Міністерство освіти і науки України
НТУ «Дніпровська політехніка»
Ройтлінгенський університет техніки та економіки (Німеччина)
Еслінгенський університет прикладних наук (Німеччина)
Технічний університет Фрайберзька гірнича академія (Німеччина)
Університет Кобленц-Ландау (Німеччина)
Краківська гірничо-металургійна академія (Польща)
Вроцлавський технічний університет (Польща)
Дніпропетровський національний університет імені Олеся Гончара
ДКХ «Дніпровський машинобудівний завод»

ПРОБЛЕМИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ОСВІТІ, НАУЦІ ТА ПРОМИСЛОВОСТІ

XIX МІЖНАРОДНА КОНФЕРЕНЦІЯ

27 листопада 2024 року
м. Дніпро

Збірник наукових праць
№ 9

Дніпро
НТУ «ДП»
2024

УДК 622 (06)
П78

П78 **Проблеми** використання інформаційних технологій в освіті, науці та промисловості: XVIII міжнар. конф. (27 листопада 2024 р., м. Дніпро): зб. наук. пр. / ред. кол.: А.А. Азюковський та ін.; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро: НТУ «ДП», 2024. – № 9. – 236 с.

ISBN 978-966-350-760-6

Подано результати теоретичних та експериментальних досліджень з різних аспектів використання інформаційних технологій в освіті, науці та керування промисловістю. У публікаціях розглянуто питання створення та вдосконалення програмних засобів обробки та передачі інформації, математичного моделювання, дистанційної освіти, інформаційної безпеки та телекомунікації.

Для наукових, інженерно-технічних співробітників і студентів, які спеціалізуються в галузі обчислювальної техніки та інформаційних технологій.

УДК 622 (06)

ISBN 978-966-350-760-6

РОЗДІЛ 1

МІЖНАРОДНЕ СПІВРОБІТНИЦТВО У СФЕРІ ОСВІТИ, НАУКИ І ВИРОБНИЦТВА

УДК 004.415.3:681.6

Г.Г. Швачич[1], П.О. Іщук[1], П.О. Щербина[1], О.А. Дмитрієва[2]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна
[2]Institute for modelling hydraulic and environmental systems, University of Stuttgart

МОДЕЛІ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ НА ЧИСЕЛЬНО-АНАЛІТИЧНИХ СХЕМАХ ПІДВИЩЕНОГО ПОРЯДКУ ТОЧНОСТІ

Анотація. Робота присвячена побудові чисельно-аналітичним методом конструювання ефективних алгоритмів для розв’язування задач параболічного типу. Використовуючи апріорну інформацію про гладкість розв’язку, основна увага приділяється побудові рішень високого порядку точності.

***Ключові слова**: паралельні обчислення, моделі, високий порядок точності, розпаралелювання обчислень, диференціальні рівняння, математична фізика.*

Вступ. Паралельні обчислювальні системи розвиваються дуже швидко. З появою обчислювальних кластерів паралельні обчислення сталі доступними багатьом. Для їх побудови, як правило, використовуються масові процесори, стандартні мережеві технології і вільно поширюване програмне забезпечення. Помітимо, під обчислювальним кластером розуміють сукупність комп'ютерів, об'єднаних у рамках деякої мережі для вирішення одного завдання. Невеликий кластер з 6-у обчислювальних лез може ефективно використовуватися невеликими підрозділами. У зв'язку з цим нині проблема розробки ефективного програмного забезпечення виявилася однією з центральних проблем паралельних обчислень в цілому [1, 2]. Створення паралельних обчислювальних систем зажадало розробки математичних концепцій побудови паралельних алгоритмів, тобто алгоритмів, пристосованих до реалізації на подібних обчислювальних системах.

Основний зміст роботи. Отже, стає очевидним, що розробка паралельних обчислювальних систем вимагала створення математичної концепції для побудови паралельних алгоритмів, тобто алгоритмів, адаптованих для реалізації на таких системах. Основою побудови паралельного алгоритму може служити як послідовний алгоритм, так і задача сама по собі [3, 4]. При розпаралелюванні послідовного алгоритму найбільш розумним видається прагматичний підхід, а

саме в послідовних алгоритмах виявляються елементи, що часто зустрічаються, які потім трансформуються в паралельну форму.

Для побудови чисельно-аналітичної схеми на підставі апріорної інформації шукана функція представляється у вигляді ряду Тейлора:

$$Y_{p+\varepsilon_x,1}(t,x)=\sum_{n=0}^{\infty}\varepsilon_x^n\cdot Y_{p,n+1}(t), \tag{1}$$

де

$$\begin{cases}\varepsilon_x=\dfrac{x-x_p}{x_{p+1}-x_p}\in[+1,-1],\\ Y_{p,n+1}=\dfrac{Dx1^n}{n!}\cdot\dfrac{\partial Y}{\partial x^n}\Big|_{x=x_p}.\end{cases}$$

Після узгодження (1) з досліджуваним рівнянням і прирівнюючи коефіцієнти при однакових степенях ε_x^n, отримаємо систему звичайних диференціальних рівнянь (СЗДР)

$$Y'_{p,n+1}(t)=\frac{(n+1)(n+2)}{Dx1^2}\cdot Y_{p,n+3}(t) \tag{2}$$

що мають форму Коші

$$Y_{p,n+1}(0)=\varphi_{p,n+1}, \tag{3}$$

де $\varphi_{p,n+1}$ – відомі значення тейлоровских компонент початкової функції. Обмежимося в правій частині ряду Тейлора (1) кінцевим числом доданків $n=N$. Тоді отримаємо:

$$Y_{p+\varepsilon_x,1}(x,t)=\sum_{n=0}^{N}\varepsilon_x^n\cdot Y_{p,n+1}(t), \tag{4}$$

де N – ціле число. Щоб апроксимувати досліджуване рівняння в точці (x_p,t) введемо в розгляд замикаючі зв'язки

$$\begin{Bmatrix}Y_{p,N+1}\\ Y_{p,N}\end{Bmatrix}. \tag{5}$$

Поклавши в (4) $\varepsilon_x = \pm 1$, отримують на триточковому шаблоні систему з двох рівнянь алгебри

$$\begin{cases} Y_{p,N+1} + Y_{p,N} = \left[Y_{p+1,1} - \sum_{n=0}^{N-2} Y_{p,n+1} \right], \\ Y_{p,N-1} - Y_{p,N-1} = (-1)^N \cdot \left[Y_{p-1,1} - \sum_{n=0}^{N-2} (-1)^n \cdot Y_{p,n+1} \right]. \end{cases} \tag{6}$$

Знайдемо

$$\begin{Bmatrix} Y_{p,N+1} \\ Y_{p,N} \end{Bmatrix} = \frac{1}{2} \cdot \left\{ \left[Y_{p+1,1} \pm (-1)^N \cdot Y_{p-1,1} \right] - \sum \varphi_n^{\pm} \cdot Y_{p,n+1} \right\}, \tag{7}$$

де

$$\varphi_n^{\pm} = 1 + (-1)^{n+N}, \quad N = 2, 3, 4, . \tag{8}$$

нормуючі множники.

Порядок точності обчислень визначається утриманим значенням *N*. Так, при *N = 2* значення $n = \overline{0,0}$ і тоді отримують

$$\begin{cases} Y_{p,2} = \frac{1}{2} \cdot \left[Y_{p+1,1} - Y_{p-1,1} \right], \\ Y_{p,3} = \frac{1}{2} \cdot \left\{ \left[Y_{p+1,1} + Y_{p-1,1} \right] - 2 \cdot Y_{p,1} \right\}. \end{cases} \tag{9}$$

Після підстановки (9) в (2) отримаємо СЗДР виду:

$$Y'_{p,1}(t) = \frac{1}{Dx1^2} \cdot \left\{ \left[Y_{p+1,1}(t) + Y_{p-1,1}(t) \right] - 2Y_{p,1}(t) \right\}, \quad p = \overline{1,2m-1}, \tag{10}$$

де $\{Y_{0,1}(t), Y_{2m,1}(t)\}$ граничні функції першого роду.

Помітимо, що розвинений підхід включає звичайні скінечнно-різницеві методи як окремий випадок. Так, схема (10) співпадає з класичною задачею Дириxле. Для задачі (10) характерно і те, що передача інформації на границях області в природну схему розрахунку внутрішньої точки реалізується точно без пониження порядку апроксимації.

Розроблена процедура чисельно-аналітичної дискретизації досить просто узагальнюється і на інші типи диференціальних рівнянь математичної фізики.

Зокрема, в стаціонарних задачах простіше локалізувати певні особливості і в областях гладкості використовувати схеми високого порядку точності.

Задавшись метою синтезувати паралельні алгоритми цього методу із співвідношень (10) витікає, що він вписується в концепцію необмеженого паралелізму [5]. Дійсно, якщо можна призначити один процесор на один вузол розрахункової області, то стає можливим проведення розрахунків в усіх вузлах паралельно.

Висновки. Зауважимо, що чисельні алгоритми мають дві характеристики, за якими часто судять про їх якість. Перша – складність (час, що витрачається на обчислення), друга, – чисельна стійкість (мала чутливість до помилок округлення). На зорі паралельних обчислень основне значення мала складність (тобто міра паралелізму) алгоритму. Ця концепція дістало назву концепції необмеженого паралелізму [5]. Вважалося, що процесорів може бути скільки завгодно багато, усі вони універсальні, працюють в синхронному режимі, мають загальну пам'ять, будь-які передачі інформації здійснюються миттєво і без конфліктів. Критикуючи таку концепцію, зауважимо, що побудова швидких паралельних алгоритмів завжди буде актуальним питанням.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ivaschenko V.P., Shvachych G.G., Semenov S.G. Efficient parallelization algorithms of the applied tasks in multiprocessor computing systems. *Системні технології.* Дніпро. 2017. № 2 (109). P. 57 – 66.
2. I. Mamuzić, G.G. Shvachych, P. O. Ishchuk Challenges of Parallel Numerical Integration of Pollution Transport Equations. *Book of abstracts of the 17th International Symposium of Croatian Metallurgical Society – SHNM '2004 Materials and metallurgy (published in Metalurgija,* 63 (2024) 3-4), Croatian metallurgical society. P. 477-492.
3. Ivashchenko V., Shvachych G., Udovyk I., Moroz D., Mamuzić I. On the problem of the efficiency of computations of multiprocessor systems when solving applied problems. *Metallurgy and related tropics* : *Proceedings of 15th international symposium of Croatian metallurgical society*. Zagreb, 22–23 March 2022. Zagreb, Croatia.
4. Shvachych G., Vozna N., Ivashchenko O., Bilyi O., Moroz D. Efficient algorithms for parallelizing tridiagonal systems of equations. *System technologies*. Dnipro. 2021. № 5 (136). P. 110 – 119.

УДК 519.6:534.2:004.94

Л.С. Коряшкіна[1], С.Д. Приходченко[1], Є.О. Загинайло[1], О.І. Сироткина[2]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна
[2]School of Computer Science University of Windsor, Windsor, Ontario, Kanada

ВИКОРИСТАННЯ ФРЕЙМВОРКУ PYROOMACOUSTICS ДЛЯ ПРОГРАМНОГО МОДЕЛЮВАННЯ ЗВУКОВИХ ЯВИЩ В ЗАМКНЕНОМУ ПРОСТОРІ

Анотація. У даній роботі представлено комплексне акустичне моделювання середовища за допомогою фреймворку Pyroomacoustics, що дозволяє точно імітувати взаємодію звуку з поверхнями та оптимізувати локалізацію джерела звуку без необхідності створення фізичних прототипів. Результати підтверджують ефективність методики для проведення наукових досліджень та розробки нових технологій у галузі аудіо-інженерії та моделювання акустичних середовищ.

***Ключові слова**: акустичне моделювання, Pyroomacoustics, звукові хвилі, мікрофонний масив, локалізація джерела звуку, комп'ютерна симуляція.*

Вступ. За останні 10 років програмне моделювання акустичних явищ значно розвинулось завдяки прогресу в обчислювальних технологіях, новим алгоритмам та інтеграції машинного навчання. Збільшення продуктивності обчислень [1], особливо завдяки графічним процесорам та паралельним системам, дозволило ефективніше розв'язувати складні задачі акустики, що раніше були недосяжними. Фізичне моделювання звуку досягло нового рівня реалістичності завдяки точнішим методам, таким як числовий аналіз скінченних елементів та граничних елементів. Важливим напрямком стало створення інтерактивних систем для громадянської інженерії, аудіопроцесингу та віртуальної реальності [2], де акустика в реальному часі стала критично важливою для іммерсивних досвідів. Машинне навчання і AI активно застосовуються для прогнозування результатів складних розрахунків, класифікації звуків та оптимізації середовищ. Нові інструменти та програмне забезпечення полегшили роботу для фахівців, а відкрите ПЗ розширює можливості адаптації під конкретні задачі. Продовжується акцент на мультидисциплінарність, що інтегрує акустику з іншими науками, створюючи нові горизонти для досліджень та застосувань.

Проблема локалізації об'єкту за даними акустичного контролю є складною задачею, яка вимагає інтеграції різних наукових підходів та технологій. У центрі цієї проблеми знаходиться аналіз акустичних сигналів для визначення положення об'єкту в просторі. Основна складність полягає у тому, що звукові хвилі можуть бути значною мірою зрушені або змінені середовищем, включаючи поширення через різні матеріали, розсіяння на поверхнях та абсорбцію енергії. Для точного визначення позиції необхідні розроблені алгоритми, які враховують ці фактори.

Один з ключових підходів – використання масивів мікрофонів [3] для отримання багатоканальних даних, які потім аналізуються для визначення часу прибуття сигналу (Time of Arrival, TOA) чи різниці часу прибуття (Time Difference of Arrival, TDOA). Однак, реальні умови часто сприяють появі шумів і мультипатів, що знижує точність таких методів. Щоб протистояти цьому, активно використовуються статистичні та фільтрувальні техніки, такі як Калманівський фільтр або баєсові моделі.

Машинне навчання та штучний інтелект [4] також займають важливе місце в цьому контексті, оскільки вони дозволяють автоматично вчитуватися на великих наборах даних, що включають різні умови середовища та типи шумів. Це дає можливість розробляти адаптивні системи, які можуть автоматично коригувати свої параметри для досягнення кращої точності. Проте, тренування таких моделей вимагає значних обчислювальних ресурсів і великої кількості анотованих даних, що є додатковим викликом.

Для підвищення точності локалізації важливо враховувати геометрію середовища, особливо коли вона складна або динамічна. Методи комп'ютерного зору і сенсорних мереж можуть допомогти вирішити цю проблему, поєднуючи акустичні дані з візуальними, що дозволяє отримати більш повну картину середовища. Такі мульти-сенсорні підходи стають все більш популярними завдяки своїй ефективності у складних умовах.

Таким чином, локалізація об'єкту за даними акустичного контролю потребує комплексного підходу, який об'єднує передові алгоритми, машинне навчання, обробку сигналів та мульти-сенсорні системи. Незважаючи на прогрес у цій області, залишаються виклики, зокрема пов'язані з адаптацією до динамічних середовищ і забезпеченням достатньої точності в присутності шумів і мультипатів.

Моделювання акустичного середовища програмними засобами критично для розробки ефективних систем, що враховують складність звукових хвиль у різних умовах. Це дозволяє передбачувати поведінку звуку, оптимізувати архітектуру приміщень та вдосконалювати технології обробки сигналів. Програмні моделі дають можливість аналізувати взаємодію звуку з матеріалами, шумом і різноманітними поверхнями, що важливо для досягнення реалістичності в аудіоприкладних системах, таких як віртуальна реальність чи інтерактивні середовища. Також це сприяє підвищенню точності локалізації джерел звуку та розвитку інтелектуальних систем моніторингу.

Постановка задачі. Створення моделей просторів за допомогою фреймворку Pyroomacoustics для програмного моделювання звукових явищ в замкнених просторах є важливим завдяки його можливостям точної інтерпретації складних акустичних процесів. Цей інструмент забезпечує науковцям та розробникам зручний спосіб створювати реалістичні моделі акустики, враховуючи складність взаємодії звуку з поверхнями та матеріалами. Pyroomacoustics дозволяє швидко протестувати різні сценарії, оптимізувати параметри середовища та аналізувати результати звукової пропагації. Крім того,

його можливості інтеграції з іншими інструментами та підтримка машинного навчання роблять його незамінним для розробки передових аудіо-систем, таких як віртуальна реальність або інтерактивні середовища. Це значно сприяє підвищенню точності та продуктивності роботи з акустичними даними..

Основний зміст роботи. Фреймворк Pyroomacoustics [1, 2] пропонує значні переваги для програмного моделювання звукових явищ завдяки своїй гнучкості, ефективності та простоті використання. Воно забезпечує науковцям і розробникам можливість швидко створювати реалістичні моделі акустики, враховуючи складність взаємодії звуку з поверхнями та матеріалами. Його вбудовані алгоритми дозволяють точно аналізувати різні аспекти звукової пропагації, що критично важливо для оптимізації акустики приміщень. Крім того, Pyroomacoustics підтримує інтеграцію з машинним навчанням, що сприяє автоматизації процесів аналізу та передбачення поведінки звуку. Це також полегшує тестування різних сценаріїв без необхідності великих обчислювальних ресурсів, що значно скорочує час на розробку та адаптацію систем. Всього це робить Pyroomacoustics незамінним інструментом для сучасних аудіо-технологій.

Розберемо приклад роботи із цим фреймворком.

Фрагмент програми використовує бібліотеку Pyroomacoustics для створення та візуалізації моделі акустичного простору. Спочатку визначаються координати кутів двомірної кімнати у формі масиву, який потім передається функції pra.Room.from_corners для побудови моделі кімнати. Далі за допомогою методу plot реалізовується графічне зображення цього простору на площині з встановленням меж для осей X та Y. Для переходу до тримірної моделі використовується метод extrude, який "піднімає" двомірну фігуру на задану висоту, тим самим формує тримірний простір. Після цього повторно виконується візуалізація з додатковим розташуванням осі Z, що дозволяє отримати повну просторову картину акустичної кімні. Це дозволяє аналізувати або моделювати звукові явища у складних геометричних середовищах (рис.1).

Далі моделювання акустичного середовища має отримати джерело звукового сигналу та мікрофонний масив, або ж декілька. Спочатку завантажується аудіофайл за допомогою функції wavfile.read, що надає частоту дискретизації fs та сам звуковий сигнал. Далі береться модель кімнати за допомогою методу pra.Room.from_corners, де враховуються параметри, такі як частота дискретизації, трасування променів та поглинання звуку повітрям. До цього простору додається джерело звуку з координатами [1., 1.] і завантаженим сигналом. Потім визначається кругова конфігурація масиву мікрофонів за допомогою функції pra.circular_2D_array з центром у точці [2., 2.], кількістю елементів M=6 та радіусом 0.1. Цей масив мікрофонів додається до моделі кімнати для запису звукових даних. Виконуються графічні візуалізації кімнати після додавання мікрофонів, що дозволяє аналізувати розташування джерел і приймачів звуку в просторі. Це забезпечує можливість детального дослідження акустичної взаємодії у складному середовищі

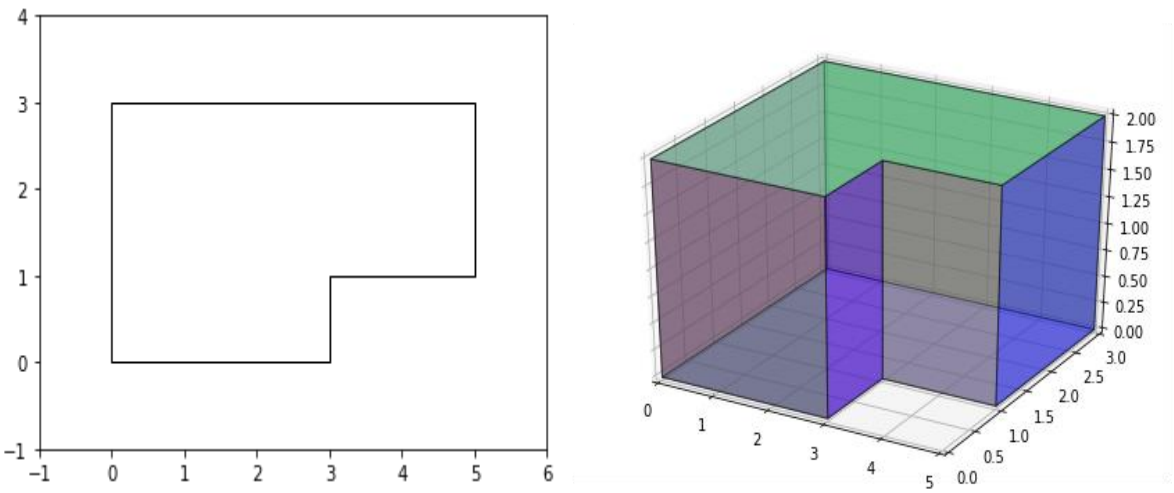


Рис. 1. План кімнати та її тривимірне представлення

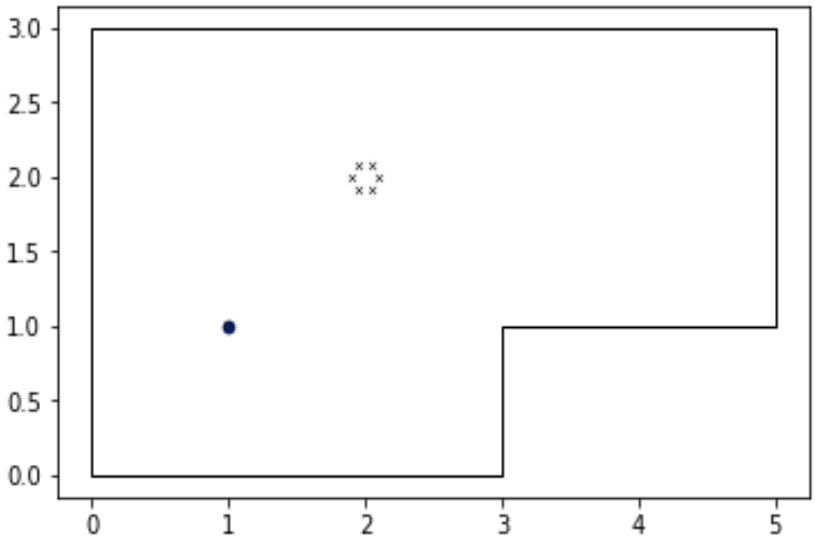


Рис. 2. План кімнати із нанесеними джерелом звуку (1,1), та масивом мікрофонів (навкруги 2,2)

З метою перевірки правильності роботи розробленого моделі з використанням Pyroomacoustics було створено модель великої залу з розмірами 500 на 650 метрів у плані та висотою 3 метри. Цей простір імітує реальні умови для проведення акустичних експериментів. Для отримання точних даних про розповсюдження звуку був розташований масив мікрофонів, кожен з яких розташований на відстані 100 метрів один від одного. Таке розташування дозволяє аналізувати взаємодію звукових хвиль у великому обсязі простору, враховуючи різні матеріали поверхонь та можливі шуми. Модель забезпечує можливість тестування різних сценаріїв звукової пропагації та впевнитися у точності результатів, що критично важливо для досягнення реалістичності при моделюванні складних акустичних середовищ.

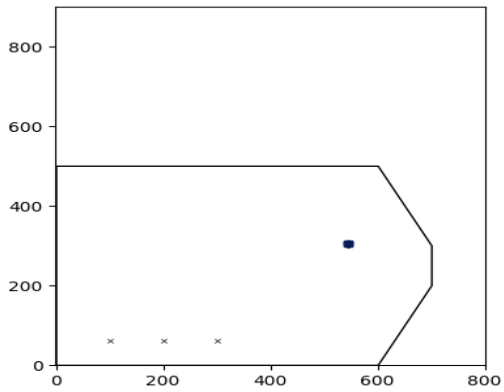


Рис. 3. План модельної зали із нанесеними джерелом звуку (точка), та масивом мікрофонів (хрестики)

Моделювання прийому звуку на мікрофони, розташовані на значній відстані один від одного, також продемонструвало прогнозовані результати. Зокрема, було зафіксовано різницю у часі прибуття звукових хвиль до кожної точці прийому, що залежить від відстані між ними (рис. 4).

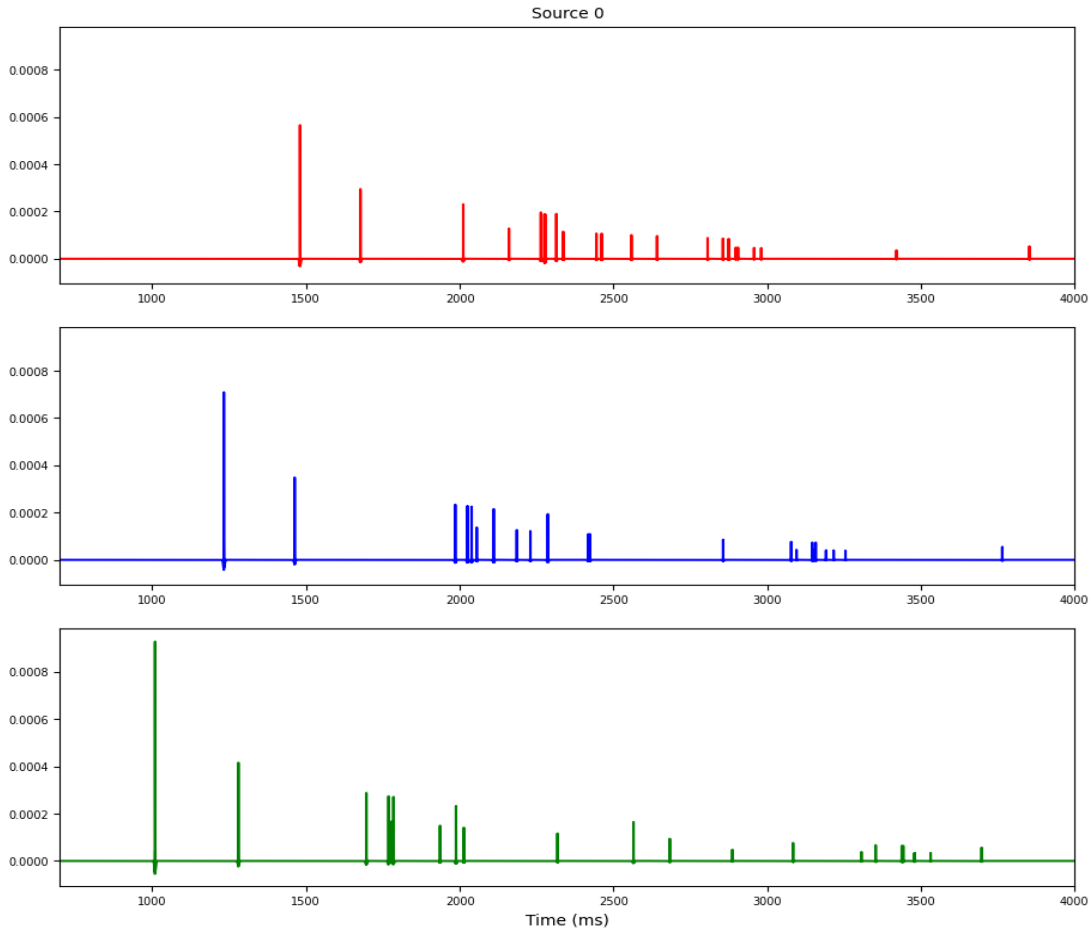


Рис. 4. Прийом звуку мікрофонами. (Три мікрофони і одне джерело звуку. Перший мікрофон – найвіддалений)

Це спостереження підтверджує можливість проведення модельних експериментів без необхідності створення фізичних макетів, що значно спрощує процес досліджень. Такий підхід дозволяє ефективніше оптимізувати локалізацію джерела звуку в просторі, оскільки можна швидко та точно аналізувати різні конфігурації і параметри середовища. Використання комп'ютерного моделювання для таких задач забезпечує гнучкість та швидкість отримання результатів, що є критично важливим для розробки нових аудіотехнологій та систем моніторингу. Більш того, це дозволяє проводити експерименти в складних або неможливих для реального життя умовах, що посилює можливості наукових досліджень.

Висновки. В результаті було спроектовано комплексну модель акустичного середовища з використанням Pyroomacoustics, що дозволяє точно імітувати різні умови. У тому числі, було обрано оптимальні параметри для моделювання прийому звуку на масив мікрофонів, а також змодельовано взаємодію звукових хвиль з поверхнями різних матеріалів. Це надало можливість проводити детальні експерименти без створення фізичних прототипів, що значно спростило процес аналізу та оптимізації систем локалізації джерела звуку. Більше того, дане підходи забезпечили гнучкість та швидкість отримання результатів, що критично важливо для подальших наукових досліджень та розробки нових технологій.

ПЕРЕЛІК ПОСИЛАНЬ

1. Scheibler, Robin & Bezzam, Eric & Dokmanic, Ivan. (2017). Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms. 10.1109/ICASSP.2018.8461310.
2. SALMON, François & VERRON, Charles & CAMIER, Cédric & MALGRANGE, Marina. (2024). Adaptation and evaluation of Pyroomacoustics for auralization purposes. INTER-NOISE and NOISE-CON Congress and Conference Proceedings. 270. 5906-5917. 10.3397/IN_2024_3660.
3. C. Othmani et al., "A review of the state-of-the-art approaches in detecting time-of-flight in roomimpluse responses," Sensors and Actuators: A. Physical, vol. 374, doi : 10.1016/j.sna.2024.115467.
4. Sanguano, Daniel & Lucio Naranjo, Jose & Tenenbaum, Roberto & Sampaio-Regattiere, G.B.. (2023). Real-time impulse response: a methodology based on Machine Learning approaches for a rapid impulse response generation for real-time Acoustic Virtual Reality systems. Intelligent Systems with Applications. 21. 200306. 10.1016/j.iswa.2023.200306.

УДК 004.932:528.854

Л.І. Мещеряков[1], І.М. Удовик[1],М.В. Пімахов[1], Я. Сконечний[2]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна
[2]Вроцлавский технологический университет, Польша

ВІЗУАЛІЗАЦІЯ ВІРТУАЛЬНИХ СЕРЕДОВИЩ

Анотація. В роботі представлено процес практичної реалізації створення додатку віртуального середовища архітектурної візуалізації аудиторії університету за допомогою системи програмування Blueprint.

Ключові слова: *віртуальний реалізм, інтерактивне, ArchViz, Unreal Engine, Blueprint, Niagara, ігровий рушій, архітектурна візуалізація.*

Вступ. Величезна обчислювальна потужність сучасних персональних комп’ютерів знайшла застосування в різних інформаційних сферах, серед яких є і множина комп'ютерних ігор. Процедура створення відеоігор вимагає використання ігрового рушія і не зовсім відомо, що ігрові рушії можуть слугувати більш ширшим цілям, ніж просто розробка ігор. Так їх можна використовувати і для формування віртуальних середовищ, для створення інформативних демонстрацій та презентацій продуктів або фізичних просторів, які не лише захоплюють, але й занурюють кінцевого користувача, дозволяючи йому, ґрунтуючись на проектах архітектурної візуалізації (*ArchViz*) з високим ступенем занурення, відчути простір так, ніби він присутній там фізично.

Використовуючи можливості ігрових рушіїв, компанії та розробники можуть отримати безліч можливостей для презентації своїх продуктів та візуалізації архітектурних споруд. Незалежно від того, чи це демонстрація проекту нерухомості, нової лінійки продуктів, чи захоплююча віртуальна прогулянка майбутнім закладом, потенціал використання ігрових рушіїв для демонстрацій є досить величезним. Завдяки своїй здатності створювати візуально приголомшливе інтерактивне середовище, ці додатки пропонують безпрецедентний рівень залучення потенційних інвесторів та клієнтів. Використовуючи можливості ігрових рушіїв та найсучасніші технології, компанії можуть створювати візуально вражаючі та захоплюючі проекти *ArchViz*, які занурюють користувачів у відповідне віртуальне середовище. Оскільки застосування інформаційних технологій продовжує стрімкий розвиток, можливості використання ігрових рушіїв для демонстрації різнотипових проектів будуть звичайно тільки швидко розширюватися,

Аналіз останніх публікацій. Ігрові рушії слугують основою розробки ігор, пропонуючи широкий спектр можливостей та функцій, які спрощують процес та дають можливість розробникам втілювати свої творчі задуми в життя. Їх можна визначити як програмний фреймворк або платформу, яка надає розробникам набір інструментів, бібліотек та систем для створення, розробки і

розгортання відеоігор, що слугує проміжною ланкою між кодом гри та апаратним забезпеченням, дозволяючи розробникам зосередитися на логіки та дизайні гри. Функціонально ігрові рушії можливо розділити на: рушій рендерингу, фізичні рушії, звукові рушії, включаючи штучний інтелект, сценарії та мови програмування, конвеєри ресурсів та редактори гри.

На даний час із популярних ігрових рушіїв можна виділити перед усе такі як *Unity, Unreal Engine* та *CryEngine. Unity* широко використовуваний ігровий рушій, відомий своєю універсальністю та простотою використання. *Unreal Engine* відомий своїми дуже широкими візуальними можливостями та високою точністю комп'ютерної графіки. Він надає такі розширені можливості як трасування променів світла у реальному часі та динамічне глобальне освітлення, що дозволяє розробникам створювати візуально вражаючі ефекти занурення. *CryEngine* в основному зосереджений на створенні найсучаснішої комп'ютерної графіки та реалістичних середовищ. Він відмінно справляється з рендерингом великих відкритих просторів, динамічних погодних систем та деталізованих моделей персонажів. Серед розглянутих, найбільш широко використовуваних двигунів, для розробки *ArchViz*-проекту найкращим виступає *Unreal Engine 5,* так як він є оптимальним за багатьма параметрами з усіх запропонованих, містить найновітніші технології та має можливість програмувати за допомогою системи *Blueprint*, яка є візуальною адаптацією мови *C++*.

Метою статті є представлення процесу практичної реалізації створення додатку віртуального середовища архітектурної візуалізації однієї з аудиторій університету за допомогою системи програмування Blueprint.

Виклад основного матеріалу. Вирішальне значення для ефективної розробки майбутнього проекту являється вивчення вимог до неї. Щоб розробити програмну систему, ці вимоги повинні бути визначені, виміряні, протестовані та пояснені. Специфікації функціональних вимог проекту, що формується − це опис функцій та їхніх атрибутів, який не містить жодних винятків чи протиріч. Також додаток має бути фотореалістичним та деталізованим, а також мати змогу використовувати систему сучасного реалістичного освітлення.

Опис використаних в проекті елементів та плагінів. Застосована в проекті система *VFX*-частинок реалізована за допомогою системи *Niagara*, ефекти для камери сформовані за допомогою елементу *PostProcessVolume*, погода реалізована завдяки паку *Ultra Dynamic Sky*, реалістичні матеріали були використані з безкоштовної бібліотеки *Quixel Bridge Megascans*, також для матеріалу візуалізації скла було застосовано плагін *Advanced Glass Material Pack,* для імпорту 3D-моделей проекту представленої аудиторії був використаний плагін *Datasmith*, а для *UI*-частини застосовано елемент *Widget,* та в цілому все програмувалось на основі системи *Blueprint*.

Основа опису функціоналу *Niagara* – це система візуальних ефектів наступного покоління в *Unreal Engine 5*. За допомогою *Niagara* технічний художник має можливість створювати додаткову функціональність самостійно, без допомоги програміста. Система адаптивна та гнучка. Початківці можуть

почати з модифікації шаблонів або прикладів поведінки, а досвідчені користувачі можуть створювати свої власні модулі. У системі *Niagara* є чотири основні компоненти: *Системи, Емітери, Модулі* та *Параметри* [1]. Система *Niagara* – це контейнер для всього, що може знадобитись при побудові ефекту. Всередині цієї системи можуть бути розташовані різні блоки, які складаються в модуль для створення загального ефекту [2].

Опис функціоналу *Post Process Volume* – це спеціальний тип об'єму, який можна додати до рівня для доступу к функціям пост-обробки. Декілька об'ємів можуть бути розміщені для визначення вигляду певної області або встановлені для впливу на всю сцену [3]. Можна додати *Post Process Volume* у свій рівень за допомогою панелі *Place Actors*. Після розміщення на рівні використовується панель *Details* при доступі до всіх можливих властивостей та функцій [4]. Налаштування *Post Process Volume* є специфічними налаштуваннями для цього розміщеного об'єму та його взаємодії зі сценою та будь-якими іншими об'ємами *Post Process*, з якими вони можуть перекриватися. Наприклад, можна перемикати властивість *Infinite Extent*, щоб зробити цей об'єм *Post Process*, що впливає на все, що розміщено на сцені, або залишити його невстановленим, щоб вплинути лише на певну область сформованої сцени.

Функціонал пакету *Ultra-Dynamic Sky (UDS)* – це пак для *Unreal Engine 5*. *UDS* – це система неба, розроблена для більш динамічного та природного вигляду, ніж більшість попередніх рішень по візуалізації неба, пропонує велику гнучкість та можливості налаштування з інтерфейсом, що розроблений для підвищення простоти та швидкості обробки [5]. За допомогою *Ultra-Dynamic Sky* можна налаштувати час доби і всі аспекти неба будуть оновлюватися разом з ним. Система має повністю динамічні хмари, місяць та зорі. Вбудоване освітлення з сонцем, місяцем та освітленням синхронізується з небом. Можна налаштувати хмарність від ясного неба до похмурого. Також є повна система погоди – *Ultra-Dynamic Weather*, яка додає до сформованої сцени дощ, сніг, блискавку та інші погодні сутності [6].

Основа опису функціоналу *Quixel Bridge* – це плагін для *Unreal Engine 5,* який дозволяє отримати повний доступ до бібліотеки *Megascans* прямо у редакторі рівнів. Є можливість переглядати колекції, шукати конкретні активи та додавати активи до проектів *Unreal Engine* [7]. *Quixel Bridge* для *Unreal Engine* встановлюється та активується за замовчуванням. Можна відкрити його з головної панелі інструментів: для цього потрібно натиснути кнопку *Створити*, а потім відповідно вибрати *Quixel Bridge*. Після відкриття *Bridge* потрібно увійти у свій обліковий запис *Epic Games*.

Опис функціоналу системи програмування *Blueprints* – це система візуального скриптування в *Unreal Engine 5*, яка дозволяє створювати геймплей без написання коду. За допомогою *Blueprint* можливо створювати логіку гри, інтерактивні об'єкти, інтерфейс користувача та багато іншого [8]. *Blueprints* мають графічний інтерфейс, де можливо перетягувати та з'єднувати вузли для створення структури потрібної логіки. Вони мають ряд основних компонентів,

таких як функції, змінні та події. Можна створювати свої власні *Blueprint* класи або унаслідуватися від існуючих класів *Unreal Engine*. Саме тому вся програмна частина розробляється логічними блоками через систему *Blueprints*.

Опис створення сцени. Керування від першого обличчя. Для створення *ArchViz* додатку було синтезовано новий проект *Unreal Engine 5* із шаблоном *First Person*. Після чого відкриється початковий рівень та інші панелі. Далі необхідно відкрити *BP_FirstPersonCharacter*, що знаходиться за шляхом *All->Content->FirstPerson->Blueprints* та видалити в ньому елемент *FirstPersonMesh*, щоб в *ArchViz*-проекті не було недоречних «роботичних рук». Також необхідно встановити постійну швидкість руху – для цього в елементі *Character Movement* встановлюється значення *Max Walk Speed* на 300. Також для того, щоб користувач не міг випадково застрягти у геометрії рівня, потрібно ввести заборону і для цього потрібно вимкнути параметр *Can Jump*.

Імпорт моделей та створення матеріалів. Для створення віртуальної аудиторії потрібно відкрити новий рівень та імпортувати заздалегідь сформовану *3D*-модель аудиторії. Тут використовується плагін *Datasmith* для *Autodesk 3ds Max* та *Datasmith Importer* для *Unreal Engine 5* [9].

Так як *Datasmith* – не досить досконалий плагін, то може виникнути проблема зникнення текстур з усіх об’єктів. Вона вирішується імпортом реалістичних матеріалів з безкоштовної бібліотеки *Quixel Bridge*, вбудованої у двигун. З цієї бібліотеки було імпортовано двадцять один матеріал, серед яких: різновиди дерев’яних і мармурових підлог, дерево, пластик, тканини та метал. Також були сформовані власні матеріали завдяки створенню *Master Material’y* (рис. 1), що дозволяє синтезувати безліч матеріалів, беручи за референс один і той самий, таким чином позитивно впливаючи на оптимізацію всього проекту.

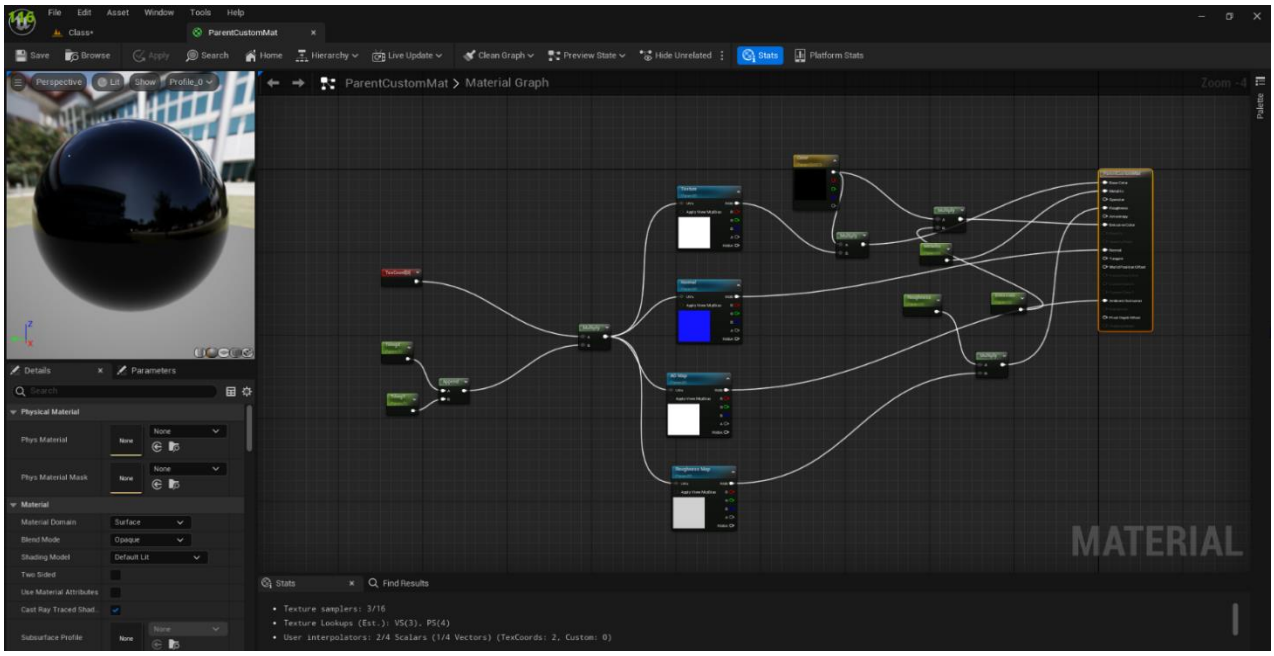


Рис. 1. Master Material

Master Material був створений з урахуванням усіх подальших потреб проекту, а саме: можливість зміни текстур, карт нормалей, карт *Ambient Occlusion*, карт шорсткості, можливість зменшувати та збільшувати текстури та карти, змінювати ступінь металевості матеріалу, ступінь його освітленості, ступінь його шорсткості та можливість зміни кольорової палітри текстури [10]. Після створення цього *Master Material'y* було синтезовано множину об'єктів типу *Material Instance*, кожен з яких має свої значення.

Також було сформовано два окремих матеріали, що не відносяться до *Master Material*. Серед них перший – матеріал для жалюзі, що має у деякій мірі трохи пропускати сонячне проміння, тобто застосована технологія поверхневого розсіювання *Subsurface Shading*, а *Master Material*, створений раніше, не має такої можливості, тому цей матеріал було створено окремо. І окремо було створено матеріал *LampMaton* та *LampMaton_Inst* (*Material Instance*) для ламп на стелі, а також створено *LampMatParameter* для функціонування цих ламп [11].

Оптимізація об'єктів та застосування матеріалів. Після завантаження та створення власних матеріалів, потрібно їх застосувати до *3D*-моделей так, щоб це не шкодило загальній оптимізації. Для цього достатньо об'єднати складні об'єкти в одну модель за допомогою опції *ConvertActors To Static Mesh* [12].

Після об'єднання, потрібно видалити схожі об'єкти та скопіювати один декілька разів у координати схожих об'єктів. Таким чином було об'єднано деталі однієї парти, деталі однієї навчальної дошки, видалено інші парти, та другу дошку, залишено лише один стілець, одну лампу на стелі, два види жалюзі (довгі і короткі) та повторно скопійовано всі ці об'єкти, після чого до кожної моделі на сцені було застосовано свій *Material Instance* (рис. 2).

Рис. 2. Створене віртуальне середовище із застосованими матеріалами

Задання глобального освітлення та спецефектів. Існує декілька шляхів задання глобального освітлення: власноруч або плагінами/паками (наборами

пакетів). В даному проекті використовується придбаний пак *Ultra-Dynamic Sky* задля економії часу та отримання фотореалістичного результату. Цей пак додається до проекту через лаунчер *Epic Games*. Таким чином, у вкладці *Details* для *Ultra_Dynamic_Sky* було задано стартовий час доби 09:36 ранку, ввімкнене параметр *Animate Time of Day*, параметру *Volumetric Cloud Rendering Mode* задано значення *Full Cinematic Quality*, ввімкнене параметри *Enable Fog Inside Clouds, Two Layers, Use Cloud Shadows, Simulate Real Sun, Simulate Real Moon* та *Simulate Real Stars*, задано дату 26.04.2024, задано координати широти і довготи згідно координат університету, узятих з *Google Maps*, а саме: для *Latitude* задано значення 48.455312, для *Longitude* задано значення 35.061639.

Рис. 3. Результат рендеру віртуального середовища в реальному часі

Також, для пост-процесінгу, до сцени було додано *PostProcessVolume* із набору стандартних інструментів, в якому були ввімкнені хроматична аберація, віньєтування, ввімкнене параметр *Infinite Extend (Unbound*) у розділі *Bloom* параметру *Method* задане значення *Convolution*, в розділі *Exposure* для *Metering Mode* задано значення *Manual*, ввімкнене параметр *Apply Physical Camera Exposure*, задані значення *Min Brightness*: 20, *Max Brightness*: 20, *Speed Up*: 3, *Speed Down*: 1, було ввімкнене *Dirt Mask* з інтенсивністю 2.048 та змінені інші основні параметри. На основі системи частинок *Niagara* для задання атмосфери сцени було створено систему літаючих пилинок. При налаштуванні системи було змінено такі параметри, як *Emitter State, Spawn Rate, Initialize Particle, Shape Location, Add Velocity, Particle State, Scale Sprite Size* (рис. 3), *Scale Color* (рис. 4), *Curl Noise Forse, Sprite Renderer* [12].

Таким чином, в результаті проведених проектних робіт із можливими спецефектами та освітленням, був досягнутий досить задовільний

фотореалістичний *3D*-рендер в реальному часі для створеного віртуального середовища однієї з аудиторій університету (рис. 3).

Висновки. Розроблений інтерактивний *ArchViz*-додаток, може використовуватись при проектуванні приміщень, так як має привабливу фотореалістичну комп'ютерну графіку та містить в собі інтерактивні можливості взаємодії з оточенням: можливості підібрати підлогу зі списку запропонованих дизайнерами, або змінити колір крісла на один із списку, або взагалі замінити деякі крісла на диван и побачити, як саме його потрібно буде збирати. Можливо змінювати рішення з приводу проектування приміщення, дивлячись на те, як воно виглядатиме вдень, вночі, під світлом ламп, при ясній погоді або грозовій. Даний приклад *Arch-Viz* додатку може вплинути на уявлення проектування приміщення в цілому та створити попит на білдингові компанії, що використовують подібні програмні рішення через отримані враження користувача на етапі, коли реальне приміщення ще не існує. Варто зазначити, що при розробці було використано лише візуальне програмування *Blueprint*, що прискорює розробку подібного додатку через можливість побачити логіку наочно, і це ніяк не впливає на оптимізацію проекту та є досить простим в розумінні задач програмної інженерії.

ПЕРЕЛІК ПОСИЛАНЬ

1. Overview of Niagara Effects for Unreal Engine | Unreal Engine 5.0 Documentation: https://docs.unrealengine.com/5.0/en-US/overview-of-niagara-effects-for-unreal-engine/
2. Creating Visual Effects in Niagara for Unreal Engine | Unreal Engine 5.2 Documentation: https://docs.unrealengine.com/5.2/en-US/creating-visual-effects-in-niagara-for-unreal-engine/
3. Quick Start for Niagara Effects in Unreal Engine | Unreal Engine 5.0 Documentation: https://docs.unrealengine.com/5.0/en-US/quick-start-for-niagara-effects-in-unreal-engine/
4. Post Process Effects in Unreal Engine | Unreal Engine 5.2 Documentation: https://docs.unrealengine.com/5.2/en-US/post-process-effects-in-unreal-engine/
5. Ultra Dynamic Sky Content Detail: https://www.unrealengine.com/marketplace/en-US/product/ultra-dynamic-sky
6. Ultra Dynamic Sky - Product Video + Quick Start (UE5 Version): https://youtu.be/b52npy-XUdQ
7. Getting started with Bridge in UE5: https://youtu.be/OYP0EcAEfsQ
8. Навчальний посібник з Unreal Engine 5 для початківців - початковий курс UE5: https://youtu.be/k-zMkzmduqI
9. Unreal Engine 5 Import A Scene By Datasmith From 3DS Max 2022.2: https://youtu.be/kSDDetL3bYg
10. Master Material Basics! Unreal 5 (Must KNOW workflow!): https://youtu.be/tT0ANO5sXso
11. Material Parameter Collection | 5-Minute Materials [UE5]: https://youtu.be/J2Qf5v9_uSY
12. How to convert placed actors/meshes into blueprints UE5: https://youtu.be/MSYki36PJh8

УДК 004.942

Г.Г. Швачич[1], П.О. Іщук[1], С.Г. Семенов[2]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна
[2]University Education Commission, Krakow, Poland

РОЗПОДІЛЕНЕ МОДЕЛЮВАННЯ ЗАДАЧ ІДЕНТИФІКАЦІЇ ДОВКІЛЛЯ

Анотація. На основі одновимірного рівняння перенесення шкідливих домішок розроблено алгоритм ідентифікації для екологічних задач. Ефективність алгоритму ілюструється порівняльного аналізу розв'язку конкретних задач, як для послідовних, так і для багатопроцесорних обчислювальних систем.

***Ключові слова**: шкідливі домішки, розподілене моделювання, алгоритм, багатопроцесорні системи, система рівнянь.*

Основу сучасних моделей динаміки атмосфери складають закони збереження маси, моменту і енергії, які разом із законами хімії і термодинаміки описують процеси, що відбуваються в атмосфері, океані, грунті, а також їхню взаємодію [1]. У математичному представленні це системи багатовимірних нелінійних диференціальних рівнянь, які вирішуються в припущенні, що зовнішнім джерелом є сонячна енергія [2, 3]. Ці системи включають ряд вхідних параметрів, відомих з експерименту при певній похибці.

Дослідження спрямовано на реалізацію алгоритму ідентифікації в задачах екології на основі застосування одновимірного рівняння перенесення шкідливих домішок. Алгоритм ілюструється на прикладах розв'язку тестових задач, як для послідовних, так і для багатопроцесорних обчислювальних систем.

У простому випадку при моделюванні перенесення шкідливих домішок на тлі атмосферних процесів, коли параметри довкілля вважаються відомими, математично це завдання сформулюється наступним чином. В області визначення шуканої функції $y \in (-\infty, +\infty)$ необхідно знайти розв'язок рівняння перенесення домішок:

$$u\varphi'(y) + \sigma\varphi(y) = \mu\varphi''(y) + Q\delta(y - y_0), \qquad (1)$$

за наступних умов обмеженості шуканого рішення в області визначення:

$$\left.\begin{aligned} \varphi(y)_{|y\to+\infty} = 0 \\ \varphi(y)_{|y\to-\infty} = 0 \end{aligned}\right\}. \qquad (2)$$

У даних дослідженнях побудова математичної моделі реалізовано методом прямих [4]. Область визначення шуканого рішення задачі (1),

визначається на прямій $y \in [y_0, y_1]$, покритій рівномірно вузлами $p = \overline{1,2m-1}$. Для кожного сіткового вузла розв'язок подається сукупністю двох рішень. Поєднання розв'язків рівняння по вузлам реалізується за допомогою граничних умов четвертого роду. У такій постановці остаточний вигляд математичної моделі подається системою лінійних алгебраїчних рівнянь трьохдіагональної структури:

$$C_p \varphi_{p+1} - \varphi_p + D_p \varphi_{p-1} = f_p, \tag{3}$$

де

$$\left.\begin{aligned} & C_p = \frac{e^{-\beta_1}}{Det}, \; D_p = \frac{e^{-\beta_2}}{Det}, \\ & f_p = -\frac{Q_p Dy1}{2\Omega\mu} \frac{(1-e^{-2\Omega_p})}{Det}, \\ & Det = (1+e^{-2\Omega_p}), \; Dy1 = \frac{Y_L - Y_0}{2m}, \end{aligned}\right\} \tag{4}$$

– відомі величини, а індекс *p* характеризує можливість обліку залежності вхідних параметрів *R*, включаючи і потужність Q_p точкових джерел забруднення, від просторової координати y.

Процес моделювання співвідношень (3), (4) реалізовано засобами паралельного програмування [5]. Відомо, що найбільший ефект від паралельного процесора досягається в тих випадках, коли він застосовується для виконання матричних обчислень лінійної алгебри [5, 6]. Для даного завдання розпаралелювання стає можливим тільки за умов певної рахунковості вузлів області визначення системи лінійних алгебраїчних рівнянь (3), коли параметр $m = 2^k, k \in Z$. Таким чином, процес розпаралелювання системи лінійних алгебраїчних рівнянь (3) полягає у відображенні шуканих змінних φ_p у вузли відповідного графа. Обчислювальний алгоритм, що досить просто реалізовує цю процедуру, конструюється елементарними діями з схеми «непарно-парної» редукції рядків системи лінійних алгебраїчних рівнянь (3) [5].

ПЕРЕЛІК ПОСИЛАНЬ

1. Гоков О. М. Процеси в атмосфері і атмосферно-іоносферна взаємодія [Електронний ресурс] : монографія / О. М. Гоков. - Х. : ХНЕУ ім. С. Кузнеця, 2017. - 135 с.
2. Ivaschenko V.P., Shvachych G.G., Semenov S.G. Efficient parallelization algorithms of the applied tasks in multiprocessor computing systems. *Системні технології*. Дніпро. 2017. № 2 (109). P. 57 – 66.
3. I. Mamuzić, G.G. Shvachych, P. O. Ishchuk Challenges of Parallel Numerical Integration of Pollution Transport Equations. *Book of abstracts of the 17th International Symposium of Croatian Metallurgical Society – SHNM '2004 Materials and metallurgy (published in Metalurgija,* 63 (2024) 3-4), Croatian metallurgical society. P. 477-492.

4. Ivashchenko V., Shvachych G., Udovyk I., Moroz D., Mamuzić I. On the problem of the efficiency of computations of multiprocessor systems when solving applied problems. *Metallurgy and related tropics* : *Proceedings of 15th international symposium of Croatian metallurgical society*. Zagreb, 22–23 March 2022. Zagreb, Croatia.

УДК 553.252

Л.І. Мещеряков[1], М.М. Одновол[1], Д.Д. Сень[1], О.А. Дмитрієва[2]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна
[2]Institute for modelling hydraulic and environmental systems, University of Stuttgart

ГЕНЕРАТИВНІ МЕТОДИ У 3D-МОДЕЛЮВАННІ

Анотація. Проведено всебічний аналіз основних методів генерації тривимірних моделей, включаючи генеративні змагальні мережі (GAN), варіаційні автоенкодери (VAE) та дифузійні моделі. Встановлено, що дифузійні моделі забезпечують високий рівень деталізації об'єктів, особливо у випадках із великою кількістю кроків генерації. Дослідження текстових промптів показало їх критичну роль у визначенні характеристик згенерованих об'єктів: чіткіші описи приводять до більш якісних результатів. Оцінено практичне застосування цих методів у різних галузях: архітектурній візуалізації, ігровому дизайні, технічному моделюванні.

Ключові слова: *генеративне 3D-моделювання, дифузійні моделі, генеративні алгоритми, текстові промпти, оптимізація параметрів генерації, комп'ютерна графіка, архітектурна візуалізація.*

Вступ. Розвиток технологій генеративного штучного інтелекту (ГШІ) відкриває нові горизонти у галузі комп'ютерної графіки, зокрема у 3D-моделюванні. Використання генеративних алгоритмів у поєднанні з сучасними обчислювальними системами дозволяє створювати складні тривимірні об'єкти з високим ступенем деталізації та відповідності технічним завданням. На відміну від традиційних методів моделювання, що вимагають значних зусиль кваліфікованих фахівців, генеративні системи здатні автоматизувати процес створення, значно скорочуючи час і ресурси, необхідні для виконання завдання.

Особливу актуальність ці технології набувають у зв'язку зі зростанням попиту на високоякісні 3D-моделі у різних галузях: від архітектурного дизайну та віртуальної реальності до технічного моделювання та ігрової індустрії. Сучасні методи генерації 3D-графіки включають генеративні змагальні мережі (GAN), варіаційні автоенкодери (VAE), трансформери та дифузійні моделі. Кожен із цих підходів має свої переваги й обмеження, що визначає їхню придатність для різних завдань.

Важливим аспектом використання генеративних методів є їх здатність автоматизувати складні процеси створення об'єктів на основі текстових описів або інших вхідних даних. Наприклад, дифузійні моделі, які демонструють найвищу ефективність у завданнях високодеталізованого моделювання,

працюють шляхом поступового видалення шуму з вхідних даних, перетворюючи їх у реалістичні об’єкти. Водночас, правильне налаштування ключових параметрів генерації, таких як кількість кроків дифузії, масштаб керування або текстові промпти, є критичним для досягнення бажаного результату.

Науковий інтерес до цієї галузі обумовлений не лише технологічними досягненнями, але й прагненням зробити процес генерації більш доступним і менш залежним від обчислювальних потужностей. В умовах, коли сучасні генеративні системи, такі як Meshy AI, часто потребують значних ресурсів і працюють на комерційних платформах, виникає необхідність у розробці альтернативних підходів, які дозволять знизити витрати і зробити генерацію доступнішою для широкого кола користувачів.

Генеративні методи мають величезний потенціал для інтеграції в існуючі робочі процеси. В архітектурній візуалізації вони дозволяють автоматизувати процес створення інтер’єрів, у технічному моделюванні сприяють швидкому прототипуванню об’єктів, а в ігровому дизайні забезпечують швидке створення контенту для віртуальних середовищ. Водночас, важливим є врахування практичних аспектів, таких як вибір оптимальних параметрів генерації, що забезпечує баланс між якістю результату та витратами на обчислення.

Ця робота спрямована на всебічне дослідження сучасних генеративних методів 3D-моделювання, аналіз їх переваг і недоліків, а також оцінку їхньої ефективності у вирішенні практичних завдань. Особливу увагу приділено вивченню впливу параметрів генерації на якість моделей, використанню текстових промптів для налаштування характеристик об’єктів та розробці рекомендацій для їх оптимального застосування.

Таким чином, результати цього дослідження сприятимуть подальшому розвитку генеративних методів у комп’ютерній графіці, знижуючи бар’єри для їх використання та відкриваючи нові можливості для автоматизації складних творчих і технічних процесів.

Генеративні методи у 3D-моделюванні. Генеративні методи є важливою частиною сучасного 3D-моделювання, оскільки вони дозволяють створювати тривимірні об’єкти із заданими характеристиками, використовуючи алгоритми штучного інтелекту. Основними типами генеративних методів, які активно використовуються в 3D-графіці, є генеративні змагальні мережі (GAN), варіаційні автоенкодери (VAE), трансформери та дифузійні моделі. Кожен із цих підходів має свої теоретичні основи, принципи роботи та галузі застосування.

Генеративні змагальні мережі (Generative Adversarial Networks) складаються з двох нейронних мереж: генератора $G(z)$ і дискримінатора $D(x)$. Генератор створює нові зразки даних $G(z)$, отримуючи на вхід випадковий шум z, тоді як дискримінатор намагається відрізнити справжні дані x від згенерованих. Мережі навчаються у змагальному режимі, що формалізується через задачу мінімакс:

$$\min_G \max_D E_{x\sim P_{data(x)}} [\log D(x)] + E_{z\sim P_z(z)}[\log(1 - D(x))]$$

Основна перевага GAN полягає у здатності створювати реалістичні об'єкти на основі навчання на великому наборі даних. У 3D-моделюванні GAN використовують для автоматизації процесів текстуризації, створення анімацій та формування геометрії.

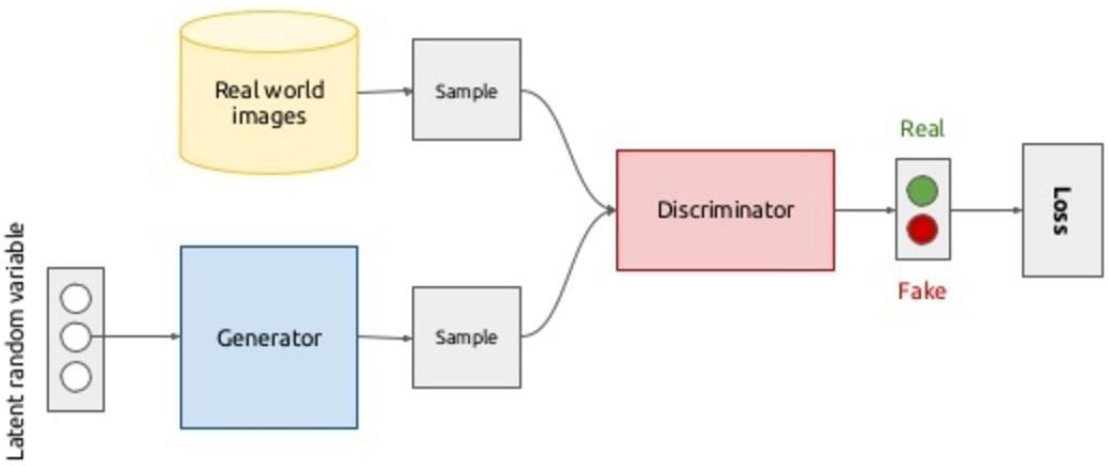


Рис. 1. Принцип роботи змагальних мереж

Варіаційні автоенкодери є підходом до генеративного моделювання, який поєднує ідеї байєсового висновку та глибокого навчання. Архітектура VAE складається з енкодера $q(z|x)$, що перетворює вхідні дані x у латентний простір z, і декодера $p(x|z)$, який реконструює дані з латентного простору. Функція втрат для навчання VAE включає дві частини:

1. **Реконструктивну похибку**, яка вимірює, наскільки відновлені дані $\hat{x}$ схожі на вхідні:

$$L_{recon} = -E_{q(z|x)}[log\ p(x \mid z)].$$

2. **Регуляризацію**, яка забезпечує наближення до нормального розподілу:

$$L_{KL} = D_{KL}(q(z \mid x) \parallel p(z)).$$

Повна функція втрат записується як:

$$L = L_{recon} + \beta L_{KL}$$

VAE використовуються для генерації тривимірних об'єктів із заданими атрибутами, наприклад, створення моделей на основі стилю або текстових описів.

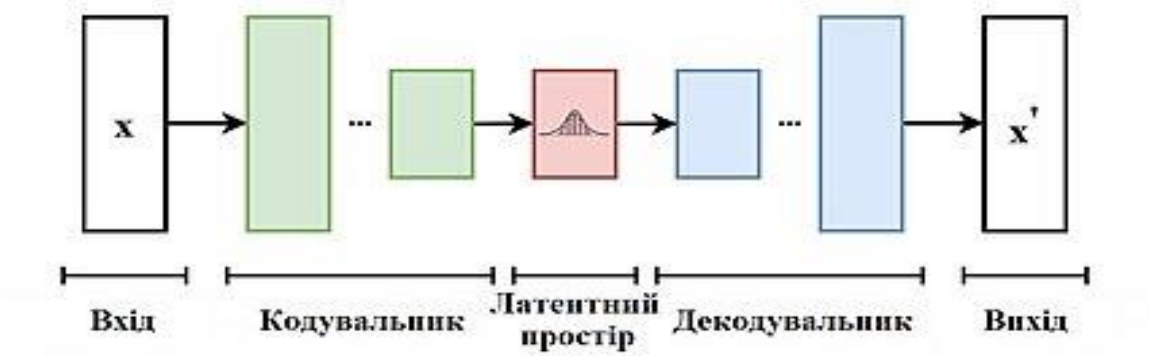


Рис. 2. Принцип роботи варіаційних автоенкодерів

Трансформери стали популярними у 3D-моделюванні завдяки здатності ефективно обробляти послідовності даних та їхню здатність до паралельного навчання. У задачах генерації тривимірних об'єктів трансформери використовують для створення послідовностей точок або трикутників, що визначають геометрію моделі.

Основна ідея трансформера полягає у використанні механізму самозвернення, який визначається як:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$$

де Q, K, V – матриці запитів, ключів і значень, d_k – розмірність ключів.

Завдяки трансформерам, можливо реалізувати генерацію складних об'єктів, таких як 3D-архітектурні моделі чи моделі персонажів.

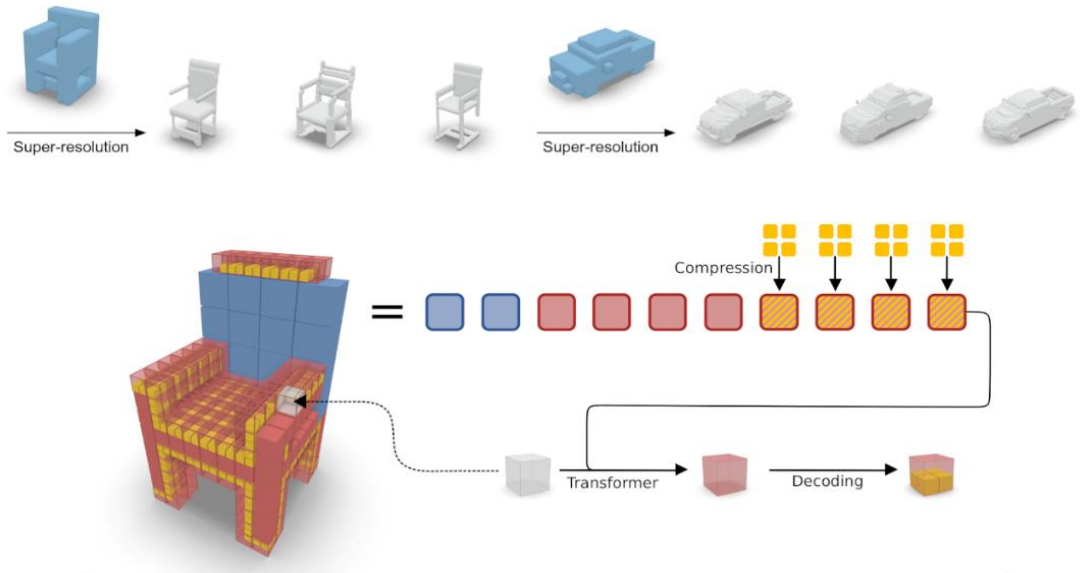


Рис. 3. Принцип роботи варіаційних автоенкодерів

Дифузійні моделі є одним із найбільш перспективних підходів у генеративному моделюванні. Ці моделі базуються на поступовому введенні шуму у дані, що потім видаляється під час генерації. Дифузійний процес моделюється через функції перехідних ймовірностей $q(x_t|x_{t-1})$:

$$q(x_t \mid x_{t-1}) = N(x_t; \sqrt{a_t}\, x_{t-1}, (1 - a_t)I).$$

де α_t – параметр, який визначає рівень шуму на кроці t.

Процес відновлення даних описується функцією:

$$p(x_{t-1} \mid x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \sum\nolimits_\theta (x_t, t)).$$

Генерація 3D-моделей за допомогою дифузійних моделей дозволяє отримувати високоякісні результати, особливо для об’єктів із складною геометрією.

Основною перевагою генеративних методів є їхня гнучкість, можливість створення об’єктів із заданими характеристиками та автоматизація рутинних завдань. Проте виклики, такі як необхідність значних обчислювальних ресурсів і складність налаштування параметрів, залишаються актуальними.

Генеративні методи мають потенціал для інтеграції в різні робочі процеси, зокрема у галузях архітектурного дизайну, ігрової індустрії та медичного моделювання, забезпечуючи значне підвищення ефективності.

Практичне застосування та перспективи. Генеративні методи у 3D-моделюванні відкривають широкі можливості для оптимізації творчих та виробничих процесів, забезпечуючи новий рівень автоматизації й ефективності. Завдяки їх здатності створювати тривимірні об'єкти на основі текстових описів, ці системи стають важливим інструментом у таких галузях, як архітектура, кінематограф, анімація, ігрова індустрія та промисловий дизайн.

Одним із найбільш вагомих напрямів застосування генеративних систем є кінематограф та анімація. Візуальні ефекти (VFX) та анімаційні сцени вимагають детального опрацювання моделей і текстур, що займає багато часу при традиційних підходах. Наприклад, генеративні методи дозволяють створювати унікальні персонажі, такі як *"фентезійний дракон із лускою, яка відблискує на світлі"*, або детальні фони для сцен, наприклад, *"похмуре місто майбутнього з неоновими вивісками"*. У кінематографі, де реалізм і деталізація є критичними, такі інструменти дають змогу прискорити роботу, залишаючи більше часу на творчий етап.

У сфері архітектури генеративні системи використовуються для створення концептуальних проєктів. Їхня здатність швидко генерувати 3D-моделі на основі текстових описів, таких як "мінімалістичний приватний будинок з великими

панорамними вікнами", дозволяє архітекторам швидко візуалізувати ідеї. Водночас використання параметрів, таких як розміри приміщень або стиль інтер'єру, дозволяє досягти високої точності у відповідності до запиту клієнта.

В анімації генеративні моделі сприяють створенню великих обсягів фонів, персонажів та об'єктів. Наприклад, запит "*ліс із фантастичними рослинами, які світяться"* дозволяє отримати унікальні сцени, що не потребують довготривалого ручного моделювання. Водночас такі інструменти надають можливість гнучко налаштовувати параметри для досягнення специфічного художнього стилю, наприклад, стилізованої 2D-анімації або реалістичного 3D.

У ігровій індустрії генеративні методи використовуються для створення об'єктів, персонажів та цілих локацій. Наприклад, генерац*ія "покинутого замку в готичному стилі"* дозволяє отримати детальний 3D-ландшафт для інтеграції в гру. Важливим є те, що генеративні моделі дозволяють створювати безліч варіацій об'єктів, зберігаючи їхню художню якість і відповідаючи стилю гри.

У промисловому дизайні генеративні системи допомагають швидко створювати прототипи, оптимізуючи дизайн під технічні вимоги. Наприклад, промт *"ергономічний стілець для офісу"* може враховувати як естетичні, так і функціональні аспекти моделі. Налаштування параметрів, таких як розміри, матеріали та форма, дозволяють створювати варіанти, оптимальні для виробництва.

Ключовим аспектом ефективного використання генеративних систем є правильне формулювання текстових запитів. Наприклад, при створенні архітектурного дизайну важливо деталізувати запит*: "двоповерховий будинок у готичному стилі із відкритою терасою та басейном" (див. рис 4, 5)*. Для ігрової сцени можна додати опис деталей, таких як "*стара бібліотека з полицями, заповненими книгами"*.

Рис. 4. Двоповерховий будинок із відкритою терасою та басейном

Перспективи генеративних систем включають їх подальше вдосконалення для інтеграції в комплексні робочі процеси, таких як автоматизація підготовки даних для 3D-друку чи інтеграція з інструментами доповненої реальності.

Завдяки цим можливостям генеративні системи продовжуватимуть змінювати галузь 3D-моделювання, роблячи її більш доступною, ефективною та творчою.

Рис. 5. Двоповерховий будинок у готичному стилі із відкритою терасою та басейном

Висновок. Результати дослідження підтверджують значну роль генеративних методів у трансформації процесів 3D-моделювання. У межах проведеної роботи було здійснено глибокий аналіз сучасних генеративних підходів, таких як генеративні змагальні мережі (GAN), варіаційні автоенкодери (VAE) та дифузійні моделі, які демонструють високий потенціал для застосування у сфері 3D-графіки. Одним із ключових аспектів є їх здатність автоматизувати процеси створення складних 3D-об’єктів. Завдяки інтеграції генеративних систем стало можливим не лише оптимізувати виробничі процеси, але й значно розширити межі творчості. Застосування таких методів забезпечує можливість швидкої генерації великої кількості варіантів моделей, що суттєво знижує витрати часу на етапах прототипування та дизайну.

Проведені дослідження також підтвердили ефективність генеративних моделей у створенні об’єктів із заданими характеристиками. Наприклад, налаштування параметрів, таких як рівень деталізації або стиль об’єкта, дозволяє адаптувати результати до потреб конкретного проєкту. При цьому було виявлено важливість точного формулювання запитів (промтів) для отримання бажаних результатів. Наприклад, промти з чітко окресленими деталями, такими як *"футуристичний міст із підвісними ліхтарями",* демонструють вищу ефективність порівняно з загальними описами.

Особливу увагу варто приділити практичному застосуванню генеративних методів у таких галузях, як кінематограф, архітектура та ігрова індустрія. У кінематографі та анімації ці методи дозволяють швидко створювати реалістичні локації, персонажів і спецефекти, зменшуючи потребу в трудомісткому ручному моделюванні. В архітектурі генеративні системи відкривають нові можливості для створення концептуальних проєктів і підвищують ефективність роботи з клієнтами. В ігровій індустрії вони спрощують створення складних ігрових сцен і персонажів, що сприяє зростанню різноманітності контенту.

Значні перспективи мають дослідження оптимальних параметрів генерації. Наприклад, для дифузійних моделей було встановлено, що налаштування кількості ітерацій і рівня шуму суттєво впливають на деталізацію та якість результатів. Зниження кількості ітерацій може прискорити генерацію, проте за рахунок деталізації об'єкта. Водночас високий рівень шуму може призвести до спотворень, що потребує додаткових налаштувань.

У рамках дослідження також визначено рекомендації для інтеграції генеративних методів у робочі процеси. Одним із ключових аспектів є навчання користувачів ефективному формулюванню запитів і розумінню базових принципів роботи генеративних моделей. Це дозволить знизити бар'єри входу для фахівців, які не мають глибоких технічних знань, і сприятиме ширшому впровадженню цих систем.

Таким чином, генеративні методи стають невід'ємною частиною сучасних підходів до 3D-моделювання, забезпечуючи не лише підвищення ефективності робочих процесів, але й розширення творчих можливостей. Подальші дослідження у цій галузі спрямовані на вдосконалення алгоритмів, підвищення точності моделей та їх інтеграцію у нові сфери, що сприятиме ще більшому поширенню генеративних технологій у сучасній індустрії.

ПЕРЕЛІК ПОСИЛАНЬ

1. Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. (2022). *Image Style Transfer Using Convolutional Neural Networks* https://www.cvfoundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf
2. Tero Karras, Samuli Laine, Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. https://www.researchgate.net/publication/338514531_A_Style-Based_Generator_Architecture_for_Generative_Adversarial_Networks
3. Goodfellow, I., Pouget-Abadie, J., Xu, B., Warde-Farley, D., Courville, A., & Bengio, Y. (2014). *Generative Adversarial Networks*. https://doi.org/10.3390/inorganics11030090
4. Brownlee, J. (2023). *What Are Generative Adversarial Networks (GANs)?*. https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
5. Bernstein, M. (2024). *Variational Autoencoders Explained.* https://mbernste.github.io/posts/vae/
6. Karagiannakos, S., & Adaloglou, N. (2022). *How Diffusion Models Work: The Math from Scratch*. https://theaisummer.com/diffusion-models/
7. Sharvit, H.. (2023). *Latent Diffusion Models*. https://www.itshadar.com/latent-diffusion-models
8. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). *Generative Adversarial Networks*. https://arxiv.org/abs/1406.2661
9. *How Transformers Work: A Detailed Exploration of Transformer Architecture.* (2024). https://www.datacamp.com/tutorial/how-transformers-work
10. Bergmann, D. (2024). *What Are Diffusion Models?*. https://www.ibm.com/think/topics/diffusion-models
11. Karagiannakos, S., & Adaloglou, N.. (2022). *How Diffusion Models Work: The Math from Scratch*. https://theaisummer.com/diffusion-models/
12. Bergmann, D. (2024). Meshy AI: Is it the Best 3d Model Generator?. https://medium.com/@bestaitoolz/meshy-ai-is-it-the-best-3d-model-generator-9a20446a12ff

РОЗДІЛ 2

ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ

УДК 004.415.3:681.6

Л.В. Кабак[1], Б.І. Мороз[1], К.С. Родна[1], М.О. Гречкін[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ПОРІВНЯЛЬНИЙ АНАЛІЗ МІКРОСЕРВІСНОЇ ТА МОНОЛІТНОЇ АРХІТЕКТУР

Анотація. Проведено порівняльний аналіз мікросервісної та монолітної архітектури.
Ключові слова: *мікросервісна архітектура, монолітна архітектура, веб-додаток, мікросервіси, моноліт.*

Вступ. У сучасному світі люди очікують від додатків швидкодії, ефективності та надійності. Якщо клієнт має швидкісне інтернет-з'єднання та потрібний додаток, то такі якості, як надійність та ефективність, можуть бути забезпечені лише програмістом. Тому важливо обрати відповідну архітектуру програмного забезпечення перед тим, як реалізовувати функціональності, що були визначені для проекту.

Основний зміст роботи. В останні роки розробники у більшості використовували монолітну архітектуру. У межах моноліту всі модулі – інтерфейс користувача, бізнес-логіка, доступ до даних, служби авторизації та інші – об'єднані в одну програму, яка розгортається на одній платформі як єдине ціле [1]. Переваги монолітної архітектури полягають у простоті розробки та розуміння коду. Оскільки існує єдина кодова база, розробникам легше орієнтуватися в коді та додавати нові зміни. Це спрощує процес розробки, оскільки всі компоненти доступні в одному місці. Крім того, розгортання додатка стає легшим, оскільки потрібно розгорнути лише один файл або пакет. Монолітна архітектура має високу продуктивність за рахунок того, що в централізований кодовій базі та репозиторії, одне API може виконувати таку ж функцію, для якої в інших видах архітектури доводиться використовувати декілька [2]. Це зменшує складність, пов'язану з мережевими протоколами і серіалізацією даних, що позитивно впливає на швидкодію системи. Спрощене тестування досягається завдяки тому, що всі компоненти знаходяться в одному додатку, що полегшує налаштування тестового середовища та відслідковування помилок (рис.1).

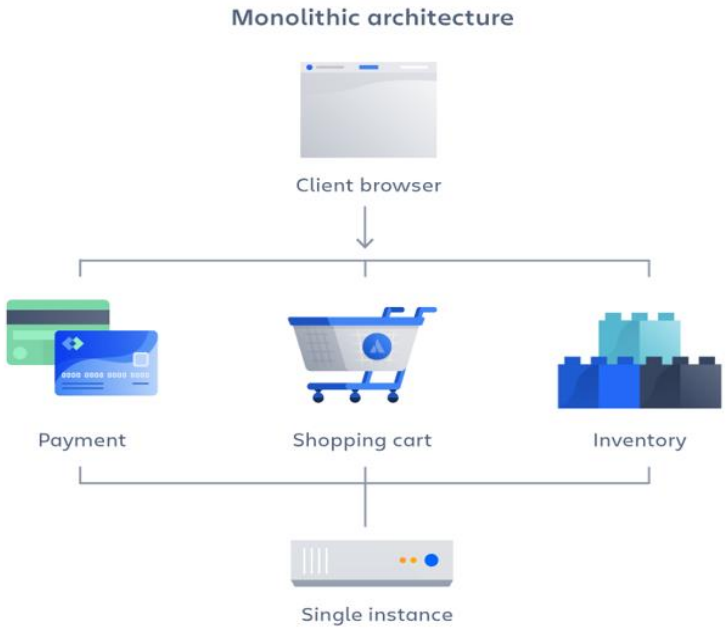


Рис. 1. Схема монолітної архітектури

Однак монолітна архітектура має і недоліки. Складність підтримки та розвитку збільшується зі зростанням розмірів додатка. Кодова база може стати громіздкою та важкою для розуміння, а внесення змін може призвести до непередбачуваних наслідків в інших частинах системи, що ускладнює підтримку та розвиток. Обмежена масштабованість є ще одним недоліком: монолітні додатки важко масштабувати ефективно. Якщо один модуль стає «вузьким місцем», неможливо масштабувати лише його; потрібно масштабувати весь додаток, що може бути неекономічно [3].

Низька гнучкість у виборі технологій обмежує можливість впровадження нових, більш підходящих технологій для окремих компонентів, оскільки всі модулі моноліту зазвичай написані на одній мові програмування та використовують однакові технології. Ризик збоїв зростає, адже помилка в одному модулі може призвести до відмови всього додатку. Відсутність ізоляції між компонентами підвищує ризик простоїв та знижує надійність системи. Конфлікти при злитті коду, різні стандарти кодування та комунікаційні проблеми можуть негативно впливати на продуктивність розробки. Повільна адаптація до змін ринку через складність та ризики, пов’язані з внесенням змін, ускладнює впровадження нових функцій або реагування на зміну вимог ринку, що знижує конкурентоспроможність бізнесу. Простота архітектури дозволяє командам швидко створювати функціональний продукт без значних початкових витрат на інфраструктуру та організацію процесів. Однак зі зростанням проєкту виникають проблеми масштабування та підтримки. Наприклад, при високих навантаженнях необхідно збільшувати ресурси для всього додатка, навіть якщо лише окремі модулі потребують додаткової продуктивності, що призводить до неефективного використання ресурсів та ріст витрат на інфраструктуру [4].

Відсутність модульності ускладнює впровадження нових технологій. Якщо з'являється потреба використовувати нову базу даних або фреймворк для певного модуля, це може бути складно або неможливо в контексті моноліту без значних переробок. У контексті DevOps практик монолітна архітектура може стати перешкодою. Автоматизація розгортання, безперервна інтеграція та доставка (CI/CD) ускладняються через розмір та складність додатка. Цикли розгортання стають довшими, що суперечить принципам швидкої та гнучкої розробки.

Мікросервісна архітектура – це підхід до розробки програмного забезпечення, при якому додаток складається з набору невеликих, автономних сервісів, що взаємодіють між собою через чітко визначені інтерфейси (API). Кожен сервіс відповідає за конкретну бізнес-функцію і може розроблятися, розгортатися та масштабуватися незалежно від інших. Цей підхід базується на принципах розподілених систем і спрямований на підвищення гнучкості, масштабованості та швидкості розробки складних програмних продуктів.

Переваги мікросервісної архітектури полягають у її здатності забезпечувати високу масштабованість та гнучкість системи. Оскільки сервіси є незалежними, вони можуть масштабуватися окремо відповідно до навантаження, що дозволяє ефективніше використовувати ресурси та оптимізувати продуктивність. Такий підхід також сприяє швидшому циклу розробки та розгортання: команди можуть працювати паралельно над різними сервісами, впроваджуючи нові функціональні можливості без необхідності пере розгортання всього додатка. Мікросервіси дозволяють використовувати різні технологічні стеки та мови програмування для кожного сервісу (рис.2), що надає розробникам свободу вибору інструментів, найбільш підходящих для конкретних задач [5].

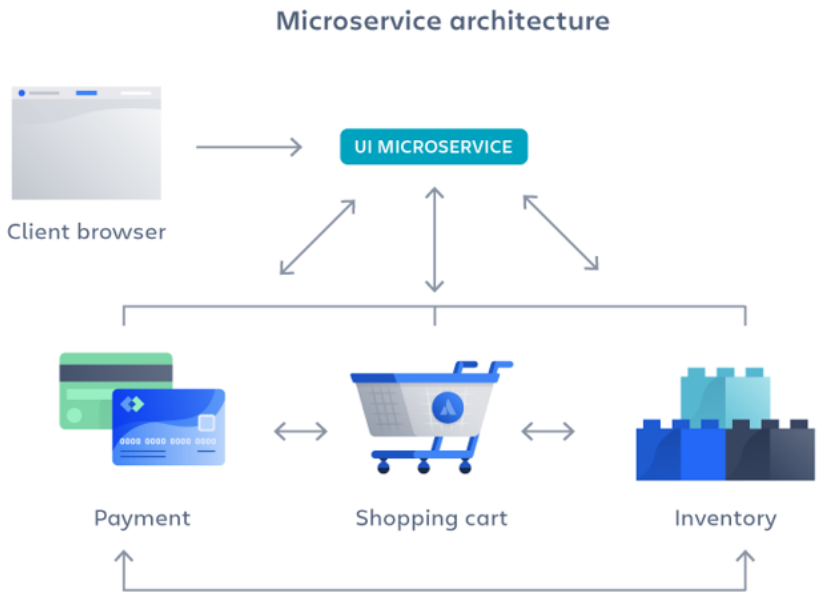


Рис. 2. Схема мікросервісної архітектури

Крім того, мікросервісна архітектура підвищує стійкість системи до збоїв. Якщо один сервіс виходить з ладу, це не обов’язково впливає на роботу інших сервісів, що забезпечує безперебійну роботу додатка в цілому. Це також полегшує процес оновлення та підтримки: виправлення помилок або впровадження нових функцій можна здійснювати на рівні окремих сервісів без ризику порушити роботу всієї системи.

Переваги мікросервісної архітектури полягають у її здатності забезпечувати високу масштабованість та гнучкість системи. Оскільки сервіси є незалежними, вони можуть масштабуватися окремо відповідно до навантаження, що дозволяє ефективніше використовувати ресурси та оптимізувати продуктивність. Такий підхід також сприяє швидшому циклу розробки та розгортання: команди можуть працювати паралельно над різними сервісами, впроваджуючи нові функціональні можливості без необхідності перерозгортання всього додатка. Мікросервіси дозволяють використовувати різні технологічні стеки та мови програмування для кожного сервісу, що надає розробникам свободу вибору інструментів, найбільш підходящих для конкретних задач [5].

Крім того, мікросервісна архітектура підвищує стійкість системи до збоїв. Якщо один сервіс виходить з ладу, це не обов’язково впливає на роботу інших сервісів, що забезпечує безперебійну роботу додатка в цілому. Це також полегшує процес оновлення та підтримки: виправлення помилок або впровадження нових функцій можна здійснювати на рівні окремих сервісів без ризику порушити роботу всієї системи.

Якщо ж говорити про недоліки, то одним з основних є підвищена складність управління системою. Розподілена природа мікросервісів ускладнює налагодження, моніторинг та тестування додатка. Комунікація між сервісами відбувається через мережу, що може призводити до затримок, проблем з пропускною здатністю та необхідності обробки помилок мережі. Це вимагає впровадження складних механізмів маршрутизації, балансування навантаження та управління мережею.

Забезпечення цілісності даних та транзакцій стає більш складним завданням, оскільки дані можуть зберігатися в різних базах даних, керованих різними сервісами. Це ускладнює підтримку узгодженості даних і вимагає використання розподілених транзакцій або патернів, таких як сага, для управління складними бізнес-процесами.

Тестування мікросервісів також стає більш складним. Необхідно не тільки перевіряти коректність роботи кожного окремого сервісу, але й забезпечувати правильну взаємодію між ними. Це вимагає налаштування складних тестових середовищ та розробки інтеграційних і end-to-end тестів. Крім того, автоматизація процесів розгортання, моніторингу та управління конфігураціями стає критично важливою, що може збільшити витрати на інфраструктуру та вимоги до кваліфікації персоналу.

Також варто зазначити, що розподілена природа мікросервісів може призвести до підвищених витрат на мережеві ресурси та інфраструктуру. Комунікація між сервісами може стати «вузьким місцем», особливо при високих навантаженнях, що вимагає додаткових зусиль для оптимізації та забезпечення стабільної роботи системи.

Висновки. Вибір між монолітною та мікросервісною архітектурами залежить від специфіки проєкту, його масштабів, ресурсів та стратегічних цілей. Монолітна архітектура є оптимальним вибором для невеликих або середніх проєктів, де важливі швидкий старт, простота розробки та розгортання. Вона підходить для команд з обмеженими ресурсами, які бажають швидко вивести продукт на ринок без значних інвестицій у складну інфраструктуру. Моноліт дозволяє зосередитися на функціональності, забезпечуючи при цьому просте управління кодовою базою та тестування.

З іншого боку, мікросервісна архітектура стає доцільною для великих, складних систем, які вимагають високої масштабованості, гнучкості та стійкості до збоїв. Вона підходить для проєктів з великими командами розробників, де необхідно паралельно працювати над різними компонентами, використовуючи різні технологічні стеки. Мікросервіси дозволяють масштабувати окремі сервіси незалежно, оптимізуючи використання ресурсів і підвищуючи продуктивність системи. Однак цей підхід вимагає значних інвестицій у інфраструктуру, автоматизацію процесів розгортання, моніторингу та управління конфігураціями.

ПЕРЕЛІК ПОСИЛАНЬ

1. Newman, S. (2021). *Building Microservices: Designing Fine-Grained Systems* (2nd ed.). O'Reilly Media.
2. *Monolithic vs microservice architecture*: https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith.
3. Zabierowski. W. *The Comparison of Microservice and Monolithic Architecture*: https://www.researchgate.net/publication/341956559-The-Comparison-of-Microservice-and-Monolithic-Architecture.
4. *5 advantages of microservices architecture*: https://www.atlassian.com/microservices/cloud-computing/advantages-of-microservices.
5. Dushenin O. *Monolithic Architecture. Advantages and Disadvantages*: https://datamify.medium.com/monolithic-architecture-advantages-and-disadvantages-e71a603eec89.

УДК 004.415.3:681.6

М.К. Чурсина[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ЗАСТОСУВАННЯ АЛГОРИТМІВ ХЕШУВАННЯ ДЛЯ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ДАНИХ В SQL SERVER

Анотація. Розглянуто вбудовані можливості хешування у системі управління реляційними базами даних Microsoft SQL Server. Зроблено порівняння підтримуваних алгоритмів хешування з аналізом щодо безпеки їх використання для конкретних задач.

Ключові слова: *інформаційна безпека, цілісність, хешування, колізія, хеш-сума, MD5, SHA-1, SHA-256, SHA-512.*

Вступ. Забезпечення цілісності даних є однією з ключових задач інформаційної безпеки, особливо у системах обробки та зберігання даних, таких як бази даних. В епоху зростання кіберзагроз перевірка автентичності даних та захист від несанкціонованих змін стають критично важливими.

Хешування, як фундаментальний інструмент криптографії, пронизує сучасний цифровий світ, залишаючись при цьому невидимим для більшості користувачів. Від простих операцій аутентифікації до складних механізмів забезпечення цілісності даних, хешування відіграє ключову роль у захисті інформації.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- дослідити вбудовану функцію Microsoft SQL Server, що використовуються для хешування даних;
- визначити підтримувані алгоритми хешування для забезпечення цілісності даних;
- виконати порівняння алгоритмів за критеріями стійкості до атак на колізії;
- дослідити швидкість роботи алгоритмів хешування для різних обсягів даних;
- виконати моніторинг використання ресурсів системи під час хешування даних;
- зробити висновок щодо використання алгоритмів хешування до конкретних задач.

Основний зміст роботи. Хешуванням називають процес перетворення вхідних даних у рядок символів фіксованого розміру, що являє собою унікальний відбиток оригінальних даних. Дані проходять через математичний алгоритм, який створює унікальне значення хешу. Алгоритм бере вхідні дані і виконує серію обчислень, щоб перетворити їх у рядок символів фіксованого розміру.

Це перетворення є детермінованим і гарантує, що певний вхідний сигнал незмінно призводить до того самого хеш-виходу. І, при цьому, має односторонній характер, який унеможливлює отримання вхідного сигналу з хеш-значення.

Microsoft SQL Server, як одна з найпопулярніших систем управління базами даних, пропонує потужні інструменти для реалізації хешування. Вбудована функція HASHBYTES() [1] дозволяє обчислювати хеш-суми за різними алгоритмами. Аргументами даної функції є обраний алгоритм та змінна, що містить дані для хешування. Функція повертає значення varbinary, де зберігаються двійкові дані в форматі змінної довжини. Функція HASHBYTES() має обмеження розміру вхідних даних – 8000 байтів, тому необхідним є розробка стратегії ефективних методів обробки даних, яку буде розглянуто пізніше. Вбудована функція обчислення хеш-суми HASHBYTES() підтримує наступні алгоритми: MD2, MD4, MD5, SHA, SHA-1, SHA-256, SHA-512.

Офіційний стандарт безпечних хеш-функцій США FIST PUB 180-4 [2] визначає сукупність критеріїв та алгоритмів, що гарантують високий рівень безпеки хешування. Зазначеними в цьому документі є алгоритми сімейства SHA. Відсутність інших алгоритмів хешування в цьому документі може свідчити про те, що ці алгоритми більше не вважаються безпечними і не рекомендуються для використання в нових системах. Подальшому розгляду в рамках дослідницької роботи підлягають алгоритми хешування MD5, SHA-1, SHA-256, SHA-512.

Досліджувані алгоритми засновані на ітераційній схемі ітераційній схемі Меркаля-Демгарда [3], ідеєю якого є функція стиснення. Вхідні значення поділяють на блоки, кратні сталої величини, шляхом додавання біта “1” та необхідної кількості бітів “0”. Перед початком хешування виконується ініціалізація набору 32-бітних регістрів сталої величини. І вже після цього виконується сам алгоритм з відповідною кількістю ітерацій, використовуючи різні операції. В таблиці 1 виконано порівняльну характеристику досліджуваних алгоритмів.

Застосування хеш-функцій відіграє ключову роль у багатьох аспектах обробки даних. Шляхом обчислення хешу від таблиць або окремих рядків можна відстежувати зміни. У роботі з великим обсягом даних є доцільним виконання запитів на пошук полів за індексом, створеним на основі хеш-функції. Після відновлення резервної копії даних є важливим перевірка хешу даних на збіг для уникнення втрати даних.

Одним із найважливіших критеріїв при виборі алгоритму хешування для обробки даних є стійкість до колізії [4]. Колізією називається ситуація, коли для двох різних вхідних даних обчислюється однакове значення хеш-суми. Наявність таких ситуацій може призвести до серйозних помилок в обробці даних, таких як втрата та деформація даних. Одним із типових прикладів є некоректна ідентифікація записів в таблиці за унікальним ідентифікатором, що містить хеш значення. Якщо два поля мають однакове значення хеш-суми через колізію, то система може змішати дані при оновлені запису.

Таблиця 1

Порівняльна характеристика досліджуваних алгоритмів хешування

Назва	Рік розробки	Автори	Розмір блоку	Кількість регістрів станих	Кількість ітерацій	Застосовані над операції блоками	Вихідне хеш значення
MD5	1991 рік	Рон Ріверст	512 біт	4	4 раунди по 16 кроків	Логічні операції та зсуви	128 біт, результат об'єднання виходу регістрів
SHA-1	1995 рік	Агентство національної безпеки США разом з Національним інститутом стандартів та технологій	512 біт	5	4 раунди по 20 кроків	Логічні операції, зсуви та додавання	160 біт, результат конкатенації регістрів
SHA-256	2002 рік		512 біт	8	64 кроки	Логічні операції, зсуви та окремі нелінійні функції	256 біт, результат об'єднання виходу регістрів
SHA-512			1024 біт	8	80 кроків		512 біт, результат об'єднання виходу регістрів

На міжнародній конференції з теорії та застосування криптографічних методів від травня 2005 року група китайських вчених та математиків продемонстрували свою роботу над дослідженням вразливостей алгоритму хешування MD5 до колізії [5]. У своїй роботі вони використали метод диференціального криптоаналізу для визначення впливу змін вхідних даних на зміни в результаті хешування, звівши проблему пошуку колізії до вирішення системи нелінійних рівнянь. Після цього відкриття MD5 більше не вважається безпечним алгоритмом хешування.

Пізніше, у серпні цього ж року, була опублікована робота [6], в якій автори описали розроблений метод атаки, що дозволив їм знайти колізії в повному 80-крововому алгоритмі SHA-1 зі складністю, менше ніж 2^{69} хеш операцій, коли

межею є 2^{80}. Розроблений новий підхід до диференціального криптоаналізу з використанням властивостей булевих функцій дозволив знайти слабкі місця в структурі SHA-1.

Одним з важливих викликів при роботі з даними є забезпечення високої швидкості обробки запитів. Ефективність алгоритмів значною мірою залежить від розміру оброблювальних даних. В рамках дослідження за допомогою модулю hashlib мови програмування Python було реалізоване тестування швидкості роботи алгоритмів хешування MD5, SHA-1, SHA-256 та SHA-512 для даних розміром 100 Мб, 1024 Мб та 10240 Мб. Отримані результати представлені на гістограмі впливу розміру даних на швидкість роботи алгоритмів хешування (рис. 1).

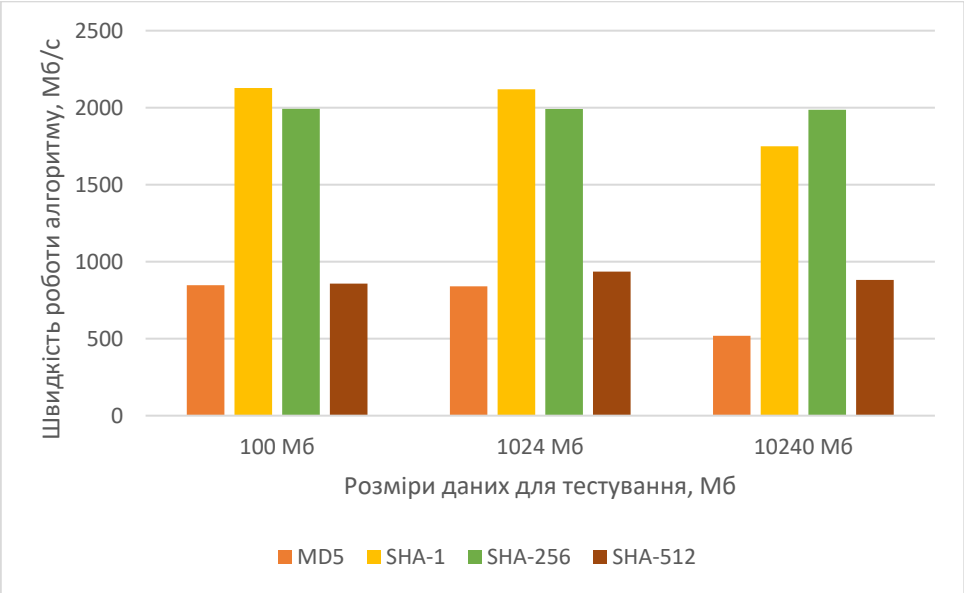


Рис. 1. Гістограма впливу розміру даних на швидкість роботи алгоритмів хешування

Продуктивність алгоритмів MD5 та SHA-1 на достатньо великих обсягах даних падає, що пов’язано з недостатньою оптимізацією та затримки через стару архітектуру даних алгоритмів. Алгоритми SHA-256 та SHA-512 краще оптимізовані для сучасних процесорів для роботи з даними через структуру, що пристосована для паралельної обробки.

Для більш детального розуміння як алгоритми хешування впливають на систему проведено моніторинг використання системних ресурсів у реальному часі. За допомогою модулю psutil мови програмування Python здійснено моніторинг навантаження ЦП та використання оперативної пам’яті під час виконання алгоритмів хешування на даних об’ємом 10 Гб. Вимірювання проводиться за інтервалом 0,1 секунди. Результати моніторингу навантаження системних ресурсів у форматі графіків проілюстровані на рис. 2.

Всі алгоритми демонструють значні коливання використання ЦП протягом часу виконання, що пов’язані зі структурою алгоритмів, розміром блоків

обробки даних. Алгоритм MD5 демонструє більш стабільне навантаження ЦП, оскільки використовує відносно простий набір операцій в роботі. Алгоритм сімейства SHA мають більш складну структуру та відображають більші коливання навантаження.

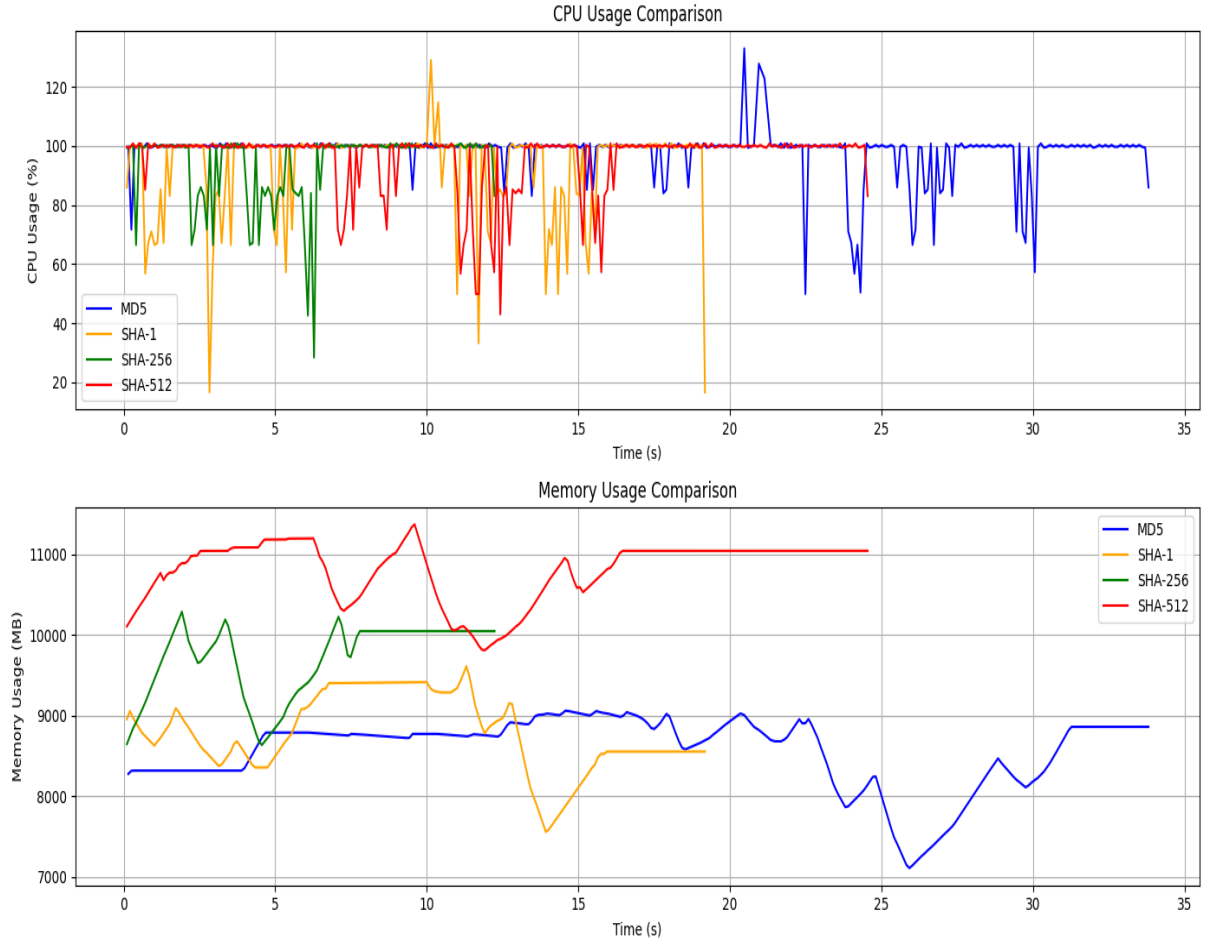


Рис. 2. Динаміка використання ЦП та пам’яті при виконанні алгоритмів хешування MD5, SHA-1, SHA-256 та SHA-512

Використання пам’яті для всіх алгоритмів зростає з часом і досягає стабільного рівня після завершення обчислень, що пояснюється виділенням пам’яті під проміжні результати обчислень. Значно більший обсяг пам’яті використовують алгоритми SHA-256 та SHA-512 через більшу кількість обчислень та проміжних результатів.

На основі проведених досліджень надано наступні практичні рекомендації щодо використання алгоритмів хешування:

1.Алгоритм MD5 слід використовувати лише для невеликих внутрішніх задач, де безпека не є критичною, через свою низьку стійкість до колізій та структуру, що не є оптимізованою для роботи з великим обсягом даних. Прикладом може стати перевірка контрольних сум файлів або блоків даних.

2.Алгоритм SHA-1 є достатньо швидким при обробці даних різного розміру, але через свою низьку стійкість до колізії не рекомендований до використання для критично важливих задач.

3.Алгоритм SHA-256 є найбільш ефективним для роботи з великим обсягом даних, має високу стійкістю до атак. Рекомендовано використовувати для роботи з перевіркою цілісності резервних копій без даних.

4.Алгоритм SHA-512 має найвищу стійкість до атак серед перерахованих алгоритмів. Рекомендовано застосовувати для завдань з найвищими вимогами до безпеки.

Наукова новизна полягає у комплексному досліджені вбудованих функцій хешування у контексті забезпечення цілісності даних в Microsoft SQL Server з наданням рекомендацій щодо оптимізації використання алгоритмів хешування.

Висновки. У роботі було виконане дослідження вбудованої функцій Microsoft SQL Server для виконання хешування даних. Визначено підтримувані алгоритми хешування для забезпечення цілісності при обробці даних. Для оцінки та подальшого сформування рекомендацій щодо застосування алгоритмів хешування для конкретних задач було виконане тестування швидкості алгоритмів та моніторинг використання системних процесів під час хешування даних за допомогою окремих модулів мови програмування Python.

ПЕРЕЛІК ПОСИЛАНЬ

1. Офіційна документація Microsoft SQL Server, HASHBYTES. – [Електронний ресурс] Режим доступу : https://learn.microsoft.com/ru-ru/sql/t-sql/functions/hashbytes-transact-sql?view=sql-server-ver16
2. FIST PUB 180-4: Secure Hash Standard. – [Електронний ресурс] Режим доступу: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.180-4.pdf
3. Harshvardhan Tiwari. Merkle-Damgård Construction Method and Alternatives: A Review – Journal of Information and Organizational Sciences 41(2):283-304
4. Козак Р.О. Методи пошуку колізій криптографічних хеш-функцій, 2011.
5. Xiaoyun Wang and Hongbo Yu. How to Break MD5 and Other Hash Functions – Advances in Cryptology – EUROCRYPT 2005, pp. 19-35. – [Електронний ресурс] Режим доступу: https://www.iacr.org/archive/eurocrypt2005/34940019/34940019.pdf.
6. Xiaoyun Wang, Yiqun Lisa Yin and Hongbo Yu. Finding Collisions in the Full SHA-1 – Advances in Cryptology – CRYPTO 2005, pp. 17-36. – [Електронний ресурс] Режим доступу: https://www.iacr.org/archive/crypto2005/36210017/36210017.pdf.

УДК 004.415.3:681.6

В.В. Спірінцев[1], А.Л. Ширін[1], Д.В. Курілех[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ЗНАЧЕННЯ ТА ВПЛИВ ШРИФТІВ У ВЕБРОЗРОБЦІ

Анотація. Описано важливість підбору шрифтів при розробці вебзастосунків та розглянуто основні поради вибору шрифтів для сайтів.

Ключові слова: *шрифт, вебдизайн, підбір шрифту, зручне сприйняття, вебсторінка, типографіка, UI/UX.*

Вступ. Типографіка є однією з найважливіших складових вебдизайну, що відіграє вирішальну роль у створенні ефективного користувацького досвіду. У сучасному світі, де інформація сприймається в основному через текстовий контент, вибір шрифтів може значно впливати на те, як користувачі сприймають вебсторінки та взаємодіють з ними. Відповідний шрифт може покращити читабельність тексту, викликати позитивні емоції у користувачів і навіть підвищити їхню довіру до контенту. Навпаки, неправильно обраний шрифт може зробити текст важким для сприйняття, викликати роздратування або навіть призвести до втрати аудиторії.

Основний зміст роботи. Сьогодні питання вибору шрифтів стало більш комплексним, враховуючи не лише естетичні, а й технічні аспекти, такі як доступність та адаптація до різних пристроїв. Різні шрифти можуть викликати різні асоціації та емоції у користувачів, тому важливо враховувати ці фактори для створення позитивного користувацького досвіду [1].

Однією з основних проблем вебдизайну є недостатня увага до вибору шрифтів. Дизайнери часто віддають перевагу візуальній привабливості шрифту, нехтуючи його читабельністю або психологічним впливом на користувача. Неправильний вибір може зробити сайт невиразним і незапам'ятовуваним. В результаті текст може викликати у користувача негативні емоції або бути важким для читання, що погіршує загальне враження від сайту.

Використання непродуманих шрифтів може призвести до відмови від подальшого перегляду сайту. Це особливо важливо для ресурсів, які мають на меті залучення і утримання аудиторії, таких як новинні портали або онлайн-магазини. Чималу роль відіграє відсутність адаптивності - багато шрифтів не відображаються коректно на різних пристроях, що може створити труднощі в читанні, особливо на мобільних пристроях. Це впливає на досвід користувачів, які звикли до зручного перегляду на різних екранах [2].

Використання несумісних шрифтів може призвести до хаосу в дизайні. Наприклад, комбінація складних декоративних шрифтів з простими може відволікати увагу користувачів від контенту (рис.1).

Рис. 1. Поєднання стандартних та декоративних шрифтів у тексті

Розглянемо два різні шрифти для порівняння їхнього впливу на користувачів:

1) Arial є одним із найпоширеніших шрифтів, який використовується завдяки своїй нейтральності та зручності для читання (рис.2) тому він вважається стандартним для усіх документів Європи. Він підходить для будь-якого контенту, але його надмірне використання може зробити сайт невиразним.

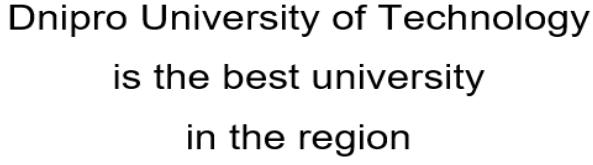


Рис. 2. Шрифт Arial

2) Times New Roman є традиційним і асоціюється з офіційністю та професійністю (рис.3). Він вважається стандартом для усіх документів України. Хоча він ідеально підходить для текстів великого обсягу, у певних контекстах він може здаватися застарілим або надто формальним.

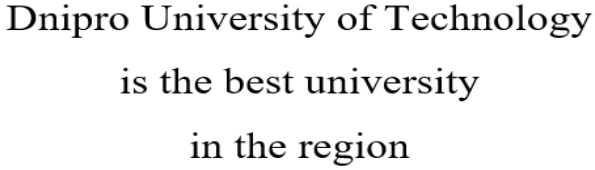


Рис. 3. Шрифт Times New Roman

Щоб уникнути поставлених проблем щодо підбору шрифтів, необхідно керуватися комплексним підходом, який включає в себе:

1. Аналіз аудиторії та контенту. Цей крок є важливим аспектом перед вибором шрифту. Важливо розуміти, хто є вашою цільовою аудиторією та який тип контенту буде представлений на сайті. Наприклад, для молодіжних порталів варто обирати сучасні, легко читабельні шрифти, тоді як для офіційних сайтів доречні шрифти, що викликають довіру;
2. Читабельність шрифтів. Для основного тексту слід використовувати шрифти, які добре сприймаються на екранах будь-якого розміру. Такі шрифти, як Open Sans, Roboto, або Lato, забезпечують високу читабельність навіть при різних розмірах екрана. Рекомендується проводити тестування шрифтів на

різних пристроях і при різних умовах освітлення, щоб переконатися, що текст залишається легко читабельним;

3. Аналіз впливу шрифтів на емоційний стан. Шрифти мають здатність впливати на емоції користувачів. Необхідно провести аналіз і вибрати шрифти, які підсилюють довіру, комфорт або інші бажані емоційні реакції користувачів. Регулярно аналізуйте використовувані шрифти, зважаючи на зворотний зв'язок від користувачів, щоб переконатися, що вони відповідають поточним потребам аудиторії;

4. Використання адаптивних шрифтів. Необхідно застосовувати шрифти, які автоматично адаптуються до розмірів екрана, такі як Google Fonts [5] або Adobe Fonts [6], тестувати їх на різних платформах і пристроях, щоб переконатися в коректному відображенні та збереженні читабельності.. Вони забезпечують високу якість відображення на всіх пристроях;

5. Використання медіа-запитів CSS. У верстці слід впроваджувати медіа-запити для зміни розміру, міжрядкових інтервалів та інших параметрів шрифту залежно від розміру екрану. Це забезпечить оптимальне відображення тексту на всіх пристроях [4];

6. Створення гармонійних комбінацій шрифтів. Важливо розробити стратегію вибору комбінацій шрифтів, які доповнюють один одного. Наприклад, для заголовків можна використовувати більш виразні шрифти, тоді як для основного тексту слід обирати прості та легко читабельні [3].

Висновки. Вибір шрифтів є критично важливим елементом веброзробки, який впливає на користувацький досвід. Дизайнери повинні усвідомлювати, як шрифти впливають на сприйняття і емоції користувачів, і враховувати це при розробці вебресурсів. В цей час розробники мають правильно налаштувати виведення шрифтів. Використання відповідних шрифтів може покращити читабельність і загальне враження від сайту, що, у свою чергу, впливає на його успіх.

ПЕРЕЛІК ПОСИЛАНЬ

1. Типографіка в UI/UX: Ключові аспекти та практичні поради. [електронний ресурс]. URL: https://dou.ua/forums/topic/43688 (дата звернення: 20.09.2024);
2. Значення типографії в сучасному вебдизайну. [електронний ресурс]. URL: https://coi.ua/blog/DesignCo/the-significance-of-typography-in-modern-web-design (дата звернення: 20.09.2024);
3. Важливість вибору шрифту в дизайні інтерфейсів. [електронний ресурс]. URL: https://www.seobanda.com/uk/blog/fonts-in-design (дата звернення: 20.09.2024);
4. CSS Медіа запити. [електронний ресурс]. URL: https://w3schoolsua.github.io/css/css3_mediaqueries_ex.html#gsc.tab=0 (дата звернення: 20.09.2024);
5. Google Fonts: Browse fonts. [електронний ресурс]. URL: https://fonts.google.com (дата звернення: 20.09.2024);
4. Adobe Fonts | Explore unlimited fonts. [електронний ресурс]. URL: https://fonts.adobe.com (дата звернення: 20.09.2024).

УДК 004.896:519.6:535.6

А.В. Клименко[1], В.В. Анісімов[2], В.В. Рулікова[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна
[2]ННІ «Український державний хіміко-технологічний університет» УДУНТ

РЕАЛІЗАЦІЯ НАПРЯМЛЕННЯ РОБОТА ВЗДОВЖ КОЛЬОРОВОЇ ЛІНІЇ ЗА ПРОПОРЦІЙНИМ АЛГОРИТМОМ НА БАЗІ ROMEO BLE

Анотація. Представлено реалізацію руху робота вздовж кольорової лінії за пропорційним алгоритмом з використанням плати Romeo BLE. Описано апаратну складову та алгоритм, що враховує зовнішнє освітлення

Ключові слова: *робототехніка, автоматизація, пропорційний регулятор, Romeo BLE, навчання, алгоритм відсікання освітлення.*

Вступ. Роботизація та автоматизація є ключовим напрямком технологічного розвитку сьогодні. Одним із цікавих завдань в робототехніці є реалізація автономного руху робота вздовж кольорової лінії, що може знаходити своє застосування в логістичних процесах, таких як управління складськими приміщеннями. У роботі пропонується програмна реалізація руху робота на основі плати Romeo BLE, що дозволяє вирішувати задачу з урахуванням зовнішніх умов освітлення.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані і вирішені такі завдання:

- розробити програмне забезпечення для напрямлення робота вздовж кольорової лінії з використанням пропорційного регулятора;
- забезпечити незалежність роботи системи від зовнішніх джерел освітлення за допомогою алгоритму відсікання;
- підвищити ефективність програмного забезпечення шляхом його структурування.

Основний зміст роботи. У сучасному світі автоматизація та роботизація набувають все більшого значення. Один з перспективних способів реалізації руху робота є його рух вздовж лінії певного кольору. Такий принцип вже зараз можна побачити у складських приміщеннях світових логістичних компаній. Тому актуальним питанням є програмна реалізація руху робота вздовж кольорової прямої та демонстрація цього молодому поколінню, щоб виховати для країни покоління висококваліфікованих фахівців.

Апаратною основою для розробленої програми є плата Romeo BLE з наступними підключеними компонентами (рис. 1).

Якщо увімкнути підсвітку світлодіодом 2 та виміряти інтенсивність освітлення датчиком, то ми отримаємо показники, що враховують зовнішнє освітлення та освітлення від світлодіода. В такому випадку датчик відбитого світла ***чутливий*** до зовнішніх джерел освітлення.

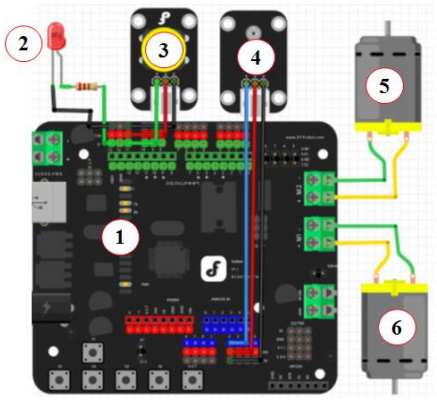


Рис. 1. Схема підключення електронних компонентів:
1 – плата Romeo BLE;
2 – світлодіод;
3 – кнопка:
4 – датчик освітленості;
5 – двигун лівого повороту;
6 – двигун правого повороту.

У програмі при визначенні інтенсивності відбитого світла використовуємо алгоритм відсікання зовнішнього освітлення:

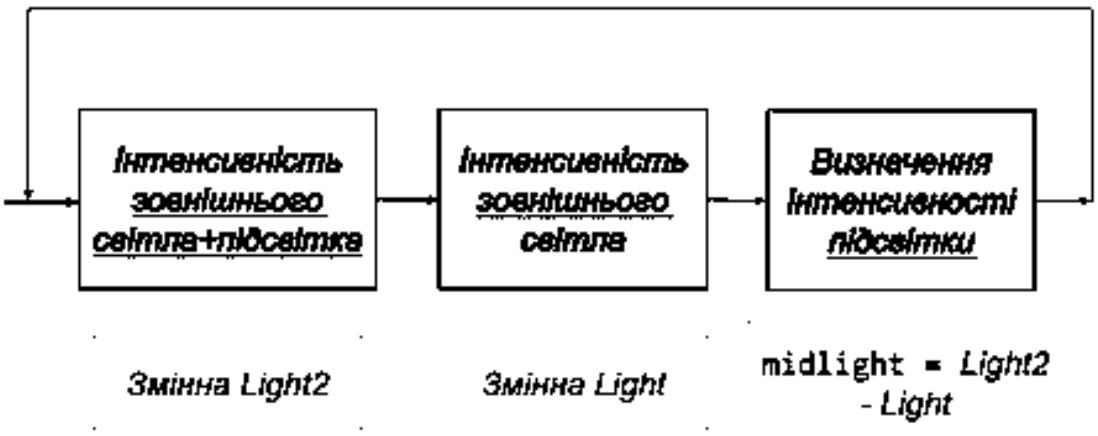


Рис. 2. Алгоритм відсікання зовнішнього освітлення

Пропорційний регулятор – це такий регулятор, у якого чим сильніший зовнішній вплив, тим сильніше опір цьому впливу.

В даному випадку інтенсивність зовнішнього впливу визначається зміною інтенсивності відбитого світла: чим менша/більша інтенсивність – тим більшим є зовнішній вплив.

В результаті маємо наступний алгоритм, закладений у програму:

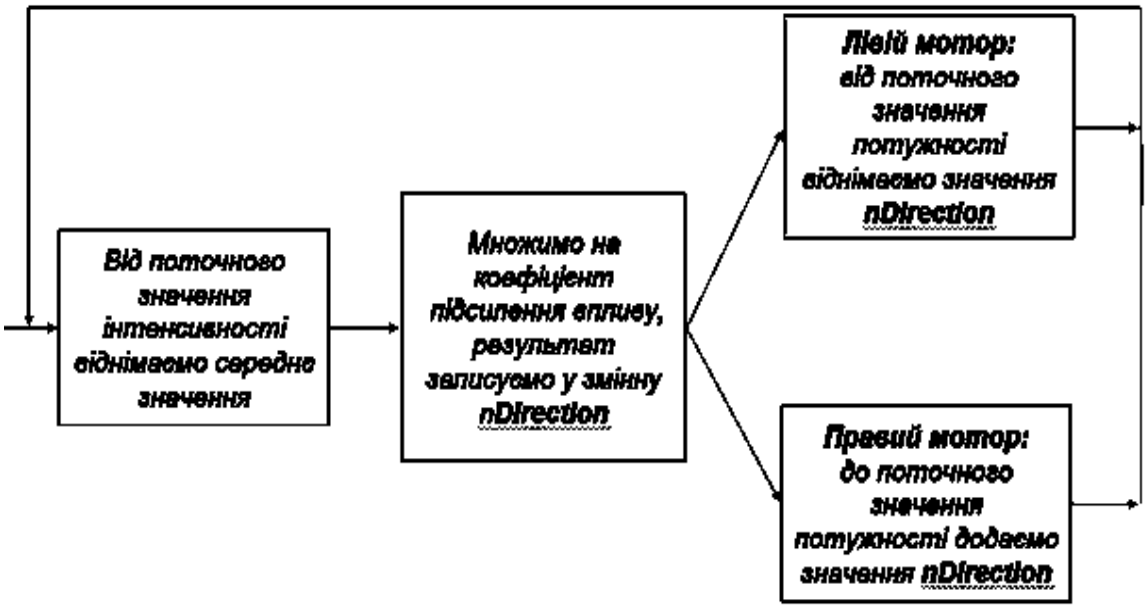


Рис. 3. Загальний алгоритм роботи програми

Реалізація описаного алгоритму є основою комп'ютерної програми для напрямлення робота вздовж кольорової лінії за пропорційним алгоритмом. Наведемо основні компоненти програмного коду даної програми. Нижче представлено основну функцію програми.

```
void loop(){
  while (!digitalRead(Button)) {
    int deviation = reflectLight() - 25;
    int nDirection = deviation * 3;
    motors_run(150 - nDirection, 150 + nDirection);
  }
  motors_run(0, 0);
  delay(400);
  while (!digitalRead(Button)) {}
  delay(400);
}
```

Представлена вище основна функція програми виконується періодично (в даному випадку раз в 400 мс). Поки кнопка Button не натиснута, працює алгоритм пропорційного регулятора: двигуни отримують значення потужності, пропорційне відхиленню від чорної лінії. Число 25 – показання датчика освітлення на границі чорної лінії з білим полем. Число 3 – емпірично визначний коефіцієнт пропорційності.

Нижче наведено функцію визначення поточної яскравості відбитого світла з врахуванням зовнішнього освітлення.

```
int reflectLight(void) {
  digitalWrite(LED1, HIGH);
  delay(5);
  int light = analogRead(LSens);
  digitalWrite(LED1, LOW);
  delay(5);
  int light2 = analogRead(LSens);
  int midlight = light - light2;
  return midlight;
}
```

Також наведемо функцію керування керування швидкістю і напрямком обертання двигунів, яка забезпечує поворот.

```
void motors_run(int a, int b){
  analogWrite(E1, a);
  digitalWrite(M1, HIGH);
  analogWrite(E2, b);
  digitalWrite(M2, HIGH);
}
```

Особливість вищеописаного алгоритму та програми на його основі в тому, що він є незалежний від зовнішнього освітлення. Також алгоритм є структурованим у декілька взаємозалежних функцій, що дозволяє легко переносити його і пере використовувати у інших проектах.

Наукова новизна. Основною новизною даної роботи є запропонований алгоритм пропорційного регулятора, який забезпечує точний рух робота вздовж лінії незалежно від умов зовнішнього освітлення. Структурована програна реалізація дозволяє з легкістю інтегрувати ii в інші проекти пов'язані з робототехнікою.

Висновки. В результаті було представлено реалізацію руху робота вздовж кольорової лінії за пропорційним алгоритмом з використанням плати Romeo BLE. Описано апаратну складову та алгоритм, що враховує зовнішнє освітлення.

ПЕРЕЛІК ПОСИЛАНЬ

1. Клименко А.В., Анісімов В.В., Анісімов В.М. Віртуалізовані інформаційні системи керування виконавчими механізмами // Матеріали доповідей ii міжнародної науково-практичної конференції «Логістика і транспортна безпека: проблеми та перспективи розвитку в контексті аналізу сучасних викликів і загроз» - Дніпро: Середняк Т.К., С. 23-27.
2. Клименко А.В., Анісімов В.В., Анісімов В.М. Дистанційні системи керування в машинобудуванні на базі робототехнічних комплексів з можливістю віддаленого спостереження та корекції роботи у реальному часі // Матеріали міжнародної науково-технічної конференції «Нові иа нетрадиційні технології в ресурсо- та енергозбереженні» - Одеса, Україна. - 2023 р., - С. 121-122.

3. Підключення світлодіода RGB до arduino. [Електронний ресурс]. URL: http://robotechnics18.rf/connection-rgb-LED-to-arduino/
4. Documentation for app developers [Електронний ресурс]. – Режим доступу: https://developer.android.com/docs.
5. Фріцінг. [Електронний ресурс]. URL: http://fritzing.org/home/

УДК 519.6:534.2:004.94

А.Л. Ширін[1], Р.Є. Слободнюк[2], С.О. Хричов[2]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна
[2]Дніпровський технолого-економічний фаховий коледж, Дніпро, Україна

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ЯК КЛЮЧ ДО ІННОВАЦІЙНОЇ STEAM-ОСВІТИ ЗА СПЕЦІАЛЬНІСТЮ 181 ХАРЧОВІ ТЕХНОЛОГІЇ

Анотація. Описано важливість використання сучасних інформаційних технологій в освітньому процесі. Надано приклади використання програмного забезпечення для підготовки здобувачив за спеціальністю 181 Харчові технології галузі знань 18 Виробництво та технології.

Ключові слова: *інформаційні технології, харчові технології, хімічні процеси, ChemSketch, моделювання, програмне забезпечення.*

Вступ. Сучасний освітній процес у галузі знань 18 Виробництво та технології неможливий без використання інформаційних технологій [1]. Інноваційні засоби навчання сприяють більш ефективному засвоєнню матеріалу, розвитку професійних компетенцій і підвищенню якості освіти [2]. Одним із потужних інструментів у цій галузі є програмне забезпечення ACD/ChemSketch, яке дозволяє студентам оволодівати складними поняттями та моделювати хімічні процеси [6].

Інформаційні технології дозволяють створювати інтерактивні навчальні матеріали, доступ до яких можливий із будь-якої точки світу [4]. Особливо актуальним такий метод є для студентів галузі харчових технологій, де суттєва увага приділяється вивченню хімічного складу продукту, розробці нових рецептур і контролю якості харчової продукції [3].

Основний зміст роботи. В рамках навчального процесу студенти можуть виконувати завдання з моделювання хімічних реакцій, створення молекулярної структури харчових компонентів та оцінювання їх стабільності [6]. Наприклад, ACD/ChemSketch – це потужна програма для створення хімічних структур і моделювання хімічних процесів. У галузі харчових технологій вона має ключове застосування [5]. Програма дозволяє студентам створювати структурні формули харчових добавок, жирів, білків і вуглеводів, що є основним для розуміння їх властивостей і взаємодій [6]. ACD/ChemSketch допоможе візуалізувати реакції,

які відбуваються під час обробки харчових продуктів, таких як ферментація, окиснення або карамелізація. Завдяки програмі студенти можуть досліджувати фізико-хімічні характеристики молекул, що мають значення для створення безпечних і корисних продуктів [4]. Програма широкого використання для створення презентацій, публікацій і дипломних робіт, що включають хімічні формули і реакції, які можуть здобувачам взаємодії з навчальним матеріалом, що сприяє глибшому засвоєнню знань [6]. На основі цього інформаційні технології забезпечують зручність і доступність, електронним ресурсам, включаючи бази даних і симулятори, доступні в будь-який час, що робить навчальний процес більш гнучким [3].

Avogadro – це відкрите програмне забезпечення для тривимірного моделювання молекул, яке розширюється через архітектуру плагінів та широко використовується для вивчення органічної та неорганічної хімії. Він подається як редактор і візуалізатор молекул, призначений для крос-платформного використання в обчислювальній хімії, молекулярному моделюванні, біоінформатиці, матеріалознавстві та суміжних областях. Авогадро надає студентам можливість моделювати взаємодію молекул, що має значення для розуміння процесів емульгування, стабілізації розчинів та інших явищ, характерних для харчової промисловості [2].

Virtual Chemistry Lab – віртуальна хімічна лабораторія, яка дає можливість проводити експерименти в інтерактивному середовищі. Вона призначена для того, щоб дозволити студентам зрозуміти навчальну програму та забезпечити рівні можливості щодо дослідження хімічних реакцій. Virtual Chemistry Lab дозволяє експериментувати в електронному середовищі і набувати досвіду навчання на практиці, деякі експерименти дають можливість робити помилки, таким чином дозволяючи ставити запитання і використовувати метод проб і помилок. Лабораторія має 2 різні секції. Є частина, де проходять більш традиційні хімічні процеси та експерименти, які ми називаємо "point and click" (навести і клацнути). Є розділ, який включає в себе "фізичні взаємодії", пропонуючи більш реалістичний і близький до віртуальної реальності досвід. Тут ви можете налаштувати експериментальні набори та виконати експерименти. Оскільки експерименти працюють з математичними формулами, вони дають результати, наближені до реальності в цьому відношенні. Для студентів харчових технологій це гарна можливість безпечно досліджувати вплив різних хімічних факторів на харчові продукти, таких як pH, температура чи концентрація реагентів, без потреби в реальних реактивах чи лабораторному устаткуванні [5].

Molecular Workbench – програма орієнтована на створення інтерактивних симуляцій хімічних і фізичних процесів, яка дозволяє студентам моделювати динамічні процеси, наприклад, термодинамічні зміни або розподіл речовин у сумішах, які стосуються харчової промисловості [4]. Типовий модуль Molecular Workbench – це комплексний навчальний пакет, що складається з ряду структурованих сторінок, які містять текст, симуляції, інструменти, елементи

керування, графіки, навігаційні посилання та вбудовані оцінки. Користувальницькі інтерфейси моделювання в Molecular Workbench можна налаштувати для студентів різних курсів. Ця унікальна функція дозволяє підтримувати широкий спектр навчальних стратегій, таких як навчання на основі запитів, відкриттів та проблемного навчання.

Висновки. Завдяки використанню таких програм, як ACD/ChemSketch і Avogadro, студенти можуть детально аналізувати хімічний склад харчових продуктів, розуміти механізми їх взаємодії та прогнозувати зміни, які відбуваються під час термічної або механічної обробки. Програми типу Virtual Chemistry Lab та Molecular Workbench можуть імітувати процеси, які мають місце в харчовому виробництві, наприклад, пастеризацію, ферментацію або стабілізацію емульсії, яка допомагає студентам оволодівати практичними навичками без потреби в дорогому забезпеченні. Крім того, за допомогою використання програмного забезпечення студенти можуть створювати нові рецептури та досліджувати вплив різних компонентів на смакові й текстурні характеристики продукту, що є місцем для інноваційної діяльності в харчовій промисловості. Моделювання хімічних реакцій і експериментів у віртуальному середовищі дозволяє також уникнути ризиків, пов'язаних із використанням небезпечних речовин, і знижує витрати на навчання.

Таким чином, інформаційно-комунікаційні технології відіграють важливу роль у підготовці фахівців з харчових технологій. Використання таких програм, як ACD/ChemSketch, Avogadro, Virtual Chemistry Lab та Molecular Workbench, дозволяє інтегрувати хімічну теорію з практичними потребами харчової галузі [1, 3]. Такі інструменти допомагають студентам не тільки засвоювати знання, а й застосовувати їх у реальних виробничих умовах, що сприяє підготовці висококваліфікованих спеціалістів і розвитку галузі в цілому [2, 6].

ПЕРЕЛІК ПОСИЛАНЬ

1. Мельник О. Ф. Міжпредметні зв’язки як засіб реалізації принципу фундаменталізації в процесі формування професійної компетентності техніків─ технологів виробництва харчової продукції //Педагогічні науки: теорія, історія, інноваційні технології. – 2015. – №. 8. – С. 153-164.
2. Мельник О. Ф. Роль і місце природничих дисциплін у процесі професійної підготовки майбутніх техніків-технологів виробництва харчової продукції //Проблеми освіти: Наук-метод. зб. / Інститут інноваційних технологій і змісту освіти МОН України. Київ, 2015. – Вип. 85. – С. 140-147.
3. Мідак Л. Я., Кузишин О. В., Базюк Л. В. Методичні вказівки до самостійної роботи з курсу «Сучасні інформаційні технології (за професійним спрямуванням)». – Калуш: ФО-П Петраш К.Т., 2018. – 92 с. – 100 пр.
4. Нетрибійчук О. Використання інформаційно-комунікаційних технологій у навчанні хімії //Біологія і хімія в рідній школі. – 2018. – Т. 3. – №. 126. – С. 30-38.
5. Слободнюк Р. Є. Демонстрація хімічних явищ за допомогою інформаційно-комунікаційних технологій при дистанційній формі навчання //Матеріали Всеукраїнської науково-практичної конференції «Проблеми розвитку професійних компетентностей вчителів природничо-математичного напрямку». – 2021. – С. 181-183.

6. Слободнюк Р. Є. Педагогічні умови використання редактора хімічних формул в освітньому процесі // Матеріали Всеукраїнської науково-практичної конференції «Проблеми розвитку професійних компетентностей вчителів природничо-математичного напрямку» //КЗВО ДАНО ДОР. Дніпро, 2019. – Т. 1. – С. 20-22.

УДК 004.056:004.94

О.О. Сафаров[1], Д.Г. Масло[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РОЗРОБКА ТА ДОСЛІДЖЕННЯ ЗАСОБІВ МОНІТОРИНГУ ТА ФІЛЬТРАЦІЇ ФАЙЛОВОЇ СИСТЕМИ ОС WINDOWS

Анотація. Розглянуто особливості вдосконалення засобів захисту файлової системи ОС Windows за рахунок розробки і впровадження фільтрів файлової системи. Розроблено дві програмні реалізації фільтру, які мають однаковий функціонал, однак використовують різні технології та працюють у різні режими роботи процесору. Фільтрація полягає у перехопленні таких файлових операцій як відкриття, запис, читання та видалення файлів за заданими правилами доступу. Проведено аналіз і порівняння двох варіантів реалізації та визначено умови надання переваги тому чи іншому фільтру.

***Ключові слова**: інформаційна безпека, несанкціонований доступ, Windows, файлова система, перехоплення, фільтрація, блокування доступу, шкідливе програмне забезпечення*

Вступ. В наш час більшість комп’ютерних систем працюють на операційній системі Windows, яка користується популярністю завдяки зручному користувацькому та програмному інтерфейсу, що приваблює різні групи користувачів. Однак зі збільшенням кількості користувачів росте і кількість зловмисників, які створюють шкідливе програмне забезпечення. Однією з найпоширеніших цілей шкідливого коду є отримання несанкціонованого доступу до файлових даних. З огляду на те, що існує безліч шляхів інфікування персонального комп’ютеру подібним кодом, відповідно існує і велика ймовірність того, що рядовий користувач рано чи пізно зіткнеться з ним. Наприклад, запустивши неперевірений програмний засіб, користувач створює процес, що з точки зору операційної системи належить користувачу системи, тому жоден вбудований механізм захисту не захистить файлову систему від такої атаки. Однак фільтри файлової системи надають можливість визначити файли, спроби доступу до яких повинні контролюватися окремим засобом і назви процесів, які мають доступ до цих файлів. В результаті при спробі шкідливого програмного забезпечення отримати доступ до захищеного файлу, воно отримає помилку відмови доступу. Таким чином, дане дослідження спрямоване на вдосконалення засобів захисту важливих файлових даних від несанкціонованого доступу зі сторони шкідливого програмного забезпечення та розробку для цього програмних засобів фільтрації, їх порівняння і аналіз.

Постановка задачі. Одним з ключових завдань є аналіз загрози несанкціонованого доступу до файлів з боку шкідливого програмного забезпечення. З огляду на його результати – дослідження засобів протидії даній загрозі за рахунок фільтрації файлових операцій процесів, що виконуються у системі. Також завданням є розробка двох варіантів фільтру файлової системи, перший з яких працюватиме виключно у режимі користувача, не втручаючись у роботу ядра, а другий – у режимі ядра [1], являючи собою міні-фільтр драйвер, який перехоплює та фільтрує файлові операції на значно нижчому рівні системи. Після чого необхідно провести аналіз отриманих програмних реалізацій, визначення сильних та слабких сторін, різниці у навантаженні на систему, складності розробки та умов надання переваги тому чи іншому фільтру файлової системи.

Основний зміст роботи. Шкідливе програмного забезпечення, яке може бути завантажено і запущено користувачем у системі під виглядом звичайного додатку може приховувати у своєму коді можливості пошуку важливих файлових даних системи і спроби порушення конфіденційності, цілісності та доступності інформації користувача. У цьому випадку операційна система сприймає даний процес як цілком легітимний, адже він виконується під обліковим записом довіреного користувача і ним же запущений. Тому вбудовані механізми захисту не захистять файлові ресурси від такого роду атак.

Додатковий програмний засіб, який виконує фільтрацію на інших принципах, які базуються на більш конкретних правилах захисту дозволяють надійно захистити файли від неперевірених програмних засобів.

Принцип фільтрації файлових операцій полягає у їх перехопленні на шляху до драйверу файлової системи, який виконує фактичні дії з фізичним носієм інформації, на якому збережено файловий об’єкт. У разі виявлення операції, яка згідно правилам відбувається над захищеним файлом, але у списку дозволених програм немає назви процесу, який виконує дану операцію, виконуються відповідні дії з блокування доступу до файлу та повернення коду помилки програмному засобу.

Даний принцип є спільним для обох програмних реалізацій фільтрів файлової системи, однак виконується він по-різному за рахунок виконання у різних режимах процесору та різного рівня доступу до ресурсів системи.

Перший фільтр працює виключно у режимі користувача, що означає, що він не відноситься до компонентів ядра операційної системи і з її точки зору є звичайним програмним засобом. Однак в операційній системі Windows один процес не має можливості виконувати моніторинг та фільтрувати файлові операції іншого процесу. Це гарантується таким механізмом захисту як віртуальний адресний простір [2]. Тому необхідне втручання у захищений простір кожного процесу для встановлення функцій перехоплення («хук-функцій») на системні виклики взаємодії з файлами. Робиться це за допомогою інжекції динамічної бібліотеки (DLL) [3]. При виконанні певної послідовності дій, які включають в себе розширення віртуального адресного простору, запис

туди шляху до бібліотеки фільтрації і створення віддаленого потоку виконання функції завантаження динамічної бібліотеки, можна примусово завантажити DLL у певний процес. При під'єднанні вона виконує функцію входу, так само як звичайні програми виконують функцію main. Назва даної функції – DllMain. Саме у її тілі записуються операції, які будуть виконуватися вже всередині процесів та файлову активність необхідно фільтрувати.

Для перехоплення функцій в режимі користувача використовується так званий механізм «хуків», що дозволяє встановити функції перехоплення, які мають ті самі прототипи, що і оригінальні функції, але викликаються замість них. Завдяки цьому хук-функції мають можливість аналізувати параметри, передані кодом програмного засобу. На їх основі визначається назва файлу, до якого додаток прагне отримати доступ, та назва самого процесу. Якщо файл є захищеним, а назви процесу немає у списку довірених додатків, операція блокується за рахунок повернення коду помилки відмови доступу. Якщо ж файл не є захищеним або процес є довіреним, викликається оригінальна функція і файлова операція вважається дозволеною.

Іншим варіантом програмної реалізації є фільтр режиму ядра. Він виконується у контексті ядра операційної системи Windows, що надає йому значно ширший спектр функціональних можливостей та рівень доступу до ресурсів [4]. За рахунок того, що код режиму ядра може втручатися у будь-які процеси операційної системи, він не потребує додаткової процедури інжекції у кожен окремий процес, а аналізує усі файлові операції, що відбуваються у системі. В основі даного рішення лежить технологія міні-фільтрів операційної системи Windows. Вони дозволяють встановити функції оберненого виклику на вказані типи пакетів вводу-виводу, які передають інформацію про файлову операцію від користувацького додатку до драйверів взаємодії з накопичувачам. Міні-фільтр драйвер розташовується між менеджером вводу-виводу, який отримує запит на доступ до файлу, і драйвером файлової системи, який обробить цей запит. Таке його знаходження дозволяє перехоплювати файлові операції ще до їх обробки та виконання фактичних дій з файлом.

У функціях оберненого виклику обробки пакетів вводу-виводу виконується та сама процедура фільтрації доступу до файлу, що і у фільтрі режиму користувача. Виконується збір даних про шлях до файлу і назву процесу, який виконує над ним операцію. У разі наявності файлу у списку захищених виконується аналіз назви процесу на предмет його наявності серед назв додатків, які є довіреними для даного ресурсу. Якщо ж для захищеного файлу дане ім'я процесу відсутнє, драйвер встановлює дані, що свідчать про відмову доступу та повертає дані менеджеру вводу-виводу, який після цього повідомляє користувацький додаток про відмову доступу. Тобто далі пакет вводу-виводу не проходить і файлова операція не відбувається. В інших випадках міні-фільтр драйверу пропускає файлову операцію дані і драйвер файлової системи виконує відповідні дії над файлом.

Таким чином обидві програмні реалізації фільтру файлової системи виконують однакову процедуру захисту важливих файлів від несанкціонованого доступу (рис. 1), але роблять це на принципово різних рівнях.

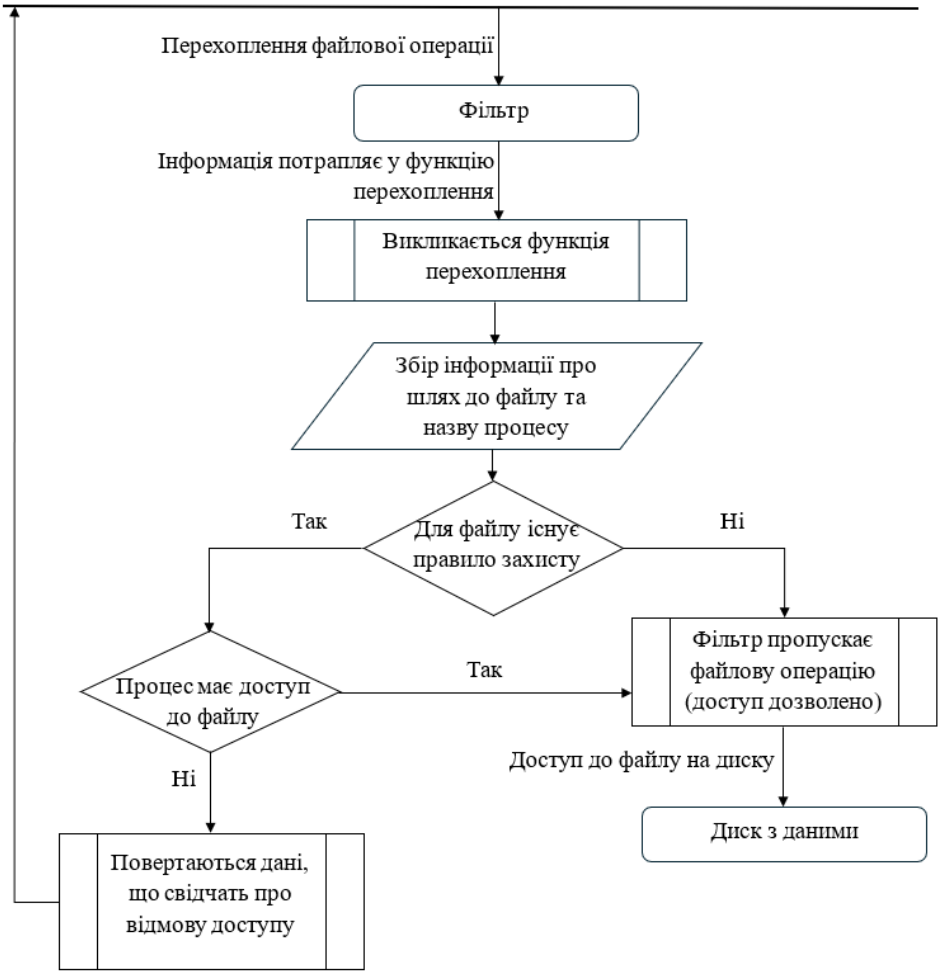


Рис. 1. Алгоритм фільтрації доступу процесу до файлу

Однією з основних переваг фільтру файлової системи режиму користувача є простота його створення. Він не вимагає написання драйверів режиму ядра, що є доволі складною задачею, не потребує процедури відлагодження драйверів, яка займає багато часу і загалом є суттєво дешевшою. Головною перевагою фільтру режиму користувача є більша надійність за рахунок виконання в привілейованому режимі та доступ до усіх ресурсів системи, за рахунок чого використовується менший процесорний час та об’єм оперативної пам’яті.

Наукова новизна розробки двох фільтрів файлової системи полягає у вдосконаленні засобів захисту файлової системи він несанкціонованого доступу з боку шкідливого програмного забезпечення та наданні альтернативних варіантів досягнення даної мети з подальшим аналізом їх плюсів та мінусів.

Висновки. У роботі проведено аналіз загрози шкідливого програмного забезпечення для конфіденційності, цілісності та доступності файлів

користувача. Запропоновано засоби вирішення даної проблеми з конкретним описом дій. Програмно реалізовано два альтернативні засоби фільтрації, які використовують різні технології, працюють у різних режимах процесору та різняться за складністю реалізації. Проаналізовані особливості їх роботи та можливості у сфері фільтрації файлових операцій. Розглянуті переваги та недоліки кожного з них та обґрунтована доцільність їх реалізації та використання.

ПЕРЕЛІК ПОСИЛАНЬ

1. Операційні системи: [Електронний ресурс]: навч. посіб. для студ. спеціальності 123 «Комп'ютерна інженерія» / В. Г. Зайцев, І. П. Дробязко. – Київ: КПІ ім. Ігоря Сікорського, 2019. – 240 с.
2. Windows 2010 : навчальний посібник / Укладач: Дячук С. Ф. – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2021. – 144 с.
3. Process Injection Part 1: The Theory [Електронний ресурс] –https://secarma.com/process-injection-part-1-the-theory/.
4. Filter Manager Concepts [Електронний ресурс] – https://learn.microsoft.com/en-us/windows-hardware/drivers/ifs/filter-manager-concepts.

УДК 004.946

Д.І. Клевченко[1], А.Т. Харь[1], Л.В. Кабак[1]

[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

МОДИ В ІГРОВІЙ ІНДУСТРІЇ: ТЕХНІЧНІ ВИКЛИКИ ТА МОЖЛИВОСТІ

Анотація. Дослідження присвячене вивченню технічних викликів та можливостей, пов'язаних зі створенням модифікацій (модів) до відеоігор. Основною метою дослідження є аналіз процесу розробки модів з точки зору програмування, використання SDK, рушіїв та бібліотек. На прикладах відомих ігор, таких як Minecraft, The Sims та S.T.A.L.K.E.R., розглядаються особливості створення модів, а також аналізуються технічні виклики, з якими стикаються розробники. У статті також досліджуються можливості, які відкриваються завдяки модам, включаючи розширення геймплею, розвиток навичок та роль спільнот моддерів. Висновки підкреслюють перспективи та майбутнє модифікацій у відеоіграх в умовах розвитку нових технологій.

Ключові слова: *модифікації відеоігор, програмування, SDK, ігрові рушії, бібліотеки, Minecraft, The Sims, S.T.A.L.K.E.R., технічні виклики, можливості модів, спільноти моддерів, розширення геймплею, розвиток навичок, нові технології в іграх, AI, VR.*

Вступ. У сучасній ігровій індустрії модифікації (моди) відіграють важливу роль, дозволяючи гравцям розширювати функціональні можливості ігор та створювати нові, унікальні враження. Від перших простих змін до повноцінних додаткових сюжетних ліній та нових світів, моди значно збільшують іграбельність ігор і сприяють формуванню активних спільнот навколо них [4].

Моддинг зародився після створення першої ж гри. У гравців виникали запитання “А якщо можна було б змінити це?”, “Було б класно, якщо замість цього було б це”. Потрібно зазначити, що не лише це вплинуло на користувачів. Багато хто ознайомився з цією галуззю через лінь деяких розробників виправити баги у своїх проектах. Тут моди також допомогають. З цього і почались експерименти, які створили нову галузь геймдеву, яка була нейтральною, адже приносила як проблеми для компаній, так і поле для розробки.

У презентації CAPCOM R&D, що стосується боротьби з шахрайством та піратством у комп'ютерних іграх, зазначено: «Гравці можуть не тільки створювати моди, але й поширювати піратські копії» [6].

Інакше кажучи, Capcom турбується про можливі фінансові втрати через піратські копії та модифікації і визнає, що точний розмір збитків невідомий. Відсутні конкретні дані, з якими можна було б порівняти ситуацію.

На перший погляд, моддинг може виглядати як шахрайство, адже гравці використовують інтелектуальну власність розробників і змінюють її на власний розсуд. Проте деякі великі розробники підтримують і заохочують таку творчість фанатів, оскільки моди можуть позитивно впливати на ігри, роблячи їх цікавішими та продовжуючи їх життєвий цикл.

Моддинг дав життя деяким з найпопулярніших та найуспішніших ігор в індустрії, таким як Counter-Strike, DotA та популярний нині жанр Battle Royale. І це лише кілька прикладів; творчість моддерів виходить за межі ігор. Яскравий приклад — Sims, що дозволяє перетворювати ігри на серіали [1].

Постановка задачі. Для досягнення поставленої мети в роботі сформовані такі завдання:

- Технічні виклики при створенні модів (мови програмування, движки, бібліотеки, інструменти та SDK);
- Можливості, що відкриваються завдяки модам;
- Аналіз популярних модів для відомих ігор (Minecraft, The Sims, S.T.A.L.K.E.R).

Основний зміст роботи. Найчастіше запитання, що виникає у тих, хто планує поринути у світ модингу це “Яку мову програмування обрати?”.

Вибір мови програмування залежить від гри, яку ви хочете модифікувати. Наприклад, якщо ви хочете створювати моди для Minecraft, вам потрібно вивчити Java, оскільки Minecraft написаний на цій мові. Однак Java — це особливий випадок, більшість ігор розробляються з використанням C/C++.

Java дозволяє модифікувати ігри, написані на Java, тому що ви можете взяти файли Java (.jar) і перетворити їх назад у вихідний код. З C або C++ так зробити не можна; найкраще, що ви можете зробити з їх виконуваними файлами, це перетворити їх у асемблерний код, який є надзвичайно складним для освоєння.

Однак розбивка гри на вихідний код є крайнім випадком. Багато ігор надають інструменти для створення модів. Наприклад, Skyrim постачається з

інструментом під назвою "Creation Kit", який можна використовувати для створення модів, а деякі розробники створюють спеціальні файли, щоб полегшити геймерам у модифікації ігор. Так зробили Том Холл та Джон Кармак, засновники Id Software яка випустила Doom. Вони випустили WAD-файл, що містить усі текстури, спрайти та дизайни карток для Doom. Це те, що пізніше будуть називати комплектом розробки ПЗ, SDK. [7]

Друга проблема, це вибір SDK, що розшифровується як "Набір засобів розробки програмного забезпечення" (Software Development Kit). Його основне призначення — спростити та полегшити процес створення додатків або ігор за допомогою різних інструментів, таких як редактори сцен, середовища для написання сценаріїв тощо. Можна розробити цілу гру за допомогою SDK, якщо він підтримує всі необхідні вам функції. Проте, вам все ще потрібно створювати власні медіа-ресурси, такі як персонажі, зброя, індикатори здоров'я, звуки тощо.

SDK часто використовуються для системи створення світу, що може бути досить складним і трудомістким процесом, якщо програмувати його самому або використовувати інші SDK, наприклад, DirectX. Unity3D, UDK та CryEngine є, напевно, лідерами у сфері простого та сучасного створення ігор на даний момент.

Однак слід знати, що якщо ви новачок у розробці ігор, ви не зможете просто завантажити SDK і відразу почати реалізовувати свої ідеї. Вам знадобиться чимало часу і ресурсів, а також знання ігрового рушія і навички в написанні сценаріїв або програмуванні.

Розглянемо використання ігрових рушіїв. Це біль та страх більшості новачків. Спочатку ти не знаєш, який рушій обрати, а потім витрачаєш купу часу на його вивчення, поки не вирішуєш, що ти досить старий для геймдеву.

Так який рушій обрати? Зараз найбільш популярні такі титани як Unreal Engine та Unity. Вибір залежить від багатьох факторів.

Unity, безумовно, широко використовується в мобільному геймдеві. Сучасний ринок розробки дозволяє вивчати будь-які інструменти, які вам більше подобаються, і завжди можна знайти роботу в цій сфері. Для новачків вибір Unity також є чудовим рішенням, оскільки цей рушій широко використовується, має великий досвід роботи та доброзичливе ком'юніті.

Якщо говорити про графіку, то якщо ви розумієте як оптимізувати шейдери, ефекти графіки, то вам із будь-яким ігровим рушієм буде комфортно. У Unreal Engine краще підтримуються технології Nvidia. Також є підтримка Nvidia GameWorks, зокрема, Nvidia Flex та інші технології. Ці особливості вже є тонкими нюансами, які залежать від контексту вашого проекту. Загалом, з точки зору рендерингу, ігрові рушії, якщо й не ідентичні, то дуже схожі. А в поєднанні з безкоштовними плагінами, кожен рушій має свої особливості в тонких деталях. Unity ж випускає багато інструментів для 2D розробок.

Unity має дві основні проблеми. По-перше, досі немає офіційної підтримки мобільних браузерів. Хоча на сучасних смартфонах технологія працює, використання її для виробничих рішень є ризикованим, оскільки вона не має офіційної підтримки. По-друге, та найголовніше — це час завантаження.

Основною перевагою веб-технологій є швиде подання контенту користувачам. Тому довгий час завантаження у Unity повністю перекреслює цю перевагу [2].

Бібліотеки грають важливу роль у розширенні функціоналу модів, надаючи розробникам готові рішення для виконання складних завдань і зменшуючи необхідність розробки функціоналу з нуля. Вони забезпечують моддерів потужними інструментами для підвищення ефективності розробки та інтеграції нових можливостей у відеоігри.

Бібліотеки забезпечують моддерів попередньо написаним кодом для виконання певних завдань, таких як графічний рендеринг, обробка даних, створення користувацького інтерфейсу та інші функціональні можливості. Це дозволяє зосередитися на креативній частині моддингу, замість того щоб витрачати час на реалізацію базових функцій. Вони можуть включати:

1. Фреймворки для рендерингу: допомагають з покращенням графіки та візуальних ефектів у модах.
2. Інструменти для обробки даних: забезпечують ефективну обробку і збереження даних, що використовуються модами.
3. Бібліотеки для розробки UI: дозволяють створювати користувацькі інтерфейси, такі як меню або інтерфейси налаштувань.

Приклади популярних бібліотек для розробки модів JDK (Java Development Kit)

- **Опис**: Набір інструментів і бібліотек для розробки програм на Java, що є основною мовою програмування для модів Minecraft.
- **Функціональність**: Містить інструменти для компіляції і запуску Java-програм, а також стандартну бібліотеку Java, яка надає основні функції, такі як обробка вводу/виводу, колекції і інше.

1. **Forge API (для Minecraft)**
   - **Опис**: Популярний фреймворк для розробки модів для Minecraft.
   - **Функціональність**: Надає багатий набір інструментів для взаємодії з ігровим двигуном Minecraft, полегшуючи інтеграцію нових елементів гри, таких як блоки, предмети, біоми і багато іншого.
2. **ModLoader (для Minecraft)**
   - **Опис**: Інструмент для модифікації Minecraft, який дозволяє легко завантажувати і керувати модами.
   - **Функціональність**: Спрощує процес модифікації гри, надаючи API для взаємодії з грою та завантаження модифікацій.
3. **OpenVR**
   - **Опис**: Бібліотека для розробки віртуальної реальності, підтримує різні VR-гарнітури.
   - **Функціональність**: Дозволяє інтегрувати функції VR у моди, надаючи можливості для створення ін immersive досвіду.
4. **Lwjgl (Lightweight Java Game Library)**
   - **Опис**: Бібліотека для створення графічних ігор на Java.

- **Функціональність**: Надає доступ до функцій OpenGL, OpenAL і інших технологій, що використовуються для графічного рендерингу та аудіо в іграх.

5. **GSON (для Java)**
   - **Опис**: Бібліотека для перетворення Java-об'єктів у JSON і навпаки.
   - **Функціональність**: Зручна для обробки даних, що зберігаються у форматі JSON, що є популярним у багатьох ігрових проектах.
6. **SDL (Simple DirectMedia Layer)**
   - **Опис**: Кросплатформена бібліотека для управління графікою, введенням і звуком.
   - **Функціональність**: Часто використовується для створення кросплатформених ігор та модів, забезпечуючи доступ до основних функцій мультимедіа.
7. **LibGDX**
   - **Опис**: Кросплатформений фреймворк для розробки ігор на Java.
   - **Функціональність**: Забезпечує зручний API для роботи з графікою, звуком і фізикою, а також можливість розробки для різних платформ, таких як Windows, Linux, Android і iOS.

Хоча бібліотеки суттєво полегшують розробку модів, вони також вносять певні труднощі та проблеми. Ретельне управління версіями, вивчення документації, моніторинг продуктивності та безпеки, а також правильне розуміння ліцензійних умов можуть допомогти ефективно усунути ці виклики та забезпечити успішну розробку модів.

Використання бібліотек для розробки модів має багато переваг, але також супроводжується рядом проблем, які можуть ускладнити процес розробки. Однією з основних труднощів є сумісність версій. Оновлення бібліотек може призвести до конфліктів з різними версіями ігрових рушіїв або іншими бібліотеками, що може вимагати значних зусиль для вирішення. Наприклад, нова версія бібліотеки може бути несумісною з попередніми версіями гри або з іншими бібліотеками, що використовуються, що може затримати процес розробки і вимагати переписування частин коду або додаткових налаштувань.

Якість документації бібліотек також є критично важливою для їх ефективного використання. Неповна або застаріла документація може ускладнити розуміння API бібліотеки та її можливостей. Відсутність активної підтримки з боку розробників бібліотеки може залишити моддерів без необхідних відповідей і рішень для вирішення проблем, що виникають у процесі розробки.

Процес навчання та освоєння нових бібліотек може зайняти значний час і зусилля, особливо якщо бібліотека має складний API або специфічні вимоги. Моддери часто витрачають багато часу на вивчення та адаптацію до нових інструментів, що може затримати розробку модів і зменшити продуктивність.

Бібліотеки зазвичай мають власні залежності, які потрібно інсталювати і підтримувати. Управління цими залежностями може бути складним, особливо

якщо деякі з них оновлюються або мають конфлікти з іншими частинами проекту. Неправильне управління залежностями може призвести до помилок у коді та ускладнити розробку модів.

Використання бібліотек може також вплинути на продуктивність модів. Якщо бібліотеки містять неефективний код або надмірні функціональні можливості, це може зменшити загальну швидкість ігрового процесу та вплинути на досвід користувачів. Тому важливо перевіряти продуктивність модів після інтеграції бібліотек, щоб уникнути негативних наслідків.

Безпека є ще одним важливим аспектом при використанні сторонніх бібліотек. Бібліотеки, які містять уразливості або не підтримуються, можуть створити ризики для безпеки даних користувачів або системи. Моддери повинні бути уважними до безпеки бібліотек і регулярно перевіряти їх на наявність відомих вразливостей.

Ліцензійні умови бібліотек також можуть створювати проблеми. Різні бібліотеки мають різні ліцензійні умови, які можуть обмежувати їх використання або вимагати виконання певних умов. Неправильне розуміння або порушення цих умов може призвести до правових проблем, тому важливо ретельно перевіряти ліцензійні умови кожної бібліотеки і забезпечити відповідність проекту цим умовам.

Останньою проблемою є підтримка різних платформ. Бібліотеки можуть підтримувати кілька платформ, але це може створити проблеми при розробці для різних систем. Платформенна специфікація бібліотек може ускладнити тестування і забезпечити правильну роботу модів на різних операційних системах, таких як Windows, macOS або Linux.

Аналіз популярних модів для відомих ігор.

Щоб почати створювати моди для Minecraft, важливо вивчити основи програмування на Java. Існує безліч безкоштовних онлайн-уроків, які допоможуть вам у цьому. Важливо знати, що версії Minecraft 1.12 та 1.16 використовують Java 8, тоді як Minecraft 1.17 та більш нові версії починають використовувати Java 16. Основи програмування залишаються схожими, тому навіть старі уроки можуть бути корисними.

Далі необхідно ознайомитися з інструкціями, специфічними для вашого мод-лоадера. На сьогоднішній день найпопулярнішими мод-лоадерами є Forge та Fabric.

Щодо версій, наразі актуальною є версія 1.16. Версія 1.12 давно залишена розробниками модів, а версія 1.17 існує, проте не має великої популярності серед розробників модів, за винятком портингу для версії 1.18, яка запланована до виходу до кінця року. На момент, коли ви будете готові створити свій мод, ситуація може змінитися [7].

Підтримка модів для версії 1.16 скоро закінчиться, але гравці можуть ще використовувати її протягом року чи більше, поки нові версії не стануть достатньо розвиненими для загального використання. Випуск мода для версії 1.16 залишає вам небагато часу для розробки та виправлення помилок. Ваші

варіанти будуть або розчарувати людей, залишивши мод без підтримки, або повернутися і працювати над застарілою версією, що відволікає час і зусилля від розробки для майбутніх версій. Використання модів і розробка модпаків часто відокремлені від роботи розробника модів. Гравці часто продовжують грати в старі версії, а автори модпаків віддають перевагу трохи старішим версіям через велику кількість стабільних і відомих модів.

Розробка модів рухається швидко, тому підтримка кількох версій може бути надзвичайно складною. Коли моди починають оновлюватися, старі версії ставляться на "життєву підтримку" (виправлення помилок за необхідності і можливі вже розроблені, але не випущені функції), а потім повністю покидаються, коли нова версія стає офіційно випущеною основною версією. Моди, які мене цікавлять, або вже оновлені, або в процесі оновлення, або виглядають забутими. Я б вважав версію 1.12 застарілою.

Fabric вже не має обмежену кількість модів, і зараз обидва мод-лоадери, Fabric і Forge, мають рівну кількість модів для нових версій.

Деякі обирають Fabric, але перед початком розробки варто обов'язково вивчити основи Java. Багато речей, які потрібно буде дізнатися, вимагають принаймні базових знань цієї мови. Існує безліч уроків на YouTube, тому рекомендую ознайомитися з ними. Розробка модів — це досить серйозна справа. Якщо ви новачок у моддингу, як і я, я б порекомендував звернути увагу на MCreator. Він значно спрощує процес кодування і робить його більш інтуїтивно зрозумілим, але має обмежену функціональність. Наприклад, створення моба, який може і ходити, і літати, є досить складним і вимагає створення кількох мобів.

На даний момент MCreator підтримує лише Forge, хоча популярні моди, такі як Kimetsu No Yaiba, були створені за допомогою цього інструменту. Найбільша проблема з MCreator полягає в тому, що нещодавно було припинено офіційну підтримку версій Minecraft до 1.19.4.

Створення модів для серії ігор **The Sims** включає в себе специфічні особливості, які відрізняють цей процес від моддингу для інших ігор. Зокрема, розробка модів для **The Sims** вимагає використання власних механізмів гри для управління контентом та інтеграції нових функцій.

Однією з основних мов програмування, яка використовується для створення модів для **The Sims 3** і **The Sims 4**, є **Python**. Python дозволяє розробникам писати скрипти для реалізації нових функцій і взаємодій у грі. За допомогою Python можна змінювати поведінку персонажів, додавати нові сценарії та вдосконалювати геймплей.

Ще однією важливою мовою для модифікацій є **XML** (Extensible Markup Language). XML використовується для налаштування характеристик ігрових об'єктів, соціальних взаємодій та інших елементів. Моддери створюють та редагують XML-файли, щоб визначити параметри нових об'єктів та функцій, що додаються до гри. Наприклад, XML-файли можуть описувати нові стосунки, соціальні взаємодії або модифікації ігрового процесу.

Серед інструментів, які допомагають у створенні модів для **The Sims 4**, особливу роль відіграє **Sims 4 Studio**. Цей інструмент надає простий інтерфейс для роботи з різними типами контенту, такими як одяг, меблі, персонажі та сценарії. За допомогою Sims 4 Studio можна імплементувати нові текстури, моделі та інші елементи гри, що робить цей процес значно легшим і зручнішим.

TS4 Tools також є важливими інструментами для моддерів, які дозволяють редагувати текстури, створювати нові об'єкти та виконувати інші завдання. Крім того, для створення нових 3D-моделей, таких як меблі або одяг, моддери часто використовують **Blender**. Це потужний інструмент для моделювання і анімації, який дозволяє створювати детальні моделі для гри. Для редагування текстур та створення нових шкір або малюнків моддери можуть використовувати графічні редактори, такі як **GIMP** або **Photoshop**.

Створення модів для **The Sims** має свої особливості, такі як кастомізація персонажів, додавання нових об'єктів і меблів, а також розширення соціальних взаємодій і сценаріїв. Моддери можуть додавати нові опції для кастомізації персонажів, такі як нові стрижки, одяг і аксесуари. Також вони можуть розширювати можливості гри, додаючи нові предмети меблів і декор для оформлення будинків і ігрових просторів.

Окрім цього, моди можуть додавати нові соціальні взаємодії і сценарії, які змінюють ігровий процес, включаючи нові можливості для персонажів або нові кар'єри. Розробники модів можуть також працювати над виправленням помилок або покращенням існуючих функцій гри, щоб оптимізувати ігровий процес і змінити баланс гри.

Таким чином, створення модів для **The Sims** є складним і творчим процесом, який вимагає знання Python і XML, а також уміння працювати з такими інструментами, як Sims 4 Studio, Blender і Photoshop. Цей процес дозволяє моддерам не лише додавати новий контент до гри, але й змінювати і розширювати її функціональність, створюючи унікальний ігровий досвід для гравців.

Створення модів для **Skyrim** є складним, але захоплюючим процесом, що вимагає знання спеціалізованих інструментів і технік. Одним з основних інструментів для моддингу в **Skyrim** є **Creation Kit** (CK), який надає всі необхідні можливості для редагування ігрового контенту, створення нових квестів і налаштування взаємодії з персонажами [4].

Перш ніж розпочати моддинг, важливо освоїти основи роботи з **CK**. Існують різноманітні навчальні матеріали та вики, які допоможуть з основами створення модів, особливо для роботи з послідовниками та квестами. Ці ресурси можуть бути дуже корисними, коли йдеться про дрібні деталі, які легко забути навіть після тривалого використання CK.

При роботі з скриптами може виникнути проблема з тим, що CK має невелику помилку, пов'язану з шляхами до скриптів з основної гри. Якщо ви стикаєтеся з труднощами при редагуванні або компіляції скриптів, можливо, вам доведеться розпакувати файл Scripts.zip, який знаходиться у папці даних, і

скопіювати або перемістити вміст з Data/Source/Scripts у Data/Scripts/Source. Знання **C#** буде корисним, оскільки дозволить вам писати патчі для синтезу.

Крім того, рекомендується постійно мати відкритим ресурс **ck.uesp.net**. Цей сайт містить корисну інформацію, зокрема сторінки про функції умов та список функцій Papyrus, що є важливими для роботи з модами. Ознайомлення з **Papyrus Primer** також допоможе розширити ваші знання про скрипти і можливості модифікацій [1].

Щоб ефективно організувати вашу гру, важливо встановити **LOOT** і розуміти, як працює overwrite, а також створити вихідні папки для інструментів. Це дозволяє вам організувати моди в різні категорії, такі як «перформанс», «SKSE», «текстури», «тіла» та інші, що допоможе підтримувати порядок у вашій модифікаційній бібліотеці [5].

Процес створення списку модів починається з основних або базових модів, які є виправленнями помилок і покращеннями продуктивності. Рекомендується перевіряти категорії модів на Nexus, починаючи з найбільш завантажених та нових модів. Найзручніше починати з модів, які не викликають конфліктів і просто доповнюють гру, особливо модів для SKSE.

Встановлюючи моди, слід дотримуватися принципу: встановлювати один мод за раз. Після встановлення кожного мода перевіряйте наявність конфліктів і вирішуйте їх перед переходом до наступного мода. Кожні 10-20 плагінів рекомендується запустити **LOOT** для перевірки конфліктів і оптимізації порядку модів. Після встановлення всіх модів корисно запустити **Cathedral Optimizer** для старих модів і перевірити наявність патчів та інструкцій установки [5].

Для подальшої оптимізації використовуйте **SkyPatcher**, **SPID**, **KID**, **OAR/DAR**, **SKSE** версії та **ESL** плагіни. Ці інструменти допоможуть вам мати більше функцій, кращу оптимізацію та покращити ігровий досвід.

Крім того, **Wabbajack** може бути корисним для завантаження основних моддингових пакетів, таких як «Skyrim Modding Essentials». Ознайомтеся з профілями в **MO2** (Mod Organizer 2) і створюйте різні профілі для тестування. Читайте описи модів, перевіряйте вимоги, сумісність та інструкції з установки на сторінках Nexus.

Важливо пам'ятати, що порядок модів у лівій панелі MO2 не є тим самим, що порядок модів ESP у правій панелі. Зазвичай порядок у правій панелі вказується на багатьох сторінках модів. Для переопрацювання NPC слід дотримуватися однакового порядку з обох боків [2].

Ось кілька порад щодо організації модів: спочатку встановлюйте основні моди та виправлення помилок, потім аудіо моди, моди для квестів, заміни сіток, покращення оточення та погоди, моди для NPC, зміни в містах і ландшафтах, модифікації інтер'єрів, зміни тварин і створінь, оновлення текстур, зміни ігрових механік, оновлення AI, броню, одяг і зброю, а також будь-які моди для покращення занурення в гру. В кінці додайте зміни UI та будь-які виходи MO2.

Таким чином, моддинг для **Skyrim** вимагає детального планування, організації та знання специфічних інструментів та технік, що дозволяє створити стабільний і приємний ігровий досвід.

Наукова новизна. В роботі набули подальшого розвитку методи розробки модів з використанням різноманітних бібліотек. Та досліджені переваги та недоліки створення модів з їх використанням. Але не потрібно думати, що моди для комп'ютерних ігор не будуть мати попит. Продовжує існувати запитання щодо інтелектуальної власності, адже навіть у наш час деякі розробники модів продовжують нею нехтувати.

Відсутність монополії в моддингу – це, безумовно, добре, але в цьому конкретному випадку це створює проблеми у технічному плані. Розробникам модів потрібно витрачати багато часу на адаптацію для всіх завантажувачів або ж створювати моди лише для одного завантажувача, що призводить до ексклюзивності модів для конкретного завантажувача і фактично створює монополію на цей мод [3].

В результаті менше людей виграють через таку кількість завантажувачів, а також потрібно враховувати різні версії гри, що ще більше розподіляє моди (з цією проблемою здебільшого зіштовхнулись розробники модів Minecraft). Це ускладнює життя як користувачам модів, так і їхнім творцям.

Щоразу додаються нові інструменти, елементи та механіки у рушіях ігор.

Висновки. Моди у відеоіграх, відзначені своєю універсальністю та впливовістю, стали важливим компонентом сучасної ігрової індустрії. Через їхню здатність змінювати та вдосконалювати ігровий досвід, модифікації відкривають нові горизонти як для розробників, так і для гравців.

Технічні виклики, з якими стикаються моддери, є суттєвими, але подолання цих перешкод дозволяє реалізувати надзвичайно різноманітні проекти. Розробка модів вимагає знання специфічних мов програмування, таких як Java, Python і C++, а також вміння користуватися інструментами та SDK, такими як Forge, The Sims Mod Constructor. Робота з різними ігровими рушіями, такими як Unreal Engine і Unity, додає ще один рівень складності, але й розширює можливості для створення новаторських модів. Важливість бібліотек і SDK у цьому процесі не можна переоцінити, оскільки вони надають важливі інструменти для розширення функціоналу ігрових модифікацій.

Моди здатні не лише розширювати ігровий процес, надаючи нові можливості для гравців, але й служать потужним інструментом для навчання і розвитку професійних навичок у геймдеві. Вони стимулюють інновації та допомагають будувати кар'єру в індустрії, створюючи умови для розвитку спільнот та колаборацій. З огляду на майбутнє, модифікації у відеоіграх продовжать еволюціонувати під впливом нових технологій, таких як штучний інтелект і віртуальна реальність. Це може привести до нових тенденцій і можливостей, що ще більше розширять потенціал моддингу. Моди не тільки підтримують ігрову індустрію, а й формують її майбутнє, визначаючи нові стандарті і відкриваючи нові шляхи для креативності та інновацій.

ПЕРЕЛІК ПОСИЛАНЬ

1. Arthmoor. Simple Papyrus Script Failing to Compile. AFK Mods. [Електронний ресурс]. – Режим доступу: https://www.afkmods.com/index.php?/topic/5617-simple-papyrus-script-failing-to-compile/.
2. Bethesda. Build, share and find community-made content in Skyrim Special Edition. [Електронний ресурс]. – Режим доступу: https://bethesda.net/en/article/52xMsb1fD2nTiNBkiWCbxq/build-share-and-find-creations-skyrim-special-edition.
3. Cloud. [Електронний ресурс]. – Режим доступу: https://cloudloader.org/.
4. GitHub. Skyrim mods and SDK. [Електронний ресурс]. – Режим доступу: https://github.com/topics/skyrim.
5. LOOT. Load Order Optimization Tool at Modding Tools - Nexus Mods. [Електронний ресурс]. – Режим доступу: https://www.nexusmods.com/skyrimspecialedition/mods/22773.
6. Mod Panic. Capcom Expresses Concern Over Mods That Are 'Offensive to Public Order and Morals' [Електронний ресурс] // IGN Nordic. – Режим доступу: https://nordic.ign.com/resident-evil-2-1/74928/news/capcom-expresses-concern-over-mods-that-are-offensive-to-public-order-and-morals.
7. Steam Store. [Електронний ресурс]. – Режим доступу: https://bethesda.net/en/article/52xMsb1fD2nTiNBkiWCbxq/build-share-and-find-creations-skyrim-special-edition.

УДК 519.6:534.2:004.94

В.В. Спірінцев[1], О.В. Іванченко[1], О.В. Голінько[1], І.С. Чернявський[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ АВТОМАТИЧНОЇ ПЕРЕВІРКИ ЗАЯВОК З ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ У РОЗРОБЦІ TELEGRAM-ЧАТБОТУ ВОЛОНТЕРСЬКОЇ ПІДТРИМКИ

Анотація. Робота присвячена розробці Telegram чат-бота, який застосовує методи машинного навчання для автоматичної перевірки заявок на допомогу. Досліджено використання моделей машинного навчання для верифікації заявок, що містять текстову та візуальну інформацію. Метою роботи є вивчення потенціалу машинного навчання для забезпечення точності та ефективності перевірки заявок у великих обсягах. Для реалізації використано передові технології машинного навчання, що дозволяють автоматизувати процес перевірки запитів.

Ключові слова: *чатбот, Telegram, машинне навчання, автоматична перевірка заявок, штучний інтелект, розпізнавання документів, інноваційні методи Python, NLP, OpenCV .*

Вступ. В сучасних умовах компанії часто стикаються з необхідністю обробляти велику кількість заявок, що містять текстову та візуальну інформацію. Перевірка таких заявок вручну є складним та трудомістким завданням, що вимагає впровадження інноваційних технологій для автоматизації процесу.

Використання машинного навчання дозволяє значно спростити верифікацію заявок, знижуючи людський фактор та підвищуючи точність перевірки.

Застосування алгоритмів машинного навчання в Telegram чат-боті дозволяє автоматизувати процес перевірки заявок. Чат-бот здатний обробляти текстові запити та документи у вигляді зображень, здійснюючи їх первинну верифікацію. Тому актуальним є створення системи, яка автоматизує процес перевірки, зберігаючи високий рівень точності та мінімізуючи участь людини. Метою дослідження є розробка Telegram чат-бота для автоматичної перевірки заявок на допомогу за допомогою методів машинного навчання, що дозволить оцінити ефективність цих підходів у реальних умовах.

Постановка задачі. Дослідження ефективності автоматичної верифікації заявок на допомогу в Telegram-чатботі із застосуванням методів машинного навчання охоплює аналіз текстових та візуальних даних, поданих користувачами. Основні виклики включають створення алгоритмів для обробки природної мови (NLP), які дозволяють ідентифікувати критичну інформацію, таку як тип запиту, його терміновість, автентичність і відповідність заданим критеріям.

Додатково досліджується автоматизація аналізу зображень, що надходять у вигляді підтверджувальних документів. Це вимагає впровадження технологій комп'ютерного зору (Computer Vision) та оптичного розпізнавання символів (OCR), які здатні ефективно працювати із зображеннями різної якості та формату. Особливу увагу приділено виявленню можливих шахрайських дій, таких як підробка документів або маніпуляції з їх вмістом.

Інтеграція системи з базою даних SQLite дозволяє забезпечити зберігання й аналіз структурованих даних, включаючи текстові заявки, зображення та історію змін статусів заявок. Такий підхід сприяє створенню масштабованої архітектури, здатної працювати з великим обсягом даних у реальному часі.

Ще одним важливим аспектом є забезпечення продуктивності Telegram-чатбота при обробці паралельних запитів від багатьох користувачів. Для цього використовуються оптимізовані алгоритми та бібліотеки Python, такі як TensorFlow і PyTorch, які дозволяють ефективно реалізовувати моделі машинного навчання. Тому актуальним є створення системи, яка одночасно забезпечує автоматизовану обробку текстових та візуальних даних, аналіз її продуктивності та надійності, а також формулювання рекомендацій щодо подальшого вдосконалення процесу автоматичної верифікації заявок. Це сприятиме не лише підвищенню ефективності волонтерської діяльності, а й забезпеченню кращого користувацького досвіду.

Основний зміст роботи. Для розробки чат-бота було використано мову програмування Python, яка відома своєю простотою, широким набором бібліотек і потужними інструментами для реалізації задач у сфері машинного навчання. Python дозволяє швидко інтегрувати інструменти для аналізу тексту, зображень та створення моделей машинного навчання, що робить її оптимальним вибором для реалізації нашого проекту.

База даних, яка використовується у чат-боті, побудована на основі SQLite. Це простий і надійний варіант, що дозволяє ефективно працювати з локальними даними без складних налаштувань. SQLite ідеально підходить для зберігання заявок, статусів, документів і даних користувачів у контексті розробки невеликого, автономного проекту.

Основний акцент у нашій роботі зроблено на використанні технологій машинного навчання для автоматизації перевірки заявок на допомогу. У системі використано кілька ключових бібліотек і моделей, кожна з яких виконує конкретні завдання.

Для аналізу текстових заявок коли користувач надає запит на допомогу у текстовій формі, використовуються бібліотеки spaCy та NLTK, які забезпечують функції для аналізу природної мови (Natural Language Processing). Ці інструменти призначені для:

- попередньої обробки тексту: токенізації, очищення від зайвих символів та стоп-слів;
- ідентифікації ключових слів та семантичного аналізу: визначення намірів і категорії заявки;
- класифікації тексту за категоріями (наприклад, гуманітарна чи фінансова допомога).

Також використовується модель BERT (Bidirectional Encoder Representations from Transformers) для високоточних текстових класифікацій. BERT аналізує текст, враховуючи контекст кожного слова, що дозволяє навіть коротким або неструктурованим заявкам бути правильно класифікованими.

Коли користувач надає документи у вигляді фото або сканів, використовується бібліотека OpenCV для попередньої обробки зображень такої як покращення якості зображень: вирівнювання, корекція освітленості, видалення шумів, підготовка зображення для подальшого аналізу тексту. Для розпізнавання тексту використовується Tesseract OCR, а для обробки складних зображень нейронні мережі на основі CNN (Convolutional Neural Networks). CNN дозволяє з високою точністю витягувати текстову інформацію навіть із низькоякісних чи пошкоджених документів.

Щоб підтвердити, що користувач є реальною особою він повинен надати відео свого обличчя зазвичай до 10 секунд і для порівняння фотографії з особою на відео застосовуються такі етапи:

- Обробка фотографії паспорта, коли користувач завантажує фото паспорта або документа, на якому є його фотографія, використовується бібліотека OpenCV для попередньої обробки зображення корекція освітленості та контрасту, а також виділення області з фото за допомогою алгоритмів комп'ютерного зору, що дозволяють виділити область, яка містить саме фото користувача.
- Обробка відеопідтвердження, коли користувач надає коротке відео, на якому він має продемонструвати своє обличчя.

Для обробки цього відео застосовуються наступні етапи:

1. Виділення кадрів з відео: за допомогою бібліотеки OpenCV відео розбивається на окремі кадри (наприклад, один кадр на секунду) для аналізу.

2. Виявлення обличчя на відео: використовується алгоритм Haar Cascades або MTCNN (Multi-task Cascaded Convolutional Networks), які дозволяють визначити положення обличчя на кадрі навіть при змінному освітленні та куті зйомки.

Після того, як кадри відео оброблені, необхідно порівняти обличчя на фото з тим, що зафіксоване на відео. Для цього використовуються технології розпізнавання обличчя і однією з найбільш поширених моделей є FaceNet, яка генерує векторні представлення обличчя і дозволяє порівняти, наскільки схожі ці вектори між собою. Між фото користувача з паспорта і його обличчям на відео проводиться порівняння за допомогою косинусної подібності між векторами обличчя. Якщо подібність перевищує певний поріг, заявка вважається підтвердженою.

Для реалізації автоматичної перевірки автентичності текстових заявок та розпізнавання тексту з документів, що додаються до заявок застосовуються фреймворки TensorFlow та PyTorch:

- TensorFlow використовується для створення моделей на основі глибоких нейронних мереж, що дозволяє масштабувати рішення і працювати з великими даними.
- PyTorch забезпечує гнучкість в побудові експериментальних моделей, дозволяючи швидко вносити зміни і тестувати нові ідеї.

У Telegram чатботі використано передтреновані моделі

- Модель BERT – використовується для класифікації текстових заявок. Завдяки попередньому тренуванню на великих наборах даних вона здатна точно аналізувати текст, враховуючи контекст кожного слова. У нашому випадку модель використовувалась для автоматичного визначення типу заявки (гуманітарна, фінансова) та оцінки її коректності.
- Tesseract OCR – це інструмент для оптичного розпізнавання тексту, який також має передтреновані моделі для різних мов. У нашому чат-боті застосовувалася україномовна модель, яка дозволяє точно витягувати текст із фото-документів навіть за наявності шуму або низької якості зображення.
- Convolutional Neural Networks (CNN). Мережі CNN, зокрема моделі VGG16 та ResNet використовуються для обробки зображень документів. Вони дозволяють розпізнавати текст на документах, що додаються користувачами. Завдяки попередньому тренуванню на великих наборах зображень, ці моделі здатні ефективно працювати навіть з поганою якістю фото.

Машинне навчання значно перевищує ефективність ручної перевірки заявок, оскільки автоматизує процес і мінімізує ризик людських помилок. Оцінка ефективності показує:

- Точність обробки текстових заявок: понад 90%.
- Точність розпізнавання тексту на зображеннях: понад 85% навіть для низькоякісних зображень.

- Скорочення часу обробки однієї заявки: до 5 секунд.
- Зменшення навантаження на модераторів: автоматична перевірка дозволяє модераторам концентруватися лише на складних випадках.

Наукова новизна. Дослідження ефективності автоматичної перевірки заявок на допомогу з використанням методів машинного навчання в межах розробки Telegram чат-бота для волонтерської підтримки представляє новизну у контексті інтеграції сучасних технологій верифікації для систем допомоги. В роботі проведено детальний аналіз існуючих методів перевірки заявок, що дозволяє виявити їхні обмеження і визначити можливості для вдосконалення процесу. Пропонується інтеграція моделей машинного навчання для обробки текстових і візуальних даних, що дозволяє автоматизувати верифікацію запитів у реальному часі з високою точністю.

Окремим внеском є розгляд застосування глибоких нейронних мереж для класифікації текстових заявок та використання алгоритмів комп'ютерного зору для обробки доданих фотографій і документів. Ці технології дозволяють здійснювати точну перевірку автентичності заявок, зокрема через автоматичну обробку даних з різних джерел (текстових і візуальних), що значно підвищує ефективність і скорочує час, необхідний для обробки запитів. Новизна дослідження також полягає в розробці системи інтеграції машинного навчання для забезпечення більш гнучкої та масштабованої системи верифікації для великих обсягів користувачів і заявок у рамках волонтерських ініціатив.

Висновки. Розроблений Telegram чат-бот, який використовує методи машинного навчання для автоматичної перевірки заявок на допомогу, показав високу ефективність у порівнянні з традиційними методами верифікації. Застосування алгоритмів NLP для обробки текстових запитів, а також технологій комп'ютерного зору та OCR для аналізу візуальних даних дозволяє здійснювати точну та швидку перевірку заявок. Оцінка ефективності показала значне скорочення часу обробки заявок, зменшення навантаження на модераторів та підвищення точності верифікації до 90% для текстових заявок і 85% для тексту на зображеннях, навіть при низькій якості. Таким чином, результат дослідження підтверджує можливість використання машинного навчання для автоматизації процесу перевірки заявок, що значно підвищує ефективність волонтерських ініціатив і забезпечує кращу підтримку користувачів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili, Machine Learning with PyTorch and Scikit-Learn. Packt. 2022. – 716 c.
2. François Chollet, Deep Learning with Python. Manning Publications, 2021. – 114 c.
3. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2022. – 76 c.
4. Adrian Rosebrock, Deep Learning for Computer Vision with Python. PyImageSearch, 2019. – 144c.
5. Rajalingappaa Shanmugamani, Deep Learning for Computer Vision. Packt Publishing, 2018. – 157 c.

РОЗДІЛ 3

ПРОГРАМНІ ЗАСОБИ УПРАВЛІННЯ, ЗБОРУ, ОБРОБКИ І ПЕРЕДАЧІ ІНФОРМАЦІЇ

УДК 519.6:534.2:004.94

А.Л. Ширін[1], О.В. Удовик[1], С.М. Мацюк[1], А.Д. Балян[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

БІБЛІОТЕКА REACT.JS ЯК ІНСТРУМЕНТ ОПТИМІЗАЦІЇ ВЕБЗАСТОСУНКІВ

Анотація. В роботі наведено інструменти оптимізації продуктивності вебзастосунків в React.js. Розглянуто принципи роботи бібліотеки React.js для створення швидких, інтуїтивних та адаптованих до різних пристроїв інтерфейсів, що суттєво покращує взаємодію з користувачем.

Ключові слова: *React, JavaScript, вебзастосунок, оптимізація, продуктивність, адаптація, інтерфейс, DOM, UI/UX.*

Вступ. React.js – це одна з найпоширеніших бібліотек JavaScript (рис.1), створена компанією Facebook для розробки користувацьких інтерфейсів [1]. Завдяки компонентній архітектурі, React надає можливість будувати масштабовані та модульні вебзастосунки. Це особливо корисно для проектів з динамічним контентом, де необхідно ефективно керувати великою кількістю інтерактивних елементів.

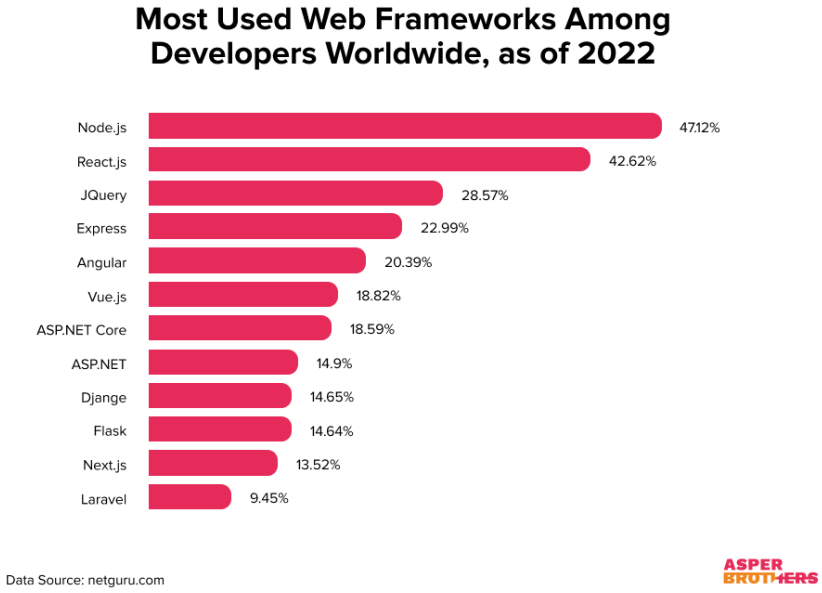


Рис. 1. Популярність React.js

Основною перевагою React є можливість повторного використання компонентів, що суттєво полегшує процес розробки та подальшого підтримання коду. Односпрямований потік даних дозволяє більш чітко контролювати стан компонентів і забезпечує стабільну роботу інтерфейсу, що робить React ідеальним рішенням для побудови систем управління завданнями, нагадуваннями та іншими динамічними системами.

Основний зміст роботи. React.js будується навколо кількох ключових концепцій: компонентів, стану (state) та властивостей (props). Кожен компонент є окремою одиницею інтерфейсу зі своєю логікою, що дозволяє легко керувати кодом і робити його більш зрозумілим та підтримуваним [2].

Основні принципи роботи React наведено нижче:

– Компоненти. React надає можливість створювати незалежні, багаторазові елементи інтерфейсу, що сприяє спрощенню розробки. Наприклад, один і той самий компонент можна використовувати в різних частинах застосунку.

– Стан і його оновлення. Компоненти в React можуть мати власний стан, який керує їхньою поведінкою. Це дозволяє створювати динамічні інтерфейси, де зміни у стані компонентів автоматично призводять до оновлення відповідних частин сторінки.

– Віртуальний DOM. React використовує віртуальний DOM для більш ефективного управління змінами у реальному DOM. Це дозволяє застосункам швидко реагувати на зміни даних, оновлюючи тільки ті частини інтерфейсу, які потребують оновлення.

Процес створення вебзастосунку за допомогою React розпочинається з визначення компонентів, які відповідатимуть за різні частини інтерфейсу. Наприклад, для систем управління завданнями можна створити окремі компоненти для списку завдань, форми для додавання нових елементів та функцій редагування або видалення.

React пропонує сучасний підхід до роботи зі станом за допомогою хуків (React Hooks). Зокрема, хук `useState` дозволяє керувати станом компонентів, а `useEffect` – виконувати додаткові дії при рендерингу чи зміні стану. Це спрощує роботу з асинхронними даними та забезпечує гнучкість у побудові логіки застосунку.

Оптимізація продуктивності в React.js. Одним із найбільших переваг React є висока продуктивність (рис. 2) завдяки використанню віртуального DOM [3]. Проте, для великих і складних проектів, де є багато компонентів, можна додатково оптимізувати роботу застосунку.

Інструменти, що дозволяють оптимізувати застосунок [4]:

– React.memo. Ця функція допомагає запобігти зайвим перерендерингам компонентів, оновлюючи їх лише тоді, коли змінюються їхні властивості. Це суттєво підвищує продуктивність, особливо в застосунках з великою кількістю динамічних елементів.

– useMemo. Використання цього хука дозволяє кешувати результати обчислень, що залежить від певних змінних, щоб не виконувати їх щоразу під час оновлення компонента.

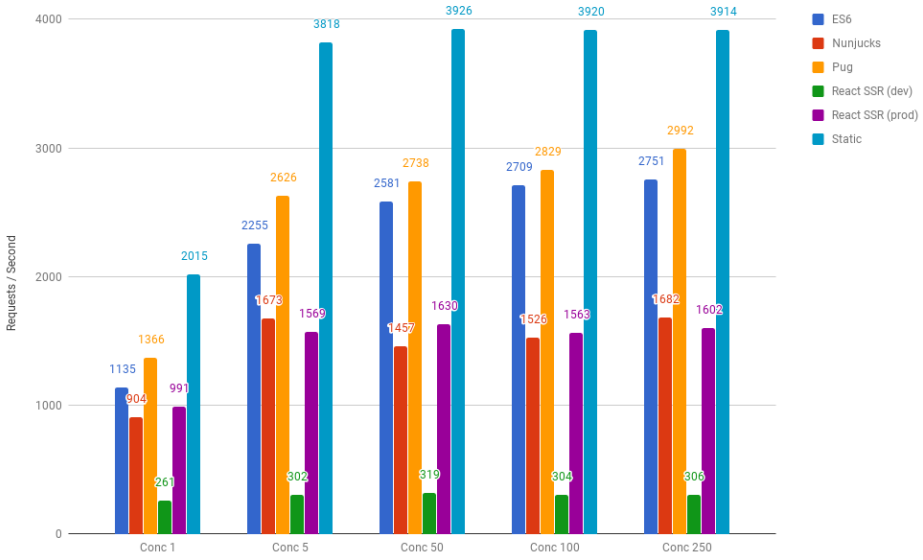


Рис. 2. Швидкість роботи різних технологій (важливий ES6 та React)

– React.lazy. Ефективна практика lazy-завантаження компонентів, що дозволяє завантажувати лише ті частини інтерфейсу, які необхідні в конкретний момент часу. Це зменшує загальний час завантаження сторінки та підвищує швидкість реакції вебзастосунку.

Адаптивність і покращення взаємодії з користувачем. Для того щоб вебзастосунок був зручним на різних пристроях, важливо забезпечити його адаптивність. Використання таких інструментів, як Tailwind CSS або Styled Components, допомагає швидко налаштовувати інтерфейс під різні розміри екранів, зберігаючи при цьому консистентний зовнішній вигляд компонентів.

Асинхронна взаємодія з сервером, яку можна реалізувати за допомогою сучасних JavaScript API (наприклад, `fetch` або `axios`), дозволяє користувачам працювати з вебзастосунком без необхідності постійного перезавантаження сторінки [5]. Це покращує швидкість взаємодії та робить використання застосунку більш комфортним.

Висновки. Застосування React.js для розробки вебзастосунків відкриває широкі можливості для створення швидких, інтуїтивних та продуктивних інтерфейсів користувача. Завдяки компонентній архітектурі, ефективній роботі з даними через віртуальний DOM і сучасним інструментам для оптимізації, React.js забезпечує надійні рішення для динамічних проектів з великою кількістю інтерактивних елементів. Гнучкість React дозволяє розробникам легко масштабувати проекти, а використання таких інструментів, як хуки та менеджери стану, значно спрощують процес підтримки та розвитку застосунків. Завдяки цьому, бібліотека React дійсно є одним із найкращих ресурсів для

створення сучасних вебзастосунків, орієнтованих на ефективну взаємодію з кінцевим користувачем.

ПЕРЕЛІК ПОСИЛАНЬ

1. Aleksander Furgal. Next JS vs React: Understanding the Differences and Making the Right Choice. 9 May 2023. [електронний ресурс]. URL: https://asperbrothers.com/blog/next-js-vs-react/ (дата звернення: 21.10.2024).
2. React vs JavaScript 2024, Advantages, Benefits, and Comparison [електронний ресурс]. URL: https://appwrk.com/reactjs-vs-plain-javascript (дата звернення: 21.10.2024).
3. The Performance Cost of Server Side Rendered React on Node.js [електронний ресурс]. URL: https://malloc.fi/performance-cost-of-server-side-rendered-react-node-js (дата звернення: 21.10.2024).
4. useMemo [електронний ресурс]. URL: https://react.dev/reference/react/useMemo (дата звернення: 21.10.2024).
5. React Optimization Techniques to Help You Write More Performant Code [електронний ресурс]. URL: https://www.freecodecamp.org/news/react-performance-optimization-techniques/ (дата звернення: 21.10.2024).

УДК 004.415.3:681.6

В.В. Спірінцев[1], О.С. Генчук[1]

[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ПРОКСІ-СЕРВЕРА НА NODE.JS

Анотація. В роботі досліджено підходи до розробки проксі-сервера на платформі Node.js, описано процес створення власної реалізації проксі-сервера, включаючи архітектурні рішення та використані модулі. Проведено аналіз існуючих рішень у цій галузі. Здійснено тестування продуктивності розробленого рішення, результати якого порівняно з показниками інших популярних реалізацій. На основі отриманих даних виконано аналіз ефективності запропонованого проксі-сервера.

***Ключові слова**: проксі, сервер, Node.js, швидкодія, ефективність, оптимізація, Squid Proxy, Tinyproxy.*

Вступ. Проксі-сервери є важливим елементом сучасної інтернет-інфраструктури, виконуючи різноманітні функції для покращення роботи з мережевим трафіком. Вони сприяють оптимізації, підвищенню рівня безпеки та ефективному управлінню даними, що проходять через мережу [1]. Завдяки своїй гнучкості та широким можливостям, проксі-сервери стали невід'ємною складовою багатьох технологічних рішень. Вони активно використовуються в корпоративних мережах для забезпечення контролю доступу, у хмарних сервісах для розподілу навантаження, а також у персональних рішеннях для захисту конфіденційності та анонімності в Інтернеті. Крім того, проксі-сервери можуть

виконувати роль кешуючих вузлів, зменшуючи затримки доступу до ресурсів і знижуючи навантаження на основні сервери.

Постановка задачі. Основними вимогами до будь-якого проксі-сервера є висока швидкодія та тривалий час безперервної роботи. Дане дослідження виконано в межах розробки програмного пакету для створення проксі-сервера, головною особливістю якого є забезпечення максимально довгого часу безперервної роботи, а також зручність в налаштуванні. Досягнення цієї мети стало можливим завдяки впровадженню незалежних підслужб на базі Node.js. При цьому пріоритет було надано стабільності роботи, тоді як оптимізація швидкодії відійшла на другий план. Основна мета дослідження полягає у тестуванні швидкодії створеного проксі-сервера на Node.js, його порівнянні з іншими популярними рішеннями та аналізі отриманих результатів.

Основний зміст роботи. Node.js пропонує широкий набір інструментів для розробки проксі-серверів завдяки своїй подієво-орієнтованій архітектурі та ефективній роботі з асинхронними операціями. Базові модулі http і https дозволяють створювати прості проксі-сервери для обробки HTTP та HTTPS-запитів. З їх допомогою можна перехоплювати запити клієнтів, модифікувати їх або перенаправляти на інші сервери. Модуль http-proxy, створений спільнотою, значно спрощує розробку, надаючи високорівневий API для реалізації функцій проксі-сервера, таких як маршрутизація, балансування навантаження та підтримка безпеки через SSL/TLS. Крім того, бібліотеки, такі як express або koa, у поєднанні з цими модулями дозволяють швидко додавати функції, такі як кешування чи автентифікація. Завдяки своїй масштабованості Node.js є гарним вибором для створення проксі-рішень, особливо для обробки високонавантажених систем у реальному часі.

Для дослідження було розроблено прототип проксі-сервера на основі вбудованого модуля http (рис. 1). Цей модуль відповідає за обробку вхідних HTTP-запитів, забезпечуючи базову функціональність обслуговування клієнтів. Для створення мережевих з'єднань та передачі даних між клієнтом і цільовим сервером використовується модуль net, що дозволяє реалізувати низькорівневі операції з TCP-з'єднаннями.

Розроблений додаток демонструє приклад найпростішої реалізації проксі-сервера, головним завданням якого є підміна IP-адреси користувача. Завдяки використанню цих модулів забезпечується базова функціональність проксі-сервера, що дозволяє перенаправляти запити через нього, приховуючи справжню IP-адресу клієнта.

Існує велика кількість рішень для створення проксі-серверів: Squid Proxy, Tinyproxy, Privoxy, HAProxy, Dante тощо. В цьому дослідженні буде розглянуто найпопулярніші, Squid Proxy і Tinyproxy.

Squid Proxy – це потужний та популярний проксі-сервер, який спеціалізується на кешуванні веб-контенту та оптимізації роботи мереж. Він широко використовується для зменшення затримок під час доступу до часто запитуваних ресурсів, оскільки зберігає копії статичного контенту, як-от

зображення, сторінки та інші файли, що скорочує навантаження на мережу та віддалені сервери. Squid підтримує кілька протоколів, включаючи HTTP, HTTPS, FTP, і може працювати як прозорий проксі, забезпечуючи мінімальне втручання в мережеву інфраструктуру [2].

```
const http = require('http')
const port = 9191
const net = require('net')
const url = require('url')
const requestHandler = (req, res) => {
  res.writeHead(405, {'Content-Type': 'text/plain'})
  res.end('Method not allowed')
}
const server = http.createServer(requestHandler)
const listener = server.listen(port)
server.on('connect', (req, clientSocket, head) => {
  const {port, hostname} = url.parse(`//${req.url}`, false, true)
  if (hostname && port) {
    const serverErrorHandler = (err) => {
      console.error(err.message)
      if (clientSocket) {
        clientSocket.end(`HTTP/1.1 500 ${err.message}\r\n`)
      }
    }
    const serverEndHandler = () => {
      if (clientSocket) {
        clientSocket.end(`HTTP/1.1 500 External Server End\r\n`)
      }
    }
    const serverSocket = net.createConnection({
      port: port,
      host: hostname,
      localAddress: '10.0.1.177'
  })
    const clientErrorHandler = (err) => {
      console.error(err.message)
      if (serverSocket) {
        serverSocket.end()
      }
    }
    const clientEndHandler = () => {
      if (serverSocket) {
        serverSocket.end()
      }
    }
    clientSocket.on('error', clientErrorHandler)
    clientSocket.on('end', clientEndHandler)
    serverSocket.on('error', serverErrorHandler)
    serverSocket.on('end', serverEndHandler)
    serverSocket.on('connect', () => {
      clientSocket.write([
        'HTTP/1.1 200 Connection Established',
        'Proxy-agent: Node-VPN',
      ].join('\r\n'))
      clientSocket.write('\r\n\r\n')
      serverSocket.pipe(clientSocket, {end: false})
      clientSocket.pipe(serverSocket, {end: false})
    })
  } else {
    clientSocket.end('HTTP/1.1 400 Bad Request\r\n')
    clientSocket.destroy()
  }
})
```

Рис. 1. Код досліджуваного додатка

Завдяки можливості фільтрувати контент, обмежувати доступ до небажаних ресурсів та аналізувати мережевий трафік, Squid часто використовується для забезпечення безпеки та контролю в корпоративних мережах. Його конфігурація гнучка та дозволяє тонко налаштовувати правила кешування, маршрутизації та аутентифікації, що робить його одним із найбільш універсальних проксі-рішень.

Tinyproxy – це легкий, швидкий і простий у налаштуванні проксі-сервер, орієнтований на мінімальне споживання системних ресурсів [3]. Завдяки своїй компактній архітектурі він ідеально підходить для невеликих мереж або пристроїв із обмеженими ресурсами, таких як маршрутизатори чи IoT-пристрої. Tinyproxy підтримує HTTP-протокол і може використовуватись як стандартний або прозорий проксі-сервер. Його конфігурація здійснюється через один текстовий файл, що робить процес налаштування швидким і зрозумілим навіть для новачків. Сервер дозволяє застосовувати контроль доступу, блокувати або дозволяти трафік на основі IP-адрес або доменів, що забезпечує базовий рівень безпеки. Tinyproxy також підтримує зворотний проксі, що дає можливість використовувати його для перенаправлення запитів на внутрішні сервери. Завдяки простоті використання та низькому споживанню ресурсів Tinyproxy є популярним вибором для легких і цільових проксі-рішень.

Тестовим стендом є віртуальний виділений сервер орендований на сервісі onecloudplanet.com, специфікації наведено на рис.2. Сервер розташований у місті Київ. Для тестування швидкодії використано програму FOGLDN Proxy Tester, запущену локально на персональному комп’ютері у місті Дніпро.

Рис. 2. Специфікації орендованого серверу

При тестуванні перевіряється підключення до сторінок: https://www.google.com/, https://www.youtube.com/, https://github.com/, https://x.com/home і https://stackoverflow.com/. До кожної сторінки зроблено 100 ітерацій запитів, по черзі через кожне із досліджуваних рішень. В результатах розглядається середня затримка в мілісекундах.

Результати тестування, наведені в табл.1, демонструють, що розроблений проксі-сервер на базі Node.js показує середню продуктивність у порівнянні з іншими популярними рішеннями. Зокрема, його швидкодія виявилася на 2.7% нижчою у порівнянні зі Squid Proxy, який є одним із найбільш оптимізованих і

широко використовуваних проксі-серверів. Однак, у порівнянні з Tinyproxy, який має легшу архітектуру, сервер на Node.js виявився на 39.2% швидшим.

Таблиця 1

Результати тестування швидкодії

	Проксі-сервер на Node.js	Squid Proxy	Tinyproxy
https://www.google.com/	111	108	159
https://www.youtube.com/	109	108	145
https://github.com/	123	120	166
https://x.com/home	151	145	193
https://stackoverflow.com/	89	86	140

Ці показники свідчать про те, що проксі-сервер на Node.js здатний забезпечувати конкурентоспроможну продуктивність навіть без глибокої оптимізації швидкодії. Варто зазначити, що такі результати були досягнуті завдяки використанню асинхронної архітектури Node.js, яка дозволяє ефективно обробляти численні запити одночасно. Аналіз показників також підкреслює перспективність застосування Node.js для створення проксі-серверів у сценаріях, де важливий баланс між продуктивністю та простотою реалізації. Подальша оптимізація, наприклад, через використання кешування або спеціалізованих алгоритмів маршрутизації, може ще більше скоротити розрив із високопродуктивними рішеннями на кшталт Squid Proxy.

Наукова новизна. Наукова новизна роботи полягає у дослідженні можливостей платформи Node.js для розробки проксі-серверів з урахуванням сучасних вимог до продуктивності та стабільності. У роботі детально проаналізовано вплив подієво-орієнтованої архітектури Node.js на швидкодію проксі-сервера, що є новим кроком у порівнянні з традиційними рішеннями, такими як Squid Proxy та Tinyproxy. Було розроблено та протестовано прототип проксі-сервера на Node.js, орієнтований на простоту налаштування та адаптацію під сценарії з помірними вимогами до продуктивності. Отримані результати розширюють наукове уявлення про можливості використання Node.js у мережевих рішеннях, акцентуючи увагу на балансі між ефективністю, простотою реалізації та стабільністю роботи.

Висновки. У рамках виконаного дослідження було розглянуто та реалізовано прототип проксі-сервера на платформі Node.js із використанням базових модулів http і net. Результати тестування продуктивності показали, що розроблене рішення демонструє конкурентоспроможність у порівнянні з популярними проксі-серверами, такими як Squid Proxy та Tinyproxy.

ПЕРЕЛІК ПОСИЛАНЬ

1. The Role of Proxy Servers in Web Security [Електронний ресурс]. – Режим доступу: https://cloudinfrastructureservices.co.uk/the-role-of-proxy-servers-in-web-security/ (дата звернення: 19.11.2024)
2. Squid: Optimising Web Delivery [Електронний ресурс]. – Режим доступу: https://www.squid-cache.org/ (дата звернення: 20.11.2024)
3. Tinyproxy lightweight http(s) proxy daemon [Електронний ресурс]. – Режим доступу: https://tinyproxy.github.io/ (дата звернення: 20.11.2024

УДК 519.6:534.2:004.94

Л.І. Мещеряков[1], А.Т. Харь[1], М.В. Куваєв[2], А.І. Політов[3]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна
[2]Інститут транспортних систем і технологій НАН України, Дніпро, Україна
[3]Дніпровський національний університет імені Олеся Гончара, Дніпро, Україна

ОРГАНІЗАЦІЯ ЛЮДИНО-МАШИННОГО ІНТЕРФЕЙСУ ДЛЯ НАЛАГОДЖЕННЯ І СУПРОВОДЖЕННЯ НА ОБ'ЄКТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМ КЕРУВАННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСОМ

Анотація. Описані особливості підходів до реалізації людино-машинного інтерфейсу для налагодження і супроводження програмного забезпечення багатозадачних систем керування технологічним процесом, що дають можливість скоротити час налагодження та налаштування програмного забезпечення систем на об'єкті керування пошуку і порушень які виникають при експлуатації таких систем.

Ключові слова: *системи керування технологічним процесом, людино-машинний інтерфейс, програмне забезпечення.*

Вступ. Одним з порівняно маловитратних способів підвищення техніко-економічних показників роботи діючих технологічних ліній є модернізація систем керування як окремими агрегатами і ділянками таких ліній, так і технологічним процесом в цілому без заміни самого технологічного обладнання. Така модернізація на діючим технологічному об'єкті передбачає поетапне оновлення систем автоматизації без тривалих зупинок технологічного процесу – з монтажем і налаштуванням як у планові ремонти і зупинки технологічного процесу, так і під час штатної роботи. При розробці систем автоматизації традиційним став підхід, що базується на застосуванні дворівневої ієрархічної системи автоматизації, в якій на нижньому рівні ієрархії комп'ютерна система автоматизації забезпечує отримання інформації з об'єкта керування і видачу на останній керуючих впливів, у тому числі – в автоматичному режимі, а на верхньому – реалізується людино-машинний інтерфейс оператора технолога.

Найбільш широке застосування при розробці людино-машинних інтерфейсів систем керування технологічним процесом знайшли програмні засоби SCADA [1,2,3]. Такі системи надають широкий спектр інструментарію для програмування людино-машинного інтерфейсу, але вони орієнтовані за своїм функціоналом на операторів-технологів, що безпосередньо керують технологічним процесом, залишаючи поза уваги питання зручностей експлуатації та налаштування таких систем, зокрема їх програмного забезпечення. Особливо це відноситься до процесу налаштування систем керування технологічним процесом в процесі їх роботи з мінімізацією часу непродуктивних простоїв для пошуку помилок і корегування програмного забезпечення.

Постановка задачі. Виходячи з викладеній у вступі проблеми, виникає задача необхідності організації людино-машинного інтерфейсу для налаштування і супроводження програмного забезпечення систем керування технологічним процесом для скорочення часу пошуку порушень, які виникають в процесе налаштування таких систем, та скорочення часу налаштування таких систем і непродуктивних простоїв що пов’язані з корегуванням їх програмного забезпечення.

Основний зміст роботи. При впроваджені систем автоматизації технологічних процесів одним з найбільш відповідальним етапом є налагодження системи на технологічному об’єкті керування і введення її в експлуатацію. На нових об’єктах системи автоматизації налагоджуються в рамках загального налаштування і відпрацювання всього технологічного процесу, і практично не впливають на час вводу об’єкта керування в експлуатацію. Зовсім інша картина має місце при модернізації систем автоматизації на діючих об’єктах – процес налагодження супроводжується позаплановими, або запланованими зупинками технологічного процесу для перевірки системи автоматизації, яка налагоджується, що призводить до втрат у виробництві. Сучасні комп’ютерні технології зі створення систем автоматизації технологічних процесів дозволяють відпрацювати більшість алгоритмів на імітаторах до встановлення обладнання на об’єкті, але питання динамічної взаємодії окремих задач і впливу на роботу прикладного програмного забезпечення системи позаштатних ситуацій, що супроводжують функціонування технологічного об’єкта керування, потребують відпрацювання програм в умовах діючого технологічного процесу. Ще більша важливість у інструментарії, який би полегшив відпрацювання прикладного програмного забезпечення безпосередньо на об’єкті керування, виникає у випадку розширенні функціональних можливостей діючої комп’ютерної системи автоматизації. Таке розширення відбувається, як правило, додаванням у існуюче обладнання (контролери) відповідних модулів вводу/виводу і інтеграцію в діюче прикладне програмне забезпечення прикладних програм, що реалізують додатковий функціонал. Слід зазначити, що питання створення інструментарію, який полегшує відпрацювання і супроводження прикладного програмного

забезпечення безпосередньо на об’єкті керування, залишилися поза увагою розробників і користувачів програмних пакетів SCADA-систем.

Задача створення зручного людино-машинного інтерфейсу для відпрацювання і супроводження багатозадачного програмного забезпечення безпосередньо на діючому об’єкті керування складної комп’ютерної системи, критичної до режиму реального часу, вирішувалася при впровадженні і супроводжені комп’ютерної системи керування швидкісним режимом прокатки сортової лінії дрібносортно-дротового стана ДСДС 250/150 та комп’ютерної системи керування швидкісним режимом прокатки чистової групи клітей дрібносортного стана ДСС 250-2 ПАТ «АрселорМіттал Кривий Ріг».

З функціональною структурою і основними програмними рішеннями системи для дрібносортно-дротового стана ДСДС 250/150 можна ознайомитися в [4]. Людино-машинний інтерфейс для супроводження як технічного забезпечення [5], так і програмного забезпечення комп’ютерної системи керування швидкісним режимом прокатки, створювалися як окремі підсистеми, доступ до яких здійснювався з HMI оператора технолога, опис наведено в [6].

Система для лівої чистової групи клітей дрібносортного стана ДСС 250-2 за основними рішеннями аналогічна системі для ДСДС 250/150, але передбачалось її подальше розширення на другу, праву, чистову і чорнову групи клітей цього стана. Загальна структура технічних засобів системи включає такі комп’ютерні вузли, як основний і резервний керуючи обчислювальні вузли, що встановлені у машинному залі прокатного стана, та вузол оператора-технолога лівої чистової групи клітей (кліті №8-№15), що встановлений на посту керування ПК-6. Передбачалося подальше розширення системи на праву чистову групу клітей (кліті №16-№23), що керувалася з поста керування ПК-5, та чорнову групу клітей (кліті №1-№7), що керувалася з поста керування ПК-4, з встановлення відповідних вузлів керування операторов-технологов на ПУ-4, ПУ-5.

При створенні людино-машинного інтерфейсу супроводження програмного забезпечення для систем керування швидкісним режимом прокатки сортової лінії дрібносортно-дротового стана ДСДС 250/150 і чистової групи клітей дрібносортного стана ДСС 250-2 використовувалися ті ж самі підходи, що і при створенні підсистем технічного супроводження цих систем, яке описано у [5], але вони були адаптовані до особливостей структури програмного забезпечення і інформації, яка є актуальною при аналізі його функціонування.

У програмному забезпеченні системи дрібносортного стана ДСС 250-2, на відміну від програмного забезпечення систем керування швидкісним режимом прокатки сортової лінії дрібносортно-дротового стана ДСДС 250/150, що наведена в [4], в який кожна функціональна задача працює на обчислювальному вузлі як окремий процес, такі задачі працюють як потоки в складі єдиного процесу для всього обчислювального вузла. Це суттєво спростило і скоротило час обміну даними між функціональними задачами, що працюють на відповідному обчислювальному вузли.

Розглянемо структуру і функціонал людино-машинного інтерфейсу супроводження програмного забезпечення на прикладі такого інтерфейсу системи керування швидкісним режимом прокатки чистової групи клітей дрібносортного стана ДСС 250-2. Вигляд його основних сторінок наведено на рис.1-3.

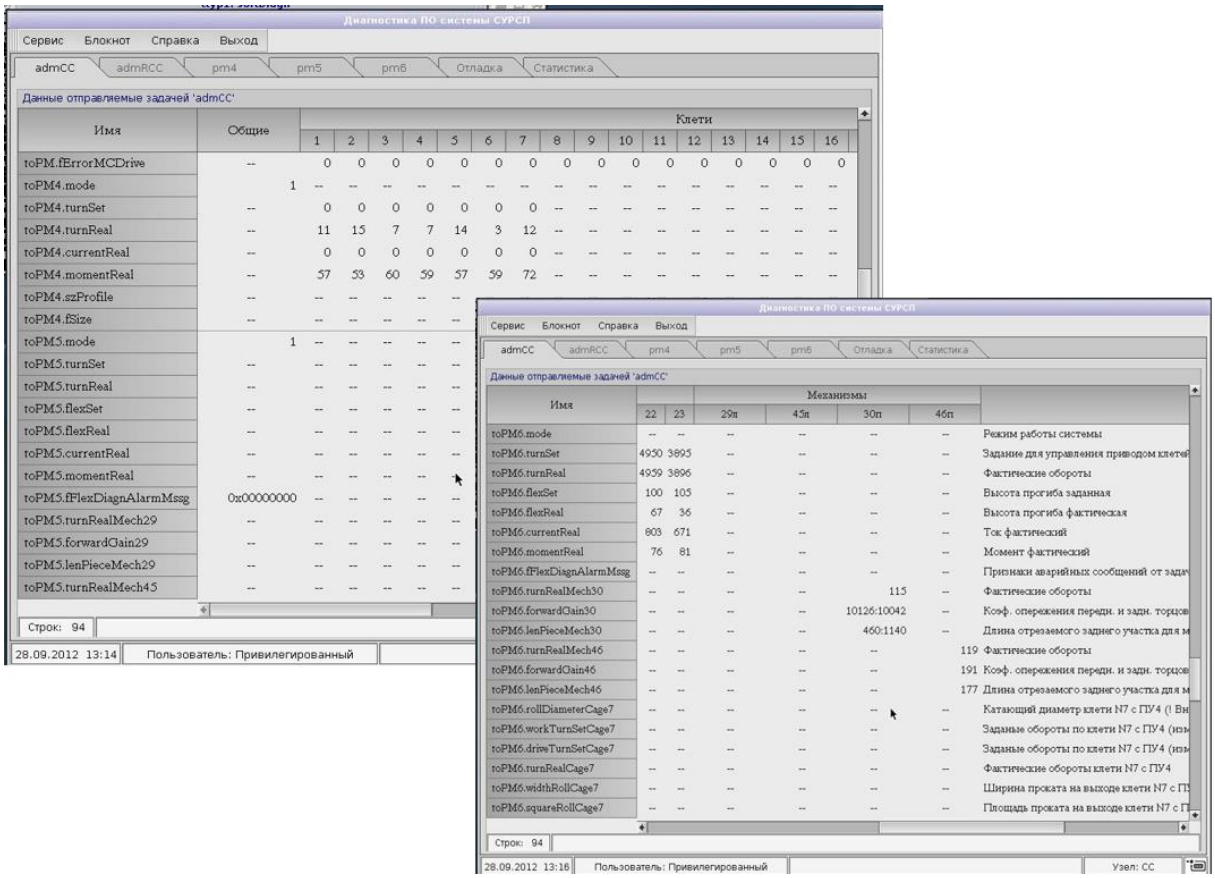


Рис.1. Людино-машинний інтерфейс супроводження програмного забезпечення системи керування швидкісним режимом прокатки чистової групи клітей дрібносортного стана ДСС 250-2.

Як видно з рис.1, 2 всі змінні за стовпчиками згруповані відповідно до клітей, з якими вони пов’язані, а за рядками – за назвою параметра, що описує відповідна змінна. Окремо виділено стовпчик, в якому розміщено параметри, що є спільними для всіх клітей одночасно, або для вузла в цілому. Крайній правий рядок дає розшифровку параметра, який описує кожна змінна. Таке надання інформації дає змогу, з одного боку, комплексно аналізувати поточний стан кожної кліті, а з іншого – динаміку реакції програмного забезпечення обчислювального вузла на перехідні режими роботи прокатного стану, наприклад, послідовного проходження клітей заготовкою в процесі її прокатки.

Комплексний аналіз роботи програмного забезпечення системи керування швидкісним режимом прокатки здійснюється по інформації, що надається на

сторінки «Статистика» (див. рис.3). Інформація, що надається на цій сторінці, дає змогу проаналізувати поточний стан потоків, дотримання поточним програмним забезпеченням вимог режиму реального часу його функціонування та наявність резервів щодо розширення його функціонального завантаження.

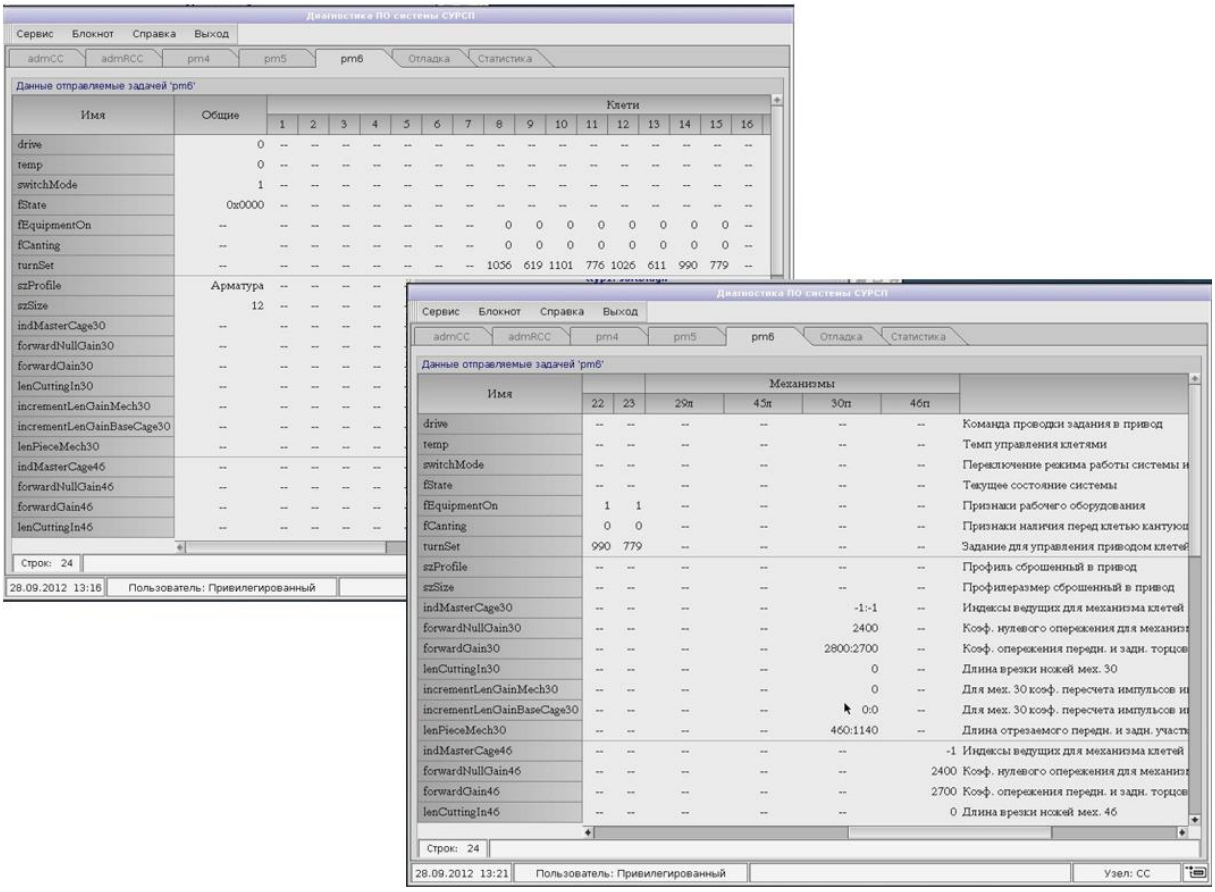


Рис. 2. Сторінка керуючого обчислювального вузла **admCC**

Сторінка блокнота «Статистика» включає такі стовбці: «Ім’я», «Вузол», «Розмір повідомлення», «Стан», «Watchdog лічильник», «Період роботи [мс]».

Стисло розглянемо інформацію, що наведена в рядках. В стовбці «Ім’я» наведені імена процесів (притиснуті до лівої границі стовбця), а безпосередньо під кожним ім’ям процесу, зі зсувом відносно лівої границі стовбця, в рядках надані імена потоків цього процесу. У наступному стовпці «Вузол», у рядку, в якому надано ім’я процесу, надана абревіатура обчислювального вузла, на якому працює цей процес, а в стовпці «Розмір повідомлення» – розмір повідомлення у байтах, яким цій процес чи потік обмінюється з підсистемою супроводження програмного забезпечення. В стовбці «Стан» визначено, чи є цей процес чи потік у нормальному активному стані (стан «норма»), чи у стані «зомбі», тобто у стані завершення, чи закритий (стан «вмер»).

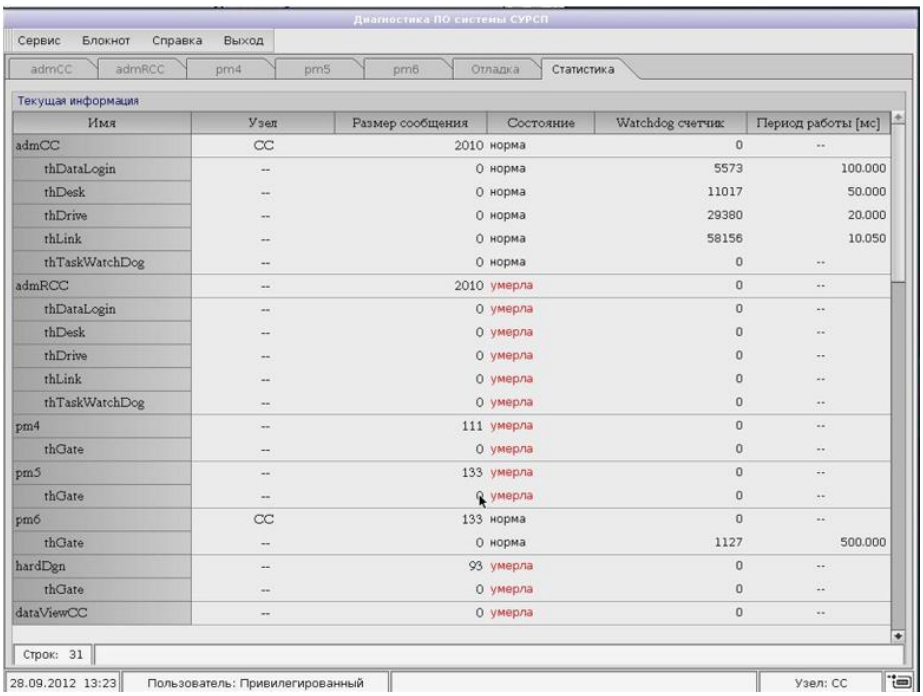


Рис. 3. Людино-машинний інтерфейс супроводження програмного забезпечення системи керування швидкісним режимом прокатки чистової групи клітей дрібносортного стана ДСС 250-2. Сторінка **«Статистика»**

Стовпчики «Watchdog лічильник» та «Період роботи [мс]» дають змогу контролювати дотримання режиму реального часу кожного з потоків і, в кінцевому випадку, оцінити запас часу для розширення функціонального завантаження програмних потоків та можливостей до додавання нових потоків в програмне забезпечення обчислювальних вузлів. «Watchdog лічильник» інкрементується у кожному окремому потоці системи керування швидкісним режимом прокатки з періодичністю роботи цього потоку. За зміною «Watchdog лічильника» розраховується періодичність роботи відповідного потоку, яка відображається у комірки стовпчика «Період роботи [мс]» відповідного потоку. У разі виходу значення періоду за запланований поріг, залежно від важливості завдання, що виконується потоком, приймається одна з наступних дій:

- розширюється запланований період роботи потоку;
- змінюється пріоритет виконання потоку відповідно до інших потоків;
- змінюється алгоритм роботи потоку таким чином, щоб він вкладався у відведений йому час виконання.

Сторінка «Налагодження» дозволяє за потреби програміста виводити на екран будь яку контрольну інформацію, яку програміст запрограмував заздалегідь, і активував при перезавантаженні системи керування.

Досвід налагодження складних багатофункціональних багатозадачних та критичних до режиму реального часу систем керування швидкісним режимом

прокатки показав доцільність розробки в їх складі підсистем програмного супроводження, що забезпечує скорочення часу налагодження системи, і дозволяє виконувати більш глибокий аналіз коректності функціонування програмного забезпечення безпосередньо в процесе керування системою промисловим об’єктом.

Наукова новизна полягає у обґрунтуванні рішень по організації людино-машинного інтерфейсу налаштування і супроводження програмного забезпечення систем керування технологічним процесом, які скорочують час пошуку помилок у програмному забезпеченні, і запобігає непродуктивним втратам часу на аналіз функціонування програмного забезпечення таких систем.

Висновки. При розробці складних інформаційно-керуючих систем автоматизації критичних до режиму реального часу слід передбачати крім традиційного людино-машинного інтерфейсу оператора-технолога ще й людино-машинний інтерфейс програмного налаштування і супроводження цієї системи, що скорочує витрати часу на її налагодження і супроводження. Такий інтерфейс повинен забезпечувати швидкий і зручний доступ до програмних змінних, що характеризують процес керування технологічним об’єктом, та надавати інструментарій для контролю дотримання режиму реального часу та дозволяти оцінити резерви часу для розширення функціональних можливостей системи автоматизації.

ПЕРЕЛІК ПОСИЛАНЬ

1. Пупена О.М. Розроблення людино-машинних інтерфейсів та систем збирання даних з використанням програмних засобів SCADA/HMI: Навч. посіб. Київ: Видавництво Ліра-К, 2020. — 594 с.
2. Bailey D., Wright E. Practical SCADA for Industry. - Newnes, 2005. – 304 p.
3. Автоматизована система диспетчерського керування WindEx. [Электрон. ресурс]. Режим доступу: https://activolt.com.ua/wp-content/uploads/2021/04/windex-.2_2021.pdf (дата звернення 24.04.2023).
4. Kuvaiev, V., Ishchuk, P., Politov, A., Buriak, V. (2021). Prohramni rishennia po zabezpechenniu nadiinoho funktsionuvannia skladnykh informatsiino-keruiuchykh system krytychnykh do rezhymu realnoho chasu [Software solutions to ensure the reliable operation of complex information and control systems critical to real-time]. Information Technology: Computer Science, Software Engineering and Cyber Security, 1, 16–24. [in Ukrainian]. doi: https://doi.org/10.32782/IT/2021-1-3
5. Куваєв В., Мещеряков Л., Харь А., Політов А. Інтерфейс технічного супроводження складних інформаційно-керуючих систем автоматизації. Information Technology: Computer Science, Software Engineering and Cyber Security, 2023, № 2, с. 26-33. doi: https://doi.org/10.32782/IT/2023-2-3
6. Куваєв В., Мещеряков Л., Харь А., Політов А. Розробка інтерфейса оператора складних інформаційно-керуючих систем критичних до режиму реального часу. Information Technology: Computer Science, Software Engineering and Cyber Security, 2023, № 1, с. 50-57. doi: https://doi.org/10.32782/IT/2023-1-7

УДК 004.415.3:681.6

В.В. Спірінцев[1], А.Л. Ширін[1], М.Д. Плющов[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ВАЖЛИВІСТЬ КОЛЬОРІВ У ВЕБДИЗАЙНІ. ВЕБРЕСУРС REALTIME COLORS

Анотація. Описано важливість підбору кольорів при розробці вебсторінок та розглянуто основні ресурси генерації кольорових схем для сайтів.

Ключові слова: *кольорова схема, сайт, підбір кольорів, зручне сприйняття, вебсторінка, UI/UX.*

Вступ. Підбір кольорів для вебсайту є одним із найважливіших аспектів вебдизайну, який впливає на сприйняття користувачів, зручність навігації та загальну естетику ресурсу. Невдало підібрана кольорова схема може спричинити низку проблем: зниження привабливості сайту, погіршення користувацького досвіду та навіть втрату аудиторії. Отже, розуміння принципів колористики, їхнього впливу на користувачів та на здатність сприйняття інформації стає критично важливим при створенні ефективного вебсайту.

Основні виклики підбору кольорів для вебсайту полягають у забезпеченні балансу між комфортом для очей і естетикою, з урахуванням психологічного впливу кольорів, інклюзивності та доступності для людей із порушеннями кольоросприйняття, а також гармонії між естетикою та функціональністю. Надмірно яскраві чи невідповідні відтінки можуть викликати втомленість або негативне емоційне сприйняття, тоді як слабкий колірний контраст може зробити елементи дизайну важкими для сприйняття та знизити ефективність передачі інформації.

Кольори можуть допомогти сформувати ідентичність бренду, спрямовувати користувачів через вебсайт та покращувати загальний досвід користувача. Тому розуміння кольорів та правильне їх використання є важливими для створення візуально привабливих і ефективних вебсайтів [1].

Основний зміст роботи. Насправді, для підбору гармонійних кольорів не обов’язково бути професійним дизайнером. Завдяки систематизації колірних відносин це завдання успішно вирішується програмними алгоритмами. Розглянемо деякі з них:

- Coolors (рис.1);
- WebFx Color Picker (рис.2);
- Realtime Colors (рис.3).

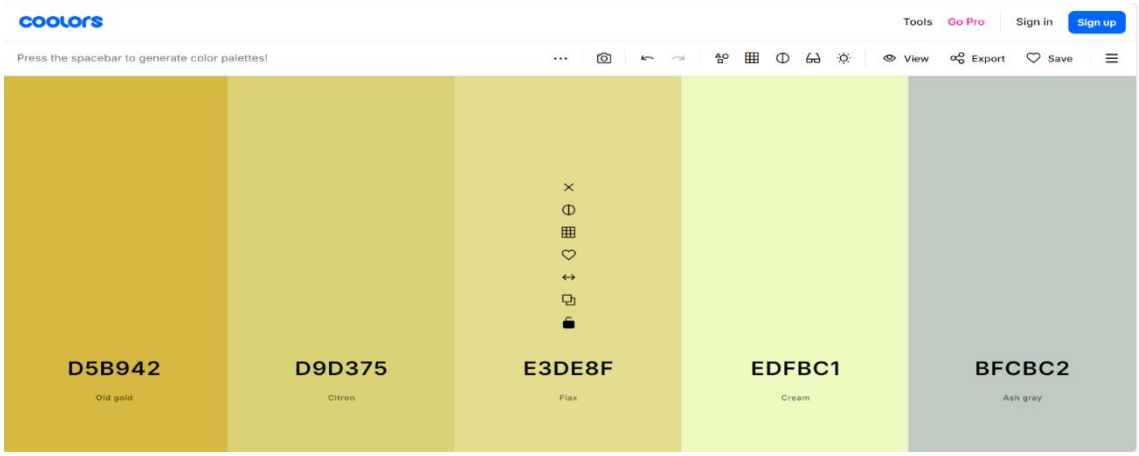


Рис. 1. Інтерфейс вебресурсу Coolors [2]

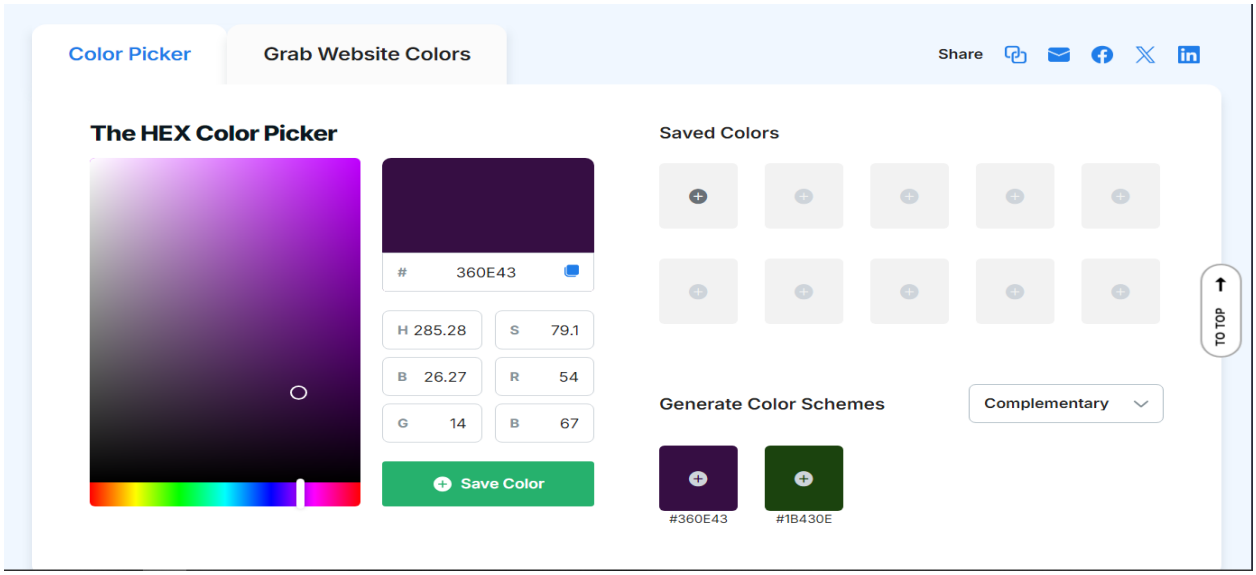


Рис. 2. Інтерфейс вебресурсу WebFx Color Picker [3]

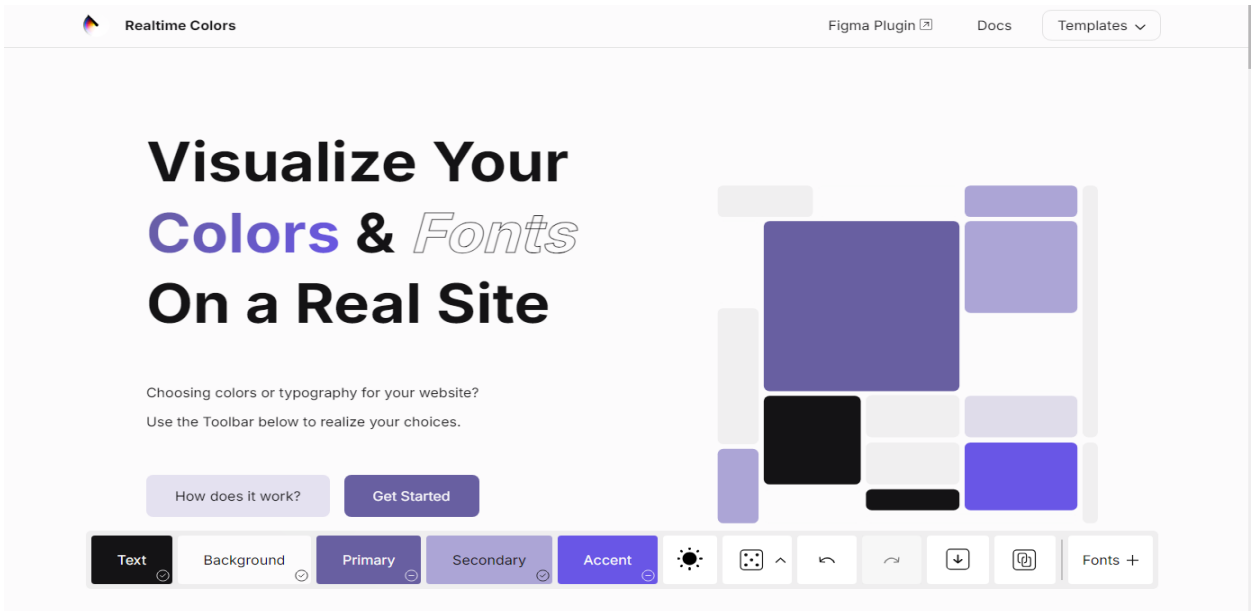


Рис. 3. Інтерфейс вебресурсу Realtime Colors [4]

Кожен з наведених вебресурсів дозволяє згенерувати палітру з 5 кольорів та скопіювати HEX код обраного кольору. Якщо розглядати всі з наведених інструментів окремо, то:

- Coolors має не інтуїтивний дизайн, а також випадково згенеровані палітри рідко підходять для використання на вебсайтах;
- WebFx Color Picker, на відміну від минулого інструмента, має вибір кольорової схеми, але перший колір користувач має обрати сам;
- Realtime Colors пропонує палітру для тексту, фону, основного, другорядного та акцентного кольорів. Палітра генерується випадково, але з урахуванням закономірностей і гармонійних взаємозв'язків згідно з колірною теорією і правилами вебдизайну.

Realtime Colors є одним з ресурсів для підбору кольорів, який має значні переваги. Його сильними сторонами є інтуїтивно зрозумілий інтерфейс, що дозволяє користувачам легко експериментувати з кольоровими схемами, використовуючи інтерактивні елементи для вибору кольорів та їхніх комбінацій. Варто зазначити, що у ньому наявна унікальна функція, сутність якої заключається у можливості бачити зміни в реальному часі, що дозволяє одразу оцінити, наскільки вдалою є обрана палітра.

Також з переваг можна виділити:

- Колірні схеми для світлих і темних тем вебсайту;
- Декілька шаблонів для перегляду палітри на реальному сайті;
- Відсутність реєстрації та реклами при використанні ресурсу.

Висновки. Підбір кольорів для вебсайту є важливим аспектом, що впливає на сприйняття та зручність навігації. В ході дослідження спеціалізованих вебдодатків, Realtime Colors показав себе як створений для максимальної зручності використання інструмент у вебдизайні – кожен колір можна скопіювати у зручному для користувача форматі або імпортувати багатьма способами, а головна можливість описана у самій назві інструмента – кольори у реальному часі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Nikola Misic. The Importance of Color Theory in Web Design: Complete Guide. 2024 [електронний ресурс]. URL: https://cliowebsites.com/color-theory-in-web-design-complete-guide/ (дата звернення: 25.09.2024).
2. Fabrizio Bianchi. Generate a palette - Coolors. 2024 [електронний ресурс]. URL: https://coolors-help.zendesk.com/hc/en-us/articles/360010581980-Generate-a-palette (дата звернення: 25.09.2024).
3. Free Color Picker Tool & Palette Generator. 2021 [електронний ресурс]. URL: https://www.webfx.com/web-design/color-picker/ (дата звернення: 25.09.2024).
4. Juxtopposed. Real-time Colors. 2023 [електронний ресурс]. URL: https://github.com/juxtopposed/realtimecolors (дата звернення: 25.09.2024).

УДК 004.415.3:681.6

Д.В. Вереда[1], В.В. Анісімов[1], В.М. Сушкін[1], А.В. Клименко[2]
[1]ННІ «Український державний хіміко-технологічний університет» УДУНТ
[2]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ПОРІВНЯЛЬНЕ ОЦІНЮВАННЯ ЯКОСТІ ВІДПОВІДЕЙ ЧАТ-БОТІВ CHATGPT І CLAUDE НА ОСНОВІ СТУДЕНТСЬКИХ ОЦІНОК

Анотація. Описано процес оцінки відповідей чат-ботів ChatGPT і Claude на основі визначених критеріїв якості, таких як точність, змістовність і стиль. Проведено анкетування серед студентів, які оцінювали відповіді моделей за шкалою від 0 до 100 балів, на основі чого було побудовано середні оцінки, інтервали оцінок і графічне представлення результатів. Це дослідження дозволяє зробити висновки про сильні та слабкі сторони кожної з моделей, а також можливості їхнього подальшого застосування залежно від контексту запитів.

Ключові слова: *чат-бот, ChatGPT, Claude, точність, змістовність, стиль, анкетування.*

Вступ. З розвитком інформаційних технологій і штучного інтелекту, чат-боти стали важливою частиною багатьох систем для надання консультацій, підтримки користувачів та поширення знань. Однак якість роботи таких систем залежить від їхньої здатності надавати точні, змістовні та стилістично коректні відповіді на запити користувачів. Це особливо актуально для освітніх та інформаційних установ, де коректність і повнота відповідей безпосередньо впливають на сприйняття інформації користувачами. Метою цього дослідження було оцінити якість роботи двох сучасних чат-ботів: ChatGPT і Claude на основі інформації про кафедру інформаційних систем УДУНТ. Потрібно було з'ясувати, наскільки добре ці моделі відповідають на поставлені питання за трьома критеріями: точність, змістовність і стиль. Оцінювання проводилося шляхом анкетування студентів, які аналізували відповіді обох чат-ботів та виставляли оцінки за зазначеними критеріями.

Основний зміст роботи. Було створено анкету, яка містила 10 питань (рис. 1). Відповіді обох моделей на ці запитання оцінювалися студентами за трьома критеріями: точність, змістовність і стиль. Оцінювання здійснювалося за шкалою від 0 до 100 балів. Студентам було роз'яснено, що означає кожен із критеріїв і як вони повинні керуватися цими критеріями для виставлення оцінок.

Для кожного студента було розраховано середню оцінку для кожного критерію окремо для кожної моделі. Результати були згруповані за інтервалами оцінок, де зазначалося кількість студентів, які поставили оцінку у відповідному інтервалі. На основі зведених даних були побудовані графіки розподілу оцінок за кожним критерієм для ChatGPT і Claude.

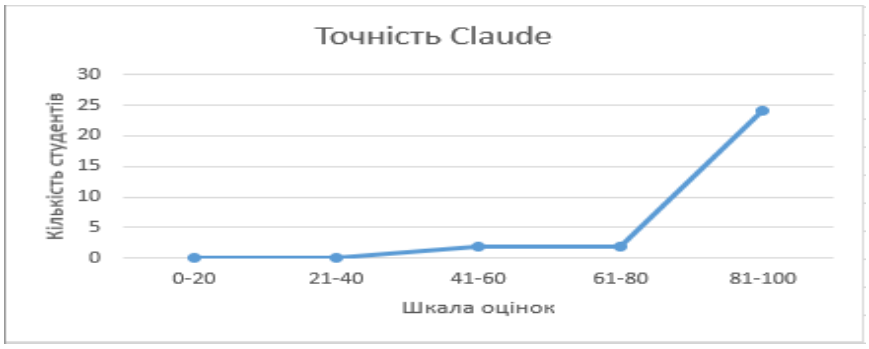
Критерії оцінювання		Ваша	Вся інформація для бази знань була взята з сайту кафедри інформаційних систем - https://udhtu.edu.ua/ftk/knti/kafis
Точність – цей критерій визначає, наскільки точно відповідь чат-бота відповідає змісту бази знань. Висока точність означає, що відповідь не містить фактичних помилок, перекручень або пропусків ключових фактів. Наприклад, інформація про дати, імена, цифри або деталі, що стосуються питання, має бути коректною.			
Змістовність – цей критерій оцінює повноту та корисність відповіді. Відповідь має бути достатньо інформативною і давати чітке пояснення для користувача, не упускаючи важливих аспектів запиту. Якщо чат-бот дає тільки часткову відповідь або не розкриває запитане питання в повній мірі, змістовність знижується.			
Стиль – цей критерій враховує, наскільки добре відповідь структурована і легка для розуміння. Висока оцінка за стиль означає, що відповідь чітка, без зайвих повторів і складних конструкцій. Якщо чат-бот відповідає не тією мовою, якою був задан запит, оцінка за стиль знижується.			

Питання	Відповідь ChatGPT	Відповідь Claude	Точність (ChatGPT), 0..100	Точніть (Claude), 0..100	Змістовність (ChatGPT), 0..100	Змістовність (Claude), 0..100	Стиль (ChatGPT), 0..100	Стиль (Claude), 0..100
1. Коли розпочалася підготовка фахівців з галузі "Комп'ютерні науки" в ДВНЗ "УДХТУ"?	Підготовка фахівців з галузі "Комп'ютерні науки" в ДВНЗ "УДХТУ" розпочалася в 1999 році. Спочатку...	Підготовка фахівців з галузі "Комп'ютерні науки" в ДВНЗ "УДХТУ" розпочалася у 1999 році...						
2. Який факультет здійснював підготовку фахівців з "Комп'ютерних наук" до 2009 року?	До 2009 року підготовка фахівців з "Комп'ютерних наук" здійснювалася на ...	According to the context provided, from 1999 to 2009, the training of specialists in "Computer Science"....						

Рис. 1. Анкета оцінювання відповідей

На рисунку 2 представлено порівняння точності відповідей двох моделей чат-ботів: ChatGPT і Claude. Відповідно до проведеного опитування, значна кількість студентів оцінила точність відповідей ChatGPT і Claude у діапазоні 81-100 балів, що вказує на високу якість роботи обох моделей з точки зору відповідності змісту базі знань. Водночас, можна помітити, що Claude має трохи більше оцінок у діапазонах 41-60 і 61-80, у порівнянні з ChatGPT, що свідчить про трохи більшу варіативність оцінок серед студентів. Загалом, більшість респондентів оцінили точність обох моделей високо, але ChatGPT отримав більше максимальних оцінок, що підкреслює його стабільність у наданні відповідей.

а.



б.

Рис. 2. Графіки точності: а – ChatGPT, б – Claude

а.

б.

Рис. 3. Графіки змістовності: а – ChatGPT, б – Claude

На рисунку 3 відображено порівняння змістовності відповідей моделей ChatGPT і Claude. Як видно, більшість оцінок студентів знаходиться у діапазоні 81-100 балів, що свідчить про те, що обидві моделі зазвичай надають розгорнуті, зрозумілі й корисні відповіді. Це особливо важливо для коректного розкриття тематики та задоволення інформаційних потреб користувачів.

У той же час спостерігається незначна різниця у розподілі оцінок в діапазонах 41-60 і 61-80: Claude має більше оцінок у цих категоріях, що може вказувати на інколи менш повне або детальне розкриття теми. Загалом, ChatGPT частіше отримував максимальні оцінки, що свідчить про його здатність краще структурувати та розгорнуто пояснювати відповіді для студентів.

На рисунку 4 відображено оцінки стилю відповідей ChatGPT та Claude. Зазначений критерій оцінює зрозумілість, структурованість і легкість для сприйняття відповідей моделей.

Переважна кількість студентів поставили обидвом моделям високі бали в діапазоні 81-100, що вказує на хорошу структуру й зрозумілість відповідей. Однак варто відзначити, що Claude має трохи ширший розподіл у діапазонах 61-80 і 41-60, що може свідчити про інколи менш чіткі або менш структуровані відповіді у порівнянні з ChatGPT. З іншого боку, ChatGPT демонструє стабільну тенденцію до отримання найвищих оцінок, що підкреслює його здатність адаптуватися до запитів користувачів і надавати лаконічні та чіткі відповіді.

а.

б.

Рис. 4. Графіки стилю: а - для чат-бота ChatGPT, б - для чат-бота Claude

Наукова новизна. Використання підходу з багатокритеріальним аналізом якості відповідей (точність, змістовність, стиль) дозволив оцінити ефективність використання бази знань у роботі моделей ChatGPT та Claude. Використання спеціалізованого контексту дає можливість адаптувати чат-боти до вузькопрофільних запитів, забезпечуючи їхню відповідність потребам користувачів у межах певної тематики.

Висновки. З отриманих даних було виявлено наступне:

- за критерієм точності середні оцінки ChatGPT демонструють дещо вищі результати, ніж Claude, особливо в інтервалі 81-100 балів, де більшість студентів поставили максимальні оцінки. Це свідчить про те, що модель ChatGPT краще впорається із завданням точного відтворення відповідей на базі наявної інформації.
- за критерієм змістовності результати ChatGPT і Claude виявилися подібними. В обох моделях спостерігається пік у вищому інтервалі оцінок (81-100 балів). Проте є відмінності у деталізації відповідей, де за певними питаннями Claude демонстрував глибше опрацювання змісту.
- за критерієм стилю ChatGPT отримав більш високі середні оцінки у порівнянні з Claude, що можна пояснити кращою структурованістю та чіткістю відповідей ChatGPT. Його відповіді часто мали зрозумілішу і легше сприйману форму.

Дослідження показало, що обидві моделі чат-ботів мають свої сильні сторони. ChatGPT переважає за точністю, тоді як Claude демонструє переваги у стилістичному та, частково, змістовному аспектах відповідей. Подальше дослідження може включати розширення вибірки оцінювачів, використання додаткових критеріїв оцінювання та аналіз контексту залежно від специфіки запитів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Introduction in LangChain. URL: https://python.langchain.com/docs/introduction/
2. Langchain. PyPI. URL: https://pypi.org/project/langchain/
3. Build a Retrieval Augmented Generation (RAG) App URL: https://python.langchain.com/docs/tutorials/rag/

УДК 004.415.3:681.6

В.В. Спірінцев[1], А.Л. Ширін[1], А.А. Бондаренко[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РОЗРОБКА ВЕБПРОЄКТІВ, АДАПТОВАНИХ ДО РІЗНИХ ПРИСТРОЇВ У ФРЕЙМВОРКУ BOOTSRAP

Анотація. В рамках опанування дисципліни Web-технології та Web-дизайн було розроблено вебпроєкт для вигаданої компанії таксі, адаптований до різних пристроїв із використанням CSS-фреймоворку Bootstrap. В роботі висвітлено основні переваги цього фреймоворку та наведено приклад використання його інструментів у розробці простих вебдодатків.

Ключові слова: *Bootstrap, CSS-фреймоворк, типографіка, шаблони, фронтенд, розробка, вебсайт, вебдодаток, вебпроєкт.*

Вступ. Bootstrap є одним із найпопулярніших CSS-фреймворків за останні п’ять років, що підтверджується його стабільно високим рівнем використання розробниками [1]. Цей показник, за даними статистичної агенції stateofcss.com у 2023 році склав 80,3% [2]. Це свідчить про його надійність та актуальність для розробників вебінтерфейсів, незважаючи на появу нових альтернатив (рис.1).

Рис. 1. Рейтинг використання CSS-фреймоворків (2019 – 2023 рр.) за даними статистичної агенції stateofcss.com

Основний зміст роботи. Bootstrap – це безкоштовний та відкритий фреймворк на основі HTML, CSS і JavaScript, призначений для швидкої розробки адаптивних вебсайтів та вебдодатків, який найчастіше використовують для створення інтерфейсів фронтенду. Завдяки готовим шаблонам і стилям для

типографіки, форм, кнопок, таблиць, навігації та інтегрованим JavaScript плагінам, Bootstrap значно спрощує процес верстки сайту.

Bootstrap має кілька ключових переваг. По-перше, його легко використовувати: навіть базових знань HTML і CSS достатньо для початку роботи. По-друге, фреймворк дозволяє швидко створювати якісну адаптивну верстку. Він також надає можливість налаштувати компоненти під конкретний проєкт, при цьому всі елементи виконані в єдиному стилі. Адаптивний CSS автоматично підлаштовується під різні пристрої – телефони, планшети та настільні комп'ютери. Крім того, Bootstrap сумісний із усіма сучасними браузерами, що забезпечує коректне відображення сайту в різних середовищах.

Ключовими елементами роботи Bootstrap є використання заздалегідь створених стилів та компонентів, які дозволяють легко створювати сучасні інтерфейси. Наприклад, розробник може використовувати стандартні класи для кнопок, форм, навігації та інших елементів, що значно скорочує час на верстку та забезпечує єдиний стиль. Також Bootstrap підтримує адаптивний дизайн, завдяки системі сіток (grid system), що дозволяє сторінкам автоматично підлаштовуватися під різні екрани (рис.2). Для розширення функціональності Bootstrap використовує JavaScript-плагіни, що додають інтерактивність без необхідності писати код з нуля.

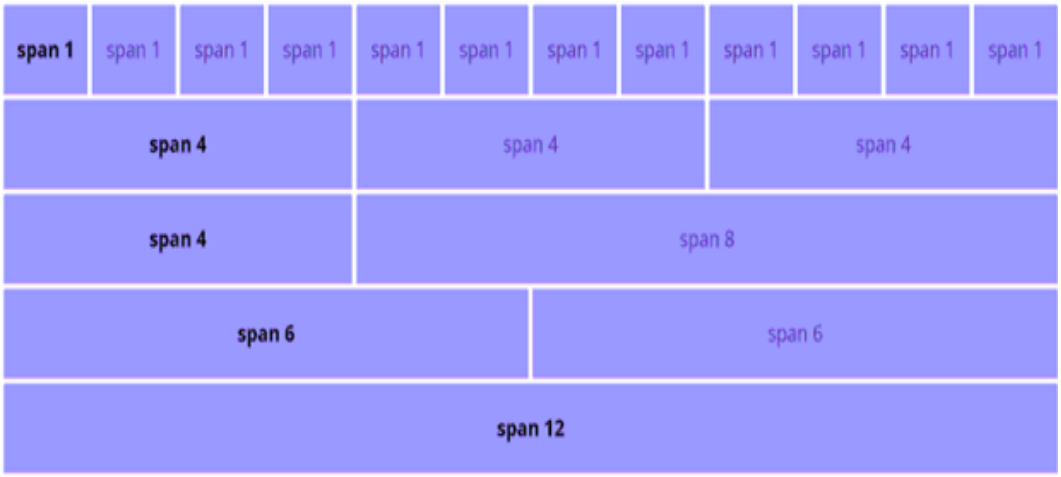


Рис. 2. Приклад використання системі сіток у Bootstrap

Процес створення вебсайту у Bootstrap починається з імпорту CSS та JavaScript бібліотек, після чого компоненти додаються до HTML (рис.3).

Наприклад, для створення адаптивної навігаційної панелі або модальних вікон можна використати готові класи та плагіни Bootstrap. Це дозволяє заощадити час і зусилля, зосередившись на функціональності, а не на деталях верстки. Для покращення продуктивності й гнучкості проєктів Bootstrap надає можливість кастомізації, що дозволяє налаштувати вигляд і поведінку компонентів, мінімізуючи код і розмір фінальних файлів. Зокрема, ви можете налаштувати сітку, зміну кольорів, типографіку та багато іншого.

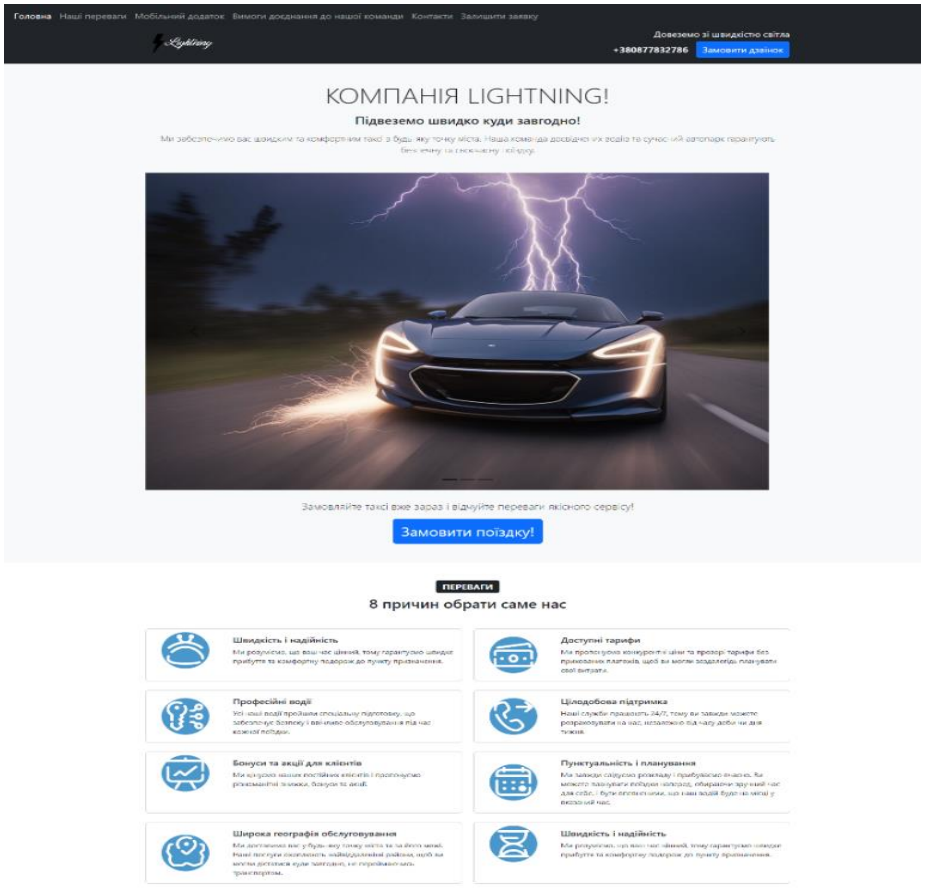


Рис. 3. Приклад розробленого сайту за допомогою Bootstrap

В якості прикладу використання CSS-фреймоворку Bootstrap було розроблено простий вебдодаток Lightning (рис.3), який адаптований під усі мобільні пристрої та планшети (рис.4).

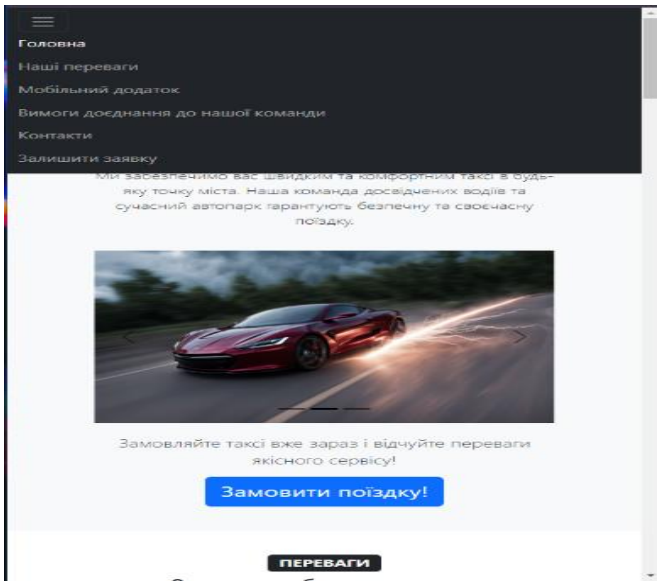


Рис. 4. Приклад адаптації сторінки

Висновки. Bootstrap ідеально підходить для швидкого створення адаптивних дизайнів завдяки готовим компонентам і системи сіток. Проте для великих проєктів з унікальним дизайном його стандартні стилі можуть стати обмеженням. У таких випадках доцільно комбінувати Bootstrap з кастомними стилями або іншими фреймворками.

ПЕРЕЛІК ПОСИЛАНЬ

1. Денисенко Андрій. 10 популярних CSS-фреймворків. 2022. [електронний ресурс]. URL: https://robotdreams.cc/uk/blog/423-10-populyarnih-css-freymvorkiv (дата звернення: 22.10.2024).
2. State of CSS 2023: CSS Frameworks. StateOfCSS-2023 [електронний ресурс]. URL: https://2023.stateofcss.com/en-US/css-frameworks/ (дата звернення: 22.10.2024).
3. Bootstrap documentation [електронний ресурс]. URL: https://getbootstrap.com/docs/5.3/getting-started/introduction/ (дата звернення: 22.10.2024).
4. What is Bootstrap? [електронний ресурс]. URL: https://itmaster.biz.ua/ programming/web-prohramuvannia/bootstrap.html (дата звернення: 22.10.2024).
5. State of CSS 2023: CSS Frameworks [електронний ресурс]. URL: https://2023.stateofcss.com/en-US/css-frameworks/ (дата звернення: 23.10.2024).

УДК 004.415.3:681.6e

Л.В.Кабак[1], Д.М. Мороз[1], О.О.Алексєєв[1]

[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

СТВОРЕННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ КЕРУВАННЯ ФІНАНСАМИ З ВИКОРИСТАННЯМ ШТУЧНОГО ІНТЕЛЕКТУ

Анотація. У роботі представлено розробку мобільного застосунку для керування фінансами з використанням штучного інтелекту, побудованого на фреймворку Flutter. У застосунку реалізовано основні функції для запису, редагування, видалення та аналізу фінансових транзакцій користувача, а також використано модель машинного навчання tflite, яка здатна самонавчатися та здійснювати аналіз витрат на основі Machine Learning у Firebase. Отримані результати можуть бути застосовані в розробці подібних фінансових застосунків для персонального використання та бізнесу.

***Ключові слова**: мобільний застосунок, управління фінансами, штучний інтелект, машинне навчання, Flutter, Firebase, транзакції, аналіз витрат.*

Вступ. У сучасному світі цифрових технологій мобільні застосунки для управління фінансами стають невід'ємною частиною щоденного життя багатьох людей. Розробка таких застосунків є складним процесом, який включає як базову функціональність для обліку фінансів, так і інноваційні методи аналізу даних, що базуються на штучному інтелекті та машинному навчанні. Ці технології

дозволяють здійснювати автоматизований аналіз витрат і надавати користувачам корисні рекомендації, що сприяє покращенню їх фінансових рішень [1].

Мобільний застосунок для фінансового управління має відповідати вимогам високої продуктивності, зручності використання та безпеки, оскільки він обробляє чутливу фінансову інформацію користувачів. Це вимагає ретельного підходу до вибору інструментів для розробки, забезпечення зберігання даних та підтримки інтерактивного інтерфейсу [2]. Використання хмарних сервісів для зберігання даних дозволяє значно підвищити гнучкість і масштабованість застосунків, що є особливо важливим для забезпечення стабільної роботи при великій кількості користувачів [3].

Важливим аспектом є також інтеграція самонавчаючих моделей, які здатні покращувати аналіз фінансових даних на основі накопиченого досвіду, що забезпечує гнучкість системи та підвищує її цінність для користувачів. Ці моделі самостійно адаптуються до поведінки користувача, надаючи персоналізовані рекомендації, що сприяє більш відповідальному фінансовому плануванню [4]. Отримані результати й досвід розробки можуть бути корисними для створення майбутніх фінансових застосунків, які здатні ефективно використовувати штучний інтелект для покращення користувацького досвіду та оптимізації управління фінансами.

Постановка задачі. Основна задача роботи полягає у створенні мобільного застосунку для керування фінансами, який підтримує введення, редагування та видалення транзакцій, а також здійснює аналіз витрат користувача з використанням штучного інтелекту.

Основний зміст роботи. У роботі детально розглянуто ключові етапи розробки мобільного застосунку для управління фінансами, з акцентом на інтеграцію штучного інтелекту, ефективне управління станом та забезпечення зручності для користувача. Під час планування розробки був проведений аналіз вимог та можливої архітектури, яка буде відповідати всім вимогам. Щоб швидко та ефективно створити додаток було використано фрейворк Flutter, який створенний для розробки мобільних проєктів, саме те, що було потрібно. Для зберігання даних користувача самим оптимальним рішенням було використання БД SQLite, а в якості інтеграції штучного інтелекту та машинного навчання – Firebase. В поєднані ці інструменти дали змогу створити якісний додаток, який досить легкий у використанні, дає поради користувачеві стосовно витрат, ґрунтуючись на його операціях.Під час роботи було використано наступне:

- Flutter – фреймворк для розробки мобільних застосунків
- SQLite - полегшена реляційна система керування базами даних
- Firebase - платформа для Backend-as-a-Service (BaaS)
- Python – мова програмування, яку використував для написання self-learning моделі
- TensorFlowLite - бібліотека програмного забезпечення з відкритим кодом для машинного навчання та штучного інтелекту

- Figma – веб-застосунок для створення дизайну застосунку

Для розробки мобільного застосунку було використано фреймворк Flutter, який надає інструменти для створення кросплатформених застосунків з високою продуктивністю. Для зберігання даних використано SQLite, що є полегшеною реляційною системою керування базами даних. Firebase використано як Backend-as-a-Service (BaaS) для забезпечення хмарного зберігання даних та автентифікації. Для розробки самонавчаючоїся моделі було використано мову програмування Python разом з бібліотекою TensorFlowLite, що дозволило інтегрувати функціонал машинного навчання та штучного інтелекту. Дизайн інтерфейсу розроблено в Figma, що забезпечило зручність у створенні та налаштуванні UI/UX елементів.

Наукова новизна. Наукова новизна роботи полягає у розробці мобільного застосунку для керування фінансами, який використовує самонавчаючуся модель tflite, інтегровану з Firebase для аналізу витрат користувача. Такий підхід дозволяє застосунку самостійно покращувати результати аналізу та адаптуватися до індивідуальних фінансових звичок користувача.

Висновки. У результаті роботи було розроблено мобільний застосунок для керування фінансами, який включає основні функції для введення, редагування, видалення та аналізу фінансових транзакцій, а також використовує штучний інтелект для аналізу витрат. Проведене дослідження дозволяє розробникам мобільних застосунків зрозуміти, як інтеграція машинного навчання може покращити функціональність та користувацький досвід у фінансових застосунках. Отримані знання можуть бути корисними в освітніх програмах для вивчення принципів використання штучного інтелекту в мобільних застосунках. Результати роботи також мають практичну цінність для компаній та розробників, які прагнуть підвищити ефективність своїх продуктів для управління фінансами.

ПЕРЕЛІК ПОСИЛАНЬ

1.Smith, J. "Mobile Applications in Financial Management: Trends and Challenges." Journal of Financial Technologies, 2022.
2.Brown, T. "Security in Fintech Applications." International Journal of Information Security, 2021.
3.Davis, L. "Cloud Services in Mobile App Development." Journal of Cloud Computing, 2020.
4.Johnson, M. "Machine Learning for Personal Finance Apps." Artificial Intelligence in Business, 2023.

УДК 519.6:534.2:004.94

В.В. Спірінцев[1], А.Л. Ширін[1], А.Р. Курдюкова[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ОПТИМІЗАЦІЯ СТВОРЕННЯ МАКЕТІВ У FIGMA. ЕФЕКТИВНІ ІНСТРУМЕНТИ ТА ПЛАГІНИ ДЛЯ WEB-ДИЗАЙНУ

Анотація. У роботі розглянуті основні інструменти і плагіни вебсервісу Figma для спрощення та покращення користувацького інтерфейсу при розробці дизайну вебсторінок. Досліджено роль Design System у проектуванні дизайн-макетів, проведено аналіз методології Atomic Design та наведено приклади застосування цих підходів у Figma.

Ключові слова: *Figma, плагін, вебдизайн, інтерфейс, component, auto layout, design system, atomic design, variables plugi.*

Вступ. У сучасному світі веброзробка стала важливим аспектом для будь-якої компанії. Незалежно від її розміру чи сфери діяльності, інтернет-присутність для компанії – це вже не просто додатковий плюс, а критична необхідність. Це не тільки підвищує обізнаність щодо діяльності компанії, але й допомагає брендам залучити нових клієнтів та збільшити продажі через цифрові канали. Перед початком розробки важливо розуміти, що якісний дизайн є ключем до успіху. Перше враження користувачів формується за секунди: невдалий дизайн може викликати сумніви, а застарілий вигляд – знизити продажі. Сучасний та функціональний дизайн підвищує довіру до бренду та залучає користувачів з першого кліку. Проте розробка якісного дизайну web-сторінок є складним і трудомістким процесом.

Основний зміст роботи. Вебдизайнери часто витрачають багато часу на створення макетів, працюючи над деталями інтерфейсу, що може сповільнювати запуск проекту. Ця проблема посилюється, коли відсутні зручні інструменти, які дозволяють ефективно впорядковувати процес. Серед безлічі програмних рішень для дизайну Figma виділяється, як інструмент, який не лише безкоштовний для початку роботи, але й пропонує широкий набір функцій і плагінів для оптимізації роботи. Основні інструменти та можливості ресурсу Figma наведено нижче.

Component (*укр. компонент*) – це елемент дизайну, який можна використовувати повторно. Створивши компонент, можна вставляти його в різні частини проекту, а зміни, внесені до основного компонента, автоматично відображатимуться у всіх його копіях. Основними функціями компоненту у Figma є:

Variants – створення різних станів одного компонента, налаштування властивостей для різних станів і легкий перехід між варіантами;

Instance Syncing – автоматичне оновлення всіх екземплярів при зміні елементів базового компонента;

Constraints – забезпечують адаптивність компонентів при зміні розміру фрейму, дозволяючи налаштовувати прив'язку елементів до країв фрейму та визначають, як вони поводитимуться при зміні розміру.

Auto layout (*укр. автоматична верстка*) – автоматизує розміщення елементів, дозволяючи створювати адаптивні макети, які легко підлаштовуються під різні розміри екранів. Завдяки інструменту Auto layout, елементи в дизайні автоматично вирівнюються, масштабуються та підтримують структуру, що забезпечує зручність у налаштуванні та зміні інтерфейсу. Щоб реалізівати цю функцію у Figma потрібно виділити об’єкти, а потім натиснути комбінацію клавіш Shift + A або кнопку «Auto layout» з контекстного меню. В Auto layout доступні такі основні функції:

Direction – горизонтальна або вертикальна орієнтація елементів;

Spacing between items – інтервал між елементами;

Padding – область всередині контейнера;

Alignment – розташування елементів (або по центру, або зліва, або справа);

Resizing – елементи можуть змінювати свої характеристики з твердим, обмеженим контентом або заповнюючи контейнер.

Design System (*укр. Система дизайну*) – це структурований набір стандартів, компонентів і правил, що забезпечують узгодженість зовнішнього вигляду користувацького інтерфейсу у цифрових продуктах. Основою цієї системи є методологія атомного проектування (*англ. Atomic Design*), поляризована Бредом Фростом у 2013 році. Вона передбачає поділ інтерфейсу на менші частини (атоми), такі як кнопки, поля введення та іконки. Вони комбінуються в більш складні компоненти, відомі як молекули. Наприклад, форма, що складається з кнопки та поля введення, є молекулою. Ці молекули формують організми, які можуть бути більш складними інтерфейсами, такими як картки або навігаційні панелі. Організми можуть бути використані для створення шаблонів, які, зрештою, формують певні сторінки.

Існує поширений міф, що системи дизайну пригнічують естетику, обмежуючи креативність дизайнерів і забезпечуючи загальну однорідність дизайну. Це не є правдою, бо такі системи підтримують дизайнерів, знаходячи шаблони, що повторюються, звільняючи їх від зайвої роботи, для вирішення інших завдань. Використовуючи систему дизайну, дизайнери можуть повторно використовувати узгоджені компоненти, налаштовувати варіанти для плавного перемикання між режимами та розмірами екрана, і все це без необхідності копіювати та вставляти ті самі проекти знову і знову. Оновлення можна робити в одному місці та поширювати по всій системі, залишаючи всіх на одній сторінці.

Важливо наголосити, що система дизайну може мати свої виклики, наприклад, необхідність постійного обслуговування та оновлення компонентів. Незважаючи на це, переваги, які вона надає, такі як оптимізація робочого процесу, підвищення узгодженості та поліпшення користувацького інтерфейсу, значно переважають над можливими недоліками.

Режим прототипування. Цей режим дозволяє дизайнерам створювати інтерактивні моделі своїх макетів, щоб показати, як буде працювати кінцевий продукт. Цей інструмент допомагає не просто бачити екрани окремо, а й налаштовувати переходи, анімації та взаємодії між ними, що значно покращує візуалізацію. При створенні анімацій у Figma важливо використовувати компоненти. Це дозволяє задавати різні стани елементів – натискання правою чи лівою кнопкою мишки або наведення на елемент. Наприклад, якщо ви анімуєте шапку макету на головному компоненті, ці анімації автоматично оновлюються на всіх копіях сторінок макету, що означає можливість переходити на інші сторінки, змінивши лише головний компонент.

Зі зростанням досвіду дизайнера, розмір проектів збільшується, в анімаційних потоках стає складніше орієнтуватися, особливо при переході з режиму прототипування у режим дизайну, стрілки зникають з поля зору, а було б дуже зручно бачити всю картину взагалі. Для вирішення цієї проблеми існує плагін *ProToFlow*, який автоматично відновлює візуальні зв'язки між фреймами у макеті.

Не секрет, що web-дизайнери часто знаходять натхнення з робіт колег. Для цієї мети існує плагін *html.to.design*, який за допомогою наданого посилання на сайт, автоматично будує макет сторінки та компоненти до неї у форматі HTML.

Висновки. Зручні інструменти Figma дозволяють ефективно впорядковувати процес розробки дизайну вебпроєктів. Платформа працює повністю в браузері, що робить її доступною з будь-якого пристрою або операційної системи, не вимагаючи встановлення додаткового програмного забезпечення, а використання широкого набору функцій, плагінів та компонентів дозволяє стандартизувати та узгоджувати дизайн у великих проектах. Результати досліджень підкреслюють потужність web-сервісу Figma, який надає безліч зручних інструментів для економії ресурсів та часу.

ПЕРЕЛІК ПОСИЛАНЬ

1. TD Sunshine. Figma Components 101 | Learn about Variants and Components Properties | Figma tutorial, 2024. [електроний ресурс] URL: https://www.youtube.com/watch?v=JDHHRN5kNU8.
2. Arnau Ros. Build a Complete Figma Design System (Freelance Course Preview), 2024. [електроний ресурс] URL: https://www.youtube.com/watch?v=w-r6quQx0zA
3. DesignSpo. 9 FREE Plugins For Figma Designers! (Best Figma Plugins 2024), 2023. [електроний ресурс] URL: https://www.youtube.com/watch?v=SBhcN1gYQKA
4. Figma and Responsiveness: Adaptive Design in Practice. *No-code and low-code mobile app builder for IOS and Android*. [електроний ресурс] URL: https://www.moxly.io/blog/figma-and-responsiveness-adaptive-design-in-practice#:~:text=Fundamentals%20of%20Adaptive%20Design%20in%20Figma%20Responsive%20Grids,that%20should%20adapt%20together%20to%20different%20screen%20sizes.
5. DesignWithArash. Responsive Design in Figma Using Breakpoints, 2022. [електроний ресурс] URL: https://www.youtube.com/watch?v=c0ZTaDMGb20 (date of access: 01.11.2024).

УДК 004.9

М.О. Ткач[1], О.О.Козорог[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РОЗРОБКА GNSS RTK СИСТЕМИ ДЛЯ СІЛЬСЬКОГОСПОДАРСЬКОЇ ПРОМИСЛОВОСТІ

Анотація. Дослідження присвячене розробці GNSS RTK системи, оптимізованої для застосування в сільськогосподарській промисловості. Запропонована система забезпечує високу точність навігації та передачі поправок завдяки використанню резервного каналу зв'язку та алгоритмів автоматичного перемикання. Основна увага приділяється модульній архітектурі, легкому шифруванню даних методом XOR та зниженню вартості впровадження.

***Ключові слова**: GNSS RTK, точне землеробство, резервний канал, шифрування XOR, RTCM3, LoRa, ZigBee, модульна архітектура.*

Вступ. Зростання використання точного землеробства ставить перед агропромисловим сектором нові виклики. Одним із ключових аспектів є забезпечення високої точності навігації сільськогосподарської техніки. Це досягається за допомогою GNSS RTK технологій, які дозволяють отримати позицію з сантиметровою точністю. Проте проблема полягає у залежності від стабільності сигналу GNSS, який часто втрачається через перешкоди або погодні умови. Це створює ризик неточностей, що впливає на ефективність землеробства. Метою даної роботи є розробка GNSS RTK системи з резервним каналом зв’язку, яка забезпечує стабільність та безпеку передачі даних навіть у складних умовах.

Постановка задачі. Метою даного дослідження є створення GNSS RTK системи з шифруванням данних, яка включає резервний канал зв’язку та алгоритм автоматичного перемикання між основним і резервним каналами. Для досягнення мети необхідно вирішити такі завдання:

- Проаналізувати існуючі проблеми точного землеробства та способи їх вирішення.
- Визначити основні, мінімально необхідні компоненти системи
- Реалізувати легкий і швидкий метод шифрування XOR для захисту даних.
- Спроектувати модульну архітектуру, яка дозволить легко масштабувати систему.
- Виконати тестування і оцінку ефективності системи в реальних умовах.
- Провести аналіз цінових показників та різниці між даною роботою та повноцінними комерційними моделями на ринку

Основний зміст роботи. Для створення GNSS RTK системи використовувалися наступні інструменти:
– Модулі LoRa для основного каналу передачі даних на далекі відстані.
– CDM26-u-center як програмне забезпечення для обробки GNSS поправок.

– GNSS приемник ZED-F9P u-blox
– МК для шифрування даних методом XOR для забезпечення базової безпеки передачі даних.

Запропонована GNSS RTK система забезпечує високу точність навігації та стабільність зв'язку завдяки резервному каналу. Основний канал на базі LoRa передає поправки RTCM3, а резервний канал автоматично активується у разі втрати сигналу. Це особливо важливо для сільськогосподарських задач, де переривання роботи може призвести до значних втрат.

Безперервність передачі даних завдяки автоматичному перемиканню між каналами. - Використання швидкого XOR - шифрування з низькими ресурсними затратами. Стійкість до умов високої вологості, перешкод і складної місцевості.

Розробка системи складалася з кількох етапів:

- Проектування основного каналу передачі даних на базі LoRa.
- Розробка резервного каналу зв'язку.
- Інтеграція алгоритму перемикання між каналами.
- Тестування шифрування XOR та загальної працездатності системи.

Приймач GNSS отримує сигнали від кількох супутників і визначає своє місцезнаходження. Дані передаються на базову станцію для обробки. Базова станція виконує корекцію отриманих координат, використовуючи алгоритми обчислення поправок. Вона враховує затримки в атмосфері та інші фактори, що викликають похибки. Базова станція передає корекційні дані через основний канал зв'язку (LoRa). У разі збоїв автоматично активується резервний канал зв'язку, що забезпечує стабільність роботи системи. TKLib, встановлений на кінцевому пристрої, отримує корекційні дані, обчислює уточнені координати та передає їх на сільськогосподарську техніку. Перед відправленням дані шифруються за допомогою XOR для забезпечення конфіденційності.

Математична модель розробки GNSS RTK системи полягає в інтеграції апаратно-програмного комплексу, який забезпечує передачу корекційних даних через основний та резервний канали. Для забезпечення високої надійності та точності використовується математична модель, яка описує процеси передачі, обробки та корекції даних. Основні етапи розробки базуються на багатокритеріальному методі оцінки оптимальних параметрів компонентів системи.

- **X** – множина вхідних параметрів, що містить сигнали, отримані GNSS-приймачем, а також корекційні дані від базової станції.
- **Y** – множина вихідних параметрів, що включає скориговані координати техніки.
- **Z** – множина зовнішніх впливів, таких як радіоперешкоди, втрати сигналу, атмосферні ефекти.
- **A** – оператор, що реалізує алгоритми обробки сигналів, корекції даних і перемикання каналів.

Математично модель виглядає так:

$$Y = A(X, Z)$$

Ключові сутності моделі:

Корекційні дані (C) – інформація про поправки, що генеруються базовою станцією:

$$C = \{c_1, c_2, \dots, c_{n(C)}\}$$

де $n(C)$– кількість поправок у потоці даних.

Модулі передачі даних (T) – пристрої (LoRa), що передають корекції між базовою станцією та GNSS-приймачем:

$$T = \{t_1, t_2, \dots, t_{n(T)}\}$$

де $n(T)$– кількість типів модулів у системі.

Помилки передачі (EE) – множина можливих помилок у передачі даних, наприклад втрати сигналу або затримки:

$$E = \{e_1, e_2, \dots, e_{n(E)}\}$$

де $n(E)$– кількість можливих помилок.

Зв’язки між класами

- G– зв'язок між сигналом, отриманим GNSS-приймачем, і модулем обробки сигналу.
- P – зв'язок між модулем передачі та перешкодами, що впливають на сигнал.

Наукова новизна розробки полягає у застосування багатокритеріального аналізу дозволяє вибирати оптимальні параметри для роботи системи залежно від зовнішніх умов. Інтеграція резервного каналу з метою безперервності передачі даних навіть у випадку втрати основного сигналу. Використання XOR шифрування для мінімізації затримки при передачі, забезпечуючи захист даних у реальному часі. Розробка GNSS RTK системи для сільськогосподарської промисловості враховує всі критичні аспекти, забезпечуючи надійність, точність та масштабованість у використанні.

Висновки. У результаті проведених досліджень і розробок створено ефективну та надійну GNSS RTK систему, призначену для точного землеробства. Розроблена GNSS RTK система демонструє високу точність позиціонування (до 1-2 см), надійність зв’язку навіть у складних умовах і доступність для впровадження в сільськогосподарській промисловості. Вона дозволяє автоматизувати ключові процеси, такі як точне висівання, внесення добрив і управління агротехнікою, що сприяє підвищенню ефективності та зменшенню витрат. Запропонована система також є перспективною для масштабування,

інтеграції нових каналів зв'язку або функцій моніторингу, що робить її адаптивним рішенням для широкого спектра фермерських господарств.

ПЕРЕЛІК ПОСИЛАНЬ

1. Hofmann-Wellenhof, B., Lichtenegger, H., & Collins, J. Global Positioning System: Theory and Practice / Springer – Berlin, Heidelberg, 2001. – 382 с.
2. Misra, P., & Enge, P. Global Positioning System: Signals, Measurements, and Performance / Ganga-Jamuna Press – Lincoln, Massachusetts, 2006. – 411 с.
3. Farrell, J. A., & Barth, M. The Global Positioning System and Inertial Navigation / McGraw-Hill – New York, 2008. – 496 с.
4. Langley, R. B. RTK GPS / Geomatics World – London, 1998. – 112 с.
5. Азаров Ю.В. Комп'ютерна схемотехніка / Видавництво НТУУ «КПІ», Київ, 2018. – 305 с.
6. Habr. Використання RTK для точного позиціонування // https://habr.com/ru/articles/648247/
7 SystemNet. Скільки каналів потрібно приймачу GNSS? 184 канали X5R – більше ніж достатньо // https://systemnet.com.ua/skilki-kanaliv-potribno-prijmachu-gnss-184-kanali-x5r-bilshe-nizh-dostatno/

УДК 004.415.3:681.6

В.В. Спірінцев[1], В.В. Веселов[1]

[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ВИКОРИСТАННЯ JETPACK COMPOSE В ПОРІВНЯННІ З МОВОЮ XML ПРИ РОЗРОБЦІ МОБІЛЬНИХ ДОДАТКІВ ДЛЯ ОС ANDROID

Анотація. Розглянуто ключові аспекти дослідження ефективності технологій XML та Jetpack Compose для розробки графічного інтерфейсу мобільних додатків для ОС Android. В роботі проаналізовано поточний стан розробки графічних інтерфейсів для Android-додатків, здійснено порівняльний аналіз технологій XML та Jetpack Compose, а також розглянуто їх особливості. Особлива увага приділяється здійсненню оцінки ефективності використання технологій XML та Jetpack Compose для розробки інтерфейсу мобільних додатків, що дозволяє здійснити раціональний вибір технологій для створення якомога ефективніших додатків за використанням ресурсів - CPU та пам'яті, розміром, часом запуску додатків та рендерингу кадрів, кількістю кадрів в секунду (FPS), та при роботі з навантаженням.

Ключові слова: *Jetpack Compose, XML, Android, мобільні додатки, графічний інтерфейс, ефективність, розробка графічного інтерфейсу.*

Вступ. В сучасному світі люди використовують мобільні девайси частіше ніж комп'ютери [1]. Варто зазначити, що кількість користувачів мобільних девайсів постійно зростає. Це підтверджує, що розробка програмного забезпечення для мобільних девайсів є досить важливим напрямком сьогодення. Крім того, кількість пристроїв з ОС Android перевищує кількість пристроїв з будь-якою іншою операційною системою [2]. Графічний інтерфейс є однією з

найважливіших складових будь-якого Android-додатку. Насьогодні існує дві основні технології для розробки графічного інтерфейсу для додатків ОС Android, а саме: XML та Jetpack Compose. Дослідження ефективності використання зазначених технологій є досить важливим питанням, оскільки від вибору технології для створення інтерфейсу користувача залежить ефективність роботи всього додатку, що, в свою чергу, впливає на загальний успіх додатку серед користувачів.

Постановка задачі. Мета даного дослідження полягає у порівнянні ефективності використання технологій XML та Jetpack Compose для розробки графічного інтерфейсу мобільних додатків для ОС Android, що дозволить підвищити ефективність прийняття рішень керівництвом IT-компаній при розробці інтерфейсу користувача додатків для даної операційної системи. Серед основних задач, які необхідно виконати для того, щоб досягнути мету дослідження, можна виділити наступні: здійснити аналіз особливостей інструменту Jetpack Compose та мови XML; дослідити ефективність використання вказаних технологій шляхом тестування двох однакових мобільних додатків для ОС Android, один з яких використовує XML, а інший - Jetpack Compose, за наступними параметрами: використання ресурсів - CPU та пам’ять, час запуску додатку та час рендерингу кадрів, кількість кадрів в секунду (FPS), навантаження з різною кількістю запитів, розмір додатку, наявність помилок та збоїв; проаналізувати та порівняти отримані результати тестування по кожному додатку, а також зробити висновки про ефективність використання інструменту Jetpack Compose в порівнянні з мовою XML.

Основний зміст роботи. На перших етапах розробки інтерфейсу користувача мобільних додатків для ОС Android використовувалась лише мова розмітки XML. Такий підхід є досить простим, а також дозволяє розділити логіку додатку від графічного інтерфейсу, проте код інтерфейсу користувача стає сильно громіздким в проектах зі складним UI. Згодом було розроблено інструмент Jetpack Compose, який надає декларативний підхід до дизайну UI, де інтерфейс описується в термінах станів.

Насьогодні мова розмітки XML зустрічається у відносно старих проектах для ОС Android, а також у всіх проектах, які написані мовою програмування Java. Це пов’язано з тим, що Jetpack Compose можна використовувати лише в проектах, написаних мовою Kotlin. Тому він застосовується в більшості випадків у сучасних проектах. Крім того, в деяких старих проектах відбувається міграція від XML до інструменту Jetpack Compose.

XML (eXtensible Markup Language) – це набір правил для визначення семантичних тегів, які розділяють документ на частини та визначають різні частини документа. XML - це розширювана мова мета-розмітки, яка визначає синтаксис, що використовується для визначення інших предметно-спеціальних, семантичних, структурованих мов розмітки [3, с. 3]. При розробці мобільних додатків для ОС Android, XML використовується для визначення макетів екранів додатку, а також елементів графічного інтерфейсу.

Jetpack Compose - це рекомендований сучасний набір інструментів Android для створення нативного інтерфейсу користувача. Він спрощує та прискорює розробку інтерфейсу користувача для ОС Android [4]. Jetpack Compose надає декларативний API, що робить можливим відтворювати інтерфейс користувача без обов'язкової зміни зовнішніх інтерфейсів. Jetpack Compose використовує декларативну модель інтерфейсу, що дозволяє уникнути складності ручного оновлення ієрархії представлень станів.

Серед основних особливостей XML в розробці додатків для ОС Android, варто зазначити наступні. По-перше, кожний макет екрану додатку зберігається в окремому XML-файлі. Очевидно, що при створенні складного графічного інтерфейсу в цих файлах утворюється значна вкладеність, що робить код важким для сприйняття та підтримки. По-друге, розмітка мовою XML відокремлюється від коду, написаного мовою програмування Kotlin або Java, що, в свою чергу, використовує розмітку графічного інтерфейсу, посилаючись на відповідні XML-файли. Це дозволяє відокремити інтерфейс від логіки додатку, що підвищує читабельність та спрощує розуміння коду. Крім того, XML може використовуватись як з мовою програмування Java, так і з Kotlin.

Щодо особливостей Jetpack Compose, то варто відзначити наступні. Jetpack Compose надає можливість розробляти графічний інтерфейс шляхом винесення графічних елементів у окремі функції, в яких визначаються необхідні атрибути для кожного графічного компонента. Це дозволяє викликати ці функції з інших функцій, що полегшує сприйняття коду на екранах додатку з складним графічним інтерфейсом, а також вирішує проблему зі значною вкладеністю, присутньою в XML. Такий підхід дає можливість повторно використовувати визначені функції, що зменшує повторюваність коду. Код, написаний з використанням Jetpack Compose, є значно лаконічнішим, ніж код XML. При використанні Jetpack Compose весь код пишеться мовою програмування Kotlin, а відокремлення логіки додатку від графічного інтерфейсу дещо розмивається.

Кастомізація інтерфейсу користувача в XML здійснюється шляхом використання призначених для цього атрибутів. Також існують спеціальні контейнери, наприклад, LinearLayout (лінійне розташування компонентів по горизонталі або вертикалі), ConstraitLayout (розташування компонентів, вказуючи обмеження), в які можна вкладати графічні компоненти, а також налаштовувати атрибути цих контейнерів. В XML контейнери та графічні компоненти утворюють ієрархічну структуру.

Кастомізація графічного інтерфейсу в Jetpack Compose виконується за допомогою використання спеціальних модифікаторів. Вони дозволяють налаштувати вирівнювання, розмір, відступ, колір фону, а також інші властивості графічних елементів. Для розміщення графічних елементів використовуються спеціальні функції, наприклад, Column та Row – для розміщення графічних компонентів вертикально та горизонтально відповідно. В Jetpack Compose всі графічні елементи визначаються у вигляді Composable-функцій, структура графічних компонентів має ієрархічний вигляд.

В ході тестування двох однакових додатків, проте один – з використанням XML, а інший – з Jetpack Compose, було визначено, що розмір додатку з XML становить 7.22 мегабайтів, а розмір додатку з Jetpack Compose – 11.64 мегабайтів. Таким чином, додаток з XML займає менше пам’яті пристрою, ніж додаток з використанням Jetpack Compose. Ніяких помилок та збоїв у функціонуванні обох додатків не було виявлено.

Для об’єктивності отриманих результатів в ході тестування ефективності додатків, тестування за кожним параметром виконується три рази з подальшим розрахунком середнього значення. Тестування ефективності використання ресурсів CPU та пам’яті виконується за допомогою використання інструмента Profiler в Android Studio на кожному кроці взаємодії з пристроєм та графічними компонентами інтерфейсу. Для тестування ефективності використання ресурсів було виділено такі кроки:

- початковий стан графічного інтерфейсу додатку;
- натискання кнопки для пошуку;
- натискання на текстове поле для вводу;
- введення тексту в текстове поле;
- натискання кнопки для пошуку після введення тексту;
- вибір елементу із списку та перехід на екран з детальною інформацією;
- додавання елементу в список обраних;
- видалення елементу із списку обраних;
- повернення на попередній екран за допомогою кнопки «На попередній екран»;
- перехід на екран із списком обраних;
- повернення на попередній екран за допомогою системної кнопки пристрою;
- переворот екрану та повторне створення компоненту Activity.

Отримані результати тестування ефективності використання ресурсів представлені в таблиці 1.

Таким чином, з отриманих результатів можна спостерігати, що в світлій темі додатку Jetpack Compose використовує CPU ефективніше, проте потребує більше пам’яті, ніж XML. Крім того, в темній темі додатку XML використовує ефективніше і пам’ять, і CPU.

Для вимірювання часу запуску додатків та часу тривалості кадрів у мілісекундах доцільно використати інструмент Macrobenchmark. Отримані результати вимірювання для обох тем додатку зображені на рис. 1.

З отриманих результатів можна зробити висновок, що для обох додатків час запуску додатку та час тривалості кадрів приймає більші значення в темній темі додатку. Крім того, запуск додатку з використанням XML займає менше часу, ніж запуск додатку з Jetpack Compose в обох темах. Час тривалості кадрів приймає більші значення на перших ітераціях з використанням XML, а на останніх ітераціях – Jetpack Compose.

Таблиця 1.

Результати порівняння ефективності використання ресурсів додатків

№ кроку	1	2	3	4	5	6	7	8	9	10	11	12
XML (Світла тема), середні значення												
CPU, %	7	3,7	3,7	1,3	1,3	1,3	1,3	1,3	1,3	0	2,3	10,7
Пам’ять, МБ	116,7	106,5	107,7	106,7	106,7	108,5	108,5	108,6	107,1	109,1	108,3	116,2
XML (Темна тема), середні значення												
CPU, %	7,7	1,7	0	0,3	0	0	0	0	0	0	0	7,7
Пам’ять, МБ	114,6	116,3	116,5	116	116,2	117,1	117,1	117,2	124,4	125,7	113,2	108,9
Jetpack Compose (Світла тема), середні значення												
CPU, %	3,3	0	0,3	0	0	0	0	1	0	0	1	9,7
Пам’ять, МБ	163,5	161,8	162	162,5	163,1	171	171,1	171,2	161,9	162,6	170,7	68
Jetpack Compose (Темна тема), середні значення												
CPU, %	10,7	0	0	0	0,3	0	0	0	0		0	13
Пам’ять, МБ	175,6	174,8	175,6	175,9	176,4	183,6	183,2	183,2	176,5	177,7	181,9	180,1

Рис. 1. Результати вимірювання часу запуску та тривалості кадрів

Вимірювання кількості кадрів у секунду (FPS) здійснюється по аналогічним крокам, як і при тестуванні використання ресурсів. Результати наведені в таблиці 2.

Таблиця 2.

Результати вимірювання кількості кадрів в секунду

№ кроку	1	2	3	4	5	6	7	8	9	10	11	12
XML (Світла тема), середні значення												
FPS	60,1	60,1	60	60	60	60,1	60	60,2	60	60	60	60,1
XML (Темна тема), середні значення												
FPS	60,1	60	60	60	60	60, 1	60	60,1	60	60	60	60
Jetpack Compose (Світла тема), середні значення												
FPS	60	60	60,1	60,1	60,1	60	60	60	60,1	60	60	60
Jetpack Compose (Темна тема), середні значення												
FPS	60,1	60	60	60	60,2	60	60,1	60,1	59,3	60	60	60,1

З результатів видно, що різниця у кількості кадрів в секунду не значна. Додаток з XML має більшу кількість кадрів в секунду в обох темах додатку, проте ненабагато.

Для тестування навантаження вимірюється час до початку запиту, здійснюється натискання на кнопку, вимірюється час після запиту та знаходиться час, витрачений на здійснення запиту. Таким чином, визначаються сумарний та середній час виконання запитів. Функціонування додатку доцільно тестувати з різною кількістю запитів, а саме 1, 5, 10 та 20 запитів. Результати тестування навантаження показані на рис. 2.

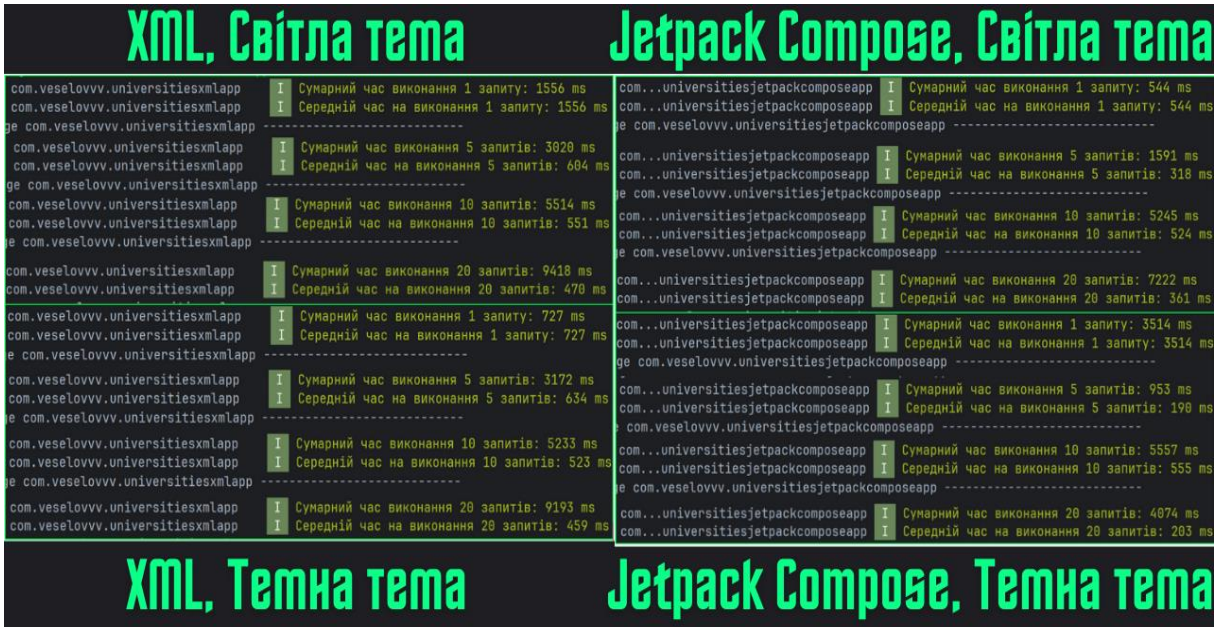


Рис. 2. Результати тестування навантаження

З отриманих результатів можна зробити висновок, що при навантаженні в світлій темі додатку, запити в додатку з Jetpack Compose виконуються швидше. Крім того, в темній темі додатку при навантаженні з кількістю запитів 5 та 20, запити виконуються швидше в додатку з Jetpack Compose, а з кількістю запитів 1 та 10 – в додатку з XML.

Наукова новизна. Новизна досліджень полягає у комплексному рішенні щодо оцінки ефективності використання технологій XML та Jetpack Compose при розробці графічного інтерфейсу для мобільних додатків ОС Android. Результати дослідження надають комплексний порівняльний аналіз технологій XML та Jetpack Compose, враховуючи особливості функціонування інтерфейсу користувача в світлій та темній темах додатку, що дозволяє здійснити раціональний вибір технологій для створення якомога ефективнішого графічного інтерфейсу мобільних додатків для ОС Android.

Висновки. В даній роботі було здійснено порівняльний аналіз ефективності технологій XML та Jetpack Compose. Результати дослідження показують, що мова XML є ефективнішою в порівнянні з Jetpack Compose за такими чотирма параметрами: розмір додатку, використання ресурсів - CPU та пам’яті, час запуску додатку та рендерингу кадрів, кількість кадрів в секунду (FPS), але менш ефективною при тестуванні навантаження. Таким чином, з огляду на ефективність за даними параметрами, доцільніше використовувати XML для розробки більш ефективних та якісних додатків.

ПЕРЕЛІК ПОСИЛАНЬ

1. GlobalStats. Desktop vs Mobile vs Tablet Market Share Worldwide - August 2024. [Electronic resourse]. – Access mode: https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide– Title from the screen.
2. GlobalStats. Operating System Market Share Worldwide - August 2024. [Electronic resourse]. – Access mode: https://gs.statcounter.com/os-market-share– Title from the screen.
3. Elliotte Rusty Harold. XML Bible, 1999. - 974 p.
4. Build better apps faster with Jetpack Compose. [Electronic resourse]. – Access mode: https://developer.android.com/compose– Title from the screen.

УДК 004.8

Р.І. Яшкін[1], М.Г. Бердник[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ЗАСТОСУВАННЯ МЕТОДІВ FINE-TUNING ТА ТРАНСФЕРНОГО НАВЧАННЯ ДЛЯ ДІАГНОСТИКИ ХВОРОБИ ПАРКІНСОНА ЗА МОВЛЕННЯМ

Анотація. У даній роботі застосовано методи fine-tuning та трансферного навчання для адаптації попередньо натренованих мовних моделей до специфічних завдань діагностики хвороби Паркінсона за мовленням. Реалізація стратегій заморожування шарів, оптимізації параметрів і аугментації даних дозволила значно підвищити точність класифікації патологічних зразків навіть за умов обмеженої кількості даних.

Ключові слова: *хвороба Паркінсона, fine-tuning, трансферне навчання, мовлення, діагностика, аугментація*

Вступ. Діагностика захворювань, таких як хвороба Паркінсона, за допомогою аналізу голосу стає дедалі поширенішою у сучасній медицині. Протягом останнього десятиліття значно розвинулася методика дослідження голосових зразків, що дозволяє виявляти навіть незначні зміни у мовленні пацієнтів. Ранні підходи, засновані на навчанні моделей з нуля, часто стикались з проблемою обмеженої кількості даних, що впливало на точність виявлення патологічних ознак.

Основний зміст роботи. Сьогодні, інтеграція сучасних технологій штучного інтелекту дозволяє проводити більш комплексний аналіз мовлення, враховуючи індивідуальні особливості та різноманіття мовних характеристик пацієнтів. Розробка універсальних діагностичних систем відкриває нові перспективи для раннього виявлення захворювань та моніторингу стану здоров’я.

У даній роботі було застосовано сучасні методи fine-tuning та трансферного навчання (далі донавчання) для адаптації попередньо натренованої мовної моделі до специфічних завдань діагностики хвороби Паркінсона (ХП) за допомогою аналізу голосових зразків. Основною ідеєю є використання моделі, що спочатку навчається на великих загальних наборах даних англійською мовою, а потім донавчається на обмежених клінічних даних пацієнтів українською мовою. Такий підхід дозволяє зберегти базові характеристики моделей і одночасно набути здатність виявляти тонкі шаблони патологічного мовлення [1].

Першочерговим етапом є стратегія заморожування нижніх шарів мережі, що зменшує ризик перенавчання та дозволяє зосередити адаптацію на верхніх шарах, які відповідають за специфічні ознаки захворювання. Вибір оптимальних шарів для донавчання ґрунтувався на попередніх дослідженнях, де було

показано, що комбінація характеристик з різних рівнів моделі сприяє покращенню точності класифікації [1].

Для зменшення негативного впливу обмеженості клінічних даних застосовано низькі коефіцієнти навчання, регуляризацію та ранню зупинку, що дозволило уникнути перенавчання. Такий підхід є критично важливим у роботі з медичними даними, де якість класифікації має безпосереднє значення для подальшої діагностики [2].

Ще одним напрямком стало використання методів аугментації даних. За допомогою змін висоти тону, розтягнення сигналу у часі та додавання шуму було розширено обсяг навчальної вибірки, що дозволило збалансувати класи та підвищити стійкість моделі до варіативних особливостей мовлення [3].

Особливу увагу приділено доменному попередньому навчанню, коли модель спочатку навчалась на споріднених завданнях, зокрема на AudioSet або LibriSpeech, що дозволяло підібрати оптимальний набір даних для донавчання. Цей підхід, а також застосування міжмовного трансферного навчання, є актуальним для адаптації моделей до української мови, оскільки більшість загальнодоступних баз даних містять зразки мовлення іншими мовами [3].

Також у роботі було використано спеціалізовані архітектури, зокрема PD-ResNet. У 2021 році було запропоновано PD-ResNet для аналізу ходи, використовуючи резидуальні блоки для класифікації ХП на основі патернів ходьби [4]. Хоча ця модель була розроблена для аналізу рухів, ідея адаптації ResNet для виявлення ХП знайшла застосування і в аналізі мовлення. Поєднання акустичного аналізу з розпізнаванням мовлення дозволило досягти високої точності та стабільності діагностики, а комплексне використання методів донавчання, аугментації даних та спеціалізованих архітектур підтверджує ефективність застосування штучного інтелекту для діагностики захворювань [5].

Висновки. Запропонований підхід із застосуванням донавчання для діагностики ХП за мовленням продемонстрував високий потенціал і ефективність. Використання стратегій заморожування шарів, аугментації даних і доменного попереднього навчання забезпечило покращення точності класифікації патологічних ознак. Спеціалізовані архітектури, такі як PD-ResNet і донавчена Wav2Vec 2.0, дозволили адаптувати моделі до вимог клінічної практики. Отримані результати відкривають нові перспективи для розробки систем ранньої діагностики та моніторингу неврологічних захворювань. Комплексний аналіз досліджень підтверджує наукову обґрунтованість підходу і та доцільність його застосування у сучасній медицині. Дослідження свідчать, що інтеграція цих передових технологій, значно підвищує ефективність діагностики пацієнтів із ХП.

ПЕРЕЛІК ПОСИЛАНЬ

1. Wiepert, D. A., Utianski, R. L., Duffy, J. R., Stricker, J. L., Barnard, L. R., Jones, D. T., & Botha, H. (2024). Exploring transfer learning for pathological speech feature prediction: Impact of layer selection. URL: https://arxiv.org/html/2402.01796v1

2. Wu, H., Soraghan, J., Lowit, A., & Di Caterina, G.. A Deep Learning Method for Pathological Voice Detection Using Convolutional Deep Belief Networks. In *Interspeech (2018*). URL: https://www.isca-archive.org/interspeech_2018/wu18b_interspeech.pdf
3. Pepino, L., Riera, P., & Ronchetti, F. (2023). Voice Disorder Analysis: A Transformer-based Approach. URL: https://arxiv.org/html/2406.14693v1
4. Yang X, Ye Q, Cai G, Wang Y, Cai G. PD-ResNet for Classification of Parkinson's Disease From Gait (2022). doi: 10.1109/JTEHM.2022.3180933
5. Klempíř, J., Mekyska, J., Brabenec, L., Rektorová, I., & Smékal, Z. (2024). Analyzing Wav2Vec Embeddings in Parkinson's Disease Speech: A Study on Cross-database Classification and Regression Tasks. *medRxiv*. URL: https://www.medrxiv.org/content/10.1101/2024.04.10.24305599v1

УДК 004.4

В.В. Спірінцев[1], М.О. Михайленко[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ВИЯВЛЕННЯ КРОСБРАУЗЕРНИХ НЕСУМІСНОСТЕЙ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ ВІЗУАЛЬНОГО ТЕСТУВАННЯ

Анотація. В роботі висвітлено особливості кросбраузерної несумісності і представлено підхід до зменшення її негативного впливу на процес веб-орієнтованої розробки. Зокрема, запропоновано фреймворк автоматизованого кросбраузерного візуального тестування, що базується на методах комп'ютерного зору та дозволяє виявляти візуальні прояви кросбраузерних несумісностей і, тим самим, підвищити продуктивність праці розробників та тестувальників веб-застосунків, а також знизити витрати на розробку програмного продукту. Запропоноване рішення базується на комбінації підходу порівняння знімків вікна веб-сторінки, відкритого за допомогою базового браузера і тестованих браузерів, з підходом аналізу DOM властивостей веб-елементів.

***Ключові слова**: кросбраузерні несумісності (XBI), браузер, автоматизоване тестування, візуальне тестування, DOM, Java, Selenium WebDriver, OpenCV.*

Вступ. Наразі сучасний світ неможливо уявити без веб-технологій, вони стали невід'ємною частиною людського життя, надаючи велику кількість можливостей для спрощення виконання багатьох задач, таких як: пошук інформації, соціальна взаємодія, бізнес процеси, комерція тощо. Тому одним із найбільш популярних видів програмного забезпечення є веб-застосунки. Для користування якими необхідними умовами є лише доступ до інтернету і наявність веб-браузеру. На даний момент існує велика кількість браузерів, кожен з яких має власні особливості і відмінності, починаючи від зовнішнього вигляду інтерфейсу і закінчуючи компонентами та організацією архітектури побудови. Це все впливає на відображення та функціонування веб-застосунків, а саме спричиняє появу кросбраузерних несумісностей (XBI) – різниць в поведінці веб-додатків залежно від їх запуску в тому чи іншому браузері. До їх прикладів

можна віднести: відсутність веб-елементу на сторінці, некоректне відображення елемента, його розміру, кольору, розташування тощо. Через некоректність веб-застосунку, його популярність може знижуватися серед користувачів. Тому веб-розробники та тестувальники приділяють певну частину часу розробці на урегулюванню проблеми кросбраузерної сумісності, що в свою чергу збільшує витрати проекту.

Визначають декілька підходів виявлення кросбраузерних несумісностей [1]: використання структури DOM, використання комп'ютерного зору і комбінація попередніх двох. Існує велика кількість досліджень, де були розроблені різні варіації інструментів ідентифікації XBI, до них відносяться наступні інструменти: WebDiff [2], Browserbite [3], X-Pert [4], CrossCheck [5], X-Check [6] та інші. Однак, проблема кросбраузерної сумісності все ще залишається актуальною і потребує подальших досліджень для усунення невирішених викликів.

Визначення проблеми. Головна причина проблематики кросбраузерної сумісності пов'язана із відмінностями між браузерами, а особливо між архітектурою браузерів.

До складу основних компонентів базової архітектури браузера входять: інтерфейс користувача, двигун браузера, рендеринговий двигун, JavaScript двигун, мережевий модуль, аналізатор XML, бекенд інтерфейсу користувача і система зберігання даних. Однак безпосередніми компонентами, які мають вплив на появу кросбраузерних несумісностей є: двигун компанування, JavaScript двигун і двигун обробки подій.

Двигун компанування є частиною рендерингового двигуну, реалізуючи з'єднання структурної частини веб сторінки з таблицями стилів. Він є головною причиною появи XBI. JavaScript двигун відповідає за інтерпретацію коду JavaScript, що вбудовано в веб-сторінку. Цей компонент викликає кросбраузерні несумісності через відмінності виконання коду в браузерах. Двигун обробки подій складається з комбінації рендерингового і JavaScript двигунів, де перший відповідає за обробку подій DOM, а другий – за динаміку взаємодії подій і пов'язаних з ними функцій.

Базуючись на цьому прояви кросбраузерних несумісностей поділяють на два типи: візуальні і функціональні. Перший тип стосується саме відмінностей у візуальному відображенні веб-сторінки та її елементів, таких як розмір, розташування чи взагалі присутність елемента. В той час як другий тип стосується різниці у виконанні програми та її функціональності.

Основні методи виявлення XBI використовують структуру DOM та комп'ютерний зір, або їх комбінацію. До них включають [1,7]: порівняння знімків екрану, адаптивне випадкове тестування, структурний аналіз DOM, ізоморфізм графів, машинне навчання, відносне розташування, Record-n-play, евристичне оцінювання і статичний аналіз.

Існує велика кількість розробок і рішень, що використовує перераховані вище методи [1,7], але незважаючи на це, певні складнощі виявлення

кросбраузерних несумісностей все ще залишаються невирішеними. До основних викликів, з якими найчастіше зустрічаються, відносяться [7]: недосяжні стани, відсутність автоматизації інструменту, хибнопозитивні та хибнонегативні результати, різні DOM-моделі веб-сторінки, зміни стану тригера і виявлення динамічних, змінних та інтерактивних елементів. Для вирішення визначеної проблеми, в даній роботі пропонується фреймворк автоматизованого кросбраузерного візуального тестування, який призначений для виявлення візуальних проявів кросбраузерних несумісностей.

Основний зміст роботи. Архітектура фреймворку складається з декількох компонентів. Перший з яких – це тестовий адаптер, в якості якого виступає Selenium WebDriver [8]. Компонент логування реалізується за допомогою бібліотеки. Джерело тестових даних фреймворку являють собою файли формату JSON. Компонент самих тестів використовує методи візуального тестування, що реалізуються за допомогою бібліотеки OpenCV. Окрім цього до основних компонентів представленого фреймворку можна віднести компонент звітування.

Також варто зазначити, що в основі архітектури запропонованого фреймворку лежить набір шаблонів проектування, що включає Factory, PageObject і Steps.

На рис.1 представлено діаграму класів для демонстрації реалізації процесу виявлення кросбраузерних несумісностей. До складу якої увійшли класи: «Screenshot», «Comparison», «VisualDifference», «DOMdifference», «Incompatibility» і «Report».

Екземпляр класу «Screenshot» уособлює себе об’єктом скріншоту веб-сторінки. Його атрибути включають: шлях до файлу збереженого скріншоту, ім’я веб-сторінки і назву браузера, де було зроблено скріншот. Щодо методів, то окрім конструктора і гетерів, клас «Screenshot» включає методи обробки скріншотів та виявлення на них візуальних відмінностей.

Клас «Comparison» пов’язаний з класом «Screenshot» відношенням агрегації. Його екземпляр представляє собою і водночас виконує функцію порівняння скріншотів веб-сторінок з двох різних браузерів.

Клас «VisualDifference» пов’язаний з класом «Screenshot» відношенням залежності. Екземпляр цього класу відповідає за об’єкт візуальної відмінності між скріншотами веб сторінок базового і тестованого браузерів. До його головних методів відносяться: ідентифікація веб-елемента та виявлення відмінності у його DOM властивостях.

Клас «DOMdifference» пов’язаний з класом «VisualDifference» відношенням залежності. Екземпляр цього класу відповідає за об’єкт відмінності DOM властивостей веб-елементів між скріншотами веб-сторінок базового і тестованого браузерів. До його головних методів відносяться: класифікація відмінності і ідентифікація кросбраузерної несумісності.

Клас «Incompatibility» пов’язаний з класом «DOMdifference» відношенням залежності. Екземпляр цього класу відповідає за об’єкт кросбраузерної несумісності.

Клас «Report» пов’язаний з класом «Incompatibility» відношенням агрегації. Екземпляр цього класу відповідає за об’єкт звіту виявлення кросбраузерних несумісностей. До його головних методів відносяться створення, збереження і виведення звіту.

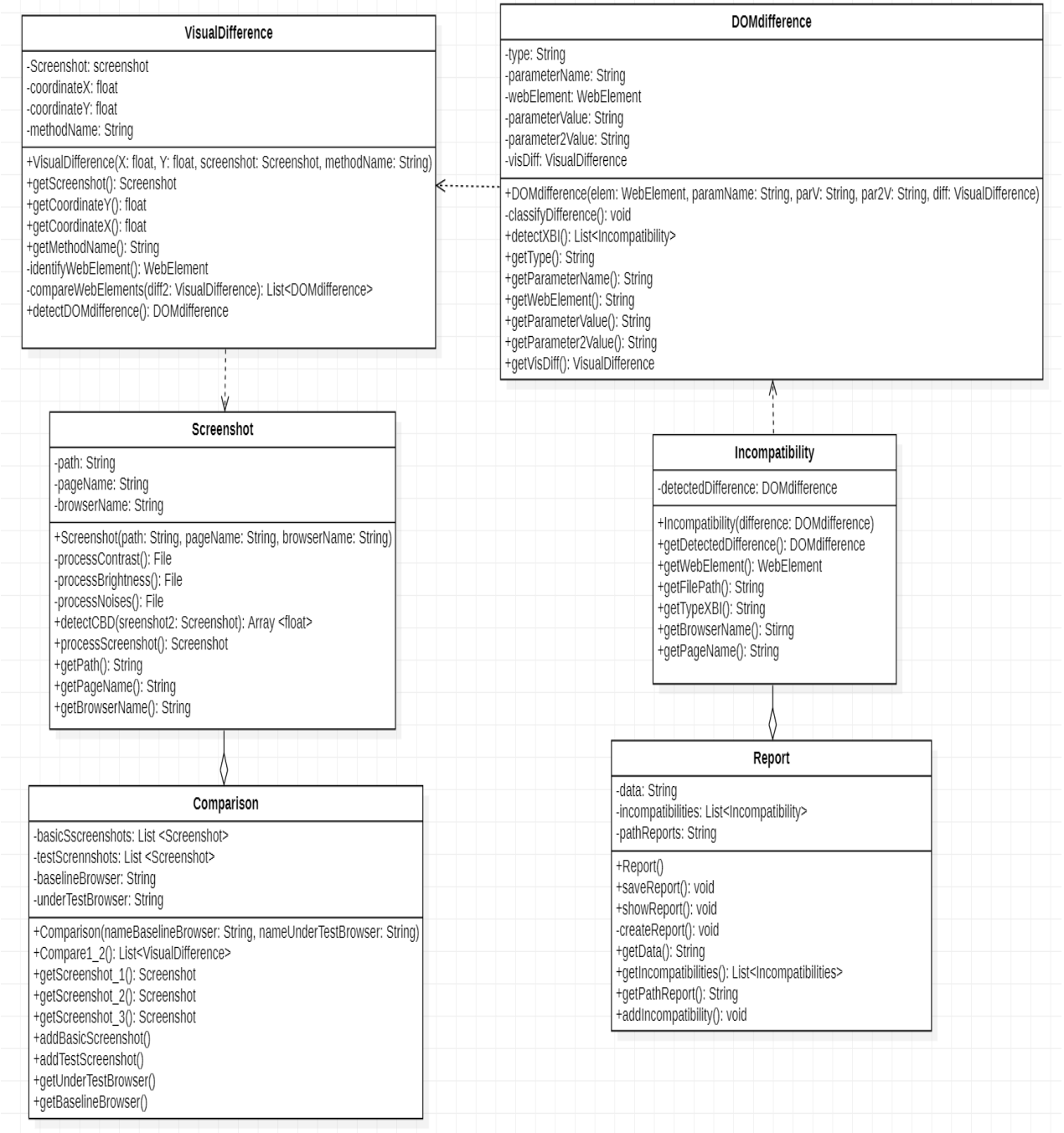


Рис. 1. Діаграма класів модуля виявлення XBI

Критерії, на базі яких відбувається пошук і класифікація кросбраузерних несумісностей, включають:

- відмінності у видимості – веб-елемент є не видимий у тестованому браузері, при цьому будучи видимим у базовому;

- відмінності положення – положення елемента відрізняється більш ніж на 40 пікселів по горизонталі чи по вертикалі;
- відмінності у розмірі – властивості висоти та/або ширини одного й того самого елемента, в різних браузерах, перевищує 15 пікселів;
- відмінності зовнішнього вигляду – відрізняються одна чи всі з наступних властивостей веб елемента: стиль шрифту, розмір шрифту, колір елемента та вміст елемента.

Алгоритм роботи фреймворку включає наступні етапи:

1. Відкриття тестованого веб-застосунку у базовому браузері, перехід на певні веб-сторінки, захоплення скріншотів з подальшим збереженням.
2. Відкриття тестованого веб-застосунку у тестуючих браузерах, перехід на ті самі веб-сторінки, захоплення їх скріншотів з подальшим збереженням.
3. Порівняння скріншотів з різних браузерів, результат якого – виявлення візуальних відмінностей, їх координат розташування та класифікація.
4. Ідентифікація потенційного веб-елемента з XBI за допомогою використання отриманих координат на попередньому кроці.
5. Порівняння DOM властивостей ідентифікованого веб-елемента відносно двох браузерів.
6. Виявлення кросбраузерної несумісності, її властивості та проведення її класифікації.
7. Відкриття інших сторінок тестованого веб-застосунку і застосування до них попередньо описаних кроків.
8. Створення звіту про виявлені кросбраузерні несумісності та їх характеристики.
9. Виведення звіту на консоль і збереження звіту у текстовий файл.

Класифікація візуальних відмінностей базується на розмірі фрагмента, координати якого виявляється під час порівняння скріншотів. Виділяються наступні види:

- велика проблема з макетом;
- проблема горизонтального положення;
- проблема вертикального положення;
- незначний візуальний збій.

При цьому класифікація саме кросбраузерних несумісностей базується на основі раніше описаних критеріїв. Тобто виділяється наступні типи XBI, на виявлення яких розрахованих запропонований фреймворк: різний зовнішній вигляд; різний розмір; різна видимість; різне позиціювання; різний контент; відсутність елемента.

Якщо під час проведення тестування не було ідентифіковано веб-елемент, то тип кросбраузерної несумісності матиме назву «Веб елемента не було ідентифіковано».

Структура звіту (табл.1) складається з даних про кожну виявлену кросбраузерну несумісність, включаючи: тип несумісності, додаткові деталі, тип

візуальної відмінності, веб-елемент, назви і версії браузерів, шлях розташування файлів скріншотів, розмір вікна, і назва веб-сторінки. При цьому основні з перерахованих даних підраховуються згідно тестуючих браузеру та веб сторінки і включається в відокремлений розділ звіту «Узагальнення».

Таблиця 1.

Структура звіту

<table>
<tr><td colspan="6">Номер та назва звіту</td></tr>
<tr><td colspan="6">Дата і час</td></tr>
<tr><td colspan="6">Кількість тестованих браузерів</td></tr>
<tr><td colspan="6">Кількість тестованих сторінок</td></tr>
<tr><td rowspan="6">Назва та версія тестованого браузеру №1…n</td><td colspan="5">Назва та версія базового браузеру</td></tr>
<tr><td rowspan="5">Назва сторінки №1…k</td><td colspan="4">Шлях розташування скріншоту та його розмір</td></tr>
<tr><td rowspan="4">Кросбраузерні несумісності</td><td rowspan="4">Несумісність №1…m</td><td>Веб-елемент</td></tr>
<tr><td>Візуальна різниця</td></tr>
<tr><td>Тип</td></tr>
<tr><td>Деталі</td></tr>
<tr><td rowspan="5">Узагальнення</td><td rowspan="5">Назва тестованого браузеру №1…n</td><td colspan="4">Розмір вікна</td></tr>
<tr><td colspan="4">Кількість несумісностей</td></tr>
<tr><td rowspan="3">Назва сторінки №1…k</td><td>Веб-елементи</td><td colspan="2">Назва та кількість</td></tr>
<tr><td>Візуальна різниця</td><td colspan="2">Назва та кількість</td></tr>
<tr><td>Тип XBI</td><td colspan="2">Назва та кількість</td></tr>
</table>

Наукова новизна. Отримав подальший розвиток алгоритм автоматизованого виявлення візуальних кросбраузерних несумісностей, що дозволяє опрацьовувати певний набір веб-сторінок тестованого веб-застосунку відразу в кількох браузерах з можливістю аналізу та дослідження особливостей кросбраузерної сумісності. Запропоноване рішення дозволяє збільшити продуктивність праці веб-розробників за рахунок скорочення часу розробки веб-застосунків, що витрачається на урахування відмінностей між браузерами.

Висновки. В роботі висвітлено особливості кросбраузерної несумісності і представлено підхід до зменшення її негативного впливу на процес веб-орієнтованої розробки. Зокрема, запропоновано фреймворк автоматизованого кросбраузерного візуального тестування, що базується на методах комп'ютерного зору та дозволяє виявляти візуальні прояви кросбраузерних несумісностей і, тим самим, підвищити продуктивність праці розробників та тестувальників веб-застосунків, а також знизити витрати на розробку програмного продукту. Запропоноване рішення базується на комбінації

загальних підходів, що включають такі основні етапи, як: порівняння скріншотів, виявлення візуальних невідповідностей, ідентифікація веб-елемента, порівняння його DOM властивостей, виявлення кросбраузерної несумісності, її класифікація та створення звіту. Модуль виявлення XBI, що включає перераховані етапи, реалізований за допомогою шістьох класів, пов'язаних між собою, а саме: «Screenshot», «Comparison», «VisualDifference», «DOMdifference», «Incompatibility» і «Report». Основні компоненти архітектури фреймворку і використані технології включають: мову програмування Java, адаптер Selenium WebDriver, бібліотеку логування log4j, бібліотеку обробки зображень OpenCV та такі шаблони проектування, як Factory, PageObject і Steps. При цьому основне джерело тестових даних – це JSON файли.

Подальший розвиток досліджень доцільно здійснювати у напрямку удосконалення розглянутого алгоритму виявлення кросбраузерних несумісностей, які проявляються у веб-елементах саме з динамічною поведінкою. Це може бути виправданим тим, що представлена модель виявлення XBI в даному проекті є придатною для реалізації підходу Record-n-Play, який призначений для подібних цілей.

ПЕРЕЛІК ПОСИЛАНЬ

1. Watanabe, Willian Massami et al. "Towards Cross-browser Incompatibilities Detection: Asystematic Literature Review." International Journal of Software Engineering & Applications 10 (2019): 17-32.
2. S. Choudhary, H. Versee, and A. Orso, "Webdiff: Automated identification of cross-browser issues in web applications," in Proc. of the 2010 IEEE International Conference on Software Maintenance (ICSM 2010), Sept 2010, pp. 1–10.
3. N. Semenenko, M. Dumas, and T. Saar, "Browserbite: Accurate cross-browser testing via machine learning over image features," in Proc. of the 29th IEEE International Conference on Software Maintenance (ICSM 2013), Sept 2013, pp. 528–531.
4. S. R. Choudhary, M. R. Prasad, and A. Orso, "X-pert: A web application testing tool for crossbrowser inconsistency detection," in Proc. of the 2014 International Symposium on Software Testing and Analysis (ISSTA 2014). New York, NY, USA: ACM, 2014, pp. 417–420. [Online]. Available: http://doi.acm.org/10.1145/2610384.2628057
5. S. Choudhary, M. Prasad, and A. Orso, "Crosscheck: Combining crawling and differencing to better detect cross-browser incompatibilities in web applications," in Proc. of the 5thInternational Conference on Software Testing, Verification and Validation (ICST 2012), April 2012, 171–180.
6. M. He, G. Wu, H. Tang, W. Chen, J. Wei, H. Zhong, and T. Huang, "X-check: A novel crossbrowser testing service based on record/replay," in 2016 IEEE International Conference on Web Services (ICWS), June 2016, pp. 123–130.
7. Sabaren, Leandro N., Maximiliano Agustin Mascheroni, Cristina L. Greiner and Emanuel Agustín Irrazábal. "A Systematic Literature Review in Cross-browser Testing." J. Comput. Sci. Technol. 18 (2018): 03.
8. Selenium. [Електронний ресурс]. – Режим доступу: https://www.selenium.dev/documentation/webdriver/ (дата звернення: 19.11.2024).

УДК 004.4

А.Л. Ширін[1], І.С. Кійко[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

НАЙБІЛЬША ЗРОСТАЮЧА ПІДПОСЛІДОВНІСТЬ

Анотація. В роботі висвітлено роз'яснення щодо вирішення олімпіадної задачі з програмування по знаходженню найбільшої зростаючої підпослідовності на мові С++. Для дотримання вимог щодо швидкості виконання розробленої програми в роботі використовується структура даних *set*.

Ключові слова: *розв'язок, рішення, задача, олімпіада, програмування, масив підпослідовність, структура, set, дані, С++.*

Вступ. В даній роботі надається рішення однієї з задач І туру всеукраїнської олімпіади з програмування з онлайн-платформи Algotester, що пропонує інструменти для тестування алгоритмів та розв'язання задач з програмування [1].

Умови задачі. Вам задано послідовність із n цілих чисел a_i. Ваша задача – знайти довжину найбільшої зростаючої підпослідовності (LIS – Longest Increasing Subsequence) заданої послідовності.

Ліміти. Час виконання не більше 2 секунд. Використання пам'яті не більше 256 МіВ.

Вхідні дані. У першому рядку задано ціле число n. У наступному рядку задано n цілих чисел послідовності a_i.

Вихідні дані. Виведіть довжину найбільшої зростаючої підпослідовності.

Обмеження: $1 \le n \le 100$; $1 \le a_i \le 10^9$.

Приклад.

Вхідні дані:

5

4 7 2 10 3

Вихідні дані:

3.

Підхід до вирішення. Головна ідея полягає в збереженні кінцевих елементів усіх можливих зростаючих підпослідовностей у впорядкованій структурі. Це дозволяє ефективно підтримувати найдовшу зростаючу підпослідовність.

Сортування та зберігання мінімальних кінцевих елементів будемо робити за наступним принципом. Для кожного елемента послідовності буде перевірятися наступна умова:

– якщо елемент більший за максимальний із збережених кінцевих елементів, додаємо його як новий у кінець.

– якщо ні, замінюємо найменший елемент, який більший або рівний поточному.

Для збереження кінцевих елементів будемо використовувати впорядковану структуру даних *set*, що підтримує логарифмічну складність для вставки та пошуку [2].

Довжина масиву даних в кінці виконання програми буде дорівнювати довжині найбільшої зростаючої підпослідовності, що і буде результатом рішення задачі.

Нижче наведено кроки алгоритму вирішення даної задачі:

1. Зчитуємо дані.
2. Ініціалізуємо порожню структуру set.
3. Проходимо кожен елемент послідовності a:
   – якщо елемент більший за найбільший у множині, додаємо його.
   – інакше замінюємо перший елемент у множині, який більший або рівний поточному.
4. Після завершення роботи розмір множини дорівнює довжині LIS.

Приклад вирішення.

Вхідні дані:

5

4 7 2 10 3

Порядок дій:

1. Зчитуємо дані.

   n = 5

   a = [4, 7, 2, 10, 3]
2. **Ініціалізуємо порожню структуру даних *set.***
3. **Проходимо кожен елемент послідовності a** (рис. 1).

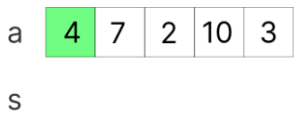


Рис. 1. Обробка першого елемента

Так як, множина *s* порожня, то просто добавляємо в неї елемент (рис. 2).

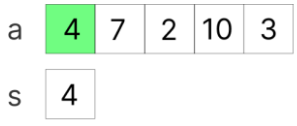


Рис. 2. Результат роботи після обробки першого елемента

Так як, другий елемент більший за найбільший елемент в множині *s*, то просто добавляємо його в кінець множини (рис. 3).

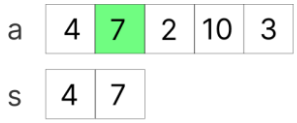


Рис. 3. Результат роботи після обробки другого елемента

Так як, третій елемент менший за найбільший елемент в множині s (рис.4), то замінюємо перший елемент у множині, який більший або рівний поточному (рис. 5).

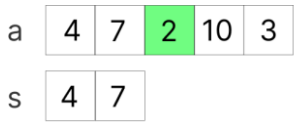


Рис. 4. Обробка третього елемента

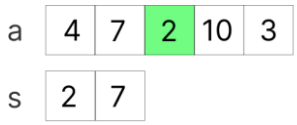


Рис. 5. Результат роботи після обробки третього елемента

Так як, четвертий елемент більший за найбільший елемент в множині s, то просто добавляємо його в множину (рис.6).

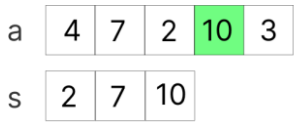


Рис. 6. Результат роботи після обробки четвертого елемента

Так як, п’ятий елемент менший за найбільший елемент в множині s (рис.7), то замінюємо перший елемент у множині, який більший або рівний поточному (рис. 8).

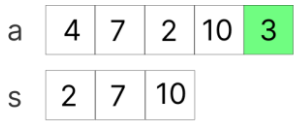


Рис. 7. Обробка п’ятого елемента

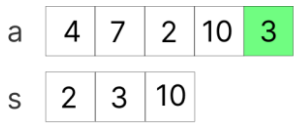


Рис. 8. Результат роботи після обробки п’ятого елемента

4. Після проходження всіх елементів в масиві *a*, розмір множини *s* і буде розміром найдовшої послідовності, тобто результат буде дорівнювати 3.

Аналіз складності.

Часова складність: для кожного елемента виконується пошук та вставка в O(log(n)), тобто загальна складність буде O(n log(n)).

Просторова складність: у найгіршому випадку в структурі буде O(n) елементів.

Висновок. Реалізований алгоритм є оптимальним для рішення даної задачі. Він забезпечує ефективне розв'язання навіть за великих обмежень *n* та a_i. Структура даних *set* дозволяє підтримувати порядок і ефективно виконувати операції пошуку та заміни.

Нижче наведено лістинг коду на мові програмування C++:

```
#inlude <bits/stdc++.h>
using namespace std;
int main()
{
   int n;
   cin >> n;  // Зчитування розмір масиву.
   vector<int> a(n);
   for (int i = 0; i < n; i++)
   {
      cin >> a[i]; // Зчитування елементів масиву
   }
   set<int> s; // ініціалізація впорядкованої структури даних
   for (int num : a)
   {
      if (s.empty() || *s.rbegin() < num) // Якщо елемент більший за найбільший елементи в множині, то додаємо його
      {
         s.insert(num);
      }
      else // В протилежному випадку замінюємо елемент в множинні, який більше або дорівнює поточному елементу
      {
         s.erase(s.lower_bound(num));
         s.insert(num);
```

```
            }
        }
        cout << s.size() << endl;
    }
```


ПЕРЕЛІК ПОСИЛАНЬ

1. Algotester - інструменти для розв'язання задач з програмування [електронний ресурс]. URL: https://algotester.com/uk (дата звернення: 20.12.2024);
2. Куток програміста: шаблонний клас std::set у C++ [електронний ресурс]. URL: http://www.kytok.org.ua/post/std-set-u-cplusplus (дата звернення: 20.12.2024);
3. Задача “Найбільша зростаюча підпослідовність”. URL: https://algotester.com/uk/ArchiveProblem/DisplayWithFile/40782.

УДК 004.4

А.Л. Ширін[1], І.С. Кійко[1]

[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ПЕРЕСТАНОВКИ ДВОХ МАСИВІВ

Анотація. В роботі висвітлено роз’яснення щодо вирішення олімпіадної задачі з програмування по знаходженню найбільшої суми різниці двох масивів на мові C++. Для дотримання вимог щодо швидкості виконання розробленої програми в роботі використовується сортування.

Ключові слова: *розв’язок, рішення, задача, олімпіада, програмування, масив, сортування, перестановка, C++.*

Вступ. В даній роботі надається рішення однієї з задач І туру всеукраїнської олімпіади з програмування з онлайн-платформи Algotester, що пропонує інструменти для тестування алгоритмів та розв'язання задач з програмування [1].

Умови задачі. Вам дано дві послідовності: a з довжиною n і b з довжиною m. Нехай $k = \min(n,m)$. Ви хочете вибрати перестановку c з послідовності a і перестановку d з послідовності b для того, щоб максимізувати наступний рахунок:

$$\sum_{i=1}^{k} |c_i - d_i|$$

Ліміти. Час виконання не більше 2 секунд. Використання пам’яті не більше 512 МіВ.

Вхідні дані. В першому рядку знаходяться два цілочислений числа n і m – довжини послідовностей a і b відповідно. В другому рядку знаходяться n цілочислених чисел a_i. В третьому рядку знаходяться m цілочислених чисел b_i.

Вихідні дані. В єдиному рядку вивести цілочислене число – максимальний рахунок.

Обмеження: $1 \le n, m \le 10^5$; $0 \le a_i, b_i \le 10^9$.

Приклади.

Вхідні дані:	Вхідні дані:
4 4	4 7
4 7 7 4	1 2 3 4
44 47 4 7	10 20 30 40 50 60 70
Вихідні дані:	Вихідні дані:
86	210

Підхід до вирішення. Для розв'язання цієї задачі головна ідея полягає в тому, щоб максимізувати абсолютну різницю для кожної пари елементів із послідовностей. Основні інструменти для цього – сортування та двовказівниковий підхід.

Нижче наведено кроки алгоритму вирішення даної задачі:

1. Зчитати вхідні дані для *n, m*, послідовностей *a* і *b*.
2. Відсортувати обидві послідовності.
3. Якщо потрібно, обміняти *a* та *b*, щоб $n \le m$.
4. Використовувати двовказівниковий підхід:

- порівнювати найменші та найбільші значення з обох кінців.
- вибирати пари, які дають найбільшу абсолютну різницю, і оновлювати вказівники.

5. Вивести накопичений результат.

Приклад вирішення.

Вхідні дані:

4 5
20 200 13 11 201
10 12 100 1

5. Зчитуємо дані:

n = 4
m = 5
a = [20, 200, 13, 11, 201]
b = [10, 12, 100, 1]

6. Сортуємо послідовності:

a = [11, 13, 20, 200, 201]
b = [1, 10, 12, 100]

7. Так, як n ≥ m, то міняємо послідовності місцями

n = 5
m = 4
a = [1, 10, 12, 100]
b = [11, 13, 20, 200, 201]

8. Використання двовказівникового підходу:

Ставимо вказівники в початкове положення (рис. 1).

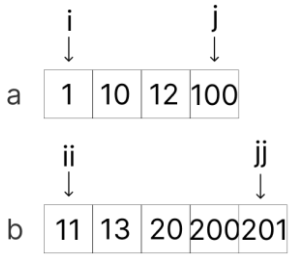


Рис. 1. Початкове положення вказівників

Обчислюємо, які вказівники дають найбільшу різницю:

$|a_i - b_{ii}| = |1 - 11| = 10$
$|a_i – b_{jj}| = |1 - 201| = 200$
$|a_j – b_{ii}| = |100 - 11| = 89$
$|a_j – b_{jj}| = |100 - 201| = 101$

Додаємо найбільшу різницю з вказіників, а саме з вказівників *i* та *jj* та здвигаємо їх:

sum = 0 + 200 = 200 (рис. 2).

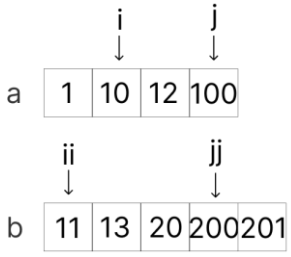


Рис. 2. Нове положення вказівників

Обчислюємо, які вказівники дають найбільшу різницю:

$|a_i - b_{ii}| = |10 - 11| = 1$
$|a_i – b_{jj}| = |10 - 200| = 190$
$|a_j – b_{ii}| = |100 - 11| = 89$
$|a_j – b_{jj}| = |100 - 200| = 100$

Додаємо найбільшу різницю з вказіників, а саме з вказівників *i* та *jj* та здвигаємо їх:

sum = 200 + 190 = 390 (рис. 3).

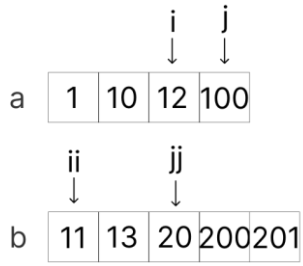


Рис. 3. Нове положення вказівників

Обчислюємо, які вказівники дають найбільшу різницю:

$|a_i - b_{ii}| = |12 - 11| = 1$
$|a_i – b_{jj}| = |12 - 20| = 8$
$|a_j – b_{ii}| = |100 - 11| = 89$
$|a_j – b_{jj}| = |100 - 20| = 80$

Додаємо найбільшу різницю з вказіників, а саме з вказівників *j* та *ii* та здвигаємо їх:

sum = 390 + 89 = 479 (рис. 4).

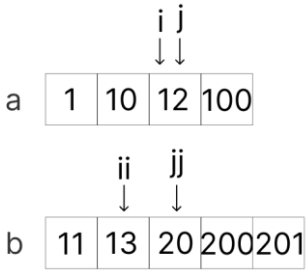


Рис. 4. Нове положення вказівників

Обчислюємо, які вказівники дають найбільшу різницю:

$|a_i - b_{ii}| = |12 - 13| = 1$
$|a_i – b_{jj}| = |12 - 20| = 8$
$|a_j – b_{ii}| = |12 - 13| = 1$
$|a_j – b_{jj}| = |12 - 20| = 8$

Додаємо найбільшу різницю з вказіників, а саме з вказівників *i* та *jj* та здвигаємо їх:

sum = 479 + 8 = 487 (рис. 5).

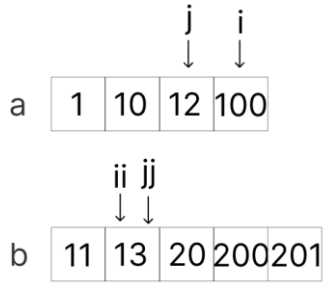


Рис. 5. Нове положення вказівників

9. Так як, вказівник *i* знаходиться правіше за вказівник *j*, це означає, що всі елементи з найменшої послідовності були пройдені, тому можна виводити накопичений результат, а саме значення *sum*, яке дорівнює 487.

Аналіз складності

Часова складність. Сортування: *O(nlogn+mlogm)*. Цикл з двовказівниками: *O(n)*. Загалом: *O(nlogn+mlogm)*, що є ефективним для великих вхідних даних.

Просторова складність: *O(n+m)* для векторів.

Висновок. Цей підхід ефективно максимізує результат завдяки використанню сортування та двовказівникової стратегії. Рішення оптимізовано для великих обсягів вхідних даних завдяки часовій складності O(nlogn). Задача демонструє, як базові техніки, такі як сортування та жадібний підхід, можуть бути використані для розв'язання реальних обчислювальних проблем.

Нижче наведено лістинг коду на мові програмування C++:

```
#include <bits/stdc++.h>
using namespace std;
#define int long long
#define inf INT_MAX
#define MOD 1000000007
int32_t main()
{
   int n, m; // Зчитування розмірів масивів
   cin >> n >> m;
   vector<int> a(n), b(m);
   for (int i = 0; i < n; i++)
   {
      cin >> a[i]; // Зчитування елементів першого масиву
   }
   for (int i = 0; i < m; i++)
   {
      cin >> b[i]; // Зчитування елементів другого масиву
   }
```

```
// Сортування масивів
    sort(a.begin(), a.end());
    sort(b.begin(), b.end());
    if (a.size() > b.size()) // Якщо перший масив більший за другий, то поміняти їх місцями
    {
        swap(a, b);
        swap(n, m);
    }
    int res = 0;
// Ініціалізація початкових вказівників
    int i = 0;
    int j = n - 1;
    int ii = 0;
    int jj = m - 1;
    while (i <= j)
    {
        int first = max(abs(a[i] - b[ii]), abs(a[i] - b[jj])); // Максимальне значення, яке можна отримати, якщо використовувати початковий вказівник першого масиву
        int second = max(abs(a[j] - b[ii]), abs(a[j] - b[jj])); // Максимальне значення, яке можна отримати, якщо використовувати кінцевий вказівник першого масиву

        if (first >= second) // Максимальне значення з початкового вказівника більше за максимальне значення з кінцевого вказівника
        {
            if (abs(a[i] - b[ii]) >= abs(a[i] - b[jj])) // Різниця початкового вказівника першого масиву і початкового вказівника другого масиву більше за різницю початкового вказівника першого масиву і кінцевого вказівника другого масиву
            {
                res += abs(a[i] - b[ii]);
                ii++; // Зсування вказівника
            }
            else // В іншому випадку
            {
                res += abs(a[i] - b[jj]);
                jj--; // Зсування вказівника
            }
            i++; // Зсування вказівника
        }
        else // В іншому випадку
```

```
        {
            if (abs(a[j] - b[ii]) >= abs(a[j] - b[jj])) // Різниця кінцевого вказівника першого масиву і початкового вказівника другого масиву більше за різницю кінцевого вказівника першого масиву і кінцевого вказівника другого масиву

            {
                res += abs(a[j] - b[ii]);
                ii++; // Зсування вказівника
            }
            else // В іншому випадку
            {
                res += abs(a[j] - b[jj]);
                j--; // Зсування вказівника
            }
            j--; // Зсування вказівника
        }
    }
    cout << res << endl;
}
```

ПЕРЕЛІК ПОСИЛАНЬ

1. Algotester - інструменти для розв'язання задач з програмування [електронний ресурс]. URL: https://algotester.com/uk (дата звернення: 20.12.2024);
2. Задача “Permutations of Two Arrays” [електронний ресурс]. URL: https://algotester.com/en/ArchiveProblem/DisplayWithFile/71118 (дата звернення: 20.12.2024).

РОЗДІЛ 4

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ ДЛЯ ВИРІШЕННЯ ЗАВДАНЬ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ

УДК 004.4`2

О.М. Алексєєв[1], А.В. М'якенький[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДВОНАПРЯМНА LSMT МОДЕЛЬ УСУНЕННЯ НЕОДНОЗНАЧНОСТІ СЛІВ У ТЕКСТІ ПРИ МОДЕЛЮВАННІ КОГНІТИВНОГО ПРОЦЕСУ РОЗУМІННЯ

Анотація. В даній роботі описано модель для розв'язання задачі усунення неоднозначності слів в текстах української мови, побудованою за архітектурою двонапрямної LSTM. Результати, отримані з використанням набору даних на основі словника СУМ, показали ефективність обраної архітектури у порівнянні з однонапрямною архітектурою LSTM.

Ключові слова: *когнітивне моделювання, когнітивний процес, NLP, WSD, LSTM, Bi-LSTM, tensorflow.*

Вступ. Задача виокремлення значень слів у тексті має назву усунення неоднозначності слів або word sense disambiguation (WSD), та відноситься до вищих когнітивних функцій мозку, зокрема до когнітивного процесу розуміння. Вона має на меті визначення можливих значень слова в заданому контексті та може використовуватися при розв'язанні інших задач таких як ідентифікація перифразів, машинний переклад тощо [1].

Розв'язання задачі WSD залежить від мови текстів та вимагає адаптації побудованих моделей під особливості обраної мови. Це робить створення моделі для роботи з українськими тестами актуальною, адже більшість сучасних прикладних рішень розроблені для роботи з текстами англійської мови.

Постановка задачі. Для досягнення поставленої мети в роботі сформовані та вирішені такі завдання:

- аналіз методів проектування моделей за архітектурою двонапрямної LSTM;
- аналіз структури набору даних для навчання та тестування моделей побудованих за обраною архітектурою;
- створення набору даних для задачі WSD;

- розробка та навчання моделі за обраною архітектурою, шляхом аналізу напрацювань у цій сфері та вибору найбільш відповідних технологій;
- формування висновків щодо ефективності побудованої моделі, використовуючи тестові дані з набору даних.

Основний зміст роботи. Задача усунення неоднозначності слів належить до задач обробки природної мови та має багато підходів та методів вирішення, одним з яких є використання моделей нейронних мереж, що побудовані за архітектурою LSTM. Дана архітектура є одним з видів архітектури рекурентної мережі, основною ідеєю якої є робота з послідовностями даних у часі та збереження довгострокових залежностей у часі.

Архітектура LSTM складається з комірки, стан якої визначає поточну довгострокову пам'ять мережі, прихований стан, що є вихідними даними мережі з попереднього кроку та поточні вхідні дані мережі. Регулювання інформації, яка надходить на кожному кроці роботи мережі регулюється трьома шлюзами: шлюз забуття, вхідний шлюз та вихідний шлюз [2].

Вдосконалена архітектура LSTM, а саме двонапрямна LSTM або Bi-LSTM також містить у собі усі структурні елементи звичайної LSTM, але складається з двох її шарів. Це дозволяє зберігати залежності послідовних даних не тільки у прямому, а також і у зворотному напрямку, що в контексті задачі WSD може використовуватися для обчислення повного контексту цільового слова, враховуючи слова як до нього, так і після.

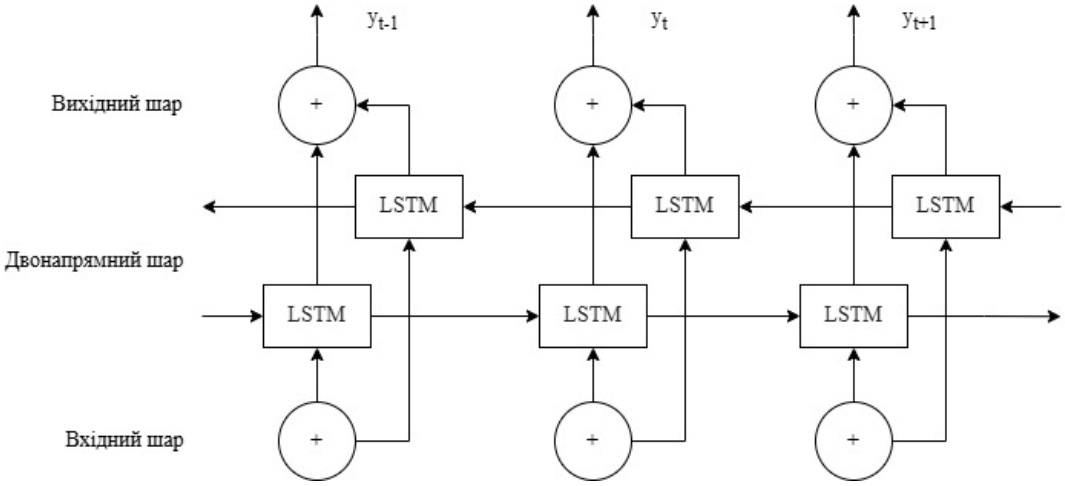


Рис. 1. Архітектура Bi-LSTM

Методологія. Для навчання моделі прогнозувати значення слів в україномовних тестах, був сформований набір даних на основі словника української мови СУМ українського мовно-інформаційного фонду НАН України. Для кожного слова зі словника було сформовано відповідність між цільовим словом, його визначеннями, а також прикладами використання слова у цьому значенні.

Наступним кроком, з метою формалізації, для отриманих даних був використаний алгоритм вбудовування слів. Вбудовування слів є технікою проекції слова у вбудований багатовимірний простір у вигляді вектора зі збереженням його семантичної та синтаксичної інформації. Для отримання вбудовування з набору даних був використаний алгоритм Glove, в якому проекція слів у вбудований простір відбувається шляхом максимізації подібності кожної пари слів, базуючись на співвідношенні появ цієї пари у корпусі [3].

Отримані закодовані значення слів контексту, що оточують цільове слово, використовуються як вхідні дані для прямого та зворотного шару двонапрямної LSTM. Кожний вхід моделі маркується значенням слова у відповідному прикладі. На виході моделі очікується прогнозоване значення цільового слова у заданому контексті.

Точність отриманих результатів обчислюється за допомогою косинуса подібності – метрики, що визначає коефіцієнт схожості двох векторів у просторі незалежно від їх величини. Косинусна подібність вимірюється між парою прогнозованого значення слова з реальним значенням слова з тестового набору даних [4]. Прогнозоване значення з максимальним значенням подібності вважається вихідним значенням моделі.

Тренування моделі. Для отримання даних для навчання та тестування, набір даних був відфільтрований для слів з більше ніж двома прикладами. Після цього перший приклад використання кожного слова був виокремлений до навчальної вибірки, а решта - до тестової.

Для реалізації моделі була використана бібліотека tensorflow мови програмування Python. Для оптимізації процесу навчання також був реалізований механізм ранньої зупинки, заснований на мінімізації loss функції, а також був використаний Nadam (Nesterov Adam) — оптимізатор, що призначений для покращення продуктивності моделей глибокого навчання [5].

Результати. Значення точності розробленої моделі на тестовій вибірці даних складає 83%. Для покращення точності роботи моделі була використана технологія виключення, або dropout. Алгоритм полягає в випадковому «вимкненні» нейронів на кожному етапі навчання та дозволяє зменшити перенавчання моделі.

Наукова новизна полягає у розробці моделі за архітектурою Bi-LSTM для розв’язання задачі неоднозначності слів в україномовних текстах та у створенні відповідного набору даних на основі словника СУМ.

Висновки. У даній роботі було розроблено модель розв’язання задачі WSD, як одного з підходів до моделювання когнітивного процесу розуміння, для розпізнавання сенсу заданих слів у контексті, використовуючи архітектуру двонапрямної LSTM. Результати навчання показали значення точності прогнозування на тестовому наборі даних у 83%, що показує ефективність обраної архітектури при розв’язанні задачі WSD.

ПЕРЕЛІК ПОСИЛАНЬ

1. Ranjan Pal A. Word Sense Disambiguation: A Survey / Alok Ranjan Pal, Diganta Saha // International Journal of Control Theory and Computer Modeling. – 2015. – Т. 5, № 3. – С. 1–16.
2. Popov A. Word Sense Disambiguation with Recurrent Neural Networks / Alexander Popov // RANLP 2017 - Student Research Workshop. – [Б. м.], 2017.
3. Pennington J. Glove: Global Vectors for Word Representation / Jeffrey Pennington, Richard Socher, Christopher Manning // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar. – Stroudsburg, PA, USA, 2014.
4. Gunawan D. The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents / D. Gunawan, C. A. Sembiring, M. A. Budiman // Journal of Physics: Conference Series. – 2018.
5. TensorFlow: Machine Learning Using Heterogeneous Edge on Distributed Systems / R. Ganesh Babu [та ін.] // Deep Learning in Visual Computing and Signal Processing. – Boca Raton, 2022. – С. 71–90.

УДК 004.932:528.854

Л.І. Мещеряков[1], Н.П. Уланова[1], А.В. Кожевніков[1], Б.С. Нановський[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

СКЛАДОВА МОДЕЛІ УДОСКОНАЛЕННЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ З ЗАСТОСУВАННЯМ МОМЕНТНИХ ФУНКЦІЙ

Анотація. Розроблено математичний апарат інформаційних характеристик асиметрійних функцій, що в силу своїх аналітичних та структурних особливостей мають чутливість до градієнта нелінійного зв'язку по горизонталі щільності розподілу ймовірностей значень сигналів з датчиків контролю оперативного технологічного та технічного станів гірничих електромеханічних комплексів і обумовлюють підвищення точності та достовірності визначення поточного стану останніх, що може бути використано для збільшення інформаційного забезпечення керування у відповідності сучасним вимогам інтегрованих задач автоматизованих систем керування технологічних процесів.

Ключові слова: *математичний апарат інформаційних характеристик, супутні випадкові сигнали, чутливість до градієнта нелінійного зв'язку, електромеханічні комплекси, асиметрійні функції.*

Вступ. Основа підвищення точності та надійності процесів в сучасних автоматизованих системах керування гірничими електромеханічними комплексами на даний час не забезпечуються в повному обсязі необхідною оперативною інформацією, що обумовлено як складною структурою та важкими умовами робочого функціонування обладнання в гірничий промисловості, так і недостатніми науковими дослідженнями в даному напрямі. На цьому і ґрунтується функціональне протиріччя між вимогою забезпечення оптимальної якості сучасного програмного керування технологічними процесами в гірничих електромеханічних комплексах та обмеженням необхідної та доступної для забезпечення цієї вимоги інформації. Таким чином, формується важлива наукова

задача пошуку нових якісних оцінок інформаційних характеристик діагностичних сигналів, що супутні робочим режимам гірничих електромеханічних комплексів. Аналіз структур та інформаційних властивостей дисперсійних функцій дає підставу за аналогією з ними сформувати та запропонувати до застосування при дослідженні динамічних та інформаційних характеристик нелінійних об'єктів в автоматизованих системах керування технологічними процесами гірничих електромеханічних комплексів нових функцій – асиметрійних.

Широко відомі одномірні кореляційні функції та їхні регресійні залежності, отримані за умовними математичними очікуваннями адекватні базовому визначенню і добре описують лінійні системи. Однак реально технологічні та технічні процеси, що протікають в гірничих електромеханічних комплексах є суттєво нелінійними, самі комплекси відповідно виступають як нелінійні, і апроксимація їх лінійними моделями вносить істотні похибки. Тому їх ідентифікацію доцільно здійснювати на базі класу моментних функцій, а саме за допомогою дисперсійних функцій, що дозволяють розкрити внутрішні нелінійні структурні зв'язки. Для розширення інформаційного забезпечення автоматизованих систем керування технологічними процесами гірничих електромеханічних комплексів перспективною являється ідея використати в структурі дисперсійних функцій і інші статистичні характеристики енергоінформаційних сигналів. При цьому можливі варіації одномірних автодисперсійних функцій випадкового сигналу входу $U(t)$ з заміною складової умовного математичного очікування на вищі умовні ймовірнісні оцінки, що визначить їх теж невипадковими функціями двох аргументів, які для кожної пари значень t, v будуть дорівнювати відповідно дисперсіям умовних дисперсій, дисперсіям умовних асиметрій та дисперсіям умовних ексцесів відповідних перетинів сигналу.

Одномірна автодисперсійна функція $\theta_{uu}(t,v)$ як і одномірна автокореляційна функції $K_{uu}(t,v)$ відображає в собі внутрішню структуру та зв'язки випадкового сигналу входу $U(t)$. При цьому ступінь розбіжності цих функцій показує ступінь нелінійності внутрішньої структури сигналу входу $U(t)$, а міра цієї розбіжності може бути використана як критерій при заміні нелінійної екстраполяції випадкового сигналу входу $U(t)$ його лінійною моделлю. Це стосується і інших інформаційних сигналів – стану $X(t)$ та виходу $Y(t)$. Так здійснюється формування принципів розширення інформаційного забезпечення керування гірничими електромеханічними комплексами та автоматизованими системами на основі інформаційних характеристик асиметрійних функцій.

Виклад основного матеріалу. При формалізації задач керування складними гірничими електромеханічними комплексами досить перспективною інформаційно потужною оцінкою виступає одномірна асиметрійна функція (функція асиметрії) випадкового сигналу $U(t\ \ )$, що визначається як невипадкова

функція двох аргументів $\gamma_{Миu}(t,v)$, яка для кожної пари значень t і v дорівнює асиметрії умовного математичного очікування відповідних перетинів значень сигналу $U(t\ \ )$ за виразом

$$\gamma_{Миu}(t,v)=\frac{1}{\sigma_{uu}^3}M\left[M\left(U_t\,|\,U_v\right)-MU_t\right]^3, \tag{1}$$

де $M(U_t|U_v)$ – умовне математичне очікування значення U_t випадкової функції вибірки сигналу входу $U(t)$ при довільному значенні аргументу t відносно значення u_v цієї ж функції при іншому довільному значенні аргументу v; MU_t – математичне очікування генеральної вибірки сигналу входу $U(t)$; σ_{uu} – середнє квадратичне відхилення вибірки випадкового сигналу входу $U(t)$.

Відповідно нормоване значення оцінки одномірної асиметрійної функції визначається формулою

$$\lambda_{Миu}^3(t,v)=\frac{\gamma_{Миu}(t,v)}{AU_t}, \tag{2}$$

де $\gamma_{Muu}(t,v)$ – одномірна автоасиметрійна функція випадкового сигналу входу $U(t)$; AU_t – асиметрії генеральної вибірки випадкового сигналу входу $U(t)$.

Для розширення інформаційного забезпечення автоматизованих систем керування технологічними процесами в гірничих електромеханічних комплексах перспективне використати і в структурі асиметрійних функцій інших статистичних характеристик енергоінформаційних сигналів. Так можливі варіації одномірних автоасиметрійних функцій випадкового сигналу входу $U(t\ \ )$ з заміною складової умовного математичного очікування на вищі умовні оцінки, що визначить їх теж невипадковими функціями двох аргументів, які для кожної пари значень t і v будуть дорівнювати відповідно асиметрії умовних дисперсій, асиметрії умовних асиметрій та асиметрії умовних ексцесів відповідних перетинів значень сигналу входу

$$\gamma_{Duu}(t,v)=\frac{1}{\sigma_{Duu}^3}M\left[D\left(U_t\,|\,U_v\right)-DU_t\right]^3,$$

$$\gamma_{Auu}(t,v)=\frac{1}{\sigma_{Auu}^3}M\left[A\left(U_t\,|\,U_v\right)-AU_t\right]^3, \tag{3}$$

$$\gamma_{Euu}(t,v)=\frac{1}{\sigma_{Euu}^3}M\left[E\left(U_t\,|\,U_v\right)-EU_t\right]^3,$$

де $D(U_t|U_v)$ – умовна дисперсія значення U_t випадкової функції вибірки сигналу входу $U(t)$ при довільному значенні аргументу t відносно значення u_v цієї ж функції сигналу при іншому довільному значенні аргументу v; DU_t – дисперсія генеральної вибірки випадкового сигналу входу $U(t\ )$; $A(U_t|U_v)$ – умовна асиметрія значення U_t випадкової функції вибірки сигналу входу $U(t)$ при довільному значенні аргументу t відносно значення u_v цієї ж функції при іншому довільному значенні аргументу v; AU_t – асиметрія генеральної вибірки випадкового сигналу входу $U(t\ )$; $E(U_t|U_v)$ – умовний ексцес значення U_t випадкової функції вибірки сигналу входу $U(t)$ при довільному значенні аргументу t відносно значення u_v цієї ж функції при іншому довільному значенні аргументу v; EU_t – ексцес генеральної вибірки випадкового сигналу входу $U(t\ )$; σ_{uu} – середнє квадратичне відхилення вибірки випадкового сигналу входу $U(t)$.

За представленими структурами виразів (3) формуються нормовані значення можливих варіацій оцінок одномірних автоасиметрійних функцій випадкового сигналу входу $U(t)$ відповідно за формулами

$$\lambda^3_{Duu}(t,v)=\frac{\gamma_{Duu}(t,v)}{AU_t}$$

$$\lambda^3_{Auu}(t,v)=\frac{\gamma_{Auu}(t,v)}{AU_t} \qquad (4)$$

$$\lambda^3_{Euu}(t,v)=\frac{\gamma_{Euu}(t,v)}{AU_t}$$

де $\gamma_{Duu}(t,v)$ – одномірна автоасиметрійна функція умовних дисперсій входу $U(t)$; $\gamma_{Auu}(t,v)$ – одномірна автоасиметрійна функція умовних асиметрій входу $U(t)$; $\gamma_{Euu}(t,v)$ – одномірна автоасиметрійна функція умовних ексцесів входу $U(t)$; AU_t – асиметрія генеральної вибірки випадкового сигналу входу $U(t)$.

Очевидно, що значно більш цінною інформативною оцінкою є одномірна взаємоасиметрійна функція, що визначається як невипадкова функція декількох аргументів $\gamma_{yxu}(s,v,t)$ і яка для кожної пари значень t, v і s дорівнює асиметрії умовного математичного очікування перетину одного випадкового сигналу щодо перетинів інших сигналів. Наприклад, для векторів значень сигналів виходу $Y(s)$ і входу $U(t)$ функція відобразиться таким чином

$$\gamma_{Myu}(s,t)=\frac{1}{\sigma_y^3}M\left[M\left(Y_s\,|\,U_t\right)-MY_s\right]^3 \qquad (5)$$

де $M(Y_s|U_t)$ – умовне математичне очікування значення Y_s випадкової функції вибірки сигналу виходу $Y(s)$ при довільному значенні аргументу s відносно значення u_t випадкової функції вибірки сигналу входу $U(t)$ при його довільному значенні заданого аргументу t; MY_s – математичне очікування генеральної вибірки сигналу виходу $Y(s)$; σ_{uu} – середнє квадратичне відхилення вибірки випадкового сигналу входу $U(t)$.

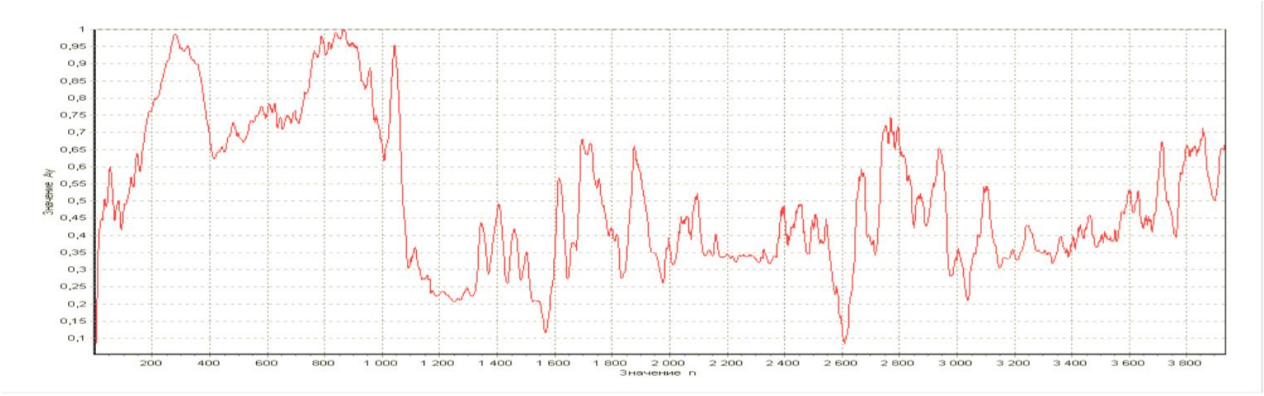
Інші варіації одномірних взаємоасиметрійних функцій випадкових сигналів входу $U(t)$ і виходу $Y(t)$, які визначені через вищі умовні оцінки, що пропонуються до застосування, визначають теж невипадкові функції двох аргументів, які для кожної пари значень *t* і *v* будуть дорівнювати відповідно вже взаємозв'язку через асиметрії умовних дисперсій, умовних асиметрій та умовних ексцесів відповідних перетинів різних сигналу (наприклад входу і виходу) будуть відображатися виразами

$$\gamma_{Dyu}(s,t)=\frac{1}{\sigma_{Dy}^3}\left[D(Y_s|U_t)-DY_s\right]^3$$

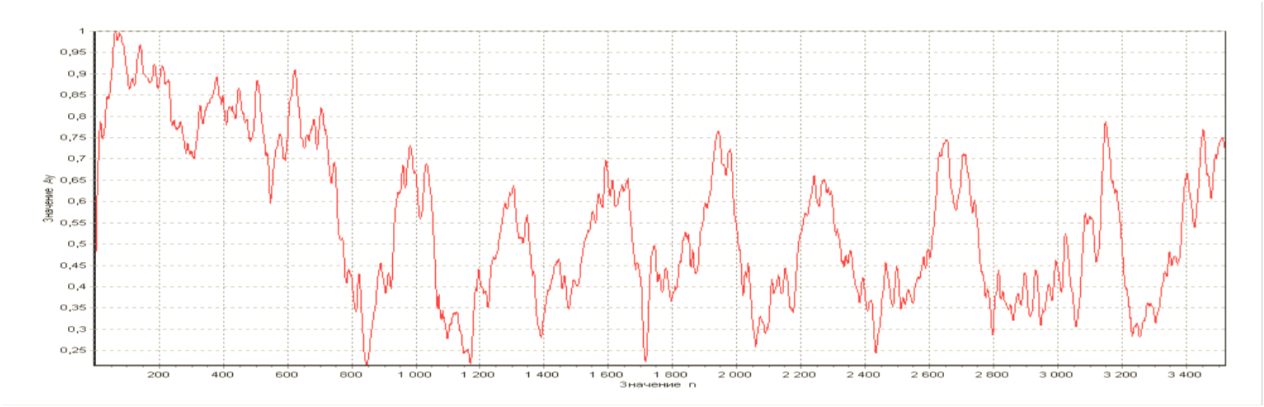
$$\gamma_{Ayu}(s,t)=\frac{1}{\sigma_{Ay}^3}\left[A(Y_s|U_t)-AY_s\right]^3 \qquad (6)$$

$$\gamma_{Eyu}(s,t)=\frac{1}{\sigma_{Ey}^3}\left[E(Y_s|U_t)-EY_s\right]^3$$

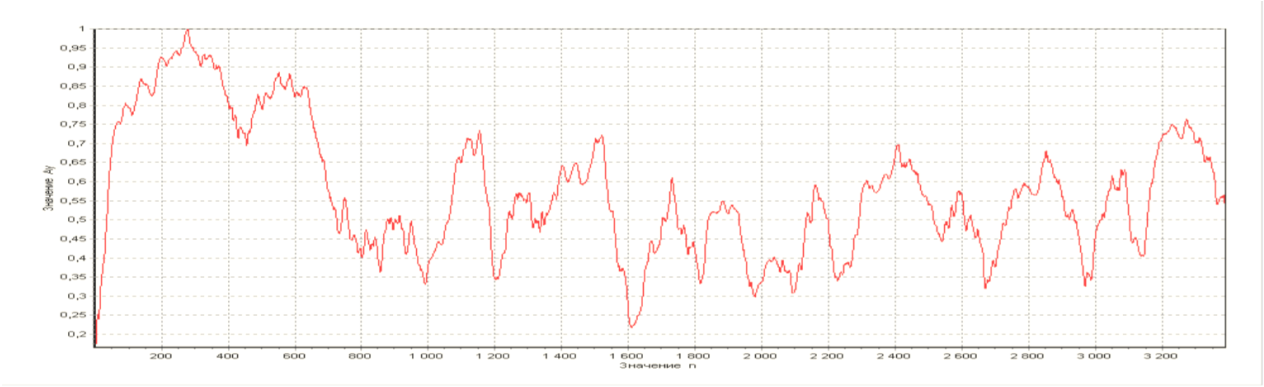
де $D(Y_s|U_t)$ – умовна дисперсія значення Y_s випадкової функції вибірки сигналу виходу $Y(s)$ при довільному значенні аргументу s відносно значення u_t випадкової функції вибірки сигналу входу $U(t)$ при довільному значенні аргументу t; DY_s – дисперсія генеральної вибірки випадкового сигналу виходу $Y(s)$; $A(Y_s|U_t)$ – умовна асиметрія значення Y_s випадкової функції вибірки сигналу виходу $Y(s)$ при довільному значенні аргументу s відносно значення u_t випадкової функції вибірки сигналу входу $U(t)$ при довільному значенні аргументу t; AY_s – асиметрія генеральної вибірки випадкового сигналу виходу $Y(s)$; $E(Y_s|U_t)$ – умовний ексцес значення Y_s випадкової функції вибірки сигналу виходу $Y(s)$ при довільному значенні аргументу s відносно значення u_t випадкової функції вибірки сигналу входу $U(t)$ при довільному значенні аргументу t; EY_s – ексцес генеральної вибірки випадкового сигналу виходу $Y(s)$; σ_y – середнє квадратичне відхилення вибірки випадкового сигналу виходу $Y(s)$.



а.



б.



в.

Рис. 1. Автоасиметрійні функції сигналу витрат потужності приводу бурового комплексу типу ЗИФ-1200МР: а – вихід на сталий робочий режим буріння; б – виникнення стану підвищеного зносу алмазної коронки в прижоговій ситуації на фоні вібрацій; в – стан буріння з різною швидкістю шпинделя станка

В якості практичних прикладів інформаційних властивостей оцінок, що пропонуються для інтелектуального діагностування оперативного стану складних гірничих електромеханічних комплексів представлені типові інформаційні характеристики отриманих нормованих асиметрійних функцій експериментальних сигналів активної потужності, що споживається, привідними

електродвигунами бурових комплексів, при різних технологічних та технічних режимах процесу буріння (рис. 1).

Не важко побачити виявлені аналітикою автоасиметрійних функцій приховані коливальні нелінійні зв’язки в структурі сигналів, що розглядаються. Безумовно, виникає потреба подальшого, більш детального вже спектрального аналізу даних сигналів з формуванням по достовірним експериментальним вибіркам інформативного простору ознак та найефективніших вирішальних правил компоненту задач розпізнавання стану об’єктів, що досліджуються.

Висновки. Таким чином, розроблено математичний апарат інформаційних характеристик асиметрійних функцій, що в силу своїх аналітичних та структурних особливостей мають чутливість до градієнта нелінійного зв'язку по горизонталі щільності розподілу ймовірностей значень сигналів з датчиків контролю оперативного технологічного та технічного станів гірничих електромеханічних комплексів і обумовлюють підвищення точності та достовірності визначення останніх, що може бути використано для збільшення інформаційного забезпечення керування у відповідності сучасним вимогам інтегрованих задач автоматизованих систем керування технологічними процесами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Piwniak G.G., Kaliski M., Zieba A., Mieszczerjakow L.J., Dudla M.A. Diagnostyka urzadzen wiertniczych. – Krakow, Dniepropietrowsk, 2004. – 174 c.
2. Мещеряков Л.І. Аналітичне та програмне забезпечення енергетичного діагностування гірничих комплексів на основі моментних функцій / Л.І. Мещеряков, О.М. Галушко, О.І. Сироткіна, С.Д. Приходченко // Гірнича електромеханіка та автоматика: Науково-технічний збірник, 2019. – вип.101.С.29 – 38.
3. Мещеряков Л.І. Статистичні моментні зв’язки інформаційних сигналів в системах розпізнавання / Л.І. Мещеряков, Н.П. Уланова, Л.В. Карманова, А.Л. Ширін // Гірнича електромеханіка та автоматика: Науково-технічний збірник, 2017. – вип.99. – С.46 – 51
4. Мещеряков Л.І. Інтелектуальна діагностика бурових комплексів / Л.І. Мещеряков // Сб. науч. трудов НГАУ. – Днепропетровск, 2007. – № 29. – С. 132–141.

УДК 004.415.3:681.6

О.О. Сітнікова[1], М.О. Мельник[1], О.П. Сирота[1]
[1]Приватна установа «Університет науки, підприємництва та технологій»
SET University, Київ, Україна

ОПТИМІЗАЦІЯ ПЛАНУВАННЯ У ГЕТЕРОГЕННИХ РОЗПОДІЛЕНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ ЗА ДОПОМОГОЮ МЕТАЕВРИСТИЧНИХ АЛГОРИТМІВ

Анотація. У доповіді розглядається проблема оптимального планування завдань у гетерогенних розподілених обчислювальних системах (HDCS) з використанням метаевристичних алгоритмів. Основну увагу приділено аналізу ефективності генетичного алгоритму (GA) та універіантного маргінального розподільчого алгоритму (UMDA) для розподілу паралельних завдань у розподілених середовищах. Запропоновані підходи оцінюються за такими критеріями, як час виконання (makespan), час очікування (waiting time) та використання ресурсів (resource utilization). Результати експериментів показують, що UMDA демонструє швидшу збіжність та ефективність у великих системах, тоді як GA забезпечує кращий баланс навантаження та рівномірний розподіл задач між процесорами.

***Ключові слова**: Гетерогенні розподілені обчислювальні системи, планування завдань, метаевристичні алгоритми, генетичний алгоритм, універіантний маргінальний розподільчий алгоритм, makespan, розподілені обчислення, оптимізація ресурсів.*

Вступ. Гетерогенні розподілені обчислювальні системи (HDCS) є основою для ефективної обробки великих обсягів даних та виконання складних обчислень. Оптимізація використання ресурсів у таких системах є складною задачею через їхню неоднорідність, що включає відмінності у швидкості процесорів, кількості ядер, топології мереж та комунікаційних затримках.

Одним з найбільш поширених підходів до організації паралельних обчислень є представлення задач у вигляді графів паралельних задач (Parallel Task Graphs, PTGs). Планування таких графів на HDCS є NP-складною задачею, що потребує застосування евристичних та метаевристичних методів. У цій роботі розглядається підхід до оптимізації планування PTGs за допомогою універіантного маргінального розподільчого алгоритму (Univariate Marginal Distribution Algorithm, UMDA), який порівнюється з генетичним алгоритмом.

Основний зміст роботи. Розподілене планування PTGs можна формалізувати через орієнтовані ациклічні графи (DAG), де вузли відповідають підзадачам, а ребра відображають залежності між ними.

Нехай

$$T=(N,E),$$

де: $N=\{\eta_i: i=1,\ldots,N\}$ – множина вузлів (підзадач),

$E=\{E_{i,j}: i,j=1,\ldots,N\}$ – множина орієнтованих ребер, що визначають потік виконання.

Критичний шлях (CP) визначається як найдовший шлях у графі за критерієм суми часу виконання підзадач:

$$CP=max \sum PW_i,$$

де W_i – час виконання підзадачі i, а P – множина всіх шляхів від початкового до кінцевого вузла.

Крім критичного шляху, важливими характеристиками є:

Щільність графа (Density) – показує кількість інформаційних зв'язків між підзадачами.

Шарування (Layering) – визначає розподіл підзадач по рівнях DAG.

Для розподілу задач використовується матриця ресурсів, що описує характеристики доступних обчислювальних вузлів.

Гетерогенні розподілені обчислювальні системи потребують ефективних методів планування завдань, оскільки оптимальний розподіл ресурсів безпосередньо впливає на загальну продуктивність системи. Однією з найважливіших проблем при цьому є визначення оптимального способу розподілу підзадач між процесорами та мінімізація часу виконання завдань. Існує багато підходів до вирішення цієї проблеми, однак найбільш поширеними є генетичний алгоритм (GA) та унівaріантний маргінальний розподільчий алгоритм (UMDA). Обидва ці методи є евристичними, тобто вони не гарантують абсолютного оптимального рішення, але дозволяють знайти прийнятний варіант розподілу за прийнятний час.

Генетичний алгоритм заснований на принципах природного добору та еволюції. Він імітує біологічні процеси, такі як спадковість, відбір, мутації та рекомбінація. Його робота починається з формування початкової популяції можливих рішень, які представляють різні варіанти розподілу підзадач між процесорами. Кожне з цих рішень кодується у вигляді хромосом, які містять інформацію про розподіл задач. Після ініціалізації виконується оцінка пристосованості кожного рішення за допомогою цільової функції, яка враховує такі параметри, як час виконання завдань (makespan) та час очікування (waiting time).

Після цього застосовуються оператори селекції, схрещування та мутації. Селекція відбирає найкращі рішення, які мають найменший makespan, для подальшого відтворення. Схрещування об'єднує частини різних хромосом, створюючи нові варіанти розподілу задач, які потенційно можуть бути кращими за батьківські. Мутація вносить випадкові зміни у деякі розподіли, щоб запобігти передчасному застряганню у локальному мінімумі. Після кількох ітерацій алгоритм поступово покращує розподіл, знаходячи все більш ефективні рішення.

Однак, незважаючи на свою ефективність, генетичний алгоритм має певні недоліки. По-перше, він може потребувати великої кількості ітерацій для

знаходження оптимального розподілу, особливо якщо початкова популяція була сформована неефективно. По-друге, процес схрещування та мутації може призводити до втрати хороших рішень або створення неефективних конфігурацій, що подовжує загальний час виконання алгоритму. Крім того, у складних багаторівневих графах задач генетичний алгоритм може не завжди забезпечувати рівномірний розподіл завдань між ресурсами, що призводить до їх неефективного використання.

На відміну від генетичного алгоритму, UMDA (Univariate Marginal Distribution Algorithm) належить до класу алгоритмів оцінки розподілу (EDA – Estimation of Distribution Algorithms). Замість використання операторів схрещування та мутації, UMDA працює зі статистичним моделюванням. Його основна ідея полягає у побудові розподілу ймовірностей для кожного параметра планування та генерації нових рішень на основі цього розподілу.

Процес роботи UMDA складається з таких основних кроків. Спочатку створюється початкова популяція можливих рішень, подібно до генетичного алгоритму. Однак, на відміну від GA, у UMDA обирається лише певна частина найкращих рішень, на основі яких будується статистична модель розподілу. Ця модель дозволяє оцінити ймовірності того, що певний розподіл підзадач буде ефективним. Потім нові рішення генеруються відповідно до отриманих ймовірностей, а процес повторюється до досягнення стабільного результату.

Такий підхід має кілька важливих переваг. По-перше, він дозволяє значно зменшити кількість ітерацій, необхідних для досягнення гарного результату, оскільки не потребує часу на операції схрещування та мутації. По-друге, він забезпечує швидшу адаптацію до нових вхідних даних та більш точний розподіл ресурсів. По-третє, завдяки використанню статистичної моделі UMDA більш ефективно знаходить баланс між розподілом завдань та комунікаційними витратами.

Експерименти показали, що UMDA демонструє значно швидший час виконання в порівнянні з генетичним алгоритмом. У випадках, коли потрібно швидко адаптуватися до змінюваного навантаження, цей метод є більш ефективним. Однак, слід зазначити, що UMDA не завжди забезпечує найкращу якість розподілу ресурсів, оскільки він не використовує складних механізмів оптимізації, таких як мутація та рекомбінація. У цьому відношенні генетичний алгоритм має перевагу, оскільки він може знайти більш якісні конфігурації розподілу, хоча і за рахунок збільшеного часу виконання.

Таким чином, вибір між GA та UMDA залежить від конкретної задачі. Якщо головним критерієм є мінімізація часу виконання, UMDA є кращим варіантом. Якщо ж пріоритетом є максимальна ефективність використання ресурсів, генетичний алгоритм може бути більш придатним. У майбутньому перспективним напрямком може бути розробка гібридного підходу, що поєднує швидкість UMDA із точністю GA, що дозволить отримати оптимальні рішення як за швидкістю, так і за якістю розподілу завдань.

Для оцінки ефективності алгоритмів планування у гетерогенних розподілених обчислювальних системах (HDCS) було проведено серію експериментів. Головною метою цих досліджень було визначення, який з алгоритмів – генетичний алгоритм (GA) чи уніваріантний маргінальний розподільчий алгоритм (UMDA) – забезпечує кращий баланс між швидкістю планування, часом очікування задач у черзі та якістю використання доступних обчислювальних ресурсів.

Опис експериментального середовища

Для проведення тестів використовувалася кластерна система, що складалася з трьох серверів з різною конфігурацією обчислювальних потужностей. Кожен кластер містив від 12 до 48 процесорних ядер, що забезпечувало тестування алгоритмів у реальних умовах гетерогенності.

Обчислювальні вузли були з’єднані мережею з високою пропускною здатністю, однак передбачалися мережеві затримки, що відображали реальні умови розподілених систем. Використовувалася черга задач з пріоритетною обробкою критичних шляхів (Critical Path, CP), щільністю графа (Density) та шаруванням (Layering), що дозволяло аналізувати різні сценарії планування.

Для тестування використовувалися як синтетичні задачі, згенеровані спеціальними алгоритмами, так і реальні сценарії, такі як:

– метод LU (LU Decomposition) – розклад матриці на верхню та нижню трикутні матриці;

– метод Гаусса-Жордана – застосовується для розв’язку систем лінійних рівнянь;

– швидке перетворення Фур'є (Fast Fourier Transform, FFT) – широко використовується у цифровій обробці сигналів;

– молекулярна динаміка (Molecular Dynamic Code) – моделює взаємодію атомів у фізичних та біологічних системах.

Висновки з експериментів.

1. UMDA забезпечує значно швидший час виконання, що робить його ідеальним для сценаріїв з високими вимогами до швидкості обчислень.
2. GA ефективніше використовує доступні ресурси, що може бути корисним у випадках, коли кількість процесорів обмежена.
3. Час очікування у UMDA значно менший, що дозволяє використовувати його у завданнях, де важлива швидка реакція системи.
4. UMDA перевершує GA у великих DAG із великою кількістю взаємозалежних підзадач.
5. GA показує кращу стабільність при зміні навантаження, особливо у випадках з обмеженими ресурсами.

Загалом, UMDA є кращим варіантом для систем реального часу, а GA – для довготривалих оптимізаційних задач з обмеженими ресурсами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Zhang, Y., & Wang, Y. (2020). A Hybrid Genetic Algorithm for Task Scheduling in Heterogeneous Distributed Computing Systems. Algorithms, 13(10), 245. DOI: 10.3390/a13100245
2. Li, J., & Chen, H. (2019). An Ant Colony Optimization Approach for Workflow Scheduling in Cloud Computing. Applied Sciences, 9(16), 3201. DOI: 10.3390/app9163201
3. Wang, L., & Zhang, G. (2021). A Particle Swarm Optimization Algorithm for Task Scheduling in Cloud Computing. Electronics, 10(5), 567. DOI: 10.3390/electronics10050567
4. Chen, X., & Liu, Y. (2018). A Hybrid Heuristic Algorithm for Task Scheduling in Heterogeneous Computing Systems. Mathematics, 6(12), 316. DOI: 10.3390/math6120316
5. Liu, F., & Tang, M. (2019). An Improved Genetic Algorithm for Task Scheduling in Cloud Computing. Future Internet, 11(3), 63. DOI: 10.3390/fi11030063

УДК 004.415.3:681.6

А.Л. Ширін[1], А.А. Мартиненко[1], О.С. Шевцова[1], Д.Ю. Буслов[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДОСЛІДЖЕННЯ МАТЕМАТИЧНИХ МОЖЛИВОСТЕЙ ОБРОБКИ ПРИРОДНОЇ МОВИ ТА РІЗНИЦІ У ЗДАТНОСТЯХ ДО ЛОГІЧНОГО МИСЛЕННЯ У РІШЕННІ ПРОСТИХ ЗАДАЧ

Анотація. В роботі розглядається здатність великих моделей обробки природної мови (LLMs) вирішувати складні математичні завдання, що виходять за межі стандартних тестів. Проаналізовано їх ефективність на новому двоступеневому тесті, який вимагає інтеграції кількох математичних концепцій для вирішення комплексних проблем. Результати показують, що, незважаючи на високі досягнення на традиційних тестах, LLMs демонструють значний розрив у результатах при вирішенні завдань, що потребують композиційного підходу.

Ключові слова: *великі мовні моделі, шкільна математика, композиційне мислення, математичні завдання, логічне міркування, тестування LLMs, GSM8K.*

Вступ. У сучасному світі великі мовні моделі (LLMs) дедалі частіше демонструють вражаючі результати в багатьох сферах, включаючи математичне мислення. Вважалося, що такі моделі вже повністю опанували елементарну математику завдяки високій продуктивності на тестах рівня старшої школи. Проте, недавні дослідження завдань зі шкільної математики показали, що, хоча деякі передові моделі незначно підходять під завдання, інші демонструють систематичну переадаптацію, ймовірно через протікання тестового набору [1].

У цій роботі детально показано крихкість здатностей мовної моделі Grade School Math до логічного мислення та оцінено, наскільки добре LLMs можуть поєднувати вивчені концепції для вирішення задач. Щоб краще дослідити як LLMs розуміють математичні концепції та оцінити їхні реальні здібності, у роботі було використано композиційний тест, який складається з двох

пов’язаних питань, де відповідь на перше слугує змінною у другому. Це дослідження спрямоване на перевірку, наскільки LLMs здатні комбінувати вивчені знання для розв'язання більш складних задач.

Основний зміст роботи. Для вирішення простих математичних задач на рівні шкільної математики існуючи мовні моделі використовують Grade School Math DataSet (GSM8K) – це набір даних із 8,5 тисяч високоякісних лінгвістично різноманітних математичних словесних задач для початкової школи. Набір даних сегментується на 7,5 тисяч тренувальних задач і 1 тисячу тестових задач. Для вирішення цих задач потрібно від 2 до 8 кроків, а рішення в основному передбачають виконання послідовності елементарних обчислень з використанням базових арифметичних дій (+ − ×÷) для отримання остаточної відповіді [2].

Для того, щоб оцінити здатність великих мовних моделей до математичного мислення було створено композиційний тест (КТ) на тому ж рівні математичної складності як GSM8K, де кожне завдання складається з двох пов'язаних питань, так що відповідь на перше питання використовується як змінна у другому питанні. У зв'язку з тим, що LLMs можуть легко розв'язувати шкільні математичні завдання, вони також повинні бути спроможні вирішувати їх комбінації. Саме тому було вимірено розрив між виконанням КТ та окремих задач з GSM8K. Зокрема, тестування проводилось на відкритих та закритих LLMs, включаючи сімейства Gemini, Gemma2, LLAMA3 та GP.

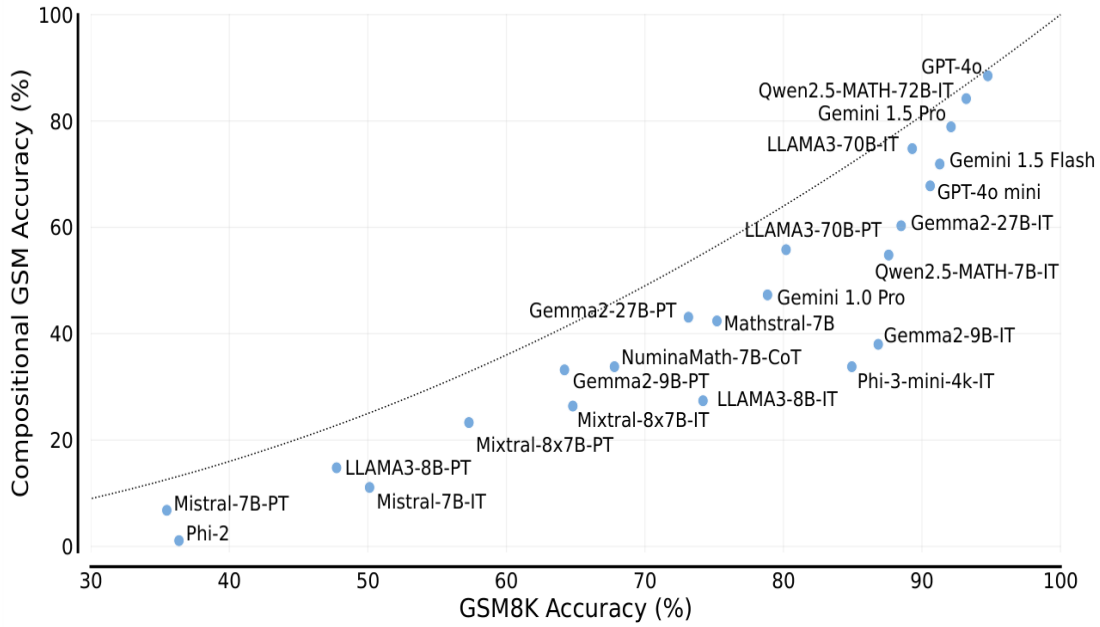


Рис. 1. Результат аналізу існуючих LLM у рішенні простих задач

Більшість моделей продемонстрували помітний розрив показників між КТ та задач з GSM8K (рис.1). Результати проведення аналізу наведено нижче:

– більшість моделей демонструють очевидну різницю між виконанням КТ та задач з GSM8K, що підриває їх надійність та здатність до логічного мислення.

– розрив у логічних можливостях найбільш виражений у менших та більш ефективних моделях, що знижує їх практичну користь.

– налаштування на слідування інструкцій впливає на LLMs різних розмірів по-різному, що вимагає повторного розгляду стандартних тренувальних алгоритмів.

– додаткове навчання з використанням або людських, або синтетичних даних на завданнях GSM8K приводить до специфічної переадаптації завдання з довготривалим навчанням.

Розроблений КТ передбачає аналіз двох запитань: Запитання-1 та Запитання-2, які були відібрані із підмножини оригінального тестового набору GSM8K. Особливістю цього підходу є те, що відповідь на Запитання-1 виступає змінною в Запитанні-2, яку ми позначимо як *X*. Як показано на рис.2, заключна відповідь на Запитання-2 отримується шляхом підстановки *X* та розв'язання цього рівняння. Для досягнення нової остаточної відповіді в Запитанні-2 було змінено певне число так, щоб нова відповідь була різною від попередньої, але залишалася позитивним цілим числом, недалеко розташованим від попередньої відповіді.

Нехай **X** є відповіддю на питання 1:
Q1: У світі залишилось 27 єдинорогів. Третина з них мешкає у Шотландському нагір'ї. Дві третини шотландських єдинорогів - самки. Скільки всього самок шотландських єдинорогів?
Розв'яжіть **Q1** і використовуйте значення **X** для розв'язання **Q2**. Поясніть свою відповідь крок за кроком. **Q2:** Шафка Зака вдвічі менша за шафку Тімоті. Шафка Петра на 1/6 більша за шафку Зака. Якщо шафка Петра має об'єм **X** $м^3$, то який об'єм шафки Тимоті буде в $м^3$?

Рис. 2 – Приклад задачі з композиційного тесту.

Відповіддю на питання *Q1* є змінна *X* у питанні *Q2*. Модель повинна правильно розв'язати перше питання для того, щоб розв'язати друге. Нова остаточна відповідь на питання *Q2* обчислюється шляхом модифікації її коду та його виконання.

Для автоматичного отримання нової остаточної відповіді на питання *Q2* у кодовому розв'язку змінювали на певне число, щоб розподіли остаточних відповідей у модифікованому та оригінальному тестових наборах були схожими за величиною.

Щоб впевнитися в осмисленості та логічності модифікованих запитань, було згенеровано 16 кандидатів-рішень для кожного запитання за допомогою моделей GPT-4o та Gemini 1.5 Pro. Ті запитання, чий результат відповідей у менш ніж 4 випадках із 16 співпав із очікуваним результатом виконання коду, були додатково вичитані вручну та модифіковані (близько 25% запитань). Це дослідження допомогло глибше зрозуміти, як можна адаптувати питання для

підвищення розуміння моделі і надійності у відповідях, що особливо актуально для розробки та тестування інтелектуальних систем.

Наукові результати. Для оцінки здатність великих мовних моделей до математичного мислення було проаналізовано такі моделі, як GPT-4o, GPT-4o mini, LLAMA3-70B та 8B (PT та IT), Phi 2, Phi-3-mini-instruct, Gemini 1.0, 1.5 Flash та 1.5 Pro, Gemma2 9B та 27B (PT та IT), Mistral-7B (PT та IT), Mixtral-8x7B (PT та IT) та спеціалізовані на математиці LLM, включаючи Numina-7B, Mathstral-7B, Qwen2.5-7B та Qwen2.5-72B.

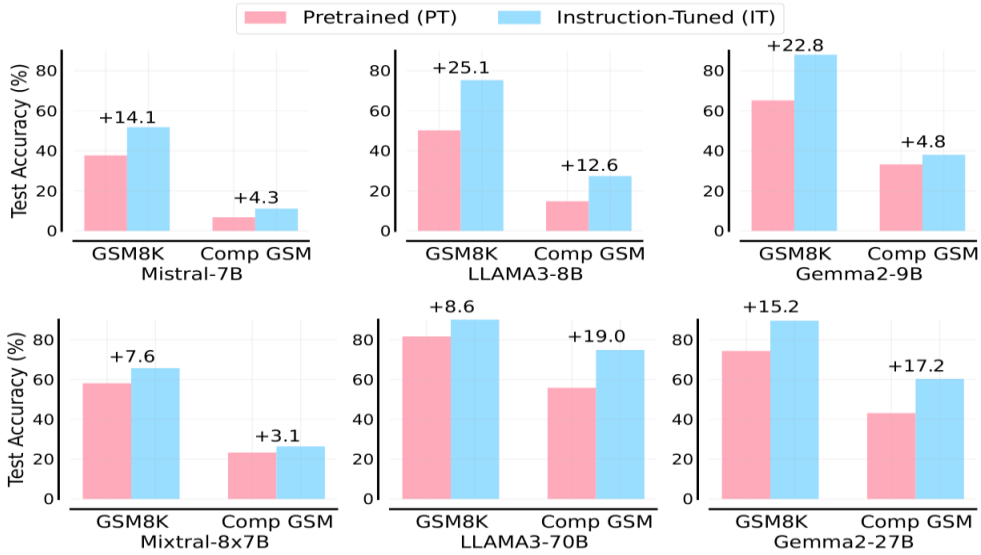


Рис. 3. Вплив налаштування за інструкцією на композиційний тест в великих мовних моделях Mistral, LLAMA3 та Gemma2.

Всі моделі були оцінені при температурі 0, а pass@1 (це метрики, що використовується для оцінювання продуктивності мовних моделей на завданнях програмування або інших завданнях, де потрібно отримати точну відповідь з першої спроби – вона показує, наскільки часто модель успішно розв'язує задачу за першого ж виклику (однієї генерації)) використовувалося для вимірювання продуктивності на кожному тестовому наборі.

Було виявлено, що більшість великих мовних моделей не відповідали очікуванням на КТ, що демонструвало значний розрив у здатності до міркування (рис. 3). Зокрема менші моделі LLM демонстрували суттєвіший розрив у порівнянні з закритими передовими великими мовними моделями.

Висновки. У роботі досліджується здатність великих мовних моделей (LLMs) до математичного мислення за допомогою композиційного тесту, що містить завдання з двома взаємопов'язаними питаннями. Результати показують значний розрив між продуктивністю моделей на стандартному композиційному тесті та задач з набору GSM8K, підкреслюючи виклики в здатності до інтегрованого мислення. Більші моделі, такі як GPT-4o та LLAMA3, демонструють кращі результати, тоді як менші моделі стикаються з труднощами

в адаптації та вирішенні складних задач. Аналіз мовних моделей показав на необхідність вдосконалення тренувальних методик та підходів до генерації рішень для покращення здатності моделей розуміти та вирішувати комплексні математичні завдання.

ПЕРЕЛІК ПОСИЛАНЬ

1. Zhang, H., Da, J., Lee, D., Robinson, V., Wu, C., Song, W., Zhao, T., Raja, P., Slack, D., Lyu, Q., Hendryx, S., Kaplan, R., Lunati, M., and Yue, S. (2024). A careful examination of large language model performance on grade school arithmetic. CoRR, abs/2405.00332. doi: 10.48550/ARXIV.2405.00332. URL: https://doi.org/10.48550/arXiv.2405.00332.
2. Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, Kaiser L, Plappert M, Tworek J, Hilton J, Nakano R, Hesse C. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168. 2021 Oct 27.
3. Srivastava, S., Menon, S., Sukumar, A., Philipose, A., Prince, S., and Thomas, S. (2024). Functional benchmarks for robust evaluation of reasoning performance, and the reasoning gap. CoRR, abs/2402.19450. doi: 10.48550/ARXIV.2402.19450. URL: https://doi.org/10.48550/arXiv.2402.19450.
4. Riviere, G. Team M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., et al. (2024). Gemma 2: Improving open language models at a practical size. arXiv preprint arXiv:2408.00118.
5. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

УДК 004.89:621.311.243

О.В. Помазан[1], В.В. Анісімов[1], А. В. Клименко[2]

[1]Український державний університет науки і технологій, Дніпро, Україна

[2]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ОПТИМАЛЬНЕ РІШЕННЯ РОЗКЛАДКИ ПАНЕЛЕЙ НА ДАХУ ЗА ДОПОМОГОЮ ГЕНЕТИЧНОГО АЛГОРИТМУ

Анотація. Представлено підхід до оптимального розміщення сонячних панелей на дахах із використанням генетичного алгоритму. Алгоритм враховує геометричні обмеження дахів та максимізує використання доступної площі. Отримані результати демонструють ефективність алгоритму, зокрема досягнення заповнення площі до 95% із дотриманням технічних вимог.

Ключові слова: *генетичний алгоритм, оптимізація, сонячні панелі, розміщення панелей, фітнес-функція, поновлювана енергетика, стрінг.*

Вступ. Сучасні виклики енергетичної галузі, зокрема необхідність переходу до екологічно чистих та поновлюваних джерел енергії, вимагають нових технічних рішень [1]. Сонячна енергетика займає провідне місце серед

таких джерел завдяки своїй доступності, екологічності та стрімкому розвитку технологій. Одним із ключових завдань під час проектування сонячних станцій є оптимальне розміщення сонячних панелей, що дозволяє максимально ефективно використовувати доступну площу та забезпечувати необхідні електротехнічні параметри.

Постановка задачі. Мета роботи полягає у розробці алгоритму, який забезпечить оптимальне розміщення сонячних панелей на даху. Задача включає такі підцілі:

1. Максимізація використання площі даху. Панелі мають бути розміщені так, щоб використовувати максимальну доступну площу.
2. Уникнення накладень панелей. Панелі не повинні перекривати одна одну.
3. Формування стрінгів. Панелі повинні бути згруповані у стрінги (послідовності панелей, з'єднаних одна з одною), кожен із яких забезпечує сумарну напругу максимально близьку до оптимальної (наприклад для Deye Sun 6K це 370 В).
4. Адаптація до геометрії даху. Алгоритм повинен враховувати не тільки різні розміри та форми дахів, а і наявність недоступних зон, наприклад, димарі, мансардні вінка та інше.

Використання генетичних алгоритмів для вирішення цієї задачі є перспективним підходом, який поєднує гнучкість і здатність знаходити оптимальні рішення в умовах складних обмежень. Генетичні алгоритми, засновані на принципах природного відбору, дозволяють ітеративно покращувати рішення, враховуючи критерії, що мають найбільший вплив на результат.

У межах даного дослідження було реалізовано генетичний алгоритм для розв'язання задачі оптимального розміщення сонячних панелей на прямокутному даху. Запропонований алгоритм забезпечує формування конфігурації панелей, що максимізує використання площі даху, дотримуючись геометричних обмежень, і формує стрінги з напругою, яка задовольняє вимоги ефективності електричних систем.

Основний зміст роботи.

Для розв'язання задачі було використано генетичний алгоритм [2], який включає такі етапи:

Ініціалізація популяції. Створюється початкова популяція рішень, у кожному з яких панелі розміщуються випадковим чином із врахуванням обмежень. Геномом особини є вектор координат розміщення панелей.

Фітнес-функція. Основний критерій оцінювання рішень включає:

- Загальну площу, яку займають панелі.
- Відсутність перекриттів панелей.
- Відповідність напруги у стрінгах заданим вимогам.

– Штрафи за невиконання обмежень (перекриття, недотримання напруги).

Генетичні оператори.

– Кросовер: комбінування двох рішень для отримання нового [3]. Два геноми (назвемо їх parent1 і parent2) вибираються випадковим чином або з урахуванням їхньої оцінки фітнес-функції (частіше беруть найкращі). Випадковим чином обирається точка, яка визначає, скільки панелей береться від першого батька, а скільки – від другого. Наприклад, якщо точка кросоверу знаходиться після 3-ї панелі, то перші три панелі беруться з parent1, а решта – з parent2. Формується новий геном, який поєднує частину генів (панелей) першого батька з частиною другого. Якщо виникають конфлікти (наприклад, перекриття панелей), то геном коригується, щоб відповідати обмеженням. Результатом є новий геном (child), який може містити переваги обох батьків.

Приклад роботи кросоверу:

Початкові батьки:

Parent1:

[(1, 1), (3, 1), (5, 1), (7, 1)]

(панелі розміщені у рядку)

Parent2:

[(2, 2), (4, 2), (6, 2), (8, 2)]

(панелі розміщені в іншому рядку)

Вибір точки кросоверу:

Наприклад, точка кросоверу – після другої панелі:

З Parent1 береться: [(1, 1), (3, 1)].

З Parent2 береться: [(6, 2), (8, 2)].

Результат:

Child: [(1, 1), (3, 1), (6, 2), (8, 2)].

Панелі комбінуються з двох рішень.

Важливість кросоверу:

1. Кросовер зберігає успішні патерни розташування панелей з обох батьків.

2. У поєднанні з мутацією та відбором кросовер сприяє поступовій оптимізації рішення.

3. Це дозволяє алгоритму "вчитися" від кращих геномів та покращувати адаптацію до вимог задачі.

– Мутація: невеликі випадкові зміни в розміщенні панелей для дослідження нових варіантів.

Еволюція поколінь. Кожне нове покоління створюється шляхом вибору найкращих рішень із попереднього покоління та їхньої комбінації. Процес триває до досягнення стабільного результату або заданої кількості ітерацій.

Фітнес-функція розраховує придатність кожного рішення на основі таких компонентів:

– Площа покриття: чим більша площа даху покрита панелями, тим вищий результат.

– Штрафи за перекриття: накладення панелей одна на одну знижує значення функції.

– Напруга у стрінгах: кожен стрінг повинен забезпечувати напругу, близьку до 370 В. Штрафи застосовуються за недотримання цього параметра.

Завдяки цим компонентам алгоритм спрямований на пошук рішень, які максимізують площу покриття та відповідають технічним вимогам.

Результати та обговорення.

Для візуалізації результатів було використано бібліотеку Matplotlib, що дозволяє оцінити прогрес алгоритму у кожному поколінні.

Проведені симуляції на дахах різних розмірів і конфігурацій показали такі результати:

1. У середньому 85–90% площі даху було ефективно використано для розміщення панелей.
2. Панелі організовані у стрінги, що відповідають технічним вимогам щодо напруги.
3. Алгоритм успішно адаптується до унікальних геометричних характеристик дахів, генеруючи унікальні рішення для кожного випадку.
4. Візуалізація процесу демонструє поступове покращення рішень у кожному поколінні.

На рисунку 1 показано приклад оптимального розміщення панелей на прямокутному даху з розмірами 10 х10 м.

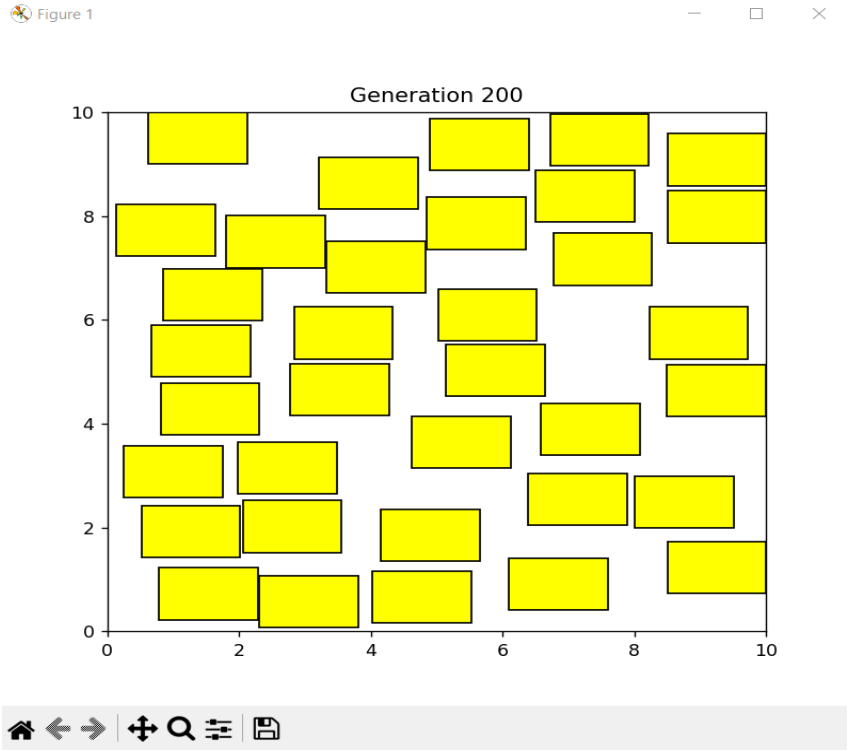


Рис. 1. Результат роботи генетичного алгоритму

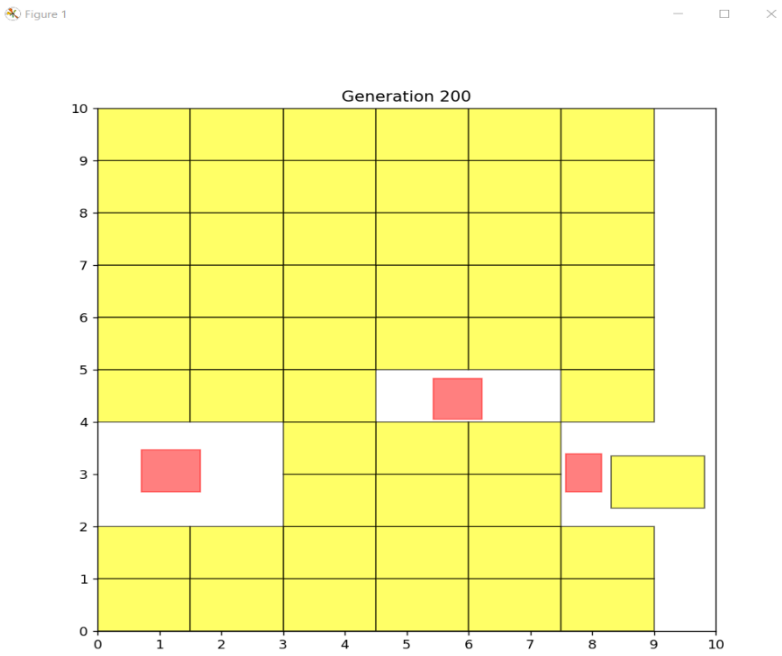


Рис. 2. Покращений результат роботи генетичного алгоритму

Завдяки доопрацюванню програми та впровадженню додаткових обмежень вдалося покращити якість отриманих рішень (Рисунок 2). Зокрема:

– Врахування реальних обмежень даху дозволило уникнути розміщення панелей у непридатних для цього зонах, таких як вентиляційні шахти або виступи.

– Організація панелей у стрінги забезпечила дотримання вимог щодо напруги (наближення до 370 В) та покращила ефективність електричного з'єднання.

– Оптимізація використання площі даху дозволила досягти заповнення до 90–95%, що перевищує попередні результати.

Висновки. Розроблений генетичний алгоритм вирішує задачу оптимального розміщення сонячних панелей на дахах із врахуванням технічних обмежень. Отримані результати підтверджують його придатність для використання у реальних умовах. Подальша робота спрямована на вдосконалення фітнес-функції, розширення алгоритму для складніших геометричних конфігурацій та інтеграцію з іншими системами планування сонячних електростанцій.

ПЕРЕЛІК ПОСИЛАНЬ

1. Goldberg D.E. "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley, 1989.
2. Michalewicz Z. "Genetic Algorithms + Data Structures = Evolution Programs". Springer, 1996.
3. Штучний інтелект у задачах оптимізації. Навчальний посібник / [В. І. Петренко, О. М. Смірнов]. – Київ: Техніка, 2020.

УДК 004.89

М.Г. Бердник[1], І.П. Стародубський[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

АВТОМАТИЧНА АДАПТАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДО ХМАРНИХ ТА ПЕРИФЕРІЙНИХ ПЛАТФОРМ МЕТОДАМИ МАШИННОГО НАВЧАННЯ

Анотація. У даній роботі розроблені методи автоматичної адаптації програмного забезпечення для хмарних та периферійних платформ із використанням машинного навчання. Реалізовано методи, що базуються на наглядовому та підкріплювальному навчанні, які дозволяють ефективно керувати розподілом ресурсів і навантаженням. Продемонстровано переваги ML-адаптації у підвищенні продуктивності та гнучкості обчислювальних систем.

***Ключові слова**: автоматична адаптація, машинне навчання, хмарні обчислення, периферійні обчислення, підкріплювальне навчання, оптимізація ресурсів.*

Сучасні програмні системи дедалі частіше працюють у хмарних та периферійних обчислювальних середовищах, які характеризуються високою динамічністю. Навантаження на сервіси може коливатися, можливі збої, а доступні ресурси постійно змінюються. Для забезпечення надійної роботи в таких умовах програмне забезпечення повинно адаптуватися – змінювати кількість виділених ресурсів, масштабувати компоненти або мігрувати між вузлами хмари і периферії залежно від ситуації. Визначити, коли і як саме слід виконати адаптацію, є нетривіальною задачею, оскільки традиційні платформи покладаються на прості порогові правила (наприклад, авто-масштабування AWS). Особливо складною є адаптація у гетерогенному середовищі "хмара–периферія", де потрібно вирішувати, на якому рівні (центральному або периферійному) виконувати ті чи інші компоненти системи.

В цій роботі було розроблено систему автоматичної адаптації, що складається з кількох основних компонентів:

1. Модуль прогнозування навантаження, що реалізований на основі рекурентних нейронних мереж (LSTM), та дозволяє аналізувати історичні тренди навантаження та прогнозувати можливі сплески активності. Реалізовано механізм динамічного коригування моделей на основі даних в реальному часі. Це забезпечило стабільну реакцію системи на змінні умови.

2. Система прийняття рішень на основі підкріплювального навчання: Впроваджено нейромережевий агент на основі алгоритму Deep Q-Network (DQN), який навчався на реальних даних обчислювальної інфраструктури та розробив ефективну стратегію міграції навантаження. Реалізовано систему багатокритеріальної оцінки продуктивності для оптимізації використання процесорного часу та енергоспоживання. Адаптовано алгоритм PPO (Proximal Policy Optimization) для підвищення стійкості рішень у нестабільних середовищах.

3. Онлайн-адаптація, де використано методику Transfer Learning для оновлення моделі без необхідності її повного перенавчання. Додано систему самооновлення моделей на основі потокового аналізу даних, що зменшує ризик деградації якості передбачень. Реалізовано механізм «екстреного перемикання» алгоритмів у разі виявлення несподіваних аномалій у роботі інфраструктури. Це дозволяє системі зберігати знання, отримані на попередніх даних, та швидко адаптуватися до змін у середовищі.

4. Інтеграція з реальними обчислювальними середовищами: система була розгорнута на кластері Kubernetes, що поєднує периферійні вузли та хмарну платформу AWS. Впроваджено автоматичний балансувальник навантаження для рівномірного розподілу завдань між периферійними вузлами. Моніторинг продуктивності здійснювався через Prometheus, а результати масштабування аналізувалися у Grafana.

Результати експериментів (рис. 1) підтвердили ефективність реалізованої системи автоматичної адаптації програмного забезпечення.

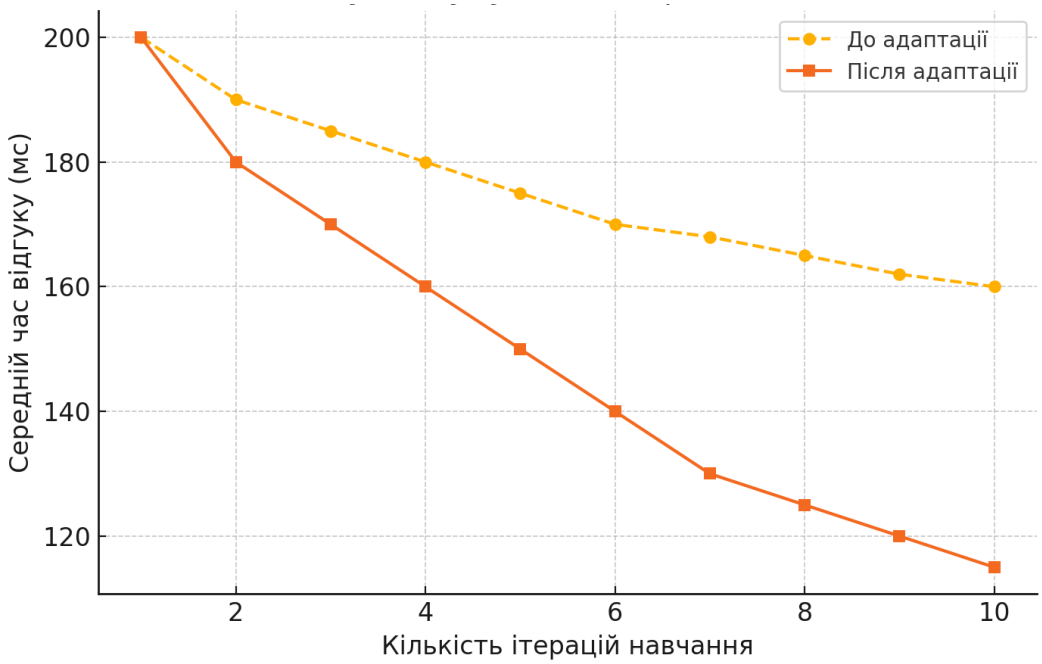


Рис. 1. Час відгуку після впровадження адаптації.

Впровадження методів машинного навчання дозволило значно покращити продуктивність хмарно-периферійних систем та оптимізувати використання обчислювальних ресурсів.

УДК 004.932:528.854

С.Л. Нікулін[1], В.Ю. Каштан[1], О.В. Коробко[1], К.А. Сідаш[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

КОМПЛЕКСНЕ ВИКОРИСТАННЯ МЕТОДІВ K-MEANS КЛАСТЕРИЗАЦІЇ ТА ARIMA ДЛЯ МОДЕЛЮВАННЯ ТЕНДЕНЦІЙ СМЕРТНОСТІ В ЄВРОПІ ВІД ІНФЕКЦІЙНИХ ЗАХВОРЮВАНЬ

Анотація. Представлено інтеграцію методів K-means кластеризації та ARIMA для комплексного аналізу та прогнозування смертності в Європі. Цей підхід дозволяє отримати нові знання про вплив різних факторів на смертність, а також покращити точність прогнозування за рахунок використання сучасних статистичних моделей і алгоритмів машинного навчання.

***Ключові слова**: K-means кластеризація, ARIMA, моделювання, прогнозування, кластеризація даних, інфекційні захворювання.*

Вступ. Інфекційні хвороби, як старі, так і новітні, є однією з найбільших загроз для здоров'я людей у всьому світі. Вони є причиною значної кількості смертей, особливо серед дітей, та становлять серйозну проблему для систем охорони здоров'я багатьох країн. Поява нових, висококонтагіозних вірусів, таких як COVID-19, підкреслює актуальність цієї проблеми.

За даними Світового банку, 50% випадків смерті дітей віком до 5 років у світі спричинені інфекційними захворюваннями (патологія органів дихання, кишкові інфекції, кір, малярія, СНІД та інші), а у структурі захворюваності у цій віковій групі інфекційні хвороби становлять 80%, [1]. За даними Всесвітньої організації охорони здоров’я (ВООЗ), смертність внаслідок інфекційних хвороб у деяких країнах світу посідає друге місце у структурі загальної смертності. Наприкінці XX століття і на початку XXI століття загострилась ситуація з таких широко відомих захворювань, як чума, холера, жовта лихоманка, з’явилось більше 30 нових раніше невідомих, але небезпечних хвороб: високо контагіозні COVID-19, геморагічні лихоманки Ласса, Ебола, Марбург; ВІЛ-інфекція, легіонельоз, пташиний та свинячий грип та інші, [2]. Інфекційні хвороби, незважаючи на досягнення сучасної медицини, залишаються надзвичайно важливою медико-соціальною та економічною проблемою для суспільства. Зокрема, це є вкрай актуальним для України, особливо у скрутні часи економічної кризи, коли питання контролю епідемічної ситуації, управління інфекційною захворюваністю під час пандемії COVID-19 у державі набуває статусу національної безпеки.

Метою роботи є розробка комплексного підходу до аналізу та прогнозування тенденцій смертності в Європі за допомогою поєднання методів K-means кластеризації для виявлення груп та ARIMA для прогнозування часових рядів, що дозволяє більш точно оцінити динаміку смертності на основі різних статистичних і аналітичних підходів.

Постановка задачі. Актуальною науково-прикладною задачею для досягнення мети роботи є прогнозування тенденцій смертності в Європі від інфекційних захворювань. Для цього необхідно вирішити такі завдання:

– зібрати та підготувати дані про смертність в Європі, зокрема щодо інфекційних захворювань, виконати попередній аналіз для виявлення аномалій;

– застосувати метод кластеризації K-середніх для аналізу рівнів смертності та виявлення однорідних груп населення;

– розробити інтегровану методологію прогнозування тенденцій смертності із використанням K-means кластеризації та моделі ARIMA.

Вихідні дані. Історичні дані про смертність населення від інфекційних захворювань. Дані взяті з відкритих джерел, таких як офіційні статистичні агентства, наукові бази даних або медичні звіти. Інформація охоплює різні часові періоди, регіони, вікові групи та інші фактори, що впливають на смертність. Після завантаження даних була проведена їх структуризація та очищення від пропусків, дублікатів або аномальних значень. Цей етап критично важливий для забезпечення надійності подальших етапів аналізу, [3].

Методи досліджень. Сучасні методи аналізу смертності від інфекційних захворювань часто не враховують багатофакторного впливу, що обмежує точність прогнозування та ускладнює розробку ефективних стратегій реагування. Більшість підходів зосереджуються лише на певних аспектах, таких як часові чи географічні чинники, але не забезпечують цілісного підходу до аналізу. Існує потреба у створенні інтегрованих методів, які одночасно враховують різні фактори смертності та дозволяють проводити більш точні прогнози з використанням сучасних статистичних моделей і алгоритмів машинного навчання.

Методологія дослідження включає застосування K-means кластеризації та моделі ARIMA для аналізу та прогнозування тенденцій смертності. Структурна схема методології представлена на рис.1.

На першому етапі здійснюється завантаження історичних даних про смертність населення від інфекційних захворювань. Дані можуть бути взяті з відкритих джерел, таких як офіційні статистичні агентства, наукові бази даних або медичні звіти. Інформація повинна охоплювати різні часові періоди, регіони, вікові групи та інші фактори, що впливають на смертність.

Далі дані кластеризуються за допомогою алгоритму K-means, що автоматично створює три класи смертності: низька, середня і висока. Цей процес дозволяє групувати країни Європи за подібними рівнями смертності. Основні етапи включають: Вибір кількості кластерів (k=3), що відповідає трьом класам: низька, середня та висока смертність. Алгоритм випадковим чином визначає початкові центроїди для кожного з трьох кластерів.

Обчислення відстані кожного спостереження до найближчого центроїда і призначення його до відповідного класу. Після кількох ітерацій алгоритм визначає стабільні класи, що дозволяє розподілити дані на три категорії.

На третьому етапі на основі кластеризованих даних для кожного класу будується модель ARIMA. Вона допомагає прогнозувати майбутні зміни рівнів смертності в кожному з трьох класів. Перед побудовою моделі здійснюється перевірка часових рядів на стаціонарність за допомогою тесту Дікі-Фуллера. У разі виявлення нестаціонарності дані трансформуються (шляхом диференціювання або інших методів), щоб забезпечити виконання умов стаціонарності. Це дозволяє коректно застосувати модель ARIMA для прогнозування майбутніх змін рівнів смертності в кожному з трьох класів.

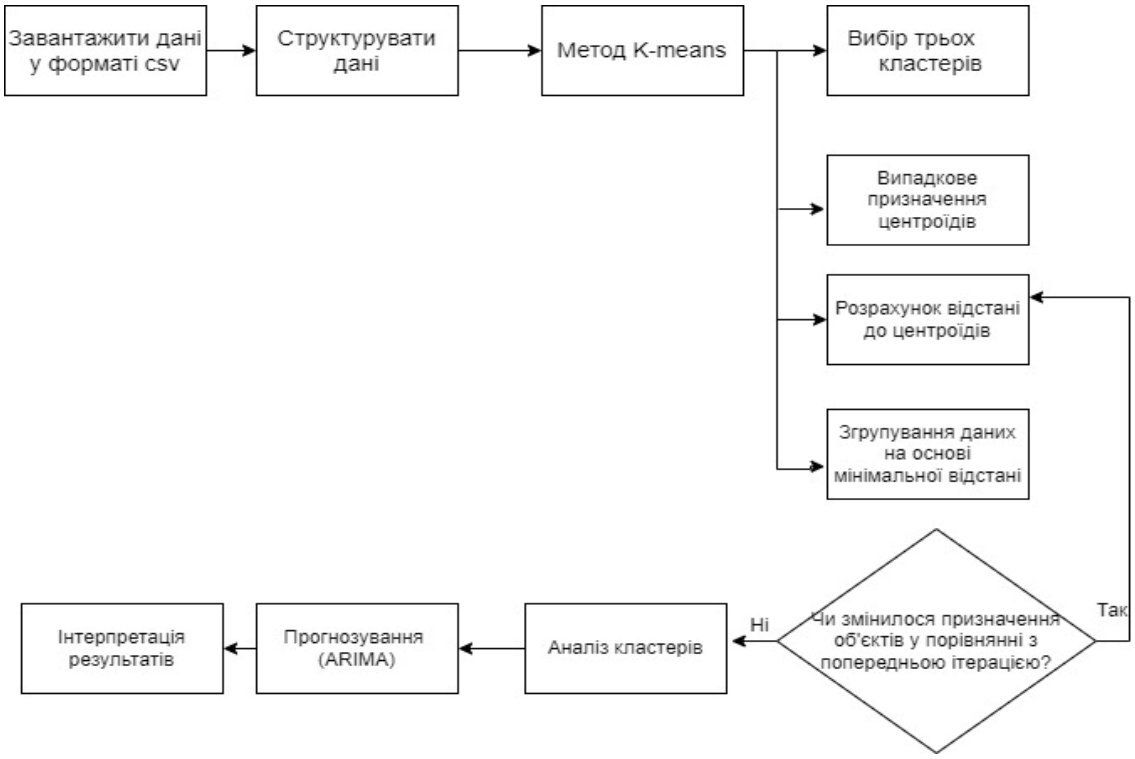


Рис. 1. Структурна схема методології прогнозування смертності

Після кластеризації та прогнозування здійснюється інтерпретація отриманих результатів. Кожен із трьох класів аналізується окремо для виявлення факторів, що впливають на рівень смертності в цих групах. Прогнози аналізуються та порівнюються з реальними даними для оцінки точності моделі.

Обговорення результатів. Аналіз результатів показав, що комбіноване використання методів K-means та ARIMA продемонструвало позитивні результати в оцінці та прогнозуванні показників смертності. Зниження значення середньої квадратної помилки (MSE) свідчить про підвищення точності прогнозування, а покращення метрики силуету вказує на кращу якість кластеризації.

Визначено, що метод ARIMA, що враховує часову залежність, в поєднанні з K-means, який ідентифікує подібності в даних, забезпечує більш точні та надійні прогнози, (табл. 1).

Таблиця 1.

Результати оцінки метрик для ARIMA та K-means

Метод	Параметри	MSE	Силует
ARIMA	(p=1, d=1, q=1)	23.45	N/A
K-means	k=3	N/A	0.45
Комбінований	ARIMA + K-means	19.32	0.50

Таким чином, інтеграція методів K-means кластеризації та ARIMA для комплексного аналізу та прогнозування смертності в Європі. дозволяє отримати нові знання про вплив різних факторів на смертність, а також покращити точність прогнозування за рахунок використання сучасних статистичних моделей і алгоритмів машинного навчання. Дослідження також підкреслює важливість географічного контексту у вивченні проблеми смертності.

Висновки. В роботі визначено та оцінено наявні моделі смертності для розуміння впливу різних факторів на показники смертності населення, зокрема у контексті інфекційних захворювань.

Проаналізовано методи кластеризації, такі як K-means, для визначення закономірностей у даних та поділу населення на гомогенні групи. Використано ARIMA для прогнозування смертності з урахуванням часових залежностей.

Проведено оцінку точності прогнозів за допомогою метрик, таких як середня квадратна помилка (MSE) для точності прогнозування та метрика силуета для оцінки якості кластеризації.

Отримані результати підтверджують доцільність використання комплексного підходу до аналізу та прогнозування часових рядів, що дозволяє приймати обґрунтовані рішення на основі даних.

.

ПЕРЕЛІК ПОСИЛАНЬ

5. Закон України «Про захист населення від інфекційних хвороб» від 06.04.2000 р. No 1645-III. [Електронний ресурс]. Режим доступу: zakon.rada.gov.ua.
6. Андрейчин М.А. Біологічна зброя і біотероризм / М.А. Андрейчин, В.С.Копча. Здоров’я України: медична газета. 2004. No 5. С. 40–41.
7. Статистичні дані [Електронний ресурс] Режим доступу: https://catalog.data.gov/dataset/?tags=deaths&res_format=CSV.
8. Onambele, Luc & Guillen-Aguinaga, Sara & Guillen-Aguinaga, Laura & Ortega-Leon, Wilfrido & Montejo, Rocio & Alas-Brun, Rosa & Aguinaga-Ontoso, Enrique & Aguinaga-Ontoso, Ines & Guillen-Grima, Francisco. (2023). Trends, Projections, and Regional Disparities of Maternal Mortality in Africa (1990–2030): An ARIMA Forecasting Approach. Epidemiologia. 4. 322-351. 10.3390/epidemiologia4030032.
9. Doukhan, P., Pommeret, D., Rynkiewicz, J., and Salhi, Y. (2017). A class of random eld memory models for mortality forecasting. Insurance: Mathematics and Economics 77, pp. 97-110. doi: 10.1016/j.insmatheco.2017.08.010.
10. Human Mortality Database (2019). University of California, Berkeley (USA), and Max Planck Institute for Demographic Research (Germany). Available at www.mortality.org or www.humanmortality.de (data downloaded on 2019-01-28).

РОЗДІЛ 5

ІНФОРМАЦІЙНА БЕЗПЕКА

УДК 004.056:004.94

В.І. Корнієнко[1], Д.О. Савченко[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РОЗРОБКА МЕТОДИКИ ВИЯВЛЕННЯ ЗАГРОЗ У РЕАЛЬНОМУ ЧАСІ НА ПЛАТФОРМІ KUBERNETES ЗА ДОПОМОГОЮ СИСТЕМИ FALCO

Анотація. Досліджено загрози для інформаційної безпеки контейнерних середовищ на платформі Kubernetes. Розроблено методику виявлення загроз у реальному часі за допомогою системи Falco за рахунок активації в ній аналізу ризиків та створенню правил для виявлення аномалій. Проведено тестування методики у змодельованих умовах для забезпечення ефективного реагування на кіберзагрози.

Ключові слова: система Falco, платформа Kubernetes, контейнеризація додатків, хмарні технології, моніторинг безпеки, кібербезпека.

Вступ. У сучасному світі інформаційних технологій зі зростанням популярності контейнеризації та мікросервісної архітектури, інформаційна безпека стала ключовим пріоритетом. Платформа Kubernetes [1], як провідна система організації контейнерів з відкритим вихідним кодом, забезпечує масштабованість, гнучкість та ефективність в управлінні контейнерними додатками. Водночас, зростання складності цієї системи вимагає підвищеного рівня кібербезпеки шляхом її моніторингу для своєчасного виявлення і реагування на інциденти.

Особливу роль у забезпеченні безпеки займає розробка ефективних методів виявлення загроз у реальному часі. Система Falco [2] – це інструмент з відкритим вихідним кодом, пропонує потужні можливості для моніторингу поведінки контейнерних додатків і виявлення аномалій у Kubernetes. Таким чином, це дослідження спрямоване на розробку методології використання системи Falco для забезпечення високого рівня захисту платформи Kubernetes від сучасних кіберзагроз.

Постановка задачі. Ключовим завданням є розробка методики виявлення потенційних загроз та аномалій в реальному часі в середовищі Kubernetes з використанням інструменту безпеки Falco, що забезпечить більшу автоматизацію в обробці безпекових подій і спростить адміністрування безпеки кластерів.

Для цього необхідно: розглянути архітектуру оркестратора платформи Kubernetes і системи Falco; визначіти критерії та параметри для виявлення загроз, що базуються на реальному середовищі використання Kubernetes та розробити сценарії для Falco, що відповідатимуть цим критеріям. Це потребує розгляду процесів встановлення та налаштування Falco у середовищі Kubernetes, інтеграції з існуючими інструментами моніторингу та сповіщення, а також проведення серії тестів для перевірки ефективності розробленої методики.

Основний зміст роботи. На основі глибокого аналізу контейнерних середовищ у Kubernetes у якості основних загроз інформаційної безпеки визначені: вразливі контейнерні образи, незахищені API, неправильні конфігурації кластерів та недостатній контроль доступу. Це дозволяє обґрунтовано вибрати систему Falco як ключовий інструмент для моніторингу поведінки контейнерів у реальному часі, яка інтегрується з Kubernetes для забезпечення проактивного виявлення загроз.

На основі детального аналізу архітектури Kubernetes визначено, що ключовими є компоненти API Server, etcd, Kubelet, Kube-Proxy і Scheduler [3], які забезпечують функціонування кластера, але можуть стати потенційними точками атаки. Для протидії цьому доцільно використання мережевих політик, RBAC (контроль доступу на основі ролей) і шифрування для захисту кластерів.

Архітектура системи Falco (рис. 1) включає модуль ядра та бібліотеки Libscap і Libsinsp, які забезпечують перехоплення системних викликів та їх аналіз.

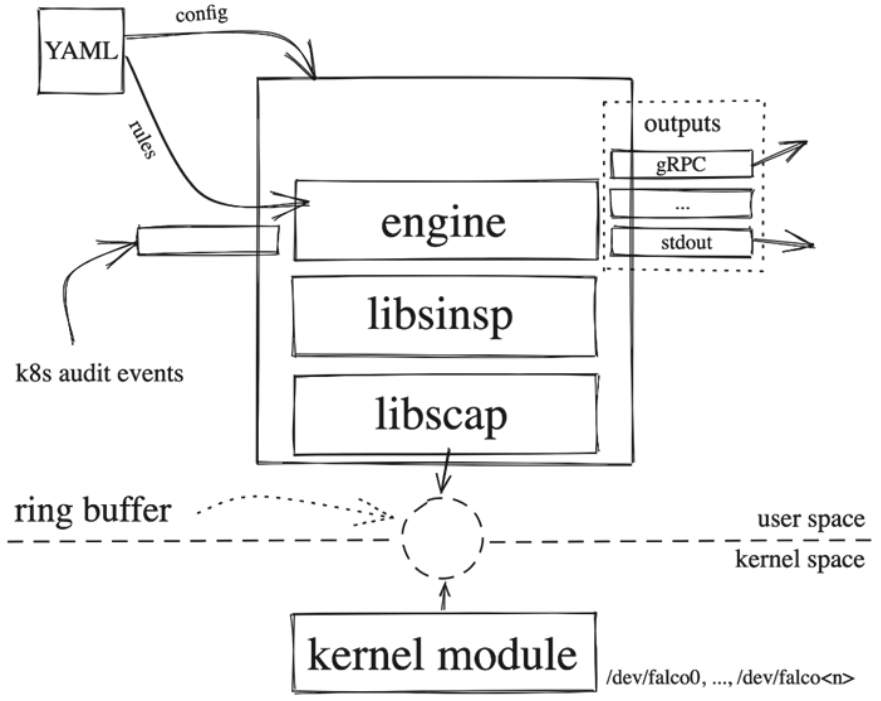


Рис 1. Архітектура системи Falco

Система Falco дозволяє створювати кастомні правила для виявлення ненормальної поведінки, наприклад, спроб несанкціонованого доступу до системних файлів або виконання небезпечних команд.

Проведено серію тестів у змодельованому середовищі платформи Kubernetes для оцінки ефективності системи Falco. Створено кілька сценаріїв загроз, включаючи несанкціонований доступ до файлів системи, атаки через вразливі API та спроби підвищення привілеїв. У всіх випадках Falco своєчасно виявляв аномалії та генерував сповіщення з різними рівнями пріоритету. Інтеграція із корпоративним месенджером та системою сповіщення Slack забезпечила швидке інформування команди реагування про інциденти (рис. 2).

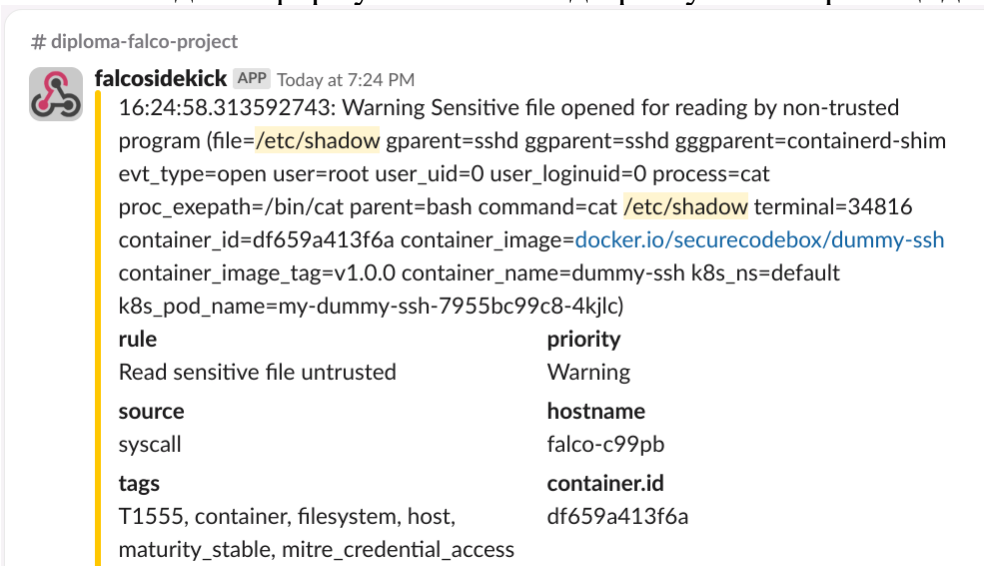


Рис 2. Повідомлення системи Falco про загрозу безпеці у Slack

Також створено механізм інформування, який дозволяє відправляти детальні сповіщення з контекстною інформацією про загрозу. Це допомагає фахівцям швидко оцінювати ситуацію та приймати рішення щодо нейтралізації інформаційних загроз.

Наукова новизна роботи полягає у підвищенні кібербезпеки IT-додатків шляхом розробки та використання методики виявлення загроз у реальному часі для платформи Kubernetes з використанням системи Falco. Запропонована методика забезпечує високий рівень ймовірності виявлення аномалій завдяки поєднанню поведінкового моніторингу та адаптивних правил, що дозволяє оперативно реагувати на сучасні кіберзагрози. На відміну від існуючих підходів, дослідження пропонує інтеграцію системи Falco з системами сповіщення, що підвищує ефективність управління безпекою у динамічних контейнеризованих середовищах.

Висновки. У роботі проведено аналіз загроз безпеки, що впливають на стан захисту контейнеризованих середовищ на платформі Kubernetes, визначено основні вимоги до забезпечення безпеки. Обґрунтовано вибір системи Falco як інструмента моніторингу та виявлення загроз у реальному часі. Розроблено та

протестовано набір правил для виявлення аномалій, що забезпечує оперативне реагування на інциденти безпеки.

Проведені дослідження підтвердили ефективність використання системи Falco у поєднанні з системами сповіщення для підвищення рівня кібербезпеки контейнерних додатків. Основними складовими розробленої методики є аналіз поведінки контейнерів, створення адаптивних правил та інтеграція з механізмами інформування. Це дозволяє знизити ризики кіберзагроз і забезпечити надійний захист IT-інфраструктури в умовах динамічних середовищ Kubernetes.

ПЕРЕЛІК ПОСИЛАНЬ

1. Що таке Kubernetes? Офіційний сайт проєкту Kubernetes: [Електронний ресурс] – Режим доступу до ресурсу: https://kubernetes.io/docs/concepts/overview/
2. Що таке Falco? Офіційний сайт проєкту Falco: [Електронний ресурс] – Режим доступу до ресурсу: https://falco.org/about/
3. Архітектура Kubernetes. Офіційний сайт проєкту Kubernetes: [Електронний ресурс] – Режим доступу до ресурсу: https://kubernetes.io/docs/concepts/architecture/

УДК 004.056.5:004.71

С.М. Грось[1], Д.С. Тимофєєв[1]

[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

АНАЛІЗ ВРАЗЛИВОСТЕЙ ПРОТОКОЛУ BGP

Анотація. Наведено загальний аналіз вразливостей протоколу BGP, актуальні засоби захисту і запропоновані рекомендації з протидії вразливостям. Представлено реальні інциденти, що підкреслюють важливість покращення безпеки маршрутизації для стабільності інтернету.

Ключові слова: *мережа, протокол, маршрутизація, трафік, вразливість, криптографія, цифровий підпис, інцидент.*

Вступ. Глобальна мережа об'єднує мільярди пристроїв по всьому світу, забезпечуючи передавання даних, комунікацію, комерцію, навчання, розваги та багато іншого. Тому стабільність інтернету є критично важливою для безперервної роботи цих сервісів. Для забезпечення стабільної та надійної роботи необхідна ефективна маршрутизація трафіку між мільйонами мереж. Протокол BGP (Border Gateway Protocol) є основним засобом, який керує маршрутизацією між автономними системами, що є основою глобальної інфраструктури інтернету.

Постановка задачі:

- розглянути принцип роботи протоколу BGP;
- проаналізувати актуальні вразливості пов’язані з протоколом BGP;

- проаналізувати засоби захисту від вразливостей;
- представити рекомендації з протидії вразливостям.

Основний зміст роботи. BGP (Border Gateway Protocol) - це протокол маршрутизації, який використовується для обміну інформацією про маршрути між різними автономними системами. Кожна автономна система є своєрідною мережею або групою мереж, які мають спільний політичний або адміністративний контроль. Протокол забезпечує обмін інформацією про доступні маршрути між системами і дозволяє вибирати найбільш оптимальні для передачі даних. BGP працює на прикладному рівні та є вектором шляхів (path vector protocol), що означає, що маршрутизатори BGP обмінюються списками шляхів до різних мереж і вибирають найбільш ефективні, залежно від різних параметрів, таких як довжина шляху, довжина маски та інші атрибути.

Типи BGP [1].

iBGP (internal BGP) - використовується в межах однієї автономної системи для обміну маршрутами між маршрутизаторами.

eBGP (external BGP) - використовується для обміну маршрутами між автономними системами, а саме між різними мережами або провайдерами інтернету.

Механізм роботи BGP. BGP використовує кілька ключових атрибутів для обрання маршруту [1]:

- AS_PATH - атрибут, який містить список автономних систем, через які пройшов маршрут. Чим довший список, тим менш бажаний цей шлях, оскільки це означає більшу кількість проміжних точок і тим самим більше час затримки;
- NEXT_HOP - вказує адресу маршрутизатора, через який потрібно передавати пакети;
- LOCAL_PREF - пріоритет маршруту всередині автономної системи, що дозволяє адміністраторам налаштовувати пріоритет одного маршруту перед іншим;
- MED (Multi-Exit Discriminator) - використовується для визначення пріоритету між кількома виходами з однієї автономної системи до іншої;
- COMMUNITY - додатковий атрибут, який дозволяє групувати маршрути та застосовувати правила для певних груп мереж.

BGP також використовує оновлення (UPDATE) для повідомлення про нові або змінені маршрути, а також для відправлення повідомлень про недоступні маршрути.

BGP забезпечує роботу глобальної маршрутизації, що дозволяє мережам по всьому світу передавати дані один з одним. Наприклад, якщо користувач з України хоче звернутися до вебсайту з Німеччині, BGP використовує маршрути, щоб визначити найкоротший шлях, через який дані повинні пройти, щоб досягти свого кінцевого пункту. Це може включати маршрути через кілька різних автономних систем.

BGP забезпечує масштабованість і гнучкість інтернету, дозволяючи новим мережам без труднощів підключатися до глобальної мережі і автоматично визначати оптимальні шляхи для передачі трафіку. Завдяки цьому, протокол може підтримувати величезну кількість маршрутів, оскільки таблиці маршрутизації в BGP можуть включати мільйони записів.

Аналіз реальних інцидентів із BGP.

Хайджек YouTube з боку Пакистану (2008) [2]. У 2008 році компанія Pakistan Telecom (PTCL) випадково заблокувала доступ до YouTube в усьому світі, намагаючись заблокувати платформу лише на території Пакистану. PTCL використала BGP для локального блокування, але помилково передала цей маршрут по міжнародних каналах, що спричинило значне глобальне відключення YouTube для користувачів по всьому світу. Цей випадок демонструє, як BGP-оголошення для локального контролю можуть мати серйозні наслідки, якщо їх обмежити не належним чином

Хайджек криптовалюти MyEtherWallet (2018) [3]. Під час цілеспрямованого нападу на криптогаманець MyEtherWallet, хакери перенаправили DNS-запити користувачів на зловмисні сервери через переналаштування маршрутів BGP. Це призвело до того, що користувачі підключалися до підробленого сайту, через що втрачали свої кошти. Інцидент підкреслює небезпеку BGP-хайджеків у фінансовому секторі, де перехоплення трафіку може призвести до величезних втрат.

Пролонговані перебої через уразливість BGP [4]. У червні 2023 року, дослідник з BGP.Tools виявив нову атаку, яка дозволяє створювати BGP UPDATE-повідомлення з модифікованими атрибутами. Ця атака могла спричиняти збій в певних маршрутизаторах при передачі пошкоджених повідомлень, при цьому відключаючи мережі від інтернету. Це сталося, коли невеликий бразильський провайдер поширив некоректний маршрут, що спричинив значні перебої в інших мережах.

Ці інциденти демонструють, що BGP все ще має критичні вразливості, які можуть вплинути на безпеку та стабільність глобальної мережі.

Аналіз вразливостей протоколу.

BGP є потужним і ефективним протоколом, проте він має декілька проблем, які можуть призвести до порушення стабільності мережі:

- цикли маршрутизації: якщо маршрутизатори передають один одному непра- вильні або застарілі маршрути, можуть виникати цикли, коли пакети без кінця передаються між маршрутизаторами, не досягаючи призначення. Зловмисники можуть спеціально вводити некоректні маршрути, створюючи цикли для атаки на доступність (DoS-атака);
- маршрутизація за замовчуванням: у випадку, коли маршрутизатор не має точної інформації про маршрут до призначеного місця, він може направити трафік на загальний, менш ефективний маршрут, що може спричинити затримки або втрату пакетів. Як приклад, зловмисник може перенаправляти трафік через

свій вузол, що створює ризик перехоплення або модифікації даних (атака "людина посередині").;

- масштабованість: з кожним роком інтернет зростає, в результаті кількість автономних систем збільшується, що призводить до зростання таблиць маршрутизації. Це навантажує маршрутизатори, що може призвести до збільшення часу обробки і затримки та збільшення ризику відмови або інших помилок. Зловмисник може навмисно додавати велику кількість маршрутів (атака BGP Table Overflow), щоб маршрутизатор вийшов з ладу;
- відсутність валідації маршрутів: BGP не має надійного механізму для перевірки достовірності маршрутів, тому некоректні або зловмисні маршрути можуть бути прийняті, що спричиняє перенаправлення трафіку. Зловмисник може використати атаку BGP Hijacking для перенаправлення або крадіжки трафіку;
- залежність від централізованих джерел: залежність від RPKI (Resource Public Key Infrastructure) для перевірки маршрутів може призвести до збоїв або компрометації RPKI, які в свою чергу можуть спричинити масштабні проблеми з маршрутизацією. Зловмисник може атакувати RPKI для введення неправдивих маршрутів або позбавлення легітимних вузлів доступу до мережі.

Аналіз механізмів захисту протоколу.

Протокол BGP, хоч і є основним для маршрутизації в інтернеті, але має декілька серйозних вразливостей. Для того, щоб зменшити ризики і забезпечити стабільність мережі, були розроблені різні методи захисту. Однак, навіть з цими механізмами захисту, питання безпеки залишаються досить актуальними.

BGPSEC (BGP Security Extensions) [5]. BGPSEC - це розширення протоколу BGP, яке включає криптографічну автентифікацію для кожного маршруту, що дозволяє перевіряти, чи є визначені маршрути автентичними та чи не були вони змінені. Кожен анонс маршруту в BGPSEC підписується за допомогою цифрового підпису, що в свою чергу дозволяє забезпечити надійність маршруту.

Механізм роботи BGPSEC: для кожної автономної системи в мережі забезпечується цифровий підпис для всіх маршрутів, що вона оголошує. Маршрутизатори в процесі обміну інформацією можуть перевірити підпис кожного маршруту на автентиність і впевнитися, що він дійсно походить від відповідної автономної системи. Тому ця криптографічна перевірка сприяє запобіганню атакам типу BGP hijacking, оскільки зловмисник не може просто оголосити фальшивий маршрут, не маючи відповідного підпису.

RPKI (Resource Public Key Infrastructure) [6]. RPKI є системою для автентифікації маршрутів, яка дозволяє верифікувати право власності на IP-простори та автономні системи, які оголошують маршрути через BGP.

Принцип роботи RPKI: кожен провайдер інтернету реєструє свої IP-діапазони і автономні системи в реєстрах RPKI. З цими реєстраційними записами асоціюється цифровий сертифікат, який засвідчує право на володіння

маршрутом для кожної автономної системи. Коли маршрутизатор отримує маршрут, система перевіряє, чи має цей маршрут сертифікат у базі даних RPKI. Якщо сертифікат недійсний або не збігається з анонсом, маршрут відкидається як потенційно фальсифікований.

IRR (Internet Routing Registry) [7]. IRR - це набір публічних реєстрів, де автономні системи публікують свої маршрути. Інші мережі можуть використовувати ці реєстри для перевірки легітимності оголошених маршрутів. Хоча IRR не має криптографічного захисту, він все ж дає змогу адміністраторам перевіряти коректність маршрутів, що анонсуються.

BGP Monitoring Protocol (BMP) [8]. Цей інструмент має важливе значення з точки зору безпеки мережі, оскільки він дозволяє моніторити та аналізувати обмін інформацією про маршрути в мережах, що використовують BGP, а також отримувати детальну статистику про всі BGP-сесії, включаючи інформацію про кількість оновлень маршрутів, тривалість сесій, стан з'єднань і т. д. Ця інформація є досить корисною для виявлення потенційних атак або неправильних конфігурацій, таких як помилки в анонсуванні маршрутів або неправильне налаштування фільтрів маршрутизації, що може призвести до небезпек для мережевої безпеки.

Наведені захисні механізми для BGP, хоч і підвищують безпеку маршрутизації, проте можуть негативно впливати на глобальну мережу через зростання затримок, збільшення обсягу даних, централізацію управління, ризик помилок у конфігурації та фрагментацію інформації. Ці фактори можуть призводити до збоїв у маршрутизації, уповільнення роботи мережі та складності впровадження нових технологій. Щоб мінімізувати ці ризики, важливо впроваджувати автоматизацію, резервування, проводити аудити та глобальну координацію між усіма учасниками мережі.

Загальні рекомендації щодо мінімізації негативного впливу на глобальну мережу:

- глобальна координація: узгодження стандартів і підходів впровадження між провайдерами та автономними системами для уникнення фрагментації;
- навчання персоналу: проведення тренінгів для мережевих адміністраторів щодо запровадження і використання нових засобів безпеки;
- поетапне впровадження: використання пробних проектів для тестування впливу захисних механізмів перед їх глобальним розгортанням;
- моніторинг і аудит: постійний моніторинг маршрутів і перевірка коректності налаштувань для якнайшвидкого виявлення проблем;
- адаптація до навантаження: інтеграція засобів безпеки із сучасними рішеннями, які можуть обробляти великий обсяг даних без значного збільшення затримок;
- використання RPKI для перевірки автентичності маршрутів перед їх прийняттям у таблицю маршрутизації, що допомагає уникнути зловмисних або некоректних маршрутів;

- забезпечення багаторівневої маршрутизації з резервними маршрутами, щоб уникнути втрати доступності в разі відмови основного маршруту;
- зменшення залежності від централізованих точок обробки даних і покращення стійкості за допомогою децентралізованих механізмів маршрутизації;
- впровадження спеціалізованих рішень для фільтрації трафіку, спрямованого на перевантаження таблиць маршрутизації;
- оптимізація розподілу трафіку для зменшення впливу на продуктивність маршрутизаторів при великих обсягах даних;
- впровадження вдосконалених розширень, таких як BGPsec, для підвищення безпеки протоколу на глобальному рівні.

Висновки. Протокол BGP є серцем глобальної маршрутизації, однак він має численні вразливості, особливо до атак типу hijacking та route leak. Існуючі механізми захисту, такі як BGPSEC, RPKI та IRR, дозволяють підвищити рівень безпеки маршрутизації, але вони не гарантують повної безпеки та їх використання ще не є універсальним.

З огляду на швидкий розвиток інтернету і зростаючі вимоги до безпеки, протокол BGP та інфраструктура маршрутизації в цілому, повинні адаптуватися до нових викликів, таких як високі обсяги даних, складні атаки та ін. Вдосконалення безпеки і надійності BGP є критично важливим для забезпечення стабільності інтернету в майбутньому.

ПЕРЕЛІК ПОСИЛАНЬ

1. A Border Gateway Protocol 4 (BGP-4): https://datatracker.ietf.org/doc/html/rfc4271
2. A Brief History of the Internet's Biggest BGP Incidents: https://nanog.org/stories/ articles/a-brief-history-of-the-internets-biggest-bgp-incidents/
3. A Brief History of the Internet's Biggest BGP Incidents: https://blog.lacnic.net/en/a-brief-history-of-the-internets-biggest-bgp-incidents/
4. BGP Flaw Can Be Exploited for Prolonged Internet Outages: https://www.security week.com/bgp-flaw-can-be-exploited-for-prolonged-internet-outages/
5. BGPsec Protocol Specification: https://datatracker.ietf.org/doc/html/rfc8205
6. An Infrastructure to Support Secure Internet Routing: https://datatracker.ietf.org/ doc/html/rfc6480
7. Internal Rate of Return (IRR): Formula and Examples: https://www.investopedia.com/ terms/i/irr.asp
8. BGP Monitoring Protocol (BMP): https://datatracker.ietf.org/doc/html/rfc7854

УДК 004.056.4

Д.М. Шкуренко[1], О.В. Кручінін[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ОСОБЛИВОСТІ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ДЛЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ МАЛОГО БІЗНЕСУ

Анотація. У цій статті розглянуто специфіку проведення тестування на проникнення для інформаційно-комунікаційних систем (ІКС) малих підприємств. Враховуючи обмежені фінансові та технічні ресурси малих підприємств, розглянуто основні підходи та інструменти, які можуть бути ефективно використані для підвищення рівня безпеки ІКС. Основна увага приділяється важливості тестування на проникнення для захисту критично важливої інфраструктури малих підприємств, особливо в умовах зростаючої кількості кіберзагроз. Також обговорюються методи оптимізації витрат на тестування безпеки, використання інструментів з відкритим кодом та автоматизація процесів, що дозволяє досягти високих результатів навіть за умов обмеженого бюджету.

Ключові слова: *тестування на проникнення, інформаційно-комунікаційні системи, кібербезпека, малі підприємства, вразливості, автоматизовані засоби, соціальна інженерія, інструменти з відкритим кодом.*

Вступ. У сучасних умовах зростання кіберзагроз, спрямованих на інформаційно-комунікаційні системи (ІКС) малих підприємств, стало серйозною проблемою. Оскільки ці системи є критично важливими для безперервної роботи бізнесу, їх захист від потенційних загроз стає пріоритетним завданням. На відміну від великих компаній, малі підприємства мають обмежені можливості для впровадження комплексних заходів кібербезпеки, що підвищує їх вразливість до атак.[1,5] Вразливості в системах можуть використовуватися зловмисниками для отримання доступу до конфіденційної інформації, порушення роботи або нанесення фінансових збитків. Тестування на проникнення є важливим інструментом виявлення цих слабких місць та підвищення рівня безпеки ІКС.[2] Воно дозволяє своєчасно ідентифікувати та усунути вразливості, забезпечуючи стабільність та надійність функціонування систем. Крім того, тестування на проникнення допомагає підприємствам відповідати нормативним вимогам та стандартам у сфері кібербезпеки.[3]

Постановка задачі. Метою цього дослідження є аналіз особливостей проведення тестування на проникнення для ІКС малих підприємств. Для досягнення цієї мети були поставлені такі завдання:

1. Визначити основні типи вразливостей, характерні для ІКС малих підприємств.
2. Розглянути підходи до проведення тестування на проникнення в умовах обмежених ресурсів.

3. Дослідити ефективні інструменти для проведення тестування на проникнення.

4. Запропонувати рекомендації для підвищення рівня безпеки ІКС малих підприємств.

5. Проаналізувати можливості впровадження автоматизованих засобів тестування для оптимізації процесу виявлення вразливостей.

6. Розробити методики навчання персоналу для підвищення рівня обізнаності щодо кібербезпеки.

7. Розробити підходи до моніторингу та підтримки безпеки ІКС після проведення тестування на проникнення.

Основний зміст роботи. Тестування на проникнення, є процесом моделювання потенційних атак для виявлення слабких місць у системі. Вразливості можуть мати технічний або організаційний характер і бути використані зловмисниками для здійснення несанкціонованих дій [4,5]. Для малих підприємств особливо важливо знайти оптимальний баланс між рівнем безпеки та витратами, оскільки їхні ресурси є обмеженими. Основним завданням у цьому контексті є досягнення максимально можливого рівня захисту за наявних ресурсів. Тестування на проникнення дозволяє виявити вразливі місця та впровадити заходи для їх нейтралізації, забезпечуючи захист від потенційних загроз [2].

Ефективне проведення тестування на проникнення потребує ретельного планування, визначення цілей та вибору відповідних методологій і інструментів. Для малих підприємств важливо забезпечити економічну ефективність тестування, оскільки бюджети на кібербезпеку зазвичай є обмеженими. Вибір інструментів має бути обґрунтованим та враховувати конкретні потреби та можливості підприємства [3].

У дослідженні використовувалися такі методи та інструменти:

- Методологія OWASP. Методологія OWASP (Open Web Application Security Project) дозволяє ідентифікувати основні вектори атак та вразливості, що можуть бути використані зловмисниками для компрометації системи [1]. Структурований підхід OWASP забезпечує чіткі рекомендації для зменшення ризиків і є особливо корисним для підприємств із обмеженими ресурсами. Крім того, OWASP надає стандартизовані тести, які можна адаптувати до потреб конкретного підприємства.
- Автоматизовані інструменти тестування. Використання таких інструментів, як Nmap для сканування мережі [3], Metasploit для моделювання атак [2] та Nikto для аналізу конфігурації веб-серверів,[4] дозволяє швидко ідентифікувати вразливості в ІКС. Nmap забезпечує детальний аналіз мережевої інфраструктури, дозволяючи виявити потенційні точки входу для атак. Metasploit є ефективним інструментом для імітації реальних атак, що допомагає оцінити вразливість системи. Nikto дозволяє виявити відомі конфігураційні помилки та вразливості веб-серверів, що можуть бути використані для компрометації.

- Соціальна інженерія. Людський фактор часто є слабкою ланкою в системах безпеки. Тому навіть найкращі технічні заходи можуть виявитися недостатніми, якщо працівники не підготовлені до протидії атакам із застосуванням методів соціальної інженерії. Проведення навчальних тренінгів та регулярне інформування персоналу про актуальні загрози значно знижує ризик успішних атак із використанням соціальної інженерії [2].

Обмежені фінансові ресурси малих підприємств роблять доцільним використання інструментів з відкритим кодом для забезпечення кібербезпеки. Інструменти, такі як Wireshark, OpenVAS та Burp Suite Community Edition, дозволяють виконувати аналіз безпеки без значних витрат [3]. Wireshark забезпечує моніторинг та аналіз мережевого трафіку, допомагаючи виявляти аномальні дії. OpenVAS є потужним інструментом для пошуку вразливостей у мережевих компонентах і створення детальних звітів для подальшого аналізу. Burp Suite Community Edition дає можливість тестувати безпеку веб-додатків та виявляти такі вразливості, як SQL-ін'єкції та міжсайтовий скриптинг (XSS) [1].

Крім того, впровадження автоматизованих засобів тестування дозволяє значно скоротити час, необхідний для аналізу, та підвищити його ефективність [3]. Автоматизація допомагає швидко виявляти слабкі місця системи та зменшити ризик помилок, які можуть виникати під час ручного тестування. Це особливо важливо для малих підприємств, які не мають достатніх ресурсів для проведення тривалого аналізу безпеки.

Для зниження ризиків кібератак важливо забезпечити регулярне оновлення програмного забезпечення, впровадження багатофакторної автентифікації та навчання персоналу основам кібербезпеки [4]. Регулярне оновлення програмного забезпечення допомагає запобігти використанню відомих вразливостей, а багатофакторна автентифікація забезпечує додатковий рівень захисту від несанкціонованого доступу. Навчання працівників підвищує їхню обізнаність та здатність реагувати на кіберінциденти, що зменшує ймовірність успішних атак [2]. Крім того, слід впровадити політику резервного копіювання та відновлення даних для забезпечення швидкого відновлення роботи системи після інцидентів. Використання засобів моніторингу, таких як Splunk або ELK Stack, дозволяє постійно контролювати стан ІКС та виявляти потенційні загрози на ранніх етапах, що значно підвищує рівень кібербезпеки [3].

Висновки. Тестування на проникнення є важливим елементом забезпечення кібербезпеки ІКС малих підприємств. У рамках дослідження було визначено основні вразливості таких систем та проаналізовано інструменти, що можуть бути ефективно використані для їх виявлення. Запропоновані рекомендації спрямовані на підвищення рівня безпеки ІКС малих підприємств з урахуванням обмежених ресурсів. Використання інструментів з відкритим кодом та автоматизованих засобів тестування допомагає оптимізувати витрати та підвищити ефективність заходів кібербезпеки. Регулярне оновлення програмного забезпечення, багатофакторна автентифікація та навчання персоналу є основними елементами для забезпечення ефективного захисту ІКС

малих підприємств. Подальші дослідження можуть бути спрямовані на розробку конкретних методик автоматизації тестування на проникнення, що зробить цей процес більш доступним для підприємств із обмеженими ресурсами. Крім того, впровадження комплексних заходів моніторингу безпеки та навчання персоналу є важливими складовими для забезпечення постійної підтримки високого рівня кіберзахисту.

ПЕРЕЛІК ПОСИЛАНЬ

1. OWASP Foundation. "OWASP Testing Guide". OWASP, 2021.
2. David Kennedy, Jim O'Gorman, Devon Kearns, and Mati Aharoni. "Metasploit: The Penetration Tester's Guide". No Starch Press, 2011.
3. William Shotts. "The Linux Command Line, 2nd Edition: A Complete Introduction". No Starch Press, 2019.
4. Nikto Developer Team. "Nikto Web Scanner - Documentation". CIRT.net, 2020
5. NIST. "Cybersecurity Framework". National Institute of Standards and Technology, 2020.

УДК 004.5

М. О. Бойчук[1]

[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ОСОБЛИВОСТІ ЗАБЕЗПЕЧЕННЯ ЗАХИЩЕНОСТІ КВАЛІФІКОВАНИХ ЕЛЕКТРОННИХ ПІДПИСІВ У СУДОВИХ УСТАНОВАХ УКРАЇНИ

Анотація. З огляду на важливість електронного документообігу у правосудді, впровадження зазначених рекомендацій сприятиме підвищенню стійкості інформаційних судових систем до кіберзагроз, збереженню довіри громадян та зміцненню законності.

Ключові слова: *інформаційні судові системи, електронний документообіг.*

Постановка проблеми. Кваліфікований електронний підпис (КЕП) є одним із ключових інструментів у забезпеченні тріади інформаційної безпеки (конфіденційності, цілісності (достовірності) та доступності) діловодства загальних та спеціалізованих судів. У ракурсі забезпечення безперебійності робочого процесу судів України, використання КЕП є важливим через високу юридичну значущість відомостей, що циркулюють в інформаційній системі судових установ, та необхідність підтримки оперативного доступу співробітників суду до документації судових справ. Завдання забезпечення захищеності КЕП в установах судової влади є актуальним в умовах зростання кількості і складності ворожих кібератак і ризиків компрометації інформації, що обробляється в єдиній інформаційній системі судів України.

Особливостями використання КЕП у судових установах є недопустимість помилок в алгоритмах шифрування ключів та у користувацьких паролях

співробітників, адже помилка у використанні або компрометація КЕП може призвести до недійсності документів. Файли особистих КЕП формуються та передаються співробітникам судів на захищені файлові носії (флеш-накопичувачі) уповноваженими територіальними органами Державної податкової служби України.

Рішення судів, клопотання та інші документи судових справ найчастіше містять персональні дані сторін судових процесів та іншу чутливу інформацію, що робить їх привабливими цілями для зловмисників, які мають намір отримати несанкціонований доступ до персональних даних.

Кваліфікований електронний підпис гарантує високий рівень довіри до обміну документацією між судовими установами, слідчими, прокурорами, адвокатами та іншими учасниками судових процесів.

Виклад матеріалів дослідження. Для вдосконалення інформаційної безпеки кваліфікованих електронних підписів необхідно виділити основні загрози для цих атрибутів доступу. Такими загрозами є:

- компрометація закритого ключа - якщо закритий ключ КЕП потрапляє до зловмисників, вони можуть підробляти електронні документи;
- фішинг-атаки - користувачі можуть бути спрямовані на підроблені сайти шляхом введення в оману, де їх персональний закритий ключ може бути викрадено;
- недостатній рівень захисту програмного та апаратного забезпечення - вразливості в програмному забезпеченні або обладнанні інформаційних систем судових установ можуть стати каналами для кібератак.

Можна виділити такі основні заходи для забезпечення захищеності КЕП:

- резервне копіювання - регулярне створення резервних копій даних та ключів, що дозволить відновити доступ до файлу КЕП у разі інциденту інформаційної безпеки;
- впровадження сучасних криптографічних алгоритмів - використання стійких до зламу алгоритмів, таких як RSA з довжиною ключа не менше 2048 біт або алгоритмів на основі еліптичних кривих;
- захист файлових носіїв КЕП - використання апаратних токенів або програмних засобів ієрархічного керування файловими носіями HSM (Hierarchical Storage Management) для зберігання закритих ключів, а також багаторівнева автентифікація за потреби їх застосування;
- навчання персоналу - регулярне проведення тренінгів з інформаційної безпеки та гігієни, правильного використання КЕП для співробітників судів;
- моніторинг та аудит безпеки - впровадження систем запобігання та виявлення вторгнень IPS/IDS (Intrusion Prevention System / Intrusion Detection System) та проведення регулярних перевірок на відповідність службової КСЗІ міжнародним стандартам ISO у галузі інформаційної безпеки.

Висновки. Ефективне забезпечення захищеності кваліфікованих електронних підписів у судах України потребує комплексного підходу, що включає технічні, організаційні та правові заходи. З огляду на важливість електронного документообігу у правосудді, впровадження вищезазначених рекомендацій сприятиме підвищенню стійкості інформаційних судових систем до кіберзагроз, збереженню довіри громадян та зміцненню законності у цифрову епоху.

ПЕРЕЛІК ПОСИЛАНЬ

1. ISO/IEC 27001:2022. "Information technology — Security techniques — Information security management systems — Requirements".
2. [Гусєв В.І., Підопригора О.М. "Кібербезпека в органах державної влади: аналіз сучасних загроз та способи захисту". Науковий журнал "Інформаційна безпека", 2023, №2, с. 17-25.]
3. [Ковальчук А.В. "Використання електронного цифрового підпису в судовій практиці: переваги та ризики". Юридичний вісник, 2022, №4, с. 45-52.]
4. Міністерство цифрової трансформації України. "Методичні рекомендації щодо забезпечення захисту інформації при використанні ЕЦП".

УДК 004.056.55

В.Г. Бородкін[1]

[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ПРОГРАМНЕ РІШЕННЯ ДЛЯ ЗАХИСТУ ВЕБРЕСУРСІВ ВІД SQL-ЗАГРОЗ ЗА ДОПОМОГОЮ PYTHON

Анотація. У статті розглядається проблема SQL-ін'єкцій як одного з найнебезпечніших видів кіберзагроз для вебресурсів. Досліджено програмне рішення, розроблене на Python, що дозволяє ефективно запобігати цим атакам завдяки застосуванню підготовлених запитів, автоматизованих методів перевірки введення даних і сучасних бібліотек.

Ключові слова: *SQL-ін'єкції, кібербезпека, Python, захист баз даних, економічна доцільність, OWASP ZAP.*

Вступ. У сучасному світі вебресурси стали важливою складовою різних бізнес-процесів, наукових досліджень і соціальної взаємодії. Разом із цим зростають і ризики їх уразливості до різних видів атак, серед яких SQL-ін'єкції є одними з найпоширеніших і найнебезпечніших. Такі атаки дозволяють зловмисникам отримати доступ до конфіденційної інформації, маніпулювати базами даних або навіть повністю знищувати їх.

Особливо важливою є проблема SQL-ін'єкцій для підприємств, що оперують великими обсягами даних. Недостатній рівень захисту вебдодатків може призвести до серйозних фінансових збитків, втрати репутації та

юридичних проблем. Саме тому розробка та впровадження ефективних методів захисту вебресурсів є актуальним завданням.

Постановка задачі. Метою роботи є дослідження програмного рішення для захисту вебресурсів від SQL-ін'єкцій на основі сучасних методів кібербезпеки. Для досягнення цієї мети було поставлено наступні завдання:

- Провести аналіз сучасних методів захисту вебресурсів від SQL-ін'єкцій.
- Дослідити переваги та недоліки мов програмування та фреймворків для реалізації захищеного додатку.
- Дослідити архітектуру програмного забезпечення на основі Python із використанням бібліотек SQLAlchemy і Flask.
- Реалізувати тестування рішення з використанням інструментів для оцінки кібербезпеки.
- Оцінити економічну доцільність розробленого рішення.

Основний зміст роботи. SQL-ін'єкції є одними з найсерйозніших загроз для баз даних, оскільки вони дозволяють зловмисникам вводити шкідливі команди SQL, які виконуються сервером бази даних. Згідно з дослідженнями OWASP, до 60% всіх атак на вебдодатки включають SQL-ін'єкції. Основними причинами успіху таких атак є відсутність перевірки введених даних, використання динамічних SQL-запитів без захисту та недостатня сегментація прав доступу до бази даних. Для боротьби з цими проблемами запропоновано застосування таких методів, як використання підготовлених запитів, регулярна перевірка систем на вразливості та шифрування даних. Python обрано як основну мову розробки завдяки її простоті, широкій підтримці та наявності потужних бібліотек для роботи з базами даних. Основні використані технології включають SQLAlchemy, яка є ORM-бібліотекою для роботи з базами даних і дозволяє створювати підготовлені запити та захищати дані, Flask — легкий вебфреймворк для розробки серверної частини додатка, та OWASP ZAP, який використовується для тестування безпеки вебдодатків.

Розроблене програмне забезпечення базується на архітектурному патерні MVC (Model-View-Controller). Model відповідає за роботу з базою даних через ORM-бібліотеку SQLAlchemy. View надає інтерфейс для взаємодії з користувачем, побудований на Flask. Controller обробляє запити користувачів та координує взаємодію між моделями і представленнями. Система забезпечує фільтрацію введених даних, автоматичне блокування запитів, що містять шкідливі SQL-команди, та логування підозрілої активності для подальшого аналізу. Проведено тестування системи на прикладі навчального вебдодатку, що імітує базову функціональність реального ресурсу. За допомогою OWASP ZAP було виявлено потенційні вразливості, які були усунені шляхом додаткового шифрування даних та впровадження системи багатофакторної автентифікації. Результати тестування показали, що система ефективно запобігає спробам

впровадження шкідливих SQL-команд. Час обробки запитів залишився в межах прийнятного діапазону, що підтверджує високу продуктивність рішення.

Впровадження розробленого програмного забезпечення дозволяє зменшити ймовірність фінансових втрат через несанкціонований доступ до баз даних. Було проведено розрахунки економічної ефективності, які включають вартість розробки та впровадження системи, оцінку збитків, яких вдалося уникнути завдяки захисту від атак, та аналіз потенційного збільшення доходів за рахунок підвищення довіри користувачів до захищеності вебресурсу. Результати підтвердили, що інтеграція такого рішення окупається протягом першого року використання.

Наукова новизна роботи полягає у розробці програмного рішення для захисту вебресурсів від SQL-ін'єкцій на базі Python з використанням SQLAlchemy і Flask. Інтеграція методів фільтрації даних, підготовлених запитів та автоматичного блокування шкідливої активності забезпечує високий рівень безпеки. Використання OWASP ZAP для тестування підтвердило ефективність запропонованих рішень.

Додатково розглянуто економічну доцільність впровадження, що включає зменшення фінансових ризиків та підвищення довіри користувачів, що підкреслює практичну цінність роботи.

Висновки. Досліджено рішення на основі Python та сучасних бібліотек, котрі забезпечують надійний захист вебресурсів від SQL-ін'єкцій. Проведене тестування підтвердило його ефективність та продуктивність. Інтеграція програмного забезпечення в існуючі вебсайти дозволяє не тільки знизити ризики кіберзагроз, але й підвищити загальний рівень довіри користувачів до ресурсу.

У перспективі планується розширення функціональності системи шляхом впровадження додаткових методів виявлення аномалій та інтеграції з іншими інструментами безпеки

ПЕРЕЛІК ПОСИЛАНЬ

1. Даніель, Шац; Джулі, Стіна. «До більш репрезентативного визначення кібербезпеки». Журнал цифрової криміналістики, безпеки та права.
2. Роуз, Маргарет. «Визначення соціальної інженерії». Технічна ціль.
3. Шац, Даніель; Башруш, Рабіх; Уолл, Джулі. «До більш репрезентативного визначення кібербезпеки». Журнал цифрової криміналістики, безпеки та права.
4. Стівенс, Тім. «Глобальна кібербезпека: нові напрями в теорії та методах». Політика та управління.
5. Millman, Renee. «Нове поліморфне шкідливе програмне забезпечення обходить три чверті AV-сканерів». SC Magazine Великобританія.
6. Сканнелл, Кара. «Шахрайство з електронною поштою генерального директора коштує компаніям 2 мільярди доларів». Financial Times.
7. Марсель, Себастьян; Ніксон, Марк; Лі, Стен, ред. Посібник із біометричної боротьби зі спуфінгом: надійна біометрія під спуфінгом Напади(PDF). Лондон: Springer.

УДК 004.056.55

Д.В. Кравченко[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ДЕЦЕНТРАЛІЗОВАНІ ІДЕНТИФІКАТОРИ (DID)

Анотація. Досліджуються теоретичні принципи та практичне застосування децентралізованих ідентифікаторів (DID), акцентуючи на їх актуальності в сучасному управлінні ідентифікацією, орієнтованому на конфіденційність. Описано практичну реалізацію, яка демонструє процес реєстрації та автентифікації користувача на основі DID на змодельованому прикладі.

Ключові слова: *децентралізовані ідентифікатори (DID), керування ідентифікацією, децентралізована автентифікація, конфіденційність, верифіковані дані, централізоване керування.*

Вступ. У сучасній цифровій екосистемі управління та захист онлайн-ідентичностей стає дедалі складнішим. Традиційні системи управління ідентифікацією, такі як автентифікація за допомогою імені користувача і пароля та централізовані бази даних, стикаються зі зростаючими проблемами, пов'язаними з безпекою, конфіденційністю та контролем. Ці системи часто покладаються на центральні органи влади для перевірки та зберігання інформації про користувачів, що робить їх вразливими до витоку даних, несанкціонованого доступу та зловживань конфіденційною інформацією.

Децентралізовані ідентифікатори (DID) – це новий тип ідентифікаторів, який забезпечує децентралізовану цифрову ідентичність, яку можна перевірити. DID відноситься до будь-якого об'єкта (наприклад, особи, організації, речі, моделі даних, абстрактної сутності тощо), визначеного контролером DID. На відміну від типових федеративних ідентифікаторів, DID були розроблені таким чином, щоб їх можна було відокремити від централізованих реєстрів, постачальників ідентифікаційних даних та центрів сертифікації. Зокрема, хоча інші сторони можуть бути використані для того, щоб допомогти виявити інформацію, пов'язану з DID, дизайн дозволяє контролеру довести контроль над ним без необхідності отримання дозволу від будь-якої іншої сторони. DID - це URI, які пов'язують суб'єкт DID з документом, що дозволяє здійснювати довірчі взаємодії, пов'язані з цим суб'єктом.[1]

Постановка задачі. Були сформовані та вирішені такі завдання:

- Проаналізувати актуальність децентралізованих ідентифікаторів та їх роль у інформаційній безпеці
- Розробити практичну реалізацію, яка показує правдивість твердження

Основний зміст роботи. Децентралізовані ідентифікатори (DID) – це спосіб ідентифікувати себе в Інтернеті без використання центрального органу, наприклад уряду чи компанії. Децентралізовані ідентифікатори (DID) представляють зміну в тому, як ми керуємо цифровими ідентифікаторами.

Вирішуючи основні питання безпеки, конфіденційності та контролю, DID забезпечують безпечний, приватний та ефективний спосіб обробки особистої інформації. Оскільки кіберзагрози продовжують зростати, як і очікування користувачів щодо конфіденційності, впровадження DID є важливим кроком.[2]

Більше того – і ця причина має бути найпереконливішою – надаючи людям засоби для встановлення та керування власними цифровими ідентифікаторами, DID можуть значно підвищити цифрове залучення. Це особливо важливо в цифровій економіці, де доступ до онлайн-сервісів, фінансових систем і навіть ринків праці все більше залежить від наявності цифрової ідентичності, яку можна перевірити.

Чому нам сьогодні потрібні DID: Децентралізовані цифрові ідентифікатори перетворилися з приємних інструментів на необхідні інструменти. Причини цього вагомі:

1. Боротьба з крадіжкою особистих даних: Традиційні системи ідентифікаційної інформації дуже вразливі. Лише у 2023 році кількість кібератак, націлених на системи ідентифікації, зросла на 48%. Понад 80% випадків витоку даних пов’язані зі зламаними обліковими даними. DID зменшує цей ризик, усуваючи необхідність централізованого зберігання персональних даних.

2. Ризики централізації: централізовані системи є єдиними точками збою. Злом SolarWinds 2023 року, який торкнувся тисячі організацій, підкреслив небезпеку централізованого керування ідентифікацією. Оскільки DID функціонують у блокчейні, вони децентралізовані, а тому не схильні до ризиків централізації.

3. Захист конфіденційності користувачів: зростає недовіра до того, як компанії обробляють особисті дані. За даними Identity Defined Security Alliance, 66% споживачів турбуються про конфіденційність своїх даних. Gartner повідомляє, що DID значно підвищують конфіденційність користувачів, дозволяючи окремим особам ділитися лише необхідною інформацією, тим самим зменшуючи доступ до конфіденційних даних.

4. Відповідність нормативним вимогам: завдяки законам про захист даних, таким як GDPR, підприємствам потрібні надійні рішення ідентифікації, які можуть надати DID.

5. Економічні переваги: згідно з дослідженням McKinsey, компанії можуть заощадити до 90% на витратах на перевірку особи за допомогою рішень цифрової ідентифікації.

Під час надання верифікованих облікових даних (VC) або верифікованої презентації (VP) суб’єкт надає лише DID емітенту або верифікатору. Ключ DID слід розглядати як псевдонім для ідентифікації; його можна відстежувати та корелювати. DID є частиною кожного VC та VP. Кінцеві користувачі повинні використовувати кілька DID, щоб зменшити відстеження та кореляцію, але це також вимагає, щоб кожен DID з’являвся лише в одному VC. Наявність кількох DID з однаковим вмістом VC дозволяє корелювати та ідентифікувати через вміст VC. «Умови використання» VP і VC можуть бути порушені Верифікатором або

Емітентом, діючи як один без повідомлення про дію. Увесь вміст, який надається третім сторонам, піддається загрозам конфіденційності; тому кінцевий користувач повинен ретельно оцінити, що і кому надсилати. Кінцевий користувач має повний контроль над усіма явними даними, і рішення про згоду на розголошення є кінцевим користувачем. Рекомендується видавати лише один особистий VC для даного DID, щоб мінімізувати відстеження користувача. Проте той самий VC може бути виданий декільком DID, керованим контролером DID.[3]

Таблиця 1 наглядно показує відмінності та переваги децентралізованого керування ідентифікацією над централізованим.

Таблиця 1

Децентралізоване проти централізованого керування ідентифікацією

Децентралізоване керування ідентифікацією	**Централізоване управління ідентифікацією**
Користувачі самостійно контролюють свої особисті дані	Користувачі повинні довіряти сторонньому постачальнику послуг для забезпечення безпеки своїх даних
Користувачі завжди контролюють, хто має доступ до їхніх даних	Постачальник послуг контролює, хто має доступ до даних користувача
Це більш безпечно, оскільки доступ обмежено власниками гаманців, а дані децентралізовані	Дані зберігаються в централізованій системі, що полегшує їх злам
Обмін даними має бути дозволений власниками даних	Даними можна ділитися без відома власників даних

У багатьох частинах світу мільйони людей не мають доступу до офіційної ідентифікації, що обмежує їхню можливість доступу до основних послуг, таких як банківська справа, охорона здоров’я та освіта. За даними Світового банку, понад 1 мільярд людей у всьому світі не мають жодної форми юридично визнаної ідентифікації.[4]

DID пропонують багатообіцяюче рішення цієї проблеми, надаючи безпечну цифрову ідентифікацію, яку можна перевірити, якою люди можуть володіти та контролювати. DID можуть відігравати вирішальну роль у гуманітарних зусиллях, особливо в таборах для біженців, де традиційні

ідентифікаційні документи часто губляться або знищуються. Такі організації, як Організація Об'єднаних Націй, досліджують рішення ідентифікації на основі блокчейну, щоб надати біженцям безпечні цифрові ідентифікаційні дані, які можуть допомогти їм отримати допомогу, медичне обслуговування та інші необхідні послуги.

Альянс ID2020 – це глобальне партнерство, яке прагне покращити життя за допомогою цифрової ідентичності. У них є пілотні проекти, у яких технологія блокчейн і DID використовуються для надання цифрових ідентифікаторів біженцям, гарантуючи безпеку їхніх ідентифікацій і розпізнавання за кордоном.[5]

Було розроблено практичну реалізацію, щоб впевнитись в безпечному та зручному використанню DID, навідміну від звичайних, централізованих ідентифікаторів. Мета полягала в тому, щоб продемонструвати, як DID можна використовувати для реєстрації користувачів і їх автентифікації, не покладаючись на централізованих постачальників ідентифікаційних даних. Це узгоджується з сучасними потребами щодо безпечного, орієнтованого на користувача керування ідентифікацією у все більш цифровому та децентралізованому світі. Демонстрація складається з бекенда (сервер Node.js) і інтерфейсу (обслуговується за допомогою Vite), які взаємодіють, щоб полегшити реєстрацію та автентифікацію користувачів за допомогою DID.

Процес:

1. Реєстрація користувача

Користувач вводить унікальний ідентифікатор (наприклад, своє ім'я користувача або ідентифікатор користувача) у форму реєстрації (рис. 1).

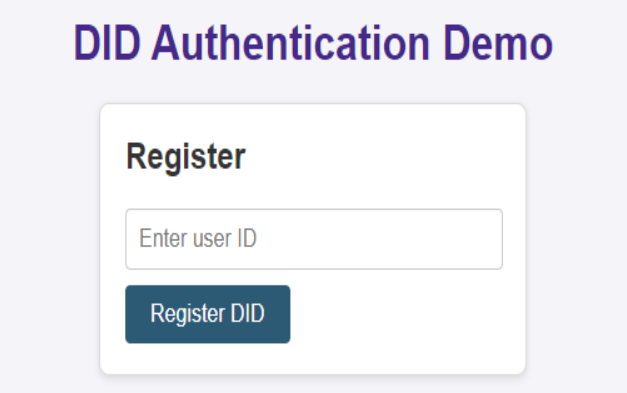


Рис. 1. Форма реєстрації

Інтерфейс надсилає цю інформацію на сервер через виклик API (/register). Новий DID створюється для користувача за допомогою постачальника криптографії Ed25519. DID і відповідне початкове значення закритого ключа надійно зберігаються на сервері (у пам'яті для цієї демонстрації). Сервер відповідає повідомленням із підтвердженням реєстрації та наданням відповідно DID (рис. 2).

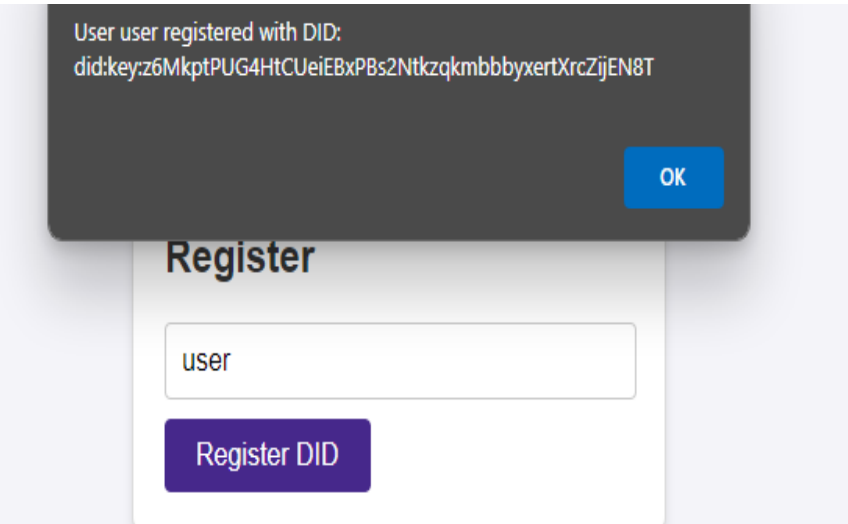


Рис. 2. Підтвердження реєстрації

2. Створення виклику

Користувач надає свій ідентифікатор користувача для отримання виклику автентифікації через API (/challenge). Сервер генерує унікальний запит із міткою часу, пов'язаний з ідентифікатором користувача. Виклик надсилається користувачеві для підписання. Виклики гарантують, що тільки користувач з відповідним DID може автентифікуватися, що унеможливлює повторні атаки або видавання себе за іншу особу.

3. Підписання виклику

Користувач підписує виклик за допомогою свого приватного ключа DID, гарантуючи, що підпис криптографічно прив'язаний до його особи. Процес підписання підтверджує, що користувач володіє закритим ключем, пов'язаним із DID, що забезпечує безпечне підтвердження права власності.

Процес підписання відбувається автоматично, без потреби користувача робити це самому. Для даної практичної реалізації цей процес був представлений за допомогою додаткових кнопок на отримання та підписання виклику (рис. 3).

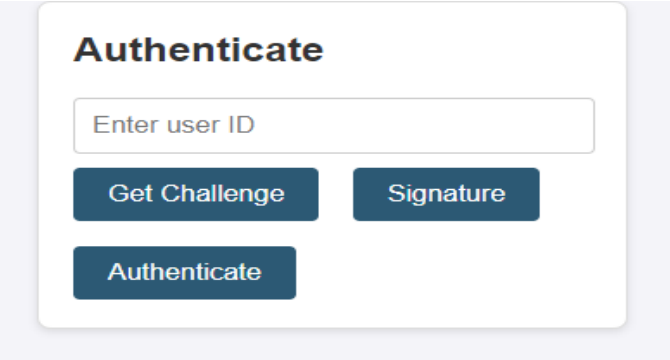


Рис. 3. Форма автентифікації

4. Автентифікація

Підписаний виклик надсилається назад до серверної частини для перевірки через API (/authenticate). Сервер звіряє підпис із DID користувача. Якщо підпис дійсний, автентифікація користувача пройшла успішно (рис. 4).

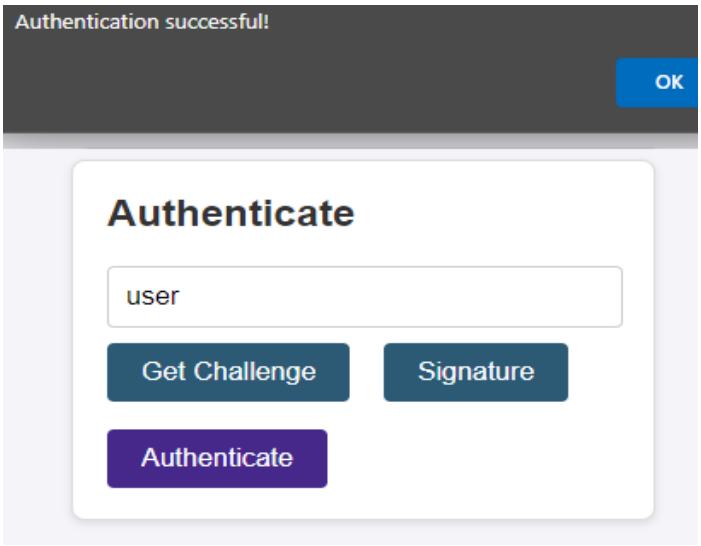


Рис. 4. Успішна автентифікація

Цей крок завершує процес автентифікації, забезпечуючи безпечний доступ без необхідності введення паролів.

Висновки. Децентралізовані ідентифікатори (DID) представляють зміну в тому, як ми керуємо цифровими ідентифікаторами. Вирішуючи основні питання безпеки, конфіденційності та контролю, DID забезпечують безпечний, приватний та ефективний спосіб обробки особистої інформації. Оскільки кіберзагрози продовжують зростати, як і очікування користувачів щодо конфіденційності, впровадження DID має бути важливим кроком стосовно інформаційної безпеки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Decentralized Identifiers (DIDs) v1.0. Core architecture, data model, and representations. W3C Recommendation. URL: https://www.w3.org/TR/did-core/#:~:text=Decentralized%20identifiers%20(DIDs)%20are%20a,the%20controller%20of%20the%20DID.
2. Guide to Decentralized Identifiers (DIDs). URL: https://www.dock.io/post/decentralized-identifiers
3. DID Method for Natural Persons. URL: https://hub.ebsi.eu/vc-framework/did/natural-person
4. World Bank – 1.1 Billion ‘Invisible’ People without ID. URL: https://www.worldbank.org/en/news/press-release/2017/10/12/11-billion-invisible-people-without-id-are-priority-for-new-high-level-advisory-council-on-identification-for-development#:~:text=WASHINGTON%2C%20October%2012%2C%202017%E2%80%94,are%20children%20who%20are%20unregistered.
5. Альянс ID2020. URL: https://www.id2020.org/

УДК 004.056.55

В.М. Кононець[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

МЕТОДИ ТЕОРІЇ ІГОР У КІБЕРБЕЗПЕЦІ: ВІД ПРОАКТИВНИХ СТРАТЕГІЙ ДО ЗНИЖЕННЯ ЗБИТКІВ

Анотація. Дана стаття розглядає методи теорії ігор, що застосовуються в галузі кібербезпеки, зокрема, для аналізу взаємодії між атакуючими й захисниками інформаційних систем. Представлено практичний приклад використання теорії ігор для моделювання реальних сценаріїв, таких як DDoS-атаки та управління ризиками, з акцентом на баланс між витратами на захист і потенційними збитками.

Ключові слова: *теорія ігор, байесова гра, реактивна стратегія, мінімакс, модель матричної гри*

Вступ. Кібербезпека сьогодні є однією з найбільш динамічних галузей науки та техніки, яка вимагає використання новітніх підходів для боротьби з еволюціонуючими загрозами. Одним із перспективних напрямів є застосування теорії ігор, яка дозволяє моделювати взаємодію між атакуючими та захисниками у кіберпросторі.

Сучасні підходи до кіберзахисту базуються на аналізі історичних даних про атаки, однак вони не враховують стратегічний характер поведінки атакуючих. Теорія ігор пропонує математичний апарат для аналізу цих взаємодій, прогнозування атак і вибору оптимальних стратегій захисту.

Постановка проблеми. Сучасні системи кібербезпеки стикаються зі зростаючою складністю та частотою атак, які постійно змінюються та адаптуються. Реактивні підходи, які базуються на аналізі історичних даних про атаки, втрачають ефективність через неможливість врахувати стратегічний характер поведінки атакуючих. У такій ситуації постає необхідність розробки підходів, які дозволяють моделювати взаємодію між атакуючими й захисниками, враховуючи невизначеність, асиметрію інформації та обмежені ресурси захисту. Методи теоретико-ігрового підходу стають у нагоді в такому випадку, бо мають потужний математичний апарат.

Теоретичні основи

Основою дослідження є поняття теорії ігор, а саме:

1) Гра з нульовою сумою, де атакуючий і захисник мають протилежні інтереси, що дозволяє застосувати моделі мінімаксу для аналізу [1].

2) Гра з неповною інформацією, де захисники не завжди знають стратегії та цілі атакуючих, але можуть використовувати ймовірнісні методи для їхнього прогнозування.

3) Динамічні ігри, де відбувається моделювання багатоетапних атак із поступовим коригуванням стратегій[1].

У контексті теорії ігор у кібербезпеці стратегії поділяють на реактивні (захисні дії після атаки) та проактивні (захисні дії до виникнення атаки). Ці підходи мають різний характер і застосовуються залежно від наявних ресурсів, рівня ризиків і стратегій атакуючого. Реактивні стратегії орієнтовані на відповіді після початку атаки [2]. Вони фокусуються на мінімізації шкоди та відновленні роботи систем після порушення. До них можуть належати ігри з нульовою сумою, де атакуючий намагається завдати максимальної шкоди, а захисник – мінімізувати втрати. Проактивні стратегії базуються на передбаченні можливих атак і превентивних діях [2]. Їх мета – знизити ймовірність успішної атаки або її вплив на систему (наприклад, байесові ігри).

Можливості застосування методів теорії ігор у кібербезпеці

Ігрові моделі дозволяють визначити, як розподілити обмежені ресурси (фінанси, обчислювальну потужність) між критично важливими вузлами системи. Наприклад, під час DDoS-атаки важливо швидко визначити найбільш уразливі ділянки мережі та зосередити ресурси на їхньому захисті. Атакуючі завжди змінюють тактики, тому ефективний захист має бути динамічним. Завдяки іграм із неповною інформацією можна розробляти стратегії, що реагують на поведінку противника у реальному часі.

Ігрові підходи добре працюють у багатокомпонентних системах, таких як IoT-мережі, де різні вузли мають різний рівень критичності [2]. Наприклад, розробка стратегій кооперативної гри для виявлення шкідливих вузлів. Якщо застосувати алгоритми машинного навчання разом із теорією ігор, це дозволить імітувати можливі сценарії атак і підготувати превентивні заходи. Теоретико-ігрові моделі визначають також і оптимальний спосіб захисту від перевантаження серверів шляхом аналізу можливих стратегій атакуючого.

Розглянемо приклад, де сервер зазнає DDoS-атаки. Атакуючий може спрямувати атаку на сервер із різною інтенсивністю [3]:

a_1 – слабка атака.

a_2 – середня атака.

a_3 – масована атака.

Захисник в свою чергу ж може вибрати стратегії:

d_1 – розширення пропускної здатності мережі.

d_2 – впровадження фільтрації підозрілих запитів.

d_3 – відключення окремих серверів для мінімізації втрат.

Матриця втрат для захисника

Втрати захисника у цьому випадку будуть залежати від параметрів вказаних в таблиці 1.

Таблиця 1.

Параметр	Опис	Позначка
Інтенсивность атаки	чим ефективність вища, тим більші втрати	L_a
Ефективность обраного захисту	ймовірність успіху захисту d_j проти атаки a_i	E_{ij}
Ціна впровадження захисту	вартість впровадження мір захисту	C_j

Таблиця 1. **Необхідні параметри для моделі витрат**

З даних параметрів наша модель витрат буде мати наступний вигляд (див. формула 1)

$$L_{ij} = \left(1 - E_{ij}\right) * L_a + C_j\,, \tag{1}$$

Втрати від атак: L_a={50 тис. грн, 200 тис. грн., 1 млн. грн.} для атак - a_1, a_2, a_3.

Вартість захисту: C ={10 тис. грн на добу, 20 тис. грн на добу, 5 тис. грн.}.

Також маємо оцінку ефективності в умовах української інфраструктури:

$$\boldsymbol{E} = \begin{bmatrix} 0.8 & 0.5 & 0.2 \\ 0.9 & 0.7 & 0.4 \\ 0.6 & 0.8 & 0.9 \end{bmatrix}$$

Очікувані втрати для кожної комбінації обчислюються за формулою 2

$$L_{ij} = \left(1 - E_{ij}\right) * L_{a_i} + C_j\,, \tag{2}$$

Звідси отримуємо матрицю втрат для захисника

$$\boldsymbol{L} = \begin{bmatrix} 60000 & 105000 & 40000 \\ 80000 & 85000 & 125000 \\ 800000 & 625000 & 105000 \end{bmatrix}$$

Так як захисник (β) намагається мінімізувати максимальні втрати (a_{ij}), то використовуємо критерій мінімаксу (формула 3)

$$\beta = \max_j \beta_j = \min_j \max_i a_{ij}\,, \tag{3}$$

Захисник обирає стратегію за якої збитки будуть мінімальними – d_3 (локалізація трафіку), бо максимальні втрати мінімальні (125000 грн). І в даному випадку немає стратегії краще.

Теоретико-ігровий потенціал на цьому не вичерпаний, адже наприклад, для захисту банківських систем використання байесових ігор [2] для виявлення шахрайства при транзакціях є більш доречним. І це є безумовним плюсом для такого потужного математичного апарату, адже він має певний тип ігор, який може стати у нагоді при різноманітних кібератаках. Кажучи про розподіл ресурсів у хмарних системах [4], навіть у такому випадку на основі аналізу ймовірностей атак на конкретні вузли, можливо розробити стратегії захисту.

Удосконалення ігрових моделей у кібербезпеці можливе також через інтеграцію з алгоритмами машинного навчання, які забезпечують прогнозування атак (Random Forest, SVM, LSTM), адаптацію стратегій (Q-Learning), виявлення вразливостей у реальному часі (K-Means, DBSCAN).

Аналогічні дослідження

У роботі Roy та Ellis (2017) досліджено застосування байесових ігор для виявлення шахрайства у фінансових транзакціях. Це дослідження підкреслило важливість врахування асиметрії інформації, коли атакуючий володіє перевагою в знаннях про вразливості.

В одній із статей була запропонована модель, що інтелектуально обирає найбільш підходящий модуль для виявлення атаки на основі сигнатур, аномалій та honeypot-детекторів [5]. Особлива увага була привернута на рівновагу змішаної стратегії Неша та результати моделювання.

Дослідження Adamson та співавторів (2017) акцентує увагу на управлінні ресурсами у хмарних системах за допомогою теорії ігор. У ньому було запропоновано використання кооперативних ігор для оптимального розподілу обчислювальних потужностей, що мінімізує ризики перевантаження.

Перспективи розвитку: розширення можливостей алгоритмів машинного навчання у поєднанні з теорією ігор для розробки інноваційних рішень у сфері кібербезпеки.

Висновки. Розробка і впровадження нових підходів теорії ігор у кібербезпеці дозволяє суттєво підвищити рівень захисту інформаційних систем. Зокрема, це стосується адаптивного захисту, оптимізації використання ресурсів та прогнозування загроз.

Розгляд такого методу у питаннях кібербезпеки має свої плюси та мінуси, зокрема одним із мінусів може бути необхідність у точних даних про ймовірності подій, витрати, ефективність стратегій для побудови моделей. У багатьох випадках ці дані є невідомими або важкодоступними [6]. Але у міру появи інформації про нові атаки можна уточнювати ймовірності відповідно до попередніх гіпотез, що є певним плюсом.

ПЕРЕЛІК ПОСИЛАНЬ

1. Osborne M. J., Rubinstein A. A. Course in Game Theory. MIT Press, – 1994.

2. Roy S., Ellis C. A. Game Theory for Cyber Security. Springer, – 2017.
3. T. Spyridopoulos, G. Karanikas, T. Tryfonas, G. Oikonomou. A game theoretic defence framework against DoS/DDoS cyber attacks. – 2013.
4. G. Adamson, L. Wang, M. Holm, and P. Moore, “Cloud manufacturing – a critical review of recent development and future trends,” Int. J. Comput. Integr. Manuf., – 2017.
5. Komal Singh, Sharad Saxena, Anju Sharma. GTM-CSec: Game theoretic model for cloud security based on IDS and honeypot,– 2020.
6. О.П. Ігнатенко. Теоретико-ігровий підхід до проблеми безпеки мереж – 2017.

УДК 004.056.55

М.В.Харченко[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ВИКОРИСТАННЯ БЛОКЧЕЙН-ТЕХНОЛОГІЙ ДЛЯ ЗАХИСТУ МЕДИЧНИХ ДАНИХ У ЦИФРОВИХ СИСТЕМАХ ОХОРОНИ ЗДОРОВ’Я

Анотація. Дане дослідження аналізує застосування блокчейн-технологій для захисту медичних даних, їх переваги перед централізованими системами, а також виклики інтеграції у медичну сферу. Розглянуто основні кіберзагрози, проблеми централізованих баз даних і можливості блокчейну для забезпечення конфіденційності, цілісності та прозорості даних

Ключові слова: *блокчейн, захист медичних даних, децентралізовані системи, смарт-контракти, кібербезпека, цифрові системи охорони здоров’я, прозорість даних, цілісність інформації, конфіденційність, інтеграція медичних систем*

Вступ. Сучасні системи охорони здоров’я все більше орієнтуються на цифровізацію, прагнучи підвищити ефективність надання медичних послуг, полегшити доступ до даних пацієнтів та забезпечити якісну взаємодію між медичними установами. Ця трансформація, хоча й створює безпрецедентні можливості, супроводжується новими викликами, зокрема в контексті захисту чутливої медичної інформації. Уразливості традиційних підходів до зберігання та обміну даними роблять такі системи привабливою мішенню для кіберзлочинців [1], [2], [3].

Блокчейн-технології пропонують інноваційні рішення для захисту медичних даних, об’єднуючи принципи децентралізації, прозорості та високого рівня безпеки [4]. Завдяки своїй архітектурі блокчейн забезпечує захист від несанкціонованого доступу, цілісність даних та можливість відстеження будь-яких змін [5]. Ці характеристики стають надзвичайно важливими у сфері охорони здоров’я, де кожна помилка або витік інформації може мати серйозні наслідки для пацієнтів.

На відміну від централізованих баз даних, що мають один або кілька вразливих вузлів, блокчейн функціонує як розподілена система, у якій усі записи є незмінними та надійно захищеними. Ця технологія дозволяє не лише

підвищити безпеку, але й розширює можливості для інтеграції медичних систем, спрощуючи обмін даними між установами та країнами. Наприклад, з використанням смарт-контрактів можна автоматизувати доступ до даних лише для уповноважених осіб, тим самим мінімізуючи ризик витоку конфіденційної інформації [4], [5].

Попри значний потенціал, впровадження блокчейн-технологій у сферу охорони здоров’я залишається складним завданням. Воно вимагає врахування правових, технічних та етичних аспектів, а також подолання перешкод, пов’язаних із масштабованістю та продуктивністю мереж [4], [5]. Зважаючи на актуальність проблеми захисту медичних даних, інтеграція блокчейну у цифрові системи охорони здоров’я є перспективним напрямом, що заслуговує на всебічне вивчення та практичне впровадження [5].

Основний зміст роботи. Кіберзагрози в медичній сфері стали серйозною проблемою, оскільки зростаюча цифровізація привела до підвищення вразливості медичних установ до різноманітних атак [1]. Відзначено значне збільшення кількості кіберзлочинів у цій галузі, оскільки медичні дані мають високу вартість на чорному ринку, а їх використання може призвести до серйозних наслідків, таких як шахрайство, маніпуляції або навіть нанесення шкоди пацієнтам [2], [3]. Згідно зі звітом Beazley за 2022 рік, медичний сектор став найбільш атакованим серед усіх галузей, що перебувають під впливом кіберзлочинців [1]. Зокрема, за даними HIMSS (Healthcare Information and Management Systems Society) за 2021 рік, 42% організацій у сфері охорони здоров'я відзначили напади за допомогою програм-вимагачів (ransomware), що є значним показником, порівняно з іншими галузями [2]. Це також підтверджується дослідженнями Verizon (2021), які вказують, що 34% всіх витоків даних у всіх секторах сталися саме в медичній сфері, що вдвічі більше, ніж у фінансовій сфері (17%) [3].

Особливу увагу слід приділити атакам програм-вимагачів, які є однією з основних загроз для медичних установ. Згідно з дослідженням HIMSS (Healthcare Information and Management Systems Society) за 2021 рік, 42% організацій у сфері охорони здоров’я відзначили напади за допомогою програм-вимагачів (ransomware), що свідчить про значний ризик для цієї галузі [2]. Це не тільки ускладнює роботу медичних працівників, але й може призвести до значних фінансових витрат, зокрема через вимогу викупу або витрати на відновлення даних [1], [3].

Іншою серйозною проблемою є крадіжка медичних даних. За звітами Verizon (2021), витоки медичних даних становлять 34% усіх витоків у всіх секторах, що вдвічі більше, ніж у фінансовій сфері (17%) [3]. Основні причини включають відсутність належних заходів безпеки, використання застарілих систем або людський фактор, зокрема, помилки персоналу або відповідні зловмисні дії [1], [2].

Таким чином, статистика свідчить про серйозну проблему кіберзагроз у медичному секторі, що вимагає термінових заходів щодо посилення безпеки

медичних даних [3]. Ці загрози можуть мати серйозні наслідки для пацієнтів, організацій і всієї сфери охорони здоров'я, підкреслюючи необхідність впровадження нових технологій для захисту даних, таких як блокчейн [4], [5].

Традиційні системи зберігання та обробки медичних даних зазвичай базуються на централізованих базах даних, які використовуються лікарнями, клініками, лабораторіями або страховими компаніями. Однак їхня централізація робить їх уразливими до атак, у той час як блокчейн дозволяє розподілену обробку даних із високим рівнем захисту [4], [5]. Ці системи мають низку уразливостей, які знижують їхню ефективність та безпеку:

Централізована структура як ключова слабкість. Вони залежні від одного або кількох серверів, які виступають точками контролю. Це створює кілька основних ризиків:

- Єдина точка відмови (Single Point of Failure): якщо центральний сервер зазнає збою (наприклад, через фізичну поломку, кібератаку або природну катастрофу), система перестає функціонувати, що призводить до втрати доступу до важливих даних.
- Атаки на центральний вузол: зловмисники часто націлюються на головні сервери, оскільки компрометація такого вузла дає доступ до всієї системи даних.

Нестача ефективного шифрування. Попри використання шифрування в багатьох традиційних системах, вони часто:

- не застосовують шифрування "на рівні даних" (data-at-rest encryption), залишаючи файли вразливими під час зберігання;
- не забезпечують наскрізного шифрування під час передачі даних між системами;
- покладаються на застарілі криптографічні протоколи, які вже не відповідають сучасним стандартам.

Відсутність механізмів відстеження змін (audit trail). У традиційних базах даних зазвичай складно або неможливо відстежити всі зміни, зроблені у записах. Це створює кілька загроз:

- Неможливість виявити несанкціоновані зміни: зловмисник може змінити дані без слідів, що може мати критичні наслідки для пацієнта, наприклад, зміна інформації про алергію чи історію лікування.
- Обмежена прозорість: як медичні установи, так і пацієнти не можуть бути впевненими у тому, хто і коли отримував доступ до їхніх даних.

Блокчейн-технології відіграють ключову роль у трансформації підходів до захисту медичних даних у цифрових системах охорони здоров'я. Вразливість традиційних методів зберігання та обміну інформацією, таких як централізовані бази даних, стала причиною багатьох витоків і атак. Наприклад, за звітами Verizon (2021), 34% усіх витоків даних відбуваються саме у медичній сфері, що свідчить про серйозну проблему [3]. Це підкреслює необхідність інноваційних рішень, серед яких блокчейн займає провідне місце.

Блокчейн пропонує децентралізований, прозорий і захищений спосіб зберігання даних, який суттєво знижує ризики несанкціонованого доступу, крадіжки або втрати інформації. Основною перевагою цієї технології є її незмінність даних. Як зазначено в дослідженні Kuo та інших (2017), кожен запис у блокчейні захищений криптографічними методами і після підтвердження не може бути змінений або видалений без згоди учасників мережі [4]. Це є надзвичайно важливим у сфері охорони здоров'я, де цілісність інформації про пацієнтів відіграє вирішальну роль у наданні якісних послуг.

Наприклад, збереження історій хвороби, результатів аналізів і планів лікування в блокчейні гарантує їх точність і незмінність, що допомагає запобігти помилкам у діагностиці та лікуванні. За дослідженням Angraal та ін. (2017), блокчейн також дозволяє відстежувати зміни в даних, забезпечуючи повну прозорість для медичних установ [5].

Ще однією важливою перевагою є децентралізація, яка робить блокчейн ідеальним для захисту медичних даних. У традиційних системах дані зберігаються на централізованих серверах, які стають головною мішенню для кіберзлочинців. За даними HIMSS (2021), медичні організації все частіше стають жертвами програм-вимагачів (ransomware), що призводить до блокування доступу до важливих медичних даних [2]. У блокчейні ж інформація розподіляється між багатьма вузлами мережі, що значно ускладнює централізовані атаки. Як зазначено в звіті Beazley (2022), цей підхід забезпечує вищий рівень безпеки навіть у разі компрометації одного з вузлів [1].

Блокчейн також дозволяє забезпечити прозорість і контроль доступу до медичних даних. Використання смарт-контрактів – автоматизованих програм, що виконують задані умови, дозволяє надавати доступ до даних тільки уповноваженим особам або організаціям. Наприклад, пацієнт може самостійно визначити, хто і коли матиме доступ до його медичної інформації, та переглядати всі дії, пов'язані з цими даними. Це значно підвищує рівень довіри між пацієнтом і медичними установами. Цей підхід підкріплюється дослідженнями Angraal та ін. (2017), які наголошують на важливості забезпечення безпечного доступу до медичних даних за допомогою блокчейну [5].

В умовах зростання міжнародної співпраці у сфері охорони здоров'я блокчейн може спростити обмін медичними даними між установами, країнами та континентами, забезпечуючи їх безпеку і доступність. Наприклад, у випадку пацієнта, який проходить лікування в різних країнах, блокчейн дозволить лікарям оперативно отримати необхідну інформацію про його стан без ризику порушення конфіденційності. Це важливо в контексті глобальної цифровізації медицини, що відзначено у звітах HIMSS (2021) [2] та Beazley (2022) [1].

Крім того, блокчейн-технології допомагають боротися з проблемами фальсифікації даних, зокрема медичних звітів або результатів досліджень. Завдяки незмінності записів та можливості відстеження кожної транзакції у ланцюгу блоків, можна швидко виявити будь-які спроби маніпуляції даними. Це особливо важливо для фармацевтичної індустрії, де достовірність даних

досліджень має вирішальне значення, як зазначено у дослідженнях Kuo та інших (2017) [4].

Однак варто зазначити, що впровадження блокчейну у медицину має й певні виклики, зокрема пов'язані зі складністю масштабування технології, високими енергетичними витратами (у разі використання Proof of Work) і необхідністю адаптації законодавства. Незважаючи на ці труднощі, потенціал блокчейн-технологій для захисту медичних даних є надзвичайно високим. Він дозволяє створити безпечне, прозоре та ефективне середовище, що відповідає сучасним вимогам до цифровізації медицини, як підкреслюють дослідження Angraal та ін. (2017) [5] і Beazley (2022) [1].

Таким чином, блокчейн-технології забезпечують не лише підвищення рівня безпеки медичних даних, але й сприяють розвитку нових підходів до обміну інформацією, інтеграції медичних систем та підвищення довіри до цифрових рішень у сфері охорони здоров'я. Це робить їх ключовим інструментом у сучасному світі, де захист конфіденційної інформації є одним із головних пріоритетів, що повністю підтверджується у звітах HIMSS та Verizon (2021) [2][3].

Новизна цієї роботи полягає у всебічному дослідженні можливостей застосування блокчейн-технологій для захисту медичних даних у цифрових системах охорони здоров'я. У роботі пропонується інноваційний підхід до інтеграції децентралізованих реєстрів для забезпечення конфіденційності, цілісності та доступності медичної інформації. Зокрема, увага зосереджується на використанні смарт-контрактів для автоматизації управління доступом до даних та мінімізації ризиків несанкціонованого втручання.

Новизна також проявляється у комплексному аналізі переваг і викликів блокчейн-технологій, включаючи їхню адаптацію до медичних систем різного масштабу. Окрім цього, робота пропонує перспективні сценарії використання блокчейну для створення глобальних медичних мереж, що забезпечують безпечний обмін інформацією між країнами та установами.

ПЕРЕЛІК ПОСИЛАНЬ

1. Cyber Services Snapshot: Healthcare Sector Insights [Electronic resourse]. - Access mode: https://www.beazley.com/en-US/cyber-services-snapshot/
2. Healthcare Cybersecurity Survey Report [Electronic resourse]. - Access mode: https://www.himss.org/
3. Data Breach Investigations Report [Electronic resourse]. - Access mode: https://www.verizon.com/about/news/verizon-2021-data-breach-investigations-report
4. Kuo, T. T., Kim, H. E., & Ohno-Machado, L. (2017). Blockchain distributed ledger technologies for biomedical and healthcare applications. Journal of the American Medical Informatics Association, 24(6), 1211–1220. doi: https://doi.org/10.1093/jamia/ocx068
5. Angraal, S., Krumholz, H. M., & Schulz, W. L. (2017). Blockchain technology: Applications in health care. Circulation: Cardiovascular Quality and Outcomes, 10(9), e003800. doi:https://doi.org/10.1161/CIRCOUTCOMES.117.003800

УДК 004.056:336.7

А.О. Мущак[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

КІБЕРБЕЗПЕКА У ФІНАНСОВИХ І ПЛАТІЖНИХ СИСТЕМАХ

Анотація. У роботі розглянуто основні кіберзагрози у фінансових і платіжних системах, запропоновано методи їх нейтралізації, зокрема багатофакторну автентифікацію, використання штучного інтелекту та моніторинг аномалій у реальному часі.

Ключові слова: *кібербезпека, фінансові системи, платіжні системи, багатофакторна автентифікація, машинне навчання, штучний інтелект, виявлення шахрайства, інфраструктура безпеки, захист даних, моніторинг у реальному часі.*

Вступ. Фінансові системи виступають базисом для стабільності та зростання економік усього світу. Зі зростанням обсягів безготівкових розрахунків зростають і ризики атак на платіжні системи. За даними звіту McAfee за 2023 рік, щороку обсяг збитків від кібератак на фінансові установи перевищує $3 трлн .

Однією з причин уразливості є цифровізація всіх етапів фінансових процесів, що робить системи більш відкритими для атак. Іншою проблемою є недостатнє оновлення систем безпеки, що дає змогу зловмисникам використовувати відомі уразливості.

Метою даної роботи є аналіз актуальних загроз у фінансовій галузі та розробка рекомендацій для підвищення кібербезпеки.

Постановка задачі. Сучасні фінансові та платіжні системи стикаються з постійними викликами, пов’язаними із забезпеченням кібербезпеки. Основною проблемою є високий рівень складності цих систем, де поєднуються численні користувачі, пристрої, транзакції та інтеграція з глобальними мережами.

Задачі дослідження включають:

- Визначення основних векторів атак на фінансові системи.
- Розробка підходів для автоматичного виявлення підозрілих транзакцій.
- Інтеграція сучасних технологій, таких як штучний інтелект і машинне навчання, у процеси моніторингу та аналізу загроз.
- Аналіз ефективності багатофакторної автентифікації та інших методів захисту користувачів.
- Створення рекомендацій для мінімізації ризиків та підвищення надійності платіжних систем.

Класифікація загроз

1. Вектор атак на клієнтів. До основних типів атак на клієнтів належать:

- **Фішинг**. Зловмисники розсилають підроблені листи або створюють вебсайти, що імітують справжні фінансові платформи.
- **Злом облікових записів**. За допомогою крадіжки даних для входу або brute-force атак зловмисники отримують доступ до банківських рахунків.

2. Загрози для інфраструктури

- Інфраструктура фінансових систем стає ціллю атак, таких як:
- **DDoS-атаки** для виведення з ладу серверів.
- **Використання уразливостей ПЗ** для проникнення в мережу банку.

3. Атаки на мобільні платформи.

Мобільні додатки є привабливими цілями, оскільки вони часто містять недоліки у кодуванні або недостатньо захищені API.

Методи виявлення шахрайства. Для підвищення ефективності захисту фінансових систем застосовуються сучасні методи аналізу великих даних і моделі машинного навчання. Одним із популярних підходів є використання логістичної регресії.

Формула логістичної регресії

Логістична регресія широко використовується для оцінки ймовірності шахрайства. Її основне рівняння:

$$P(y = 1 \mid X) = \frac{1}{1+e-(\beta 0+\beta 1x1+\beta 2x2+\cdots+\beta nxn)} \quad (1)$$

де:

$P(y = 1 \mid X)$ – ймовірність того, що транзакція є шахрайською.

$x_1, x_2, \ldots, x_n$ – характеристики транзакції (сума, місце проведення, час).

$\beta_0, \beta_1, \ldots, \beta_n$ – параметри моделі, які налаштовуються під час навчання.

Ця модель дозволяє автоматично класифікувати транзакції на основі аналізу їх характеристик.

Підходи до забезпечення безпеки:

1. Використання штучного інтелекту.

AI дозволяє проводити аналіз у реальному часі та автоматично виявляти відхилення в поведінці клієнтів. Приклади застосування:

- Аналітика великих даних для виявлення аномалій.
- Генерація попереджень про підозрілі транзакції.

2. Захист каналів передачі даних

Для зниження ризиків атаки "людина посередині" використовується шифрування TLS

3. Крім того, впроваджуються технології, такі як HSTS, для забезпечення захищеного з'єднання.

3. Резервні копії та планування відновлення

У разі атак на інфраструктуру фінансові установи повинні мати плани відновлення (Disaster Recovery Plan), які включають:

- Резервне копіювання важливих даних.
- Відновлення систем у найкоротший термін.

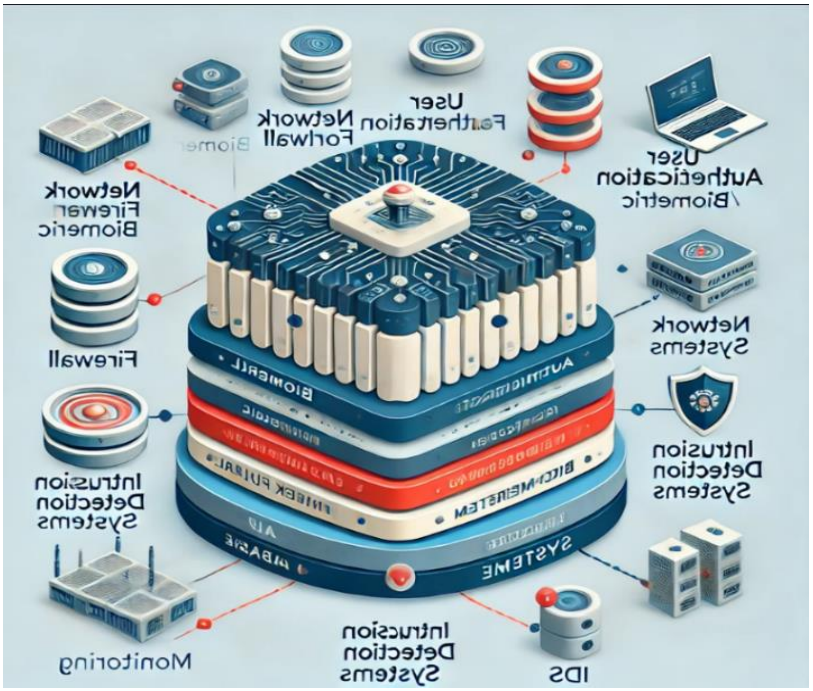


Рис. 1. Схема виявлення шахрайства у фінансовій системі.

Таблиця 1.

Порівняння основних методів захисту

Метод	Переваги	Недоліки
Логістична регресія	Простота впровадження	Менш ефективна для складних даних
Глибоке навчання	Висока точність	Високі обчислювальні витрати
Багатофакторна автентифікація	Простота використання	Потребує технічної підготовки користувачів

Рекомендації

- **Використання передових технологій**. Впровадження AI та моделей машинного навчання для автоматизації процесів.
- **Навчання персоналу.** Регулярне проведення тренінгів із кібергігієни.

- **Моніторинг аномалій**. Реалізація SIEM-систем для моніторингу у реальному часі.
- **Регулярний аудит**. Проведення перевірок на відповідність стандартам кібербезпеки, наприклад, ISO 27001.

Висновки. Сучасні загрози вимагають інтеграції технологій і заходів безпеки. Використання моделей прогнозування ризиків, моніторинг транзакцій і захист інфраструктури дозволяють значно знизити рівень ризику. У майбутньому розвиток квантових обчислень відкриє нові перспективи для забезпечення безпеки.

ПЕРЕЛІК ПОСИЛАННЯ

1. Symantec. "Internet Security Threat Report", 2023.
2. McAfee. "Financial Cybersecurity Trends", 2023.
3. ISO/IEC 27001:2013. "Information Security Management".
6. OWASP Foundation. "Top 10 Security Risks for Web Applications".
7. Bishop, M. "Introduction to Computer Security", 2022.

УДК 004.056

О.С. Ткачов[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

КІБЕРБЕЗПЕКА ЯК КЛЮЧОВА КОМПЕТЕНЦІЯ ДЛЯ ЦИФРОВОЇ ЕКОНОМІКИ.

Анотація. Розглянуто значення кібербезпеки як критично важливої компетенції для цифрової економіки, яка базується на інтернет-технологіях, хмарних обчисленнях і великих даних. Проаналізовано основні загрози цифрового середовища, які ставлять під загрозу організації та галузі. Визначено ключові напрями розвитку компетенцій у сфері кібербезпеки.

Ключові слова: *інновації у кібербезпеці, цифрова економіка, компетенції, кібергігієна, кібератаки, інформаційна безпека, критична інфраструктура.*

Вступ. Цифрова економіка є рушієм глобальних змін у бізнесі, промисловості та державному управлінні. Її основою є інтернет, який забезпечує безперервний обмін даними, однак ця відкритість робить її вразливою до кібератак. За даними Cybersecurity Ventures, до 2025 року глобальні збитки від кібератак можуть досягти $10,5 трлн щорічно. Це ставить кібербезпеку в центрі уваги як держав, так і приватних організацій. Цифрова економіка в Україні розвивається. Державні послуги цифровізуються, додаток Дія популяризується, це створює нові виклики у сфері кібербезпеки. З’являються глобальні ініціативи у сфері кібербезпеки, такі як ENISA в ЄС, ініціативи США щодо захисту критичної інфраструктури.

Основна мета цієї роботи – підкреслити важливість кібербезпеки як компетенції, яка має бути розвинута на всіх рівнях: від індивідуального користувача до корпоративного сектора та урядових структур. Успіх цифрової економіки напряму залежить від здатності суспільства захистити дані, системи та процеси від загроз. Кібербезпека стає не лише технологічним завданням, але й ключовою компетенцією для кожного учасника цифрового середовища.

Постановка задачі. Для досягнення поставленої мети проведено аналіз ролі кібербезпеки в сучасній цифровій економіці, визначення ключових навичок, необхідних для забезпечення безпеки, та обґрунтування шляхів інтеграції цих компетенцій у різні сфери діяльності. Основні задачі:

1. Дослідити основні загрози цифрової економіки.
2. Розробити систему формування компетенцій у сфері кібербезпеки яка враховує нові типи загроз, як-от атаки на штучний інтелект чи постквантові криптографічні ризики.
3. Визначити методи підвищення обізнаності та розвитку компетенцій.

Основний зміст роботи.Загрози цифрової економіки. Цифровізація створює нові ризики, серед яких:

- Фішингові атаки: за статистикою Verizon Data Breach Report, близько 36% зломів починаються з фішингових листів.
- Ransomware-атаки(атаки програм-вимагачів): за даними Cybersecurity Ventures середній розмір викупу зріс до $1.85 млн у 2023 році.
- Кібершпигунство: атаки на інтелектуальну власність стають основною проблемою для компаній, які займаються інноваціями.
- DDoS-атаки: за статистикою, їхня частота збільшується щороку на 15%.
- Експлойти нульового дня: середній час на виправлення критичних вразливостей складає 205 днів.

Критична інфраструктура, включаючи енергетичні системи та транспорт, також стає мішенню для кіберзлочинців. Наприклад, атака на Colonial Pipeline у США в 2021 році паралізувала постачання палива в декількох штатах.

Ключові навички, необхідні для сучасної економіки: розуміння основ кібергігієни, безпечне використання паролів, захист особистої інформації, використання менеджерів паролів, виявлення підозрілих дій.

Технічні компетенції: налаштування мережевих захистів, використання інструментів моніторингу трафіку, управління інцидентами, впровадження моделі нульової довіри та мінімально необхідного доступу, наприклад, хмарні платформи AWS GuardDuty, Google Chronicle, можуть працювати в рамках Zero Trust підходу, контролюючи доступ до ресурсів на основі багатофакторної автентифікації та постійної перевірки.

Аналітичні навички: розуміння методів соціальної інженерії та прогнозування загроз.

Підходи до розвитку кіберкомпетенцій:

А) Освітні програми: наприклад, у країнах ЄС впроваджуються програми Cybersecurity Skills Development для школярів і студентів. В Україні подібний

курс "Кібербезпека" став частиною програми для спеціалістів із інформаційної безпеки. Україна є однією з перших країн-партнерів НАТО (разом з Тунісом), де створюється курс за програмою Generic Reference Curriculum on Cybersecurity. Прикладом успішної освітньої програми є проект Cyber Patriot у США, який орієнтований на підлітків.

Б) Технологічні інновації: використання штучного інтелекту, таких як Darktrace чи IBM QRadar, для автоматизованого виявлення аномального трафіку або інструментів блокчейну для захисту транзакцій. Машинне навчання дозволяє ідентифікувати патерни атак і створювати моделі для їхнього попередження. Впровадження блокчейн-технологій, децентралізовані мережі унеможливлюють злом баз даних, такі технології вже впроваджуються у Європі під назвою QKD (Quantum Key Distribution), а також блокчейн забезпечує децентралізацію та зменшує ризики від маніпуляції даними в енергетичних системах. У промислових системах блокчейн може використовуватися для відстеження змін у конфігурації обладнання. Також важливим є ізолювання пристроїв IoT від основної мережі, що дозволяє обмежити потенційний вплив атак, та використання IoT Gateway для забезпечення безпечного обміну даними між IoT-пристроями та основними системами.

В) Міжсекторальна співпраця: створення спільних платформ для обміну інформацією про кібератаки, наприклад, через ініціативи CERT (Computer Emergency Response Team). Тісна співпраця з командою CERT-UA, яка функціонує в складі Державної служби спеціального зв’язку та захисту інформації України. Захист комунікацій між критичними вузлами за допомогою квантових алгоритмів, які забезпечують небувалий рівень шифрування.

Систематичний підхід до формування кіберкомпетенцій охоплює три рівні розвитку: базовий, середній і високоспеціалізований. Кожен рівень спрямований на окремі аудиторії та має відповідні інструменти для навчання:

Базовий рівень - навички кібергігієни.

Цільова аудиторія: школярі, студенти неспеціалізованих спеціальностей, працівники нефахових галузей.

Основні теми: створення надійних паролів, розпізнавання фішингових атак, принципи захисту особистих даних.

Інструменти навчання: онлайн-курси, наприклад, "Cyber Hygiene Essentials" від провідних платформ (Coursera, Udemy). Вікторини для перевірки знань (наприклад, використання платформ Kahoot). Кампанії з підвищення обізнаності через соціальні мережі та відеоролики.

Середній рівень - компетенції для корпоративного сектору.

Цільова аудиторія: працівники компаній, IT-фахівці середньої ланки.

Основні теми: управління доступом, безпечна робота з корпоративними системами, базові знання про інструменти моніторингу загроз.

Інструменти навчання: Сертифікаційні курси, наприклад, CompTIA Security+ або Certified Information Security Manager (CISM). Проведення

кібернавчань (симуляція атак). Використання корпоративних політик для забезпечення кібербезпеки (BYOD, політики оновлення ПЗ).

Високоспеціалізований рівень - експертні навички.

Цільова аудиторія: кіберфахівці, співробітники CERT, аналітики SOC.

Основні теми: розробка захисних систем, аналіз шкідливого ПЗ, тестування на проникнення.

Інструменти навчання: Професійні сертифікати: Certified Ethical Hacker (CEH), GIAC Certified Incident Handler (GCIH). Навчальні платформи для спеціалістів: Hack The Box, TryHackMe, симуляційні платформи (RangeForce, Cyberbit). Лабораторії на базі віртуальних середовищ (Metasploit, Kali Linux).

Наукова новизна У статті запропоновано систематичний підхід до формування кіберкомпетенцій, що охоплює як базові, так і високоспеціалізовані навички. Визначено залежність між рівнем розвитку кібербезпеки та економічною стійкістю держави. Розроблено рекомендації щодо інтеграції інноваційних технологій для захисту критичних інфраструктур. Підхід інтеграції кібербезпеки в критичні інфраструктури доповнений сучасними технологіями, як-от блокчейн та AI, що не було так широко висвітлено раніше. Поєднання кіберкомпетенцій з технологіями, такими як QKD та AI, відкриває нові можливості для захисту критичних систем.

Висновки. Кібербезпека є однією з ключових компетенцій для сучасної цифрової економіки. Вона потребує розвитку через освіту, інновації та партнерство. Чим вищий рівень кібербезпеки, тим стійкішою є економіка держави до внутрішніх і зовнішніх загроз. Інвестиції в кібербезпеку слід розглядати як довгострокову стратегію економічного зростання. Реалізація комплексного підходу до формування цих навичок допоможе мінімізувати ризики, пов’язані з цифровізацією, та підвищити конкурентоспроможність компаній і держав на світовій арені. Важливо підтримувати розвиток національних CERT, стимулювати співпрацю між державним і приватним секторами, а також залучати інвестиції в інноваційні дослідження кібербезпеки. Запропонована система кіберкомпетенцій може бути впроваджена в освітні програми, корпоративні політики та державні ініціативи. Це дозволить досягти не лише підвищення безпеки, але й зміцнення довіри до цифрової економіки.

ПЕРЕЛІК ПОСИЛАНЬ

1. URL: https://www.enisa.europa.eu/
2. URL:https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf
3. URL:https://www.bdo.ua/uk-ua/insights-2/information-materials/2024/zero-trust-a-revolutionary-approach-in-modern-cybersecurity
4. URL:https://www.nato.int/cps/en/natohq/news_159840.htm?selectedLocale=en
5. URL:https://www.nato.int/cps/en/natohq/topics_157591.htm
6. URL:https://cert.gov.ua/about-us

УДК 004.415.3:681.6

Е.Д. Ібрагімов[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У ЗАХИСТІ ІНФОРМАЦІЇ

Анотація. Розглядається роль штучного інтелекту (ШІ) у забезпеченні інформаційної безпеки в умовах зростаючих кіберзагроз. Підкреслюється, що ШІ дозволяє автоматизувати процеси виявлення та протидії атакам, забезпечуючи проактивний підхід до захисту даних. Зазначено ключові напрямки застосування ШІ, зокрема аналіз аномалій, прогнозування поведінки загроз і автоматизацію управління ризиками. Вказано, що впровадження технологій ШІ сприяє зміцненню кіберзахисту у критичних галузях, таких як фінанси, державне управління та оборонна промисловість. Також акцентовано увагу на потенціалі інтеграції ШІ з іншими технологіями (блокчейн, квантові обчислення), що забезпечить нові можливості для інформаційної безпеки.

Ключові слова. *Штучний інтелект, ШІ, кібербезпека, інформація, інформаційна безпека, захист, процесси, аналітика загрози.*

Вступ. Сучасний світ цифрових технологій створює безпрецедентні можливості для обробки та збереження даних, проте разом із цим зростають і загрози інформаційній безпеці. У таких умовах штучний інтелект стає ключовим інструментом у боротьбі з кіберзлочинністю. Його застосування дозволяє не лише автоматизувати процеси виявлення загроз, а й забезпечувати проактивний підхід до захисту інформації. Завдяки алгоритмам машинного навчання, аналізу великих даних та здатності швидко адаптуватися до нових видів атак, ШІ стає основою сучасних систем кібербезпеки. Це не лише зміцнює захист критичної інфраструктури, а й сприяє створенню безпечного цифрового середовища для бізнесу та суспільства загалом.

Постановка задачі:

1. Дослідити вплив сучасних кіберзагроз на інформаційну безпеку.
2. Вивчити роль і можливості штучного інтелекту у забезпеченні захисту інформації.
3. Розробити рекомендації щодо впровадження ШІ для підвищення кібербезпеки у критичних галузях.
4. Оцінити перспективи інтеграції ШІ з іншими передовими технологіями для створення надійних систем кіберзахисту.
5. Визначити значення державної політики в розвитку технологій ШІ для забезпечення національної безпеки.

Основний зміст роботи:

1. Аналіз сучасних загроз інформаційній безпеці в умовах глобалізації та цифровізації.

2. Огляд можливостей штучного інтелекту у виявленні, аналізі та прогнозуванні кіберзагроз.

3. Розгляд застосування алгоритмів машинного та глибокого навчання для підвищення точності та ефективності систем кіберзахисту.

4. Визначення ключових напрямів використання ШІ:

   a. моніторинг мережевого трафіку;

   b. виявлення фішингових атак;

   c. запобігання вторгненням у критичну інфраструктуру.

5. Розгляд інтеграції ШІ з новітніми технологіями (блокчейн, квантові обчислення) для посилення захисту даних.

6. Аналіз значення інформаційної безпеки як складової національної безпеки.

7. Оцінка загроз та переваг використання ШІ в інформаційному захисті для бізнесу, держави та суспільства.

Штучний інтелект не тільки допомагає реагувати на загрози, але й змінює підхід до кіберзахисту, орієнтуючись на випередження атак. Завдяки технологіям аналізу аномалій, прогнозуванню поведінки загроз та автоматизації управління ризиками, сучасні системи захисту стають більш точними та ефективними. Зокрема, ШІ активно використовується для моніторингу мережевого трафіку, виявлення фішингових атак і запобігання вторгненням. Такі рішення знаходять своє застосування у фінансовому секторі, державному управлінні та інших критичних галузях, де інформаційна безпека має вирішальне значення.

Важливою невід’ємною складовою національної безпеки є інформаційна безпека держави, на яку впливають внутрішні та зовнішні чинники, зокрема, загальний рівень економічного, соціального й інформаційного розвитку країни та політична ситуація в країні та в цілому в світі. Зацікавленими у розвитку даного сектора є не тільки бізнес-структури, що активно застосовують нововведення у даній сфері, а й державні установи, для яких найголовнішим є питання національної безпеки.

В сучасних умовах глобалізації інформаційна безпека виступає одним з найголовніших чинників забезпечення умов реалізації національних інтересів, спроможності держави долати кризові явища при зовнішній агресії. Своєчасні ефективні заходи щодо управління інформаційною безпекою з боку держави, як основного суб’єкта забезпечення інформаційної безпеки, здатні подолати загрози соціально-економічному та політичному життю країни. Сфера оборони та безпеки в світі – це галузь номер один і вона зазнає серйозних змін від впровадження технологій штучного інтелекту, що змінює баланс сил між державами. Штучний інтелект представляє собою результат людської діяльності здатний до логічного мислення, управління своїми діями, обґрунтовування своїх рішень, які не може коректувати в разі зміни умов.

У майбутньому очікується подальша інтеграція штучного інтелекту з іншими технологіями, такими як блокчейн і квантові обчислення, що відкриє

нові горизонти для посилення кіберзахисту. Це підтверджує, що ШІ стає невід'ємною частиною сучасної цифрової епохи, формуючи надійнішу основу для збереження конфіденційності та цілісності даних.

На даний час в світі суспільний розвиток характеризується формуванням інформаційного суспільства. Україна активно приймає участь у формуванні єдиного світового ринку інформації та в процесах глобальної інформатизації. Саме інформаційний чинник відіграє значну роль у відстоюванні державних інтересів у державотворчому процесі та на міжнародній арені. Ключовою складовою процесу управління процесами та інститутами стає широкий та оперативний доступ до підвищення ефективності інформації. Масове впровадження новітніх систем обробки інформації та телекомунікацій супроводжує інформатизацію сучасного суспільства. Одним з чинників, що сприяє забезпеченню інформаційної безпеки є застосування технологій штучного інтелекту. Адже штучний інтелект є одним з трендових напрямів, яким охоплені всі розвинуті країни світу. У випадку відсутності необхідної уваги зі сторони держави до проблем штучного інтелекту Україна ризикує втратити можливість технологічного прориву. Глобальний ринок технологічних рішень на основі штучного інтелекту буде розділений між країнами-конкурентами, що ускладнить розвиток держави в стратегічно важливих галузях економіки та уповільнить її розвиток. Під штучним інтелектом розуміється комплекс технологічних рішень, що дозволяє імітувати когнітивні функції людини та отримувати при виконанні конкретних завдань результати, що дорівнюють результатам інтелектуальної діяльності людини.

Активний розвиток інформаційних технологій обумовлює актуальність вивчення проблем інформаційної безпеки, а саме:

- загроз для інформаційних ресурсів, різних засобів та заходів захисту;
- бар’єрів для проникнення;
- вразливих місць в системі захисту інформації.

Штучний інтелект (ШІ) відіграє важливу роль у кібербезпеці. Спочатку, ШІ використовував прості правила для відстеження мережевого трафіка та дій користувачів. Ці правила, створені людьми, допомагали виявляти підозрілу активність, але мали обмеження.Згодом, наприкінці 20-го століття, ШІ став більш розвиненим завдяки прогресу в машинному навчанні. Тепер він може самостійно встановлювати правила, зменшуючи потребу в ручному введенні даних. Використання алгоритмів глибокого навчання дозволило йому ефективніше виявляти потенційні загрози, роблячи кібербезпеку більш надійною. Сучасний генеративний штучний інтелект або інтелектуальний ШІ — третя хвиля, яка забезпечує багато переваг у сфері кібербезпеки, від автоматизації повторюваних завдань і зменшення людських помилок до використання прогнозової аналітики для підтримки виявлення загроз. ШІ також виявився безцінним в автоматизації реагування на інциденти безпеки.Штучний

інтелект (ШІ) дуже допомагає в автоматизації реагування на кібератаки. Команди з кібербезпеки використовують ШІ для швидкого аналізу загроз і відповіді на них. Технології, як-от машинне навчання та глибоке навчання, які є частинами ШІ, змінили спосіб роботи кібербезпеки. Сучасний ШІ може аналізувати великі обсяги даних, вчитися на них, виявляти закономірності та приймати рішення для виявлення та усунення потенційних загроз.

Прогнозуючий штучний інтелект може оптимізувати виявлення загроз і створювати передові рішення з кібербезпеки, узгоджені з ландшафтом загроз, що постійно змінюється. Ці системи штучного інтелекту самоконтролюються та самонавчаються і можуть застосовувати аналіз у непередбачуваних ситуаціях і приймати рішення на основі власних спостережень.ШІ може сприяти виявленню та передбаченню загроз такими способами:

- Аналізуйте великі набори даних у режимі реального часу: активний моніторинг вашої мережі загроз вимагає сортування величезних обсягів структурованих і неструктурованих даних. Без ШІ цей процес потребував би значного часу та людських ресурсів. ШІ може автоматизувати моніторинг загроз, зменшуючи кількість людських помилок і роблячи виявлення ефективнішим;
- Визначте незвичайну діяльність: ідентифікація ризиків має вирішальне значення для прогнозування ШІ в кібербезпеці. ШІ може аналізувати зміни мережі та використовувати ці шаблони для прогнозування. Він може використовувати поведінкову аналітику для виявлення незвичайних дій у межах і за межами вашої системи;
- Передбачте потенційні вектори атак: методи або шляхи, які кіберзловмисники використовують для доступу до системи або мережі, називаються векторами атак. ШІ відіграє вирішальну роль у прогнозуванні векторів атак на основі історичних даних. Він використовує методи машинного навчання для аналізу історичних даних на наявність шаблонів і аномалій, що може допомогти передбачити майбутні вектори атак. Алгоритми штучного інтелекту можуть аналізувати величезні масиви даних і виявляти приховані зв’язки в подіях, які не можливо виявити «неозброєним оком». Чим більше даних ви надаєте, тим більше він навчається, тому штучний інтелект з часом поліпшує свої можливості прогнозування.

В загальному під інформаційною безпекою необхідно розуміти сукупність засобів, методів та процесів (процедур), які забезпечують захист інформаційних активів та, як наслідок, гарантують збереження ефективності та практичної корисності як технічної інфраструктури інформаційних систем, так і відомостей, які в таких системах зберігаються та обробляються. Особливого значення та актуальності в спектрі суспільних відносин набуває використання штучного інтелекту в системі забезпечення інформаційної безпеки, оцінка та аналіз інформаційних загроз та практичне застосування штучного інтелекту в контексті інформаційного опору з метою захисту територіальної цілісності України та суверенітету, що належить до концептуальних основ діяльності суспільства.

В Україні важливою складовою реформування оборонно-промислового комплексу є впровадження інноваційних технологій штучного інтелекту. Адже провідні країни світу вивчають та використовують можливості штучного інтелекту в галузі оборони. Так, сучасна міжнародна група Thales Group3, що функціонує в сфері цифрової ідентифікації безпеки створює й випускає інформаційні системи для авіакосмічної та військової галузей, що є потужною основою для швидкого ефективного захисту різних секторів критичної інфраструктури (енергетика, хімічна промисловість, транспорт, екологія та ін.). Дана група сприяє в забезпеченні інформаційної безпеки державним органам, приватним компаніям та власникам об'єктів критичної інфраструктури. На перший погляд впровадження штучного інтелекту здається дорогим та неперевіреним рішенням, яке приймається тільки інноваційними компаніями з великим бюджетом. Однак його використання в управлінні інформаційною безпекою гостро необхідно для всіх компаній, незалежно від їх масштабів та сфери діяльності. На даний час кількість атак зростає, а ландшафт загроз змінюється з блискавичною швидкістю. Наприклад, продукти Kaspersky попереджають більше 700 млн онлайн-атак на квартал по всьому світу, а Cisco блокує 20 млрд мережевих атак в день. Беззаперечно, при такому обсязі злочинної діяльності зловмисники активно застосовують засоби автоматизації кібератак, в тому числі використовують технології штучного інтелекту для їх вдосконалення та трансформації, а також для обходу відомих засобів захисту.

У 2019 році світовий ринок технологій штучного інтелекту в інформаційній безпеці оцінюється експертами (MarketsandMarkets, Zion Market Research) у $8 млрд, з досягненням $30 млрд у 2025 році та щорічним зростанням на 23% (рис.1).

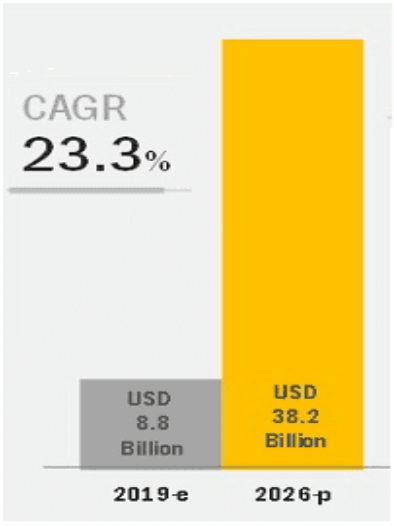


Рис. 1. Прогноз обсягу світового ринку технологій штучного інтелекту в інформаційній безпеці на 2019-2025 роки, за даними MarketsandMarkets

Організації, що впроваджують технології штучного інтелекту для поведінкового аналізу та передиктивної аналітики, отримують відчутні результати у вигляді підвищення ефективності виявлення атак, скорочення часу реагування та витрат на організацію безпеки. заявляють, що технології штучного інтелекту скорочують витрати на виявлення та реагування на загрози безпеці, та близько 75% заявляють про скорочення часу реагування (до 12%).

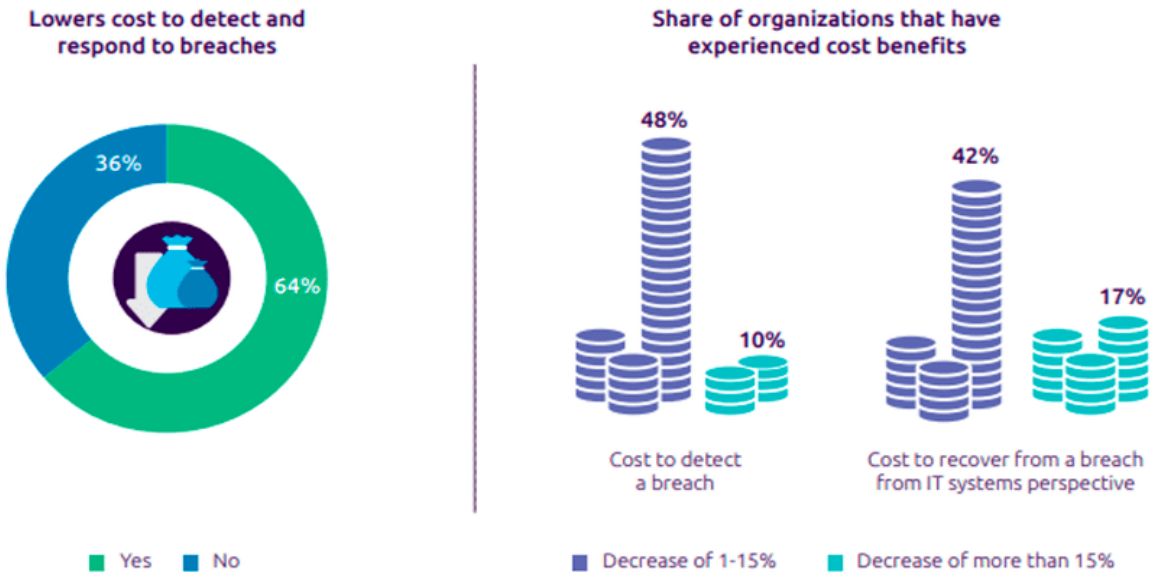


Рис. 2. Статистика скорочення витрат на детектування та реагування на інциденти під час використання технологій ІІ

Наукова новизна: Дослідження пропонує інноваційний підхід до використання штучного інтелекту (ШІ) у сфері інформаційної безпеки. Новизна полягає в інтеграції ШІ з сучасними технологіями, такими як блокчейн і квантові обчислення, для створення нових рішень у боротьбі з кіберзагрозами. Робота акцентує увагу на впровадженні алгоритмів прогнозування загроз, які дозволяють реалізувати проактивний підхід до захисту даних, та автоматизації реагування на інциденти безпеки, що знижує ризики людських помилок. Також обґрунтовано значення ШІ для формування національної політики у сфері кібербезпеки та його вплив на критичну інфраструктуру.

Висновоки: Штучний інтелект стає ключовим елементом сучасних систем кіберзахисту, забезпечуючи ефективну автоматизацію процесів виявлення та нейтралізації загроз. Його використання сприяє проактивному підходу до захисту інформації через аналіз аномалій та прогнозування атак. Інтеграція ШІ з іншими інноваційними технологіями, такими як блокчейн та квантові обчислення, відкриває нові перспективи для створення стійких до загроз систем. Водночас розвиток ШІ стає критично важливим для забезпечення національної безпеки та збереження конкурентоспроможності на міжнародній арені.

Інвестиції у дослідження та впровадження ШІ є необхідною умовою для зміцнення інформаційної безпеки держави.

ПЕРЕЛІК ПОСИЛАНЬ

1. Применение технологий искусственного интеллекта в информационной безопасностиhttps://www.anti-malware.ru/analytics/Technology_Analysis/using-artificial-intelligence-technologies-in-information-security
2. Роль штучного інтелекту в кібербезпеці: передбачення та запобігання атакам https://www.bdo.ua/uk-ua/insights-2/information-materials/2024/the-role-of-ai-in-cybersecurity-anticipating-and-preventing-attacks
3. "Укроборонпром" хоче використовувати штучний інтелект в "обороні" https://www.epravda.com.ua/news/2021/08/26/677230/
4. Artificial Intelligence (AI) In Cyber Security Market Will Reach to USD 30.9 Billion By 2025: Zion Market Research
5. https://www.globenewswire.com/news-release/2019/8/28/1907655/0/en/Artificial-Intelligence-AI-In-Cyber-Security-Market-Will-Reach-to-USD-30-9-Billion-By-2025-Zion-Market-Research.html

РОЗДІЛ 6

ПРОБЛЕМИ ДИСТАНЦІЙНОЇ ОСВІТИ

УДК 004.42

Л.В. Кабак[1], І.Г. Гуліна[1], Д.М. Мороз[1], В.В. Мамешин[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РОЗРОБКА СИСТЕМИ ДЛЯ РОЗПОДІЛЕНОГО НАВАНТАЖУВАЛЬНОГО ТЕСТУВАННЯ ТА АНАЛІЗУ РЕЗУЛЬТАТІВ У РЕАЛЬНОМУ ЧАСІ НА БАЗІ KUBERNETES

Анотація. У роботі розглянуто інструменти для тестування навантаження системи і її продуктивності. Наведено ключові принципи для створення архітектури системи розподіленого навантажувального тестування інтегрованої в CI/CD-конвеєр.

Ключові слова: *тестування, навантажувальне тестування, Kubernetes, контейнер, масштабування, Prometheus, CI/CD конвеєр.*

Вступ. Сучасні програмні додатки часто стикаються з викликами, пов’язаними з високим навантаженням, особливо в пікові моменти, коли кількість користувачів досягає рекордних значень. У таких умовах важливо не тільки перевірити, як система справляється з цими навантаженнями, а й забезпечити безперервну доступність та швидкість реагування.

Розподілене навантажувальне тестування є важливим інструментом для оцінки стабільності та продуктивності програмних систем під час пікових навантажень. Воно дозволяє моделювати великий обсяг запитів з різних географічних точок, виявляючи проблеми з продуктивністю, швидкодією і масштабованістю та допомагає усувати вузькі місця до розгортання.

Правильні інструменти та методології тестування продуктивності допомагають організаціям створювати високопродуктивні, надійні та масштабовані додатки, що забезпечує задоволення потреб клієнтів і сприяє зростанню бізнеса в конкурентному цифровому середовищі.

Постановка задачі. Для досягнення поставленої мети в роботі були сформовані наступні задачі:

- Провести аналіз існуючих рішень для проведення розподіленого навантажувального тестування
- Проаналізувати запропоноване рішення і оцінити його ефективність в порівнянні з іншими рішеннями
- Спроектувати архітектуру системи та розробити програмну частину
- Оцінити результати роботи системи і зробити висновки щодо переваг запропонованої архітектури

Основний зміст роботи. Для забезпечення масштабованості і відмовостійкості системи розподіленого навантажувального тестування використовувується технологія Kubernetes [1], яка відповідає за оркестрацію контейнерів з інструментами для навантаження

Kubernetes дозволяє автоматично масштабувати кількість контейнерів, що працюють в рамках тестування. Це дозволяє безперешкодно збільшувати або зменшувати їх кількість, щоб підвищити або знизити навантаження на систему. Це особливо важливо для масштабованих тестів, де потрібно генерувати великий обсяг запитів без перевантаження окремих вузлів.

Інструментом для проведення навантажувального тестування обрано k6 [2] Він дозволяє писати тести на JavaScript, які потім перетворюються на запити для імітації навантаження на веб-сайт або API. Завдяки інтуїтивному API сценаріїв і інтеграції CI/CD, k6 є чудовим вибором для постійного тестування продуктивності.

Для збору метрик продуктивності системи та результатів навантажувального тестування використовується Prometheus [3]. Це система для моніторингу та збору метрик, яка спеціально спроектована для роботи в розподілених та контейнеризованих середовищах, таких як Kubernetes. Prometheus автоматично здійснює опитування метрик з різних компонентів інфраструктури, таких як сервіси, додатки та контейнеризовані сервіси в Kubernetes.

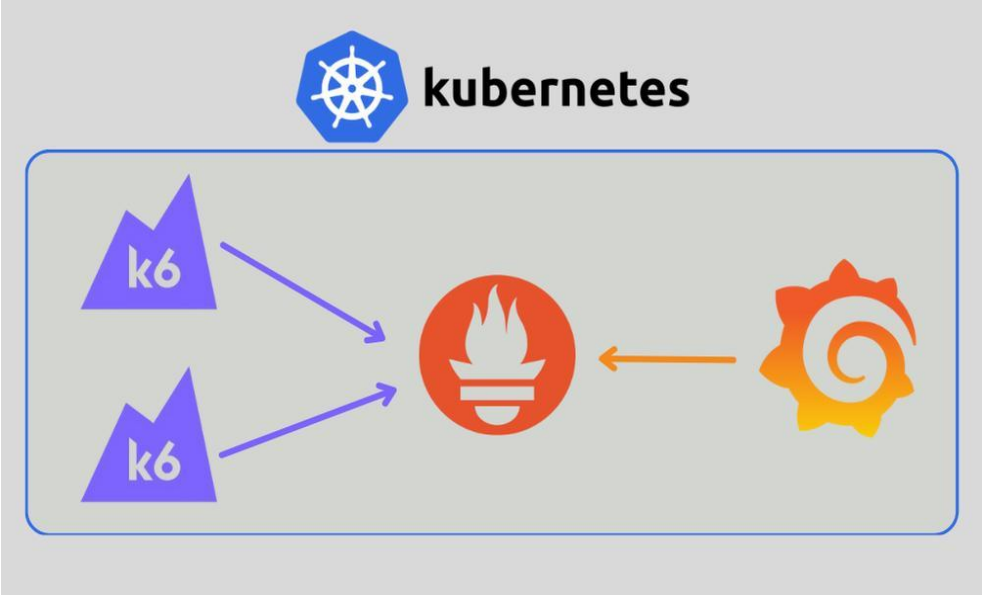


Рис. 1. Інструменти для тестування k6, Prometheus, Grafana

Зібрані метрики можуть бути використані для аналізу та візуалізації в реальному часі через інтерфейс Grafana. Grafana [4] дозволяє відображати зібрані дані про продуктивність у вигляді графіків, гістограм і таблиць, що дозволяє

здійснювати оперативний аналіз і швидко реагувати на можливі проблеми в системі.

Jenkins [5] – це інструмент для автоматизації процесів розробки, інтеграції та доставки програмного забезпечення (CI/CD) [6]. Він дозволяє автоматично запускати тести, збірки та розгортання на різних етапах життєвого циклу розробки, що забезпечує безперервну інтеграцію та доставку оновлень. Для розподіленого навантажувального тестування Jenkins може автоматично ініціювати тести за графіком або в залежності від подій у репозиторії, наприклад, при внесенні змін у код.

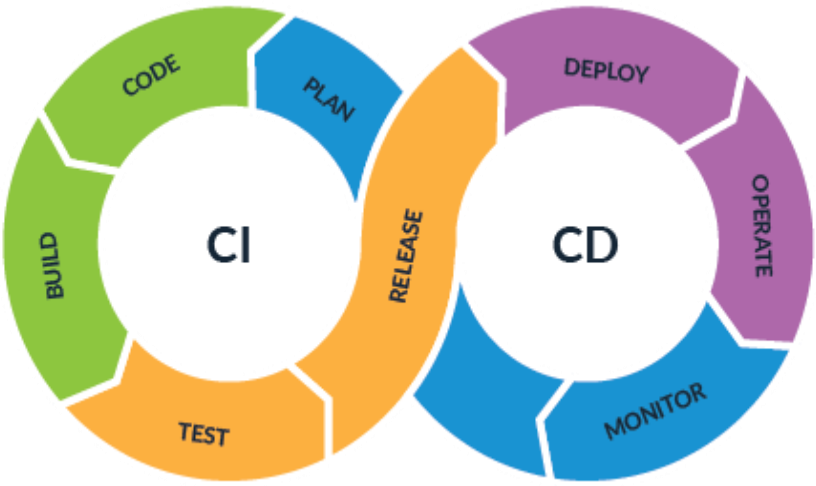


Рис. 2. Етапи CI/CD

Наукова новизна полягає у запропонованій архітектурі системи розподіленого навантажувального тестування, що має задовольняти потреби у здійсненні масштабованих тестів з великим обсягом запитів та централізованим зберіганням і відображенням метрик продуктивності і результатів тестування у реальному часі.

Висновки. У роботі були розглянуті основні необхідні компоненти системи розподіленого навантажувального тестування, зокерма технологія Kubernetes для оркестрації контейнерів та інструменти для проведення тестування, такі як k6, Prometheus, Grafana і Jenkins. Пропонована архітектура повинна ефективно масштабувати навантаження на систему та автоматизувати тестування та збір метрик.

ПЕРЕЛІК ПОСИЛАНЬ

1 Kubernetes Documentation. (Електрон. Ресурс) / Спосіб доступу: URL: https://kubernetes.io/docs/home/
2 K6 Testing Tool. (Електрон. Ресурс) / Спосіб доступу: URL: https://k6.io/
3 Prometheus. (Електрон. Ресурс) / Спосіб доступу: URL: https://prometheus.io/
4 Grafana. (Електрон. Ресурс) / Спосіб доступу: URL: https://grafana.com/

5 Jenkins Handbook. (Електрон. Ресурс) / Спосіб доступу: URL: https://www.jenkins.io/doc/book/
6 Continuous Integration. (Електрон. Ресурс) / Спосіб доступу: URL: https://martinfowler.com/articles/continuousIntegration.html
УДК 004.75: 37.091.39

УДК 004.056: 004.94

О.М. Алексєєв[1], В.В. Логвіненко[1]
[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

РОЗРОБКА ТА ВПРОВАДЖЕННЯ ПРОГРАМНОЇ СИСТЕМИ МОНІТОРИНГУ УСПІШНОСТІ СТУДЕНТІВ КУРСУ GOOGLE CLASSROOM ЗАСОБАМИ GOOGLE ТАБЛИЦЬ ТА GOOGLE APPS SCRIPT

Анотація. У роботі представлено розроблену програмну систему моніторингу успішності студентів із використанням Google Workspace, API Google Classroom та Google Apps Script. Система автоматизує збір, аналіз і візуалізацію даних, скорочуючи час моніторингу та підвищуючи точність.

Ключові слова: *Моніторинг, Google Classroom, Google Apps Script, API, UrlFetchApp, Хмарні технології, Автоматизація, Освітні платформи, Візуалізація даних.*

Вступ. У сучасному освітньому процесі, ефективний моніторинг успішності студентів є важливим аспектом забезпечення якісного навчання. Традиційні методи аналізу успішності займають багато часу, вимагають значних зусиль викладачів і не завжди дозволяють оперативно виявити проблемні моменти. У зв’язку з цим зростає потреба у впровадженні цифрових інструментів, які автоматизують ці процеси та забезпечують точний аналіз.

Одним із найбільш зручних рішень для автоматизації є використання хмарних платформ, таких як Google Workspace, які надають інструменти для інтеграції даних і їх візуалізації. Google Classroom широко використовується для організації навчання, однак засоби для аналізу успішності студентів, які вбудовані в платформу, є обмеженими [1].

Метою цієї роботи є розробка та впровадження програмної системи моніторингу успішності студентів із використанням Google Таблиць та Google Script на базі Фахового коледжу зварювання та електроніки імені Є.О.Патона. Така система дозволяє автоматизувати процес збору, обробки та аналізу даних із Google Classroom, що сприяє підвищенню ефективності викладачів і забезпечує доступ до аналітичної інформації в зручному форматі.

У роботі також розглядаються сучасні підходи до моніторингу успішності студентів та пропонуються інноваційні рішення, що базуються на інтеграції API сервісів Google.

Основний зміст роботи. Для реалізації програмної системи моніторингу успішності студентів було використано інструменти хмарної платформи Google Workspace, зокрема :

– **Google Classroom API, що** забезпечує доступ до структурованих даних певного курсу навчального процесу, таких як: список завдань, оцінки студентів, дедлайни тощо[2];

– **Google Apps Script** – мова сценаріїв, що використовується для автоматизації збору, обробки та аналізу даних[3]. Скрипти, в реалізації системи, виконують такі функції, як: отримання даних через API, фільтрацію та сортування, вивантаження результатів у Google Таблиці;

– **Google Таблиці** виконують роль засобу збереження, структурування та візуалізації даних. Вони також є основним графічним користувацьким інтерфейсом доступу викладачів до структурованої інформації певного курсу.

Алгоритм роботи та архітектура програмної системи:

1) **модуль збору даних –** інтеграція з Google Classroom API для отримання інформації (ідентифікаторів) про курси, завдання, студентів і оцінки;

2) **модуль обробки даних –** організація даних у таблицях за визначеною структурою (курси, теми, студенти, оцінки, дедлайни);

3) **модуль оновлення даних:**

– регулярне оновлення інформації за допомогою тригерів у Google Apps Script;

– застосування служби UrlFetchApp [4] для реалізації асинхронної обробки декількох курсів одночасно, що дозволяє оптимально зменшити час оновлення даних;

4) **модуль візуалізація – представлення** даних у Google Таблицях за допомогою зручних таблиць. Застосування інструменту умовного форматування Google Таблиць, до даних оцінок завдань, для ефективного відображення результатів та візуального аналізу.

Застосований підхід алгоритму та архітектура дозволяє автоматизувати обробку великих обсягів даних із мінімальними витратами часу, забезпечуючи викладачів зручними інструментами для моніторингу успішності студентів у реальному часі.

Особливістю реалізації програмної системи було застосування програмних скриптів, для інтеграції з Google Classroom API, який: викликає методи API для отримання даних про курси та оцінки студентів; зберігає отриману інформацію у Google Таблицях у вигляді таблиць із чітко визначеними полями (назва курсу, тема, прізвище та ім’я студента, завдання, статус виконання, оцінка тощо).

Для обробки даних використовуються функції сортування, фільтрації та агрегації, зокрема – генерації таблиць за певними темами курсу.

Система також використовує тригери Google Apps Script[5], які автоматично оновлюють дані у визначений час.

Інтерфейс користувача представлений Google Таблицями, які слугують як інструментом для візуалізації, так і точкою взаємодії з викладачем. Включені наступні елементи:

- **таблиці** з відфільтрованими даними за темами курсу та завданнями у порядку призначення;
- **панелі керування** для виконання додаткових дій – оновлення даних.

Представлена система дозволяє інтегрувати процеси моніторингу успішності в освітній процес, спрощуючи роботу викладачів і забезпечуючи якісну аналітичну інформацію в реальному часі. Розроблена програмна система моніторингу успішності студентів продемонструвала високу ефективність під час тестування та впроваджені на реальних даних із курсів у Google Classroom Фахового коледжу зварювання та електроніки імені Є.О. Патона.

У ході тестування було зібрано та оброблено дані з 50 навчальних курсів 1 семестру 2024-2025 н.р., з інтервалом оновлення – 2 години та часом оновлення 10 хв (всі 50 курсів, середній час оновлення одного курсу складає 3 секунди), що включали:

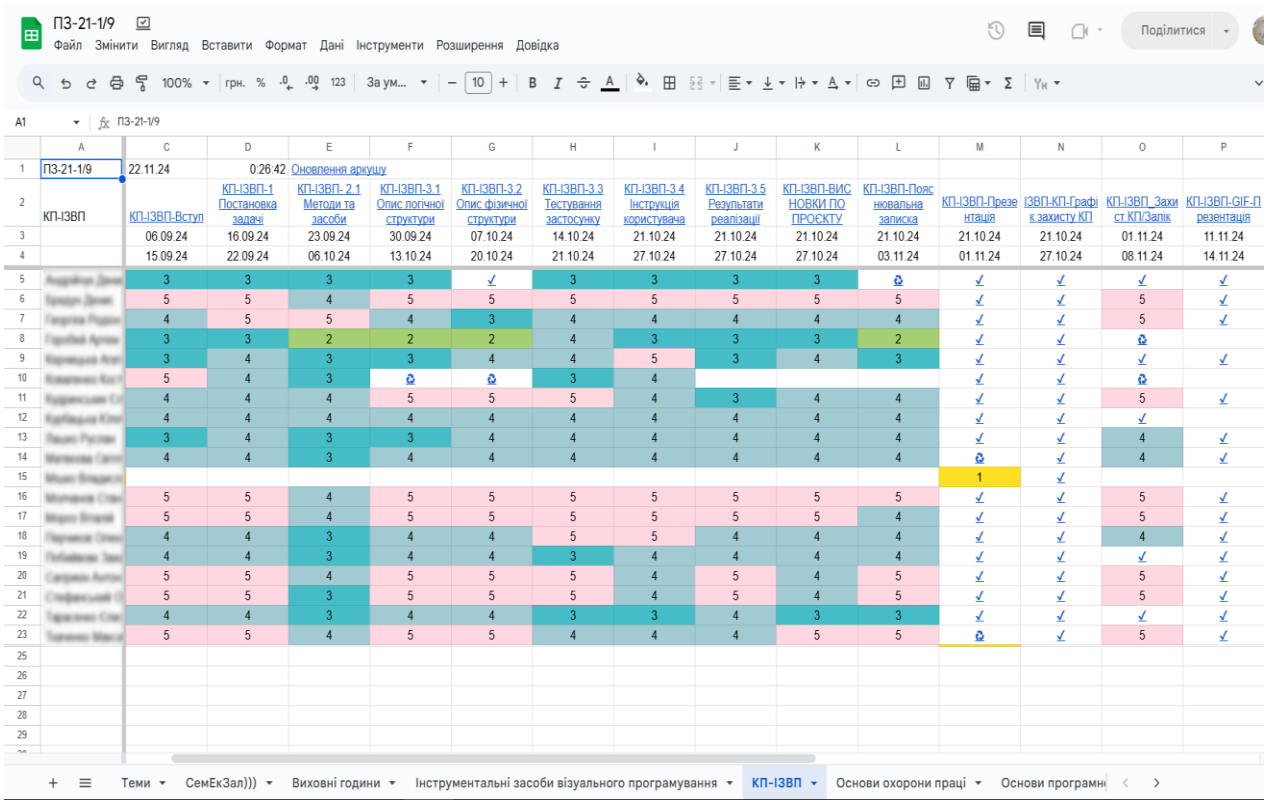
- понад 900 студентів;
- понад 500 тем;
- понад 5000 завдань різного типу (завдання, тести, запитання);
- інформацію про дати створення завдань, дедлайни, статус виконання та оцінки.

Система дозволила автоматично отримати структуровану інформацію та представити її у вигляді окремих документів Google Sheets за 50 курсами. Кожний документ містить Аркуші з зі зведеними таблицями за Темами Курсу, де таблиці містять дані виконаних та оцінених завдань за кожним студентом. На рис. 1 наведений, приклад табличного представлення, виконання та оцінювання завдань Теми – «КП_ІЗВП», Курсу – «ПЗ-21-1/9» навчальної групи.

Алгоритм роботи системи дозволяє автоматично формувати Аркуші з даними зведених оцінок за Темами, та тим самим формувати відомості з результатами: «Рубіжного оцінювання» та оцінок результатів Заліків та Екзаменів за семестр (див. рис.2).

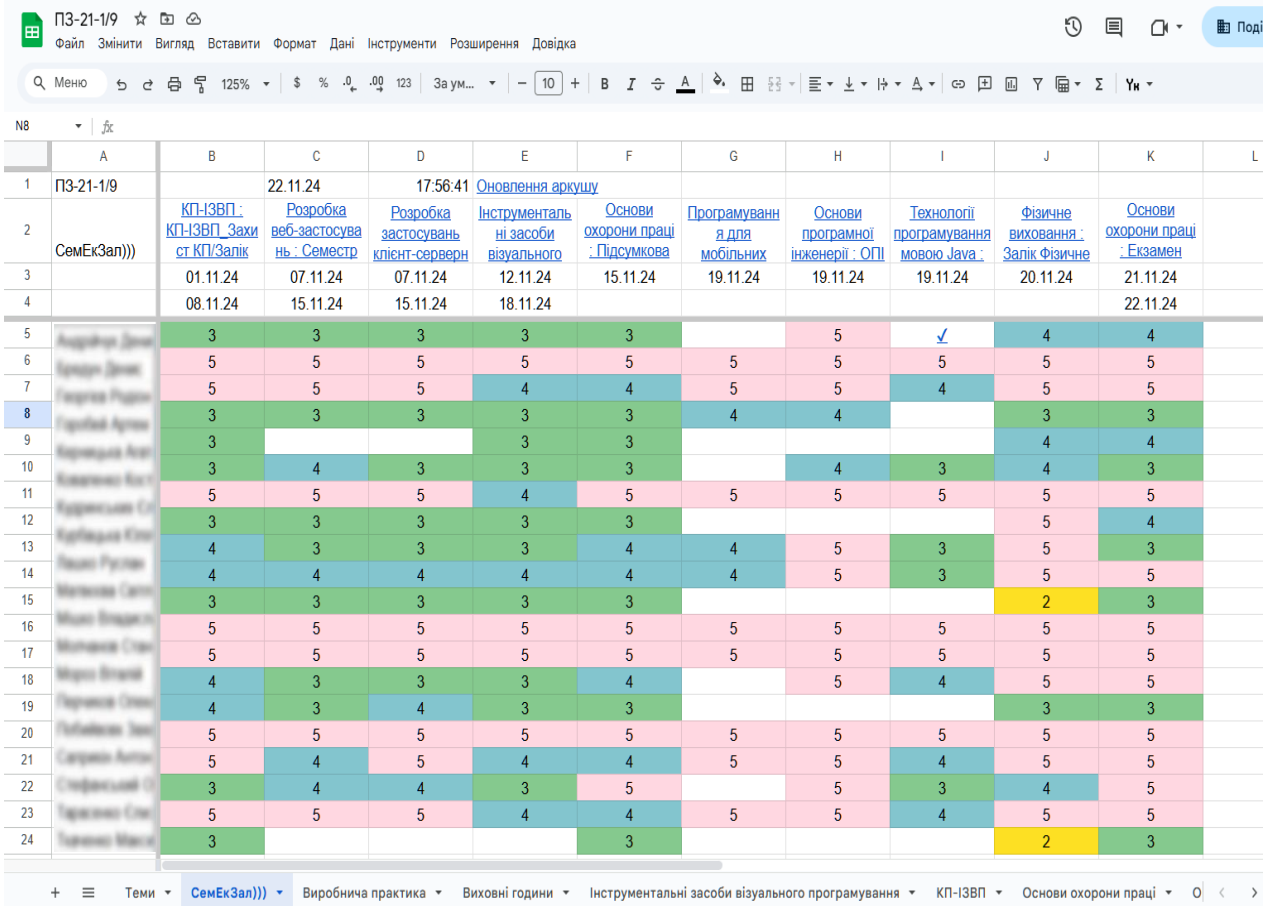
Аналіз ефективності у порівняні з ручними методами моніторингу показало, що:

- час на обробку даних зменшився у ∞ рази;
- зросла точність аналізу, оскільки система виключає людський фактор;
- викладачі отримали доступ до актуальної інформації в реальному часі, що дозволило швидко реагувати на проблеми (наприклад, відсутність виконання завдань окремими студентами).



	A	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	ПЗ-21-1/9	22.11.24	0:26:42	Оновлення аркушу											
2	КП-ІЗВП	КП-ІЗВП-Вступ	КП-ІЗВП-1 Постановка задачі	КП-ІЗВП- 2.1 Методи та засоби	КП-ІЗВП-3.1 Опис логічної структури	КП-ІЗВП-3.2 Опис фізичної структури	КП-ІЗВП-3.3 Тестування застосунку	КП-ІЗВП-3.4 Інструкція користувача	КП-ІЗВП-3.5 Результати реалізації	КП-ІЗВП-ВИСНОВКИ ПО ПРОЄКТУ	КП-ІЗВП-Пояснювальна записка	КП-ІЗВП-Презентація	ІЗВП-КП-Графік захисту КП	КП-ІЗВП_Захист КП/Залік	КП-ІЗВП-GIF-Презентація
3		06.09.24	16.09.24	23.09.24	30.09.24	07.10.24	14.10.24	21.10.24	21.10.24	21.10.24	21.10.24	21.10.24	21.10.24	01.11.24	11.11.24
4		15.09.24	22.09.24	06.10.24	13.10.24	20.10.24	21.10.24	27.10.24	27.10.24	27.10.24	03.11.24	01.11.24	27.10.24	08.11.24	14.11.24
5	[redacted]	3	3	3	3	✓	3	3	3	3	✪	✓	✓	✓	✓
6	[redacted]	5	5	4	5	5	5	5	5	5	5	✓	✓	5	✓
7	[redacted]	4	5	5	4	3	4	4	4	4	4	✓	✓	5	✓
8	[redacted]	3	3	2	2	2	4	3	3	3	2	✓	✓	✪	
9	[redacted]	3	4	3	3	4	4	5	3	4	3	✓	✓	✓	✓
10	[redacted]	5	4	3	✪	✪	3	4				✓	✓	✪	
11	[redacted]	4	4	4	5	5	5	4	3	4	4	✓	✓	5	✓
12	[redacted]	4	4	4	4	4	4	4	4	4	4	✓	✓	✓	
13	[redacted]	3	4	3	3	4	4	4	4	4	4	✓	✓	4	✓
14	[redacted]	4	4	3	4	4	4	4	4	4	4	✪	✓	4	✓
15	[redacted]											1	✓		
16	[redacted]	5	5	4	5	5	5	5	5	5	5	✓	✓	5	✓
17	[redacted]	5	5	4	5	5	5	5	5	5	4	✓	✓	5	✓
18	[redacted]	4	4	3	4	4	5	5	4	4	4	✓	✓	4	✓
19	[redacted]	4	4	3	4	4	3	4	4	4	4	✓	✓	✓	✓
20	[redacted]	5	5	4	5	5	5	4	5	4	5	✓	✓	5	✓
21	[redacted]	5	5	3	5	5	5	4	5	4	5	✓	✓	5	✓
22	[redacted]	4	4	3	4	4	3	3	4	3	3	✓	✓	✓	✓
23	[redacted]	5	5	4	5	5	4	4	4	5	5	✪	✓	5	✓

Рис. 1. Результати роботи системи моніторингу оцінок курсу ClassRoom



	A	B	C	D	E	F	G	H	I	J	K	L
1	ПЗ-21-1/9		22.11.24	17:56:41	Оновлення аркушу							
2	СемЕкЗал)))	КП-ІЗВП : КП-ІЗВП_Захист КП/Залік	Розробка веб-застосувань : Семестр	Розробка застосувань клієнт-серверн	Інструментальні засоби візуального	Основи охорони праці : Підсумкова	Програмування для мобільних	Основи програмної інженерії : ОПІ	Технології програмування мовою Java :	Фізичне виховання : Залік Фізичне	Основи охорони праці : Екзамен	
3		01.11.24	07.11.24	07.11.24	12.11.24	15.11.24	19.11.24	19.11.24	19.11.24	20.11.24	21.11.24	
4		08.11.24	15.11.24	15.11.24	18.11.24						22.11.24	
5	[redacted]	3	3	3	3	3		5	✓	4	4	
6	[redacted]	5	5	5	5	5	5	5	5	5	5	
7	[redacted]	5	5	5	4	4	5	5	4	5	5	
8	[redacted]	3	3	3	3	3	4	4		3	3	
9	[redacted]	3			3	3				4	4	
10	[redacted]	3	4	3	3	3		4	3	4	3	
11	[redacted]	5	5	5	4	5	5	5	5	5	5	
12	[redacted]	3	3	3	3	3				5	4	
13	[redacted]	4	3	3	3	4	4	5	3	5	3	
14	[redacted]	4	4	4	4	4	4	5	3	5	5	
15	[redacted]	3	3	3	3	3				2	3	
16	[redacted]	5	5	5	5	5	5	5	5	5	5	
17	[redacted]	5	5	5	5	5	5	5	5	5	5	
18	[redacted]	4	3	3	3	4		5	4	5	5	
19	[redacted]	4	3	4	3	3				3	3	
20	[redacted]	5	5	5	5	5	5	5	5	5	5	
21	[redacted]	5	4	5	4	4	5	5	4	5	5	
22	[redacted]	3	4	4	3	5		5	3	4	5	
23	[redacted]	5	5	5	4	4	5	5	4	5	5	
24	[redacted]	3				3				2	3	

Рис. 2. Результати збору оцінок з заліків та екзаменів за Темами курсу ClassRoom

Розроблена програмна система отримала позитивний відгук від викладачів та адміністрації ЗО, які відзначили її зручність у використанні, адаптивність під різні курси на базі платформи Google Classroom і можливість швидкої інтеграції у вже існуючий навчальний процес.

Перспективи використання за результати тестування свідчать про високий потенціал системи для впровадження в освітній процес, зокрема для:

- моніторингу успішності великих груп студентів;
- аналізу результатів за навчальними модулями;
- підготовки звітності для адміністрації навчальних закладів.

Переваги запропонованого підходу. Розроблена програмна система продемонструвала значний потенціал для автоматизації моніторингу успішності студентів. До її основних переваг можна віднести:

- **ефективність та економія часу:** автоматизований збір та обробка даних значно скоротили час, необхідний для аналізу успішності, у порівнянні з традиційними методами;
- **простота інтеграції -** використання Google Workspace[6] забезпечило безшовну інтеграцію із платформою Google Classroom без необхідності залучення стороннього програмного забезпечення;
- **гнучкість –** система дозволяє адаптувати скрипти під специфічні потреби викладачів, такі як додаткові метрики або фільтри;
- **зручність візуалізації** – результати моніторингу представлені у вигляді таблиць, що полегшує їх інтерпретацію.

Незважаючи на досягнуті результати, система має певні обмеження, зокрема:

- **залежність від інфраструктури Google –** використання Google Classroom API та Google Таблиць передбачає наявність стабільного доступу до інтернету та обмеження, пов’язані з квотами API;
- **проблеми з безпекою даних –** передача даних через хмарні сервіси потребує ретельного налаштування доступу та забезпечення конфіденційності інформації про студентів;
- **технічна складність.** Хоча система є зручною у використанні, її початкове налаштування вимагає базових знань програмування (Google Apps Script) та роботи з API Google Apps.

Для вдосконалення системи та розширення її функціональних можливостей перспективними є такі напрями:

1) **Графіки успішності:**

– динаміка виконання завдань у часі;

– порівняння середнього бала студентів за різними групами (за курсом та окремими темами);

– пропущених дедлайнів – візуалізація частки невиконаних завдань для окремих студентів чи груп.

2) **Зведені таблиці:**

– рейтинг студентів за успішністю;

– список проблемних зон (Завдання чи Теми, які викликали найбільші труднощі).

– кількість виконаних та невиконаних завдань кожного студента;

– відсоток виконання завдань у задані строки.

3) **Прогнозування успішності:** впровадження алгоритмів машинного навчання для передбачення результатів студентів на основі історичних даних.

4) **Розширення функціональності:** додавання інструментів для аналізу участі студентів у дискусіях, кількості переглядів навчальних матеріалів тощо.

Таким чином, система вже зараз забезпечує значний внесок у підвищення ефективності моніторингу успішності студентів, а подальший розвиток може зробити її незамінним інструментом для сучасного освітнього процесу.

Наукова новизна дослідження розробки полягає у створенні програмної системи, що інтегрує хмарні технології Google Workspace для автоматизованого моніторингу успішності студентів. Вперше:

– автоматизовано збір і аналіз даних з Google Classroom за допомогою API, забезпечуючи швидкість і точність обробки;

– поєднано автоматизацію та візуалізацію, що дозволяє викладачам отримувати інтерактивну аналітику в реальному часі;

– розширено можливості моніторингу - система відображає динаміку виконання завдань, пропущені дедлайни;

– забезпечено адаптивність - архітектура системи дозволяє легко адаптувати її до різних навчальних курсів.

Ця система вдосконалює цифрові інструменти для моніторингу освітнього процесу, сприяючи автоматизації та підвищенню ефективності аналізу.

Висновки. Представлено розробку та впровадження програмної системи моніторингу успішності студентів на базі ЗО, що базується на інтеграції Google Classroom API, Google Apps Script та Google Таблиць [3]. Результати роботи демонструють, що використання хмарних технологій значно підвищує ефективність аналізу навчальних даних, автоматизуючи трудомісткі процеси та забезпечуючи викладачів зручними інструментами для оперативного прийняття рішень. Впровадження системи у навчальний процес дозволяє оптимізувати роботу викладачів, підвищити прозорість та об’єктивність оцінювання, а також забезпечити якісний аналіз успішності студентів. Отримані результати підтверджують доцільність використання таких інструментів в умовах цифрової трансформації освіти.

ПЕРЕЛІК ПОСИЛАНЬ

1. Vasenko, O., Yakuba, V., & Havrylov, I. (2023). Automating data analysis in scientific research of future education specialists using the Google Apps Script platform. Professional Education: Methodology, Theory and Technologies, (18), 48-65. [Електронний ресурс] Режим доступу: https://doi.org/10.31470/2415-3729-2023-18-48-65

2. Google Classroom API Documentation. [Електронний ресурс] Режим доступу: https://developers.google.com/classroom
3. Google Apps Script Documentation. [Електронний ресурс] Режим доступу: https://developers.google.com/apps-script
4. Srivastava Sachin. Let's Learn Google Apps Script: Customize and Automate Google Applications using Apps Script, Independently published, 2021. - 314 p.
5. Mcpherson Bruce. Going GAS: From VBA to Google Apps, O'Reilly Media, 2016. - 456p.
6. Sanderson D. Програмування Google App Engine: Build and Run Scalable Web Apps on Google's Infrastructure, 2nd Edition, O'Reilly Media, 2012. - 536 p. - ISBN10: 1-4493-9826-X, ISBN13: 978-1-4493-9826-2.

УДК 811.111:37.091.33-027.22

N.I. Bilan[1], K.V. Maurina[1]
[1]Dnipro University of Technology, Dnipro, Ukraine

ENHANCING ENGLISH SPECIALIST TEXT READING COMPREHENSION COMPETENCE BY COMBINING CONTENT VISUALIZATION METHODS

Abstract. Описано використання графічних методів візуалізації смислової структури англомовних фахових текстів в процесі роботи студентів університету над англомовними фаховими текстами. Обґрунтовано ефективність поєднання існуючих технік візуалізації для покращення розуміння фахових текстів, що мають комплексну змістовну структуру.
Key words*: *English specialist texts, semantic structure, visualization methods, arrow analysis

Introduction. The experience of conducting classes in the online mode accumulated by teachers of higher education institutions of Ukraine since 2019 provides extensive factual material which should be analysed and systematised on the basis of the methodology principles of teaching English in higher educational institutions in order to use those methods and types of work that have proved their effectiveness during the specified period.

The aim of this study is to analyse the use of graphical methods used to represent the semantic structure of English scientific and technical texts in the process of studying the topics of module blocks 2 and 3 of the discipline "Foreign language for professional purposes (English/German/French)".

The main body. These module blocks 2 and 3 topics are "Searching, Reading and Processing Foreign Language Information. Pre-reading, Skimming, Scanning and Active Reading of English Specialist Texts. Reading on Print and Electronic Media" and "Written Communication. Application Procedure".

While mastering active or intensive reading, students use such methods as asking questions, annotating the text, highlighting key points, taking notes and summarization to study academic material and complex subjects in depth, enhance their

comprehension and retention of information, develop critical thinking and analyze the content of assigned specialist texts.

In addition to these time-tested teaching methods, the discipline's programme includes the use of various means of visualizing scientific information. Students are encouraged to create and comment on a range of graphical means of presenting the information of the materials studied – graphs, schemes, diagrams, Mind Maps. Overall, this variety of methods of work with technical and scientific literature certainly results in students achieving the desired goal – deeper understanding and memorization of the content of texts on their speciality.

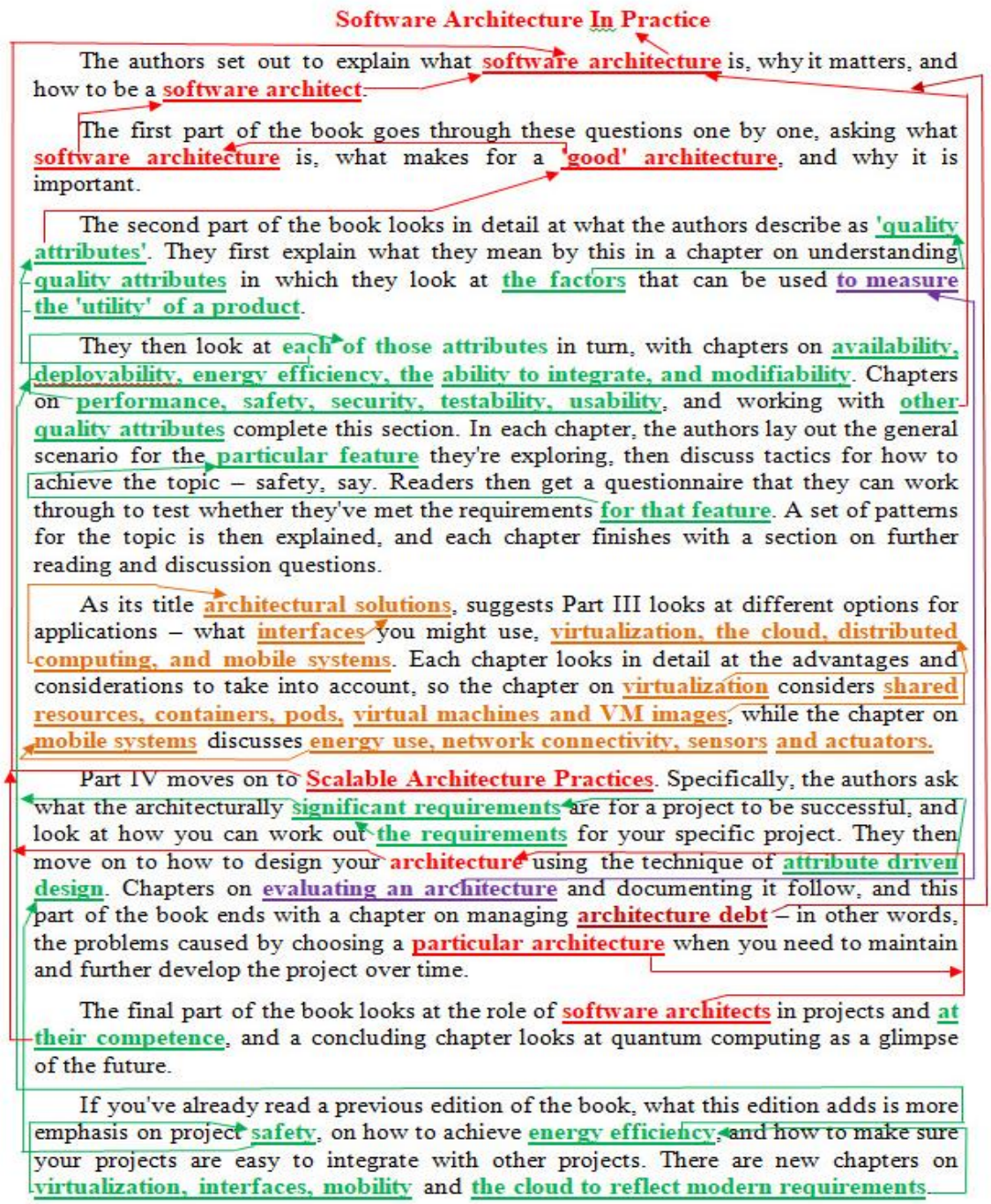
However, the practice of working on visualization tasks shows that not always 1st year students can successfully create graphic means of presenting information of complex academic materials. When exposed to the complex meaning representation systems of scientific texts, students find it difficult to identify larger meaning representation blocks within a whole text and the meaning representation units that make up those larger blocks.

To solve this problem, we used the method of the arrow analysis which allows establishing all the relationships between all the key concepts that form the semantic structure of a scientific text.

The effectiveness of the arrow analysis in identifying semantic relations not only in texts, but also in various systems is confirmed by numerous scientific and practical studies covering numerous spheres of life of modern humankind. We will mention only the fields of education [1] and processes occurring in socio-ecological systems [2]. Taking into account the efficiently and intensively used semantic potential of arrow symbols, information technology specialists are searching for solutions to the urgent task of making computers interpret arrow symbols, which are a frequent and universal component of communication diagrams [3].

Our choice of the method of analyzing the system of semantic units and blocks of scientific texts proved to be successful and the following example of an arrow analysis of a text combining excerpts from "Software Architecture in Practice" by Bass L., Clements P., Kazman R. [4] performed by the then first year student Maurina K. illustrates the case (pic.1).

According to Maurina K., her perception of the semantic structure of this text changed dramatically after applying the arrow analysis – the abundance of concepts presented a separate problem for comprehension which, combined with the ambiguity of the systemic relations existing between them, became even more difficult. However, the explicitness of the system of relations between the semantic elements of the analyzed text provided by the use of the arrow analysis not only explained the interconnection of these concepts, but also resulted in a deeper understanding of the latter.



Picture 1. The arrow analysis of the text “Software Architecture in Practice” performed by Maurina K.

Highlight colour differentiation of words and phrases expressing key concepts in the text:

- – the generalizing concepts of Software Architecture and software architect.
- – the “quality attributes” of 'good' Software Architecture.
- – the architectural solutions
- – measuring the product 'utility' and evaluating the “quality attributes” of 'good' Software Architecture.
- – the architecture debt

Conclusion. The conducted research and practical implementation of graphic methods of visualizing the semantic structure of English specialist texts indicates the effectiveness of combining existing visualization techniques in the process of university students' work on academic texts in the online mode. When the level of complexity of the texts being studied is rather high, the use of arrow analysis proves to be an effective solution due to its unique explanatory potential.

REFERENCES:

1. Essay writing - Achieving coherence. URL: https://flexiblelearning.auckland.ac.nz/essay-writing-2018/8.html (accessed: 20.01.2025)
2. Banitz, T., T. Hertz, L.-G. Johansson, E. Lindkvist, R. Martínez-Peña, S. Radosavljevic, M. Schlüter, K. Wennberg, P. K. Ylikoski, and V. Grimm. 2022. Visualization of causation in social-ecological systems. Ecology and Society 27(1):31. https://doi.org/10.5751/ES-13030-270131 URL: https://ecologyandsociety.org/vol27/iss1/art31/ (accessed: 22.01.2025)
3. Kurata, Y., & Egenhofer, M. J. (2008). The Arrow-Semantics Interpreter. Spatial Cognition & Computation, 8(4), 306–332. URL: https://doi.org/10.1080/13875860802148843 (accessed: 17.01.2025)
4. Bass L., Clements P., Kazman R. Software Architecture in Practice. 4th Edition, Addison-Wesley Professional. 464 p.

УДК 004.056: 004.94

О.В. Герасіна[1], І.О. Чертков[1]

[1]Національний технічний університет «Дніпровська політехніка», Дніпро, Україна

ВИЯВЛЕННЯ СПАМУ ЕЛЕКТРОННОЇ ПОШТИ ІЗ ВИКОРИСТАННЯМ МАШИННОГО НАВЧАННЯ ТА ШТУЧНОГО ІНТЕЛЕКТУ

Анотація. Обґрунтовано актуальність фільтрації спаму електронної пошти. Проаналізовано існуючі методи виявлення спаму електронної пошти за допомогою машинного навчання та систем штучного інтелекту. Визначено напрями подальших досліджень.

Ключові слова: *спам, штучний інтелект, фільтрація, класифікація, оптимізація, машинне навчання.*

Вступ. Електронні листи − один з найбільш розповсюджених методів сучасного спілкування, оскільки це дуже зручний спосіб передачі даних. Окрім того, сучасні поштові сервіси надають надійні методи превентивного захисту користувачів. Але не дивлячись на всі засоби захисту саме e-mail листи залишаються важливим і, напевно, найбільш улюбленим інструментом кібершахраїв [1-5].

Спам − небажані повідомлення у будь-якій формі, які надсилаються у великій кількості. Загроза спаму в електронній пошті збільшується щорічно і відповідає за понад 77% усього глобального трафіку e-mail.

Головна проблема ідентифікації сучасного спаму полягає у тому, що його дуже важко відрізнити від звичайної кореспонденції. За останні роки було винайдено чимало способів боротьби зі спамом, але зловмисники стежать за цим й винаходять всі нові прийоми для обходу фільтрів. До того ж нерідко фільтрація спам-листів приносить більше шкоди, чим користі: разом з рекламою не доходять до адресата й важливі повідомлення. Таким чином, всі дослідження в області боротьби з незапитуваною кореспонденцією є надзвичайно актуальними.

Наразі комплексний захист від спаму складається з наступних етапів: аналіз відправника, використання фільтрів, аналіз змісту листа. Дані етапи базуються на двох підходах – фільтрація за формальними ознаками повідомлення (за способом посилки та оформленням) або за його змістом (семантичні методи фільтрації). Семантичні методи передбачають розпізнавання за змістом (словосполучення, евристики, статистика) або за зразками листів (за сигнатурами). Для роботи семантичних методів використовуються фільтри здатні до самонавчання, насамперед на основі систем штучного інтелекту (Artificial Intelligence – AI) та машинного навчання (Machine Learning – ML).

Отже, задача фільтрації спаму є задачею класифікації – визначення належності об'єкта (електронного повідомлення) до одного з заздалегідь виділених класів («спам» або «не спам») на підставі аналізу сукупності ознак, що характеризують даний об'єкт. Для вирішення задачі класифікації актуальним є використання методів ML / AI [6].

Ідея автоматичної класифікації спам/не-спам електронних листів за допомогою методів ML / AI наразі є цікавою для багатьох дослідників [4].

Мета роботи. Аналіз існуючих методів детектування спаму електронної пошти із використанням машинного навчання та штучного інтелекту.

Виклад основного матеріалу. Існують методи, які детектують спам електронної пошти як у форматі текстових, так і у форматі графічних даних. Зокрема методи, запропоновані у роботах [7-8] використовували набір зображень і текстових даних для виявлення спаму різними методами.

У роботі [7] авторами було використано Наївний баєсів класифікатор, алгоритм k-найближчих сусідів і зворотний алгоритм DBSCAN з експериментами на наборі даних Enron Corpus. Для розпізнавання тексту використоввалась бібліотека оптичного розпізнавання символів (OSR). Два експерименти проводились з етапами попередньої обробки та без них з оцінкою таких параметрів як Accuracy, Specificity, Sensitivity та Precision. Загалом NB працює ефективно зі значенням Accuracy 87%.

У роботі [8] авторами було розглянуто гібридний підхід вибору ознак TF-IDF (Term Frequency Inverse Document Frequency) і теорію наближених множин (грубих наборів, Rough Sets). Під час цього експерименту вручну створюється набір даних із 169 електронних листів, що містить як текст, так і зображення. Для

видалення несуттєвих слів і генерації правил автори використали інструмент RSES (Rough Set Exploration System). Загальна концепція порівнюється по точності (Accuracy) із TDIDF-деревом рішень і показує ефективні результати.

Більшість з дослідників зосередилися лише на класифікації електронного спаму на основі тексту, оскільки спам на основі зображень можна відфільтрувати на початковому етапі попередньої обробки. Для інтелектуального аналізу текстових даних дуже часто використовують метод опорних векторів (SVM – Support Vector Machines). Деякі автори використовували SVM окремо (як у роботі [9]), деякі в інтеграції з іншими концепціями, такими як SVM та Наївний баєсів класифікатор (у роботі [10]), SVM та адаптивним кроковим алгоритмом пошуку зозулі (у роботі [11]), а також SVM та машинами екстремального навчання (у роботі [12]).

У роботі [9] авторами було використано SVM для класифікації виявлення e-mail спаму разом із латентно-семантичним аналізом для вибору характеристик. При цьому, для вилучення ознак авторами було запропоновано використання TF-IDF (Term Frequency – Inverse Document Frequency). Було встановлено, що запропонована концепція працює краще з точки зору точності (Accuracy) у порівнянні з іншими існуючими підходами (з оптимізацією роєм часток (Particle Swarm Optimization – PSO), з нейронною мережею та без використанням латентно-семантичного аналізу).

У роботі [10] авторами було запропоновано концепцію SVM та Наївного баєсового класифікатору для фільтрації e-mail спаму. При цьому останній може обробляти великий набір даних, а SVM – створювати розподіл на основі гіперплощини між різними категоріями об’єктів. Автори використовували набір даних Chinese Spam Email Dataset, отриманий від DATAMALL. Виходячи з результатів, автори повідомили про кращу ефективність запропонованого підходу у порівнянні з окремим використанням цих алгоритмів.

У роботі [11] авторами було модифіковано алгоритм пошуку зозулі за допомогою розміру кроку та SVM для класифікації e-mail спаму. Модифікований алгоритм було названо Stepsize Cuckoo Search, і він був використаний для вибору найкращого набору характеристик, а SVM – для розрахунку значення придатності та остаточної класифікації. Загальна продуктивність була оцінена для трьох ядер лінійної, квадратичної та поліноміальної SVM та виявилась кращою з метриками оцінки Specificity, Sensitivity та Accuracy у порівнянні з оригінальним алгоритмом пошуку зозулі разом із SVM.

У роботі [12] авторами було запропоновано використання SVM з машинами екстремального навчання для класифікації e-mail спаму. Машини екстремального навчання – це підхід до ML, який був запропонований для подолання багаторічного недоліку прямої нейронної мережі та використовується як підхід до навчання одношарової нейронної мережі. Результати запропонованого методу було порівняно на основі точності (Accuracy) та часу, витраченого на класифікацію спаму електронної пошти з того самого набору даних. З точки зору Accuracy, SVM працює краще з 94,06 % порівняно з

машинами екстремального навчання (для них Accuracy 93,04 %). Але для кожного випадку SVM витрачає більше часу, ніж машини екстремального навчання, отже, останні кращі за SVM з точки зору витраченого часу.

Також, крім значної кількості робіт із SVM [9-12], існує багато інших робіт, в яких разом з методами ML / AI також використовуються інтелектуальні пошукові алгоритми, найчастіше – PSO. У роботі [13] авторами було запропоновано удосконалений алгоритм негативного відбору з PSO для класифікації e-mail спаму. Тут випадковий детектор алгоритму негативного відбору було покращено за допомогою PSO. Загальна функція придатності запропонованого методу із алгоритмом негативного відбору та PSO оцінювалась за допомогою коефіцієнту локального відхилення (Local Outlier Factor – LOF), й загальна точність (Accuracy) досягла 83,20 %.

У роботі [14] авторами було використано концепцію на основі штучної нейронної мережі для класифікації та ідентифікації e-mail спаму із створеного самостійно набору даних. У запропонованому підході кластеризація k-середніх використовувалась для виділення характеристик, нейронна мережа зворотного розповсюдження використовувалась для навчання набору даних, а нейронна мережа прямого розповсюдження – для класифікації та ідентифікації e-mail спаму. Результати з кластеризацією k-середніх у попередній обробці кращі, ніж без її використання.

У роботі [15] авторами було об’єднано PSO із алгоритмом дерева рішень для покращення класифікації e-mail спаму. У зазначеному методі PSO оцінювало кінцеві результати будь-якого конкретного набору даних, а дерево рішень використовувалось для класифікації повного набору даних на невеликі класи ознак. Головним недоліком підходу, наведеного у роботі [15] є те, що автори не використовували жодної техніки виділення характеристик.

Висновки. Отже, з наведених робіт можна зробити висновок, що перспективним є поєднання техніки інтелектуальної оптимізації (пошукових, агентських алгоритмів, наприклад, PSO) для покращення точності, простору пошуку та процесу класифікації, а також методів ML / AI безпосередньо для детектування спаму електронної пошти.

Подальші дослідження мають бути спрямовані на дослідження інших класифікаторів на основі ML / AI, насамперед нейронних мереж (перцептронів, нейронних мереж Хопфілда і Хемінга, нейронних мереж прямого розповсюдження тощо) та нечіткої логіки (нейро-нечітких мереж, нечіткої кластеризації), які у поєднанні з методом оптимізації дозволяють детектувати спам електронної пошти. При цьому, необхідно також зосередитись на налаштуванні параметрів обраних інтелектуальних класифікаторів.

Також подальші дослідження мають бути спрямовані на дослідження інших методів оптимізації (еволюційних, агентських тощо), які у поєднанні з класифікатором на основі нейронних мереж та / або нечіткої логіки дозволяють виявляти e-mail спам.

ПЕРЕЛІК ПОСИЛАНЬ

1. Даник Ю.Г. Основи кібербезпеки та кібероборони: підручник / Ю.Г. Даник, П.П. Воробієнко, В.М. Чернега. – [Видання друге, перероб. та доп.]. – Одеса.: ОНАЗ ім. О.С. Попова, 2019. − 320 с.
2. Інформаційна та кібербезпека: соціотехнічний аспект: підручник / [В.Л. Бурячок, В.Б. Толубко, В.О. Хорошко, С.В. Толюпа]; за заг. ред. д-ра техн. наук, професора В.Б. Толубка. – К.: ДУТ, 2015. – 288 с.
3. Даник Ю.Г. Основи кібернетичної безпеки: монографія; за заг. ред. Ю.Г. Даника. / Ю.Г. Даник, Р.В. Грищук. – Житомир: ЖНАЕУ. – 2016. – 636 с.
4. Залива В.В. Фільтрація спаму електронної пошти за допомогою машинного навчання / В.В. Залива, А.П. Бондарчук, О.А. Золотухіна // Зв'язок. ДУТ. – 2019. − № 6. – С. 61-66.
5. Spam Statistics (2024): New Data on Junk Email, AI Scams & Phishing. [Електронний ресурс]. – Режим доступу: https://www.emailtooltester.com/en/blog/spam-statistics/.
6. Корнієнко В.І. Інтелектуальне моделювання нелінійних динамічних процесів в керуванні, кібербезпеці, телекомунікаціях: підручник / В.І. Корнієнко, О.Ю. Гусєв, О.В. Герасіна. – Міністерство освіти і науки України, Національний технічний університет «Дніпровська політехніка». – Дніпро, НТУ «ДП», 2020. – 531 с.
7. Text and image based spam email classification using KNN, Naïve Bayes and Reverse DBSCAN algorithm / Harisinghaney Anirudh, Aman Dixit, Saurabh Gupta, Anuja Arora // International Conference on Optimization, Reliabilty, and Information Technology (ICROIT), IEEE, 2014. – P. 153-155.
8. Masurah M. An evaluation on the efficiency of hybrid feature selection in spam email classification / M. Masurah, A. Selamat // International Conference on Computer, Communications, and Control Technology (I4CT), IEEE, 2015. – P. 227-231.
9. Renuka Karthika D. Latent Semantic Indexing Based SVM Model for Email Spam Classification / D. Renuka Karthika, P. Visalakshi // Journal of Scientific & Industrial Research. – 2014. – Vol. 73(7). – P. 437-442.
10. A support vector machine based naive Bayes algorithm for spam filtering / F. Weimiao, J. Sun, L. Zhang, C. Cao, Q. Yang // 35th International Performance Computing and Communications Conference (IPCCC), 2016 IEEE. – P. 1-8.
11. Kumaresan T. E-mail spam classification using S-cuckoo search and support vector machine / T. Kumaresan, C. Palanisamy // International Journal of Bio-Inspired Computation. – 2017. – Vol. 9, no. 3. – P. 142-156.
12. Olatunji Sunday Olusanya. Extreme Learning machines and Support Vector Machines models for email spam detection / Olatunji Sunday Olusanya // 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 2017 IEEE. – P. 1-6.
13. A combined negative selection algorithm–particle swarm optimization for an email spam detection system / I. Ismaila, A. Selamat, N.Th. Nguyen, S. Omatu, O. Krejcar, K. Kuca, M. Penhaker // Engineering Applications of Artificial Intelligence. – 2015. – Vol. 39. – P. 33-44.
14. Kaur T.S. Email Spam filtering using BPNN classification algorithm / T.S. Kaur, N. Bogiri // International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), IEEE, 2016. – P. 915-919.
15. Harpreet K. Improved email spam classification method using integrated particle swarm optimization and decision tree / K. Harpreet, A. Sharma // 2nd International Conference on Next Generation Computing Technologies (NGCT), IEEE, 2016. – P. 516-521.

ВІДОМОСТІ ПРО АВТОРІВ

Алексєєв Михайло Олександрович – д.т.н., професор, завідувач кафедри програмного забезпечення комп'ютерних систем, НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Алексєєв Олексій Михайлович – к.т.н., доцент, доцент кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Алексєєв Олексій Олексійович – магістр групи 122м-23-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Анісімов Володимир Володимирович – к.т.н., доцент, ННІ «Український державний хіміко-технологічний університет» (УДУНТ), м. Дніпро, Україна.

Балян Андрій Давідович – магістр групи 121м-23-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Берднік Михайло Геннадійович – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Білан Наталія Іванівна – к.ф.н, доцент, доцент кафедри іноземних мов НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Бойчук Микола Олексійович – магістр кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Бородкін Владислав Григорович – магістр кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Бондаренко Андрій А.Олексійович – студент групи 121-23-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Буслов Дмитро Юріович – магістр групи 121м-22-3 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Вереда Данило Вікторович – QA Engineer ФОП Вереда

Веселов Вячеслав Вячеславович – магістр групи 122м-23-2 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Герасіна Олександра Володимирівна – к.т.н., доцент, доцент кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Генчук Олексій Станіславович – магістр групи 122м-23-3 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Голінько Олександр Васильович – аспірант кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Грось Сергій Михайлович – магістр кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Гуліна Ірина Григорівна – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Дмитрієва Ольга Анатоліївна – д.т.н., професор, Дослідницький центр моделюючих технологій (SimTech) університету Штутгарта, Інститут моделювання водно-екологічних систем університету Штутгарта, Німеччина. (Olga Dmytriyeva, d. of sc., prof., [1]Research center of simulation technologies (SimTech), University of Stuttgart, [2]Institute for modelling hydraulic and environmental systems, University of Stuttgart)

Загинайло Євгеній Олександрович – аспірант кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Іванченко Олег Васильович – д.т.н., доцент, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Ібрагімов Ельдар Джаліл Огли – магістр групи 125м-24-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Іщук Павло Олександрович – аспірант кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кабак Леонід Віталійович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Каштан Віта Юріївна – к.т.н., доцент, доцент кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кійко Ілля Сергійович – студент групи 121-23ск-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Клевченко Дар'я Ігорівна – студентка кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Клименко Антон Володимирович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кожевников Антон Вечеславович – к.т.н., доцент, доцент кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Козорог Олександр Олександрович – магістр групи 125м-21-3 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кононець Владислав М. – магістр групи 125м-24-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Коробко Ольга Валеріївна – асистент кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Корнієнко Валерій Іванович – д.т.н., професор, завідувач кафедри безпеки інформації та телекомунікацій, НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Коряшкіна Лариса Сергіївна – д.т.н., професор, професор кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кравченко Дмитро Володимирович – магістр групи 125м-24-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Кручінін Олександр Володимирович – старший викладач кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Куваєв Микола Володимирович – к.т.н., науковий співробітник Інституту транспортних систем і технологій НАН України, м. Дніпро, Україна.

Курілех Дмитро Віталійович – студент групи 122-24ск-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Курдюкова Анастасія Романівна – магістр групи 122м-23-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Логвіненко Валентина Вікторівна – магістр групи 121м-23з-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мамешин Володимир Валерійовия – магістр групи 122м-23-2 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мартиненко Андрій Анатолійович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Масло Дмитро Григорович – магістр групи 125м-23-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мацюк Сергій Михайлович – к.т.н., доцент, доцент кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мауріна Катерина Віталіївна – студентка групи 122-22-4 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мельник Маргарита Олександрівна – к.т.н., доцент, доцент Приватна установа «Університет науки, підприємництва та технологій» SET University, Київ, Україна.

Мещеряков Леонид Іванович – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Михайленко Марія Олександрівна – магістр групи 122м-23-2 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мороз Борис Іванович – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мороз Дмитро Максимович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Мущак Артем Олександрович – магістр групи 125м-24-2 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

М'якенький Арсеній Вячеславович – аспірант кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Нановський Богдан Сергійович – аспірант кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Нікулін Сергій Леонідович – д.г.н., професор, професор кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Одновол Микола Миколайович – доцент кафедри системного аналізу і управління НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Пімахов Михайло Васильович – бакалавр кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Плющов Михайло Дмитрович – студент групи 122-24ск-1 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Приходченко Сергій Дмитрович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Політов Артем Ігоревич – магістр групи КС-23м кафедри електронних обчислювальних машин Дніпровського національного університету ім. О.Гончара, м. Дніпро, Україна.

Помазан Олексій Вітальович – аспірант, Український державний університет науки і технологій, Дніпро, Україна

Родна Катерина Станиславовна – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Рулікова Валерія Віталіївна – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Савченко Данило Олександрович – магістр групи 125м-24-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Сафаров Олександр Олександрович – к.т.н., доцент, доцент кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Семенов Сергій Геннадійович – д.т.н., професор, Університет Комісії Національної освіти в Кракові, Польща (Sergey Semenjv, d. of sc., prof., University Education Commission , Krakow, Poland)

Сень Данило Дмитрович – магістр групи 121м-22-3 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Сироткина Олена Ігорівна – к.т.н, доцент, School of Computer Science University of Windsor, Windsor, Ontario, Kanada.

Сирота Олена Петрівна – к.т.н., доцент, доцент Приватна установа «Університет науки, підприємництва та технологій» SET University, Київ, Україна.

Сідаш Катерина Андріївна – магістр групи 126м-23з-1 кафедри інформаційних технологій та комп'ютерної інженерії НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Сітнікова Оксана Олександрівна – к.т.н., доцент, доцент Приватна установа «Університет науки, підприємництва та технологій» SET University, Київ, Україна.

Сконечный Ян – профессор, Вроцлавский технологический университет, Польша.

Слободнюк Р.Є. – к.т.н., викладач, Дніпровський технолого-економічний фаховий коледж, Дніпро, Україна

Спірінцев В'ячеслав Васильович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Стародубський Ігор Петрович – аспірант кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Сушкін Вадим Максимович – к.т.н., доцент, ННІ «Український державний хіміко-технологічний університет» (УДУНТ), м. Дніпро, Україна.

Тимофєєв Дмитро Сергійович – старший викладач кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Ткач Максим Олександрович – к.т.н., доцент, доцент кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Ткачов Олександр Сергійович – магістр групи 125м-24-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Удовик Ірина Михайлівна – к.т.н., декан факультету інформаційних технологій, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Удовик Олександр Васильєвич – студент третього (освітньо-наукового) рівня вищої освіти за спеціалістю 141 «Електроенергетика, електротехніка та

електромеханіка» кафедри електротехніка та електромеханіка НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Уланова Наталья Петрівна – к.т.н., доцент, доцент кафедри вищої математики НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Харь Алена Тарасовна – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Харченко Микита Васильович – магістр групи 125м-22-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Хричов Сергій Олександрович – викладач Дніпровський технолого-економічний фаховий коледж, Дніпро, Україна.

Чертков Ілля Олександрович – магістр групи 125м-24-2 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Чернявський Ігор Сергійович – магістр групи 121м-23-2 кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Чурсина Марія Кирилівна – магістр групи 125м-24-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Швачич Геннадій Григорович – д.т.н., професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Шевцова Ольга Сергіївна – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Щербина Павло Олександрович – асистент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Ширін Артем Леонідович – к.т.н., доцент, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Шкуренко Дмитро Миколайович – магістр групи 125м-23-1 кафедри безпеки інформації та телекомунікацій НТУ «Дніпровська політехніка», м. Дніпро, Україна.

Яшкін Ростислав Ігорович – аспірант кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна.

ЗМІСТ

Розділ 1 МІЖНАРОДНЕ СПІВРОБІТНИЦТВО У СФЕРІ ОСВІТИ, НАУКИ ВИРОБНИЦТВА……..…….. 3

1. **Швачич Г.Г., Іщук П.О., Щербина П.О., Дмитрієва О.А.** Моделі паралельних обчислень на чисельно-аналітичних схемах підвищеного порядку точності…….…… 3
2. **Коряшкіна Л.С., Приходченко С.Д., Загинайло Є.О. , Сироткина О.І.** Використання фреймворку Pyroomacoustics для програмного моделювання звукових явищ в замкненому просторі…….. 7
3. **Мещеряков Л.І., Удовик І.М., Пімахов М.В., Сконечний Я.** Візуалізація віртуальних середовищ…..... 13
4. **Швачич Г.Г., Іщук П.О., Семенов С.Г.** Розподілене моделювання задач ідентифікації довкілля……..……..... 20
5. **Мещеряков Л.І., Одновол М.М., Сень Д.Д., Дмитрієва О.А.** Генеративні методи у 3D-моделюванні…… 22

Розділ 2 ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ…… 30

6. **Кабак Л.В., Мороз Б.І., Родна К.С., Гречкін М.О.** Порівняльний аналіз мікросервісної та монолітної архітектур……..... 30
7. **Чурсина М.К.** Застосування алгоритмів хешування для забезпечення цілісності даних в SQL Server……..……..…….. 35
8. **Спірінцев В.В., Ширін А.Л., Курілех Д.В.** Значення та вплив шрифтів у веброзробці…….…… 41
9. **Клименко А.В., Анісімов В.В., Рулікова В.В.** Реалізація напрямлення робота вздовж кольорової лінії за пропорційним алгоритмом на базі Romeo BLE……... 44
10. **Ширін А.Л., Слободнюк Р.Є., Хричов С.О.** Інформаційні технології як ключ до інноваційної steam-освіти за спеціальністю 181 Харчові технології…… 48
11. **Сафаров О.О., Масло Д.Г.** Розробка та дослідження засобів моніторингу та фільтрації файлової системи ОС Windows……..… 51
12. **Клевченко Д.І., Харь А.Т., Кабак Л.В.** Моди в ігровій індустрії: технічні виклики та можливості……..……..…….…. 55
13. **Спірінцев В.В., Іванченко О.В., Голінько О.В., Чернявський І.С.** Дослідження ефективності автоматичної перевірки заявок з використанням машинного навчання у розробці telegram-чатботу волонтерської підтримки……..……..…… 65

Розділ 3 ПРОГРАМНІ ЗАСОБИ УПРАВЛІННЯ, ЗБОРУ, ОБРОБКИ І ПЕРЕДАЧІ ІНФОРМАЦІЇ……. 70

14. **Ширін А.Л., Удовик О.В., Мацюк С.М., Балян А.Д.** Бібліотека React.js як інструмент оптимізації вебзастосунків…….. 70

15. **Спірінцев В.В., Генчук О.С.** Дослідження ефективності проксі-сервера на Node.js……..……...… 73
16. **Мещеряков Л.І., Харь А.Т., Куваєв М.В., Політов А.І.** Організація людино-машинного інтерфейсу для налагодження і супроводження на об'єкті програмного забезпечення систем керування технологічним процесом……..... 78
17. **Спірінцев В.В., Ширін А.Л., Плющов М.Д.** Важливість кольорів у вебдизайні. вебресурс Realtime Colors……... 85
18. **Вереда Д.В., Анісімов В.В., Сушкін В.М., Клименко А.В.** Порівняльне оцінювання якості відповідей чат-ботів ChatGPT і Claude на основі студентських оцінок……. 88
19. **Спірінцев В.В., Ширін А.Л., Бондаренко А.А.** Розробка вебпроєктів, адаптованих до різних пристроїв у фреймворку Bootsrap……..……... 93
20. **Кабак Л.В., Алексєєв О.О.** Створення мобільного застосунку для керування фінансами з використанням штучного інтелекту…….. 96
21. **Спірінцев В.В., Ширін А.Л., Курдюкова А.Р.** Оптимізація створення макетів у Figma. ефективні інструменти та плагіни для web-дизайну…….……..… 99
22. **Ткач М.О., Козорог О.О.** Розробка GNSS RTK системи для сільськогосподарської промисловості……. 102
23. **Спірінцев В.В., Веселов В.В.** Дослідження ефективності використання Jetpack Compose в порівнянні з мовою XML при розробці мобільних додатків для ОС Android……..…….. 105
24. **Яшкін Р.І., Бердник М.Г.** Застосування методів fine-tuning та трансферного навчання для діагностики хвороби паркінсона за мовленням…….. 112
25. **Спірінцев В.В., Михайленко М.О.** Виявлення кросбраузерних несумісностей з використанням фреймворку візуального тестування. 114
26. **Ширін А.Л., Кійко І.С.** Найбільша зростаюча підпослідовність….. 121
27. **Ширін А.Л., Кійко І.С.** Перестановки двох масивів…….. 125

Розділ 4 МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОПТИМІЗАЦІЯ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ ДЛЯ ВИРІШЕННЯ ЗАВДАНЬ ОСВІТИ, НАУКИ І УПРАВЛІННЯ ВИРОБНИЦТВОМ..……. 132

28. **Алексєєв О.М., М'якенький А.В.** Двонапрямна LSMT модель усунення неоднозначності слів у тексті при моделюванні когнітивного процесу розуміння……. 132
29. **Мещеряков Л.І., Уланова Н.П., Кожевніков А.В., Нановський Б.С.** Складова моделі удосконалення інформаційного забезпечення комп'ютерних систем з застосуванням моментних функцій..... 135

30. **Сітнікова О.О., Мельник М.О., Сирота О.П.** Оптимізація планування у гетерогенних розподілених обчислювальних системах за допомогою метаевристичних алгоритмів……. 142
31. **Ширін А.Л., Мартиненко А.А., Шевцова О.С., Буслов Д.Ю.** Дослідження математичних можливостей обробки природної мови та різниці у здатностях до логічного мислення у рішенні простих задач…… 146
32. **Помазан О.В., Анісімов В.В., Клименко А.В.** Оптимальне рішення розкладки панелей на даху за допомогою генетичного алгоритму……. 150
33. **Бердник М.Г., Стародубський І.П.** Автоматична адаптація програмного забезпечення до хмарних та периферійних платформ методами машинного навчання……..……... 155
34. **Нікулін С.Л., Каштан В.Ю., Коробко О.В. , Сідаш К.А**. Комплексне використання методів K-means кластеризації та ARIMA для моделювання тенденцій смертності в Європі від інфекційних захворювань……. 157

Розділ 5 ІНФОРМАЦІЙНА БЕЗПЕКА…… 161

35. **Корнієнко В.І., Савченко Д.О.** Розробка методики виявлення загроз у реальному часі на платформі Kubernetes за допомогою системи Falco……... 161
36. **Грось С.М., Тимофєєв Д.С.** Аналіз вразливостей протоколу BGP.. 164
37. **Шкуренко Д.М., Кручінін О.В.** Особливості тестування на проникнення для інформаційно-комунікаційних систем малого бізнесу……... 170
38. **Бойчук М.О.** Особливості забезпечення захищеності кваліфікованих електронних підписів у судових установах України…….……... 173
39. **Бородкін В.Г.** Програмне рішення для захисту вебресурсів від SQL-загроз за допомогою Python……. 175
40. **Кравченко Д.В.** Децентралізовані ідентифікатори (DID)……... 178
41. **Кононець В.М.** Методи теорії ігор у кібербезпеці: від проактивних стратегій до зниження збитків…….……..... 184
42. **Харченко М.В.** Використання блокчейн-технологій для захисту медичних даних у цифрових системах охорони здоров’я…..…… 188
43. **Мущак А.О.** Кібербезпека у фінансових і платіжних системах…….. 192
44. **Ткачов О.С.** Кібербезпека як ключова компетенція для цифрової економіки……..… 196
45. **Ібрагімов Е.Д.** Використання штучного інтелекту у захисті інформації……. 200

Розділ 6 ПРОБЛЕМИ ДИСТАНЦІЙНОЇ ОСВІТИ……..…….. 207

46. **Кабак Л.В., Гуліна І.Г., Мороз Д.М., Мамешин В.В.** Розробка системи для розподіленого навантажувального тестування та аналізу результатів у реальному часі на базі Kubernetes…… 207

47. **Алексєєв О.М., Логвіненко В.В.** Розробка та впровадження програмної системи моніторингу успішності студентів курсу Google Classroom засобами Google таблиць та Google Apps Script…….. 210
48. **Bilan N.I., Maurina K.V.** Enhancing english specialist text reading comprehension competence by combining content visualization methods……..… 216
49. **Герасіна О.В., Чертков І.О.** Виявлення спаму електронної пошти із використанням машинного навчання та штучного інтелекту……..… 219
ВІДОМОСТІ ПРО АВТОРІВ……..……. 224

Наукове видання

XIX Міжнародна конференція

ПРОБЛЕМИ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СФЕРІ ОСВІТИ, НАУКИ ТА ПРОМИСЛОВОСТІ

Збірник наукових праць
Національний технічний університет «Дніпровська політехніка»,
№ 9
(Українською мовою)

Відповідальний за випуск д.т.н., проф. Л.І. Мещеряков

Видано в редакції авторів публікацій.

Підп. до друку 25.12.2023. Формат 30×42/4.
Папір офсет. Ризографія. Ум. друк. арк. 10,7.
Обл.-вид. арк. 13,6. Тираж 100 пр. Заказ № 847

Підготовлено до друку та видруковано
НТУ «Дніпровська політехніка»,
Свідоцтво про внесення до Державного реєстру ДК №1842 від 11.06.2004 р.