\text{Selection}(x_t)) \right), \quad (7)$$

де  $x_t$  — поточне покоління рішень; Selection, Crossover, Mutation — відповідні оператори ГА. У [10] застосування ГА дозволило знайти оптимальні щоденні раціони з урахуванням калорійних і нутрієнтних обмежень, що забезпечило високу різноманітність допустимих рішень, але вимагало значних обчислювальних ресурсів при великій популяції. У роботі [11] метод удосконалено шляхом адаптивного налаштування ймовірності мутації і поєднання з навчальними моделями, що підвищило швидкість збіжності та точність оптимізації, проте залишилася проблема локальних мінімумів у складних багатовимірних просторах.

Таким чином, ГА є потужним інструментом для глобальної оптимізації складних раціонів харчування, забезпечує гнучкість і різноманітність допустимих рішень, але потребує збалансованого налаштування параметрів для ефективної роботи.

Проте, застосування одного лише алгоритму оптимізації, навіть такого потужного як генетичний алгоритм, часто не забезпечує достатньої точності та адаптивності рішень у складних умовах багатовимірних обмежень. Для підвищення ефективності оптимізації доцільно поєднувати алгоритми з методами прогнозування, що дозволяє передбачати вплив зовнішніх і змінних параметрів на результат опти-

мізації. Таке комбіноване використання забезпечує більшу точність, різноманітність і стабільність кінцевих рішень.

У сучасних дослідженнях методи прогнозування активно використовуються для підвищення точності оптимізації раціонів та оцінки ризиків. Зокрема, в роботі [12] метод Random Forest застосовано для прогнозування ризику розвитку діабету 2 типу на основі змін рівня HbA1c. Модель продемонструвала високу точність порівняно з традиційними регресійними підходами та дозволила виявити важливі клінічні предиктори, які залишалися непоміченими іншими методами. Відмінно від Random Forest, у дослідженні [13] використано XGBoost з байєсівською оптимізацією гіперпараметрів для класифікації ризику захворювань. Підхід скоротив час підбору оптимальних гіперпараметрів і покращив узагальнення моделі, підвищуючи точність прогнозування. Нарешті, у роботі [14] застосовано глибокі нейронні мережі DenseNet для діагностики дефіциту поживних речовин у рисі за зображеннями листя; модель досягла точності 97,44 %, що підкреслює ефективність нейронних мереж у розпізнаванні складних патернів. Таким чином, поєднання цих методів прогнозування може значно підвищити ефективність оптимізаційних алгоритмів у складних системах, забезпечуючи точність, гнучкість і здатність до обробки різноманітних даних.

Аналіз [3–11] показує, що оптимізація харчових раціонів є складним і актуальним завданням, яке потребує використання різних підходів. Класичні алгоритми оптимізації — жадібний метод, лінійне програмування, метод Монте-Карло та генетичні алгоритми — мають свої переваги і обмеження:

— жадібні алгоритми швидко формують допустимі меню, але обмежені у різноманітності;

— лінійне програмування прозоре та кероване, проте потребує модифікацій для збільшення різноманітності раціонів;

— генетичні алгоритми ефективні для глобального пошуку у великих і нелінійних просторах і підвищують різноманітність допустимих раціонів.

Методи прогнозування на основі машинного навчання — Random Forest, XGBoost та нейронні мережі — дозволяють підвищити точність і гнучкість моделей: Random Forest виділяє ключові фактори, XGBoost покращує узагальнення та оптимізацію гіперпараметрів, а нейронні мережі моделюють складні нелінійні взаємозв'язки.

Таким чином, для персоналізованого планування раціону доцільно комбінувати алгоритми оптимізації і методи прогнозування, що дозволяє адаптувати рішення під конкретні дані, підвищити точність і різноманітність раціонів.

## **Мета і задачі дослідження та заплановані результати**

Метою дослідження є розробка і аналіз методів оптимізації персоналізованого раціону харчування із застосуванням класичних та інтелектуальних алгоритмів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

— проаналізувати існуючі алгоритми оптимізації і визначити їхні переваги і обмеження у формуванні раціону;

— дослідити можливості поєднання найкращого алгоритму оптимізації з методами прогнозування для підвищення точності рішень;

— оцінити якість отриманих рішень за критеріями точність, швидкість, продуктивність, різноманітності страв і можливості практичної реалізації.

Заплановані результати дослідження включають створення концептуальної моделі оптимізації раціону харчування, яка поєднує переваги різних підходів і може стати основою для практичних систем підтримки прийняття рішень у дієтології.

## **Розробка і аналіз алгоритмів оптимізації і методів прогнозування раціону харчування з урахуванням індивідуальних харчових уподобань**

Аналіз останніх наукових досліджень свідчить про значну увагу до задач оптимізації раціонів харчування, що зумовлено індивідуальними вподобаннями користувачів і практичними обмеженнями. Існує широкий спектр підходів до вирішення цієї проблеми — від класичних математичних методів до сучасних стохастичних та еволюційних алгоритмів. Серед найбільш поширених і результативних підходів виділяються метод Монте-Карло, жадібні алгоритми та генетичні алгоритми, а для підвищення точності результатів доцільним є поєднання з методами прогнозування на основі машинного навчання. Саме ці методи було обрано для дослідження у межах даної роботи.

Лінійний алгоритм базується на підході цілочисельного лінійного програмування [4], який дозволяє знаходити оптимальний раціон з урахуванням обмежень на кількість прийомів їжі, порції і калорійність. Нехай  $x_i$  позначає кількість порцій страви  $i$ , а  $y_i \in \{0,1\}$  — бінарна змінна, що показує, чи обрана страву. Кожна страву характеризується калорійністю  $cal_i$ .

Для забезпечення узгодженості вибору порцій і наявності страви вводиться наступне обмеження:

$$y_i \leq x_i \leq M * y_i, \quad i = 1, \dots, n, \quad (8)$$

де  $M$  — максимальна кількість порцій однієї страви. Цей зв'язок гарантує, що обрана страву має щонайменше одну порцію, а необрана — нуль порцій.

Відповідно до харчових правил формулюються обмеження на структуру прийомів їжі. Наприклад, якщо множини  $B$ ,  $L$  та  $D$  містять індекси страв для сніданку, обіду та вечері відповідно, то

$$\sum_{i \in B} y_i = 1, \sum_{i \in L} y_i = 1, \sum_{i \in D} y_i = 1, \quad (9)$$

де  $y_i$  — бінарна змінна, що набуває значення 1, якщо страву  $i$  включена до раціону, і 0, якщо вона відсутня;  $B$  — множина індексів можливих страв для сніданку;  $L$  — множина індексів можливих страв для обіду;  $D$  — множина індексів можливих страв для вечері. Таким чином, умови (9) забезпечують, що у сформованому раціоні буде обрано рівно одну страву для кожного з основних прийомів їжі. Це обмеження відповідає стандартним принципам структурування добового харчування та дозволяє уникнути випадків, коли користувачу пропонується кілька «сніданків» або повна їхня відсутність. Для забезпечення збалансованості добового раціону, окрім основних прийомів їжі (сніданку, обіду та вечері), враховуються також додаткові

прийоми — перекуси. Щоб уникнути надмірної кількості вибраних страв, формулюються додаткові обмеження:

$$\sum_{i \in S} y_i \leq 2, \sum_{i=1}^n y_i \leq 5, \quad (10)$$

де  $S$  — множина індексів страв, які можуть виступати в ролі перекусів;  $n$  — загальна кількість страв у базі даних. Таке формулювання (10) забезпечує раціональну структуру меню: воно не стає надмірно складним (надто багато страв), але водночас допускає можливість включення додаткових перекусів у межах розумних лімітів.

Для врахування цільової калорійності  $G$  вводиться допоміжна змінна  $\delta \geq 0$ , яка характеризує відхилення фактичної калорійності від бажаного значення. Таким чином, умови на калорійність страв у раціоні можна записати як

$$\sum_{i=1}^n cal_i x_i - G \leq \delta, \quad G - \sum_{i=1}^n cal_i x_i \leq \delta. \quad (11)$$

де  $x_i$  — бінарна змінна, що показує включення страви до раціону (1 — включена, 0 — відсутня);  $cal_i$  — калорійність страви  $i$ ;  $G$  — бажана добова калорійність;  $\delta$  — допоміжна змінна, що показує допустиме відхилення калорійності раціону від цільового значення. Мета алгоритму полягає в мінімізації  $\delta$ , тобто у знаходженні раціону, який максимально наближений до заданої добової калорійності.

Додатково застосовується механізм заборони повторних комбінацій страв, що дозволяє отримати декілька різних оптимальних варіантів раціону. Формально це можна виразити через систему додаткових обмежень. Для вже знайденого раціону  $\{x_i^*\}$  наступні рішення повинні задовольняти такі умови:

$$\sum_{i \in I} 1_{\{x_i = x_i^*\}} \leq |I| - 1, \quad (12)$$

де  $I$  — множина індексів страв у попередньому рішенні;  $1_{\{x_i\}}$  — індикаторна функція. Це забезпечує різноманітність планів харчування, не змінюючи при цьому оптимальність кожного окремого рішення.

Лінійний підхід доцільний для задач з обмеженою кількістю страв, оскільки він гарантує глобально оптимальний результат і дозволяє точно врахувати харчові вимоги користувача. Як показано у дослідженнях [3, 4], подібні методи є ефективними при формуванні раціонів і ресурсному плануванні.

Після застосування лінійних методів оптимізації, які дозволяють отримати точне рішення задачі у випадках чітко визначених обмежень і цільових функцій, постає потреба у використанні більш гнучких алгоритмів, здатних працювати в умовах невизначеності або великої розмірності простору рішень. Одним із таких підходів є метод Монте-Карло, що базується на стохастичному моделюванні та використанні випадкових вибірок для наближеного пошуку оптимальних рішень.

Метод Монте-Карло є стохастичним підходом до оптимізації, який ґрунтується на багаторазовій генерації випадкових кандидатів і оцінці їхньої якості за допомогою цільової функції [15]. У контексті побудови раціону харчування кожен кандидатський раціон складається з обов'язкових прийомів їжі — сніданку, обіду та вечері — та додаткових перекусів. Кількість перекусів може змінюватися від нуля до двох, при цьому допускається повторення однієї і тієї ж страви, що дозволяє отримати більшу кількість порцій певного продукту.

Нехай  $x_i$  — кількість порцій страви  $i$ , а  $C_i$  — її калорійність. Загальна калорійність раціону обчислюється як

$$T(x) = \sum_{i=1}^n C_i x_i. \quad (13)$$

Метою є мінімізація відхилення від заданої добової калорійності  $G$ , що виражається формулою

$$\delta = |T(x) - G| \quad (14)$$

На кожній ітерації алгоритм випадковим чином обирає базові страви та формує початковий раціон, враховуючи пріоритети користувача щодо улюблених інгредієнтів. Потім застосовується локальний жадібний пошук (hill-climb) для корекції порцій: на кожному кроці порції кожної страви можуть збільшуватися або зменшуватися на одиницю, якщо це наближає загальну калорійність до цільового значення.

Процес багаторазової генерації випадкових раціонів і застосування локального пошуку дозволяє отримати популяцію  $N$  кандидатів, яку можна ранжувати за критерієм наближеності до калорійної цілі. Це можна подати у вигляді

$$rank(x^{(k)}) = |T(x^{(k)}) - G|, \quad (15)$$

де  $x^{(k)}$  —  $k$ -й кандидат у популяції (згенерований раціон харчування);  $T(x^{(k)})$  — розрахована калорійність  $k$ -го раціону;  $rank(x^{(k)})$  — значення критерію, що відображає відхилення калорійності згенерованого раціону від цільового значення;  $G$  — задана цільова добова норма калорій. Після сортування обираються найкращі  $M$  раціонів, що забезпечує як варіативність рішень, так і їхню відповідність індивідуальним харчовим уподобанням.

Перевагою методу є його здатність працювати у великих і складних просторах рішень, де класичні лінійні методи можуть бути обмежені [16].

Метод Монте-Карло забезпечує гнучкість у побудові множини рішень, дозволяючи одночасно враховувати як харчові обмеження, так і особисті уподобання користувача, що робить його ефективним інструментом у персоналізованому плануванні харчування.

Жадібний алгоритм оптимізації раціону харчування ґрунтується на поетапному виборі локально найкращого продукту з точки зору наближення до заданих обмежень і цільових показників. На відміну від лінійних методів, що працюють з усім простором допустимих рішень одночасно, жадібний підхід формує розв'язок поступово, додаючи на кожному кроці лише один елемент.

Основна ідея полягає в тому, що на кожній ітерації розраховується «корисність» продукту, яка визначається співвідношенням між вмістом поживних речовин і калорійністю. Формально таку корисність можна подати як

$$U_i = \frac{\sum_{j=1}^m \omega_j * p_{ij}}{c_i}. \quad (16)$$

де  $U_i$  — коефіцієнт «корисності»  $i$ -го продукту;  $p_{ij}$  — вміст  $j$ -ї поживної речовини у продукті;  $\omega_j$  — ваговий коефіцієнт, що визначає значущість цієї речовини для індивідуальної дієти;  $c_i$  — калорійність продукту. Таким чином, алгоритм віддає

перевагу тим продуктам, які забезпечують максимальний приріст поживних речовин за одиницю енергетичної цінності.

Особливістю жадібного підходу є те, що він забезпечує швидке отримання результату, оскільки не потребує перебору великої кількості комбінацій. Проте його недолік полягає в ризику потрапляння в локальний оптимум: ранній вибір продуктів із високим коефіцієнтом корисності може обмежити можливість досягти більш збалансованого рішення в подальшому. Саме тому жадібні методи часто використовуються як початкове наближення до розв'язку, що потім може бути вдосконалене за допомогою більш потужних оптимізаційних підходів, зокрема генетичних алгоритмів або методів Монте-Карло.

Генетичний алгоритм, використаний для оптимізації раціону харчування, наслідує принципи біологічної еволюції, моделюючи процес природного відбору [17]. У цьому підході кожен можливий план харчування інтерпретується як індивід популяції, де його «генами» виступають окремі страви та кількість їхніх порцій. Первинна популяція формується випадково, однак з урахуванням індивідуальних харчових уподобань користувача, що дозволяє від самого початку зменшити простір пошуку до більш релевантних комбінацій.

Якість кожного індивіда оцінюється за допомогою функції пристосованості [17]. Вона визначається відхиленням добової калорійності від цільового значення:

$$F(x) = |\text{Calories}(x) - C_{goal}| \quad (17)$$

де  $\text{Calories}(x)$  — сумарна калорійність індивіда  $x$ ;  $C_{goal}$  — задана добова норма калорій. Чим меншим є значення  $F(x)$ , тим краще збалансованим вважається план харчування.

На кожній ітерації (поколінні) відбувається відбір найкращих індивідів за принципом мінімізації  $F(x)$ . Потім застосовуються оператори генетичного пошуку: схрещування та мутація. Схрещування реалізується як комбінування частини раціону одного «батька» із частиною раціону іншого, тобто:

$$\text{Child} = \{x_1, x_2, \dots, x_k\} \cup \{y_{k+1}, \dots, y_n\} \quad (18)$$

Мутація, у свою чергу, полягає у випадковій заміні однієї зі страв або зміні її кількості порцій. Імовірність мутації задається параметром  $\mu \in [0,1]$ . Такий підхід дозволяє уникнути передчасної збіжності алгоритму до локального мінімуму, оскільки вносить у популяцію додаткове різноманіття.

Процес повторюється протягом фіксованої кількості поколінь  $G$ , після чого найкращим вважається індивід із мінімальним значенням  $F(x)$ . Важливо, що завдяки включенню харчових уподобань користувача до процедури генерації індивідів, алгоритм не лише оптимізує калорійність, а й формує розв'язки, які є реалістичними з точки зору вподобань і звичок конкретної особи.

У науковій літературі неодноразово підтверджено ефективність генетичних алгоритмів у розв'язанні комбінаторних задач з великою кількістю обмежень, зокрема у сферах планування і оптимізації навчальних процесів. Так, у роботі [18] представлено модифікований генетичний алгоритм для автоматичного складання розкладу університетських занять, що дозволяє враховувати численні обмеження та знаходити ефективні рішення в умовах складних і багатокритеріальних задач.

Використання такого підходу в задачі оптимізації раціону харчування дозволяє знайти баланс між точністю відповідності дієтичним нормам і збереженням індивідуальних харчових вподобань.

Для системного порівняння ефективності алгоритмів оптимізації було сформовано табл. 1, яка демонструє ключові показники роботи чотирьох методів: лінійного програмування, жадібного алгоритму, методу Монте-Карло та звичайного генетичного алгоритму. Вибір критеріїв порівняння включає час виконання пошуку раціонів, кількість згенерованих варіантів, продуктивність, а також середнє та найкраще відхилення від цільової калорійності.

Таблиця 1. Порівняння алгоритмів оптимізації за ключовими критеріями

Критерій	LP	Жадібний алгоритм	Метод Монте-Карло	ГА
Час виконання (сек)	69,6	1,9	0,9	7,2
Кількість згенерованих варіантів	5	11	14	850
Продуктивність (варіанти/сек)	0,071	5,79	15,55	118,06
Середнє відхилення від цільової калорійності	1,75	11,9	1,75	7,4
Відхилення найкращого раціону від цільової калорійності	1,65	8,9	1,65	1,65

Аналіз даних табл. 1 дозволяє зробити низку важливих висновків щодо ефективності розглянутих алгоритмів оптимізації. По-перше, за показником часу виконання пошуку раціонів, лінійний алгоритм демонструє найбільше значення, що зумовлено детальним перебором усіх можливих комбінацій, тоді як метод Монте-Карло характеризується мінімальним часом виконання. Жадібний алгоритм і звичайний генетичний алгоритм показують проміжні значення, при цьому час виконання ГА залишається досить прийнятним для практичного застосування.

По-друге, аналіз кількості згенерованих варіантів свідчить, що генетичний алгоритм забезпечує значно ширшу популяцію рішень (850 варіантів), що підвищує ймовірність знаходження оптимального або близького до оптимального раціону. Лінійний алгоритм та метод Монте-Карло формують мінімальну кількість варіантів (5 та 14 відповідно), що обмежує їхню здатність до гнучкої персоналізації раціону. Жадібний алгоритм, хоча і швидкий, також формує невелику кількість варіантів (11), що робить його менш придатним для завдань комплексної оптимізації.

Що стосується продуктивності, виміряної як кількість варіантів на секунду, генетичний алгоритм значно перевищує всі інші методи (118,06 варіантів/сек), що свідчить про його високу ефективність при обробці великої кількості комбінацій. Метод Монте-Карло та жадібний алгоритм демонструють середню продуктивність, а лінійний алгоритм залишається найменш продуктивним.

За критерієм середнього відхилення від цільової калорійності, лінійний алгоритм і метод Монте-Карло показують найкращий результат (1,75), тоді як ГА демонструє трохи більші значення (7,4), що пояснюється більшою варіативністю

сформованих раціонів. Жадібний алгоритм показує найгірший результат (11,9), що підтверджує його обмеженість для задач персоналізації.

Нарешті, відхилення найкращого раціону від цільової калорійності практично однакове для методів Монте-Карло, лінійного алгоритму та ГА (1,65), що свідчить про те, що навіть при великій кількості сформованих варіантів генетичний алгоритм здатний забезпечити високоточне наближення до бажаної добової калорійності. Жадібний алгоритм у цьому випадку відстає, демонструючи значно більше відхилення.

Отже, на основі проведеного порівняння алгоритмів оптимізації як проміжне рішення визначено, що генетичний алгоритм є найбільш ефективним для подальшої роботи. Саме його доцільно використовувати у поєднанні з методами прогнозування для підвищення точності та гнучкості підбору персоналізованого раціону харчування.

Разом з тим, для підвищення рівня персоналізації та точності рекомендацій доцільним є поєднання обраного методу оптимізації з методами прогнозування. Це забезпечує врахування індивідуальних особливостей користувача, прогнозування потреб організму та адаптацію раціону до динамічних змін у харчових вподобаннях і фізіологічних параметрах.

Першим досліджуваним методом прогнозування є багатошарова повнозв'язна нейронна мережа (MLP). Цей підхід належить до класу штучних нейронних мереж прямого поширення сигналу та характеризується наявністю одного або кількох прихованих шарів із нелінійними функціями активації. Завдяки здатності апроксимувати складні багатовимірні залежності MLP може ефективно прогнозувати потреби користувача у поживних речовинах і калорійності на основі вхідних даних, таких як антропометричні показники, рівень фізичної активності та індивідуальні харчові вподобання.

Багатошарова повнозв'язна нейронна мережа (також відома як багатошаровий перцептрон, MLP) — це тип штучної нейронної мережі, у якій кожен нейрон одного шару з'єднаний з кожним нейроном наступного шару. Така архітектура дозволяє моделювати складні, нелінійні взаємозв'язки в даних [19].

Альтернативним підходом до прогнозування, що продемонстрував високу ефективність у задачах класифікації та регресії з багатовимірними вхідними даними, є метод Random Forest.

Random Forest — це ансамблевий метод, що базується на побудові множини рішень у вигляді дерев рішень. Кожне дерево в ансамблі приймає рішення щодо того, чи буде користувач схвалювати певний раціон, на основі ознак страв та інгредієнтів [20]. Фінальний прогноз формується як середнє значення прогнозів усіх дерев у лісі.

Класична формула прогнозу Random Forest (19):

$$\hat{y} = \frac{1}{K} \sum_{k=1}^K T_k(x_{mealplan}), k = 1, \dots, K, \quad (19)$$

де  $T_k$  —  $k$ -те дерево в ансамблі;  $x_{mealplan}$  — вхідний вектор, який описує раціон користувача (включає векторизацію інгредієнтів і назв страв);  $K$  — кількість дерев. Древа рішень формуються на випадкових підвбірках даних із застосуванням bootstrap, а на кожному вузлі відбувається розщеплення за ознакою, що найкраще

мінімізує функцію втрат. Найпоширенішим критерієм для бінарної класифікації є індекс Джині:

$$Gini = 1 - \sum_{c=1}^C p_c^2, \quad (20)$$

де  $p_c$  — частка класу  $c$  серед зразків у вузлі. XGBoost (Extreme Gradient Boosting) — це метод градієнтного бустингу, який формує послідовність дерев рішень, кожне з яких намагається скоригувати помилки попередніх дерев [21]. Такий підхід дозволяє ефективно моделювати складні залежності між інгредієнтами страв та уподобаннями користувача.

Прогноз XGBoost обчислюється за формулою

$$\hat{y}_i = \sum_{t=1}^T f_t(x_i), \quad f_t \in \mathcal{F}, \quad (21)$$

де  $T$  — загальна кількість дерев;  $x_i$  — вектор вхідного раціону;  $f_t$  — функція, що описує  $t$ -те дерево в ансамблі. Для навчання дерев використовується регуляризована функція втрат:

$$\mathcal{L} = \sum_{i=1}^N l(y_i \hat{y}_i) + \sum_{t=1}^T \Omega(f_t), \quad \Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^J w_j^2, \quad (22)$$

де  $l(y_i \hat{y}_i)$  — базова функція втрат (наприклад, бінарна крос-ентропія);  $w_j$  — вага  $j$ -го листа дерева;  $\gamma, \lambda$  — параметри регуляризації;  $J$  — кількість листів у дереві. Аналіз даних табл. 2 дозволяє детально оцінити переваги та недоліки кожного із методів прогнозування.

Таблиця 2. Порівняння методів прогнозування за ключовими критеріями

Критерій	MLP	Random Forest	XGBoost
Час навчання, секунди	3,1	0,275	0,187
Час сортування, секунди	0,335	0,304	0,291
Використання оперативної пам'яті, Мб	0,012	0,0552	0,1355

Аналіз табл. 2 показує, по-перше, за критерієм часу навчання найшвидшим є алгоритм XGBoost, який забезпечує високу швидкість завдяки використанню бустингової стратегії та ефективних механізмів оптимізації. Random Forest також показує хороші результати з низьким часом навчання. У свою чергу, базовий багатосаровий перцептрон (MLP) демонструє найбільший час навчання (3,1 с), що пов'язано з більш складною структурою моделі та потребою у численних ітераціях для досягнення збіжності. Проте навіть цей час залишається прийнятним для практичних застосувань, оскільки вимірюється в секундах.

По-друге, за показником часу сортування всі три методи демонструють близькі результати. Це свідчить про те, що на етапі класифікації і прогнозування різниця між ними практично не впливає на швидкість роботи системи.

Найбільш суттєві відмінності проявляються у використанні оперативної пам'яті. Тут MLP показує найкращий результат (0,012 Мб), що є значно нижчим порівняно з Random Forest (0,0552 Мб) та XGBoost (0,1355 Мб). Цей критерій є ключовим у контексті даного дослідження, оскільки система має обробляти великі обсяги даних при персоналізації харчових раціонів. Мінімальне використання па-

м'яті забезпечує можливість масштабування та стабільну роботу без перевантажень.

Основною перевагою MLP у контексті даного дослідження є мінімальне використання оперативної пам'яті, що є критично важливим для обробки великого обсягу даних у персоналізованому підборі раціону харчування.

## Модифікований алгоритм оптимізації і метод прогнозування

У рамках дослідження створено модифікований генетичний алгоритм (ГА), побудований на основі елітарної моделі (elitist GA) із доменно-специфічними правилами генерації і мутації індивідів. Базовий ГА передбачає бінарне кодування хромосом, випадкову ініціалізацію популяції, класичні оператори схрещування та мутації. Запропонований підхід відходить від класичної моделі, зберігаючи загальну еволюційну парадигму, проте застосовуючи адаптацію під предметну область — формування збалансованих харчових планів.

Кожен індивід  $I$  у популяції  $P$  репрезентується у вигляді плану харчування (MealPlan), тобто впорядкованої множини страв із відповідними порціями, що можна відобразити виразом

$$I = \{(d_1, s_1), (d_2, s_2), \dots, (d_k, s_k)\}, k \in \{3, 4, 5\}, \quad (23)$$

де  $d_i$  — страва, вибрана з множини доступних (сніданки, обіди, вечері, перекуси),  $s_i \in \{1, 2, \dots, 5\}$  — кількість порцій. Тоді загальна калорійність індивіда визначається формулою

$$C(I) = \sum_{i=1}^k s_i * c(d_i), \quad (24)$$

де  $c(d_i)$  — калорійність однієї порції страви  $d_i$ . Метою оптимізації є мінімізація відхилення від заданої калорійної мети  $C^*$ . Виходячи з цього, функція пристосованості формується як

$$f(I) = |C(I) - C^*|, \quad (25)$$

Таким чином, чим менше значення  $f(I)$ , тим кращим вважається індивід.

Першою модифікацією була адаптована під завдання ініціалізація популяції. На відміну від класичного ГА, де початкова популяція формується випадковим чином, у даному алгоритмі застосовується доменно-специфічна ініціалізація. При створенні індивіда з імовірністю 0,5 обираються страви, що відповідають індивідуальним уподобанням користувача, тобто

$$P(d \in D_{fav}) = 0,5, P(d \in D \setminus D_{fav}) = 0,5, \quad (26)$$

де  $D_{fav}$  — множина страв, що містять інгредієнти з переліку преференцій користувача. Другою модифікацією було використано елітний відбір, що передбачає формування нової популяції на основі найбільш придатних особин поточного покоління. Це можна виразити формулою

$$P' = \operatorname{argmin}_{I \in P} f(I), |P'| = \frac{1}{2} |P|, \quad (27)$$

де  $P$  — поточна популяція, яка складається з набору рішень (раціонів);  $f(I)$  — цільова функція, яка оцінює якість кожного рішення  $I$  (наприклад, відхилення від ці-

льової калорійності);  $\operatorname{argmin}_{I \in P} f(I)$  — означає вибір тих особин із популяції, для яких значення функції  $f(I)$  є мінімальним, тобто найближчих до оптимального рішення;  $P'$  — нова популяція, яка формується з найкращих особин;  $|P'| = \frac{1}{2}|P|$  — до нової популяції переходить рівно половина особин з початкової, причому саме та половина, яка має кращі показники (найменше відхилення від цільової функції). На відміну від рулетки чи турнірної селекції, даний метод гарантує збереження найбільш пристосованих особин.

Третя модифікація стосується операції схрещення. Вона була адаптована під структуру індивідів у задачі формування раціону. Для цього застосовано односточкове схрещення: беруться два батьківські рішення, обирається випадкова точка розділу, і новий індивід формується як комбінація першої частини одного батька та другої частини іншого. Це дозволяє природним чином поєднувати страви та порції з різних рішень, створюючи більш різноманітні та потенційно кращі раціони.

Четверта модифікація стосується процесу мутації, який у генетичних алгоритмах відповідає за випадкове внесення змін у поточні рішення, щоб уникнути застою і дати змогу дослідити нові варіанти. Вона може відбуватись у двох варіантах:

1) зміна кількості порцій — наприклад, для певної страви замінюється кількість порцій на іншу (від 1 до 5). Це дозволяє гнучко підлаштовувати калорійність під задану ціль;

2) заміна страви — вибрана страву у плані харчування може бути замінена на іншу зі своєї категорії (сніданок, обід, вечеря чи перекус). При цьому алгоритм контролює, щоб у раціоні не було повторів одних і тих самих страв, тобто унікальність завжди зберігається.

Таким чином, алгоритм є елітним генетичним алгоритмом з модифікаціями на етапах ініціалізації, селекції, схрещування та мутації, які враховують предметну специфіку задачі (формування збалансованих харчових планів). Що дозволяє підвищити якість рішень за рахунок скорочення простору пошуку завдяки уподобанням користувача та збереженню найкращих індивідів на кожному кроці при адаптованих операторах мутації, що запобігають деградації рішень.

Для оцінки ефективності алгоритмів оптимізації і модифікованого ГА алгоритму доцільно виконати їхнє порівняння за низкою критеріїв, зокрема точністю та кількістю отриманих результатів, швидкістю і продуктивністю. Таке порівняння дозволяє визначити сильні та слабкі сторони кожного підходу, а також відобразити покращення результату для модифікованого алгоритму оптимізації.

Аналіз показав, що різні алгоритми оптимізації значно відрізняються за ефективністю. У табл. 3 можна побачити це порівняння.

Порівняння за критерієм часу виконання показало суттєву різницю між алгоритмами. Це відображено в табл. 3. Найбільш тривалий процес спостерігається у випадку застосування лінійного програмування — 69,6 с, що пояснюється необхідністю врахування значної кількості обмежень і змінних. Жадібний алгоритм показав значно вищу швидкість — 1,9 с, проте його ефективність обмежується локально оптимальними рішеннями. Метод Монте-Карло виявився найшвидшим — 0,9 с завдяки випадковому генеруванню рішень, однак якість таких результатів не завжди є стабільною.

Таблиця 3. Порівняння алгоритмів оптимізації за критеріями ефективності

Критерії ефективності	Лінійне програмування	Монте-Карло	Модифікований генетичний алгоритм	Жадібний алгоритм
Час виконання	69,6 секунди	0,9 секунди	5,5 секунди	1,9 секунди
Кількості згенерованих варіантів	5	14	1000	11
Продуктивність алгоритмів	0,071 варіантів/с	15,55 варіантів/с	181,81 варіантів/с	5,79 варіантів/с
Відхиленням від цільової калорійності в середньому	1,75 калорій	1,75 калорій	7,1 калорій	11,9 калорій
Відхиленням найкращого раціону від цільової калорійності	1,65 калорій	1,65 калорій	8,9 калорій	1,65 калорій

Модифікований генетичний алгоритм за швидкістю поступається жадібному методу та Монте-Карло (5,5 с), проте забезпечує значно кращу якість оптимізації завдяки еволюційному відбору та здатності працювати з великим простором рішень.

Одним із важливих критеріїв ефективності алгоритмів оптимізації є кількість згенерованих варіантів рішень, що відображає різноманітність можливих конфігурацій оптимального раціону. Цей показник є ключовим при формуванні персоналізованих рекомендацій, оскільки більша кількість варіантів забезпечує ширші можливості для вибору відповідно до індивідуальних уподобань користувача. Аналіз кількості згенерованих варіантів, що видно в табл. 3, свідчить, що модифікований генетичний алгоритм демонструє найвищу здатність до генерації варіативних рішень. Завдяки застосуванню еволюційних операторів — кросоверу, мутації і селекції — він формує значно більший простір можливих комбінацій, що дозволяє знаходити як локально, так і глобально оптимальні варіанти. У кількісному вимірі модифікований ГА згенерував 1000 варіантів, тоді як інші методи показали набагато скромніші результати: лінійне програмування — 5 варіантів, жадібний алгоритм — 11, метод Монте-Карло — 14. Такий розрив на користь ГА свідчить про його високу гнучкість і придатність до роботи з багатовимірними задачами, де важливе значення має різноманітність можливих рішень. У практичному контексті підбору раціону це означає, що користувач отримує ширший набір персоналізованих пропозицій, що підвищує якість і корисність рекомендацій.

Ще одним ключовим критерієм оцінки є продуктивність алгоритмів, яку доцільно вимірювати через кількість згенерованих варіантів за одиницю часу. Такий підхід дозволяє оцінити ефективність використання обчислювальних ресурсів і практичну придатність алгоритмів для задач формування раціону харчування в режимі реального часу. Отримані результати продуктивності алгоритмів оптимізації, які відображені в табл. 3, підтверджують, що модифікований генетичний алгоритм демонструє найвищу продуктивність серед усіх розглянутих методів. Його показник становить 181,81 варіантів за секунду, що у десятки разів перевищує результати інших підходів. Для порівняння: метод Монте-Карло забезпечив лише 15,55 варіантів/сек, жадібний алгоритм — 5,79, а лінійне програмування — всього 0,071 варіанта/сек. Така ситуація склалася тому, що у лінійного алгоритму час виконання

великий і відповідно маленька кількість генерації варіантів. Висока продуктивність ГА пояснюється еволюційними механізмами кросовера та мутації, які дозволяють ефективно досліджувати простір рішень, зберігаючи при цьому різноманітність популяції. Натомість традиційні методи, що ґрунтуються на жорсткій детермінованій логіці, формують значно меншу кількість альтернатив, що обмежує їхнє застосування у практичних сценаріях, де важлива швидка генерація великої кількості персоналізованих варіантів.

Одним із ключових показників ефективності алгоритмів формування оптимального раціону є точність досягнення цільових значень добової калорійності. Для цього аналізується середнє відхилення отриманих раціонів від заданої норми калорійності, що дозволяє оцінити збалансованість рішень і їхню практичну придатність. Результати цього порівняння, подані в табл. 3, демонструють різну здатність алгоритмів до досягнення точності щодо цільової калорійності. Лінійний метод програмування та метод Монте-Карло показали найкращі результати, забезпечивши мінімальне середнє відхилення від заданого значення, яке становить лише 1,75 калорії. Це свідчить про їхню високу здатність до точного відтворення заданих параметрів. Модифікований генетичний алгоритм, хоча й продемонстрував дещо більший показник (7,1 калорії), залишається достатньо точним, особливо з огляду на його здатність генерувати велику кількість альтернативних варіантів і балансувати між точністю та різноманітністю рішень. Натомість жадібний алгоритм виявився найменш ефективним за цим критерієм, показавши середнє відхилення у 11,9 калорій, що вказує на його обмеженість у досягненні калорійної збалансованості та пояснює недостатню універсальність цього підходу у задачах персоналізованого підбору раціону.

І останній критерій — відхилення найкращого згенерованого раціону від цільової добової калорійності. Цей показник дозволяє оцінити, наскільки алгоритм здатний знайти оптимальний варіант раціону, максимально наближений до заданої норми калорійності, незалежно від загальної кількості згенерованих варіантів. Результати порівняння, які видно в табл. 3, демонструють, що модифікований генетичний алгоритм, разом із лінійним методом програмування та методом Монте-Карло, забезпечує мінімальне відхилення від цільової калорійності для найкращого раціону, що дорівнює 1,65 калорій. Це підтверджує високу точність цих підходів у пошуку оптимальних рішень для формування збалансованого раціону. У той же час жадібний алгоритм показав значно більший показник (8,9 калорій), що обмежує його ефективність у задачах, де ключовим є дотримання заданої калорійності.

Отримані дані свідчать про те, що модифікований генетичний алгоритм поєднує переваги точності та значної різноманітності варіантів, що робить його найбільш придатним для задач персоналізованого підбору раціону харчування.

Для підвищення точності й адаптивності формування персоналізованого раціону харчування, разом із модифікацією генетичного алгоритму оптимізації, було виконано модифікацію методу прогнозування. Це дозволяє більш коректно враховувати індивідуальні харчові уподобання.

Зокрема, було застосовано кастомну попередню обробку даних, що дозволила більш гнучко враховувати особливості вхідних ознак. Крім того, архітектура моделі була розширена за рахунок двох embedding-шарів (замість традиційного одного), що забезпечило глибше представлення дискретних ознак. Додатково було

реалізовано роботу з embedding-векторами у цілому, а не з окремими елементами, що дало змогу краще зберігати семантичні зв'язки між інгредієнтами. Також для входу використовувалася комбінація інгредієнтів і назв страв у вигляді єдиного векторного подання, тоді як зазвичай такі ознаки обробляються окремо. Для підвищення зручності навчання та моніторингу процесу було інтегровано власний Logger із кастомними повідомленнями, що дозволяло контролювати ключові параметри та результати на кожному етапі. Завдяки цим модифікаціям вдалося адаптувати MLP до специфіки задачі та досягти кращої точності прогнозування.

MLP складається з вхідного шару, одного або кількох прихованих шарів і вихідного шару. Кожен нейрон виконує лінійне перетворення своїх вхідних даних, після чого застосовує нелінійну активаційну функцію. Це дозволяє мережі навчатися розпізнавати складні шаблони та взаємозв'язки.

Вхідний шар складається із двох компонент: векторизації інгредієнтів і назв страв. Векторизація інгредієнтів — шар вбудованих представлень (*ingredient embedding*) перетворює індекси інгредієнтів на вектори розмірності *embedding*. Обчислення середнього значення векторів інгредієнтів забезпечує узагальнене представлення для кожної страви. Векторизація назв страв — аналогічно, шар *dish\_name\_embedding* перетворює індекси назв страв у вектори розмірності *embedding*. Отримані вектори об'єднуються шляхом конкатенації, формуючи вхідний вектор розмірності, який передається до прихованого шару розраховується за формулою

$$\text{embedding\_dim} = 2 \times \text{embedding}, \quad (28)$$

де *embedding\_dim* — вхідний вектор розмірності; *embedding* — вектор розмірності. Прихований шар є повнозв'язним і містить 64 нейрони. Кожен нейрон приймає вхідні дані з усіх нейронів попереднього шару. Лінійне перетворення вхідного вектора здійснюється через оператор Linear з наступною функцією активації ReLU, яка розраховується за формулою

$$h = \text{ReLU}(W_x + b), \quad (29)$$

де *h* — вектор виходів прихованого шару; *b* — вектор зміщення; *x* — вхідний вектор; *W* — матриця ваг розмірності, яка розраховується за формулою

$$W = (64 \times 2\text{embedding\_dim}). \quad (30)$$

Вихідний шар представлений одним нейроном, що приймає вхідні сигнали від кожного з 64 нейронів прихованого шару. Лінійне перетворення з функцією активації Sigmoid перетворює сумарний сигнал у значення ймовірності, яке розраховується так:

$$y = \sigma(W_o h + b_o), \quad (31)$$

де *y* — значення ймовірності; *W<sub>o</sub>* — ваги вихідного шару; *b<sub>o</sub>* — зміщення;  $\sigma(x)$  — матриця ваг розмірності, яка розраховується за формулою

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (32)$$

Вихідний нейрон прогнозує ймовірність позитивної оцінки раціону, що дозволяє класифікувати раціон як схвалений чи не схвалений.

Для аналізу ефективності модифікованого методу прогнозування було проведено порівняльний аналіз, результати якого видно в табл. 4.

Таблиця 4. Порівняння методів прогнозування за критеріями ефективності

Критерії ефективності	Модифікована багатопшарова повнозв'язна нейронна мережа	Random Forest	XGBoost
Час навчання	2,3 с	0,275 с	0,187 с
Час сортування	0,328 с	0,304 с	0,291 с
Використання пам'яті	0,0081 Мб	0,0552 Мб	0,1355 Мб

Одним із ключових критеріїв оцінки ефективності методів прогнозування є час навчання моделі. Цей показник визначає швидкість, з якою алгоритм здатний адаптуватися до наявних даних і формувати прогнозні моделі, що особливо важливо при роботі з великими обсягами інформації або в реальному часі. У табл. 4 видно, що найбільший час навчання демонструє модифікована багатопшарова повнозв'язна нейронна мережа, проте його величина залишається прийнятною та становить близько 2,3 секунди. Але цей час є вже меншим, ніж час у звичайної MLP. Методи Random Forest та XGBoost показали швидкий час навчання: 0,275 та 0,187 секунди.

Ще одним важливим показником ефективності методів прогнозування є час сортування отриманих результатів. Цей критерій характеризує швидкість впорядкування прогнозованих варіантів, що має значення при подальшій інтеграції з алгоритмом оптимізації і формуванні кінцевого персоналізованого раціону. У табл. 4 результати демонструють, що найбільший час сортування займає модифікована багатопшарова повнозв'язна нейронна мережа близько 0,328 секунди, але й інші методи мають схожий результат: 0,304 та 0,291 секунди. Тобто різниця є незначною.

Ще одним важливим критерієм ефективності методів прогнозування є обсяг використаної оперативної пам'яті. Цей показник відображає ресурсоемність алгоритму та має значення при виборі методу для інтеграції з алгоритмом оптимізації та обробки великих обсягів даних. Аналіз результатів у табл. 4 показує, що найбільш економною за споживанням пам'яті є модифікована багатопшарова повнозв'язна нейронна мережа. Метод Random Forest займає проміжне положення, тоді як XGBoost демонструє найбільше споживання пам'яті серед трьох алгоритмів, що свідчить про його високу ресурсозатратність при обробці даних.

Отже, модифікована MLP має кращі показники ніж звичайна MLP і також інші методи прогнозування.

## Висновки

У ході дослідження було розроблено та проаналізовано алгоритми оптимізації і методи прогнозування для персоналізованого підбору раціону харчування, що забезпечують урахування індивідуальних харчових уподобань користувача.

Наукова новизна роботи полягає у створенні модифікованого генетичного алгоритму для оптимізації, що забезпечує покращення результатів порівняно зі стан-

дартними підходами: модифікований ГА забезпечив вищу продуктивність і різноманітність рішень при збереженні прийнятної точності, тоді як модифікована MLP показала кращу адаптивність до специфіки даних і більш ефективне використання ресурсів.

Проведене порівняння алгоритмів засвідчило, що саме генетичний алгоритм, особливо в модифікованій формі, є найбільш ефективним: він генерує найбільшу кількість варіантів раціонів, має найвищу продуктивність пошуку оптимальних рішень і забезпечує збалансоване відхилення від цільової калорійності. Інші методи продемонстрували свої переваги у вузьких аспектах (наприклад, лінійний метод у точності калорійності), але поступаються ГА за комплексною ефективністю.

Щодо методів прогнозування, проведене порівняння показало, що модифікована MLP є найбільш придатною для задачі прогнозування у цьому дослідженні. Вона забезпечує низький час навчання, порівняний час сортування та мінімальне використання оперативної пам'яті, що критично важливо для роботи з великими обсягами даних.

Таким чином, оптимальна комбінація для персоналізованого підбору раціону харчування включає модифікований ГА як метод оптимізації та модифіковану MLP як метод прогнозування. Це дозволяє досягти високої точності, продуктивності та ефективного використання ресурсів.

З практичної точки зору, розроблену систему можна інтегрувати в мобільні застосунки чи веб-сервіси для автоматизованого підбору раціону харчування, що відкриває перспективи її застосування у сфері дієтології, спортивної медицини та систем підтримки здорового способу життя. Подальші дослідження можуть бути спрямовані на розширення набору ознак (наприклад, урахування алергій чи рівня фізичної активності) та адаптацію алгоритмів для роботи у реальному часі.

1. World Health Organization. Obesity and overweight. URL: <https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight>
2. Garnet Health/ Basal Metabolic Rate Calculator/ URL: <https://www.garnethealth.org/news/basal-metabolic-rate-calculator>
3. Miow Y.X., Mok W.K.H., Gan W.Y., Lim P.Y., Appannah G., & Adznam S.N.A. The use of linear programming approach in diet optimization among children under five: a scoping review. *BMC Public Health*. 2025. Vol. 25, art. No. 1279.
4. van Dooren C. A review of the use of linear programming to optimize diets, nutritiously, economically and environmentally. *Frontiers in Nutrition*. 2018. Vol. 5, art. No. 48.
5. Shah A., & Khan M. A survey of Monte Carlo methods for noisy and costly densities with application to reinforcement learning and ABC. *EURASIP Journal on Advances in Signal Processing*. 2020. Vol. 2020, No. 1. P. 1–22.
6. Wang L., & Li Y. A review of Monte Carlo and quasi-Monte Carlo sampling techniques. *Wiley Interdisciplinary Reviews: Computational Statistics*. 2021. Vol. 13, No. 5. P. 1–20.
7. Zhang H., & Liu J. A review of Monte Carlo methods and their application in medical physics, "University of New Hampshire Honors Theses". 2022. P. 1–15.
8. Chatzipantsiou I., Tsirigotakis N., & Demetriou S. An extensive search algorithm to find feasible healthy menus for school children. *Applied Soft Computing*. 2022. Vol. 114.
9. Khandoker D., Akter M., & Rahman R. Diet planning with machine learning: Teacher-forced REINFORCE for composition compliance with nutrition enhancement. *Journal of Computational Nutrition*. 2021. Vol. 8.
10. Heinonen P., & Juuso E.K. Development of a genetic algorithms optimization algorithm for a nutritional guidance application. Proceedings of the 9th EUROSIM & the 57th SIMS. 2017. P. 55–60.

11. Karim R., Biplob M.B., & Arefin M.S. Developing a genetic algorithm based daily caloric recommendation system for humans. *International Journal of Computer Science and Information Technology*. 2024. Vol. 16, No. 3. P. 45–52.
12. Ooka T., Johno H., Nakamoto K., Yoda Y., & Yokomichi H. Random forest approach for determining risk prediction and predictive factors of type 2 diabetes. *BMJ Nutrition, Prevention & Health*. 2021. Vol. 0. P. e000200.
13. Anam S., Purwanto I.N., Mahanani D.M., Yusuf F.I., & Rasikhun H. Health risk classification using XGBoost with Bayesian hyperparameter optimization. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*. 2025. Vol. 9, No. 3. P. 465–473.
14. Xu Z., Guo X., Zhu A., He X., Zhao X., Han Y., & Subedi R. Using deep convolutional neural networks for image-based diagnosis of nutrient deficiencies in rice. *Computational Intelligence and Neuroscience*. 2020. Vol. 2020. P. 7307252.
15. Oliinyk A., & Subbotin S. A stochastic approach for extracting association rules *Pattern Recognition and Image Analysis*. 2016. Vol. 26, No. 2. P. 419–426.
16. Glasserman P. *Monte Carlo Methods in Financial Engineering*, Springer, New York, 2023, 585 p.
17. Subbotin S.O., Oliinyk A.O., Fedorchenko Ye.M., et al. Mathematical and software tools for decision making, pattern recognition and intelligent diagnostics. Zaporizhzhia Polytechnic National University, 2020. 271 p.
18. Fedorchenko I., Oliynik A., Zaiko T., Medvedev K., Fedorchenko Y., & Khokhlov M. Development of a modified genetic method for automatic university scheduling. *CEUR Workshop Proceedings*. 2024. Vol. 3662. P. 1–15.
19. Nlunge A. A deep architecture: Multi-layer perceptron. URL: <https://medium.com/@nlunge786/a-deep-architecture-multi-layer-perceptron-164bc5ff3842>
20. Liu Z., & Liu Y. A study of random forest algorithm for data mining applications. *Proceedings of the International Conference on Applied Computing and Information Technology (ACIT)*. 2016. P. 257–261.
21. Chen T., & Guestrin C. XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016. P. 785–794.

Надійшла до редакції 10.11.2025