

УДК 004.9:004.5:004.89

DOI <https://doi.org/10.32782/2415-8151.2025.35.33>

ГЛИБОКЕ ПІДКРІПЛЕНЕ НАВЧАННЯ ДЛЯ АДАПТИВНОГО НАЛАШТУВАННЯ ВЕБІНТЕРФЕЙСУ КОРИСТУВАЧА В РЕАЛЬНОМУ ЧАСІ

Лукашук Микола Миколайович

аспірант кафедри дизайну і технологій,

Київський національний університет культури і мистецтв, Київ, Україна,

e-mail: mykola.lukashuk@gmail.com, orcid: 0009-0009-5448-4315

Анотація. Мета. Метою роботи є розробка адаптивного дизайну вебінтерфейсів з використанням методів машинного навчання з підкріпленням.

Методологія. Методологія дослідження базувалася на застосуванні глибокого навчання з підкріпленням для адаптивного налаштування вебінтерфейсу в реальному часі. Взаємодія користувача з інтерфейсом моделювалась як процес ухвалення рішень. Застосовувалася технологія процесу ухвалення рішень Маркова (MDP), де політика визначає зміни елементів інтерфейсу на основі функції винагороди, що інтегрує метрики залучення користувача (швидкість про клікування, час бездіяльності) та відповідність стандартам доступності WCAG. Для визначення ефективних патернів взаємодії використовували оцінювачі максимальної правдоподібності та градієнтна оптимізація. Аналіз часових залежностей інтеракцій здійснювався за допомогою модулів LSTA та LSTM, які фокусуються на короткострокових і довгострокових перевагах користувачів, використовуючи згорткові шари та механізм уваги для екстракції релевантних функцій.

Результати. Щоб проаналізувати шаблони поведінки користувачів, включно зі шляхами навігації, методами введення даних і параметрами налаштування, структура використовує оцінки максимальної ймовірності та оптимізацію на основі градієнта. Цей процес моделює налаштування користувача та взаємодію для ефективної оптимізації налаштувань інтерфейсу користувача. Крім того, архітектура інтегрує модуль довготривалої короткочасної пам'яті (LSTM) для захоплення тимчасових залежностей у взаємодії користувача. Обробляючи дані через згорткові шари стробування, мережа дізнається як миттєві, так і розширені налаштування користувача. Тимчасова агрегація додатково вдосконалюється за допомогою адаптивних згорткових ядер і механізмів уваги, які визначають пріоритетність критичних областей інтерфейсу на основі взаємодії користувача.

Адаптивність системи покращується завдяки включенню нечіткої логіки в механізми стробування, де нечіткі правила моделюють складні залежності між факторами, що впливають на взаємодію користувача. Ця інтеграція полегшує ухвалення надійних рішень за різних умов, забезпечуючи стабільність і запобігаючи переобладнанню. Модуль LSTM використовує згорткові операції над об'єднаними комірками та прихованими станами для виокремлення життєво важливих функцій, вирішення проблеми зникнення градієнта та покращення збереження пам'яті. Дії, згенеровані моделлю, динамічно оновлюють елементи інтерфейсу користувача, сприяючи бездоганній та персоналізованій взаємодії з користувачем.

Реалізація бекенду, побудована на FastAPI, обробляє дані взаємодії користувача в режимі реального часу, обчислює винагороди та передбачає дії для

оптимізації налаштувань інтерфейсу користувача. Сервер використовує моделі на основі TensorFlow для аналізу користувацьких показників і даних про доступність, надаючи корисну інформацію для налаштування інтерфейсу. Архітектура забезпечує ефективний зв'язок між бекендом й інтерфейсом, даючи змогу модифікувати інтерфейс користувача в реальному часі.

Наукова новизна статті полягає в розробленні адаптивної моделі вебінтерфейсу, що використовує глибоке навчання з підкріпленням для персоналізації взаємодії з користувачем у реальному часі, враховуючи часові залежності та метрики доступності.

Практична значущість визначається можливістю застосування запропонованого підходу для підвищення залучення користувачів і забезпечення інклюзивності вебресурсів, що актуально для різних галузей, включно з електронною комерцією та освітніми платформами.

Ключові слова: адаптивний інтерфейс користувача, навчання з підкріпленням, марковський процес ухвалення рішень, вебдоступність, залучення користувачів, темпоральне моделювання.

ВСТУП

Розвиток галузі розробки програмного забезпечення продемонстрував появу значних інновацій з погляду інтерактивності з користувачем та загальної вебдоступності. Однак традиційний статичний дизайн інтерфейсів часто не може задовольнити різноманітних потреб користувачів, особливо тих, які мають певні обмеження з погляду життєдіяльності, або тих, які мають особливі вимоги, зумовлені характером їхньої зайнятості.

З огляду на це, головним напрямом поточного дослідження обрано адаптивізацію користувацьких інтерфейсів у середовищі вебплатформ із використанням основних принципів машинного навчання. Розроблена методологія має враховувати характер взаємодії користувача з вебплатформою та безпосередньо її інтерфейсом, реєструючи поведінку користувача, яка, наприклад, виражається в частоті про клікування, часі бездіяльності та відповідності критеріям вебдоступності, визначеним WCAG (Web Content Accessibility Guidelines – Настанови щодо доступності вебконтенту).

АНАЛІЗ ПОПЕРЕДНІХ ДОСЛІДЖЕНЬ

У науково-дослідницькому просторі сьогодення з'являються роботи, присвячені розробці дизайну адаптивних інтерфейсів у середовищах вебплатформ.

Робота З. Дена, Ю. Чена, К. Юя, Ц. Сюя, Х. Йє [4] містить результати дослідження методів оптимізації дизайну вебінтерфейсу на основі когнітивного навантаження користувача, використовуючи вебсайт Design Intelligence Center (DIC) як тематичне дослідження. Вивчається, як теорія когнітивного

навантаження в поєднанні з відстеженням очей та аналізом поведінки користувачів може надати розуміння для покращення дизайну інтерфейсу. Теорія когнітивного навантаження, яка наголошує на споживанні когнітивних ресурсів під час обробки інформації, є основою для оцінки та редизайну вебінтерфейсів, щоб збалансувати функціональність, естетику та задоволеність користувачів. Аналізуючи взаємодію користувачів, це дослідження визначає ключові фактори, що сприяють когнітивному навантаженню, і пропонує стратегії їх пом'якшення за допомогою дизайнерських втручань [4].

Експерименти з відстеження очей використані для спостереження та кількісної оцінки поведінки користувачів під час навігації різними ітераціями вебсайту DIC. Такі дані, як моделі погляду, час фіксації та карти траєкторій очей, показали, як користувачі взаємодіють із візуальними й текстовими елементами, підкреслюючи сфери ефективної передачі інформації. Порівняльний аналіз між двома версіями вебсайту показав, що коригування дизайну, включно з модифікацією макета та візуальним пріоритетом, суттєво вплинули на здатність користувачів ефективно обробляти інформацію. Дослідження показало, що зменшення візуального безладу та підкреслення важливого вмісту покращує концентрацію і розуміння користувача, покращуючи загальний досвід перегляду [4].

Паралельно з аналізом стеження за очима досліджувалася поведінка користувачів під час взаємодії на основі завдань на вебсайті DIC. Це передбачало запис й аналіз шляхів навігації, методів введення та часу виконання завдань. Результати показали, що

логічні неузгодженості та нечіткі маркування елементів інтерфейсу збільшували когнітивні зусилля та затримували виконання завдань. Наприклад, запровадження більш чітких інструкцій під час процесу реєстрації та оптимізація функцій установлення паролів зменшили непотрібні затримки користувача. Ці висновки скеровували розробку дієвих рекомендацій щодо вдосконалення зручності та доступності вебсайту [4].

С.-К. Хсу [7] дослідив чинники, що впливають на дизайн візуального вебінтерфейсу та його оцінку користувачами, зосереджуючись на функціональності й естетичній привабливості. Це дослідження використовує підхід, орієнтований на візуальну комунікацію, щоб дослідити, як стилі візуального інтерфейсу формують сприйняття та переваги користувачів. За допомогою поєднання оглядів літератури, інтерв'ю з користувачами та експериментальних рейтингів дослідження визначає ключові критерії дизайну та вивчає їх вплив на шість поширених стилів вебінтерфейсу.

Дослідження починається з узагальнення висновків з інтерв'ю з 31 користувачем і попередніх досліджень, щоб визначити 11 критичних критеріїв дизайну для вебінтерфейсів, включно з такими елементами, як структура, зручність використання, макет, кольори та читабельність. Вони є основою для оцінки 90 вебсторінок за допомогою рейтингової системи за шкалою Лайкерта за участю 32 суб'єктів. Аналіз даних, зібраних за допомогою аналізу основних компонентів (PCA – Principal Component Analysis), виявив два основні фактори – функціональність та емоційний складник, – які лежать в основі оцінок користувачів. Фактор функціональності містить такі практичні елементи, як простота відображення інформації та інтеграція гіперпосилань, тоді як емоційний фактор відображає естетичні та сенсорні атрибути, як-от візуальна привабливість і колірна гармонія.

Згодом дослідження вивчає, як ці фактори взаємодіють із шістьма окремими стилями вебінтерфейсу: спочатку текст, рамки та кольорові блоки, раціональний макет, емоційний та вигнутий, орієнтований на зображення та схожий на мультфільм. Результати показують, що стилі, які підкреслюють збалансовану функціональність та емоційну привабливість, як-от емоційний та викривлений тип, отримують вищі оцінки користувачів. І навпаки, стилі, які надто зосереджені на функціональності чи естетиці, наприклад, текст або мультфільм, мають тенденцію погано працювати в загальній оцінці. Це підкреслює важливість інтеграції зручності використання

з привабливими візуальними елементами для оптимізації взаємодії з користувачем [7].

Крім того, варто відзначити праці таких науковців: С. Аджаягбу, С. Іву [1], С. Аль-Федагі [2], В. Бруно, Г. Аль-Каймарі [3], Н. Діссанаяке, К. Хевагамаж [5], Д. Хушяр, С. Лі, Е. Ян, Дж. Чо, Х. Їм [6], Д. Ікбал, Т. Краший [8], Ш. ЛіЦюнь, В. Вей, Л. Яньчан [9], Л. Мадейські, М. Кавалерович [10], У.-Д. Рейпс [11], Ф. Рікка, П. Тонелла [13], Х. Шах [14], Дж. Шахід, М. Хамід, І. Джавед, К. Куреші, Н. Креспі [15], Ч. Тан, М. Тан, Ф. Ся, Ц. Чен, Х. Цзян, Ю. Чжан [16], А. Тангуду, А. Чхапола, Ш. Джайн [17], К. Вакіл, К. Ісмаїл [19], М. Ван, Ц. Жень, Ю. Цао, Ю. Лі, А. Цзян, Д. Ран, І. Ху, В. Ян, Т. Се [20], та інших.

Попри високу активність сучасних дослідників у розкритті різних аспектів досліджуваної наукової проблематики, можна констатувати, що питання, пов'язане з розробкою адаптивного дизайну вебінтерфейсів, досі залишається недостатньо дослідженим та потребує подальшого опрацювання.

МЕТА

Метою роботи є розробка адаптивного дизайну вебінтерфейсів з використанням методів машинного навчання з підкріпленням. Для досягнення поставленої мети проаналізовано сучасні підходи до створення адаптивних вебінтерфейсів та визначено їх обмеження. Розроблено математичну модель, що враховує особливості поведінки користувачів, часові залежності та доступність інтерфейсу. Запропоновано алгоритм навчання з підкріпленням для персоналізації взаємодії з користувачем у реальному часі. Проведено експериментальну перевірку моделі на тестових даних, що підтвердило її ефективність у підвищенні адаптивності та зручності використання вебінтерфейсів.

РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ

Ядро запропонованого оптимізаційного підходу базується на застосуванні принципів глибокого навчання з підкріпленням до системи адаптації інтерфейсу користувача (user interface – UI). Завдяки налаштуванню процесу кастомізації інтерфейсу в ролі процесу ухвалення рішень Маркова (Markov Decision Process – MDP), алгоритм вивчає оптимальні політики, які диктують характер того, як елементи інтерфейсу мають змінюватися у відповідь на дії користувача. Формально цей процес моделюється так:

$$\pi^* = \arg \max_{\pi} E \left[\sum_{t=1}^T \gamma^{t-1} R(s_t, a_t) \right] \quad (1)$$

π репрезентує політику, яка вирішує, яким буде характер вибору дій a_t у стані s_t , $R(s_t, a_t)$ позначає функцію винагороди, що виражає задоволення користувача та відповідність критеріями вебдоступності, γ – це дисконтний коефіцієнт, який урівноважує негайні винагороди з тими, що виникнуть у майбутньому. Функція винагороди інтегрує метрики залучення користувача, як-от швидкість про клікування, час бездіяльності та частота взаємодій з показниками відповідності критеріями вебдоступності, включно з колірними контрастними коефіцієнтами, присутністю ролей ARIA (Accessible Rich Internet Applications – доступні насичені інтернет-застосунки) та відповідністю стандартам WCAG. У фрагменті коду в табл. 1 представлено те, як функція `compute_reward` агрегує зазначені метрики задля отримання композитного сигналу винагороди.

Далі, з метою аналізу патернів користувацької поведінки, як-от навігаційні шляхи, методи введення даних та преференції кастомізації, запропонований фреймворк містить оцінювачі максимальної правдоподібності та використовує концепцію оптимізації на основі градієнта. Функція правдоподібності, налаштована для моделювання вподобань користувача та характеру його взаємодій з інтерфейсом, спрощує оцінку найбільш ефективних налаштувань інтерфейсу:

$$\theta^t = \arg \max_{\theta} L_t(\{A_s, Y_s\}_{s=1}^{t-1}, \theta) \quad (2)$$

Диференціація цієї правдоподібності забезпечує наявність градієнта, необхідного для оновлення параметрів політики:

$$\nabla_{\theta} L_t(\{A_s, Y_s\}_{s=1}^{t-1}, \theta) = \sum_{s=1}^{t-1} (Y_s - P_{\theta}(Y_{s=1})) (A_{1s} - A_{2s}) - \lambda \theta \quad (3)$$

де Y_s відповідає зворотному зв'язку від користувача, $P_{\theta}(Y_{s=1})$ становить передбачувану ймовірність позитивної взаємодії та λ позначає регуляризаційний параметр, який забезпечує стабільність моделі та запобігає переобладнанню.

Далі впроваджується модуль LSTA, який відповідає за фіксацію довгострокових часових залежностей в інтеракціях користувача. У цьому модулі застосовуються воротні згорткові шари, функцією яких є обробка вхідних даних, що дає змогу мережі навчатися як на миттєвих, так і на розширених преференціях користувача. Часова агрегація досягається шляхом інтеграції адаптивних згорткових ядер, які динамічно підлаштовуються, базуючись на темпоральному контексті.

Модуль обробляє функції введення $X \in \mathbb{R}^{C \times T \times H \times W}$ так:

$$G = X \oplus \text{Conv3D}_{3 \times 3 \times 3}(\text{Conv3D}_{3 \times 3 \times 3}(X) \odot \text{Conv3D}_{1 \times 1 \times 1}(X)) \quad (4)$$

де $\oplus$ та $\odot$ позначають поелементні додавання та мультиплікацію, відповідно. Вихід G потім піддається операції розділення на короткострокові та довгострокові гілки, кожна з яких впливає на різні аспекти темпоральної екстракції функцій. Удосконалення репрезентації вилучених функцій досягається за посередництвом впровадження міжрозмірних взаємодій між просторовими та каналними розмірностями через спеціалізований механізм уваги.

Ця операція здійснюється шляхом перестановки розмірностей G з упрощенням глобального об'єднання та подальшим виконанням згорткових перетворень з метою генерації карт уваги. Ці карти уваги зосереджують фокус на критичні ділянку інтерфейсу, що є найбільш афінними до взаємодій користувача з ним. Ця вдосконалена екстракція функцій формулюється так:

Таблиця 1

Фрагмент коду – Агрегація метрик залучення користувача

```
import numpy as np
def compute_reward(user_metrics, accessibility_metrics, gamma=0.9):
    engagement_reward = (
        0.4 * user_metrics['click_through_rate'] +
        0.3 * user_metrics['inactivity_time'] +
        0.3 * user_metrics['interaction_frequency']
    )
    accessibility_reward = (
        0.5 * accessibility_metrics['color_contrast'] +
        0.3 * accessibility_metrics['aria_roles'] +
        0.2 * accessibility_metrics['wcag_compliance']
    )
    total_reward = gamma * engagement_reward + (1 - gamma) * accessibility_reward
    return total_reward
```

Джерело: розроблено автором

$$Y = \frac{1}{3}(G_C \odot w_C + G_H \odot w_H + G_W \odot w_W) \quad (5)$$

де, w_C , w_H , w_W позначають карти уваги, які відповідають різними парам розмірності. Водночас темпоральна динаміка взаємодій користувача з інтерфейсом моделюється з використанням мережі довгої короткострокової пам'яті (Long Short-Term Memory – LSTM), яка працює шляхом утримання стану комірки c_t та збереженого стану h_t , що керується так:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (6)$$

$$h_t = o_t \odot \tanh(c_t) \quad (7)$$

f_t , i_t та o_t репрезентують ворота забування, входу та виходу відповідно, контролюючи потік інформації в межах мережі. Стан комірки-кандидата $\tilde{c}_t$ враховує потенціальну нову інформацію, яка має бути інтегрована в стан комірки. Фрагмент коду в табл. 2 демонструє цей процес.

Таблиця 2

Фрагмент коду – Фіксація темпоральних взаємозалежностей

```
import tensorflow as tf
from tensorflow.keras import layers, models
class UserBehaviorModel(tf.keras.Model):
    def __init__(self, input_size, hidden_size, output_size):
        super(UserBehaviorModel, self).__init__()
        self.lstm = layers.LSTM(hidden_size, return_sequences=False)
        self.dense = layers.Dense(output_size, activation='sigmoid')
    def call(self, x):
        x = self.lstm(x)
        return self.dense(x)
input_size = 10
hidden_size = 50
output_size = 1
model = UserBehaviorModel(input_size, hidden_size, output_size)
model.compile(optimizer='adam', loss='binary_crossentropy')
```

Джерело: розроблено автором

Адаптивність модуля може бути підвищена завдяки інтеграції концепції нечіткої логіки у воротні механізми шляхом конструкції нечітких правил згідно з методом Ванг – Менделя. Ці правила фіксують взаємозалежності між різними факторами, які впливають на взаємодію користувачів та програмного забезпечення:

$$f_i(f) = \sigma(W_f^x x_i + W_f^h h_i - 1 + W_f^f r_i + b_f) \quad (8)$$

де r_i репрезентує вихід, отриманий з моделі нечіткого передбачення, W_f^x , W_f^h та W_f^f позначають вагові матриці, а σ відповідає сигмоїдній активаційній функції, яка забезпечує фіксацію ворітних значень у визначеному діапазоні. На фрагменті коду табл. 3 показано узагальнювальну репрезентацію впровадження нечітких правил.

Запобігання проблемі градієнтного зникнення та вдосконалення збереження

Таблиця 3

Фрагмент коду – Впровадження нечітких правил

```
class FuzzyLSTMCell(layers.Layer):
    def __init__(self, units, **kwargs):
        super(FuzzyLSTMCell, self).__init__(**kwargs)
        self.units = units
        self.lstm_cell = layers.LSTMCell(units)
    def call(self, inputs, states):
        h_prev, c_prev = states
        r = self.fuzzy_rules(inputs, h_prev)
        inputs = tf.multiply(inputs, r)
        output, new_states = self.lstm_cell(inputs, states)
        return output, new_states
    def fuzzy_rules(self, inputs, h_prev):
        scale = tf.sigmoid(tf.matmul(inputs, self.kernel) + tf.matmul(h_prev, self.recurrent_kernel) + self.bias)
        return scale
```

Джерело: розроблено автором

пам'яті досягається шляхом згорткової операції, здійсненої на конкатенованій комірці та захованих станах мережі:

$$s_t = \tanh(\text{Conv1D}[c_t + h_t]) \quad (9)$$

Однорозмірна згортка вилучає важливі функції з комбінованої інформації про стан, даючи змогу моделі ідентифікувати критичні патерни. Гіперболічна тангенсна функція забезпечує те, що вихід s_t залишатиметься в межах обмеженого діапазону. Потім генерується дія y_t , що здійснює оновлення елементів інтерфейсу, яка формулюється як:

$$y_t = W_y k h_t + b_y \quad (10)$$

$W_y k$ становить вагову матрицю, пов'язану з різними сегментами дій, b_y представляє термін упередження.

Насамкінець запропонована архітектура працює на базі FastAPI бекенду, який обробляє вхідні дані про інтеракції користувача з елементами інтерфейсу. На фрагменті коду в табл. 4 продемонстровано алгоритм серверної обробки попередньо визначених функцій винагороди та моделі фіксації поведінки користувача з метою оптимальних налаштувань UI.

Таблиця 4

Фрагмент коду – FastAPI бекенд-обробка в середовищі TensorFlow

```
@app.post("/api/user_interaction")
async def user_interaction(request: Request):
    data = await request.json()
    user_metrics = extract_user_metrics(data)
    accessibility_metrics = extract_accessibility_metrics()
    reward = compute_reward(user_metrics, accessibility_metrics)
    state = preprocess_state(data)
    action_prob = model(state)
    action = tf.where(action_prob > 0.5, 1.0, 0.0).numpy().flatten()[0]
    ui_changes = determine_ui_changes(action)
    return JSONResponse(content={'ui_changes': ui_changes})

def extract_user_metrics(data):
    return {
        'click_through_rate': data.get('clicks', 0) / max(data.get('pages_viewed', 1), 1),
        'inactivity_time': data.get('inactivity_time', 0),
        'interaction_frequency': data.get('interactions', 0)
    }

def preprocess_state(data):
    state = np.array([
        data.get('clicks', 0),
        data.get('pages_viewed', 0),
        data.get('inactivity_time', 0),
        data.get('interactions', 0),
    ])
    state = state.reshape(1, -1)
    return state

def determine_ui_changes(action):
    if action > 0.5:
        return {'adjustDifficulty': True, 'newDifficulty': '2'}
    else:
        return {'adjustDifficulty': False}
```

Джерело: розроблено автором

FastAPI обробляє POST запити, які надходять з фронтенд-складника, обробляючи користувацькі взаємодії, а JSONResponse надсилає зміни UI у бекенді назад у фронтенд.

ВИСНОВКИ

Загалом, упровадження концепції навчання з підкріпленням є потенціальним проливом у сфері розробки адаптивних вебінтерфейсів, оскільки вона вможливорює навчання системи, базуючись на спостереженнях щодо поведінки користувача та його дій у відповідь на інтерфейс, що в подальшому забезпечує

вдосконалення. З іншого боку, інтеграція моделювання «стан – дія» та системи винагород пропонує вдосконалення системи адаптації інтерфейсу, гарантуючи задоволення потреб користувачів з різними потребами та вподобаннями.

ЛІТЕРАТУРА

[1] Ajaegbu C. C., Iwu C. F. Dawning of progressive web applications (PWA): Edging out the pitfalls of traditional mobile development. *International Journal of Computer Applications*. 2020. № 68 (1). P. 1–7. DOI: 10.5120/ijca2020920634.

[2] Al-Fedaghi S. Developing web applications. *International Journal of Software Engineering and Its Applications*. 2011. № 5 (2). P. 1–14. DOI: 10.14257/ijseia.2011.5.2.01.

[3] Bruno V., Al-Qaimari G. Characteristics of web applications that affect usability: A review. *Journal of Web Engineering*. 2021. № 20 (3). P. 1–18. DOI: 10.13052/jwe1540-9589.2031.

[4] Deng Z., Chen Y., Yu Q., Xu Z., Ye X. An experimental study on web interface design optimization based on user cognitive load: With the Design Intelligence Center (DIC) website as an example. *Design Studies and Intelligence Engineering*. 2023. Vol. 365. P. 506–518. DOI: 10.3233/FAIA220744.

[5] Dissanayake N., Hewagamage K. Web-based applications: Extending the general perspective of the service of web. *International Journal of Advanced Computer Science and Applications*. 2020. №8 (10). P. 1–7. DOI: 10.14569/IJACSA.2017.081001.

[6] Hooshyar D., Lee S., Yang Y., Jo J., Lim H. Long-term effects of adaptive customization support on elderly people. *Cognition, Technology & Work*. 2019. № 21. DOI: 10.1007/s10111-018-0516-9.

[7] Hsu C. C. Factors affecting webpage's visual interface design and style. *Procedia Computer Science*. 2011. №3. P. 1315–1320. DOI: 10.1016/j.procs.2011.01.009.

[8] Iqbal J., Best T. Enhanced Augmented Reality with Real-Time Semantic Segmentation and Adaptive Deep Learning Models. *ResearchGate* : website. DOI: 10.13140/RG.2.2.10817.67682.

[9] LiQun S., Wei W., Liu Y. Customizable WEB UI of based on templates. *Information Technology Journal*. 2010. № 9. DOI: 10.3923/itj.2010.1677.1681.

[10] Madeyski L., Kawalerowicz M. Architectural design of modern web applications. *International Journal of Software Engineering and Knowledge Engineering*. 2020. № 30 (5). P. 1–20. DOI: 10.1142/S0218194020500180.

[11] Reips U. D. Developing web applications for psychological research. *Behavior Research Methods*. 2022. № 54 (2). P. 1–10. DOI: 10.3758/s13428-021-01600-4.

[12] Reips U., Social D. networking services in scientific research: Opportunities and challenges. *International Journal of Internet Science*. 2023. №18 (1). P. 1–9. DOI: 10.5565/rev/ijis.001.

[13] Ricca F., Tonella F. Analysis and testing of web applications. *Proceedings – International Conference on Software Engineering*. 2001. P. 25–34. DOI: 10.1109/ICSE.2001.919078..

[14] Shah H. Navigating UI Engineering Architectures: A Deep Dive into Modern Web Application Design. *ResearchGate* : website. DOI: 10.13140/RG.2.2.30307.77603.

[15] Shahid J., Hameed M., Javed I., Qureshi K., Crespi N. A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions. *Applied Sciences*. 2022. № 12. 4077. DOI: 10.3390/app12084077.

[16] Tang Z., Tan M., Xia F., Cheng Q., Jiang H., Zhang Y. AutoGameUI: Constructing High-Fidelity Game UIs via Multimodal Learning and Interactive Web-Based Tool. *ArXiv* : website. DOI: 10.48550/arXiv.2411.03709.

[17] Tangudu A., Chhapola A., Jain S. Leveraging Lightning Web Components for Modern Salesforce UI Development. *Innovative Research Thoughts*. 2023. № 9. P. 220–234. DOI: 10.36676/irt.v9.i2.1459.

[18] Wakil K. F., Ismail K. M. Extracting the features of modern web applications based on web engineering methods. *International Journal of Advanced Computer Science and Applications*. 2020. № 10 (3). P. 1–10. DOI: 10.14569/IJACSA.2019.0100310.

[19] Wakil K. F., Ismail K. M. Intelligent web applications as future generation of web applications. *Scientific Journal of Informatics*. 2020. № 6 (2). P. 213–222. DOI: 10.15294/sji.v6i2.23214.

[20] Wu M., Wang H., Ren J., Cao Y., Li Y., Jiang A., Ran D., Hu Y., Yang W., Xie T. Skill-Adaptive Imitation Learning for UI Test Reuse. *ArXiv* : website. DOI: 10.48550/arXiv.2409.13311.

REFERENCES

[1] Ajaegbu, C. C., & Iwu, C. F. (2020). Dawning of progressive web applications (PWA): Edging out the pitfalls of traditional mobile development. *International Journal of Computer Applications*, 68 (1), 1–7. <https://doi.org/10.5120/ijca2020920634> [in English].

[2] Al-Fedaghi, S. (2011). Developing web applications. *International Journal of Software Engineering and Its Applications*, 5 (2), 1–14. <https://doi.org/10.14257/ijseia.2011.5.2.01> [in English].

[3] Bruno, V., & Al-Qaimari, G. (2021). Characteristics of web applications that affect usability: A review. *Journal of Web Engineering*, 20 (3), 1–18. <https://doi.org/10.13052/jwe1540-9589.2031> [in English].

[4] Deng, Z., Chen, Y., Yu, Q., Xu, Z., & Ye, X. (2023). An experimental study on web interface design optimization based on user cognitive load: With the Design Intelligence Center (DIC) website as an example. *Design Studies and Intelligence Engineering*, 365, 506–518. <https://doi.org/10.3233/FAIA220744> [in English].

[5] Dissanayake, N., & Hewagamage, K. (2020). Web-based applications: Extending the general perspective of the service of web. *International Journal of Advanced Computer Science and Applications*, 8 (10), 1–7. <https://doi.org/10.14569/IJACSA.2017.081001> [in English].

[6] Hooshyar, D., Lee, S., Yang, Y., Jo, J., & Lim, H. (2019). Long-term effects of adaptive customization support on elderly people. *Cognition, Technology & Work*, 21. <https://doi.org/10.1007/s10111-018-0516-9> [in English].

[7] Hsu, C. C. (2011). Factors affecting webpage's visual interface design and style. *Procedia Computer Science*, 3, 1315–1320. <https://doi.org/10.1016/j.procs.2011.01.009> [in English].

- [8] Iqbal, J., & Best, T. (2024). Enhanced augmented reality with real-time semantic segmentation and adaptive deep learning models. ResearchGate. <https://doi.org/10.13140/RG.2.2.10817.67682> [in English].
- [9] LiQun, S., Wei, W., & Liu, Y. (2010). Customizable WEB UI based on templates. *Information Technology Journal*, 9, 1677–1681. <https://doi.org/10.3923/itj.2010.1677.1681> [in English].
- [10] Madeyski, L., & Kawalerowicz, M. (2020). Architectural design of modern web applications. *International Journal of Software Engineering and Knowledge Engineering*, 30 (5), 1–20. <https://doi.org/10.1142/S0218194020500180> [in English].
- [11] Reips, U. (2023). Social networking services in scientific research: Opportunities and challenges. *International Journal of Internet Science*, 18 (1), 1–9. <https://doi.org/10.5565/rev/ijis.001> [in English].
- [12] Reips, U. D. (2022). Developing web applications for psychological research. *Behavior Research Methods*, 54 (2), 1–10. <https://doi.org/10.3758/s13428-021-01600-4> [in English].
- [13] Ricca, F., & Tonella, P. (2001). Analysis and testing of web applications. *Proceedings of the International Conference on Software Engineering*, 25–34. <https://doi.org/10.1109/ICSE.2001.919078> [in English].
- [14] Shah, H. (2024). Navigating UI engineering architectures: A deep dive into modern web application design. ResearchGate. <https://doi.org/10.13140/RG.2.2.30307.77603> [in English].
- [15] Shahid, J., Hameed, M., Javed, I., Qureshi, K., & Crespi, N. (2022). A comparative study of web application security parameters: Current trends and future directions. *Applied Sciences*, 12, 4077. <https://doi.org/10.3390/app12084077> [in English].
- [16] Tang, Z., Tan, M., Xia, F., Cheng, Q., Jiang, H., & Zhang, Y. (2024). AutoGameUI: Constructing high-fidelity game UIs via multimodal learning and interactive web-based tool. ArXiv. <https://doi.org/10.48550/arXiv.2411.03709> [in English].
- [17] Tangudu, A., Chhapola, A., & Jain S. (2023) Leveraging Lightning Web Components for Modern Salesforce UI Development. *Innovative Research Thoughts*, 9, 220–234. DOI: <https://doi.org/10.36676/irt.v9.i2.1459> [in English].
- [18] Wakil, K. F., & Ismail, K. M. (2020). Extracting the features of modern web applications based on web engineering methods. *International Journal of Advanced Computer Science and Applications*, 10 (3), 1–10. <https://doi.org/10.14569/IJACSA.2019.0100310> [in English].
- [19] Wakil, K. F., & Ismail, K. M. (2020). Intelligent web applications as future generation of web applications. *Scientific Journal of Informatics*, 6 (2), 213–222. <https://doi.org/10.15294/sji.v6i2.23214> [in English].
- [20] Wu, M., Wang, H., Ren, J., Cao, Y., Li, Y., Jiang, A., Ran, D., Hu, Y., Yang, W., & Xie, T. (2024). Skill-adaptive imitation learning for UI test reuse. ArXiv. <https://doi.org/10.48550/arXiv.2409.13311> [in English].

ABSTRACT

Lukashuk M. Deep reinforcement learning for real-time adaptive user web-interface customization

Purpose. The purpose of the work is to develop adaptive web interface design using machine learning with reinforcement methods.

Methodology. The research methodology was based on the application of deep reinforcement learning for adaptive customization of the web interface in real time. User interaction with the interface was modeled as a decision-making process. The Markov Decision Process (MDP) technology was used, where the policy determines changes to interface elements based on a reward function that integrates user engagement metrics (click rate, idle time) and compliance with WCAG accessibility standards. Maximum likelihood estimators and gradient optimization were used to determine effective interaction patterns. The analysis of time dependencies of interactions was carried out using the LSTA and LSTM modules, which focus on short-term and long-term user preferences, using convolutional layers and an attention mechanism to extract relevant features.

Results. To analyze user behavior patterns, including navigation paths, input methods, and customization options, the framework uses maximum likelihood estimation and gradient-based optimization. This process models user preferences and interactions to efficiently optimize user interface settings. In addition, the architecture integrates a long-short-term memory (LSTM) module to capture temporal dependencies in user interactions. By processing data through

convolutional gating layers, the network learns both instantaneous and extended user preferences. Temporal aggregation is further enhanced by adaptive convolutional kernels and attention mechanisms that prioritize critical areas of the interface based on user interaction. The adaptability of the system is improved by incorporating fuzzy logic into the gating mechanisms, where fuzzy rules model the complex dependencies between factors that affect user interaction. This integration facilitates robust decision-making under different conditions, ensuring stability and preventing overfitting. The LSTM module uses convolutional operations on merged cells and hidden states to extract vital features, solve the problem of gradient vanishing, and improve memory retention. Actions generated by the model dynamically update user interface elements, contributing to a seamless and personalized user experience.

The backend implementation, built on FastAPI, processes user interaction data in real time, calculates rewards, and predicts actions to optimize user interface settings. The server uses TensorFlow-based models to analyze user metrics and accessibility data, providing useful information for interface customization. The architecture provides efficient communication between the backend and the frontend, allowing for real-time user interface modification.

The scientific novelty of the article lies in the development of an adaptive web interface model that uses deep reinforcement learning to personalize user interactions in real time, taking into account temporal dependencies and accessibility metrics.

The practical relevance is determined by the possibility of applying the proposed approach to increase user engagement and ensure inclusiveness of web resources, which is relevant for various industries, including e-commerce and educational platforms.

Keywords: adaptive user interface, reinforcement learning, Markov decision process, web accessibility, user engagement, temporal modeling.

AUTHOR'S NOTE:

Lukashuk Mykola, Postgraduate Student at the Department of Design and Technology, Kyiv National University of Culture and Arts, Kyiv, Ukraine, e-mail: mykola.lukashuk@gmail.com, orcid: 0009-0009-5448-4315.

Стаття подана до редакції 02.01.2025 р.