МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

АВТОМАТИЗОВАНІ СИСТЕМИ УПРАВЛІННЯ ТА ПРИЛАДИ АВТОМАТИКИ

Всеукраїнський міжвідомчий
науково-технічний збірник

Заснований у 1965 р.

Випуск 185

Харків
2025

УДК 62-50

У збірнику наведено результати досліджень, що стосуються моделювання інформаційних технологій обробки зображень, розробки моделей, методів та інформаційних технологій формування пояснень в інтелектуальних системах, моделювання процесу кросплатформної міграції програмного забезпечення інформаційної системи та вибору технічних засобів як можливих джерел ентропії для генерації випадкових чисел в мобільних системах. Запропоновано нові та вдосконалені моделі, методи та інформаційні технології в галузі управління обробкою зображень, процесами міграції ІТ-продуктів, створення інтелектуальних та мобільних систем.

Для викладачів університетів, науковців, фахівців, аспірантів.

The collection presents the results of research related to the modeling of information technologies for image processing, the development of models, methods and information technologies for the formation of explanations in intelligent systems, the modeling of the process of cross-platform migration of information system software and the selection of technical means as possible sources of entropy for the generation of random numbers in mobile systems. New and improved models, methods and information technologies in the field of image processing management, IT product migration processes, and the creation of intelligent and mobile systems are proposed.

For university teachers, scientists, specialists, and graduate students.

Редакційна колегія:

В.В. Семенець, д-р техн. наук, проф. (гол. ред.), *В.М. Левикін,* д-р техн. наук, проф. (відпов. ред.), *М.В. Євланов,* д-р техн. наук, проф. (відпов. секр.), *Є.В. Бодянський,* д-р техн. наук, проф., *І.В. Гребеннік,* д-р техн. наук, проф., *А.Л. Єрохін,* д-р техн. наук, проф., *А.О. Каргін,* д-р техн. наук, проф., *Б.І. Мороз,* д-р техн. наук, проф., *І.Ш. Невлюдов,* д-р техн. наук, проф., *К.Е. Петров,* д-р техн. наук, проф., *І.В. Рубан,* д-р техн. наук, проф., *С.Г. Удовенко,* д-р техн. наук, проф., *О.Є. Федорович,* д-р техн. наук, проф., *В.О. Філатов,* д-р техн. наук, проф., *Г.З. Халімов,* д-р техн. наук, проф.

Рішення Національної ради про реєстрацію № 1410 від 25.04.2024 р.
Ідентифікатор медіа R30-03874

Адреса редакційної колегії: Україна, 61166, Харків, просп. Науки, 14, Харківський національний університет радіоелектроніки, кімн. 254, тел. (057) 70-21-451

© Харківський національний університет радіоелектроніки, 2025

ЗМІСТ

КРУГЛИК А.С., ЛЕВИКІН В.М., ЄВЛАНОВ М.В., МОРОЗ Б.І., МОРОЗ Д.М.
МОДИФІКАЦІЯ МОДЕЛІ ОДНОМІСНОГО КОВАРІАНТНОГО ФУНКТОРА
ДЛЯ ПРОЦЕСУ КРОСПЛАТФОРМЕНОЇ МІГРАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ 5

СИТНІКОВА П.Е., ГРИЦАЙ М.О.
МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ МЕТОДІВ ДЕТЕКЦІЇ ПОГЛЯДУ КОРИСТУВАЧА
У СИСТЕМАХ ЛЮДИНО-КОМП'ЮТЕРНОЇ ВЗАЄМОДІЇ …….....23

ЧАЛИЙ С.Ф., ЛЕЩИНСЬКА І.О.
ПОБУДОВА ФУНКЦІОНАЛЬНО-ТЕМПОРАЛЬНОГО ПРЕДСТАВЛЕННЯ МЕНТАЛЬНОЇ
МОДЕЛІ РІШЕННЯ В ЗАДАЧІ ФОРМУВАННЯ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНИХ
СИСТЕМАХ …….....34

KOVTUNENKO A.R., MASHTALIR S.V.
A REVIEW OF MODERN NEURAL NETWORK ARCHITECTURES
FOR IMAGE SEGMENTATION …….....43

ЧАЛИЙ С.Ф., ЛЕЩИНСЬКИЙ В.О.
ПРИЧИННО-НАСЛІДКОВЕ МОДЕЛЮВАННЯ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНИХ
СИСТЕМАХ …….....53

ОСТАПЕЦЬ Д.О., ОПРЯТНИЙ А.О.
АНАЛІЗ ДЖЕРЕЛ ЕНТРОПІЇ ДЛЯ ГЕНЕРАЦІЇ ВИПАДКОВИХ ЧИСЕЛ НА БАЗІ
МОБІЛЬНИХ ПРИСТРОЇВ …….....63

КРАСНІКОВ В.В., СИТНІКОВА П.Е.
АЛГОРИТМ ЗАВОЮВАННЯ ДЛЯ СТОХАСТИЧНОГО ЗАПОВНЕННЯ
ДВОВИМІРНИХ ДИСКРЕТНИХ РЕШІТОК ЗВ'ЯЗАНИМИ ОБЛАСТЯМИ 70

ЧАЛИЙ С.Ф., КРАВЧЕНКО Р.В.
ГРАФОВА НЕЙРОННА МЕРЕЖА ДЛЯ ТЕМПОРАЛЬНО УПОРЯДКОВАНИХ
ДАНИХ В ЗАДАЧІ ПОБУДОВИ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНІЙ СИСТЕМІ 77

РЕФЕРАТИ …..... 86

CONTENT

KRUGLIK A., LEVYKIN V., YEVLANOV M., MOROZ B., MOROZ D.
MODIFICATION OF THE SINGLE-PLACE COVARIANT FUNCTOR MODEL
FOR THE PROCESS OF INFORMATION SYSTEM CROSS-PLATFORM MIGRATION 5

SYTNIKOVA P., HRYTSAI M.
MODELING AND RESEARCH OF USER GAZE DETECTION METHODS IN
HUMAN-COMPUTER INTERACTION SYSTEMS ..…….…….....23

CHALY S., LESHCHYNSKA I.
DEVELOPMENT OF FUNCTIONAL-TEMPORAL REPRESENTATION OF MENTAL
MODEL SOLUTIONS IN THE TASK OF EXPLANATION GENERATION IN
INTELLIGENT SYSTEMS ……..34

KOVTUNENKO A.R., MASHTALIR S.V.
A REVIEW OF MODERN NEURAL NETWORK ARCHITECTURES
FOR IMAGE SEGMENTATION …….....…….....43

CHALY S., LESHCHYNSKYI V.
CAUSAL MODELING OF EXPLANATIONS IN INTELLIGENT SYSTEMS ...….....53

OSTAPETS D., OPRIATNYI A.
ANALYSIS OF ENTROPY SOURCES FOR RANDOM NUMBER GENERATION
BASED ON MOBILE DEVICES …..……63

KRASNIKOV V., SYTNIKOVA P.
CONQUEST ALGORITHM FOR STOCHASTIC FILLING OF TWO-DIMENTIONAL
DISCRETE GRIDS WITH CONNECTED REGIONS ..…….....70

CHALY S., KRAVCHENKO R.
GRAPH NEURAL NETWORK FOR TEMPORALLY ORDERED DATA IN THE TASK
OF EXPLANATION GENERATION IN AN INTELLIGENT SYSTEM ……77

ABSTRACTS.....86

УДК 004.4`6 DOI: 10.30837/0135-1710.2025.185.005

А.С. КРУГЛИК, В.М. ЛЕВИКІН, М.В. ЄВЛАНОВ, Б.І. МОРОЗ, Д.М. МОРОЗ

МОДИФІКАЦІЯ МОДЕЛІ ОДНОМІСНОГО КОВАРІАНТНОГО ФУНКТОРА ДЛЯ ПРОЦЕСУ КРОСПЛАТФОРМНОЇ МІГРАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Розглянуто основні особливості та недоліки існуючих рішень з формального опису процесу кросплатформної міграції програмного забезпечення інформаційних систем. Для формального опису цього процесу запропоновано модифікувати модель одномісного коваріантного функтора. Розроблено два основних варіанти модифікації цієї моделі з врахуванням можливості виконання процесу кросплатформної міграції як під час експлуатації ІС, так і під час валідації ІС до початку її експлуатації. Проведено експериментальну перевірку отриманих результатів.

1. Вступ

Переведення бізнесів у цифрову площину ставить виклики до ІТ-галузі. Зокрема, постійно зростає навантаження на ІТ-інфраструктуру підприємств та обсяги оброблюваної інформації. Залежно від різновиду бізнесу, його стратегії та інших факторів, перед ІТ-сектором ставляться завдання знаходити дешевші рішення, або ж більш захищенні, або надійніші рішення. Виходячи із сукупності цих факторів, ІТ-компанії можуть приймати рішення стосовно використання тієї чи іншої платформи для роботи інформаційних систем (ІС), які автоматизують бізнес-логіку підприємства, або навіть стосовно підтримки кількох платформ одночасно.

В основі рішень фахівців ІТ-компаній щодо кросплатформної міграції або кросплатформної підтримки ІС, як правило, знаходяться дві протилежні за сенсом концепції.

Перша концепція: розробити рішення, яке буде унікальним і апріорно орієнтованим на вирішення конкретної проблеми, яка виникає для конкретної ІС конкретного підприємства-замовника тільки один раз за весь життєвий цикл цієї ІС.

Друга концепція: розробити рішення, яке буде типовим (шаблонним) і апріорно орієнтованим на вирішення множини східних між собою проблем, які виникають багато разів для тієї самої ІС та її варіантів конфігурації.

Слід зазначити, що значну увагу в галузі міграції сучасні дослідження приділяють саме розробці рішень на основі другої концепції. Прикладами таких рішень є фреймворки хмарної міграції, в яких основну увагу приділено структуруванню та визначенню окремих етапів і їх організації в єдиний життєвий цикл міграції [1], [2]. Але такі рішення вимагають формальних описів ІС та її елементів. Дослідження, проведені у 2010-2018 рр., визнали як найрозповсюдженіші такі засоби формального опису ІС:

– структуровані текстові документи, які описують вихідний програмний код, додаткову контекстну інформацію (трасування виконання та початкову сутність вихідного коду, перевірену на наявність змін) та запити на зміни [3];

– профілі прототипів програмного забезпечення (ПЗ), які використовувалися для виявлення і документування функціональних вимог до створюваного ІТ-продукту [4];

– спеціальні структурні моделі [5].

Аналізуючи досвід цих досліджень та власні доробки у даному напрямі, автори [6], [7], визнали, що основним артефактом, який відображає проєктні рішення та вплив проєктних рішень на технічні артефакти та організаційні обов’язки, є архітектура ПЗ ІС. Але саме обмеження артефактами процесу розробки ПЗ автори [6] визнали одним з

головних недоліків застосування формальних описів для підтримки і супроводження ІС. Для подолання цього недоліку у [7] було представлено інструментальний підхід для оцінки поширення змін, спричинених запитом на зміну в бізнес-процесах або програмних системах, на основі архітектури ПЗ ІС та дизайну процесу. Але застосування запропонованого у [7] підходу та аналогічних йому для супроводження та управління змінами ІС під час експлуатації і, зокрема, під час кросплатформної міграції ПЗ ІС можливе тільки за дотримання таких умов:

– наявність множини адекватних моделей, які повністю описували б бізнес-процеси підприємства, ІС та її ПЗ;

– необхідність зберігання структурних та поведінкових особливостей бізнес-процесів підприємства, ІС та її ПЗ під час трансформації моделей одна в іншу, а також моделей у відповідні фрагменти програмного коду або команди.

Дотримання цих умов вимагає існування формального механізму, який би забезпечував існування узгоджених між собою моделей бізнес-процесів підприємства, ІС та її ПЗ, а також підтримував би цілісність структурних та поведінкових особливостей цих моделей під час їх взаємної трансформації. Необхідність існування і застосування такого формального механізму непрямо підтверджується дослідженням можливості використання засобів штучного інтелекту для вирішення задачі класифікації та пріоритизації запитів на зміну у ІТ-проєкті [8]. Якщо для повних та адекватних моделей запитів на зміну та ІС застосування методу Random Forest дозволило передбачити можливу зміну пріоритетності запитів на зміну з точністю 94 % і визначити збільшення або зменшення пріоритетності цієї зміни з точністю 93 %, то в ситуаціях із спотвореними даними, на основі яких формувалися моделі запитів на зміну, точність моделі становила 68 %, що, за словами авторів [8], відповідає опублікованим результатам сучасних досліджень. Це означає, що спроби замінити цей формальний механізм методами і засобами штучного інтелекту не гарантують рішення кращого, ніж ручна обробка таких моделей кваліфікованим аналітиком. Саме тому проведення досліджень з розробки і вдосконалення такого формального механізму є актуальним з теоретичної та прикладної точок зору.

2. Аналіз літературних даних і постановка проблеми дослідження

Досвід, накопичений дослідниками та фахівцями ІТ-компаній, дозволяє серед напрямів розвитку засобів формального механізму управління підтримкою та супроводженням ПЗ особливо виділити множину семантичних методів. Серед цієї множини методів найчастіше виділяють [9]:

– методи структурної операційної семантики (описують поведінку програми у вигляді змін станів, спричинених виконанням елементарних кроків);

– методи денотаційної семантики (визначають зміни станів функціями).

Кожна з цих груп методів має власні переваги та недоліки, які ускладнюють їх прикладне застосування. Тому у [9] було запропоновано новий підхід до управління семантикою ПЗ: поведінка програм, тобто зміни станів, моделюється як категорії цих станів. При цьому категорія морфізмів виражає множину елементарних кроків виконання, а виконання конкретної програми є орієнтованим шляхом у категорії, тобто композицією морфізмів [9]. Для розвитку підходу було використано можливість зв'язування між собою категорних моделей системою функторів, які у запропонованому підході реалізовано у вигляді окремих процедур. Така реалізація дозволила формально описати повторний виклик процедур, вкладення викликів процедур та рекурсивні виклики. Хоча запропонований у [9] підхід достатньо простий і точний, він має суттєві обмеження, тому що його формальна база побудована виключно для простої процедурної мови, яка містить усі основні конструкції ван Дейкстри.

Використання математичного апарату теорії категорій як формальної основи механізму забезпечення існування та підтримки цілісності узгоджених між собою моделей бізнес-процесів підприємства, ІС та її ПЗ залишається одним з основних способів створення такого механізму. Дана точка зору обумовлена у [10] такими причинами:

– теорія категорій пропонує інтегроване бачення концепцій моделі, а також надає механізми для об'єднання моделей, механізми для міграції між моделями та механізми для побудови мостів між моделями;

– категорія, як і модель, вважається сумішшю графічної інформації та алгебраїчних операцій, тому мова категорій здається найзагальнішою для опису моделей;

– теорія категорій дозволяє спростити опис того, як людина мислить та використовує окремі моделі.

Але прикладне застосування теоретико-категорних моделей в ІТ-галузі продовжує викликати значні проблеми. Так, у [11] для створення моделі лінійки програмних продуктів з оновленнями було запропоновано розглянути системи умовних переходів в коалгебраїчному середовищі, елементами яких є еквівалентні категорії Клейслі, в яких знаходяться ці коалгебри. В цих системах функтори використано для формального опису типу розгалуження коалгебри програмного продукту. Одним з результатів використання таких категорно-коалгебраїчних систем для моделювання лінійки програмних продуктів з оновленнями є можливість створення екземпляру існуючого алгоритму мінімізації коалгебри для виведення поведінкових еквівалентностей у цьому середовищі [11]. Такий екземпляр алгоритму дозволяє забезпечити поведінкову еквівалентність оновленої версії програмного продукту із мінімальними змінами коалгебри операцій оновленої версії. Але запропоновані у [11] рішення мають два значні недоліки, які обмежують їх прикладне застосування, а саме:

– необхідність наявності у розробників формального механізму забезпечення існування та підтримки цілісності узгоджених між собою моделей бізнес-процесів підприємства, ІС та її ПЗ значного обсягу теоретичних знань в галузі сучасної загальної алгебри (теорія категорій, теорія представлень, універсальна алгебра тощо);

– перед початком програмної реалізації запропонованих систем умовних переходів необхідно довести існування адекватної реалізації розробленої формальної бази таких систем засобами конкретної мови або середовища програмування.

Крім того, запропоновані у [11] системи категорно-коалгебраїчних моделей описують виключно артефакти процесу супроводження ПЗ, що, як показано у розділі 1, є одним з головних недоліків застосування формальних описів для підтримки і супроводження ІС.

Для подолання цих недоліків у [12] було запропоновано нову формульну рамку опису моделі архітектури ПЗ, в якій теорія категорій, алгебраїчні специфікації та алгебра процесів були поєднані разом:

– компоненти ПЗ були описані за допомогою об'єктів категорій;

– різні типи морфізмів зображували семантику, що міститься у зв'язках компонентів;

– функтори теорії категорій описували зв'язки між моделями архітектур, представленими типізованими діаграмами категорій.

Отримані у [12] теоретичні результати були орієнтовані на визначення того, чи відповідає перетворення певним характеристикам або обмеженням, та реалізовані у вигляді системи спільної розробки ПЗ. Встановлено, що такий підхід розширює можливості семантичного опису моделі архітектури та може бути використаний як ефективне доповнення до існуючого методу моделювання ПЗ [12]. Але подібне рішення не враховує зміни, які можуть виникати у ПЗ внаслідок змін бізнес-процесів та ІТ-інфраструктури, в межах якої експлуатується це ПЗ.

Розвитком результатів, отриманих у [12], є розроблена у [13] єдина основна структура семантичного опису, призначена для опису та перетворення компонентно-орієнтованих моделей ПЗ, а також для підтримки та перевірки семантичних властивостей у процесі перетворення моделі. Формальною основою цієї структури семантичного опису є розширення теорії типізованих категорій до алгебри процесів. Крім цього, отримані у [13] результати орієнтовані на опис не стільки ПЗ, скільки його моделей, а саме:

– діаграма категорій використовується для опису семантики архітектурної моделі ПЗ;

– типізований морфізм описує зв'язок залежності між компонентними об'єктами категорії;

– типізований функтор використовується для опису механізму відображення до та після перетворення моделі.

Прикладні дослідження показують, що отримані результати добре відповідають сутності та вимогам процесу модельно-орієнтованої розробки та забезпечують нову основну структуру для розуміння, когнітивного навчання та просування досліджень у галузі розробки ПЗ на основі модельно-орієнтованого підходу [13].

Новою у результатах, отриманих у [13], є їхня орієнтація на формальний опис не компонентів ПЗ (фрагментів програмного коду), а моделей цих компонентів, для опису яких застосовуються згадувані у розділі 1 засоби формального опису ІС. Але при цьому залишаються малодослідженими або зовсім не дослідженими дуже багато питань, пов’язаних із прикладними аспектами застосування запропонованої у [13] структури семантичного опису. Зокрема, залишається незрозумілим спосіб забезпечення еквівалентності категорного-функторного представлення моделей ПЗ, самих моделей та їх програмної реалізації з врахуванням обмежень середовищ розробки та експлуатації ПЗ. Знову залишається невирішеною проблема відображення поточних варіантів та можливих змін у бізнес-процесах підприємства у категорно-функторні формальні описи семантики ПЗ, яке експлуатується в межах цих бізнес-процесів.

Один з поглядів на загальну перспективу застосування категорно-функторного апарату для семантичного моделювання ПЗ розглянуто у [14]. Ця перспектива має такі особливості [14]:

– моделювання денотації в межах категорій, враховуючи вхідні/вихідні значення та динаміку станів компонентів ПЗ;

– зосередження уваги на з'ясуванні зв'язків між об'єктами, що представляють стани пам'яті як денотаційні функції;

– цілеспрямоване використання коалгебр для вираження поведінки програми, особливо щодо конфігурацій пам'яті;

– представлення динаміки виконання через налаштований поліноміальний ендофунктор, що враховує ряд типів переходів.

Очікуваним результатом використання цієї перспективи є отримання категорії, адаптованої для конфігурацій пам'яті, яка ефективно відображає переходи програми, тобто встановлення переходу між описом ПЗ та пам’яттю як елементом ІТ-інфраструктури, в межах якої планується експлуатація цього ПЗ [14]. Але дослідники розуміють, що навіть така перспектива хоча й підкреслює незмінну актуальність теорії категорій у динамічному ландшафті семантичного моделювання, пропонує розуміння лише потенційних практичних застосувань у програмуванні та суміжних галузях.

Ще одним напрямом сучасних досліджень способів формального опису процесів супроводження та міграції ПЗ є використання положень теорії категорій для опису окремих випадків та різновидів трансформації ПЗ. Цей напрям, по суті, є спробою формально узагальнити прикладний досвід розробки та супроводження ПЗ різного

призначення і розглядає категорно-функторний апарат як своєрідні формальні патерни, які встановлюють загальні особливості виконання тих чи інших операцій. Як приклади робіт за цим напрямом можна вказати такі дослідження:

– розробка теоретичної бази, яка забезпечує об'єднувальну мову для різних існуючих симпліціальних фільтрацій, а також є механізмом для генерації довільно великих сімейств нових функторів фільтрації з контролем залежності/незалежності від базової точки, а також локальності фільтрації [15];

– розробка так званих функторів Меркла як складних дерев з багаторазових будівельних блоків, що включають суми, добутки та функціональні простори та є замкнутими відносно композиції та найменших точок виправлення, та застосування цих функторів як важливого елементу формальної основи протоколу Canton та для перевірки цілісності й гарантій безпеки цього протоколу, а також тестування нових фільтрацій на наборах даних у блокчейнах та інших технологіях розподілених реєстрів [16];

– використання функторів C++ як механізмів адаптації загальних алгоритмів (інтегрування, інтерполяція, мінімізація та процедури пошуку коренів) до різних типів даних та/або типів функцій у програмному пакеті C++ для розрахунку паралельних розподілів імпульсів (включаючи механізм відриву та механізм дифракційної дисоціації) важкого залишку (ядра) в реакціях нокауту окремих нуклонів, викликаних пучками стабільних та радіоактивних атомних ядер середньої енергії [17];

– використання теорії категорій для моделювання та управління потоками даних та залежностями в системі, зокрема за допомогою функторів та монад, в процесах створення адаптивних та масштабованих архітектур ігрових додатків, враховуючи специфіку компонентно-орієнтованого та сервісно-орієнтованого архітектурних стилів в умовах зміни вимог та умов експлуатації [18];

– використання категорно-функторного апарату для формального опису патернів проєктування вимог до ІС, які визначають моделі основних процесів інженерії вимог та синтезу опису архітектури створюваної ІС [19].

Проведений аналіз сучасних досліджень дозволяє зробити такі висновки:

– проблема формального опису механізму управління підтримкою та супроводженням ПЗ ІС ще досить далека від повного вирішення;

– застосування апарату функторів спрямовано, переважно, на формальний опис елементів механізму управління підтримкою та супроводженням ПЗ ІС, які забезпечують зберігання структурних особливостей програмних систем та продуктів, а також їхніх компонентів, під час трансформацій;

– існуючі категорно-функторні моделі орієнтовано, головним чином, на опис ПЗ та його окремих компонентів, як перспектива – на опис переходу між описом ПЗ та пам’яттю як елементом ІТ-інфраструктури.

– проблеми формального опису трансформації ПЗ базової (типової) ІС до особливостей ІТ-інфраструктури конкретних підприємств, які є споживачами різноманітних варіантів конфігурації цієї ІС, а також впливу на архітектуру ПЗ запитів на зміни, які виникають у бізнес-процесах підприємств, залишаються майже недослідженими.

Виходячи з цих висновків, проблему даного дослідження запропоновано сформулювати як проблему модифікації категорно-функторних моделей з врахуванням особливостей опису трансформації ПЗ ІС в процесі кросплатформної міграції, яка виникає за вимогами адаптації типової ІС до особливостей ІТ-інфраструктури конкретних підприємств-споживачів цієї ІС. Вирішення цієї проблеми дозволить отримати результати, що будуть викликати інтерес як у науковців в галузі комп’ютерних наук, так і у фахівців-практиків з міграції ПЗ ІС.

3. Мета і задачі дослідження

Метою даного дослідження є модифікація моделі функтора для опису трансформаціїї програмного забезпечення ІС в межах її кросплатформної міграції. Досягнення цієї мети дозволить визначити формальний опис елементів процесу кросплатформної міграції програмного забезпечення ІС на рівні її семантичного опису, що дасть змогу зменшити витрати ресурсів та часу на виконання цього процесу на вимоги значної кількості підприємств-споживачів ІС.

Для досягнення цієї мети треба вирішити такі задачі:

– модифікація елементів моделі одномісного коваріантного функтора для опису особливостей кросплатформної міграції програмного забезпечення ІС;

– експериментальна перевірка модифікованої моделі функтора в процесі рефакторингу вихідного програмного коду мігруючого ПЗ ІС.

4. Моделі і технології, які використовуються у дослідженні

Об'єктом даного дослідження є процес кросплатформної міграції ПЗ ІС управління підприємством.

Основною гіпотезою даного дослідження є гіпотеза про можливість застосування моделі функтора, який зв'язує дві категорні моделі ПЗ ІС, для опису елементів процесу кросплатформної міграції цього ПЗ.

Ідея використання функторів як засобів формалізованого опису елементів механізму управління підтримкою та супроводженням ПЗ ІС, які забезпечують зберігання структурних особливостей категорної моделі ПЗ ІС під час її трансформації, поставила питання про можливість розробки єдиного підходу до формалізованого опису функторів. В даному дослідженні запропоновано в загальному випадку використовувати модель одномісного коваріантного функтора. Під час побудови цієї моделі було прийнято позначення похідної категорної моделі, яка є початком функтора, як категорії A, а кінцевої категорної моделі, яка є кінцем функтора, як категорії B. З врахуванням цих позначень узагальнена модель одномісного коваріантного функтора має такий вигляд [19]-[21]:

$$\Phi_B^A = (Sc_{Tez(A)}, Ob^A, Mor^A, Sc_{Tez(B)}, Ob^B, Mor^B, \Phi_{Ob^A}^{Ob^B}, \Phi_{Mor^A}^{Mor^B}), \quad (1)$$

де $Sc_{Tez(A)}$ – рівень представлення фрагмента тезаурусу, що використовується похідною категорією A; Ob^A – клас об'єктів похідної категорії A, які описують елементи класу об'єктів похідної категорної моделі; Mor^A – клас морфізмів похідної категорії A, які описують зв'язки елементів класу об'єктів похідної категорної моделі; $Sc_{Tez(B)}$ – рівень представлення фрагмента тезаурусу, що використовується кінцевою категорією B; Ob^B – клас об'єктів кінцевої категорії B, які описують елементи класу об'єктів кінцевої категорної моделі; Mor^B – клас морфізмів кінцевої категорії B, які описують зв'язки елементів класу об'єктів кінцевої категорної моделі; $\Phi_{Ob^A}^{Ob^B}$ – система правил трансформації елементів класу об'єктів похідної категорії A в елементи класу об'єктів кінцевої категорії B; $\Phi_{Mor^A}^{Mor^B}$ – система правил трансформації елементів класу морфізмів похідної категорії A в елементи класу морфізмів кінцевої категорії B.

Модель (1) може існувати тільки при дотриманні таких умов [22]:

$$\forall a \in Ob^A \quad \exists \Phi_B^A(a) \in Ob^B \; ; \quad (2)$$

$$\forall \alpha \in H_A(a_i, a_j) \subseteq Mor^A \ \exists \Phi_B^A(\alpha) \in H_B(\Phi_B^A(a_i), \Phi_B^A(a_j)) \in Mor^B \ ; \quad (3)$$

$$\forall 1_a \in Mor^A \ \exists \Phi_B^A(1_a) = 1_{\Phi_B^A(a)} \in Mor^B \ ; \quad (4)$$

$$\forall \alpha \in H_A(a_i, a_j) \in Mor^A, \beta \in H_A(a_j, a_k) \in Mor^A \ \exists \Phi_B^A(\alpha\beta) = \Phi_B^A(\alpha)\Phi_B^A(\beta) \in Mor^B, \quad (5)$$

де a – будь-який об'єкт, що належить до класу об'єктів Ob^A похідної категорії A; $H_A(a_i, a_j)$ – множина морфізмів, визначених в похідній категорії A для об'єкта a_i як початку морфізму та об'єкта a_j як кінця морфізму, при цьому $i \neq j$ є ідентифікаторами об'єктів похідної категорії A; α – морфізм, який є елементом множини $H_A(a_i, a_j)$; $H_B(\Phi_B^A(a_i), \Phi_B^A(a_j))$ – множина морфізмів, визначених у кінцевій категорії B для об'єкта $\Phi_B^A(a_i)$ як початку морфізму та об'єкта $\Phi_B^A(a_j)$ як кінця морфізму; 1_a – одиничний морфізм, визначений для об'єкта a похідної категорії A; $1_{\Phi_B^A(a)}$ – одиничний морфізм, визначений для відповідного об'єкта кінцевої категорії B; $\alpha\beta$ – часткова бінарна операція здобутку морфізмів похідної категорії A.

Крім умов (2)-(5), обумовлених особливостями формального апарату теорії категорій, на модель (1) накладено додаткові умови існування, які мають вигляд:

$$Ob^A \subseteq \Phi_{Ob^B}^{Ob^A}(Ob^A) \subseteq Ob^B, \quad Mor^A \subseteq \Phi_{Mor^B}^{Mor^A}(Mor^A) \subseteq Mor^B. \quad (6)$$

Умову (6) для моделей, які описуються категоріями A та B, слід інтерпретувати таким чином: фрагмент тезаурусу, що утворює похідну модель, яку описано категорією A, не повинен перевищувати за розмірами та складністю фрагмент тезаурусу, що утворює кінцеву модель, яку описано категорією B [19]. Але ця умова повинна виконуватися тільки у випадках, коли рівні тезаурусів $Sc_{Tez(A)}$ та $Sc_{Tez(B)}$ категорій будуть еквівалентні один одному.

Модель (1) дозволяє зробити висновок про існування двох основних способів реалізації запропонованого формалізованого опису одномісного коваріантного функтора. Перший спосіб передбачає пошук таких систем правил, які однозначно визначали б відповідність заздалегідь заданих елементів похідної і кінцевої категорій. Другий спосіб передбачає формування таких об'єктів і морфізмів категорії B, які задовольняли б заздалегідь заданим системам правил трансформації, об'єктам і морфізмам похідної категорії A, а також враховували б різницю рівнів подання тезаурусів моделей $Sc_{Tez(A)}$ і $Sc_{Tez(B)}$. З прикладної точки зору, перший спосіб реалізації функтора передбачає формування та реалізацію якогось загального алгоритму або бази правил перетворення похідного представлення ПЗ в кінцеве представлення цього ж ПЗ або множини можливих результатів його трансформації. При цьому передбачається, що текстові конструкції або елементи візуальних моделей, що описують це ПЗ, апріорно визначені як елементи такого алгоритму або бази правил і практично не змінюються з часом. Другий спосіб реалізації функторів передбачає створення та постійний розвиток деякої множини алгоритмів або якоїсь системи правил перетворення похідного представлення ПЗ в кінцеве представлення цього ж ПЗ або множини можливих результатів його трансформації. При цьому темпи зміни цієї множини алгоритмів або систем правил визначаються темпами появи нових або

зміни існуючих понять тезаурусу похідного представлення ПЗ. Поява кожного нового, раніше невідомого поняття у такому представленні ПЗ вимагає доповнення існуючого варіанту реалізації функтора новими алгоритмами чи правилами, які встановлюють спосіб і форму опису такого нового поняття у кінцевому представленні цього ж ПЗ або множини можливих результатів його трансформації.

Для проведення експериментальних досліджень було запропоновано обрати ПЗ спеціалізованої інформаційно-аналітичної системи медичного закладу (ІАСМЗ). Головною метою створення ІАСМЗ була автоматизація процесів обліку, контролю, аналізу та прийняття рішення за результатами здійснення клінічних лабораторних аналізів. Для досягнення цієї мети в ІАСМЗ реалізовано такі функціональні модулі:

– «Облік результатів клінічних аналізів пацієнта, введених оператором з власного робочого місця»;

– «Облік результатів клінічних аналізів пацієнта, сформованих системою управління приладом, який виконував аналіз, і переданих до ІАСМЗ за допомогою спеціального бізнес-сервісу ІАСМЗ»;

– «Контроль результатів клінічних аналізів пацієнта, введених до ІАСМЗ»;

– «Аналіз введених до ІАСМЗ результатів клінічних аналізів пацієнта»;

– «Розрахунки та публікація калькуляції результатів клінічних аналізів пацієнта»;

– «Обґрунтування та підтримка прийняття рішення за результатами дослідження клінічних аналізів пацієнта».

Під час розробки ПЗ ІАСМЗ було прийнято рішення використати клієнт-серверну архітектуру із застосуванням для клієнтської та серверної частин тієї самої мови програмування. Як базову мову програмування було обрано C/C++. Головне призначення серверної частини ПЗ – реалізація бізнес-логіки функцій ІАСМЗ. Головне призначення клієнтської частини ПЗ – реалізація користувацького інтерфейсу з використанням GUI-фреймворків (QT framework, MFC framework тощо).

Під час експлуатації ІАСМЗ виникла задача підтримки та супроводження її ПЗ для платформ Linux та AIX від IBM. Основну увагу під час даного дослідження було зосереджено на вирішенні питань кросплатформної міграції та підтримки серверної частини ПЗ ІАСМЗ.

5. Вирішення задачі модифікації моделі одномісного коваріантного функтора

5.1. Результати модифікації елементів моделі одномісного коваріантного функтора

В процесі дослідження було встановлено, що хід та результати модифікації моделі одномісного коваріантного функтора з врахуванням особливостей процесу кросплатформної міграції ПЗ ІС значною мірою залежать від бажаного для виконавців робіт варіанта реалізації цієї моделі в межах відповідної інформаційної технології. При цьому уся множина можливих варіантів реалізації може бути зведена до двох базових варіантів:

Перший варіант – використання першого способу реалізації моделі одномісного коваріантного функтора для кожного окремого випадку кросплатформної міграції (результатом є база правил, кожен окремий фрагмент якої відповідає за кросплатформну міграцію ПЗ до відповідної платформи).

Другий варіант – використання другого способу реалізації моделі одномісного коваріантного функтора для усіх підтримуваних випадків кросплатформної міграції (результатом є система правил, кожен окремий фрагмент якої відповідає за формування та підтримку бази правил кросплатформної міграції ПЗ до відповідної платформи).

На формальному рівні перший варіант запропоновано представити конусом одномісних коваріантних функторів, кожен з яких описує набір правил кросплатформної

міграції з платформи *A* до платформи *B*. Цей конус має такий вигляд:

$$Sw_{op}^{i} \xrightarrow{F_j} Sw_{op}^{j}, j = 1,...,n, \quad (7)$$

де Sw_{op}^{i} – опис похідної реалізації ПЗ експлуатованої ІС, виконаної в межах платформи Pl_i, i=1,2,…; Sw_{op}^{j} – опис кінцевої реалізації ПЗ експлуатованої ІС, виконаної в межах платформи Pl_j; F_j – позначення одномісного коваріантного функтора; n – кількість платформ, для яких слід підтримувати процес кросплатформної міграції ПЗ ІС.

В цьому варіанті модель одномісного коваріантного функтора (1)-(6) повинна враховувати умову, за якою рівні тезаурусів $Sc_{Tez(A)}$ та $Sc_{Tez(B)}$ категорій еквівалентні один одному. Але при цьому неможливо або недоцільно встановити наявність якогось вищого рівня тезаурусу $Sc_{MTez(A)} \xleftarrow{iso} Sc_{MTez(B)}$, який був би єдиним для обох рівнів тезаурусів $Sc_{Tez(A)}$ та $Sc_{Tez(B)}$ категорій, що описують ПЗ ІС для платформ *A* та *B* відповідно. Тому модель одномісного коваріантного функтора запропоновано модифікувати для першого варіанту шляхом зміни умов (6) таким чином:

$$Sc_{Tez(A)} \equiv Sc_{Tez(B)},\ Ob^A \equiv \Phi_{Ob^B}^{Ob^A}(Ob^A) \equiv Ob^B,\ Mor^A \equiv \Phi_{Mor^B}^{Mor^A}(Mor^A) \equiv Mor^B. \quad (8)$$

Дотримання еквівалентності в умовах (8) дозволяє отримати для платформи *B*, де планується експлуатувати мігроване ПЗ ІС, ті самі рішення, які було отримано та перевірено під час експлуатації похідного ПЗ цієї ж ІС на платформі *A*.

Отриманий результат дозволив описати формування моделі одномісного коваріантного функтора для першого варіанту кросплатформної міграції як метод, що складається з таких етапів.

Етап 1. Визначення аналітиком множин тезаурусів $Sc_{Tez(A)}$ та $Sc_{Tez(B)}$ і перевірка умови $Sc_{Tez(A)} \equiv Sc_{Tez(B)}$. Якщо умова не виконується, то визначення неможливості виконання обраної дії з кросплатформної міграції і завершення використання методу.

Етап 2. Визначення на основі тезаурусу $Sc_{Tez(A)}$ елементів множин Ob^A та Mor^A для похідного варіанту ПЗ.

Етап 3. Визначення на основі тезаурусу $Sc_{Tez(B)}$ елементів множин Ob^B та Mor^B для кінцевого варіанту ПЗ.

Етап 4. Визначення множини правил $\Phi_{Ob^B}^{Ob^A}(Ob^A)$ та $\Phi_{Mor^B}^{Mor^A}(Mor^A)$, для яких виконуються умови $Ob^A \equiv \Phi_{Ob^B}^{Ob^A}(Ob^A) \equiv Ob^B$ та $Mor^A \equiv \Phi_{Mor^B}^{Mor^A}(Mor^A) \equiv Mor^B$ відповідно.

Етап 5. Формування опису одномісного коваріантного функтора (1) з врахуванням результатів виконання Етапів 2-4.

Етап 6. Перевірка результату виконання Етапу 5 на відповідність умовам (2)-(5). Завершення використання методу.

Опис одномісного коваріантного функтора (1), отриманий в результаті використання запропонованого методу, можна багаторазово використовувати для кросплатформної міграції інших фрагментів ПЗ ІС або нових версій даного ПЗ без додаткових витрат на оплату праці фахівця з міграції ПЗ.

Другий варіант на формальному рівні запропоновано представити як сукупність таких одномісних коваріантних функторів:

– коконус функторів G_i, кожен з яких визначає базу правил модифікації ПЗ ІС до особливостей платформи Pl_i;

– ізоморфний функтор, який встановлює взаємну відповідність між визначеною системою правил кросплатформної міграції та описами цих правил для варіанту ПЗ, що є похідним для процесу кросплатформної міграції;

– конус функторів F_i, кожен з яких визначає базу правил міграції похідного варіанту ПЗ ІС у варіант ПЗ ІС, який планується експлуатувати в межах платформи Pl_i.

Цю сукупність функторів запропоновано представити діаграмою, що має вигляд:

$$\begin{array}{ccc} K_{op} & \xleftrightarrow{Iso} & Sw_{op} \\ \uparrow G_i & & \downarrow F_i \\ K_{op}^i & & Sw_{op}^i \end{array}, \quad i = 1,...,n, \tag{9}$$

де K_{op}^i – база правил, яка визначає особливості міграції похідного ПЗ до платформи Pl_i; n – кількість платформ, на яких здійснюється чи планується експлуатація ПЗ ІС; K_{op} – система правил кросплатформної міграції, яка є загальною для усіх підтримуваних платформ, на яких здійснюється чи планується здійснювати експлуатацію ПЗ ІС; Sw_{op} – варіант ПЗ, що є похідним для процесу кросплатформної міграції; Sw_{op}^i – варіант ПЗ, який є результатом процесу кросплатформної міграції для платформи Pl_i.

Роботу з формування баз правил K_{op} та K_{op}^i запропоновано формально описати такою комутативною діаграмою:

$$\begin{array}{ccc} Sc_{Tez}^i & \xrightarrow{O_i} & Sc_{MTez} \\ \downarrow \Phi_{K_{op}^i}^{Sc_{Tez}^i} & & \downarrow \Phi_{K_{op}}^{Sc_{Mtez}} \\ K_{op}^i & \xrightarrow{G_i} & K_{op} \end{array}, \quad i = 1,...,n, \tag{10}$$

де Sc_{Tez}^i – фрагмент тезаурусу, який визначає особливості кросплатформної міграції похідного варіанту ПЗ ІС до платформи Pl_i; Sc_{MTez} – тезаурус, який визначає загальні особливості міграції ПЗ ІС до платформи, яку обрано як похідну базову платформу для експлуатації ІС; O_i – коконус функторів, кожен з яких визначає базу правил модифікації тезаурусу платформи Pl_i до особливостей похідної базової платформи для експлуатації ПЗ ІС; $\Phi_{K_{op}^i}^{Sc_{Tez}^i}$ – функтор, який визначає особливості перетворення фрагменту тезауруса на множину правил кросплатформної міграції похідного варіанту ПЗ ІС до платформи Pl_i; $\Phi_{K_{op}}^{Sc_{MTez}}$ – функтор, який визначає особливості перетворення тезауруса на множину правил міграції ПЗ ІС до платформи, яку обрано як похідну базову платформу для експлуатації ІС.

В цьому варіанті модель одномісного коваріантного функтора (1)-(6) повинна враховувати існування тезаурусу, рівень якого перевищує рівні фрагментів тезаурусів окремих платформ. Такий тезаурус повинен базуватися на фрагменті тезауруса платформи,

яку обрано як похідну базову платформу для експлуатації ІАСМЗ, і розглядатися як результат сукупності бієктивних відображень елементів тезаурусів окремих платформ Pl_i. Тому для опису другого варіанту було запропоновано модифікувати опис одномісного коваріантного функтора (1) таким чином

$$\Phi_B^A = (Sc_{MTez}, Sc_{Tez(A)} \subseteq Sc_{MTez}, Ob^A, Mor^A, Sc_{Tez(B)} \subseteq Sc_{MTez}, Ob^B, Mor^B, \Phi_{Ob^A}^{Ob^B}, \Phi_{Mor^A}^{Mor^B}). \quad (11)$$

Запропонований модифікований опис одномісного коваріантного функтора (11) є узагальненим. У випадках застосування моделі (11) для формального опису окремих конусів та коконусів функторів, визначених на діаграмах (9) та (10), один з елементів ($Sc_{Tez(A)}$ або $Sc_{Tez(B)}$) буде набувати такого вигляду:

$$Sc_{Tez(A)} = Sc_{MTez}; \quad Sc_{Tez(B)} = Sc_{MTez}. \quad (12)$$

Умови (6) для другого варіанту було запропоновано змінити таким чином:

$$Ob^A \subseteq \Phi_{Ob^B}^{Ob^A}(Ob^A) \equiv Ob^B, \; Mor^A \subseteq \Phi_{Mor^B}^{Mor^A}(Mor^A) \equiv Mor^B. \quad (13)$$

Застосування модифікованої таким чином моделі одномісного коваріантного функтора в процесі кросплатформної міграції ПЗ ІС було запропоновано представити як дві послідовно виконувані роботи цього процесу: Робота 1 (підготовча) та Робота 2 (здійснення міграції). Суть Роботи 1 полягає у формуванні баз правил K_{op}^i та системи правил K_{op} на основі результатів аналізу тезаурусів Sc_{Tez}^i та визначення тезауруса Sc_{MTez}. Узагальнений сценарій виконання Роботи 1 було запропоновано представити як метод, що складається з таких етапів.

Етап 1. Визначення аналітиком множини платформ (Pl_i), на яких планується здійснювати експлуатацію ПЗ ІС, та платформи, яку обрано як похідну базову платформу для експлуатації ІС. Формування множини тезаурусів Sc_{Tez}^i і тезауруса Sc_{MTez}.

Етап 2. Для кожного з тезаурусів множини Sc_{Tez}^i визначення на його основі елементів множин Ob^A та Mor^A для моделей функтора коконуса O_i. Для тезауруса Sc_{MTez} визначення на його основі елементів множин Ob^B та Mor^B для моделей функтора коконуса O_i.

Етап 3. Для моделей функтора коконуса O_i визначення множин правил $\Phi_{Ob^B}^{Ob^A}(Ob^A)$ та $\Phi_{Mor^B}^{Mor^A}(Mor^A)$, для яких виконуються умови (13). Формування опису функтора коконуса O_i на основі моделі (11) з врахуванням отриманих результатів. Перевірка сформованого опису на відповідність умовам (2)-(5).

Етап 4. Формування множини моделей функторів $\Phi_{K_{op}^i}^{Sc_{Tez}^i}$ та моделі функтора $\Phi_{K_{op}}^{Sc_{MTez}}$. Синтез баз правил K_{op}^i та системи правил K_{op}.

Етап 5. Синтез моделей функторів коконуса G_i. Завершення використання методу.

Суть Роботи 2 полягає у здійсненні процесів кросплатформної міграції похідного варіанту ПЗ ІС Sw_{op} до варіантів ПЗ ІС Sw_{op}^i, які повинні експлуатуватися в межах

платформ Pl_i, на основі сформованих в результаті виконання Роботи 1 баз правил K_{op}^i та системи правил K_{op}. Узагальнений сценарій виконання Роботи 2 було запропоновано представити як метод, що складається з таких етапів.

Етап 1. Перевірка наявності оновленої версії похідного варіанту ПЗ ІС Sw_{op}, для якого треба виконати процеси кросплатформної міграції. Якщо такої версії немає, то визначення неможливості виконання процесів кросплатформної міграції і завершення використання методу.

Етап 2. Визначення на основі системи правил K_{op} елементів множин Ob^A та Mor^A для похідного варіанту ПЗ.

Етап 3. Визначення на основі баз правил K_{op}^i елементів множин Ob^B та Mor^B для кінцевого варіанту ПЗ.

Етап 4. Визначення множини правил $\Phi_{Ob^B}^{Ob^A}(Ob^A)$ та $\Phi_{Mor^B}^{Mor^A}(Mor^A)$, для яких виконуються умови $Ob^A \subseteq \Phi_{Ob^B}^{Ob^A}(Ob^A) \equiv Ob^B$ та $Mor^A \subseteq \Phi_{Mor^B}^{Mor^A}(Mor^A) \equiv Mor^B$ відповідно.

Етап 5. Формування опису одномісного коваріантного функтора (11) з врахуванням результатів виконання Етапів 2-4.

Етап 6. Перевірка результату виконання Етапу 5 на відповідність умовам (2)-(5). Завершення використання методу.

Опис одномісного коваріантного функтора (11), отриманий в результаті використання запропонованих методів, можна багаторазово використовувати для кросплатформної міграції нових версій ПЗ ІС у варіанти ПЗ цієї ж ІС, які повинні експлуатуватися в межах платформ Pl_i, i=1,2,…, без додаткових витрат на оплату праці фахівця з міграції ПЗ.

Основна відмінність між першим та другим способом здійснення кросплатформної міграції за умови використання запропонованих модифікованих моделей одномісного коваріантного функтора полягає у такому. Перший спосіб дозволяє здійснити процес кросплатформної міграції безпосередньо під час експлуатації поточної чи оновленої похідної версії ПЗ ІС. При цьому під час експлуатації похідної версії ПЗ ІС можуть бути виявлені та виправлені помилки, які не були знайдені під час розробки, реалізації та верифікації цієї версії ПЗ ІС. Такий спосіб є найкращим для ІС, які експлуатуються обмеженою кількістю підприємств-замовників в межах невеликої кількості платформ.

Другий спосіб орієнтовано на здійснення процесу кросплатформної міграції після завершення верифікації ПЗ ІС (до початку або в процесі валідації цього ПЗ). При цьому для кожної платформи з множини платформ (Pl_i), на яких планується експлуатація ПЗ ІС, процес кросплатформної міграції буде здійснюватися автоматично за умови можливості формування баз правил міграції K_{op}^i та загальної системи правил міграції K_{op}. Такий спосіб є найкращим для до ІС, що експлуатуються великою кількістю підприємств-замовників в межах значного різноманіття платформ, в межах яких здійснюється експлуатація цих ІС та їхніх ПЗ.

5.2. Експериментальна перевірка результатів модифікації моделі одномісного коваріантного функтора

Під час експериментальної перевірки результатів модифікації моделі одномісного коваріантного функтора було розглянуто особливості виконання процесу кросплатформної міграції ПЗ ІАСМЗ з платформи Linux до платформи AIX від IBM. Для

цієї міграції першою роботою процесу є робота з кросплатформної міграції результатів рефакторингу вихідного коду. Необхідність цієї роботи обумовлена такими причинами. Для ПЗ ІАСМЗ певна частина програмного коду, що реалізує бізнес-логіку функцій, повинна бути присутня як на клієнтській, так і на серверній частині (наприклад, бізнес-логіка функцій калькуляції). Але решта програмного коду повинна бути присутня лише в одній частині ПЗ – або клієнтській, або серверній. Тому слід виділити в ПЗ ІАСМЗ фрагменти ПЗ, які повинні виконуватися виключно на сервері.

Оскільки ПЗ ІАСМЗ вже протягом значного часу експлуатується на платформі Linux, для здійснення процесу кросплатформної міграції було вирішено використати перший спосіб. Тому виконання роботи з кросплатформної міграції результатів рефакторингу вихідного коду здійснювалося за визначеними етапами відповідного методу.

Під час виконання Етапу 1 було встановлено, що застосування мови програмування C/C++ дало для платформи Linux можливість реалізувати виділення серверної частини ПЗ ІАСМЗ за допомогою директиви препроцесора, наведеної на рис. 1.

```
#if defined __linux__
//source code that related to server side only will be included from
this section in compilation time
#endif
//all source code from this section will be included in compilation
time
```

Рис. 1. Директива препроцесора, яка виділяє серверну частину програмного забезпечення для платформи Linux

Таке рішення забезпечило для платформи Linux виділення в серверну частину лише того коду ПЗ ІАСМЗ, який відповідає за реалізацію операцій, що складають частини сценаріїв виконання функцій ІАС, призначених для виконання в межах серверної частини. Застосування цього рішення на етапі компіляції повинно було зменшити загальний розмір модулів ПЗ, що, в свою чергу, повинно було позитивно вплинути на загальну продуктивність усієї ІАСМЗ в межах платформи Linux.

Але застосувати наведену на рис. 1 директиву «__linux__» в межах платформи AIX неможливо, тому що ця платформа не гарантує присутність запропонованої директиви. Якщо ж здійснити міграцію ПЗ ІАСМЗ до платформи AIX, не змінюючи директиву «__linux__», усі фрагменти ПЗ ІАСМЗ, які були виділені цією директивою, будуть додані до клієнтської частини, а не до серверної. Тому для забезпечення виконання умови $Sc_{Tez(A)} \equiv Sc_{Tez(B)}$ було запропоновано для платформи AIX використати директиву препроцесора, яка розпізнається AIX-компілятором (див. рис. 2).

```
#if defined (_AIX) || defined (__linux__)
//source code that related to server side only will be included from
this section in compilation time
#endif
//all source code from this section will be included in compilation
time
```

Рис. 2. Директива препроцесора, яка виділяє серверну частину програмного забезпечення для платформ Linux та AIX

Під час виконання Етапу 2 було встановлено, що наведена на рис. 1 директива є

єдиним елементом множини Ob^A для похідного варіанту ПЗ ІАСМЗ, який експлуатується на платформі Linux. Оскільки таке виділення не змінює зв'язки між модулями ПЗ, встановлені під час проєктування та розробки ПЗ ІАСМЗ, множина Mor^A була визначена як пуста множина.

Під час виконання Етапу 3 було встановлено, що наведена на рис. 2 директива є єдиним елементом множини Ob^B для похідного варіанту ПЗ ІАСМЗ, який експлуатується на платформі Linux. Оскільки таке виділення не змінює зв'язки між модулями ПЗ, встановлені під час проєктування та розробки ПЗ ІАСМЗ, множину Mor^B було визначено як пусту множину.

Під час виконання Етапу 4 було визначено, що множину правил $\Phi_{Ob^B}^{Ob^A}(Ob^A)$ повинно складати одне правило. Загальний вигляд функції «directive_server», яка реалізує правило кросплатформної міграції з платформи Linux до платформи AIX директиви препроцесора з виділення серверної частини ПЗ наведено на рис. 3.

```
string directive_server (directive_x)
{
 if (directive_x == "#if defined __linux__")
 {
     return "#if defined (_AIX) || defined (__linux__)"
 }
}
```

Рис. 3. Загальний вигляд функції «directive_server»

Використання цього правила дозволяє стверджувати, що умову $Ob^A \equiv \Phi_{Ob^B}^{Ob^A}(Ob^A) \equiv Ob^B$ для даного одномісного коваріантного функтора виконано (наведене на рис. 3 правило $\Phi_{Ob^B}^{Ob^A}(Ob^A)$ містить частини, еквівалентні як Ob^A, так і Ob^B).

Множина правил $\Phi_{Mor^B}^{Mor^A}(Mor^A)$, виходячи з того, що $Mor^A = \varnothing$ та $Mor^B = \varnothing$, теж була визначена як пуста множина. Тому умову $Mor^A \equiv \Phi_{Mor^B}^{Mor^A}(Mor^A) \equiv Mor^B$ слід вважати виконаною.

В результаті виконання Етапу 5 було отримано екземпляр моделі одномісного коваріантного функтора, який має вигляд:

$$\begin{aligned}&\Phi_{AIX}^{Linux} = (\text{"}__linux__\text{"}, \text{"}\#if\ defined\ __linux__\text{"}, \varnothing,\\ &_AIX, \text{"}\#if\ defined(_AIX) || defined(__linux__)\text{"}, \varnothing, .\\ &\text{"}string\ directive_server(directive_x)\text{"}, \varnothing)\end{aligned} \quad (14)$$

В результаті виконання Етапу 6 було встановлено, що отримана модель (14) відповідає умовам (2)-(5).

Сформована модель одномісного коваріантного функтора (14), яка описує роботу з кросплатформної міграції результатів рефакторингу вихідного коду ПЗ ІАСМЗ, дозволяє автоматизувати виконання цієї роботи для усіх подальших версій цього ПЗ. Таке рішення забезпечить коректне лінкування з API платформи, і дозволить звести до мінімуму витрати

людино-годин на міграцію ПЗ ІАСМЗ після внесення змін до похідного варіанту цього ПЗ.

6. Обговорення результатів дослідження

Основними результатами дослідження є розроблені дві модифікації моделі одномісного коваріантного функтора. Перша з цих модифікацій дозволяє описати роботи процесу кросплатформної міграції як конуси одномісних коваріантних функторів, кожен з яких встановлює правило виконання цієї роботи для похідної та кінцевої платформ. Суть цього варіанту модифікації полягає в уточненні умови (6) існування одномісного коваріантного функтора, яка, на відміну від базового варіанту, встановлює наявність еквівалентних описів елементів множин об'єктів та морфізмів похідної та кінцевої категорій та множин відповідних правил функтора. Друга з цих модифікацій дозволяє описати роботи процесу кросплатформної міграції як конуси одномісних коваріантних функторів, кожен з яких визначає базу правил міграції похідного варіанту ПЗ ІС у варіант ПЗ ІС, який планується експлуатувати в межах платформи Pl_i . Але, на відміну від першого варіанту, конус функторів у другому варіанті є результатом формування коконуса функторів, кожен з яких визначає базу правил модифікації тезаурусу платформи Pl_i до особливостей похідної базової платформи для експлуатації ПЗ ІС, та коконуса функторів, кожен з яких визначає базу правил модифікації ПЗ ІС до особливостей платформи Pl_i . При цьому отримані результати формування цих коконусів узгоджуються між собою за рахунок існування ізоморфних одномісних коваріантних функторів, які встановлюють взаємно-однозначне відображення цих результатів. Суть цього варіанту модифікації полягає в уточненні не тільки умови (6), а й базового формального опису одномісного коваріантного функтора (1).

Запропоновані модифікації моделі одномісного коваріантного функтора дозволяють, на відміну від базової моделі цього функтора (1)-(6), краще пристосувати цю модель до особливостей процесу кросплатформної міграції ПЗ ІС різного призначення. Використання модифікованих моделей функтора для формального опису робіт та діяльностей процесу кросплатформної міграції ПЗ ІС дозволяють встановити основні вимоги до інформаційних технологій, що дозволять автоматизувати виконання цього процесу. Така автоматизація дозволить скоротити витрати часу та ресурсів на виконання процесу кросплатформної міграції ПЗ ІС під час впровадження та експлуатації даного ПЗ на платформах підприємств-замовників цих ІС.

Основні характеристики запропонованих модифікацій моделі одномісного коваріантного функтора та їх значення наведено у табл. 1.

На відміну від наведених у [18] рішень, запропоновані модифіковані моделі одномісного коваріантного функтора орієнтовано на опис процесу кросплатформної міграції ПЗ саме ІС як окремого класу ІТ-продуктів. На відміну від рішень, розроблених у [19], розроблені моделі спрямовано на опис робіт та діяльностей процесів валідації, функціонування та супроводження ІС.

Головним обмеженням застосування отриманих результатів даного дослідження є спрямованість модифікованих моделей одномісного коваріантного функтора саме на опис процесу кросплатформної міграції ПЗ ІС. Тому застосування отриманих моделей для формального опису інших процесів життєвого циклу ІС вимагає проведення додаткових досліджень з модифікації або уточнення отриманих моделей до особливостей цих процесів. Ще одним досить значним недоліком отриманих результатів є необхідність додаткових витрат зусиль та часу аналітика на встановлення фактів еквівалентності елементів тезаурусів платформ, між якими повинна відбуватися міграція ПЗ ІС. Намагання ліквідувати цей недолік приводить до необхідності вирішення задачі оцінювання

Таблиця 1

Порівняльні характеристики запропонованих модифікацій моделі одномісного коваріантного функтора

Найменування характеристики	Перший варіант модифікації	Другий варіант модифікації
Формальне представлення	Конус одномісних коваріантних функторів, кожен з яких описує набір правил кросплатформної міграції ПЗ ІС з платформи A до платформи B	– коконус функторів, кожен з яких визначає базу правил модифікації тезаурусу платформи Pl_i до особливостей похідної базової платформи для експлуатації ПЗ ІС; – коконус функторів, кожен з яких визначає базу правил модифікації ПЗ ІС до особливостей платформи Pl_i; – ізоморфний функтор, який встановлює взаємну відповідність між визначеною системою правил кросплатформної міграції та описами цих правил для варіанту ПЗ, що є похідним для процесу кросплатформної міграції; – конус функторів, кожен з яких визначає базу правил міграції похідного варіанту ПЗ ІС у варіант ПЗ ІС, який планується експлуатувати в межах платформи Pl_i.
Концепція міграції	Концепція «точка – точка» (міграція відбувається між двома апріорно обраними платформами)	Концепція «загальносистемна шина» (міграція відбувається між базовою платформою та множиною платформ (Pl_i), які апріорно обрані для експлуатації ПЗ ІС)
Необхідність централізованої системи / бази правил міграції	Ні	Так
Обробка тезаурусів платформ	Здійснюється аналітиком вручну	Може бути автоматизована
Технологічний рівень реалізації	Може бути реалізована вручну або шляхом створення інформаційної технології	Потребує спеціальної інформаційної технології
Потреба в персоналі	Потрібен один фахівець, добре знайомий з тезаурусами обох платформ	Потрібна команда фахівців, знайомих з тезаурусами усіх запропонованих для міграції платформ Pl_i
Потреба в часі для налаштування моделі	Може бути налаштована за один робочий день	Потребує виділення для налаштування проміжку часу, не більше однієї ітерації ІТ-проєкту

Кінець таблиці 1

Найменування характеристики	Перший варіант модифікації	Другий варіант модифікації
Можливість повторного використання	Тільки для апріорно обраної пари платформ	Тільки для апріорно обраної множини платформ (Pl_i)
Стадія та процеси життєвого циклу ІС, в межах якої слід застосовувати модифікацію	Стадія експлуатації ІС (процеси функціонування та супроводження ІС)	Стадія впровадження ІС (процеси інтеграції та валідації ІС)
Відповідальні за застосування модифікації	Фахівці з супроводження ПЗ ІС	Фахівці з інтеграції та/або валідації ПЗ ІС на платформах підприємств-замовників ІС
Кількість впроваджень ІС, для яких доцільне застосування модифікації	Невелика (до 10-15 впроваджень)	Практично необмежена

ефективності застосування інформаційних технологій автоматизованої кросплатформної міграції ПЗ ІС, які базуються на категорно-функторному апараті, і проведення значної кількості подальших досліджень в даному напрямі.

Ці та інші обмеження і недоліки отриманих результатів з модифікації моделі одномісного коваріантного функтора визначили такі напрями подальших досліджень в цій галузі:

– дослідження технологічних і прикладних рішень з реалізації модифікованих моделей одномісного коваріантного функтора;

– дослідження варіантів модифікації моделі одномісного коваріантного функтора для формального опису інших процесів життєвого циклу ІС;

– дослідження з формального категорно-функторного опису та розробки архітектури автоматизованої інтелектуальної системи управління життєвим циклом ІС різного призначення.

7. Висновки

У ході даного дослідження було вирішено задачу модифікації моделі функтора для опису трансформації ПЗ ІС в межах її кросплатформної міграції. Під час вирішення цієї задачі було здійснено:

– модифікацію моделі одномісного коваріантного функтора (7) шляхом уточнення умов (8) існування такого функтора в процесі кросплатформної міграції ПЗ ІС під час експлуатації цієї ІС;

– модифікацію моделі одномісного коваріантного функтора (11) шляхом уточнення опису цього функтора з врахуванням особливостей процесу кросплатформної міграції ПЗ ІС під час валідації цієї ІС;

– експериментальну перевірку модифікованих моделей одномісного коваріантного функтора під час здійснення процесу кросплатформної міграції ПЗ ІАСМЗ і, зокрема, роботи з кросплатформної міграції результатів рефакторингу вихідного коду даного ПЗ.

Отримані результати підтверджують можливість автоматизованого виконання процесу кросплатформної міграції ПЗ ІС шляхом створення і використання відповідних інформаційних технологій.

Основним напрямом подальших досліджень запропоновано вважати дослідження особливостей технологічних і прикладних рішень з реалізації модифікованих моделей одномісного коваріантного функтора та оцінювання ефективності застосування подібних рішень під час функціонування та супроводження ІС різного призначення.

Перелік посилань:

1. Awasthi, Y., Zengeni, T. M., & Makambwa, T. A Framework for Cloud Migration in Academic Institutions. ResearchGate. 2024. URL: https://www.researchgate.net/publication/381952876_A_Framework_for_Cloud_Migration_in_Academic_Institutions

2. Emmanuel Cadet, Olajide Soji Osundare, Harrison Oke Ekpobimi, Zein Samira & Yodit Wondaferew Weldegeorgise. Cloud migration and microservices optimization framework for large-scale enterprises. Open Access *Research Journal of Engineering and Technology.* 2024. Vol. 7(2). P. 046–059. URL: https://doi.org/10.53022/oarjet.2024.7.2.0059

3. Gethers, M., Dit, B., Kagdi, H., Poshyvanyk, D. Integrated impact analysis for managing software changes. *34th International Conference on Software Engineering (ICSE)*. Zurich, Switzerland. 2012. P. 430-440. URL: doi: https://doi.org/10.1109/ICSE.2012.6227172

4. Manapian, A., Prompoon, N. Software time estimation model for requirements change based on software prototype profiles using an analogy estimation method. *2014 International Computer Science and Engineering Conference (ICSEC).* Khon Kaen. Thailand. 2014. P. 366-371. URL: doi: https://doi.org/10.1109/ICSEC.2014.6978224

5. Miguel, M.A., Araújo, M.A.P., David, J.M.N., Braga, R. A framework to support effort estimation on software maintenance and evolution activities. *12th Brazilian Symposium on Information Systems: Information Systems in the Cloud Computing Era*, Proceedings. Florianopolis - Santa Catarina. 2016. P. 232-239.

6. Rostami, K., Stammel, J., Heinrich, R., Reussner, R. Change Impact Analysis by Architecture-based Assessment and Planning. *Lecture Notes in Informatics*, Proceedings - Series of the Gesellschaft fur Informatik. Hannover. 2017. Vol. P267, P. 69-70.

7. Rostami, K., Heinrich, R., Busch, A., Reussner, R. Architecture-Based Change Impact Analysis in Information Systems and Business Processes. *2017 IEEE International Conference on Software Architecture (ICSA)*. Gothenburg, Sweden. 2017. P. 179-188. URL: https://doi.org/10.1109/ICSA.2017.17.

8. Dugar, M. How Machine Learning Can Help Developers. *24th International Arab Conference on Information Technology*, ACIT 2023. Ajman. 2023. Code 198200. URL: https://doi.org/10.1109/ACIT58888.2023.10453880

9. Steingartner, W., Novitzká, V., Bačíková, M., Korečko, Š. New approach to categorical semantics for procedural languages. *Computing and Informatics*, 2017. Vol. 36, Iss. 6. P. 1385-1414. URL: https://doi.org/10.4149/cai_2017_6_1385.

10. Crăciunean, D.-C. Categorical modeling method, proof of concept for the petri net language. *Proceedings of the 7th International Conference on Model-Driven Engineering and Software Development*, 2019. P. 283-291. URL: https://doi.org/10.5220/0007360602830291

11. Beohar, H., König, B., Küpper, S., Silva, A., Wissmann, T. A coalgebraic treatment of conditional transition systems with upgrades. *Logical Methods in Computer Science*, 2018. Vol. 14, Iss. 128, 19. URL: https://doi.org/10.23638/LMCS-14(1:19)2018.

12. Hou, J., Zhang, Y., Rana, A.D. Describing Approach for Model-Driven Collaborative Application Development. *International Conference on Artificial Intelligence and Advanced Manufacturing*, AIAM 2019, Proceedings. 2019. P. 336-343. URL: https://doi.org/10.1109/AIAM48774.2019.00073.

13. Hou, J., Xu, C., Zhang, Y. Architecture-Based Semantic Description Framework for Model Transformation. *5th International Conference on Natural Language Processing and Information Retrieval*, NLPIR 2021, ACM International Conference Proceeding Series, 2021. P. 73-80. URL: https://doi.org/10.1145/3508230.3508241

14. Steingartner, W. Perspectives of semantic modeling in categories. *Journal of King Saud University - Computer and Information Sciences.* 2025. Vol. 37. Iss. 3. № 19. URL: https://doi.org/10.1007/s44443-025-00010-9.

15. Chowdhury, S., Clause, N., Mémoli, F., Sánchez, J.Á., Wellner, Z. New families of stable simplicial filtration functors. *Topology and its Application.* 2020. Vol. 279. 107254. URL: https://doi.org/10.1016/j.topol.2020.107254.

16. Lochbihler, A., Marić, O. Authenticated Data Structures as Functors in Isabelle/HOL. *2nd Workshop on Formal Methods for Blockchains*, FMBC 2020, Open Access Series in Informatics, 2020, Vol. 84, 6. URL: https://doi.org/10.4230/OASIcs.FMBC.2020.6.

17. Sun Y.Z., Wang S.T. CNOK: A C++ Glauber model code for single-nucleon knockout reactions. *Computer Physics Communications*, 2023, Vol. 288, № 108726. URL: https://doi.org/10.1016/j.cpc.2023.108726.

18. Bezditnyi, V., Chebanyuk, O. Software Engineering Fundamentals to Design Application for Modern Game Engines. *2024 14th International Scientific and Practical Programming Conference*, UkrPROG 2024, CEUR Workshop Proceedings, Vol. 3806, P. 89–99.

19. Левикін В.М., Євланов М.В., Керносов М.А. Патерни проєктування вимог до інформаційної системи:

моделювання та застосування: монографія. (рос.) Харків: ТОВ «Компанія СМІТ», 2014. 320 с.

20. Левикін В.М., Євланов М.В. Задача визначення функторів між категорними моделями інформаційної системи. (рос.) *Проблеми біоніки*. 2003. Вип. 58. С. 62-67.

21. Євланов М.В. Формалізація взаємних відображень моделей інформаційних систем. (рос.) *Materialy IV Miedzynarodowej naukowi-praktycznej konferencji Nowoczesnych naukowych osiagniec - 2008*. Т. 13. Matematyka. Fizyka. Nowoczesne informacyjne technologie. Przemysl: Nauka i studia. P. 82-85.

22. Awodey S. Category Theory. 2nd ed. NY.: OXFORD UNIVERSITY PRESS, 2010. XVI+311 p.

Надійшла до редколегії 12.06.2025 р.

Круглик Андрій Сергійович, аспірант кафедри програмного забезпечення комп'ютерних систем, факультет інформаційних технологій, НТУ «Дніпровська Політехніка», м. Дніпро, Україна, e-mail: swatkrim@gmail.com.

Левикін Віктор Макарович, доктор технічних наук, професор, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: viktor.levykin@nure.ua, ORCID: https://orcid.org/0000-0002-7929-515X

Євланов Максим Вікторович, доктор технічних наук, професор, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: maksym.ievlanov@nure.ua, ORCID: https://orcid.org/0000-0002-6703-5166

Мороз Борис Іванович, доктор технічних наук, професор, професор кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна, e-mail: moroz.b.i@nmu.one, ORCID: https://orcid.org/0000-0002-5625-0864

Мороз Дмитро Максимович, доктор філософії, доцент кафедри програмного забезпечення комп'ютерних систем НТУ «Дніпровська політехніка», м. Дніпро, Україна, e-mail: moroz.d.m@nmu.one, ORCID: https://orcid.org/0000-0003-2577-3352

УДК 004.932.2 DOI: 10.30837/0135-1710.2025.185.023

П.Е. СИТНІКОВА, М.О. ГРИЦАЙ

МОДЕЛЮВАННЯ ТА ДОСЛІДЖЕННЯ МЕТОДІВ ДЕТЕКЦІЇ ПОГЛЯДУ КОРИСТУВАЧА У СИСТЕМАХ ЛЮДИНО-КОМП'ЮТЕРНОЇ ВЗАЄМОДІЇ

Реалізовано систему зорового введення, що дозволяє керувати курсором на екрані за допомогою рухів очей користувача. Рішення базується на використанні нейронної мережі для прогнозування напрямку погляду та враховує просторову орієнтацію голови, отриману за допомогою оцінки пози. Для підвищення стабільності результатів застосовано методи фільтрації шуму та згладжування координат. Програмний модуль виконує нормалізацію зображення обличчя, трансформує дані у тривимірному просторі та обчислює точку перетину вектора погляду з площиною екрана, що дає змогу точно відображати фокус уваги користувача у вигляді візуального маркера. Результати підтвердили працездатність системи в режимі реального часу.

1. Вступ

Сучасні технології взаємодії людини з комп'ютером стрімко розвиваються, наслідком чого є зростання інтуїтивності та зручності процесу керування пристроями. Одним із перспективних напрямів є зорове введення, що дозволяє користувачам керувати системами за допомогою рухів очей, зменшуючи необхідність фізичних маніпуляцій. Це особливо важливо для людей з обмеженими можливостями, а також у середовищах, де традиційні методи введення можуть бути незручними або неефективними.

Основною проблемою при розробці таких систем є точність визначення погляду, обробка великого обсягу даних у реальному часі та інтеграція цих технологій у існуючі програмні та апаратні комплекси. Сучасні методи комп'ютерного зору та машинного

навчання [1] відкривають нові можливості для покращення точності та швидкодії зорового введення, однак вони потребують детального дослідження та моделювання.

У даному дослідженні розглянуто різні підходи до моделювання систем зорового введення, проаналізовано їхні переваги та недоліки, а також досліджено перспективи застосування в різних сферах, таких як ігрова індустрія та управління складними системами.

2. Аналіз існуючих досліджень та формулювання проблеми

Розвиток технологій відстеження погляду значно розширив можливості безконтактної взаємодії людини з комп'ютером, відкриваючи нові перспективи для управління цифровими пристроями. У сучасних дослідженнях активно використовуються як класичні методи комп'ютерного зору, так і підходи, засновані на глибокому навчанні [2], що дозволяє покращити точність визначення напрямку погляду та зменшити вплив зовнішніх факторів, таких як зміна освітлення або положення голови користувача.

Аналіз існуючих технологій свідчить, що традиційні алгоритми, такі як перетворення Гафа та каскади Гаара [3], забезпечують базове виявлення очей та визначення їхнього розташування на зображенні. Проте їхня ефективність обмежена у складних умовах, зокрема при зміні освітлення або через індивідуальні особливості обличчя користувачів. Використання згорткових нейронних мереж (convolutional neural network, CNN), зокрема моделей типу GazeNet [4], дозволяє значно підвищити точність аналізу, оскільки вони можуть враховувати ширший набір параметрів і навчатися на великих обсягах даних.

Окрему увагу варто приділити фільтрації шумів, спричинених мікрорухами очей та коливаннями голови. Для цього застосовуються алгоритми, такі як фільтр Калмана [5] та методи експоненційного згладжування[6], що допомагають стабілізувати рух курсора та підвищити точність прогнозування стану та поведінки системи. Впровадження таких методів є важливим кроком для забезпечення плавності та точності керування.

Ще одним важливим аспектом реалізації системи зорового введення є механізми підтвердження введення команд. У сучасних рішеннях використовуються різні підходи, серед яких фіксація погляду на певному елементі протягом заданого часу, подвійне кліпання або інтеграція голосових команд. Аналіз цих методів свідчить про їхню ефективність у різних сценаріях використання, проте кожен із них має свої обмеження. Вибір оптимального підходу залежить від специфіки завдань, які виконує користувач.

Серед існуючих підходів до реалізації технології відстеження погляду можна виділити кілька популярних рішень:

– WebGazer.js – браузерне програмне рішення, що працює на основі вебкамери і звичайного JavaScript. Не потребує встановлення додаткового ПЗ або драйверів, але має низьку точність (~4–6°) [7] і сильну залежність від освітлення та позиції голови;

– Tobii Eye Tracker – апаратне рішення, що базується на інфрачервоному (ІЧ) освітленні та високоточних сенсорах. Забезпечує високу точність (до 0.5°) і стійкість до зовнішніх факторів, але потребує спеціалізованого обладнання[8] та має високу вартість;

– MediaPipe FaceMesh – бібліотека розроблена компанією Google[9], яка забезпечує виявлення 3D-точок обличчя в режимі реального часу. Є основою для багатьох дослідницьких і прикладних систем, має високу швидкість роботи, але не надає повноцінного прогнозування погляду.

На відміну від згаданих рішень, у даному дослідженні використано модифіковану модель на основі VGG16, з подальшим доповненням шарами згортки, уваги (SELayer) та індивідуального калібрування, яка поєднує глибоке згорткове навчання для обробки зображень обличчя та очей, а також повнозв'язні шари для прогнозування напрямку погляду. Модель приймає на вхід нормалізовані зображення обох очей та обличчя та на

виході формує два кутові параметри pitch та yaw, які описують напрямок погляду у сферичних координатах. Таким чином, мережа реалізує підхід, заснований на зображенні, що дозволяє досягти прийнятної точності без використання апаратного ІЧ-обладнання.

Отже, аналіз сучасних підходів до побудови систем зорового введення вказує на низку ключових проблем, які залишаються актуальними для дослідницької спільноти. Зокрема, більшість наявних рішень демонструють обмежену точність при зміні умов освітлення або позиції голови користувача, є чутливими до шумів і нестабільностей у відеопотоці, а також не завжди забезпечують зручні й надійні механізми підтвердження введення команд. Окрему складність становить забезпечення стабільного переміщення курсора за поглядом без випадкових коливань. У зв'язку з цим дане дослідження спрямоване на вирішення вказаних проблем шляхом інтеграції методів нормалізації зображень, фільтрації шумів та вдосконалених алгоритмів підтвердження введення, що дозволяє підвищити загальну точність і зручність взаємодії користувача з комп'ютером.

3. Мета і задачі дослідження

Метою цього дослідження є розробка та вдосконалення методів керування курсором за допомогою відстеження погляду, що забезпечують високу точність, стабільність, плавність руху та зручність використання, з урахуванням сучасних викликів у галузі комп'ютерного зору.

Для досягнення цієї мети необхідно вирішити такі задачі:

– обґрунтувати вибір структури програмної реалізації та визначити ключові компоненти системи керування курсором за поглядом;

– реалізувати ключові компоненти системи керування курсором за поглядом;

– провести тестування системи та оцінити її точність, швидкодію, адаптивність і зручність використання.

4. Технологічні аспекти реалізації системи керування курсором за поглядом

Об'єктом даного дослідження є процес зорового керування комп'ютерним курсором на основі аналізу напрямку погляду користувача.

Головною гіпотезою дослідження є припущення, що використання нейронної мережі у поєднанні з алгоритмами нормалізації, фільтрації та геометричної трансформації дозволяє реалізувати точне, стабільне та зручне керування курсором без необхідності використання спеціалізованого апаратного забезпечення.

Система керування курсором на основі погляду реалізована як модульний ІТ-продукт, що обробляє відеоінформацію у реальному часі, визначає координати напрямку погляду, фільтрує їх та відображає на екрані. У процесі розробки цієї системи було розроблено структуру її програмної реалізації, яка забезпечує повноцінний цикл обробки вхідного відеопотоку та визначення координат погляду на екрані – від захоплення зображення до проєкції координат на екран.

Система (та її програмна реалізація) складається з таких основних модулів:

– модуль відеозахоплення – здійснює захоплення кадрів з вебкамери з заданими параметрами (роздільна здатність, частота кадрів);

– модуль попередньої обробки – відповідає за виявлення обличчя та ключових точок за допомогою алгоритму MediaPipe FaceMesh, а також виконує нормалізацію зображень очей і обличчя;

– модуль прогнозування погляду – реалізований у вигляді нейронної мережі, яка отримує нормалізовані зображення та генерує прогноз напрямку погляду у вигляді кутів нахилу (pitch, yaw);

– модуль обчислення координат – перетворює кутові значення у тривимірний вектор і розраховує точку перетину цього вектору з площиною екрана;

– модуль виводу – забезпечує візуалізацію отриманої точки на екрані у вигляді лазерного курсора або траєкторії.

Структуру програмної реалізації зображено на рис. 1.

Рис. 1. Структура програмної реалізації системи керування курсором на основі погляду

Модуль калібрування виконує збір пар даних, у яких кожне зображення очей відповідає відомій координаті точки на екрані, на яку дивиться користувач. Для цього користувач фокусує погляд на заздалегідь визначених контрольних точках, а система зберігає відповідні зображення та координати. На основі цих пар (реальні координати на екрані та ознаки зображення очей) формується матриця афінного перетворення, яка дозволяє трансформувати координати, розраховані на основі напрямку погляду, у координати екранного простору. Для цього застосовується афінне перетворення вигляду:

$$xc = a_{11} * xg + a_{12} * yg + a_{13} \ ,$$

$$yc = a_{21} * xg + a_{22} * yg + a_{23},$$

де xg, yg – координати, отримані з моделі комп'ютерного зору; xc, yc – координати на екрані; $a_{ij}, i = 1, \dots, 2;\ j = 1, \dots, 3$ – коефіцієнти, серед яких:

– a_{11}, a_{21} – коефіцієнти масштабування за осями x та y; визначають, наскільки потрібно розтягнути або стиснути координати з камери, щоб привести їх до розмірів екрана;

– a_{12}, a_{22} – коефіцієнти, що відповідають за поворот координатної системи (наприклад, якщо користувач сидить трохи під кутом відносно камери);

– a_{13}, a_{23} – коефіцієнти зсуву; компенсують те, що користувач може сидіти не точно по центру перед камерою (зміщення вліво/вправо або вгору/вниз).

Їхні значення розраховуються автоматично методом найменших квадратів під час калібрування – на основі відповідності між точками, які бачить камера, і реальними точками на екрані.

Захоплення відео з вебкамери відбувається у реальному часі. Кожен кадр передається до модуля комп'ютерного зору, який за допомогою алгоритму MediaPipe FaceMesh визначає положення обличчя, очей, а також ключові орієнтири – зокрема, центр зіниці. Отримані зображення проходять попередню обробку, що включає нормалізацію яскравості, масштабування та вирівнювання з урахуванням положення голови користувача.

Нормалізовані зображення очей і обличчя надходять на вхід згорткової нейронної мережі, яка виконує прогноз напрямку погляду у вигляді кутів нахилу (pitch та yaw). Модель приймає три канали вхідних даних – ліве око, праве око та повне обличчя – і повертає два кутові значення, які описують орієнтацію погляду в сферичних координатах.

Кожне з цих зображень обробляється окремою гілкою мережі, яка складається з послідовності згорткових і активаційних шарів. Це дозволяє виділити найінформативніші ознаки для кожної з зон – очей та обличчя. Отримані ознаки з'єднуються, після чого проходять через серію повнозв'язних шарів, які формують підсумкове передбачення.

На виході мережа видає два значення: кут повороту погляду по вертикалі (pitch) та по горизонталі (yaw). Ці кути надалі використовуються для побудови просторового вектора погляду, який перетинається з площиною екрана для визначення координат, на які дивиться користувач.

Модель також враховує індивідуальні особливості користувачів – для кожного з них в мережі передбачено невелике коригування вихідних результатів, що покращує загальну точність системи.

Таким чином, нейронна мережа є центральним компонентом системи та відповідає за точне визначення напрямку погляду в режимі реального часу.

Для підвищення стабільності положення курсора застосовується фільтрація отриманих координат погляду. Через наявність шумів, спричинених мікрорухами очей або похибками камери положення курсора без фільтрації є нестабільним. Для згладжування використовуються такі два підходи.

Перший підхід – це фільтр Калмана, який поєднує попереднє передбачене положення курсора з новим виміром. Його формула в простій формі:

$$x_k = x_{k-1} + K * (z_k - x_{k-1}),$$

де x_k – оновлене положення курсора; z_k— нове виміряне положення; K – коефіцієнт корекції (Kalman Gain), який визначає, наскільки сильно система довіряє новому виміру. У простій реалізації це фіксоване значення (наприклад, 0.5), але в повноцінному фільтрі Калмана цей коефіцієнт адаптивний і змінюється в залежності від оцінки шуму та точності вимірів).

Другий підхід – експоненційне згладжування, що обраховується за формулою:

$$S_t = \alpha * x_t + (1 - \alpha) * S_{(t-1)},$$

де α – коефіцієнт згладжування, який контролює ступінь довіри до нового значення (наприклад, при α = 0.2, нове значення змінює курсор на 20 %, а 80 % залишаються від попереднього стану); x_t – поточне «сире» значення координати; S_t – згладжене значення.

Після фільтрації координати передаються до інтерфейсу керування, де положення курсора оновлюється відповідно до напрямку погляду користувача. Частота оновлення залежить від кількості кадрів за секунду (FPS) вебкамери і швидкості обробки кадру. Якщо вебкамера працює на 30 FPS і час обробки кадру становить 20 мс, затримка реакції системи буде приблизно 33–50 мс, що є прийнятним для комфортної взаємодії.

Для ініціації дій (натискань) система може використовувати різні підходи. Один з них — подвійне кліпання очима, яке фіксується і розпізнається як команда. Альтернативно, використовується фіксація погляду на елементі протягом певного часу, після чого виконується вибір. Це дозволяє уникнути випадкових спрацювань та зробити керування природнішим.

Особливістю реалізованої системи є інтеграція згорткової нейронної мережі [10], яка прогнозує напрямок погляду у вигляді кутів нахилу (pitch, yaw), з геометричними обчисленнями у тривимірному просторі. На основі передбаченого напрямку формується тривимірний вектор погляду, який перетинається з площиною екрана, заданою в координатах камери [11]. Точка перетину визначає просторову позицію фокусу уваги користувача, що згодом проєктується у координати екрана.

На відміну від типових рішень, які просто видають координати без урахування просторової сцени, запропонована система поєднує глибинне навчання [12] з математичною інтерпретацією сцени, що дозволяє легко адаптувати її до різних положень камери, розмірів екрана або положення користувача. У порівнянні з комерційними аналогами (наприклад, Tobii Eye Tracker), дана реалізація є відкритою, повністю програмною та не вимагає спеціального інфрачервоного обладнання. Для роботи достатньо звичайної вебкамери, а всі компоненти – від прогнозу до візуалізації – можуть бути змінені або вдосконалені. Новизна підходу полягає саме в поєднанні нейронного моделювання з просторовими геометричними обчисленнями та програмною фільтрацією без апаратної залежності.

5. Результати дослідження

Для оцінки роботи системи керування курсором на основі відстеження погляду було проведено серію експериментів, що включали як практичне тестування на реальному пристрої, так і моделювання поведінки алгоритмів на згенерованих даних.

У практичному тесті система запускалась на ноутбуці з операційною системою Windows 10 та вбудованою вебкамерою. Після калібрування, під час якого користувач фіксував погляд на різних точках екрана, система виводила в реальному часі червону точку на площину – вона відображала напрямок погляду. Для візуального оцінювання точності користувач виконував рух очей зліва направо, з фіксацією погляду на крайніх точках, а також описував поглядом умовне коло в центрі екрана. Суб'єктивно система демонструвала плавний рух і стабільну реакцію, однак при малому розмірі мітки були помітні мікроколивання.

Для кількісної оцінки стабільності було проведено імітаційний експеримент, у якому змодельовано синусоїдальний рух погляду — це умовний сценарій, у якому користувач рухає поглядом горизонтально з однаковою швидкістю (наприклад, при читанні тексту або скануванні лінії меню). До «ідеальної» траєкторії погляду було додано випадковий шум, який імітує нестабільність через мікрорухи очей та шум вебкамери. Далі ці шумні дані оброблялись двома методами: експоненційним згладжуванням та фільтром Калмана.

Графік результатів, представлений на рис. 2, показав, що початкові дані без фільтрації значно коливаються, що підтверджує нестабільну поведінку курсора при прямому використанні координат моделі. Після застосування експоненційного згладжування

координати стали суттєво плавнішими, а використання фільтра Калмана дозволило максимально наблизити траєкторію до ідеальної.

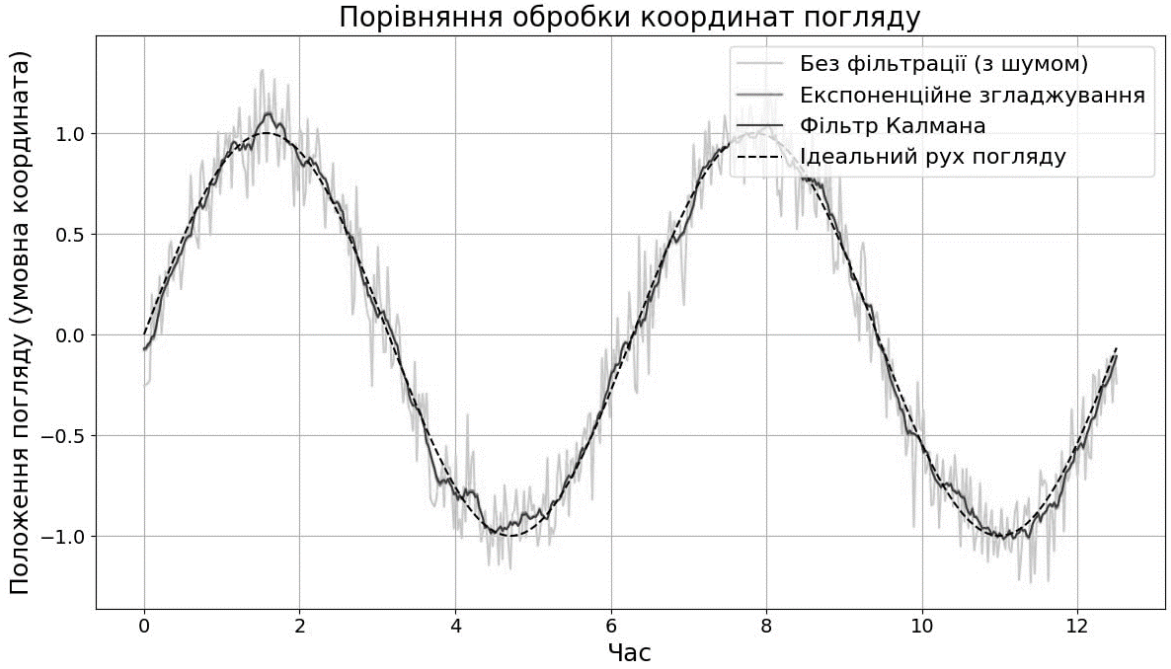


Рис 2. Порівняння ефективності фільтрації координат погляду за різними алгоритмами

Порівняльні характеристики методів фільтрації координат погляду наведено у табл. 1.

Таблиця 1

Порівняльні характеристики методів фільтрації координат погляду

Режим обробки координат	Середнє відхилення (пікселі)	Візуальна стабільність	Реакція на зміну погляду	Переваги	Недоліки
Без фільтрації	≈ 20	Низька	Швидка, але нестабільна	Висока швидкодія, проста реалізація	Тремтіння, стрибки координат
Експоненційне згладжування	≈ 12	Середня	Помірна	Простота реалізації, зменшення шуму	Затримка при різких рухах
Фільтр Калмана	≈ 8	Висока	Швидка й стабільна	Найкраще згладжування, адаптація до шуму	Складність налаштування

Крім того, для демонстрації ефективності фільтрації координат погляду було змодельовано два режими роботи системи: без згладжування та з фільтрацією. В обох

випадках користувач нібито дивиться на однакові мішені, але результати відрізняються точністю.

На рис. 3 відображено порівняння ефективності передбачення координат погляду на точках на екрані з фільтрацією та без неї та прийнято такі позначення:

– кружечок – позиції мішеней на екрані;

– хрестик – передбачені координати погляду без фільтрації);

– трикутник – координати після застосування фільтру (наприклад, експоненційного згладжування або фільтра Калмана);

– пунктирні лінії показують відхилення між gaze-точками і мішенями.

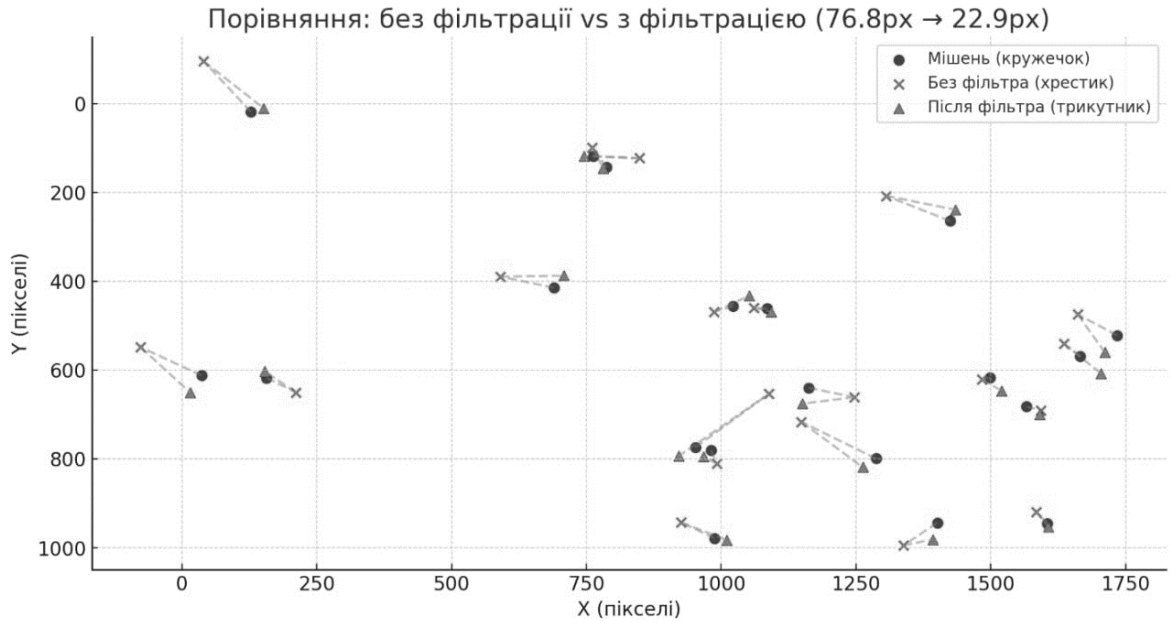


Рис 3. Порівняння ефективності передбачення координат погляду на точках на екрані з фільтрацією та без фільтрації

У результаті застосування фільтрації середня похибка зменшилась з приблизно 80 пікселів до 10, що свідчить про значне підвищення стабільності системи. Таким чином, навіть проста фільтрація помітно покращує комфорт користування системою на практиці.

На рис. 4 зображено тривимірну реконструкцію сцени, яка демонструє, як система прогнозує напрямок погляду користувача. Модель обличчя створено на основі координат ключових точок, отриманих за допомогою алгоритму MediaPipe FaceMesh. Ці точки формують базову сітку обличчя, яка дозволяє системі орієнтуватись у просторовій конфігурації голови.

З центру області очей побудовано вектор, який репрезентує прогнозований напрямок погляду. Цей вектор формується шляхом перетворення пари значень, отриманих з виходу нейронної мережі – кутів повороту погляду у сферичних координатах: pitch (нахил вгору/вниз) та yaw (нахил вліво/вправо). Кути перетворюються у тривимірний вектор за допомогою просторової тригонометрії.

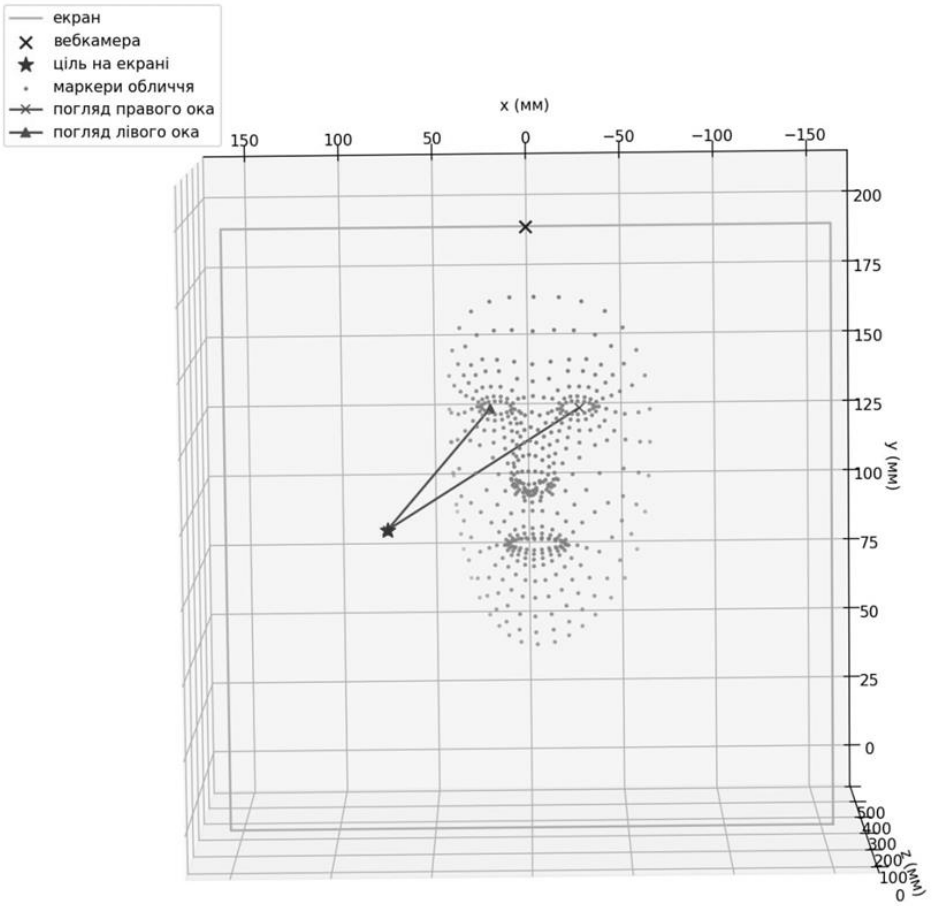


Рис. 4. Тривимірне зображення користувача з вектором погляду та точкою його перетину з площиною екрана

Площину екрана задано в системі координат камери – зазвичай вона розміщується перпендикулярно до осі Z на визначеній відстані. Розраховується точка перетину вектора погляду з цією площиною, яка і є прогнозованою точкою фіксації. У візуалізації ця точка може бути позначена кольоровою міткою (наприклад, червоною), що дозволяє візуально оцінити, куди, на думку системи, спрямовано погляд користувача.

Такий тип тривимірної візуалізації є корисним інструментом не лише для оцінки якості прогнозу, а й для калібрування та виявлення потенційних помилок у розрахунках – наприклад, коли напрямок погляду не збігається з положенням мішені. Це також дозволяє розробнику контролювати узгодженість між геометрією обличчя, положенням голови та вектором погляду.

Отримані результати підтверджують доцільність використання фільтрації при роботі з координатами погляду, особливо у випадках високочастотних шумів. Крім того, результати показують, що навіть без повної реалізації складних моделей у коді, теоретичне моделювання дає можливість оцінити ефективність обраного підходу та обґрунтувати його застосування в реальній системі.

6. Обговорення результатів дослідження

Результати дослідження показали покращену стабільність і точність керування курсором за допомогою відстеження погляду. На відміну від попередніх підходів, які здебільшого зосереджувались на визначенні координат погляду, у цьому дослідженні застосовано методи фільтрації шуму та прогнозування руху очей. Це дозволяє зменшити

випадкові коливання і забезпечити плавніше керування. Такий підхід покращує взаємодію користувача із системою і зменшує втому очей при тривалому використанні.

Серед основних переваг цієї системи — підвищена точність позиціонування курсора та зниження навантаження на процесор завдяки оптимізованим алгоритмам обробки даних. Інтеграція механізму підтвердження команд, наприклад подвійного кліку, також спрощує взаємодію з системою, дозволяючи обійтись без додаткових пристроїв введення. Важливою особливістю є можливість адаптації алгоритмів до індивідуальних потреб користувачів, що розширює потенційні сфери застосування.

Проте, окрім переваг, система має певні обмеження. Її ефективність залежить від якості відеопотоку та умов освітлення. Незважаючи на алгоритми компенсації, різкі зміни освітлення можуть вплинути на точність визначення погляду. Крім того, система може виявляти деяку нестабільність при дуже швидких рухах очей, що може спричиняти незначні затримки в переміщенні курсора.

Подальші дослідження можуть зосередитись на покращенні адаптивності алгоритмів, зокрема за допомогою методів глибокого навчання для прогнозування руху очей в реальному часі. Окрім цього, є потенціал для розширення функціональності системи через інтеграцію жестового управління або комбінованих методів введення. Можна також дослідити ефективність цього підходу у різних сценаріях, таких як управління віртуальними середовищами або підтримка людей з обмеженими можливостями. Отже, отримані результати відкривають можливості для подальших удосконалень, які зроблять систему ще точнішою, швидшою та зручнішою в експлуатації.

7. Висновки

У процесі дослідження було послідовно вирішено поставлені задачі, спрямовані на реалізацію системи керування курсором за допомогою погляду.

Визначено модульну структуру системи керування курсором за поглядом з окремими компонентами для відеозахоплення, обробки зображень, прогнозування погляду, обчислення координат та виводу результатів. Вибрано відповідні інструменти, такі як MediaPipe, PyTorch, OpenCV.

Реалізовано ключові компоненти системи керування курсором за поглядом:

– реалізовано нейронну мережу на основі згорткових шарів і повнозв’язних блоків для прогнозування напрямку погляду у вигляді кутів pitch та yaw, що забезпечило базову оцінку фокусу користувача;

– впроваджено процес нормалізації вхідних зображень очей та обличчя, включаючи вирівнювання по положенню голови, що підвищило точність роботи моделі;

– розроблено алгоритм перетворення кутових координат у просторовий вектор з подальшою проєкцією на площину екрана, що дозволяє адаптувати систему до різного положення камери та розміру дисплея;

– досліджено та реалізовано два методи фільтрації координат погляду – ковзне середнє та фільтр Калмана, що дозволило суттєво зменшити випадкові коливання та підвищити стабільність положення курсора;

– запропоновано механізм підтвердження введення команд (подвійне кліпання або фіксація погляду), що забезпечив зручність взаємодії з користувачем без використання фізичних пристроїв введення.

Проведено тестування системи в умовах реального використання. Отримані результати засвідчили прийнятний рівень затримки (до 50 мс), високу адаптивність до різних користувачів і можливість точного керування курсором без апаратних модифікацій.

Таким чином, усі задачі дослідження виконано. Отримана система є програмно гнучкою, не залежить від спеціального обладнання та може бути легко модифікована.

Подальші дослідження можуть включати вдосконалення прогнозу траєкторії погляду, врахування жестів голови або інтеграцію з розширеними інтерфейсами користувача.

Перелік посилань:

1. Zhang Z., Chen Y., Li L. Gaze Tracking Techniques and Applications. Springer, 2019. 320 p.
2. Hennadiy F., Zakharchenko A. Machine Learning Algorithms for Eye Gaze Detection. *Journal of Computational Intelligence*. 2021. Vol. 16, No. 3. P. 189–202.
3. Viola P., Jones M. Rapid Object Detection using a Boosted Cascade of Simple Features. CVPR, 2001. P. 511–518.
4. Cheng Y., Liu W., Lv F. Real-Time Gaze Estimation Based on CNN and Facial Landmark Detection Sensors. 2020. Vol. 20, No. 4. P. 1–15.
5. Welch G., Bishop G. An Introduction to the Kalman Filter. UNC Chapel Hill. 1995. 16 p
6. Васильєва І., Коваленко Ю. Техніки фільтрації шуму в системах відстеження погляду. Харків: Наукова думка, 2020. 192 с..
7. Papoutsaki A., Sadeh N., Huang J., Lasecki W.S., Bigham J.P. WebGazer: Scalable Webcam Eye Tracking Using User Interactions. IJHCS. 2016. Vol. 98. P. 102–113.
8. Tobii Eye Tracking. URL: https://www.tobii.com/ (дата звернення: 26.04.2025).
9. Медіапайп (MediaPipe). Розробка моделей для відстеження обличчя. URL: https://google.github.io/mediapipe/solutions/face_mesh (дата звернення: 26.04.2025).
10. Мартин Х., Джонс Р. Комп'ютерний зір та глибинне навчання для відстеження погляду. Київ: Наукова думка, 2021. 312 с.
11. Офіційний сайт OpenCV: Бібліотека для комп’ютерного зору та обробки зображень. URL: https://opencv.org/ (дата звернення: 26.04.2025).
12. Офіційний сайт PyTorch: Бібліотека глибинного навчання. URL: https://pytorch.org/ (дата звернення: 26.04.2025).

Надійшла до редколегії 10.05.2025 р.

Ситнікова Поліна Едуардівна, кандидат технічних наук, доцент, доцент кафедри системотехніки ХНУРЕ, м. Харків, Україна, e-mail: polina.sytnikova@nure.ua, ORCID: https://orcid.org/0000-0002-6688-4641

Грицай Микита Олегович, здобувач вищої освіти, група СПРм-23-1, факультет комп’ютерних наук, ХНУРЕ, м. Харків, Україна, e-mail: mykyta.hrytsai@nure.ua

УДК 004.8:004.9 DOI: 10.30837/0135-1710.2025.185.034

С.Ф. ЧАЛИЙ, І.О. ЛЕЩИНСЬКА

ПОБУДОВА ФУНКЦІОНАЛЬНО-ТЕМПОРАЛЬНОГО ПРЕДСТАВЛЕННЯ МЕНТАЛЬНОЇ МОДЕЛІ РІШЕННЯ В ЗАДАЧІ ФОРМУВАННЯ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ

Об'єктом дослідження є процес побудови ментальної моделі рішення для користувача інтелектуальної інформаційної системи. Метою є розробка підходу до побудови функціонально-темпорального представлення ментальної моделі рішення для користувача інтелектуальної інформаційної системи з тим, щоб забезпечити можливість формування й уточнення пояснень для користувача як з функціональними можливостями рішення такої системи, так і зі сценаріями його використання. Розроблено функціонально-темпоральне представлення ментальної моделі, яке інтегрує структуровану множину властивостей рішення у функціональному аспекті та сценарії використання рішення у темпоральному аспекті. Запропоновано метод побудови функціонально-темпорального представлення ментальної моделі, який містить етапи формування структурованого набору вхідних даних, відбору релевантних властивостей, інтеграції функціональних та темпоральних характеристик та формування доповненого представлення. Метод орієнтований на підвищення ефективності формування пояснень на основі врахування як поточних властивостей рішення, так і можливостей його подальшого використання.

1. Вступ

Сучасні інтелектуальні інформаційні системи (ІІС) використовують моделі, методи та алгоритми машинного навчання для формування рішень, які забезпечують підтримку діяльності користувачів у банківській сфері, медицині, освіті тощо [1]. Однак, внаслідок використання складних непрозорих алгоритмів машинного навчання такі системи розглядаються користувачами як «чорні скриньки», що може призводити до зниження довіри до рішень ІІС та обмежувати ефективність практичного застосування таких рішень [2]. Для того, щоб зробити прозорим процес і причини формування рішень в таких системах, використовуються методи пояснюваного штучного інтелекту (Explainable AI, XAI) [3]. Отримані пояснення мають бути зрозумілими користувачам, тобто мають узгоджуватись з ментальними моделями (ММ) рішень у цих користувачів. ММ в області когнітивної психології розглядаються як представлення зовнішньої реальності в свідомості людини [4]. ММ рішень ІІС є представленням користувача щодо властивостей та сценаріїв використання цих рішень [5]. Властивості рішень становлять функціональний аспект ММ, а сценарії використання є відображенням її темпорального аспекту. Отже, ММ відображають переваги рішень ІІС для користувача та обмеження щодо застосування цих рішень при вирішенні практичних задач користувача. Таким чином, проблема побудови ММ рішень ІІС в задачі побудови пояснень з урахуванням її функціонального та темпорального аспектів є актуальною.

2. Аналіз літературних даних і постановка проблеми дослідження

Дослідження щодо пояснювального штучного інтелекту були започатковані агентством DARPA із метою створення методів, які дають можливість розуміти причини рішень ІІС [6]. Дослідження щодо XAI проводяться в двох основних напрямках: розробка методів побудови пояснень; розробка ММ рішень для користувачів [7].

Методи побудови пояснень основані на визначенні впливу вхідних даних на рішення з використанням: локальної інтерпретованої моделі (метод LIME) [8]; теорії ігор для розподілу важливості кожної ознаки (метод SHAP) [9]; пошарового розподілу від виходу моделі до її вхідних даних через зворотнє поширення релевантності (метод LRP) [10];

моделювання причинно-наслідкових зав'язків процесу формування рішення на основі темпоральних залежностей [11]; каузальних залежностей між вхідними та проміжними станами ІІС, а також отриманим рішенням [12]; залежностей на локальному рівні представлення [13]. Проте результуючі пояснення в рамках вирішення другої задачі XAI мають бути узгоджені з ММ рішень у користувача.

Термін «ментальна модель» вперше було запропоновано в [14], з метою відобразити моделі реальності, які людина використовує для прогнозування наслідків своїх дій. Теорія ММ міркування розроблена в [15] на основі припущення, що міркування людини базується не на використанні логічних правил, а на маніпулюванні ММ можливих рішень, причому кожна модель зберігає структуру рішення. Важливу роль ММ у процесі взаємодії людини з інформаційними системами обґрунтовано в дослідженнях з проектування інтуїтивно зрозумілих інтерфейсів [16]. Особливості представлення ММ в задачі побудови пояснень в інтелектуальних системах розглянуто в [5]. Запропонована концептуальна ММ забезпечує реалізацію комбінованого підходу до побудови пояснень. З урахуванням особливостей функціонування ІІС, принципи побудови ММ були розширені принципом доповнення вхідних даних для врахування обмежень щодо використання рішень інтелектуальних систем [17]. Традиційні методи побудови ММ дають можливість відобразити структуру [17] або каузальні залежності у рішеннях ІІС [18]. Метод [19] базується на моделюванні командної роботи людей і ІІС з тим, щоб забезпечити спільне розуміння задач та рішень системи. Аналіз методів побудови ММ з позицій XAI дає можливість зробити висновок, що існуючі підходи орієнтовані або на відображення у ММ загальної структури рішення, або ж на формування залежностей, що відображають сценарій використання рішення. Зазначене свідчить про актуальність вирішення задачі побудови функціонально-темпорального представлення ММ рішення ІІС з тим, щоб забезпечити можливість відображення у поясненні як властивостей рішення, так і можливостей його використання.

3. Мета і задачі дослідження

Метою даного дослідження є розробка підходу до побудови функціонально-темпорального представлення ММ рішення для користувача ІІС з тим, щоб забезпечити можливість формування й уточнення пояснень для користувача згідно як з функціональними можливостями рішення такої системи, так і сценаріїв його використання, представлених у ММ.

Для досягнення поставленої мети під час дослідження вирішувалися такі задачі: розробка функціонально-темпорального представлення ММ користувача ІІС; розробка моделі процесу побудови ММ користувача ІІС з урахуванням функціональних особливостей рішення такої системи та сценаріїв використання цих рішень.

4. Функціонально-темпоральне представлення ментальної моделі рішення для користувача інтелектуальної інформаційної системи

Об'єктом дослідження є процес побудови ММ рішення для користувача ІІС.

Предметом дослідження є принципи та методи побудови ММ користувача ІІС.

Розроблене функціонально-темпоральне представлення базується на принципах побудови ММ: відповідності структури; множинності; неповноти вхідних даних; доповнення вхідних даних [12]-[14].

Принцип відповідності структури передбачає, що ММ та рішення ІІС мають аналогічну структуру. Відповідність структур досягається за рахунок використання теоретико-множинного опису ММ. У функціональному аспекті ММ відображає підмножину значень властивостей рішення ІІС, які обумовлюють контекст використання цього рішення. У темпоральному аспекті ММ відображає темпоральні відношення між

елементами рішення ІІС, що дає можливість сформувати опис процесу формування рішення, а також процесу використання рішення. Такий опис, в залежності від заданих темпоральних відношень, може бути сформований на різних рівнях деталізації, відображаючи роботу ІІС як «білої скриньки», «сірої скриньки» або «чорної скриньки». Наприклад, ММ модель телефону у рекомендаційній системі у функціональному аспекті відображає властивості товару (процесор, пам'ять, камера тощо). У темпоральному аспекті ММ відображає сценарії використання телефону (професійний, для фотомистецтва, ігровий, музичний тощо).

Згідно з даним принципом, кожне *n*-те рішення P_n із множини можливих рішень ІІС P, $\mathrm{P}_n \in \mathrm{P}$, задається множиною $\left\{v_{i,j}^n\right\}$ *j*-их значень *i*-их атрибутів $a_i \in A$, причому кожен атрибут a_i має кінцеву множину значень $V_i = \left\{v_{i,j}\right\}$:

$$P_n = \left\{\left\{v_{i,j}^n\right\}, R : (\forall i)\exists V_i, R \subset A \times A\right\}. \tag{1}$$

На кожній парі атрибутів $(a_i, a_m) \in R$ може бути визначено відношення між цими атрибутами. Ці відношення можуть відображати функціональність рішення, а також процес використання рішення.

Тоді функціональна компонента визначається набором значень властивостей $\left\{v_{i,j}^n\right\}$ та функціональними відношенням між цими властивостями:

$$\begin{gathered} P_n^{functional} = \\ = \left\{\left\{v_{i,j}^n\right\}, R^{functional} : (\forall i, m)\exists\left(a_i, a_m, f_m^i\right) : a_i, a_m \in A,\ f_m^i \in R^{functional}\right\}. \end{gathered} \tag{2}$$

Функціональні залежності $R^{functional}$ містять відношення з типом (роллю), з функціональною залежністю, а також обмеження.

Темпоральна компонента задає темпоральні відношення між властивостями рішення. Темпоральні відношення задаються темпоральними правилами типів «Next» та «Future». Перше правило задає безпосередню послідовність встановлення (використання) значень властивостей a_i та a_m у часі, а друге – об'єднує декілька послідовних пар (a_i, a_m), (a_m, a_k) :

$$P_n^{Temporal} = \left\{(a_i, a_m), t_m^i : (\forall i, m) t_m^i \in \left\{Next, Future\right\}, a_i, a_m \in A\right\}. \tag{3}$$

Сукупність темпоральних правил t_m^i визначає процес формування або використання рішення Π_n:

$$\Pi_n = \left\{t_2^1, t_3^2, t_2^1, ..., t_{\left|P^{Temporal}\right|}^{\left|P^{Temporal}-1\right|}\right\}. \tag{4}$$

Тоді ММ у функціональному та темпоральному аспектах, згідно з принципом відповідності структури, набуває вигляду:

$$M_n = \left\{P_n^{functional}, P^{Temporal} : (\forall n)\exists\Pi_n\right\}. \tag{5}$$

Принцип множинності ММ забезпечує формування набору ММ, що мають аналогічний набір властивостей але відмінні значення цих властивостей. Тобто вони мають забезпечити такі ж самі сценарії використання, але з урахуванням функціональних можливостей, що відрізняються. Таке множинне представлення дає можливість порівняти схожі рішення ІІС шляхом порівняння відповідних ММ користувача. ММ в даному випадку враховують не лише документально зафіксовані властивості рішення, але й особливості використання рішення ІІС іншими користувачами. Даний принцип, наприклад, забезпечує можливість порівняння двох смартфонів різних торгових марок зі схожими характеристиками. Тобто користувач може порівняти на рівні ММ цікаві йому сценарії використання конкуруючих смартфонів з урахуванням поточних значень роздільної здатності камери, частоти процесора, наявності стилуса тощо.

Згідно з даним принципом, множина ММ користувача щодо рішення ІІС складається з k-их ММ $M_{n,k}$, що відображають альтернативні варіанти $P_{n,k}$ n-го рішення :

$$M = \left\{ M_{n,k} \middle| \exists P_{n,k} \right\}. \tag{6}$$

Принцип неповноти вхідних даних відображає особливість людської психіки, яка полягає в тому, що користувач, як правило, бажає враховувати лише цікаві йому властивості рішення ІІС та ігнорувати можливі недоліки цього рішення. Тобто ММ містить лише підмножину властивостей рішення і якщо користувач не включає до цієї підмножини, зокрема, обмеження щодо використання рішення, то ММ стає упередженою. Наявність упередженості у ММ приводить до її нерелевантності та, як наслідок, помилок в оцінці та використанні рішень ІІС. Наприклад, при виборі смартфона користувач може звертати увагу на потужний процесор та яскравий екран з високою роздільною здатністю, але не враховувати обмежену місткість акумулятора. Результатом є скорочення часу автономної роботи смартфона, що може нівелювати його технічні переваги.

Згідно з даним принципом, множина атрибутів рішення A розбивається на підмножини важливих для користувача A^+ та неважливих властивостей A^- відповідно:

$$A = A^+ \bigcup A^- \left| A^+ \bigcap A^- = \varnothing. \right. \tag{7}$$

Модель M_n у відповідності до даного принципу набуває вигляду M_n^+:

$$M_n^+ = \left\{ P_n^{functional}, P^{Temporal} : (\forall i) a_i \in A^+ \right\}. \tag{8}$$

Принцип доповнення вхідних даних полягає у доповнення ММ обмеженнями щодо процесу використання рішення ІІС. Такі обмеження формуються на основі значень даних, які користувач вважає нерелевантними і які не були використані при первинній побудові ММ. Такі обмеження можуть бути отримані на основі даних щодо практики використання рішення.

Згідно з даним принципом, доповнення ММ M_n^- містить обмеження щодо властивостей рішення, які не були враховані у (8):

$$M_n^- = \left\{ P_n^{functional}, P^{Temporal} \middle| (\forall i) a_i \in A^- \right\}. \tag{9}$$

У підсумку множина ММ користувача ІІС у функціонально-темпоральному представленні набуває вигляду:

$$M = \left\{ M_{n,k} : (\forall k) M_{n,k}^{+} \bigcup M_{n,k}^{-} \left| \exists P_{n,k} \right. \right\}. \quad (10)$$

Таким чином, розроблене функціонально-темпоральне представлення ММ відповідає принципам побудови ММ та дає можливість відобразити як функціональність рішення, так і сценарії використання даного рішення.

5. Метод побудови функціонально-темпорального представлення ментальної моделі рішення для користувача інтелектуальної інформаційної системи

Процес формування ММ базується на взаємодії розглянутих в попередньому розділі принципах формування ММ. Зв'язок між цими принципами у контексті формування ММ наведено на рис. 1.

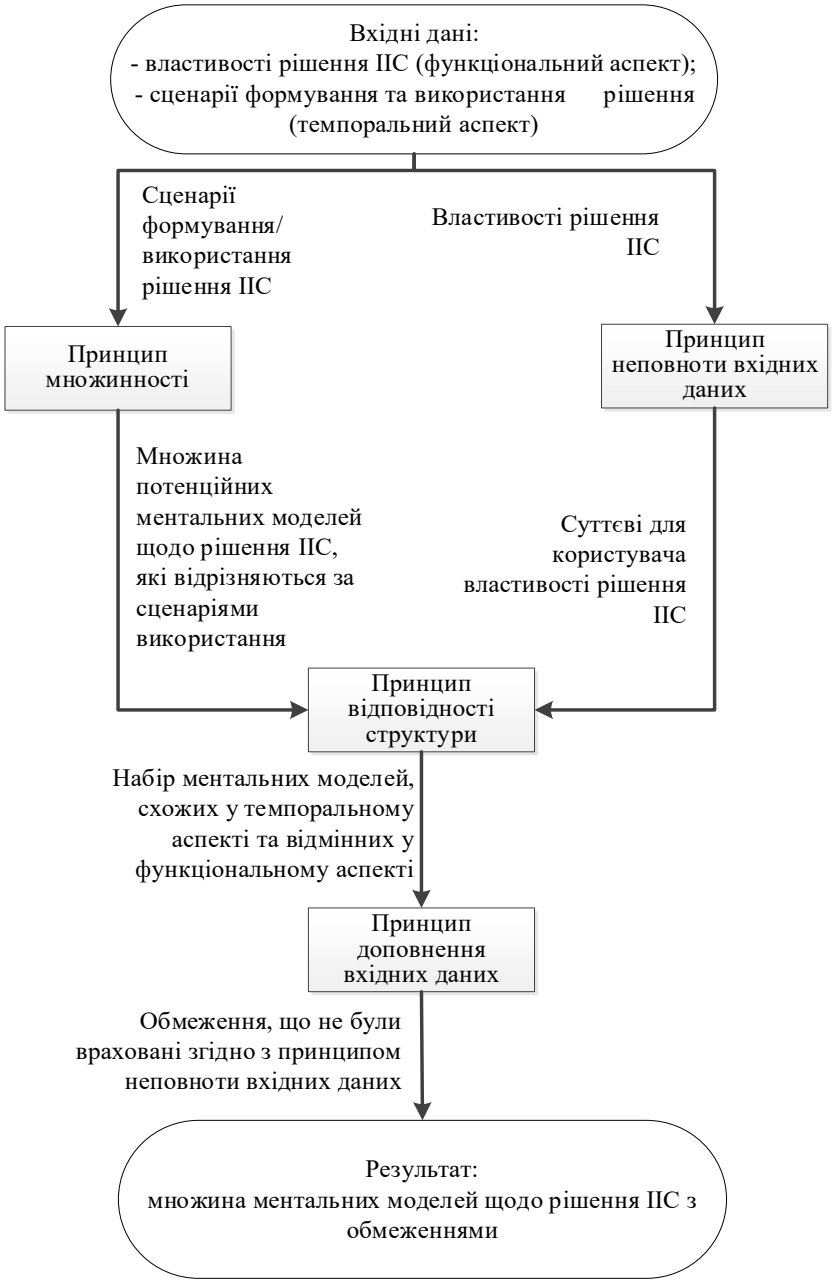


Рис. 1. Взаємодія принципів побудови ментальної моделі

Процес формування ММ використовує як вхідні дані властивості рішення і сценарії формування (використання) рішення ІІС. Згідно з принципом множинності, формується набір потенційних ММ рішення для користувача. Тобто формується «пуста» множина ММ, які відображають різні варіанти використання рішення ІІС. Згідно з принципом неповноти вхідних даних, формується набір суттєвих для користувача властивостей рішення.

За принципом відповідності структури рішення та ММ, потенційні ММ поєднуються із суттєвими властивостями рішення, що дає можливість сформувати набір альтернативних для користувача варіантів рішення. Альтернативність у даному контексті відображає різні можливості використання рішення на базі можливих варіантів властивості рішення (наприклад, якщо мова йде про схожі варіанти смартфонів з різними камерами, наявністю/відсутністю інтелектуальних можливостей тощо).

Згідно з принципом доповнення вхідних даних, кожна із ММ уточняється з використанням обмежень, які представлені окремими властивостями та їх значеннями. Наприклад, властивість «SIM-карта» може мати значення {«можливість підключення двох SIM-карт», «одна SIM-карта»}. Значення «одна SIM-карта» може виступати як обмеження, яке не було враховано користувачем згідно з принципом неповноти вхідних даних.

Метод побудови функціонально-темпорального представлення ММ рішення у задачі побудови пояснень в ІІС базується на представлених принципах побудови пояснень і містить такі етапи.

Етап 1. Формування структурованого набору вхідних даних.

Крок 1.1. Структурування властивостей вхідного об’єкта.

Крок 1.2. Виділення та оцінка набору позитивних властивостей згідно з (7).

Етап 2. Формування переліку потенційних ММ (8) за процесом використання (4).

Етап 3. Відбір релевантних властивостей рішення, що належать до A^+.

Крок 3.1. Вибір порогових значень для оцінки властивостей.

Крок 3.2. Оцінка та класифікація релевантності властивостей.

Етап 4. Побудова ММ (8) на основі інтеграції функціональних (2) та темпоральних (3) характеристик.

Етап 5. Агрегація властивостей ММ за сценаріям використання рішення (4).

Етап 6. Формування доповненого представлення ММ (10).

Результатом використання методу є ММ рішення у користувача, яка інтегрує властивості та сценарії використання рішення, що дає можливість оцінити релевантність пояснення. Наприклад, пояснення щодо характеристик смартфона, що був запропонований рекомендаційною системою, має відповідати сценаріям його використання (робота, розваги тощо). Тобто ММ формує обмеження для вибору релевантного пояснення щодо рішення ІІС.

6. Експериментальна перевірка методу побудови функціонально-темпорального представлення ментальної моделі рішення

Експериментальну перевірку розробленого методу було орієнтовано на реалізацію наведених чотирьох принципів побудови ІІС для функціонально-темпорального представлення предметної області. Як вхідні дані було використано інформацію щодо преміум-телефону на платформі електронної комерції:

– структуровану інформацію про характеристики телефону, представлену в текстовому вигляді;

– відгуки користувачів (135 відгуків), які підкреслюють позитивні та негативні властивості телефону за результатами його використання.

Фрагмент інформації про характеристики телефону наведено на рис. 2.

Приклад відгуку користувача з оцінкою характеристик телефону наведено на рис. 3. Ключові результати експерименту по етапах методу наведено в табл. 1.

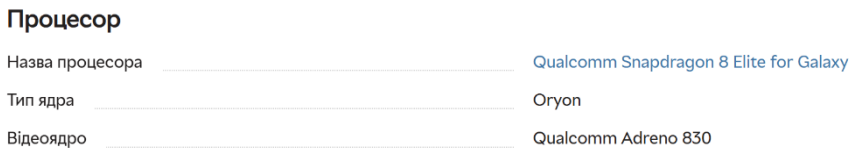
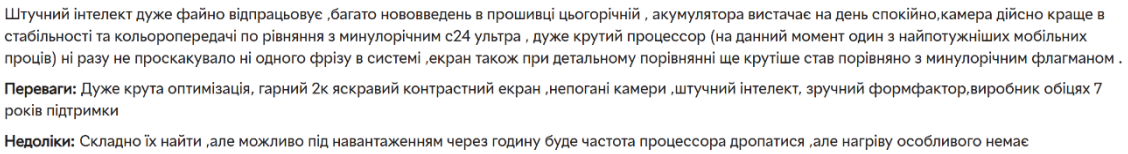


Рис. 2. Структурована інформація про характеристики вхідних даних


Штучний інтелект дуже файно відпрацьовує ,багато нововведень в прошивці цьогорічній , акумулятора вистачає на день спокійно,камера дійсно краще в стабільності та кольоропередачі по рівняння з минулорічним с24 ультра , дуже крутий процессор (на данний момент один з найпотужніших мобільних проців) ні разу не проскакувало ні одного фрізу в системі ,екран також при детальному порівнянні ще крутіше став порівняно з минулорічним флагманом .

Переваги: Дуже крута оптимізація, гарний 2к яскравий контрастний екран ,непогані камери ,штучний інтелект, зручний формфактор,виробник обіцяє 7 років підтримки

Недоліки: Складно їх найти ,але можливо під навантаженням через годину буде частота процессора дропатися ,але нагріву особливого немає

Рис. 3. Приклад відгуку з описом характеристик телефона

Таблиця 1

Поетапні результати використання методу

Етап	Результати етапу
1. Формування структурованого набору вхідних даних	**Властивості**: процесор, камера, дисплей, батарея, дизайн, наявність штучного інтелекту, ціна. **Позитивні:** швидкість (0,35 – 47 згадувань), камера (0,31 – 42), дисплей (0,29 – 39)
2. Формування переліку потенційних ментальних моделей	1.Професійне використання, робота (процесор, пам'ять, камера). 2. Звичайні користувачі, розваги (відношення ціна-якість, простота використання). 3. Ентузіасти нової техніки (наявність штучного інтелекту).
3. Відбір релевантних властивостей рішення	Високий рівень релевантності: процесор, камера, дисплей. Релевантні властивості: дизайн, наявність штучного інтелекту. Виключаються: колір, вага.
4. Побудова ментальних моделей	Функціональні властивості (контекст використання): робота (процесор, камера, батарея); розваги (камера, штучний інтелект). Темпоральні властивості (еволюція вимог до телефону на основі зміни сценарії використання).
5. Агрегація властивостей ментальних моделей	Інтегральна оцінка за сценарієм використання: робота – 0,89; розваги – 0,73. Розрахунок на основі нормованої оцінки за сумарною кількістю згадувань.
6. Формування доповненого представлення ментальних моделей	**Негативні властивості:** ціна (0,1 – 14 згадувань), батарея (0,08 – 11), нагрівання (0,09 – 9)

Оцінка властивостей є нормованою відносно кількості відгуків користувачів. Результуюча модель містить набори властивостей рішення, в тому числі обмеження, об’єднані за сценарієм використання.

7. Обговорення результатів

Розроблені функціональне-темпоральне представлення ММ рішення для користувача та метод його побудови забезпечують можливість сформувати множину таких ММ рішення для користувача, що у функціональному аспекті відображають властивості рішення, а у темпоральному – сценарії використання рішення.

Функціонально-темпоральне представлення ММ рішення відрізняється від існуючих інтеграцією можливостей рішення з контекстом його використання, що створює умови для побудови та відбору пояснень згідно як з поточними практичними потребами користувача, так і з еволюцією цих потреб.

Метод побудови функціонально-темпорального представлення ММ рішення для користувача ІІС містить етапи відбору релевантних властивостей рішення, побудови ММ, які відображають функції рішення та спосіб його використання, а також формування доповненого представлення, яке містить обмеження на використання рішення, що дає можливість обґрунтувати вибір релевантного пояснення щодо такого рішення.

Експериментальна перевірка показала, що обмеження розробленого підходу пов'язані із характеристиками вхідних даних, оскільки інформація про позитивні та негативні властивості рішення зазвичай представлена у неструктурованому текстовому форматі, що потребує використання інтелектуальних обробників вхідного тексту.

Подальші дослідження щодо побудови ММ рекомендовано орієнтувати на розробку графової нейронної мережі з блоками структуризації вхідного тексту і механізмом уваги для побудови зваженої оцінки сценаріїв в ММ.

8. Висновки

За результатами дослідження запропоновано функціонально-темпоральне представлення ММ рішення для користувача ІІС, яке містить структуровану множину властивостей рішення у функціональному аспекті, а також сценарії використання рішення у темпоральному аспекті, що дає можливість підвищити ефективність формування пояснень на основі врахування як поточних властивостей рішення, так і сценарію його подальшого використання.

Розроблено метод побудови функціонально-темпорального представлення ММ рішення для користувача ІІС, який містить етапи формування структурованого набору вхідних даних, формування переліку потенційних ММ за процесом використання, відбору релевантних властивостей рішення, побудови ММ на основі інтеграції функціональних та темпоральних характеристик рішення, агрегації властивостей ММ за сценарієм використання рішення, формування доповненого представлення ММ. Метод враховує як позитивні для користувача властивості рішення, так і обмеження на сценарії його використання, що дає можливість обґрунтовано, у відповідності до ММ, відібрати для користувача релевантне пояснення щодо отриманого рішення.

Перелік посилань:

1. Kordon, A. (2016). Intelligent systems in industry. In V. Sgurev, R. Yager, J. Kacprzyk, & V. Jotsov (Eds.), Innovative issues in intelligent systems (pp. 1-31). Springer.
2. Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 267, 1-38. https://doi.org/10.1016/j.artint.2018.07.007
3. Gunning, D., & Aha, D. (2019). DARPA's explainable artificial intelligence (XAI) program. AI Magazine, 40(2), 44-58. https://doi.org/10.1609/aimag.v40i2.2850
4. Johnson-Laird, P. N. (1983). Mental models: Towards a cognitive science of language, inference, and consciousness. Harvard University Press.
5. Чалий, С., & Лещинська, І. (2023). Концептуальна ментальна модель пояснення в системі штучного інтелекту. Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології, 1(9), 70-75. https://doi.org/10.20998/2079-0023.2023.01

6. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). IEEE Access, 6, 52138-52160. https://doi.org/10.1109/ACCESS.2018.2870052.

7. Frasca, M., La Torre, D., Pravettoni, G., Manzoni, G. M., & Caputo, A. (2024). Explainable and interpretable artificial intelligence in medicine: A systematic bibliometric review. Discovery Artificial Intelligence, 4, 15. https://doi.org/10.1007/s44163-024-00114-7

8. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765-4774.

9. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765-4774.

10. Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K. R. (2022). Explaining deep neural networks and beyond: A review of methods and applications. Proceedings of the IEEE, 109(3), 247-278.

11. Чалий С. Ф., Лещинський В. О. (2024). Темпорально-каузальні методи побудови пояснень в системах штучного інтелекту. АСУ та прилади автоматики, 181, 91-99. https://doi.org/10.30837/0135-1710.2024.181.091

12. Чалий, С., Лещинський, В., & Лещинська, І. (2022). Каузальна модель процесу побудови пояснень в інформаційній системі. Системи управління, навігації та зв'язку, 3(69), 99-103. https://doi.org/10.26906/SUNZ.2022.3.099

13. Чалий С. Ф., Лещинський В. О. (2024). Побудова пояснень на локальному рівні представлення процесу формування рішень для внутрішніх користувачів інтелектуальної системи. Біоніка інтелекту, 1 (100), 58-64. https://doi.org/10.30837/ bi.2024.1(100).08.

14. Craik, K. (1943). The nature of explanation. Cambridge University Press.

15. Johnson-Laird, P. N., & Byrne, R. M. J. (1991). Deduction. Lawrence Erlbaum Associates

16. Norman, D. A. (2014). Some Observations on Mental Models. In D. Gentner & A. L. Stevens (Eds.), Mental Models (pp. 7-14). Psychology Press.

17. Чалий, С. Ф., & Лещинська, І. О. (2024). Уточнення ментальної моделі рішення на основі доповнення вхідних даних в задачі формування пояснень в інтелектуальній системі. АСУ та прилади автоматики, 1(182), 66–72. https://doi.org/10.30837/0135-1710.2024.182.066

18. Gentner, D., & Stevens, A. L. (2014). Mental models. Psychology Press.

19. Spirtes, P., Glymour, C., & Scheines, R. (2000). Causation, prediction, and search (2nd ed.). MIT Press.

20. Andrews, R. W., Lilly, J., Srivastava, D., & Feigh, K. M. (2022). The role of shared mental models in human-AI teams: A theoretical review. Theoretical Issues in Ergonomics Science, 24(2), 129-175.

Надійшла до редколегії 25.05.2025 р.

Чалий Сергій Федорович, доктор технічних наук, професор, професор кафедри інформаційних управляючих систем ХНУРЕ, м. Харків, Україна, e-mail: serhii.chalyi@nure.ua; ORCID: 0000-0002-9982-9091

Лещинська Ірина Олександрівна, кандидат технічних наук, доцент, доцент кафедри програмної інженерії ХНУРЕ, м. Харків, Україна, e-mail: iryna.leshchynska@nure.ua, ORCID: https://orcid.org/0000-0002-8737-4595

UDC 004.83 DOI: 10.30837/0135-1710.2025.185.043

A.R. KOVTUNENKO, S.V. MASHTALIR

A REVIEW OF MODERN NEURAL NETWORK ARCHITECTURES FOR IMAGE SEGMENTATION

The paper presents a chronological review and analysis of the evolution of image segmentation methods. It examines the development of architectures from early fully convolutional networks (FCN) to modern transformer zero-shot segmentation models such as Segment Anything Model (SAM). The key architectures, their innovations, technical features, advantages, and limitations were discussed in detail. The review allows understanding the main trends in the development of segmentation methods, evaluating the effectiveness of different approaches, and making an informed choice of architectures for solving practical problems depending on the requirements for accuracy, computational resources, and application area.

1. Introduction

Computer vision has been developing rapidly and continues to do so. Every year, existing algorithms are improved, radically new methods are created, and hybrid approaches are introduced that combine the strengths of different methods to solve previous or still unsolved problems. For example, segmentation has seen a significant transformation of paradigms and approaches in the last decade, from traditional heuristic methods to fully-connected convolutional methods and deeper transformer-based models that can handle multimodal data. Transformers were originally developed for natural language processing. They were later adapted for computer vision tasks due to a key advantage: they can identify correlations between disparate regions in images regardless of how far apart these regions are. This capability was difficult to achieve with convolutional networks, as their receptive field is limited to local neighborhoods that only gradually expand with increased network depth. Notably, this development has not been linear, and early methods have not been superseded; rather, many current approaches successfully combine classical methods with neural network methods, creating hybrid solutions that outperform their predecessors. However, despite significant progress, aspects such as finding object boundaries in complex scenes, finding objects of small size, domain adaptation, utilizing prior knowledge, and computational limitations still require further research and improvement.

2. Relevance and purpose of the article

The paper's relevance is due to the rapid increase of methods and approaches for different segmentation tasks and conditions, as well as the need to understand the strengths and weaknesses of existing methods. The systematization of existing approaches in chronological order will allow not only tracing the evolution of architectural solutions, but also understanding the main ideas and research directions, as well as limitations and unsolved problems that remain relevant in the present time. This will avoid the repetition of already known limitations and mistakes from past approaches, allowing for a more informed development of new models.

The purpose of this paper is to chronologically review and analyze segmentation models to identify the main innovations, limitations, and advantages, thus allowing an informed choice of methods and approaches for solving applied problems or designing new, more effective methods.

3. Review of the models

The evolution of neural network architectures for segmentation tasks is reasonable to start with the consideration of a full convolutional network (FCN) (Fig. 1) [1]. This architecture enables the processing of images at arbitrary resolution and obtaining a segmentation result that does not require additional post-processing. This method is an adaptation of the existing SOTA (State-of-the-Art) models (VGG, AlexNet, etc.) by replacing the fully connected layers with

convolutional ones and adding layers for dimensionality recovery. However, despite these advantages, this architecture showed unsatisfactory results for the segmentation of complex scenes, small objects, and boundary detection.

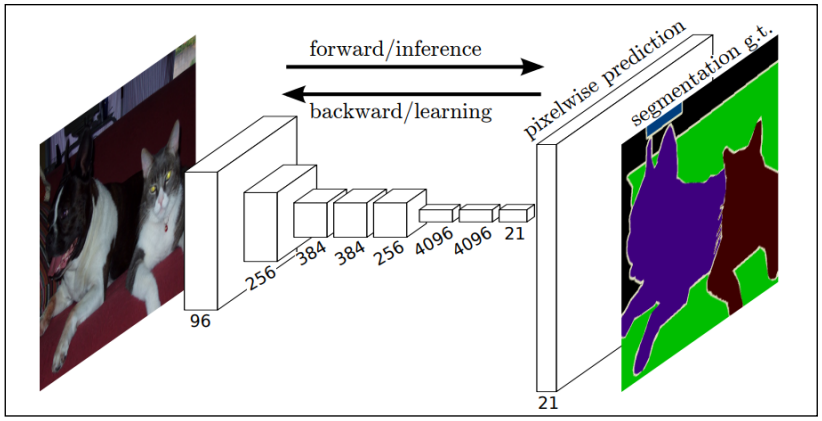


Fig. 1. FCN architecture [1]

The advent of the U-Net (Fig. 2) [2] architecture has solved some of the problems of FCN, especially in the context of handling fine details in images and boundary extraction. U-Net is also a fully convolutional neural network, which already has skip connections, allowing the model to retrieve information about objects from different levels of representations and recover their spatial information. The introduction of a weighted loss function, with increased coefficients for boundary pixels, allowed the accurate extraction of object boundaries. The architecture was originally developed for medical images and small amounts of raw data. Applying augmentation to the data allowed to achieve high recognition accuracy, but it should be noted that with insufficient variability of the data set, there is a risk of overfitting the model. The disadvantages of U-Net include computational cost, especially as the input image size increases, and the limited receptive field, which can make it difficult to analyze structures that require a larger context. To solve the problem of high-resolution image processing, the authors proposed an overlap-tile strategy.

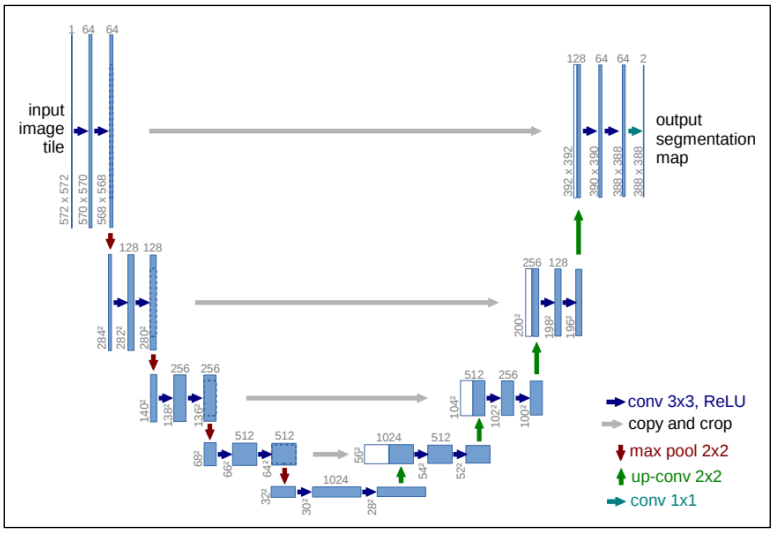


Fig. 2. U-Net architecture [2]

The SegNet (Fig. 3) [3] architecture, compared to U-Net, allows processing high-resolution

images with lower resource costs, while still finding object boundaries accurately enough. It has been specifically designed for systems requiring high-speed data processing, even in real time. SegNet is also a two-part encoder-decoder, but uses modified upsampling in the decoder module. The spatial resolution is restored using max pooling indices stored at the corresponding encoder levels during the max pooling operation. The feature maps themselves are not transferred, which significantly reduces the number of model parameters. Since skip connections are not applied, this can lead to partial loss of information about the global context of the image, which potentially reduces the effectiveness of segmentation in scenarios with complex spatial structures.

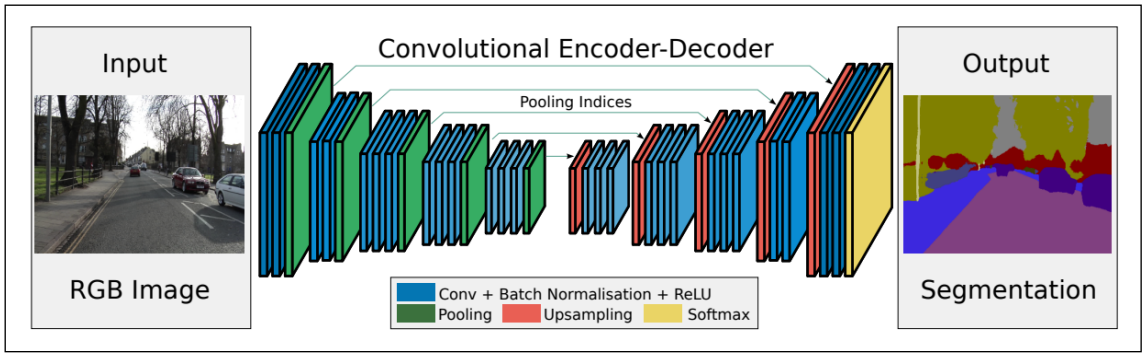


Fig. 3. SegNet architecture [3]

The DeepLab family of models (DeepLabV1 [4], DeepLabV2 [5], DeepLabV3 [6], DeepLabV3+ (Fig. 4) [7]) represents a sequential evolution of architectures aimed at improving the accuracy of semantic segmentation. Each iteration of the model has brought its additions and improvements. DeepLabV1 applied Atrous Convolutions, characterized by the introduction of specific gaps between convolutional kernel elements. This approach allows for increasing the receptive field without changing the feature map resolution. DeepLabV2 develops the concept of dilated convolutions by introducing Atrous Spatial Pyramid Pooling (ASPP) – a set of convolutions with different values of dilation parameters. This solution has significantly improved the quality of segmentation of objects of different scales in images. In DeepLabV3, the ASPP architecture was supplemented with a global pooling component. Its advantages are as follows: it takes the entire feature map and reduces it to a single vector, it captures image-level context that helps understand the overall scene, and it complements the local features captured by convolutional layers. Also, dilated convolutions were added to all parts of the model. DeepLabV3 introduces a «multi-grid» approach where different atrous rates are applied within consecutive blocks. This provides a hierarchical structure of receptive fields and shows significant improvement over using uniform rates. Despite the evolution of the DeepLab family, finding small or heavily occluded objects, or objects with rare views, is still a challenge for this architecture. In DeepLabV3+, the architecture has been changed to encoder-decoder, similar to U-Net. ResNet networks used in previous versions or a modified Xception architecture are used as an encoder-decoder in this architecture. Upsampling was made in two stages to improve granularity. At the first stage, decoder gets low-level features from the encoder and combines them with results from ASPP, which are preliminarily upsampled by a factor of four. The second stage is another 4X increase to get the final result. The authors experimented with a more complex decoder structure, including two levels of skip connections, but found no significant performance improvement over their simpler version. The complexity of the architecture improved the quality of segmentation, especially of object boundaries, but naturally led to an increase in computational cost.

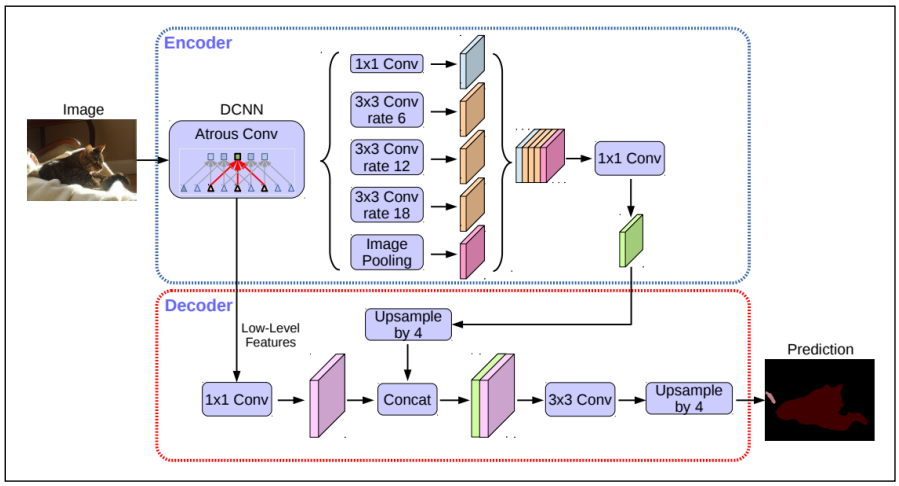


Fig. 4. DeepLab V3+ architecture [7]

Mask R-CNN (Fig. 5) [8] is an architectural evolution of the approaches laid down in Faster R-CNN, but with extended functionality to solve the instance segmentation, where it is required to separate each object within the same class. This architecture implements a two-stage approach: the first stage involves predicting regions of interest (RoI) in the input image, after which RoIAlign accurately extracts the corresponding regions from the feature maps generated by the feature extraction network (backbone). In the second stage, the extracted regions are classified, and in parallel, a separate branch of the model generates segmentation masks for each identified object. This approach became the standard for a long period and was able to predict accurate masks, even for objects with significant overlap. However, again, this approach is very expensive in computational resources and still shows suboptimal results for small-sized objects. An important advantage of Mask R-CNN is its versatility – it can be applied to classification, detection, and segmentation tasks. The disadvantages include the difficulty of training the two-stage approach on new data due to sensitivity to hyperparameters and dependence of mask prediction on the results obtained from the region proposal network (RPN).

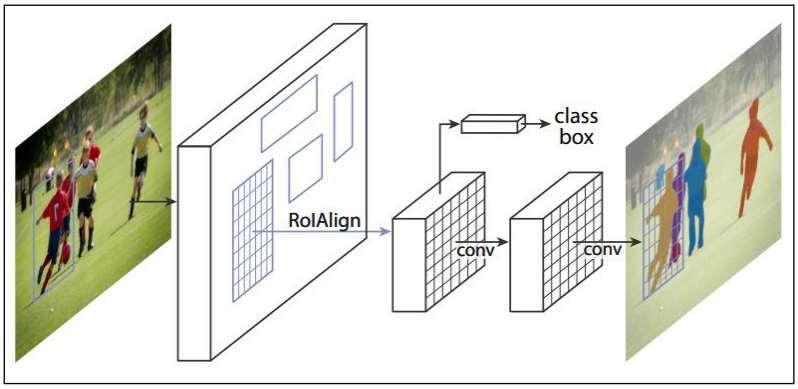


Fig. 5. Mask R-CNN architecture [8]

The problem of incorrect detection of object boundaries, especially in cases of their mutual overlap, remains one of the fundamental limitations of semantic segmentation algorithms. In the Gated-SCNN (Fig. 6) [9] method, the authors aimed to solve this problem. For this purpose, they used two branches in the architecture. The main backbone branch deals with extracting features from the input image, which are then fed via Gated Convolution Later into the second branch, which works with the original image pre-processed with the Canny filter – i.e., with gradients.

The results of these two branches are then combined via ASPP to form the final segmentation result. It should be noted that the Gated Convolution Layer module proposed within this architecture is conceptually equivalent to modern attention mechanisms. However, the main contribution of this paper is a new loss function – dual task loss, which provides the ability to compare the predicted object boundaries with the true ones. The main drawbacks of this model are its pronounced dependence on the quality of the training data annotations, because inaccuracies in the boundary labeling inevitably lead to a decrease in the accuracy of model predictions, and sensitivity to hyperparameter tuning, which is a typical characteristic of composite architectures and loss functions.

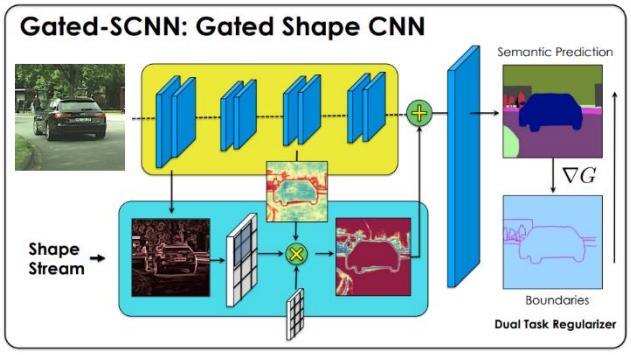


Fig. 6. Gated-SCNN architecture [9]

FastFCN (Fig. 7) [10] is another approach to semantic segmentation in which a new Joint Pyramid Upsampling (JPU) block has been proposed for upsampling. In this block, the input feature maps are first processed with standard convolution, followed by multiple separable convolutions with varying dilation rates. The authors of the research use the last three convolutional layers from the backbone network, although the number of convolutional layers can vary. This approach improves the reconstruction of fine details and significantly reduces computational cost. After JPU, there is a multi-scale/global context module to generate the final segmentation map, such as ASPP. The main advantages of this approach are the increased computational speed and the possibility to apply JPU to other already existing architectures or their modules, e.g., DeepLabV3 or other individual backbones. The main drawback, however, is that during training, the input image size had to be restricted due to memory limitations, and even a resolution difference of 96 pixels affected the performance of the model.

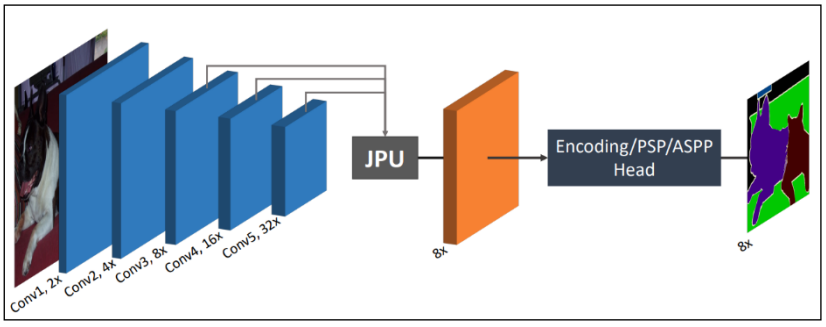


Fig. 7. FastFCN architecture [10]

MaskFormer (Fig. 8) [11] is a segmentation model that incorporates an attention mechanism into its architecture. Its fundamental conceptual difference is not in per-pixel classification or selecting

regions of interest with objects, but in first generating a set of object masks, which are then classified. To implement this approach, the backbone extracts embeddings, which are processed by a pixel decoder and a transformer decoder to produce binary object masks. The architecture of the decoders is a box-free DETR [12]. This proposed architecture demonstrates high efficiency and compatibility with various backbones. For example, compared to DeepLabV3+, MaskFormer requires fewer parameters while achieving higher accuracy when using the same backbone. The disadvantages include the need for large training datasets and careful parameter selection, as transformers tend to achieve better results when trained on large amounts of data.

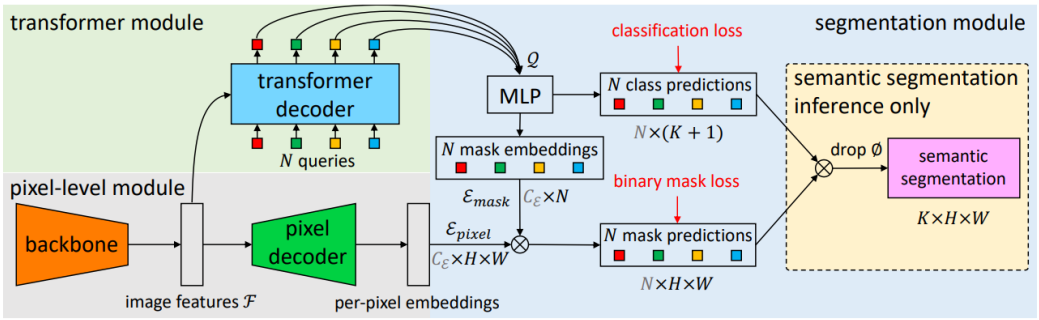


Fig. 8. MaskFormer architecture [11]

The SegFormer (Fig. 9) [13] architecture is an encoder-decoder semantic segmentation model. Its encoder is implemented as a hierarchical transformer that processes images in 4×4 patches, progressively reducing the spatial resolution and generating multi-level features. This design provides the ability to process input images of varying sizes without the need for positional encoding mechanisms. The main focus of SegFormer architecture is its minimalist decoder, which consists solely of a multilayer perceptron (MLP) and upsample layers, significantly reducing computational complexity. The decoder aggregates the feature maps produced by the encoder, aligning them to a unified resolution to produce the final result. The presented solutions and innovations in the SegFormer model provide an optimal balance between computational efficiency and segmentation accuracy, as demonstrated by SOTA results among the models of the corresponding period.

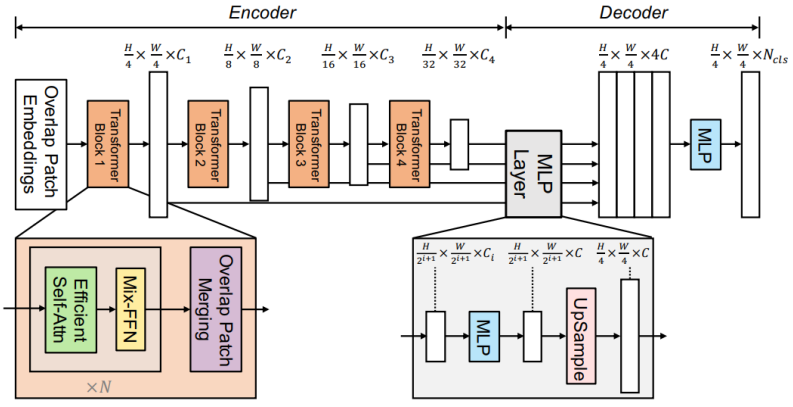


Fig. 9. SegFormer architecture [13]

Mask2Former (Fig. 10) [14] is a development of the concepts introduced in MaskFormer. The main architectural innovation is the replacement of cross-attention with masked-attention and adding multi-resolution feature maps to the pixels decoder, similar to the Feature Pyramid

Network (FPN) design. Masked attention allows the model to focus more effectively on objects or regions of interest, which accelerates the convergence process. The use of multi-scale high-resolution features in the pixel decoder allows the model to segment small objects or regions. To optimize memory usage during training, the authors proposed an approach in which only a randomly selected subset of ground-truth mask points is matched, rather than all pixels. This technique is inspired by the PointRend [15] and the Implicit PointRend [16] techniques and reduces memory consumption by up to three times. The key advantages of the model are its versatility for different types of segmentation tasks and its high accuracy. However, as with many high-performing models, these benefits come at the cost of increased computational demands and architectural complexity.

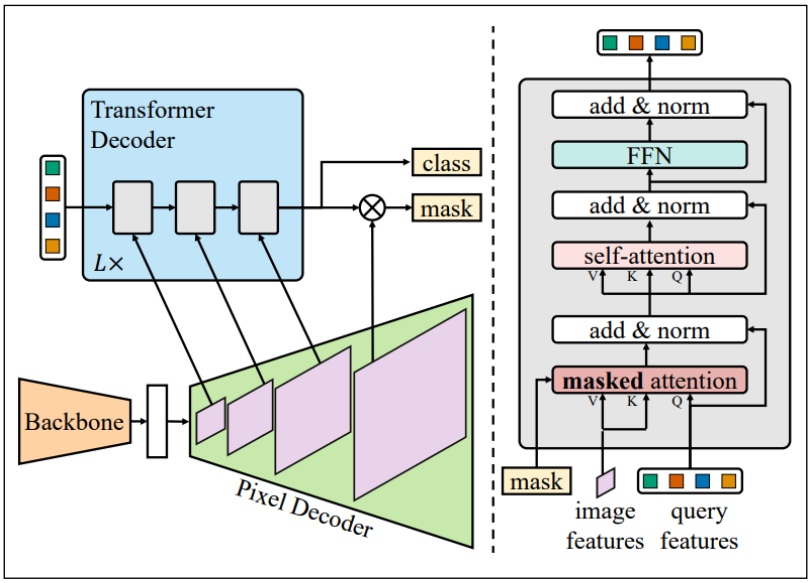


Fig. 10. Mask2Former architecture [14]

OneFormer (Fig. 11) [17] is a logical continuation of the MaskFormer and Mask2Former architectures, offering a unified approach to various segmentation tasks, including semantic, instance, and panoptic segmentation within a single model. The key difference of OneFormer is the use of a task-conditional token that defines the type of task being performed. The model architecture consists of three main components: an image encoder based on the Swin Transformer, a task token, and a decoder with multi-level attention mechanisms. The segmentation process in OneFormer involves extracting visual features from the input image, incorporating a task token, generating object queries in the decoder depending on the type of task, and predicting binary masks along with their corresponding classes. Thanks to its unified architecture and task-conditioned design, OneFormer achieves high performance across all three segmentation tasks without requiring task-specific reconfiguration. However, the model has several limitations, including high training complexity, the need for large amounts of annotated data, and significant computational cost.

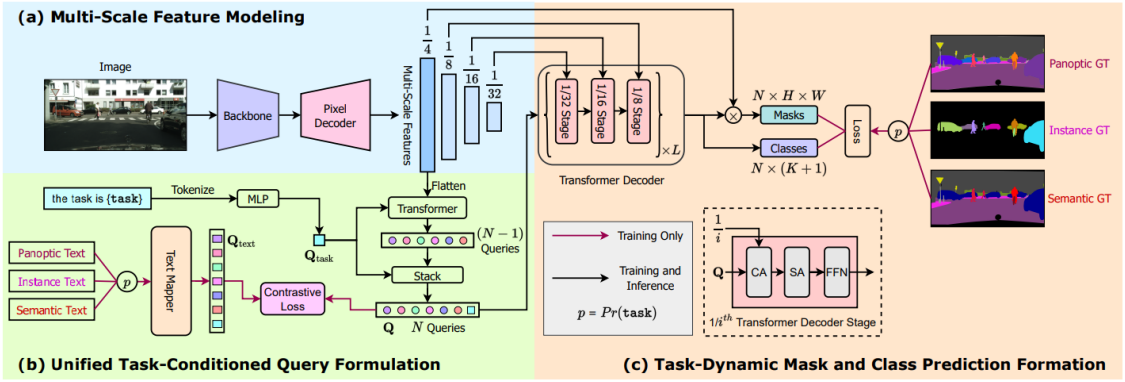


Fig. 11. OneFormer architecture [17]

Segment Anything Model (SAM) (Fig. 12) [18] is a fundamentally new paradigm in segmentation focusing on universality, interactivity, and zero-shot generalization to unseen domains. The architecture consists of three main components: an image encoder based on a modified Vision Transformer (ViT), a prompt encoder capable of handling different types of input data (e.g., text, points, boxes, masks), and a lightweight mask decoder that produces segmentation masks. What sets SAM aside from other models is its training on an unprecedentedly large dataset (more than one billion masks) and its interactive behavior that supports iterative refinement by allowing users to provide new prompts interactively. The model demonstrates strong adaptability to new domains and object types. Nevertheless, SAM has several limitations, including high computational costs, reduced accuracy for segmentation of small objects, complex or fine-grained boundaries, and difficulties with contextual understanding of complex scenes.

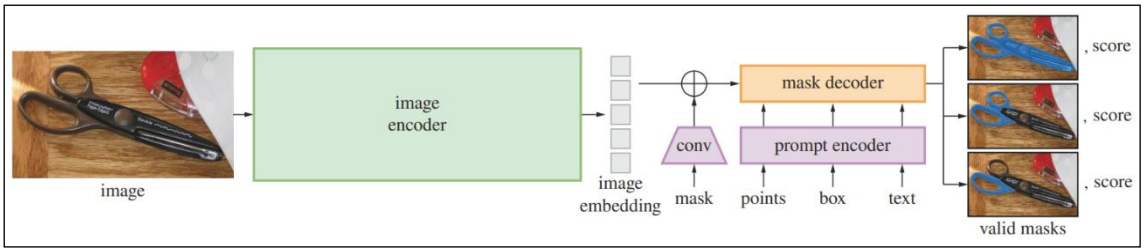


Fig. 12. SAM architecture [18]

HQ-SAM (Fig. 13) [19] is an enhanced version of the original SAM model designed to improve the quality of predicted masks, especially in the context of accurate object boundary detection. Although SAM was trained on 1B masks, its segmentation results may be suboptimal when dealing with objects that have complex structures. The HQ-SAM model addresses two main problems of the original SAM: inaccurate mask boundaries for small or thin objects, and incorrect mask predictions or fragmented predictions in complex scenes. These types of errors significantly reduce the applicability and effectiveness of SAM. To address this, the HQ-SAM architecture introduces two main components while retaining the zero-shot capabilities of the original model. The High-Quality Output Token (HQ-Output Token) is a trainable token added to the SAM mask decoder along with the original prompt and output tokens. The second innovation, Global-local Feature Fusion, combines features from the SAM mask decoder with both early-layer local features and final-layer global features from the ViT encoder. This fusion allows for taking into account boundary information, semantic context, and mask shape representation from both the ViT encoder and mask decoder to improve mask refinement. A TinyViT-based Light HQ-SAM [19] model variant was also introduced to support resource-constrained devices, since the full

HQ-SAM model has even more parameters than the original SAM.

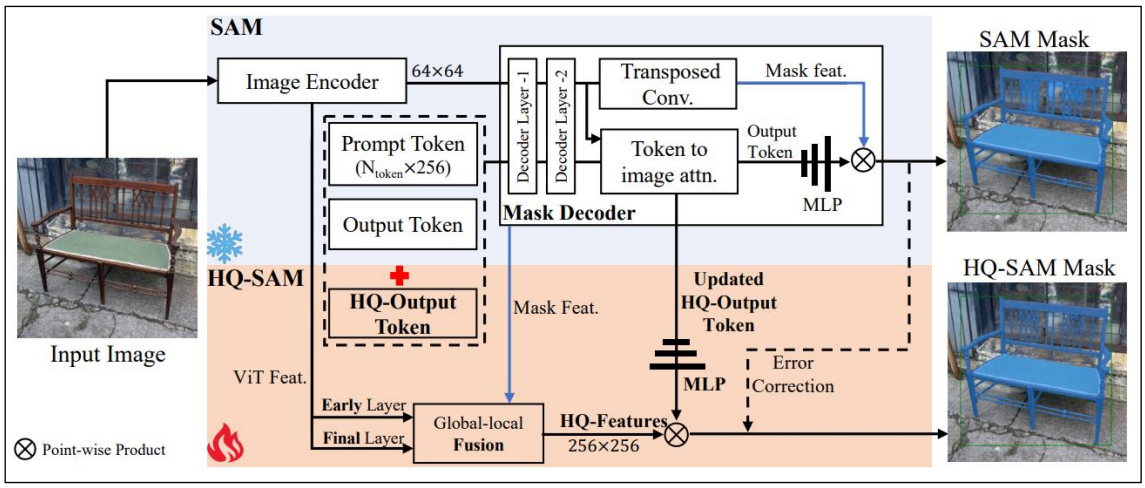


Fig. 13. HQ-SAM architecture [19]

4. Conclusions and prospects of further research

This paper provides a comprehensive review of the evolution of neural network architectures for image segmentation tasks. It examines key milestones in the development of segmentation models, from early fully convolutional networks to modern transformer-based architectures. Technical features, advantages, and limitations of different architectures were analyzed. Reviewed architectures clearly illustrate the progress of models and their approaches to problem solving. Each of the analyzed architectures has its advantages and disadvantages, which determine the areas of their optimal application in different types of segmentation. Modern architectures have successfully overcome many of the limitations of earlier approaches, offering universal solutions for various types of segmentation with an optimal balance between accuracy and efficiency. Nevertheless, several fundamental problems remain unresolved, such as domain adaptation, accurate small objects segmentation, and reducing dependence on large amounts of labeled data. Transformer-based architectures, despite their high accuracy, still require significant computational resources, which limits their application in real-time systems and highlights the need for further research to find the optimal balance between segmentation quality and computational efficiency.

References:

1. J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2015. doi:10.1109/cvpr.2015.7298965

2. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” Lecture Notes in Computer Science, pp. 234–241, 2015. doi:10.1007/978-3-319-24574-4_28

3. V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A deep convolutional encoder-decoder architecture for image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481–2495, Dec. 2017. doi:10.1109/tpami.2016.2644615

4. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Semantic image segmentation with deep convolutional nets and fully connected crfs“. arXiv preprint arXiv:1412.7062, Jun. 2016.

5. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 834–848, Apr. 2018. doi:10.1109/tpami.2017.2699184

6. L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation,” arXiv preprint arXiv:1706.05587, Jun. 2017.

7. L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for Semantic Image segmentation,” Lecture Notes in Computer Science, pp. 833–851, 2018.

doi:10.1007/978-3-030-01234-2_49

8. K. He, G. Gkioxari, P. Dollar, and R. Girshick, “Mask R-CNN,” 2017 IEEE International Conference on Computer Vision (ICCV), Oct. 2017. doi:10.1109/iccv.2017.322

9. T. Takikawa, D. Acuna, V. Jampani, and S. Fidler, “Gated-SCNN: Gated shape cnns for semantic segmentation,” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2019. doi:10.1109/iccv.2019.00533

10. H. Wu, J. Zhang, K. Huang, K. Liang, and Y. Yu, “FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation,” arXiv preprint arXiv:1903.11816, Mar. 2019.

11. B. Cheng, A. G. Schwing, and A. Kirillov, “Per-Pixel Classification is Not All You Need for Semantic Segmentation,” Advances in neural information processing systems, vol. 34, pp. 17864–17875, Jul. 2021.

12. N. Carion et al., “End-to-end object detection with Transformers,” Lecture Notes in Computer Science, pp. 213–229, 2020. doi:10.1007/978-3-030-58452-8_13

13. E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers,” Advances in neural information processing systems, vol. 32, pp. 12077–12090, May 2021.

14. B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, “Masked-attention mask transformer for Universal Image segmentation,” 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1280–1289, Jun. 2022. doi:10.1109/cvpr52688.2022.00135

15. A. Kirillov, Y. Wu, K. He, and R. Girshick, “Pointrend: Image segmentation as rendering,” 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2020. doi:10.1109/cvpr42600.2020.00982

16. B. Cheng, O. Parkhi, and A. Kirillov, “Pointly-supervised instance segmentation,” 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2022. doi:10.1109/cvpr52688.2022.00264

17. J. Jain et al., “OneFormer: One Transformer to rule Universal Image segmentation,” 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2989–2998, Jun. 2023. doi:10.1109/cvpr52729.2023.00292

18 A. Kirillov et al., “Segment Anything,” Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4015–4026, Apr. 2023.

19. L. Ke et al., “Segment Anything in High Quality,” Advances in Neural Information Processing Systems, vol. 36, pp. 29914–29934, Jun. 2023.

Надійшла до редколегії 14.05.2025 р.

Kovtunenko Andrii Romanovych, PhD student, Informatics Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, e-mail: andrii.kovtunenko@nure.ua, ORCID: https://orcid.org/0009-0004-9072-7779

Mashtalir Sergii Volodymyrovych, Doctor of Engineering Science, Professor, Informatics Department. Kharkiv National University of Radio Electronics. Kharkiv, Ukraine, e-mail: sergii.mashtalir@nure.ua, ORCID: https://orcid.org/0000-0002-0917-6622

УДК 004.8:004.9 DOI: 10.30837/0135-1710.2025.185.053

С.Ф. ЧАЛИЙ, В.О. ЛЕЩИНСЬКИЙ

ПРИЧИННО-НАСЛІДКОВЕ МОДЕЛЮВАННЯ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНИХ СИСТЕМАХ

Об'єктом дослідження є процес побудови пояснень в інтелектуальних інформаційних системах. Запропоновано причинно-наслідкову модель пояснення з ієрархічною трирівневою структурою, яка формує каузальні залежності на глобальному, процесному та локальному рівнях у темпоральному та каузальному аспектах. Запропоновано метод побудови пояснень на основі причинно-наслідкової моделі, що містить етапи визначення користувачів, побудови пояснень на кожному рівні ієрархії причинно-наслідкової моделі та формування інтегрованого трирівневого пояснення. Використання методу дозволяє деталізувати пояснення відповідно до потреб внутрішніх користувачів (розробників, адміністраторів) та зовнішніх користувачів (кінцевих клієнтів), що дозволяє підвищити прозорість рішень та довіру до отриманих рішень інтелектуальних систем.

1. Вступ

Сучасні інтелектуальні системи, які включають в себе як підсистеми штучного інтелекту, так і традиційні інформаційні підсистеми, набули поширення у таких сферах людської діяльності, як охорона здоров'я, фінансові послуги, управління підприємствами, освіта [1]. Ці інтелектуальні інформаційні системи (ІІС) забезпечують підтримку прийняття рішень, що безпосередньо впливають на життя людей та функціонування підприємств та організацій [2]. Підсистеми штучного інтелекту використовують методи машинного навчання та складні репрезентації даних [3]. Тому вони, здебільшого, функціонують як «чорні скриньки», що утруднює пояснення процесу їхньої роботи [1], [4]. Традиційні інформаційні системи зазвичай використовують детерміновані процеси із зрозумілою логікою, проте ці процеси з точки зору користувача часто мають вигляд «сірих скриньок» внаслідок великої кількості параметрів та складних алгоритмів [5]. Алгоритми та програмні рішення також можуть бути захищені в рамках комерційної таємниці, що обмежує їхню прозорість для користувача. Результатом юридичного захисту є представлення ІІС як «правової чорної скриньки». Проте використання складних непрозорих алгоритмів у процесі формування рішень в ІІС знижує довіру користувачів до цих рішень, що суттєво обмежує ефективність їх практичного застосування [6].

З метою підвищення довіри користувачів ІІС та, відповідно, ефективного використання рішень ІІС проводяться дослідження у сфері пояснюваного штучного інтелекту (XAI) [7]. Пояснення забезпечують прозорість роботи ІІС за рахунок інтерпретації причин послідовності дій процесу формування рішень.

Таким чином, проблема формування та використання пояснень в ІІС з урахуванням причин рішень та процесу формування рішень в ІІС є актуальною.

2. Аналіз літературних даних і постановка проблеми дослідження

Сучасні підходи до побудови пояснюваних ІІС включають два напрямки. Перший напрямок орієнтований на створення інтерпретованих моделей, логіка яких побудована на правилах, графах прийняття рішень тощо [8]. Другий напрямок реалізує post-hoc пояснення для моделей, які представляються для користувача як «чорні скриньки». В рамках даної групи виконується локальна апроксимація складної функції лінійною моделлю (LIME) [6] та визначення внеску значень вхідних даних за кооперативною ігровою схемою (SHAP) [9].

Обидва напрямки забезпечують інтерпретацію конкретних рішень, проте не

враховують динаміку прийняття рішень. Користувач отримує перелік факторів впливу на результат, але не має інформації, коли і як саме вони вплинули на цей результат. Така можливість є критичною для предметних областей, де послідовність дій або послідовність подання вхідних даних впливає на отримане рішення (медична діагностика, фінансовий аналіз тощо).

Темпоральні аспекти відображаються як упорядкована послідовність для пар подій. В [10], [11] запропоновано моделювати можливі причини рішення на основі темпорального порядку подій процесу формування рішення та подальшої побудови причинно-наслідкових залежностей на основі темпоральної упорядкованості. Темпоральний порядок процесу формування рішення в роботах [12]-[15] задається за допомогою темпоральних правил.

З іншого боку, дослідження у сфері причинно-наслідкового виведення [16], [17] дають можливість визначати причини отриманого результату, але вони не приділяють уваги темпоральному порядку дій процесу. Проте емпіричні спостереження в області когнітивної психології [18] підтверджують, що люди вважають пояснення переконливим лише тоді, коли воно враховує темпоральний аспект. Тому поєднання формування причинно-наслідкових залежностей на базі темпоральних правил дає можливість врахувати динаміку роботи ІІС [19].

Таким чином, існуючі підходи до побудови пояснень фокусуються переважно на визначенні факторів, які впливають на отримане рішення, і не приділяють достатньо уваги упорядкованому в часі процесу формування рішення, що і потребує розробки комплексного багаторівневого підходу до побудови пояснень. Такий підхід має виявити причини рішення на основі вхідних даних, а також дій процесу з формування рішення, що і свідчить про актуальність теми даного дослідження.

3. Мета і задачі дослідження

Метою дослідження є розробка причинно-наслідкового підходу до побудови пояснень в інтелектуальних системах з тим, щоб відобразити у формі пояснення причини отриманого рішення, а також причини дій процесу формування рішення й обмеження на ці дії, що дає можливість представити роботу інтелектуальної системи у зрозумілій для користувачів формі.

Для досягнення мети роботи вирішуються такі задачі: розробка причинно-наслідкової моделі пояснення; розробка методу побудови пояснень на основі причинно-наслідкової моделі.

4. Причинно-наслідкова модель пояснення

Об'єктом дослідження є процес побудови пояснень в ІІС.

Предметом дослідження є моделі пояснень та методи їх побудови в ІІС.

Розроблена причинно-наслідкова модель пояснення M має ієрархічну структуру та включає три базових рівні L представлення множини каузальних залежностей R, що базуються на темпоральній упорядкованості вхідних даних T_{in} та дій процесу формування рішення T_d . Ці рівні визначають такі аспекти пояснення:

– глобальний L_{Gl}, де пояснення задає тлумачення причин рішення ІІС в цілому, без деталізації внутрішнього механізму його формування;

– процесний L_{Pr}, де пояснення відображає причинно-наслідкові залежності між діями процесу формування рішення в ІІС та отриманим рішенням; ці залежності базуються на темпоральній упорядкованості дій процесу;

– локальний L_{Loc}, де описуються окремі обмеження на процес формування рішення, причому ці обмеження задаються через темпоральну упорядкованість пар дій процесу.

Наприклад, у медичній діагностичній системі модель пояснення на глобальному рівні містить причинно-наслідкові залежності із множини виду «*симптоми* $\rightarrow$ *діагноз*», на

процесному рівні модель M визначає послідовність аналізів та діагностичних дій, а на локальному рівні – обов'язкові етапи обстеження.

Формально дана модель має вигляд:

$$M = \langle L, R, T \rangle, \tag{1}$$

де $L = \{L_{\text{Gl}}, L_{\text{Pr}}, L_{\text{Loc}}\}$ – множина рівнів ієрархії пояснення; $T = \{T_{in}, T_d\}$ – темпоральна упорядкованість даних та дій процесу; R – відношення, на основі яких будується пояснення.

Така ієрархічна структура дає можливість формувати пояснення як каузальні залежності із заданим ступенем деталізації відповідно до потреб внутрішніх та зовнішніх користувачів. Внутрішні користувачі вирішують задачі із розробки та підтримки функціонування ІІС і тому в рамках пояснень потребують детального розуміння внутрішньої логіки поточного процесу прийняття рішень для того, щоб виявити можливі помилки й порушення обмежень у даному процесі, удосконалити моделі роботи ІІС та забезпечити відповідність регуляторним вимогам. Тому для цієї категорії користувачів пояснення мають відображати причини можливих послідовностей дій, а також порушення обмежень при функціонуванні ІІС. Зовнішні користувачі використовують рішення ІІС у своїй професійній діяльності, і тому пояснення для цих користувачів мають забезпечити зрозумілість причин отриманих результатів та здатність відкоригувати вхідні дані й параметри ІІС для отримання бажаних результатів.

Ключові властивості наведених рівнів представлення пояснення у причинно-наслідковій моделі наведено в табл. 1.

Таблиця 1

Властивості рівнів пояснення в причинно-наслідковій моделі

Характеристика	Глобальний рівень	Процесний рівень	Локальний рівень
Користувачі	Зовнішні користувачі (кінцеві клієнти)	Внутрішні користувачі (розробники, data scientists)	Внутрішні користувачі (системні аналітики)
Прозорість ІІС для користувача	Чорна скринька	Сіра скринька	Сіра скринька або біла скринька
Об'єкт пояснень	Причини рішення ІІС	Причини дій процесу формування рішення в ІІС	Причини обов'язкових дій у процесі формування рішення в ІІС
Вхідні дані для пояснення	Лінійно упорядковані значення вхідних змінних, рішення ІІС	Логи процесу формування рішень в ІІС	Логи процесу формування рішень в ІІС
Представлення пояснення	Можливісні причинно-наслідкові залежності між вхідними даними та рішенням ІІС	Можливісні причинно-наслідкові залежності між діями	Детерміновані або ймовірнісні обмеження щодо допустимих варіантів процесу формування рішення в ІІС

Глобальний рівень причинно-наслідкової моделі пояснення містить каузальні залежності між вхідними даними та рішенням ІІС. На даному рівні ІІС є непрозорою для кінцевого користувача, тобто розглядається як «чорна скринька». Для побудови залежностей на даному рівні причинно-наслідкового пояснення як можлива причина використовуються або безпосередньо значення вхідних змінних, або приховані фактори

(неявні вхідні дані). Крім того, на вході можуть бути використані темпоральні правила, що відображають послідовність значень цих змінних у наборі вхідних даних. Лінійний порядок вхідних даних у часі дає можливість врахувати послідовність впливу вхідних змінних на формування рішення. Наслідком є рішення системи, яке має бути використано у відповідній предметній області.

Каузальні залежності R_{Gl}, що є основою пояснення на даному рівні, визначаються в рамках теорії можливості, тобто з використанням показників можливості Π_{Gl} та необхідності N_{Gl}. Такий підхід дозволяє враховувати невизначеність, що виникає внаслідок неповноти інформації щодо внутрішнього механізму функціонування ІІС. Пояснення на глобальному рівні орієнтовані на користувачів – кінцевих клієнтів, що використовують рішення ІІС у своїй професійній діяльності.

Формально пояснення на глобальному рівні L_{Gl} задають причинно-наслідкові залежності R_{Gl} між вхідними даними X та рішенням Y з урахуванням упорядкованості вхідних даних T_{in} та прихованих факторів U:

$$L_{\mathrm{Gl}} = R_{Gl}\left(X, T_{in}, Y, U\right) : \forall R_{Gl} \exists \left(\Pi_{Gl}, N_{Gl}\right). \tag{2}$$

Множина вхідних змінних може бути упорядкована за відношенням T_{in}:

$$X = \left\{ x_i : (\forall i, i+1) \exists T_{in}^{i,i+1} \right\}. \tag{3}$$

Каузальні залежності на даному рівні визначаються шляхом оцінки можливості та необхідності конкретних значень вхідних для даних $X = x$ бути причиною конкретного рішення $Y = y$.

Можливість визначається з урахуванням прихованих факторів $u \in U$, які разом із поточними вхідними даними x приводять до поточного результату y, тобто $f(x,u) = y$:

$$\Pi_{Gl}\left(Y = y \middle| X = x\right) = \max_{u \in U} \left\{ \pi(u) : f(x,u) = y \right\}. \tag{4}$$

Наприклад, в банківській кредитній системі можливість $\Pi_{Gl} = 0,75$ рішення $Y = y : \textit{кредит схвалено}$, що отримано на основі вхідних даних $X = x : \left(\textit{дохід} = 80000\ \textit{грн}, \textit{вік} = 40\ \textit{років}\right)$ за заданих прихованих факторів $U = \left\{\textit{поточна процентна ставка, ризики, сезонні коливання}\right\}$ та $\max\limits_{u \in U} \pi(u) = \pi(\textit{поточна процентна ставка}) = 0,75$, означає, що найбільший вплив на рішення щодо схвалення кредиту має поточна процентна ставка. Значення процентної ставки можна розглядати як основну причину виділення кредиту для клієнта віком 40 років з доходом 80000 гривень.

Оцінка необхідності є дуальною до оцінки можливості Π_{Gl} та базується на оцінці можливості альтернатив до Π_{Gl} і, відповідно, визначає ступінь неминучості рішення $Y = y$ за вхідних даних $X = x$ з урахуванням прихованих факторів $u \in U$:

$$N_{Gl}(Y = y |\ X = x) = 1 - \Pi_{Gl}(Y \neq y |\ X = x). \tag{5}$$

$\Pi_{Gl}(Y \neq y |\ X = x)$ відображає можливість всіх інших результатів крім Y. Наприклад $Y \neq y : \left\{\textit{кредит відхилено}, \textit{кредит потребує додаткових документів}\right\}$. Високе значення

необхідності означає, що альтернативні результати $Y \neq y$ мають низьку можливість. Якщо $N_{Gl}(Y = y|\ X = x) = 0{,}85$, то можливість альтернативного результату не перевищує 0,15, що свідчить про «обов'язковість» результату $Y = y|\ X = x$. У відповідності до прикладу кредитування, $N_{Gl} = 0{,}85$ означає, що для представлених даних $X = x$: $(\textit{дохід} = 80000\ \textit{грн, вік} = 40\ \textit{років})$ кредит на практиці буде схвалено.

Таким чином, високі оцінки можливості та необхідності свідчать про те, що задані вхідні дані та приховані фактори є причиною отриманого рішення. Тобто оцінка можливості визначає максимальний ступінь довіри до каузальної залежності з урахуванням значень прихованих факторів u. Міра необхідності є дуальною до міри можливості та показує, наскільки неминучим є певний результат. Відповідно, пояснення на глобальному рівні формується за значенням оцінки можливості та при обмеженнях за значенням оцінки необхідності.

Процесний рівень причинно-наслідкової моделі пояснення призначений для опису механізму формування рішення в ІІС. На даному рівні пояснюється послідовність дій процесу прийняття рішень на основі формування причинно-наслідкових залежностей між цими діями. Такі залежності встановлюються як між послідовними парами дій, так і між діями, між якими існують проміжні етапи процесу формування рішення [3], [11]. Такий підхід до формування пояснення дає можливість представити користувачам не лише причини кінцевого рішення ІІС, але й логіку формування цього рішення через послідовність упорядкованих у часі дій. Причинно-наслідкові залежності на даному рівні базуються на темпоральних правилах типів «Next» та «Future», що відображають безпосередні та віддалені темпоральні залежності між діями процесу формування рішення. Правила типу «Next» задають порядок у часі для дій, які безпосередньо йдуть одна за одною. Правила типу «Future» охоплюють декілька послідовних дій і враховують опосередковані зв'язки між першою та останньою діями із цієї послідовності. Процес формування рішення у загальному випадку може мати декілька альтернативних реалізацій, які відрізняються можливими послідовностями дій. Відповідно, каузальні залежності на процесному рівні причинно-наслідкового пояснення визначаються з використанням теорії можливості на основі показників можливості та необхідності для кожної пари дій, представленої темпоральним правилом «Next» або «Future». Показник можливості дозволяє оцінити ступінь довіри до першої дії як причини другої дії із відповідної пари дій. Необхідність є оцінкою можливості альтернативної послідовності дій. Тобто необхідність обумовлює порівняння поточної причини з альтернативними або контрфактичними. Такий підхід дозволяє оцінити поточну реалізацію процесу формування рішення у порівнянні з можливими альтернативними послідовностями дій.

Формально процесний рівень $L_{\text{Рг}}$ причинно-наслідкової моделі пояснення представляється через множину дій D процесу формування рішень, упорядкованих за темпоральними відношеннями R_N та R_F типів «Next» та «Future» відповідно. Оцінка причин дій процесу виконується на базі можливості $\Pi_{\text{Рг}}$ та необхідності $N_{\text{Рг}}$:

$$L_{\text{Рг}} = \langle D, R_N, R_F, \Pi_{\text{Рг}}, N_{\text{Рг}} \rangle. \quad (6)$$

Оцінка можливості виконується для поточного рішення $Y = y$ та поточної послідовності дій $D = d$ процесу формування рішення на основі темпоральних відношень R_F, оскільки останні визначають ланцюжки дій $R_F\left(d_i, d_j\right)$, упорядкованих з використанням проміжних відношень R_N. Тобто знаходиться ключовий ланцюжок дій

$\left(d_i, d_j\right)$: $j-1>1$, який привів до отриманого результату:

$$\Pi_{\text{Pr}}\left(Y=y\middle|D=d\right)=\max_{j,j\in D}\left\{\pi\left(R_F\left(d_i,d_j\right)\right)\right\}. \quad (7)$$

Розрахунок $\max\limits_{j,j\in D}\left\{\pi\left(R_F\left(d_i,d_j\right)\right)\right\}$ виконується по всіх відомих альтернативних шляхах формування результату. Тобто $\pi\left(R_F\left(d_i,d_j\right)\right)$ визначається через нормовану кількість повторів правила $R_F\left(d_i,d_j\right)$ на відомій множині реалізацій процесу формування рішення в ІІС.

Необхідність N_{Pr} на процесному рівні розраховується аналогічно глобальному рівню як доповнення можливості Π_{Pr}:

$$N_{\text{Pr}}\left(Y=y\middle|D=d\right)=1-\Pi_{\text{Pr}}\left(Y\neq y\middle|D=d\right). \quad (8)$$

У підсумку, на даному рівні пояснення встановлюється можливість та необхідність для правил R_F, що визначають послідовність дій, яка мала найбільший вплив на отримане рішення. Наприклад, дія з призначення лікування d_j може бути визначена на основі послідовності дій після аналізу крові або ж після МРТ.

Локальний рівень причинно-наслідкового пояснення задає обмеження на множину можливих варіантів реалізації процесу формування рішення. Відповідно, локальний рівень є нижнім рівнем в ієрархії причинно-наслідкового пояснення. На даному рівні виділяються пари упорядкованих у часі дій процесу формування рішення в ІІС, які мають виконуватися для всіх або переважної більшості відомих варіантів даного процесу. Такий підхід створює умови для побудови детермінованих або ймовірнісних причинно-наслідкових залежностей між парами дій.

Формально локальний рівень причинно-наслідкового пояснення задається через множину дій D та правила C_N та C_F, які базуються на правилах R_N та R_F відповідно і виконуються для всіх варіантів процесу формування рішення в ІІС:

$$L_{\text{Loc}}=\langle D, C_N, C_F\rangle. \quad (9)$$

C_N визначається через темпоральне правило, яке виконується для всіх відомих послідовностей дій $\left(d_1,...,d_n\right)\in D$ процесу формування рішення:

$$C_N=R_N:\forall\left(d_1,...,d_n\right)\exists R_N. \quad (10)$$

Наприклад, при двохфакторній авторизації в банківській системі після дії d_1 із вводу пароля завжди виконується дія d_2 із підтвердження особи.

Процесний та локальний рівень причинно-наслідкової моделі пояснення орієнтовані, в першу чергу, на внутрішніх користувачів, тоді як глобальний рівень – на зовнішніх користувачів. Ці два класи користувачів відрізняються можливостями доступу до інформації щодо процесу роботи ІІС, а також практичними задачами у сфері їхньої професійної діяльності.

5. Метод побудови пояснень на основі причинно-наслідкової моделі

Розроблений метод містить такі етапи.

Етап 1. Визначення внутрішніх та зовнішніх користувачів.

Визначаються кінцеві клієнти, які будуть використовувати пояснення для вирішення

свої практичних задач, а також розробники, аналітики даних тощо, які будуть використовувати пояснення для удосконалення ІІС.

В залежності від результатів етапу будуть сформовані пояснення на одному, двох, або трьох рівнях.

Етап 2. Побудова пояснення на глобальному рівні.

Крок 2.1. Визначення підмножини значень вхідних змінних $X = x$, властивостей рішення $Y = y$ та внутрішніх факторів $u \in U$ для побудови пояснення.

Крок 2.2. Розрахунок можливості та необхідності для встановлення причин рішення згідно з (4) та (5).

Результатом етапу є пояснення причини рішення на основі прихованих факторів за умови поточних вхідних даних з оцінками його можливості та необхідності.

Етап 3. Побудова пояснення на процесному рівні.

Крок 3.1. Визначення множини дій процесу D.

Крок 3.2. Формування темпоральних правил R_N та R_F.

Крок 3.3. Розрахунок можливості та необхідності для ланцюжка дій – причини отриманого рішення згідно з (7) та (8).

Результатом етапу є пояснення причини рішення як ключової послідовності дій процесу формування рішення з оцінками його можливості та необхідності.

Етап 4. Побудова пояснення на локальному рівні.

Крок 4.1. Знаходження обмежень C_N на основі темпоральних правил R_N, які виконуються для всіх варіантів процесу формування рішення.

Крок 4.2. Знаходження обмежень C_F на основі темпоральних правил R_F, які виконуються для всіх варіантів процесу формування рішення.

Крок 4.3. Перевірка виконання обмежень для поточного процесу формування рішення.

Результатом етапу є пояснення у формі виконання/невиконання обмежень поточного процесу формування рішення.

Етап 5. Побудова об’єднаного трирівневого пояснення, що містить як причини рішення значення вхідних даних та дії процесу формування рішення, а також інформацію про задоволення обмежень на процес формування рішення.

6. Експериментальна перевірка методу побудови пояснень на основі причинно-наслідкової моделі

Експериментальну перевірку розробленого методу виконано з використанням логу фірми Volvo IT, який містить записи про невирішені проблеми управління сервісними процесами у інформаційній системі компанії.

Вхідний лог містить 819 трас, що відображають можливі варіанти реалізації процесу сервісного обслуговування у інформаційній системі компанії Volvo IT. Ключові дії d_i даного процесу – Accepted, Queued, Completed.

Аналіз логу на етапі 1 методу дає можливість виділити таких ключових внутрішніх користувачів: менеджери процесів, аналітики. Зовнішніми користувачами є клієнти, які надають інформацію про невирішені проблеми, та бізнес-користувачі.

За результатами етапу 2 методу виділено набір вхідних даних $X = \{Тип\ продукту\ (product)\}$, а також набір факторів $U = \{Робоча\ група\ (org:group), Країна\ (resource\ country), Вплив\ (impact)\}$, які впливають на рішення системи. Рішення містить статус процесу: $Y = \{Статус\ завершення:\ Completed\ /\ Not\ Completed\}$.

Розрахунок причин виконано для робочих груп. Результати для ключових робочих груп (з максимальним значенням можливості) наведено в табл. 2.

Таблиця 2

Можливості впливу робочих груп на результат Y процесу

Робоча група	Можливість $\pi(u)$	Необхідність $n(u)$
A2	0,85	0,62
C	0,78	0,45
G4	0,72	0,38
B	0,69	0,34

Аналіз табл. 2 дає можливість зробити висновок, що найвищу можливість успішного виконання процесу у Volvo IT має робоча група A2. Тобто якщо поточний процес обслуговує інша група, то поясненням невирішеної проблеми є недостатня кваліфікація цієї робочої групи.

На етапі 3 побудовано темпоральні правила R_N та R_F та виконано розрахунок можливості та необхідності для цих правил.

Отримано такі ключові залежності:

– можливість: $Accepted \rightarrow In\ Progress(0,95)$;

– необхідність: $Queued \rightarrow Accepted\ (0,78)$.

Тобто залежність $Accepted \rightarrow In\ Progress$ має найбільший вплив на успішне завершення процесу, а залежність $Queued \rightarrow Accepted$ у більшості випадків є необхідною для успішного завершення процесу.

На етапі 4 виконано формування обмежень на послідовність дій процесу. Виконано аналіз всіх трас процесу. Обмеженнями в рамках експерименту вважаються ті правила, які виконані на більш ніж 95 % трас.

Результати етапу наведено у табл. 3.

Таблиця 3

Обмеження на виконання процесу

Правило-обмеження	Виконання, %	Невиконання
$Queued \rightarrow Accepted$	96.8%	26 випадків
$Accepted \rightarrow Completed$	98.2%	14 випадків

На етапі 5 сформовано трирівневе пояснення для поточного варіанту процесу.

Глобальний рівень: робоча група A2 забезпечує можливість 0,85 успішного завершення процесу.

Процесний рівень: послідовність $Accepted \rightarrow In\ Progress$ забезпечує можливість 0,95 успішного завершення поточного процесу; послідовність $Queued \rightarrow Accepted$ є необхідною у поточному процесі і має значення необхідності 0,78.

Локальний рівень: наведені обмеження у поточному процесі повністю виконуються.

7. Обговорення результатів дослідження

Розроблена причинно-наслідкова модель пояснення та метод її побудови в цілому формують трирівневе представлення пояснень, придатне як для зовнішніх, так і для внутрішніх користувачів. Ієрархічне представлення пояснень на глобальному, процесному

та локальному рівнях дало змогу поєднати різні типи причин для отриманого рішення, включаючи значення вхідних змінних та обов’язкові операції процесу. Відмінність запропонованого підходу полягає в такому. На глобальному рівні враховано нечіткий характер впливу явних та неявних вхідних змінних через значення можливості та необхідності. На процесному рівні причини рішення встановлюються на основі темпоральних правил «Next»/«Future», які враховують як безпосередні, так і віддалені залежності між діями процесу формування рішення. На локальному рівні визначено детерміновані обмеження, які визначають множину допустимих варіантів виконання процесу формування рішення в ІІС.

Розроблений метод побудови причинно-наслідкових пояснень послідовно формує пояснення на кожному рівні моделі, а потім об’єднує їх у єдину структуру.

Переваги запропонованого підходу полягають у можливості порівняти значення вхідних даних та дії процесу формування рішення як причини отриманого рішення, а також встановити виконання обмежень для поточного процесу.

Проведений на даних із журналу Volvo IT експеримент дозволив виявити робочі групи з максимальною можливістю успішного вирішення проблемних ситуацій, критичні послідовності дій процесу з найбільшими значеннями можливості та необхідності, а також підтвердити виконання обмежень на процес. Аналіз всіх альтернативних варіантів виконання процесу дозволив виявити порушення обмежень у окремих альтернативах процесу, що дає змогу локалізувати та в подальшому удосконалити ці варіанти.

Обмеження запропонованого підходу пов'язані з потребою у великій кількості трас альтернативних варіантів виконання процесу з репрезентативним розподілом подій для обчислення оцінок можливості/необхідності. Крім того, лог не містить окремого маркування для неявних вхідних даних (прихованих факторів), що ускладнює побудову пояснення.

Подальші дослідження спрямовані на проведення класифікації прихованих факторів з метою підвищити точність оцінювання можливості та необхідності. Крім того, доцільно інтегрувати у запропонований підхід графові нейронні мережі для оперативного виявлення нових темпоральних патернів для великих журналів подій.

8. Висновки

Запропоновано причинно-наслідкову модель пояснення в інтелектуальній інформаційній системі, яка на глобальному рівні містить можливісні причинно-наслідкові залежності між явними й неявними вхідними даними та рішенням системи, на процесному рівні містить можливісні причинно-наслідкові залежності між діями процесу формування рішення і отриманим рішенням, на локальному рівні містить детерміновані обмеження щодо допустимих послідовностей дій із процесу формування рішення. У практичному плані модель дає можливість в рамках пояснення визначити вхідні дані як причини отриманого рішення для зовнішніх користувачів та дії процесу формування рішення як причини цього рішення для внутрішніх користувачів з тим, щоб підвищити довіру користувачів до отриманих рішень.

Запропоновано метод побудови пояснень на основі причинно-наслідкової моделі, який містить етапи визначення користувачів, побудови пояснень на глобальному, локальному та процесному рівнях, а також формування комплексного пояснення, що дає можливість підвищити ефективність функціонування інтелектуальної інформаційної системи, а також ефективність використання її рішень при вирішенні практичних задач користувачів.

Перелік посилань:

1. Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 267, 1-38. https://doi.org/10.1016/j.artint.2018.07.007

2. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. ACM Computing Surveys, 54(6), 1-35.

3. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). IEEE Access, 6, 52138-52160. https://doi.org/10.1109/ACCESS.2018.2870052

4. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), 1-42.

5. Chalyi, S., Leshchynskyi, V., & Leshchynska, I. (2019). Method of forming recommendations using temporal constraints in a situation of cyclic cold start of the recommender system. EUREKA: Physics and Engineering, 4, 34-40. https://doi.org/10.21303/2461-4262.2019.00952

6. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). «Why should I trust you?» Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135-1144.

7. D. Gunning і D. Aha, (2019) «DARPA's Explainable Artificial Intelligence (XAI) Program», AI Magazine, 40 (2), 44-58, https://doi.org/10.1609/aimag.v40i2.2850.

8. Arrieta, A. B., Díez-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A.,. & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82-115.

9. Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems, 4765-4774.

10. Chalyi, S., & Leshchynskyi, V. (2020). Temporal representation of causality in the construction of explanations in intelligent systems. Advanced Information Systems, 4(3), 113-117. https://doi.org/10.20998/2522-9052.2020.3.16

11. Чалий С. Ф., Лещинський В. О., Лещинська І. О. (2021) Контрфактуальна темпоральна модель причинно-наслідкових зв'язків для побудови пояснень в інтелектуальних системах,/ Вісник Національного технічного університету «ХПІ». Сер. : Системний аналіз, управління та інформаційні технології = Bulletin of the National Technical University «KhPI». Ser.: System analysis, control and information technology: зб. наук. пр. – Харків : НТУ «ХПІ», 2 (6), С. 41-46.

12. Chala O. (2018) Models of temporal dependencies for a probabilistic knowledge base. Econtechmod. An International Quarterly Journal. 7 (3), 53–58.

13. Levykin V., Chala O. (2018). Development of a method of probabilistic inference of sequences of business process activities to support business process management. Eastern-European Journal of Enterprise Technologies. 5/3(95), 16-24. https://doi.org/10.15587/1729-4061.2018.142664.

14. Чала О. В. (2020) Модель узагальненого представлення темпоральних знань для задач підтримки управлінських рішень. Вісник Національного технічного університету «ХПІ». Системний аналіз, управління та інформаційні технології. 1(3), 14-18. https://doi.org/10.20998/2079-0023.2020.01.03.

15. Chala O. (2018) Models of temporal dependencies for a probabilistic knowledge base. Econtechmod. An International Quarterly Journal. 7, 3, 53–58.

16. Pearl, J., & Mackenzie, D. (2018). The book of why: the new science of cause and effect. Basic books.

17. Imbens, G. W., & Rubin, D. B. (2015). Causal inference in statistics, social, and biomedical sciences. Cambridge University Press..

18. Sloman, S. A. (2005). Causal models: How people think about the world and its alternatives. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780195183115.001.0001

19. Chalyi, S., & Leshchynskyi, V. (2020). Causal reasoning in the construction of explanations in intelligent systems. Advanced Information Systems, 4(4), 105-110. https://doi.org/10.20998/2522-9052.2020.4.14

Надійшла до редколегії 25.05.2025 р.

Чалий Сергій Федорович, доктор технічних наук, професор, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: serhii.chalyi@nure.ua; ORCID: 0000-0002-9982-9091

Лещинський Володимир Олександрович, кандидат технічних наук, доцент, доцент кафедри програмної інженерії ХНУРЕ, м. Харків, Україна, e-mail: volodymyr.leshchynskyi@nure.ua; ORCID: 0000-0002-8690-5702

УДК 004.3, 004.421.5, 004.056.5 DOI: 10.30837/0135-1710.2025.185.063

Д.О. ОСТАПЕЦЬ, А.О. ОПРЯТНИЙ

АНАЛІЗ ДЖЕРЕЛ ЕНТРОПІЇ ДЛЯ ГЕНЕРАЦІЇ ВИПАДКОВИХ ЧИСЕЛ НА БАЗІ МОБІЛЬНИХ ПРИСТРОЇВ

Розглянуто основні існуючі датчики мобільних пристроїв як джерела ентропії для апаратних генераторів випадкових чисел. Сформульовано вимоги до датчиків, які мають використовуватися як джерела ентропії апаратних генераторів випадкових чисел на базі мобільних пристроїв. Проведено порівняльний аналіз датчиків мобільних пристроїв. Для подальшого використання при розробці апаратних генераторів випадкових чисел вибрано датчики акселерометра, гіроскопа і магнітометра.

1. Вступ

Через велику розповсюдженість мобільних пристроїв (смартфонів, планшетів та інших подібних засобів під керуванням мобільних операційних систем) використання їхніх елементів як джерел ентропії для реалізації генераторів випадкових чисел (ГВЧ) є перспективною задачею. Сучасні мобільні пристрої оснащені широкою номенклатурою датчиків, що можуть бути використані як джерела ентропії. З іншого боку, наявність безпровідних інтерфейсів у вказаних пристроях, дозволить використовувати дані з їхніх датчиків для генерування випадкових чисел, наприклад, на стаціонарному персональному комп'ютері. Такий підхід дозволить знизити вартість генератора, в порівнянні з відомими апаратними рішеннями.

Сучасні інформаційні технології широко використовують випадкові послідовності чисел в різних галузях науки та техніки. За допомогою випадкових чисел реалізується велика кількість задач, таких як тестування алгоритмів і систем, імітаційне моделювання, задачі чисельного аналізу, захисту інформації, криптографії та багато інших. При цьому дуже важливо використовувати якісні ГВЧ, оскільки від цього залежить якість одержуваних результатів.

Розрізняють апаратні ГВЧ (true random number generators – TRNG) та алгоритмічні (програмні, табличні) генератори псевдовипадкових чисел (ГПВЧ, pseudorandom number generators – PRNG) [1]. Багато відомих алгоритмічних ГПВЧ не здатні забезпечити необхідні характеристики випадковості та непередбачуваності генерованих послідовностей чисел. В свою чергу, апаратні ГВЧ мають вищу вартість та, як правило, нижчу швидкодію за рахунок необхідності перетворення аналогової інформації в цифрову. В апаратних ГВЧ важливо використовувати якісне джерело шуму (ентропії) [1]. Джерело ентропії – це фізичне джерело інформації, вихід якого або здається випадковим, або стає таким після застосування певного процесу фільтрації/дистиляції [2].

Як джерело ентропії ГВЧ пропонується використовувати датчики, якими оснащені сучасні мобільні пристрої. Датчики – це пристрої, які використовуються в мобільних пристроях для розпізнавання різних параметрів навколишнього середовища [3]. Серед великої номенклатури наявних датчиків для реалізації ГВЧ можуть бути застосовані ті, що мають достатньо велику чутливість (датчик має видавати «шум» без будь-яких маніпуляцій з мобільним пристроєм). Крім того, кількість даних, отримуваних за одне звернення до датчика, має істотне значення для підвищення швидкості генерування. Таким чином, виникає необхідність проведення порівняльного аналізу датчиків, які можуть бути застосовані як джерела ентропії в апаратних ГВЧ.

2. Аналіз літературних джерел та визначення проблеми дослідження

В сучасних мобільних пристроях доступно достатньо багато вбудованих датчиків, які виконують різні допоміжні функції. В основному такі датчики використовуються для проведення оцінки дій користувача та стану середовища, але, за певних умов, можуть слугувати джерелами ентропії. Наприклад, в [4] зазначено, що ГВЧ, який використовує датчики сучасного мобільного пристрою, здатний створювати високоякісні послідовності випадкових бітів. Разом з тим, датчики, подібні до датчиків мобільних пристроїв, також використовуються в пристроях Інтернету речей (Internet of Thihgs – IoT), безпілотних апаратах (дронах) тощо.

В [5] проаналізовано дані, отримані з датчиків руху та інших різних типів датчиків, які знаходяться на більшості плат, що використовуються для вузлів IoT. Аналіз виконується за трьома сценаріями (нормальний, динамічний і насичення), залежно від специфіки кожного датчика. Для оцінки рівня ентропії, автори використали методологію NIST. Беручи до уваги те, що дані є упередженими, вони використали методологію оцінки, яка використовується для незалежно та однаково розподілених джерел ентропії з використанням чотирьох оцінок (частотний тест, тест на зіткнення, частковий збір і тест на стиснення), ентропію Шеннона та мінімальну ентропію. Значення вихідної ентропії вважалося мінімальним із значень ентропії, отриманих від оцінювачів. У результаті аналізу було виявлено, що ентропія Шеннона та мінімальна ентропія навіть у п’ять разів вищі, ніж отримані іншими оцінювачами. Це може пояснити відмінності в значеннях ентропії, про які повідомляють інші автори. Хоча аналіз ентропії, проведений у цьому дослідженні, найближчий до методології NIST, кілька аспектів можуть впливати на правильність результатів. У цьому дослідженні було використано чотири оцінки порівняно з остаточною версією рекомендацій NIST, де використовуються десять оцінок. Кількість даних, зібраних для кожного експерименту, неоднакова, і це впливає на розрахункові значення ентропії. У деяких випадках з різних причин автори не могли зібрати достатню кількість даних, необхідних для хорошої оцінки ентропії.

Можливість реалізації ГВЧ з використанням даних, отриманих з акселерометра, досліджено в роботі [6]. Крім того, автори представили рішення з використанням RFID-мітки, реалізоване на пристрої з обмеженими ресурсами. Автори продемонстрували, що акселерометр генерує ентропію, навіть тоді, коли він використовується в стаціонарному режимі, і що він стійкий до різноманітних змін навколишнього середовища та агресивних маніпуляцій. Рівень ентропії розраховується для різних типів рухів, рівнів шуму, частот дискретизації, температур і навіть резонансної частоти датчика. Результати дослідження привели до висновку, що найнижчий рівень ентропії досягається, коли датчик знаходиться в стаціонарному стані.

Авторами [7] продемонстровано, що тип вимірювань освітлення навколишнього середовища та сенсорні пристрої є джерелами випадковості, особливо що стосується молодших бітів, витягнутих із виміряних даних. Отримані результати показали, що найбільші значення ентропії мали місце при використанні датчиків температури та світла. Крім того, у випадку, коли датчики були запечатані в чорному ящику, виявлено, що ентропія залежить від електронних пристроїв, за допомогою яких проводяться вимірювання. Експерименти показали високі значення ентропії на малих серіях вимірювань при використанні високоінерційних датчиків, таких як датчики температури та освітлення.

В [8] запропоновано ГВЧ, спеціально розроблений для безпілотних літальних апаратів, із вихідним кодом, отриманим на основі датчика даних, який використовується для польоту, – гіроскопа. Автори використовували лише біти з ентропією як вихідні дані

для алгоритму ГВЧ. Для цього було визначено положення бітів, які генерують ентропію, в залежності від стану дрону: «нерухомий» або «політ». Автори також порівнювали класичні ГВЧ, що використовуються для дронів, із запропонованим рішенням.

Крім того, у [8], розглянуто ГВЧ на основі датчика барометра. Для отримання випадкових чисел також використовуються біти з ентропією.

Таким чином, аналіз літературних джерел показує, що відомі спроби використання окремих датчиків мобільних пристроїв для реалізації ГВЧ. При цьому авторами не вирішувалася проблема вибору конкретного типу або декількох типів датчиків з усієї їх множини, застосування яких є найефективнішим.

Як видно з наведених даних, існує проблема вибору датчиків мобільних пристроїв для побудови та реалізації ГВЧ на їх основі.

3. Мета і задачі дослідження

Метою дослідження є розробка рекомендацій з вибору датчиків мобільних пристроїв для використання як джерела ентропії в апаратних ГВЧ на основі порівняльного аналізу їхніх основних характеристик та аналізу якості отриманих випадкових чисел. Результати дослідження можуть бути використані при реалізації засобів генерування випадкових чисел.

Відповідно до мети дослідження, поставлено такі задачі:

– огляд існуючих датчиків мобільних пристроїв;

– формування вимог до датчиків, які мають використовуватися як джерела ентропії ГВЧ;

– проведення порівняльного аналізу характеристик датчиків мобільних пристроїв;

– оцінка якості отриманих випадкових чисел та надання рекомендацій щодо використання датчиків як джерел ентропії ГВЧ.

4. Матеріали і методи дослідження

Об’єктом дослідження є процес генерування випадкових чисел з використанням датчиків мобільних пристроїв як джерела ентропії.

Головною гіпотезою даного дослідження є те, що різні датчики мобільних пристроїв при використанні їх як джерела ентропії можуть продемонструвати різну ефективність процесу генерування випадкових чисел та якість отриманих випадкових чисел.

Огляд існуючих датчиків проводився для пристроїв під керуванням мобільних операційних систем сімейства Android та iOS, таких як смартфони та планшети.

Основні датчики мобільних пристроїв, які можна використати для генерації випадкових чисел, наведено на рис. 1.

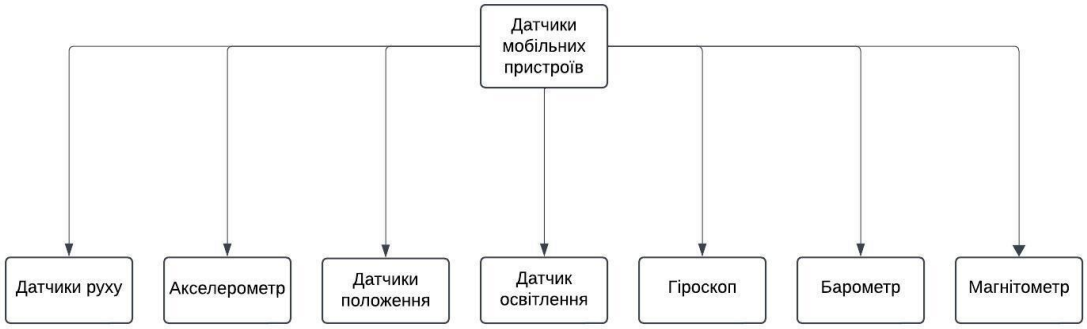


Рис. 1. Основні датчики, доступні у мобільних пристроях

Датчики руху – це датчики які використовуються для моніторингу руху пристрою. Дані рухи можуть бути такими: нахил, струс, обертання або коливання [9]. Смартфони

ідентифікують свою орієнтацію за допомогою акселерометра. Датчики руху, присутні в акселерометрі, можуть використовуватися для виявлення землетрусів або в медичних пристроях.

Акселерометр – це прилад, що вимірює силу реакції, яка індукована прискоренням або гравітацією [9]. У мобільних пристроях акселерометри здебільшого використовуються для керування орієнтацією зображення на екрані (книжковою чи альбомною). Ще одним способом застосування акселерометра може бути виконання мобільним пристроєм деяких функцій при зміні орієнтації у просторі (струшування, удар, поворот дисплею та деякі інші).

Мобільний пристрій, як правило, має два датчики, що дозволяють визначати його фізичне положення, – датчик магнітометра, комбінований з датчиком акселерометра [10].

Виробники мобільних пристроїв зазвичай використовують датчик положення, щоб з'ясувати, коли пристрій тримається близько до обличчя користувача, наприклад, під час розмови. Ці датчики корисні для визначення фізичного положення пристрою. Наприклад, можна використовувати геомагнітний датчик в поєднанні з акселерометром для визначення положення пристрою відносно північного магнітного полюса [11].

Датчик освітлення – це датчик, який регулює рівень яскравості екрана [12]. Він доступний як в дешевих, так і в дорогих мобільних пристроях. Якщо перевести пристрій у режим автоматичної яскравості, то яскравість екрана автоматично підвищуватиметься пропорційно збільшенню яскравості світла, що падає на екран пристрою. Коли ж пристрій залишиться в темряві, яскравість екрану буде мінімальною, тобто в залежності від інтенсивності світла, цей датчик керує яскравістю екрану.

Датчик гіроскопа відповідає за вимірювання швидкості обертання навколо вісі пристрою [9]. Однією з найкращих реалізацій гіроскопа є можливість плавного обертання та виконання кількох команд в іграх за допомогою 3D-рухів.

Датчик барометра використовується для визначення тиску навколишнього середовища [12]. Наприклад, програма здоров'я на смартфонах також використовує ці датчики. При підйомі сходами або переході відносно землі з одного рівня на інший кожна зміна положення точно визначається датчиком барометра, а дані надсилаються на GPS.

Магнітометр – це датчик, який вимірює магнітне поле [10]. Існує два типи магнітометрів: стаціонарні, які використовуються для вимірювання у фіксованих точках, і мобільні, які використовуються в програмах, де потрібне виявлення руху. Беручи до уваги цей аспект, можна зрозуміти, чому датчики з достатньо високою чутливістю можуть генерувати ентропію.

Для проведення експериментів розроблено програмне забезпечення, яке дозволяє отримувати дані з датчиків мобільного пристрою та передавати їх на комп'ютер, на якому, в свою чергу, на їх основі формуються послідовності випадкових чисел. Розроблене програмне забезпечення використовувалося на комп'ютері MacBook Pro M3 та смартфоні IPhone 14. Використаний програмно-технічний комплекс має клієнт-серверну архітектуру. Обмін між елементами комплексу здійснюється мережею WiFi за протоколом TCP. Крім того, розроблено протокол обміну між клієнтською і серверною частинами, який передбачає запит на встановлення режиму генерації і запит на генерацію випадкового числа, а також можливі відповіді на вказані запити. Результати генерування послідовностей випадкових чисел в різних режимах зберігаються в файли для подальшого аналізу.

5. Результати дослідження

Під час проведення порівняльного аналізу було висунуто припущення, за яким датчики, що вимірюють характеристики навколишнього середовища з використанням

властивостей мобільності та знання про місце перебування (магнітне поле наближення, географічні координати тощо), є найкращими для застосування як джерела ентропії в ГВЧ.

При виборі джерела ентропії для розробки ГВЧ на базі мобільного пристрою, на думку авторів, потрібно врахувати такі вимоги:

- чутливість датчика;
- наявність датчика у більшості мобільних пристроїв;
- швидкість оцифровування даних;
- кількість отриманих бітів за одне вимірювання.

Порівняння характеристик основних датчиків мобільних пристроїв як джерел ентропії наведено у табл. 1.

Таблиця 1

Порівняння датчиків мобільних пристроїв як джерел ентропії

Назва датчика	Тип отримуваної інформації	Кількість бітів на одне вимірювання	Доступність в пристроях	Відносна чутливість датчика
Акселерометр	Прискорення в трьох координатах (x, y, z)	16-32 для кожної координати	+	Висока
Датчик освітлення	Інтенсивність освітлення (люкс)	8-16	+	Середня
Гіроскоп	Кутові швидкості в трьох координатах (x, y, z)	16-32 для кожної координати	+	Висока
Барометр	Атмосферний тиск (бар)	16-32	Не в усіх пристроях	Низька
Магнітометр	Магнітне поле в трьох координатах (x, y, z)	16-32 для кожної координати	+	Середня

Акселерометр, гіроскоп та магнітометр наявні майже у всіх сучасних мобільних пристроях, є досить чутливими, а також дають можливість отримання найбільшої кількості бітів тому, що фіксують зміни у трьох координатах. Разом з тим, датчик освітлення та барометр фіксують лише одне значення. Тому саме датчики акселерометра, гіроскопа та магнітометра було обрано як джерела ентропії для ГВЧ.

За допомогою акселерометра, гіроскопа та магнітометра можна вимірювати значення у трьох координатах (x, y, z). Таким чином, для генерації послідовностей бітів авторами були розглянуті можливості використання таких операцій над молодшими бітами величин x, y, z: конкатенація, сумування, додавання за модулем 2 (XOR). Для кожної з вказаних операцій використано 1, 2, 4, 8, 16 та 32 біти. Мобільний пристрій перебував в нерухомому стані.

В результаті аналізу якості отриманих послідовностей для реалізації ГВЧ запропоновано використовувати операцію XOR 4 молодших бітів для акселерометра, операції XOR 16 молодших бітів для гіроскопа та операції XOR 8 молодших бітів для магнітометра.

Для перевірки якості ГВЧ з використанням вибраних датчиків, згенеровано 2000 чисел по модулю 1024 в таких режимах:

– Режим 1. Генерація випадкових чисел за допомогою акселерометра;

– Режим 2. Генерація випадкових чисел за допомогою гіроскопа;

– Режим 3. Генерація випадкових чисел за допомогою магнітометра;

– Режим 4. Генерація псевдовипадкових чисел за алгоритмом BBS із випадковим зерном, згенерованим за допомогою акселерометра;

– Режим 5. Генерація псевдовипадкових чисел за алгоритмом BBS із випадковим зерном, згенерованим за допомогою гіроскопа;

– Режим 6. Генерація псевдовипадкових чисел за алгоритмом BBS із випадковим зерном, згенерованим за допомогою магнітометра.

Вказані послідовності чисел перевірялися за допомогою кейсу тестів NIST [2] (див. табл. 2). Тестування проводилося з застосуванням 10 тестів з 15 можливих. Через невідповідність вимогам тестів щодо довжини не застосовувалися тести Overlapping Template Matching Test, Maurer's «Universal Statistical» Test, Linear Complexity Test, Random Excursions Test, Random Excursions Variant Test (позначені в табл. 2 сірим кольором). Номери режимів в табл. 2 відповідають вказаним вище; символ «+» в комірці таблиці відповідає проходженню тесту.

Таблиця 2

Результати тестування згенерованих послідовностей чисел

Назва тесту	Номер режиму генерування					
	1	2	3	4	5	6
Frequency (Monobit) Test	+	+	+	+	+	+
Frequency Test within a Block	+	+	+	+	+	+
Runs Test	+	+	+	+	+	+
Test for the Longest Run of Ones in a Block	+	+	+	+	+	+
Binary Matrix Rank Test	+	+	+	+	+	+
Discrete Fourier Transform (Spectral) Test	+	+	+	+	+	+
Non-overlapping Template Matching Test	+	+	+	+	+	+
Overlapping Template Matching Test						
Maurer's «Universal Statistical» Test						
Linear Complexity Test						
Serial Test	+	+	+	+	+	+
Approximate Entropy Test	+	+	+	+	+	+
Cumulative Sums (Cusum) Test	+	+	+	+	+	+
Random Excursions Test						
Random Excursions Variant Test						

Результати тестування (табл. 2) показали, що послідовності чисел, згенеровані в усіх шести вказаних режимах, мають ознаки випадковості (оскільки всі тести в усіх випадках пройдено).

6. Обговорення результатів дослідження

Проведене дослідження показало, що отримані послідовності чисел в усіх запропонованих режимах відповідають вимогам до випадкових послідовностей, відповідно до [2].

При цьому генерація псевдовипадкових чисел за алгоритмом BBS із випадковим зерном (режими 4-6) відбувається, принаймні, на три порядки швидше, ніж в інших режимах. Серед режимів з безпосереднім використанням датчиків (режими 1-3) найбільша швидкість генерування спостерігається в режимі генерування за допомогою гіроскопа (режим 2). Це пов'язано з тим, що в цьому режимі використовується порівняно більша кількість молодших бітів – 16, тобто за одну операцію можна отримати одразу 16 бітів випадкового числа.

Таким чином, в додатках, критичних до швидкості, рекомендується використовувати режими генерації псевдовипадкових чисел за алгоритмом BBS із випадковим зерном, згенерованим за допомогою акселерометра, гіроскопа або магнітометра. В разі необхідності застосовувати саме випадкові числа, рекомендується використовувати режим генерації за допомогою гіроскопа, як описано вище.

Надані рекомендації дозволяють значно скоротити час генерування випадкових чисел в порівнянні з описаними в доступних джерелах.

Як обмеження результатів можна зазначити те, що тестування отриманих послідовностей чисел проводилося з використанням не всіх можливих тестів з кейсу NIST [2] через вимоги до їхньої довжини.

Виходячи з цього, в подальшій перспективі потрібно провести дослідження якості отриманих на ГВЧ з використанням датчиків мобільних пристроїв послідовностей різної довжини, які задовольняють вимогам всіх зазначених вище тестів.

7. Висновки

В результаті дослідження сформовано рекомендації щодо вибору датчиків мобільних пристроїв для використання як джерела ентропії в апаратних ГВЧ на основі порівняльного аналізу їхніх основних характеристик та аналізу якості отриманих випадкових чисел.

Проведено огляд існуючих датчиків мобільних пристроїв, що можуть використовуватися як джерела ентропії (зокрема, датчик акселерометра, датчик положення, датчик освітлення, датчик гіроскопа, датчик барометра, датчик магнітометра).

Сформовано основні вимоги до датчиків, які мають використовуватися як джерела ентропії ГВЧ, серед яких чутливість датчика, наявність датчика у більшості мобільних пристроїв, кількість отриманих бітів за одне вимірювання.

Проведено порівняльний аналіз характеристик датчиків мобільних пристроїв, за результатами якого обрано датчики акселерометра, гіроскопа і магнітометра для подальшого використання як джерела ентропії в апаратному ГВЧ.

Виконано оцінку якості отриманих випадкових чисел з використанням датчиків мобільних пристроїв. Надано рекомендації щодо режимів використання вказаних датчиків як джерел ентропії ГВЧ.

Перелік посилань:

1. Ostapets D., Dziuba V., Ivin P. (2024). Hardware random numbers generator based on microcontroller, MATEC Web of Conferences 390, 04002. doi: 10.1051/matecconf/202439004002
2. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. URL: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf.
3. Sensors Overview. URL: https://developer.android.com/develop/sensors-and-location/sensors/sensors_overview.
4. Introduction to Randomness and Random Numbers. URL: https://www.random.org/randomness/.
5. Hennebert C., Hossayni H., Lauradoux C. (2013). Entropy harvesting from physical sensors, Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, 149–154. doi:10.1145/2462096.2462122
6. Voris J., Saxena N., Halevi T. (2011). Accelerometers and Randomness: Perfect Together, Fourth ACM Conference on Wireless Network Security, 115–126. doi: 10.1145/1998412.1998433
7. Pawlowski M. P., Jara A., Ogorzalek M., Jara A. J. (2015). Harvesting Entropy for Random Number Generation for Internet of Things Constrained Devices Using On-Board Sensors. doi: 10.3390/s151026838
8. Cho S.M., Hong E., Seo S.H. (2020). Random Number Generator Using Sensors for Drone. IEEE Access. doi: 10.1109/ACCESS.2020.2972958
9. Motion sensors. Android Developers. URL: https://developer.android.com/develop/sensors-and-location/sensors/sensors_motion.
10. Position sensors. Android Developers. URL: https://developer.android.com/develop/sensors-and-location/sensors/sensors_position.

11. Suciu A., Lebu D., Marton K. (2011). Unpredictable Random Number Generator Based on Mobile Sensors. IEEE International Conference on Intelligent Computer Communication and Processing. doi: 10.1109/ICCP.2011.6047913
12. Environment sensors. Android Developers. URL: https://developer.android.com/develop/sensors-and-location/sensors/sensors_environment.

Надійшла до редколегії 26.05.2025 р.

Остапець Денис Олександрович, кандидат технічних наук, доцент, доцент кафедри ЕОМ УДУНТ, м. Дніпро, Україна, e-mail: odaua@i.ua, ORCID: https://orcid.org/0000-0003-1778-7770 (науковий керівник здобувача вищої освіти Опрятного А.О.)
Опрятний Артур Олександрович, здобувач вищої освіти кафедри ЕОМ УДУНТ, м. Дніпро, Україна, e-mail: artur.opriatnyi@icloud.com, ORCID: https://orcid.org/0000-0001-7145-9677

УДК 519.21:004.021 DOI: 10.30837/0135-1710.2025.185.070

В.В. КРАСНІКОВ, П.Е. СИТНІКОВА

АЛГОРИТМ ЗАВОЮВАННЯ ДЛЯ СТОХАСТИЧНОГО ЗАПОВНЕННЯ ДВОВИМІРНИХ ДИСКРЕТНИХ РЕШІТОК ЗВ'ЯЗАНИМИ ОБЛАСТЯМИ

Запропоновано алгоритм завоювання, який дає змогу швидко заповнювати двовимірні поля неперетинними зв'язними областями заданих площ. Це робить можливим ефективну процедурну генерацію наборів даних. Роботу алгоритму продемонстровано на прикладі створення ігрового поля для модифікованої версії класичної задачі N-Queens. У підсумку набір даних генерується повністю процедурно, а ймовірність отримати дві ідентичні генерації прямує до нуля. Це відкриває можливість використовувати запропонований алгоритм у складніших системах, зокрема, для створення синтетичних зображень, ігрових рівнів або тестових вибірок.

Доведено ефективність розробленого алгоритму.

1. Вступ

При вирішенні багатьох сучасних прикладних задачах виникає потреба у процедурному заповненні двовимірних полів (матриць, решіток) зв'язними блоками різного розміру. Як очікувані результати таких задач можуть розглядатися тестові набори даних, процедурно згенеровані ігрові рівні, синтетичні набори для аугментації моделей машинного навчання, біомедичні мікроматриці тощо. Заповнення полів – це структурування простору, що створює універсальний інструмент для різних типів задач. В геоінформатиці й картографії це може бути генерація зон для карт – полігонів, кластерів, типів місцевості, при обробці зображень – сегментація зображень на зв'язні області. Задача має практичне або теоретичне застосування для багатьох галузей.

Для задоволення зазначених потреб зазвичай використовують алгоритми процедурної генерації та стохастичного моделювання, що поєднують різні підходи до отримання максимально різноманітних випадкових розподілів. Ці алгоритми відрізняються способом представлення даних і такими характеристиками, як трудомісткість та час виконання. Попри схожий напрям, кожен алгоритм формує власний тип результату, а вибір конкретного алгоритму залежить від вимог користувача й обмежень задачі.

Проблема вибору алгоритму процедурної генерації поля із заданими характеристиками є нетривіальною, оскільки жоден із наявних підходів не забезпечує гарантованої відповідності до цільових параметрів, таких як кількість зв'язаних блоків, їхні форми та просторове розташування. Це, у свою чергу, обумовлює потребу у додаткових дослідженнях, спрямованих на аналіз обмежень існуючих алгоритмів, розробку критеріїв порівняння та створення нових методів генерації, які б забезпечували

контрольоване структурування простору.

Актуальність роботи підтверджується наявністю цілого класу задач, які потребують адаптивного структурування двовимірного простору з дотриманням складних умов зв'язності та сегментації, що на сьогодні не можуть бути повноцінно розв'язані з використанням жодного із класичних алгоритмів.

2. Аналіз існуючих алгоритмів і постановка проблеми

Розглянемо модифіковану версію класичної задачі N-Queens (далі N-Queens). У класичному формулюванні мета задачі – розмістити 8 ферзів на звичайній шахівниці так, щоб у кожному рядку, стовпці й діагоналі стояв рівно один ферзь.

У модифікованому варіанті задачі N-Queens кількість ферзів дорівнює N на полі розміром N x N. Саме поле розфарбовано у N кольорів. Треба розставити ферзів таким чином, щоб у кожному кольорі опинився рівно один ферзь. Дозволяється ставити ферзів на одну діагональ, якщо вони не сусідять по діагоналі, але не на один рядок або стовпчик. Клітини поля мають бути розфарбовані з ознакою ортогональної суміжності (клітини вважаються ортогонально суміжними, якщо вони межують по горизонталі або вертикалі). Модифікація задачі N-Queens є активною дослідницькою темою, що передбачає різноманітні модифікації класичної постановки з урахуванням додаткових обмежень, евристик та алгоритмічних стратегій, як, наприклад, в [1].

Постає задача процедурної генерації розфарбованого поля таким чином, щоб задача розміщення ферзів була розв'язною.

Для такого стохастичного розфарбовування поля, яке можна подати у вигляді двовимірної матриці, а кольори – у вигляді цифр, існує кілька відомих алгоритмів та підходів:

– градієнтний шум Перліна;

– клітинні автомати;

– хвильова заливка (queue flood fill);

– випадкове блукання (random walk).

Реалізація градієнтного шуму Перліна полягає в тому, що на полі розміром N x N в кожній клітині генеруються випадкові вектори-градієнти [2]. Значення в інших клітинах отримують за допомогою плавної інтерполяції в залежності від вектора. Після цього вводять поріг P у відсотках. Всі клітини, що мають значення вище порогу можуть бути зафарбовані, наприклад, чорним кольором, а нижче порогу – білим [3].

Алгоритм забезпечує гладкі контури та легко регулює масштаб «зернистості». Проте площа зон цілком залежить від одного-єдиного значення – порогу. Цей алгоритм розфарбовує поле у два кольори. Якщо потрібні N різних кольорових зон, доведеться вводити N–1 поріг, а спрогнозувати, скільки клітин відбере кожен колір, практично неможливо [4].

Отримані надмірно гладкі межі знижують якість згенерованого поля, що віддаляє його від ряду реальних задач.

Роботу алгоритму з використанням клітинних автоматів можна описати у два етапи [5]. На першому етапі створюється поле N × N, де значення клітини задається випадково як 0 або 1 з початковою щільністю відношення кількості одиниць до розміру всього поля – ρ %. На другому етапі алгоритм змінює стан поточної клітини, спираючись на значення ортогональних сусідів. Для кожної клітини обчислюють правило переходу в інший стан в залежності від кількості ортогональних сусідів, значення яких дорівнюють одиниці. При цьому на вхід алгоритму можна подати число ітерацій зміни поточного значення клітини.

Алгоритм використовується для деяких задач стохастичного заповнення квадратного поля, але для вирішення описаної проблеми він не підходить: по-перше, він не гарантує

зв’язності отриманих областей, а по-друге, обмежений зафарбовуванням двома кольорами.

При використанні хвильової заливки (queue flood fill) для кожного кольору вибирається коренева клітина й додається до черги [6]. Крім того, задається ліміт кількості клітин, що розфарбовується одним кольором. Поки черга не порожня і ліміт не вичерпано, з черги виймається клітина, переглядаються значення її ортогональних сусідів. Якщо сусід нерозфарбований, він фарбується тим самим кольором і ставиться в кінець черги. Алгоритм простий в реалізації, забезпечує зв’язність зафарбованих областей і дає можливість задати ліміт.

Застосування цього алгоритму дає достатньо рівномірне розфарбовування, і передбачає багатократне перефарбовування клітин, що в підсумку може привести до нерозв’язності задачі, що розглядається. Крім того, при повному співпадінні стартових даних – ліміт, позиція стартових кольорів, черга розфарбовування – результат буде на 100 % детермінованим, що протирічить вимозі про стохастичність результату алгоритму.

Принцип випадкового блукання (random walk) полягає в такому. Спочатку на полі N x N випадковим чином розставляється m (m=(2,N)¯) точок – «мурах» [7]. На кожному кроці «мураха» випадковим чином обирає одне з чотирьох ортогональних напрямків з однаковою ймовірністю й переходить у сусідню клітину, зафарбовуючи її власним кольором.

Цей алгоритм дозволяє розфарбувати поле в m кольорів. При цьому він має декілька критичних для розв’язання зазначеної проблеми недоліків: не гарантує зв’язності, може залишати незаповнені клітини, а «мурахи» здатні перефарбовувати клітини одна одної.

Проведене дослідження показало, що жоден із розглянутих алгоритмів генерації кольорового поля не гарантує створення конфігурації для розв’язання модифікованої задачі N-Queens.

Таким чином, жоден із згаданих підходів не формує поле, що задовольняє вимогам, потрібним для подальшого розв’язання цієї задачі. Це підкреслює актуальність розробки нового алгоритму генерації, здатного керовано створювати ортогонально області з контрольованим числом кольорів і відповідною геометрією.

3. Мета і задачі дослідження

Метою дослідження є розробка алгоритму, який дозволив би розфарбовувати двовимірні поля розміром N x N в N кольорів при виконанні таких вимог:

– повне заповнення (жодних порожніх клітин);

– зв’язність (розфарбована область має бути ортогонально зв’язною);

– різноманітний контур (якомога більш ламані форми);

– стохастичність (можливість одержати дві однакові розфарбовані форми поля при однакових стартових даних наближається до 0);

– фіксовані витрати (час і пам’ять залишаються не гірші за $O(N^2)$).

Для досягнення поставленої мети необхідно розв’язати такі задачі:

– розробити формальний набір метрик, що дозволяє кількісно оцінити виконання вимог до алгоритму, включаючи ступінь зв’язності, геометричну різноманітність, рівень стохастичності та ресурсну ефективність.

– сконструювати та протестувати алгоритм генерації, що забезпечує задані властивості при розфарбуванні поля.

4. Матеріали дослідження

Об’єктом дослідження є процес процедурного розфарбовування двовимірного дискретного простору з топологічними обмеженнями.

Предметом дослідження є алгоритмічні підходи до генерації кольорової конфігурації поля, що забезпечує структурну зв’язність та відповідність до заданих умов задачі.

Основною гіпотезою дослідження є припущення, що застосування модифікованого алгоритму генерації поля з урахуванням ортогональної суміжності та контролю кількості кольорових компонент дозволяє створити конфігурацію, придатну для розв'язання задачі оптимального розміщення елементів з додатковими умовами.

Для розроблюваного алгоритму було обрано назву алгоритм завоювання. Для алгоритму введено поняття «агент» і «завоювання». Агент – це сутність, якій має відповідати окрема територія двовимірного поля, а завоювання – процес почергового заповнення клітин поля різними агентами. Алгоритм має такі властивості:

– агенти можуть пропускати ходи. Ймовірність пропустити хід задається окремою функцією або константою;

– кожний доступний агенту напрям має свою вагу, яка впливає на ймовірність вибору напряму;

– агенти не мають права захоплювати клітини, які вже належать іншим агентам.

Це дозволяє впливати на роботу алгоритму за допомогою різних функцій керування, тобто завдання власних формул для пропуску ходу чи калькуляції ваги напряму. А неможливість захоплювати чужі клітини забезпечує зв'язність територій агентів.

5. Результати досліджень

Розроблений алгоритм представлено у вигляді послідовності таких кроків.

Крок 1. На вхід подається матриця, на якій розташовані агенти, що мають номери від 1 до N. Всі порожні клітини заповнені нулями.

Крок 2. Номери агентів випадково сортуються і додаються до черги, за якою агенти будуть виконувати ходи.

Крок 2.1. Агент отримує право ходу, як, наприклад, у грі з кидком кубика. Визначається мінімальне і максимальне значення для випадкової генерації. Визначається порогове значення, яке буде дорівнювати кількості клітин вже захоплених агентом територій. Якщо отримане випадкове число вище або дорівнює пороговому значенню, агент може отримати хід, а якщо нижче – пропускає хід.

Крок 2.2. Перевіряються сусідні клітини за ортогональними напрямами навколо агенту, що отримав право ходу. При цьому агент не може ходити по чужих територіях, або за межі поля.

Крок 2.3. Опції, доступні агенту (доступні напрями, серед яких буде обирати агент), отримують свою вагу – абстрактну величину, за якою опції сортуються. Опція кроку на вільну клітину отримує вагу 100, а опція руху на вже захоплену тим самим агентом клітину – 50, що дозволяє одразу відсортувати їх за цією величиною. Здійснюється вибір з урахуванням останнього напряму повороту.

Якщо опцій все ще декілька, вищу вагу отримують клітини, що ближче до кордонів, бо це дозволяє швидше захопити їх і уникнути білих плям, які потім перетворяться на фігури з високим рівнем прямокутності.

Крок 3. Якщо опцій для руху по порожніх клітинах нема, агент намагається знайти найближчі незахоплені клітини і рухатись до них по території, яка йому вже належить. Після кожного ходу агент оновлює власні дані про напрям останнього руху, його теперішню позицію і останній поворот.

Крок 4. Кроки 2 і 3 виконуються ітеративно, поки поле не буде заповнено повністю, тобто не залишиться порожніх клітин (нулів).

При виконанні Кроку 1 слід зазначити, що розташування агентів не є частиною роботи алгоритму, оскільки алгоритм передбачає рух вже існуючих агентів. Для поставленої задачі агентів треба розташувати за правилами задачі N-Queens у вигляді її потенційного рішення – тобто по 1 агенту в горизонталі, вертикалі із виключенням суміжності по

діагоналі.

При виконанні Кроку 2 слід зазначити: одна ітерація алгоритму передбачає, що кожен агент або отримає право ходу, або пропускає хід і передає можливість ходу далі по черзі.

При виконанні Кроку 2.1 мінімальне і максимальне значення для випадкової генерації було обрано, відповідно, 1 та N.

При виконанні Кроку 2.3 задля збереження складності задачі краще, щоб фігури мали різноманітні форми і поле не складалось просто з різних зафарбованих прямокутників. Саме тому, крім вибору за вагою, було додано вибір з урахуванням останнього напряму повороту, тобто якщо в агента є більше одного варіанта захопити нічийну клітину, можна відкинути опцію з поворотом у тому самому напряму.

У даній реалізації обрано було додавати до ваги 20 одиниць, що дозволяє посунути цю опцію вище відповідно до пріоритету, але не є достатнім, щоб рухатись по захоплених клітинах, якщо є можливість рухатись по порожніх.

Для поставленої задачі одною з ключових вимог є вимога отримати різноманітний контур розфарбованої фігури, тобто якомога більш ламані форми. Для оцінки цього та інших параметрів були винайдені такі спеціальні метрики, що дають можливість оцінити результати роботи алгоритму:

– середня витягнутість (Avg elongation) – середнє відношення довжини до ширини описаного прямокутника, де 1 означає квадрат.

– середня прямокутність (Avg rectangularity) – частка площі фігури у площі її мінімального описаного прямокутника, де 1 є повністю заповнений.

– середня зубчастість (Avg roughness) – периметр фігури, нормований на корінь площі. Чим більший показник, тим зубчастіший конутр.

– максимальна витягнутість (Max elongation) – максимальна витягнутість серед усіх фігур розфарбованого поля.

– максимальна прямокутність (Max rectangularity) — яка фігура найбільше заповнює прямокутник.

– мінімальна зубчастість (Min roughness) – найгладкіший контур на полі.

– доля витягнутих фігур (Share elongation) > 2,5 – відсоток фігур, що мають витягнутість понад 2,5. Впродовж виконання замірювань параметр (оптимальний поріг) було виявлено експериментально.

– складність (Steps / N^2) – середня кількість кроків алгоритму, поділена на площу поля. Дає нормовану оцінку обчислювальної складності.

Розроблений алгоритм завоювання було реалізовано програмно з використанням мови C#. Крім того, було реалізовано функціонал, який дозволив зібрати оцінки метрик після певної кількості ітерацій.

Для моделювання розфарбованого поля було використано матрицю, заповнену числами, де однакові числа демонструють певний колір. В процесі тестування було отримано можливість візуально оцінити заповнену зв'язними областями матрицю, створену за допомогою реалізованого алгоритму.

Тестові результати на 1000 прогонів трьох рівнів складності 8x8, 12x12 і 20x20 продемонстровано на рис. 1-3.

6. Обговорення результатів дослідження

Для оцінки результатів було зроблено по 1000 прогонів алгоритму для різних розмірів поля, що дало можливість переконатися в коректності роботи алгоритму та отримати уявлення про прогноз поведінки генерацій на довгій дистанції.

Загалом можна сказати, що алгоритм точно відповідає заданим вимогам: отримане поле повністю заповнено, території агентів зв’язні і мають різноманітний контур. Оскільки кожен

крок випадковий, то ймовірність отримати ідентичний результат майже дорівнює нулю.

```
===== BOARD 8x8 - 1000 runs =====

--- Test result: ---
Avg elongation      : 1.75 + 0.28
Avg rectangularity  : 0.74 + 0.06
Avg roughness       : 5.17 + 0.28
Max elongation      : 3.21
Max rectangularity  : 0.98
Min roughness       : 4.18
Share elong > 2.5    : 11.4%
Steps / N^2          : 3.17
```

Рис. 1. Тестові результати для складності 8x8

```
===== BOARD 12x12 - 1000 runs =====

--- Test result: ---
Avg elongation      : 1.77 + 0.26
Avg rectangularity  : 0.65 + 0.05
Avg roughness       : 5.74 + 0.33
Max elongation      : 3.74
Max rectangularity  : 0.96
Min roughness       : 4.27
Share elong > 2.5    : 12.1%
Steps / N^2          : 5.93
```

Рис. 2. Тестові результати для складності 12x12

```
===== BOARD 20x20 - 1000 runs =====

--- Test result: ---
Avg elongation      : 1.78 + 0.22
Avg rectangularity  : 0.56 + 0.04
Avg roughness       : 6.47 + 0.31
Max elongation      : 4.42
Max rectangularity  : 0.93
Min roughness       : 4.36
Share elong > 2.5    : 12.5%
Steps / N^2          : 13.75
```

Рис. 3. Тестові результати для складності 20x20

Приклад точкової генерації поля 8x8 наведено на рис. 4.

```
5  5  2  2  2  6  6  6
7  5  2  4  4  4  4  4
7  5  3  3  3  3  3  4
7  5  3  3  3  3  1  4
7  5  3  3  3  3  1  1
7  5  5  5  8  8  8  1
8  8  8  8  8  8  8  1
8  8  8  8  8  8  1  1
```

Рис. 4. Приклад точкової генерації поля 8x8

Для оцінки результатів генерацій за зазначеними вище метриками використовувався метод Монте-Карло – багаторазове виконання генерації на однакових вхідних параметрах із подальшим збором і статистичним аналізом метрик. Результати вимірів і їхнього аналізу наведено в табл. 1.

З цієї таблиці і поданих коментарів можна зробити висновок, що всі метрики змінюються прогнозовано, отже за описаними вимогами алгоритм є стабільним і ефективним.

Таблиця 1

Результати вимірів метрик

Назва метрики	Розмір поля			Коментар
	8 × 8	12 × 12	20 × 20	
Average elongation (mean ± σ)	1.75 ± 0.28	1.77 ± 0.26	1.78 ± 0.22	Спостерігається помірне зростання. Більші поля дають трохи витягнутіші області
Average rectangularity	0.74 ± 0.06	0.65 ± 0.05	0.56 ± 0.04	Чіткий спад – контур стає більш схожим на дерево
Average roughness	5.17 ± 0.28	5.74 ± 0.33	6.47 ± 0.31	Лінійне зростання – контур стає зубчастішим
Maximum elongation	3.21	3.74	4.42	На великих полях з'являються довші «коридори»
Maximum rectangularity	0.98	0.96	0.93	Прямокутні фігури менш щільні на великих N
Minimum roughness	4.18	4.27	4.36	Мінімум гладкості контурів повільно зростає
Share elongation > 2.5 (%)	11.4	12.1	12.5	Частка довгих вузьких фігур майже стабільна (погрішність близько до 1%)
Steps / N^2	3.17	5.93	13.75	Підтверджує $O(N^2)$

Слід зауважити, що в запропонованому алгоритмі не реалізовано засоби контролю за формою утворюваних фігур. Таким чином, геометрія територій, що захоплюються агентами, визначається виключно правилами завоювання та випадковістю вибору напрямів.

В подальшому дослідженні можна оптимізувати правила вибору опцій агентами для отримання різноманітніше заповненого поля.

7. Висновки

У дослідженні було розглянуто і проаналізовано проблему стохастичного розфарбовування двовимірних полів. Було проаналізовано існуючі алгоритми і причини, через які їх не завжди можна використовувати в деяких задачах, а також згадані приклади таких задач.

За результатами вирішення поставлених задач дослідження отримано алгоритм завоювання, що демонструє стабільну зв'язність територій агентів та гнучкість налаштувань напрямів руху. Сутність отриманого результату полягає у тому, що дії агентів регулюються функціями ймовірності та вагами можливих опцій ходу, що дозволяє моделювати різні сценарії розподілу територій.

Роботу алгоритму було оцінено за допомогою фіксованих математичних метрик, які довели, що алгоритм зберігає свою надійність і має прийнятну складність при зміні величини сторони поля N.

У порівнянні з класичними методами розподілу, запропонований алгоритм забезпечує природніше формування територій без перетину володінь агентів, що сприяє покращенню узгодженості та уникненню конфліктних зон.

Задача заповнення двовимірних полів є достатньо поширеною і може бути застосована для різних областей. В залежності від напряму задачі, можна варіювати деякі параметри розробленого алгоритму, такі як значення випадкової генерації, порогове значення, пріоритет клітини тощо.

Перелік посилань:

1. Sinha, R., Kaur, N., Gupta, S., Thakur, P. (2025). N-Queen Problem Solution Using Modified Genetic Algorithm. In: Bajaj, A., Sreedhar, S., Abraham, A. (eds) Bio-Inspired Computing. IBICA 2023. Lecture Notes in Networks and Systems, vol. 1230. Springer, pp. 201–210.

2. Perlin K. (1985) An Image Synthesizer. ACM SIGGRAPH Computer Graphics, vol. 19, no. 3, pp. 287–296, https://doi.org/10.1145/325334.325247.

3. Кудінов І. П., Ягодкін Д. (2025). Алгоритми процедурної генерації шуму Перліна. Зб. тез наук. доп. здобувачів вищої освіти Бердян. держ. пед. ун-ту, Запоріжжя, Україна, трав. 2025, с. 127-130. https://doi.org/10.5281/zenodo.15548937.

4. Kopel, M., Maciejewski, G. (2020). Comparison of Procedural Noise-Based Environment Generation Methods. In: Nguyen, N.T., Hoang, B.H., Huynh, C.P., Hwang, D., Trawiński, B., Vossen, G. (eds) Computational Collective Intelligence. ICCCI 2020. Lecture Notes in Computer Science, vol 12496. Springer, Cham. pp 878–887.

5. Wu Z., Mao Y., Li Q. (2021). Procedural Game Map Generation using Multi-leveled Cellular Automata by Machine Learning, in Proc. ISAIMS 2021, Chongqing, China, Oct. 2021, https://doi.org/10.1145/3500931.3500962.

6. Fellows M. R., Rosamond F. A., da Silva M. D., Souza U. S. (2021). A Survey on the Complexity of Flood-Filling Games. Discrete Appl. Math., vol. 304, pp. 233–246, Dec. 2021, https://doi.org/10.1016/j.dam.2021.09.029.

7. Popp S., Dornhaus A. (2023). Ants Combine Systematic Meandering and Correlated Random Walks when Searching for Unknown Resources. iScience, vol. 26, no. 2, 2023, Art. no. 105916, https://doi.org/10.1016/j.isci.2022.105916

Надійшла до редколегії 10.06.2025 р.

Красніков Влад Валерійович, старший .Net розробник, компанія Pricer24, м. Харків, Україна, e-mail: krasnikov.vlad.v@gmail.com

Ситнікова Поліна Едуардівна, кандидат технічних наук, доцент, доцент кафедри системотехніки ХНУРЕ, м. Харків, Україна, e-mail: polina.sytnikova@nure.ua, ORCID: https://orcid.org/0000-0002-6688-4641

УДК 004.8:004.9 DOI: 10.30837/0135-1710.2025.185.077

С.Ф. ЧАЛИЙ, Р. В. КРАВЧЕНКО

ГРАФОВА НЕЙРОННА МЕРЕЖА ДЛЯ ТЕМПОРАЛЬНО УПОРЯДКОВАНИХ ДАНИХ В ЗАДАЧІ ПОБУДОВИ ПОЯСНЕНЬ В ІНТЕЛЕКТУАЛЬНІЙ СИСТЕМІ

Об'єктом дослідження є процес побудови пояснень в інтелектуальних інформаційних системах. Предметом дослідження є моделі та методи формування пояснень в інтелектуальних інформаційних системах. Метою роботи є розробка підходу до побудови пояснень в інтелектуальних системах на основі графових нейронних мереж, які враховують темпоральний порядок у вхідних даних. Розроблено модель графової нейронної мережі для темпорально упорядкованих даних; виконано експериментальну перевірку графової мережі в задачі побудови пояснень для системи електронної комерції. Запропонована модель графової нейронної мережі включає функціональні блоки побудови векторних представлень, виявлення темпоральних патернів з використанням мережі LSTM, формування графа мережі для заданих темпоральних інтервалів, підготовки пояснень, прогнозування, генерації пояснень з використанням механізму уваги та результатів агрегації векторного представлення мережі.

1. Вступ

Інтелектуальні інформаційні системи (ІІС) використовують методи машинного навчання для того, щоб апроксимувати складні залежності в даних й використати отримані залежності для формування рішень. Проте внаслідок непрозорості таких процесів користувачі не завжди можуть зрозуміти логіку формування рішень в ІІС [1], [2]. Непрозорість процесу формування рішень призводить до зниження довіри до результатів роботи ІІС і, як наслідок, може привести до обмеження використання отриманих рішень користувачами [3].

Для вирішення проблеми зниження довіри до ІІС в галузі штучного інтелекту розвивається науковий напрямок пояснювального штучного інтелекту (Explainable Artificial Intelligence – XAI) [4]. Напрямок XAI орієнтований на розробку моделей та методів, що забезпечують пояснення процесу та результатів роботи ІІС, забезпечуючи зрозумілість для

користувачів причин отриманих рішень [5]. Використання пояснень робить процес формування рішення інтерпретованим, що сприяє підвищенню довіри до ІІС [6]. Пояснення зазвичай відображають вплив вхідних даних на рішення ІІС [7]. Проте у випадках використання в ІІС даних із темпоральною упорядкованістю необхідно при побудові пояснень враховувати складні причинно-наслідкові зв'язки та залежності між цими даними у часі [8]. Однак існуючі методи пояснювального штучного інтелекту зазвичай не враховують темпоральну природу даних та темпоральні залежності між вхідними даними [7], що обмежує їхні можливості щодо побудови релевантних пояснень для ІІС, які обробляють послідовності подій або станів [8].Можливість обробки та аналізу складних залежностей в даних, які можуть бути представлені у формі графів, забезпечується з використанням графових нейронних мереж (GNN) [99]. Графова мережа забезпечує можливість аналізу складних залежностей та генерації на цій основі прогнозів й відповідних пояснень щодо процесу функціонування інтелектуальної системи [10], що створює умови для підвищення довіри користувачів до рішень ІІС та розширення сфери практичного застосування цих рішень. Тому проблема побудови пояснень з використанням графової нейронної мережі для темпорально упорядкованих даних є актуальною.

2. Аналіз літературних даних і постановка проблеми дослідження

Дослідження у сфері теорії графових нейронних мереж інтенсивно розвиваються в останнє десятиріччя. Теоретичні основи графових нейронних мереж як засобу обробки даних із складними зв’язками, що представляють статичні й динамічні процеси, запропоновано в [11]. Переваги й обмеження GNN щодо представлення на розрізнення графів при кодуванні суттєвих властивостей графів розглянуто в [12].

Комбінована обробка темпоральних та просторових даних на прикладі задачі прогнозування дорожнього руху розглянута в [13] та [14]. При вирішення даної задачі використовується спеціалізована архітектура темпоральної GNN, яка поєднує графову мережу та рекурентну нейронну мережу. Перша аналізує структуру дорожньої мережі, а друга враховує зміни трафіку з часом. В [15] представлено просторову-темпоральну мережу, в якій просторова складова обробляється з використанням традиційної GNN, а темпоральна – з використанням Long Short-Term Memory (LSTM). Використання механізму уваги для визначення ваги зв'язків між вершинами мережі розглянуто в [16]. Даний механізм дає можливість динамічно обраховувати важливість сусідніх вершин графу при виконанні процедури агрегації. Проте в цілому існуючі підходи до розробки GNN враховують окремі аспекти темпоральних властивостей даних і не забезпечують інтеграцію можливостей обробки змін у вхідних даних із змінами важливості цих даних, а також подальше пояснення виявлених темпоральних залежностей.

Для формування пояснень щодо темпоральних залежностей формуються каузальні правила на основі запропонованих в [18] та [19] поточних залежностей типу «Next» та довгострокових темпоральних залежностей типу «Future». Останнє з правил дає можливість виділити інтервали часу, для яких виявляються локальні закономірності. В [20] запропоновано модель казуальної залежності на основі темпоральної залежності, а в [21] – метод її побудови. В [22] в моделі каузальної залежності враховано альтернативні значення вхідних даних, що дає можливість побудувати пояснення для динамічних систем. Важливість таких залежностей базується на обчисленні частоти спільної появи значень змінних з урахуванням можливих альтернатив. Проте розроблені методи побудови пояснень для даних з темпоральними мітками орієнтовані на виділення локальних темпоральних патернів і не приділяють достатньо уваги формуванню концептуально-орієнтованих пояснень [23]. Останні можуть бути побудовані на основі моделювання ІІС з використанням підходів, що лежать в основі темпоральних GNN, та фрагментації темпоральної упорядкованості даних на основі підходів [18] і [19], оскільки така

комбінація дає можливість враховати темпоральні патерни комплексних зв'язків між вхідними даними та результатом роботи ІІС.

Таким чином, розробка нових підходів до побудови пояснень з використанням графової нейронної мережі для темпорально упорядкованих даних є актуальною задачею.

3. Мета і задачі дослідження

Метою даного дослідження є розробка підходу до побудови пояснень в інтелектуальних системах на основі графових нейронних мереж, які враховують темпоральний порядок у вхідних даних.

Для досягнення поставленої мети вирішуються такі задачі: розробка моделі графової нейронної мережі для темпорально упорядкованих даних; експериментальна перевірка графової мережі в задачі побудови пояснень для системи електронної комерції.

4. Графова нейронна мережа для обробки темпорально упорядкованих даних

Об’єктом даного дослідження є процес побудови пояснень в інтелектуальних інформаційних системах. Предметом дослідження є графові нейронні мережі, які формують пояснення на основі зв'язку між вхідними даними та отриманим рішенням ІІС. Такий зв’язок відображає ступінь впливу значень вхідних даних на рішення ІІС, обґрунтовуючи для користувача причини поточних характеристик сформованого рішення.

Головна ідея запропонованого підходу до побудови графової нейронної мережі полягає у циклічному формуванні графа мережі на визначених інтервалах часу на основі загальних правил навчання. Це дає можливість визначити поточні патерни для темпорально упорядкованих даних і на цій основі побудувати пояснення.

Блочну схему розробленої графової нейронної мережі представлено на рис. 1. Призначення кожного блоку мережі представлено в табл. 1.

Рис. 1. Блочна схема розробленої графової нейронної мережі

Таблиця 1

Призначення блоків графової нейронної мережі для обробки темпорально упорядкованих даних

Блок	Призначення
1. Блок вхідних даних	Приймає упорядковані у часі вхідні дані
2. Блок побудови векторних представлень вхідних даних	Формує векторне представлення вхідних даних
3. Блок виявлення темпоральних патернів	Виділяє патерни ,що відображають зв'язок між вхідними даними у часі; використовується мережа LSTM
4. Блок формування графа мережі	Формує граф мережі для заданого інтервалу часу; вершини графа відображають вхідні об'єкти, а дуги – різнотипні залежності між цими об’єктами
5. Згорткові шари для інформації від сусідніх вершин	Реалізують механізм передачі повідомлень для оновлення представлення вершин з урахуванням інформації від вершин-сусідів
6. Механізм темпоральної уваги	Враховує темпоральні патерни як поточного, так і для сусідніх вузлів графової мережі
7. Блок агрегації	Формує єдине векторне представлення графової мережі на основі представлень її вершин
8. Блок підготовки пояснень	Розраховує вплив вхідних факторів (значень вхідних змінних) на прогнозне значення мережі
9. Блок прогнозування	Перетворює узагальнене представлення графа мережі у скаляр, що відображає прогноз роботи системи, яка моделюється мережею
10. Блок генерації пояснень	Використовує результати роботи механізму уваги та блоку підготовки пояснень для генерації пояснення щодо впливу значень вхідних даних на прогнозний результат

Блок вхідних даних приймає дані X з темпоральними мітками. Зазвичай такі дані відображають послідовність дій певного процесу. Наприклад, послідовність покупок у системі електронної комерції, послідовність станів процесу сервісного обслуговування тощо.

Блок побудови векторних представлень вхідних даних Emb формує багатовимірні векторні представлення для кожного екземпляру вхідних даних. Таке представлення кодує властивості вхідних даних. У процесі вбудовування може бути врахований контекст. Результуюче векторне представлення відображає семантичні відношення між вхідними даними. Елементи відповідного вектору кодують, наприклад, такі властивості товару в системі електронної комерції, як ціна, популярність, категорія тощо. В даному блоці може бути використаний класичний алгоритм вбудовування на основі таблиці пошуку, яка містить код екземпляру та його векторне представлення. Даний підхід забезпечує зменшення часу обробки, оскільки оцінка складності є константною, тобто не залежить від кількості елементів у таблиці: доступ відбувається за індексом. Як альтернативу можна використати модель BERT, яка враховує контекст використання кожного елементу вхідних даних. Розмір отриманого у процесі вбудовування вектору залежить від предметної області задачі, для якої формується графова нейронна мережа. Наприклад, для пояснень у системі електронної комерції можуть бути використані вектори розміром до 100 вимірів, оскільки перелік товарів такої системи є достатньо обмеженим.

Блок виявлення темпоральних патернів $LSTM$ обробляє послідовність отриманих від попереднього блока векторів для виявлення патернів темпоральних залежностей. Даний блок використовує мережу LSTM яка є рекурентною нейронною мережею з можливостями

збереження суттєвої інформації в пам'яті. Дана мережа зберігає довгострокові патерни та адаптується до контексту, відкидаючи неактуальні залежності. Використання даної мережі дає можливість на основі даних з мітками часу отримати темпоральні патерни та використати їх для прогнозування майбутніх залежностей.

Такі патерни в подальшому служать основою для побудови пояснень у графовій нейронній мережі. Наприклад, патерн «Після початку продажів нової моделі смартфону з часом зростають продажі чохлів» дає можливість, з одного боку, прогнозувати ріст продажу чохлів для певної моделі, а з іншого боку – пояснити це збільшення продажів.

Блок формування графа мережі *B* будує граф на основі вхідних даних для визначеного інтервалу часу. Вершини графа відображають вхідні екземпляри (наприклад, товари в системі електронної комерції), а дуги – зв’язки між ними. Вершини можуть бути різних типів в залежності від категорії вхідних даних. Дуги моделюють різні типи зв’язків. Даний блок розраховує нормовані ваги дуг в залежності від частоти спільної появи елементів у вхідному наборі даних.

Згорткові шари для інформації від сусідніх вершин *GCN* призначені для передачі повідомлень між вершинами графа. На основі цих повідомлень кожен вузол оновлює своє поточне представлення. Тобто при передачі повідомлень кожна вершина виконує зважене сумування своїх ознак та ознак сусідів. Такий підхід дає можливість кожній вершині оновити свої ознаки з урахуванням контексту, тобто локального фрагменту структури графа. У запропонованій моделі використано два шари. Перший шар аналізує зв’язки вершинам-сусідами. Другий шар аналізує зв’язки між трійками вершин. Наприклад, для системи електронної комерції перший шар може аналізувати зв'язок Ноутбук – Миша, а другий шар – зв'язок Ноутбук – Миша – Килимок для миші.

Механізм темпоральної уваги ATT, який враховує два типи темпоральних патернів: внутрішні та зовнішні. Вони відображають повтор певних змін у даних через проміжки часу. Внутрішні патерни враховують темпоральні залежності для однієї вершини графової нейронної мережі, наприклад, сезонні коливання продажів в системі електронної комерції. Зовнішні патерни враховують зміни у часі, пов'язані із декількома вузлами, наприклад, затримку продажів нових товарів у роздрібних торговців порівняно із оптовиками. Механізм уваги для кожного вузла графової мережі розраховує вагу уваги до вершин-сусідів та уваги поточної вершини до своєї історії, після чого комбінує отримані результати. Головна ідея механізму уваги полягає в тому, щоб врахувати з різною вагою релевантні для поточного прогнозування та пояснення темпоральні патерни.

Блок агрегації *Pool* інтегрує представлення всіх вершин графа у єдине векторне представлення всієї мережі. Для побудови такого представлення використовується середнє значення векторів для вузлів мережі. Можуть також бути використані максимальне, зважене значення, агрегація з увагою, коли ваги вершин визначаються у відповідності до релевантності цих вершин задачі, що вирішує мережа.

Блок підготовки пояснень *PExpl* перетворює внутрішнє векторне представлення мережі у числове представлення впливу вхідних факторів на прогноз. У випадку використання агрегованого представлення мережі розраховується вплив поточної комбінації значень вхідних даних на прогнозне значення мережі. Отримане значення відображає ступінь впливу. Наприклад, для системи електронної комерції може бути розраховано вплив на результат роботи системи комбінації поточних продажів товарів через різні канали в різних регіонах. Для подальшої деталізації блок підготовки пояснень використовує як вхідні дані внутрішнє представлення вершин мережі.

Блок прогнозування *Predictor* отримує векторне представлення графа мережі та

перетворює цей вектор у числове значення, що характеризує прогнозний показник роботи системи, яка моделюється графовою нейронною мережею. У даному шарі може бути використано лінійне перетворення, що дає можливість врахувати вагу кожного елементу вектору графу мережі. Наприклад, для торгівельної системи як прогноз виступає об'єм продажів для заданої номенклатури товарів, каналів та регіонів продажів.

Блок генерації пояснень $Expl$ отримує значення (ваги) уваги для конкретних типів вхідних даних, а також для темпоральних патернів, що відображають зміни цих даних. На основі цих даних розраховується вплив ключових даних на прогнозний результат. У процесі розрахунку враховуються темпоральні закономірності, тобто тренди впливу вхідних даних на результат у часі. Отримані залежності представляються у текстовій формі. Тобто даний блок враховує суттєві компоненти вектору внутрішнього представлення мережі і узгоджує результат із ментальною моделлю користувача.

Таким чином, модель графової нейронної мережі для обробки темпорально упорядкованих даних забезпечує формування пояснення e для поточного прогнозу y через композицію блоків $Expl, Predictor, PExpl, Pool, ATT, GCN, B, LSTM, Emb$. Ця модель має такий формальний вигляд:

$$e|y = Expl\Big(Predictor \wedge PExpl\Big(Pool\Big(ATT\Big(GCN\Big(B\Big(LSTM\Big(Emb\big(X \big)\big)\big)\big)\big)\big)\big)\Big). \quad (1)$$

Граф, що формується блоком B, містить вузли, які зберігають інформацію про об'єкти, описані вхідними даними. Дуги відображають вплив вузлів. Наприклад, для системи електронної комерції вузли можуть містити інформацію про товари, категорії товарів, канали продажів тощо. Дуги відображають зв'язок між продажами товарів, залежності продажів категорій товарів від каналу, регіону тощо.

5. Експериментальна перевірка графової нейронної мережі в задачі побудови пояснень щодо продажів в системі електронної комерції

Експериментальну перевірку виконано з використанням фрагменту логу системи електронної комерції. Вхідний лог містить елементи, що відображають час кожної транзакції, а також інформацію про товари, що були продані в рамках цієї транзакції. Зокрема, лог містить інформацію про тип продукту (ноутбук, смартфон, планшет тощо), категорію продукту (електроніка, аксесуари тощо), канали продажів (веб-сайт, мобільний застосунок, маркетплейс), регіони продажів продукції (Київ, Львів, Одеса тощо), кількість та вартість проданої продукції. Фрагмент вхідних даних представлено в табл. 2.

Таблиця 2

Фрагмент вхідних даних із обчисленням зміни продажів

Дата	Вартість продукції	Зміни продажів
2025-03-24	7115,98	
2025-03-25	11949,45	1,67
2025-03-26	13637,02	1,14
2025-03-27	6733,06	0,49
2025-03-28	9336,34	1,39
2025-03-29	8401,17	0,90
2025-03-30	52336,09	0,63

Сформовано граф, який структурує продажі за тиждень. Граф містить 14 вершин та 18 дуг. Вершини розподілені за категоріями: продукти (6), класи товарів (2), канали

продажів (3), регіони (3). Кожна дуга представляє зв'язок між продуктами, класами, каналами продажів та регіонами. Дуга має вагу, яка відображає силу кожного із зв'язків. Наприклад, сила зв'язку Ноутбук – Миша становить 0,83, оскільки це комплементарні товари. Сила зв'язку Ноутбук – Клавіатура становить 0,72 оскільки додаткову клавіатуру часто купують з ноутбуком. Сила зв'язку, що визначає належність до категорій, становить 1,0. Наприклад, всі електронні пристрої мають вагу 1,0 зв'язку з категорією «Електроніка».

Модель генерує пояснення через механізм уваги та аналіз внеску кожного вузла. Пояснення відображає вплив змін у вхідних даних на результуючі продажі. Наприклад, пояснення щодо зростання продажів від 25 березня має формальний вигляд:

$$\begin{aligned} &\textit{Категорія}(\textit{електроніка}(+0,43)) \wedge \textit{Канал}(\textit{мобільні_застосунки}(+0,54)) \Rightarrow \\ &\textit{Продажі}(+0,67) \big| \textit{Дата} = 25.03.2025. \end{aligned} \tag{2}$$

Тобто зростання продажів 25 березня на 67 % порівняно з 24 березня відбулося внаслідок збільшення активності в категорії Електроніка (+45 %) та зростання продажів через канал мобільних застосунків (+32 %).

Аналогічно, зниження продажів 26 березня обґрунтовується таким поясненням:

$$\begin{aligned} &\textit{Категорія}(\textit{аксесуари}(-0,59)) \wedge \textit{Канал}(\textit{веб}-\textit{сайт}(-0,42)) \Rightarrow \\ &\textit{Продажі}(-0,51) \big| \textit{Дата} = 27.03.2025. \end{aligned} \tag{3}$$

Тобто зниження продажів обумовлено спадом попиту на аксесуари та зниженням віддачі від веб-сайту.

6. Обговорення результатів

Розроблена графова нейронна мережа на основі вхідних даних із мітками часу будує графове представлення системи, яка описана вхідними даними. На основі отриманого графового представлення мережа формує прогнозний результат роботи системи, що моделюється, а також пояснення до даного прогнозу. Система відрізняється від існуючих моделей використанням блоків підготовки та формування пояснення, що дає можливість обґрунтувати отримане рішення на основі його залежності від значень вхідних даних.

Розроблена модель відрізняється від існуючих виділенням часових інтервалів, для яких на основі загальних правил будується граф мережі. Такий темпоральний підхід дає можливість отримувати відмінні патерни на різних інтервалах часу та порівнювати ці патерни. Крім того, розроблена мережа в рамках механізму уваги комбінує темпоральні патерни для однієї вершини графа та спільні патерни для декількох вершин.

Результати експериментальної перевірки показали, що отримані в графовій нейронній мережі пояснення враховують як зважені властивості вхідних об'єктів, так і зміну значень цих властивостей з часом.

Ключове обмеження даної моделі пов'язане із кількістю категорій (класів) вхідних даних, які відображаються як вершини графа. При збільшенні кількості класів даних ускладнюється механізм уваги, що приводить до підвищення обчислювальних витрат.

Подальші дослідження у даному напрямку орієнтовані на використання контекстної векторизації та формування пояснень зі змінною деталізацією на основі векторного представлення вершин графу та механізму уваги.

7. Висновки

Розроблено модель графової нейронної мережі для темпорально упорядкованих даних в задачі побудови пояснень в інтелектуальній системі. Модель містить блоки побудови векторних представлень вхідних даних, виявлення темпоральних патернів, формування графа мережі, прогнозування, підготовки та формування пояснень на основі агрегації

представлення мережі та використання механізму уваги. Модель відрізняється від моделей існуючих темпоральних графових нейронних мереж інтеграцією модулів підготовки та формування пояснення, а також врахуванням темпоральних патернів для однієї та декількох вершин графа мережі. Модель дає можливість сформувати пояснення на основі числової оцінки впливу на прогнозний результат ключових факторів, представлених упорядкованими у часі значеннями вхідних даних, з урахування змін цих даних з часом.

Експериментальна перевірка розробленої графової нейронної мережі в задачі побудови пояснень щодо продажів в системі електронної комерції показала можливості формування пояснення на основі визначення впливу змін у продажах окремих категорій товарів через окремі канали розповсюдження на загальний показник роботи системи.

Перелік посилань:

1. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144).
2. Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI). IEEE Access, 6, 52138-52160. https://doi.org/10.1109/ACCESS.2018.2870052.
3. Чалий, С. Ф., & Лещинська, І. О. (2024). Принципи побудови ментальних моделей рішення для зовнішнього користувача в задачі формування пояснень в інтелектуальній системі. АСУ і прилади автоматики, 181, 17-25.
4. Gunning, D., & Aha, D. W. (2019). DARPA's Explainable Artificial Intelligence (XAI) program: A retrospective. AI Magazine, 40(2), 44-58. https://doi.org/10.1609/aimag.v40i2.2850.
5. Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence, 267, 1-38.
6. Чалий, С., & Лещинська, І. (2023). Концептуальна ментальна модель пояснення в системі штучного інтелекту. Вісник Національного технічного університету «ХПІ». Серія: Системний аналіз, управління та інформаційні технології, 1 (9), 70–75. https://doi.org/10.20998/2079-0023.2023.01.11
7. Arrieta, A. B., Díez-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., ... & Herrera, F. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58, 82-115.
8. Чалий, С., Лещинський, В., & Лещинська, І. (2022). Каузальна модель процесу побудови пояснень в інформаційній системі. Системи управління, навігації та зв'язку, 3(69), 99-103. https://doi.org/10.26906/SUNZ.2022.3.099.
9. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Philip, S. Y. (2020). A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, 32(1), 4-24.
10. Zheng, C., Fan, X., Wang, C., & Qi, J. (2020). GMAN: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, 34 (01), 1234-1241.
11. Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., ... & Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261.
12. Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks? arXiv preprint arXiv:1810.00826.
13. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., ... & Li, H. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 21(9), 3848-3858.
14. Jin, G., Liang, Y., Fang, Y., Huang, J., Zhang, J., & Zheng, Y. (2023). Spatio-temporal graph neural networks for predictive learning in urban computing: A survey. IEEE Transactions on Knowledge and Data Engineering, 35(8), 8075-8095.
15. Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2017). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926.
16. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.
17. Чала, О. В. (2021). Моделі, методи та інформаційні технології автоматизованого управління темпоральними базами знань для підтримки управлінських рішень [Докторська дисертація, Харківський національний університет радіоелектроніки].
18. Чала, О. В. (2018). Принципи автоматизованої побудови та використання темпоральної бази знань для підтримки прийняття рішень з управління підприємством. Системи управління, навігації та зв'язку, 6, 122-128.

https://doi.org/10.26906/SUNZ.2018.6.122

19. Chalyi, S., & Leshchynskyi, V. (2022). Temporal-oriented model of causal relationship for constructing explanations for decision-making process. Advanced Information Systems, 6(3), 60–65. https://doi.org/10.20998/2522-9052.2022.3.09

20. Чалий С. Ф., Лещинський В. О. (2024). Темпорально-каузальні методи побудови пояснень в системах штучного інтелекту. АСУ та прилади автоматики, 181, 91-99. https://doi.org/10.30837/0135-1710.2024.181.091.

21. Чалий С. Ф., Лещинський В. О., Лещинська І. О. (2021). Контрфактуальна темпоральна модель причинно-наслідкових зв'язків для побудови пояснень в інтелектуальних системах. Вісник Національного технічного університету «ХПІ». Сер. : Системний аналіз, управління та інформаційні технології: зб. наук. пр. Харків : НТУ «ХПІ», 2 (6), 41-46. https://doi.org/10.20998/2079-0023.2021.02.07.

22. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM Computing Surveys, 51(5), 1-42.

Надійшла до редколегії 25.05.2025 р.

Чалий Сергій Федорович, доктор технічних наук, професор, професор кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: serhii.chalyi@nure.ua; ORCID: 0000-0002-9982-9091

Кравченко Ростислав Вікторович, аспірант кафедри ІУС ХНУРЕ, м. Харків, Україна, e-mail: rostyslav.kravchenko1@nure.ua; ORCID: 0009-0009-0324-3597

УДК 004.4`6

Модифікація моделі одномісного коваріантного функтора для процесу кросплатформної міграції інформаційної системи / А.С. Круглик, В.М. Левикін, М.В. Євланов, Б.І. Мороз, Д.М. Мороз. АСУ та прилади автоматики. 2025. № 185. С. 5-23.

Об'єктом дослідження є процес кросплатформної міграції програмного забезпечення (ПЗ) інформаційної системи (ІС). Визначено, що існуючі методи та засоби штучного інтелекту не дозволяють повністю вирішити проблему створення та використання формального механізму підтримки цілісності структурних та поведінкових особливостей моделей ІС та її ПЗ під час їх взаємної трансформації. Сучасні дослідження в цьому напрямі спрямовані, зокрема, на визначення та реалізацію окремих функторів та їх моделей для конкретних ІТ-продуктів різного призначення. Тому проведення досліджень із застосування категорно-функторного апарату для формального опису процесу кросплатформної міграції ПЗ ІС є актуальним з теоретичної і прикладної точок зору.

Як базову модель було обрано запропоновану авторами дослідження загальну модель одномісного коваріантного функтора. Визначено особливості використання цієї моделі для формального опису процесу кросплатформної міграції ПЗ ІС. Розроблено два варіанти модифікованої моделі одномісного коваріантного функтора, які дозволяють описати процес кросплафтормної міграції ПЗ ІС під час експлуатації та під час валідації цієї ІС. Запропоновано загальні методи використання модифікованих моделей у роботах та діяльностях процесу кросплафтормної міграції ПЗ ІС.

Проведено експериментальну перевірку отриманих наукових результатів. Для такої перевірки було обрано ПЗ інформаційно-аналітичної системи управління медичним закладом. Застосування модифікованої моделі одномісного коваріантного функтора дозволило представити роботу з кросплатформної міграції рефакторингу похідного коду ПЗ як функтор, реалізований у вигляді функції мовою C/C++. Отримані результати дозволяють визнати перспективною можливість застосування модифікованих моделей одномісного коваріантного функтора для формального опису інформаційної технології автоматизованого управління процесом кросплатформної міграції ПЗ ІС.

Ключові слова: кросплатформна міграція, програмне забезпечення, інформаційна система, одномісний коваріантний функтор, рефакторинг програмного коду.

Табл. 1. Іл. 3. Бібліогр.: 22 назв.

UDC 004.4`6

Modification of the Single-place Covariant Functor Model for the Process of Information System Cross-platform Migration / A.S. Kruglik, V.M. Levykin, M.V. Yevlanov, B.I. Moroz, D.M. Moroz. Management Information System and Devices. 2025. № 185. P. 5-23.

The object of the study is the process of cross-platform migration of information system (IS) software. It was determined that existing methods and tools of artificial intelligence do not allow to fully solve the problem of creating and using a formal mechanism for maintaining the integrity of the structural and behavioral features of IS models and its software during their mutual transformation. Modern research in this area is aimed, in particular, at defining and implementing individual functors and their models for specific IT products for various purposes. Therefore, conducting research on the use of a categorical-functor apparatus for the formal description of the process of cross-platform migration of IS software is relevant from a theoretical and applied point of view.

The general model of a single-place covariant functor proposed by the authors of the study was chosen as the basic model. The features of using this model for the formal description of the process of cross-platform migration of IS software are determined. Two variants of the modified model of a single covariant functor have been developed, which allow describing the process of cross-platform migration of IS software during operation and validation of this IS. General methods of using modified models in the works and activities of the process of cross-platform migration of IS software have been proposed.

An experimental verification of the obtained scientific results has been carried out. For such verification, the software of the information and analytical system of the management of a medical institution was chosen. The use of the modified model of a single covariant functor allowed us to present the work on cross-platform

migration of the refactoring of the software source code as a functor implemented as a function in the C/C++ language. The results obtained allow us to recognize as promising the possibility of using modified models of a single-place covariant functor for the formal description of the information technology of automated management of the process of cross-platform migration of IS software.

Keywords: cross-platform migration, software, information system, single covariant functor, refactoring of program code.

Tab. 1. Fig. 3. Ref.: 22 items.

УДК 004.932.2

Моделювання та дослідження методів детекції погляду користувача у системах людино-комп'ютерної взаємодії / П.Е. Ситнікова, М.О. Грицай. АСУ та прилади автоматики. 2025. № 185. С. 23-33.

Розглянуто підходи до організації безконтактної взаємодії людини з комп'ютером на основі напрямку погляду. Наведено структуру системи, що складається з модулів виявлення обличчя, визначення орієнтації голови, аналізу положення очей та прогнозування напрямку погляду за допомогою нейромережевої моделі.

Підкреслено важливість етапу попередньої обробки та нормалізації зображень для забезпечення стабільної роботи системи в умовах змін зовнішнього освітлення та положення голови користувача. Висвітлено використання нейронної мережі, що отримує на вхід нормалізовані зображення очей та обличчя, та генерує прогноз напрямку погляду у вигляді кутів нахилу. Зазначено, що ці дані конвертуються у тривимірний вектор, який перетинається з площиною екрана для визначення координат точки, на яку спрямовано погляд.

Описано розроблений механізм підтвердження вибору за допомогою подвійного кліпання. Це рішення дозволяє виконувати команди без необхідності використання клавіатури чи миші, що є особливо актуальним для користувачів з обмеженими фізичними можливостями.

У результаті дослідження зроблено висновок, що розроблена система демонструє високу точність та стабільність при роботі в реальному часі. Запропоновані рішення відкривають перспективи для подальшого розвитку систем природної взаємодії людини з комп'ютером, зокрема у сфері асистивних технологій.

Ключові слова: комп'ютерний зір, коригування руху очей, калібрування, фільтр Калмана, експоненційне згладжування.

Табл. 1. Іл. 4. Бібліогр.: 12 назв.

UDK 004.932.2

Modeling and Research of User Gaze Detection Methods in Human-computer Interaction Systems / P.E. Sytnikova, M.O. Hrytsai. Management Information System and Devices. 2025. № 185. P. 23-33.

The article considers approaches to organizing contactless human-computer interaction based on gaze direction. The architecture of the system is presented, consisting of modules for face detection, head orientation determination, eye position analysis, and gaze direction prediction using a neural network model.

The importance of the stage of pre-processing and normalization of images is emphasized to ensure stable operation of the system under changing ambient lighting and user head positions. The use of a neural network that receives normalized eye and face images as input and generates a prediction of gaze direction in the form of tilt angles is highlighted. It is noted that this data is converted into a three-dimensional vector that intersects the screen plane to determine the coordinates of the point at which the gaze is directed.

The developed mechanism for confirming a choice using a double blink is also described. This solution allows you to execute commands without the need to use a keyboard or mouse, which is especially relevant for users with physical disabilities.

As a result of the study, it was concluded that the developed system demonstrates high accuracy and stability when working in real time. The proposed solutions open up prospects for the further development of systems for natural human-computer interaction, in particular in the field of assistive technologies.

Keywords: computer vision; eye movement correction; calibration; Kalman filter; exponential smoothing.

Tab. 1. Fig. 4. Ref.: 12 items.

УДК 004.8:004.9

Побудова функціонально-темпорального представлення ментальної моделі рішення в задачі формування пояснень в інтелектуальних системах / С.Ф. Чалий, І.О. Лещинська. АСУ та прилади автоматики. 2025. № 185. С. 34-42.

Об'єктом дослідження є процес побудови ментальної моделі рішення для користувача інтелектуальної інформаційної системи. Метою є розробка підходу до побудови функціонально-темпорального представлення ментальної моделі рішення інтелектуальної інформаційної системи. Задачі: розробка функціонально-темпорального представлення ментальної моделі рішення інтелектуальної інформаційної системи; розробка методу побудови ментальної моделі рішення інтелектуальної інформаційної системи. Наукова новизна отриманих результатів полягає в такому. Розроблено функціонально-темпоральне представлення ментальної моделі, яке містить структуровану множину властивостей рішення у функціональному аспекті, а також сценарії використання рішення у темпоральному аспекті. Запропоновано метод побудови функціонально-темпорального представлення ментальної моделі, який містить етапи формування структурованого набору вхідних даних, формування переліку потенційних ментальних моделей за процесом використання, відбору релевантних властивостей рішення, побудови ментальних моделей на основі інтеграції функціональних та темпоральних характеристик, агрегації властивостей за сценаріями використання та формування доповненого представлення ментальних моделей. Метод дає можливість обґрунтовано відібрати для користувача релевантне пояснення щодо отриманого рішення. Експериментальна перевірка розробленого методу показала можливість формування ментальних моделей, які відображають як функціональні властивості рішення, так і сценарії їх використання.

Ключові слова: пояснення, інтелектуальна система, самопояснювальний штучний інтелект, система штучного інтелекту, ментальна модель, користувачі інтелектуальної системи, функціонально-темпоральне представлення, сценарії використання

Табл. 1. Іл. 3. Бібліогр.: 20 назв.

UDC 004.8:004.9

Development of Functional-Temporal Representation of Mental Model Solutions in the Task of Explanation Generation in Intelligent Systems / S.F. Chaly, I.O. Leshchynska. Management Information System and Devices. 2025. № 185. P. 34-42.

The subject of the research is the process of constructing a mental model of a solution for a user of an intelligent information system. The aim is to develop an approach to building a functional-temporal representation of the mental model of a solution for an intelligent information system. Objectives: development of functional-temporal representation of the mental model of a solution for an intelligent information system; development of a method for constructing a mental model of a solution for an intelligent information system. Scientific novelty of the obtained results lies in the following. A functional-temporal representation of the mental model has been developed, which contains a structured set of solution properties in the functional aspect, as well as scenarios of solution use in the temporal aspect. A method for constructing functional-temporal representation of the mental model has been proposed, which includes stages of forming a structured set of input data, forming a list of potential mental models according to the usage process, selecting relevant solution properties, building mental models based on integration of functional and temporal characteristics, aggregating properties according to usage scenarios, and forming a supplemented representation of mental models. The method makes it possible to reasonably select relevant explanations about the obtained solution for the user. Experimental verification of the developed method demonstrated the possibility of forming mental models that reflect both functional properties of solutions and scenarios of their use based on analysis of user feedback from the intelligent system.

Keywords: explanation, intelligent system, self-explanatory artificial intelligence, artificial intelligence system, mental model, intelligent system users, functional-temporal representation, usage scenarios

Tab. 1. Fig. 3. Ref.: 20 items.

УДК 004.83

Огляд сучасних нейромережевих архітектур для сегментації зображень / А.Р. Ковтуненко, С.В. Машталір. АСУ та прилади автоматики. 2025. № 185. С. 43-52.

Об'єктом дослідження є методи сегментації зображень та їх еволюція. Аналіз статті показав, що сегментація зображень за останнє десятиліття зазнала значного розвитку методів: від традиційних згорткових нейронних мереж до трансформерних моделей, здатних обробляти мультимодальні дані. Актуальність статті обумовлена швидким збільшенням кількості методів та підходів до вирішення різних задач сегментації, а також необхідністю розуміння сильних та слабких сторін цих методів.

Представлено хронологічний огляд моделей сегментації, починаючи з повнозв'язних згорткових нейронних мереж (FCN), продовжуючи аналізом U-Net, SegNet, сімейства DeepLab, Mask R-CNN, Gated-SCNN, FastFCN, MaskFormer, SegFormer, Mask2Former, OneFormer і завершуючи моделями нового покоління – Segment Anything Model (SAM) та її удосконаленою версією HQ-SAM. Для кожної архітектури детально розглянуто технічні особливості, інновації, переваги та обмеження.

Проведений аналіз дав змогу виявити типові слабкі місця моделей – точне виявлення границь об'єктів, сегментація малих об'єктів, доменна адаптація та використання апріорних знань, а також обчислювальні обмеження. Показано, що розвиток моделей не був лінійним, і ранні методи не були повністю замінені. Багато сучасних підходів успішно поєднують класичні методи з нейромережевими, створюючи гібридні рішення, що перевершують попередні.

Систематизація наявних підходів у хронологічному порядку дозволяє не тільки простежити еволюцію архітектурних рішень, але й зрозуміти основні ідеї та напрямки досліджень, а також невирішені проблеми, що залишаються актуальними. Це дає змогу уникати повторення вже відомих обмежень та помилок минулих підходів при розробці нових моделей і обґрунтовано обирати архітектури для вирішення прикладних задач.

Ключові слова: комп'ютерний зір, сегментація зображень, нейронні мережі, архітектури-трансформери, сегментація без попереднього навчання, аналіз методів сегментації.

Табл. 0. Іл. 13. Бібліогр.: 19 назв.

UDC 004.83

A Review of Modern Neural Network Architectures for Image Segmentation / A.R. Kovtunenko, S.V. Mashtalir. Management Information System and Devices. 2025. № 185. P. 43-52.

The object of research is image segmentation methods and their evolution. The analysis shows that image segmentation over the past decade has undergone significant methodological development: from traditional convolutional neural networks to transformer models capable of processing multimodal data. The relevance of the work is due to the rapid increase in the number of methods and approaches to solving various segmentation tasks, as well as the need to understand the strengths and weaknesses of these methods.

The article presents a chronological review of segmentation models, starting with fully convolutional neural networks (FCN) and continuing with the analysis of U-Net, SegNet, the DeepLab family, Mask R-CNN, Gated-SCNN, FastFCN, MaskFormer, SegFormer, Mask2Former, OneFormer, and concluding with new generation models – Segment Anything Model (SAM) and its improved version, HQ-SAM. For each architecture, technical features, innovations, advantages, and limitations are examined in detail.

The analysis enabled the identification of typical weaknesses of the models - accurate detection of object boundaries, segmentation of small objects, domain adaptation, and use of prior knowledge, as well as computational limitations. It is shown that the development of models was not linear, and early methods were not completely replaced. Many modern approaches successfully combine classical methods with neural network ones, creating hybrid solutions that outperform previous ones.

The systematization of existing approaches in chronological order allows not only to trace the evolution of architectural solutions but also to understand the main ideas and research directions, as well as unsolved problems that remain relevant. This makes it possible to avoid repeating already known limitations and errors of past approaches when developing new models and to make informed choices of architectures for solving applied tasks.

Keywords: computer vision, image segmentation, neural networks, transformer models, zero-shot segmentation, analysis of segmentation methods.

Tab. 0. Fig. 13. Ref.: 19 items.

УДК 004.8:004.9

Причинно-наслідкове моделювання пояснень в інтелектуальних системах / С.Ф. Чалий, В.О. Лещинський/ АСУ та прилади автоматики. 2025. № 185. С. 53-62.

Об'єктом дослідження є процес побудови пояснень в інтелектуальних інформаційних системах. Метою дослідження є розробка причинно-наслідкового підходу до побудови пояснень в інтелектуальних системах з тим, щоб відобразити у формі пояснення причини отриманого рішення, а також причини дій процесу формування рішення й обмеження на ці дії, що дає можливість представити роботу інтелектуальної системи у зрозумілій для користувачів формі. Для досягнення мети роботи вирішуються такі завдання: розробка причинно-наслідкової моделі пояснення; розробка методу побудови пояснень на основі причинно-наслідкової моделі. Висновки. Запропоновано причинно-наслідкову модель пояснення з ієрархічною трирівневою структурою, яка структурує представлення пояснень на глобальному, процесному та локальному рівнях у темпоральному та каузальному аспектах. Модель на глобальному рівні містить можливісні причинно-наслідкові залежності між вхідними даними та рішенням системи з урахуванням прихованих факторів, на процесному рівні відображає каузальні залежності між діями процесу формування рішення на основі темпоральних правил типів «Next» та «Future», на локальному рівні задає детерміновані обмеження на послідовність дій процесу. Запропоновано темпорально-каузальний метод побудови пояснень, що містить п'ять етапів: визначення внутрішніх та зовнішніх користувачів, побудову пояснень на глобальному рівні з розрахунком можливості та необхідності, формування пояснень на процесному рівні через темпоральні правила та каузальні залежності, побудову локальних обмежень та їх перевірку, інтеграцію трирівневого пояснення. Експериментальна перевірка методу проведена на реальних даних процесу управління інцидентами.

Ключові слова: пояснення, інтелектуальна система, самопояснювальний штучний інтелект, темпоральна залежність, каузальна залежність, можливість, система штучного інтелекту.

Табл. 3. Іл. 0. Бібліогр.: 19 назв.

UDC 004.8:004.9

Causal Modeling of Explanations in Intelligent Systems / S.F. Chaly, V.O. Leshchynskyi / Management Information System and Devices. 2025. № 185. P. 53-62.

The subject of the research is the process of building explanations in intelligent information systems. The purpose of the research is to develop a causal approach to building explanations in intelligent systems in order to represent in the form of explanations the causes of the obtained decision, as well as the causes of actions in the decision-making process and constraints on these actions, which makes it possible to present the operation of the intelligent system in a form understandable to users. To achieve the goal of the work, the following tasks are solved: development of a causal explanation model; development of a method for building explanations based on the causal model. Conclusions. A causal explanation model with a hierarchical three-level structure is proposed, which structures the representation of explanations at global, process, and local levels in temporal and causal aspects. The developed model at the global level contains possibilistic causal dependencies between input data and the system's decision considering hidden factors, at the process level reflects causal dependencies between actions of the decision-making process based on temporal rules of «Next» and «Future» types, and at the local level sets deterministic constraints on the sequence of process actions. A temporal-causal method for building explanations is proposed, which includes five stages: identification of internal and external users, building explanations at the global level with calculation of possibility and necessity, formation of explanations at the process level through temporal rules and causal dependencies, construction of local constraints and their verification, integration of three-level explanation. Experimental verification of the method was conducted on real data of the incident management process.

Keywords: explanation, intelligent system, self-explanatory artificial intelligence, temporal dependency, causal dependency, possibility, artificial intelligence system.

Tab. 3. Fig. 0. Ref.: 19 items.

УДК 004.3, 004.421.5, 004.056.5

Аналіз джерел ентропії для генерації випадкових чисел на базі мобільних пристроїв / Д.О. Остапець, А.О. Опрятний. АСУ та прилади автоматики. 2025. С. 63-70.

У сучасному світі інформаційних технологій генерація випадкових чисел відіграє ключову роль у багатьох сферах. У статті проведено огляд і аналіз можливостей використання вбудованих сенсорів мобільних пристроїв як джерел ентропії для апаратних генераторів випадкових чисел (ГВЧ). Такий підхід дозволяє знизити вартість створення генераторів, зберігаючи при цьому достатній рівень випадковості.

Об'єктом дослідження є процес генерування випадкових чисел з використанням датчиків мобільних пристроїв як джерела ентропії. Показано два основні типи генераторів: програмні генератори псевдовипадкових чисел (ГПВЧ), які не завжди відповідають критеріям непередбачуваності, та апаратні генератори випадкових чисел (ГВЧ), які базуються на фізичних джерелах шуму. Для забезпечення високої якості генерації апаратні ГВЧ потребують ефективного джерела ентропії. Запропоновано використовувати сенсори мобільних пристроїв – акселерометри, гіроскопи, магнітометри, барометри, датчики освітлення тощо. Вони здатні реєструвати зміни в зовнішньому середовищі або положенні пристрою й генерувати великі обсяги даних, що можуть використовуватися як джерело ентропії.

Проведено огляд попередніх досліджень з використання датчиків мобільних пристроїв та пристроїв Інтернету речей. Сформульовано такі вимоги до датчиків: чутливість, наявність датчика у більшості мобільних пристроїв, швидкість оцифровування даних, кількість отриманих бітів за одне вимірювання. Проведено порівняльний аналіз датчиків за вказаними характеристиками. За результатами порівняння, кращі показники мають датчики акселерометра, гіроскопа і магнітометра, що і зумовило їх вибір для подальшого використання як джерел ентропії в апаратних генераторах випадкових чисел.

Ключові слова: інформаційні технології; випадкові числа; джерело ентропії; датчик; мобільний пристрій; акселерометр; гіроскоп; магнітометр.

Табл. 2. Іл. 1. Бібліогр.: 12 назв.

UDC 004.3, 004.421.5, 004.056.5

Analysis of Entropy Sources for Random Number Generation Based on Mobile Devices / D.O. Ostapets, A.O. Opriatnyi. Management Information System and Devices. 2025. P. 63-70.

In the modern world of information technology, random number generation plays a key role in many areas. The article provides a review and analysis of the possibilities of using built-in sensors of mobile devices as sources of entropy for hardware true random number generators (TRNGs). This approach allows you to reduce the cost of creating generators while maintaining a sufficient level of randomness.

The object of research is the process of generating random numbers using mobile device sensors as a source of entropy. Two main types of generators are shown: software pseudo-random number generators (PRNGs), which do not always meet the unpredictability criteria, and hardware true random number generators (TRNGs), which are based on physical noise sources. To ensure high generation quality, hardware TRNGs require an effective source of entropy. It is proposed to use sensors of mobile devices - accelerometers, gyroscopes, magnetometers, barometers, light sensors, etc. They are able to register changes in the external environment or position of the device and generate large amounts of data that can be used as an entropy source.

A review of previous research on the use of mobile device sensors and IoT devices is carried out. The following requirements for sensors were formulated: sensitivity, the presence of a sensor in most mobile devices, the speed of data digitization, the number of bits received per measurement. A comparative analysis of the sensors according to the specified characteristics is carried out. According to the results of the comparison, the best indicators are provided by accelerometer, gyroscope and magnetometer sensors, which led to their selection for further use as sources of entropy in hardware random number generators.

Keywords: information technology; random numbers; source of entropy; sensor; mobile device; accelerometer; gyroscope; magnetometer.

Tab. 2. Fig. 1. Ref.: 12 items.

УДК 519.21:004.021

Алгоритм завоювання для стохастичного заповнення двовимірних дискретних решіток зв'язаними областями / В. В. Красніков, П. Е. Ситнікова. АСУ та прилади автоматики. 2025. № 185. С. 70-77.

Об'єктом дослідження є процес процедурного розфарбовування двовимірного дискретного простору з топологічними обмеженнями.

Запропоновано новий алгоритм, що дозволяє розв'язати розповсюджену проблему генерації змістовних стохастичних зв'язаних областей для двовимірних дискретних полів. Проведено дослідження, в ході якого сформульовано відповідну чітку задачу, ілюстровану прикладом у вигляді створення ігрового поля для модифікованої версії класичної задачі N-Queens. Задача полягає у реалізації алгоритму, який дозволив би створювати будь яке поле площею N x N, де N > 0, задається вручну. Поле повинно процедурно заповнюватись зв'язними областями.

Проведено аналіз існуючих алгоритмів та підходів, які зазвичай використовуються у процедурній генерації у сумісних індустріях. Серед таких розглянуто: градієнтний шум Перліна; клітинні автомати; хвильова заливка і випадкове блукання.

Зроблено висновки, що ці алгоритми у їхній класичній репрезентації не підходять для процедурного створення ігрового поля для обраної задачі, тому у дослідженні запропоновано новий алгоритм, який є похідним від алгоритму випадкового блукання.

Цей алгоритм полягає у створенні агентів, які в даному випадку є ферзі на шахівниці. Кожен агент рухається по черзі, але можливість ходу і вибір ортогонального напряму задається за допомогою додаткових критеріїв, таких як: формула пропуску ходу, пріоритет нічийних клітин над власними, пріоритет клітин ближчих до краю поля і неможливість заходити на чужі поля.

Алгоритм реалізовано з використанням мови програмування C#. В роботу програми було також додано можливість зробити декілька генерацій поспіль і порахувати метрики для отриманих фігур на матрицях. Для дослідження успіху запропонованого алгоритму було використано такі метрики: середня витягнутість, середня прямокутність, середня зубчастість контуру, максимальна витягнутість, максимальна прямокутність і мінімальна зубчастість контуру. Було також пораховано метрику ефективності алгоритму щодо його складності тобто (кількість кроків агентів поділена на площу решітки).

Для розрахунку ефективності та збору метрик було проведено по тисячі експериментів зі значеннями розміру поля N x N, де N = 8, 12, 20. Результати роботи довели, що алгоритм працює, виконує поставлену задачу і є ефективним.

Ключові слова: алгоритм завоювання, стохастичне заповнення, двовимірна дискретна решітка, зв'язні області.

Табл. 1. Іл. 4. Бібліогр.: 7 назв.

UDK 519.21:004.021

Conquest Algorithm for Stochastic Filling of Two-dimentional Discrete Grids with Connected regions / V.V. Krasnikov, P.E. Sytnikova. Management Information System and Devices. 2025. № 185. P. 70-77.

The object of research is the process of procedural coloring of a two-dimensional discrete space with topological constraints.

A new algorithm is proposed that solves the widespread problem of generating meaningful stochastic connected regions on two-dimensional discrete fields. A study was conducted in which a clear formulation of the task was given and illustrated by constructing a game board for a modified version of the classical N-Queens problem. The task is to implement an algorithm able to create an arbitrary N x N board (for any manually specified N > 0). The board must then be filled procedurally with connected regions.

An analysis of existing algorithms commonly used in procedural generation for related industries was carried out. The following methods were examined: Gradient Perlin Noise, Cellular Automata, Wave-Based Flood Fill and Random Walk.

It was concluded that, in their classical form, these algorithms are not suitable for procedural

construction of a game board for the chosen task, so a new method derived from the random-walk approach was proposed.

The proposed algorithm introduces a set of agents, which – in this context – are the queens on a chessboard. Each agent moves in turn, but both the possibility of making a move and the choice of orthogonal direction is driven by additional criteria, as such: turn skip formula, empty cells bias against controlled ones, borderline cells bias against inner and prohibition of stepping onto the cells owned by other agents.

The algorithm was implemented in the C# programming language. The program includes the ability to run several consecutive generations and to compute metrics for the shapes obtained on the grids. To evaluate the success of the proposed algorithm the following metrics were used: average elongation, average rectangularity, average roughness, maximum elongation, maximum rectangularity and minimum roughness. In addition, an efficiency metric was calculated from the complexity standpoint – namely, the total number of agents steps divided by the grid area.

To measure efficiency and collect the metrics, one thousand experiments were carried out for the board sizes of N x N, where N = 8, 12 and 20. The results show that the algorithm operates correctly, fulfils the stated task, and is efficient.

Keywords: conquest algorithm, stochastic filling, two-dimensional discrete grid, connected regions.

Tab. 1. Fig. 4. Ref.: 7 items.

УДК 004.8:004.9

Графова нейронна мережа для темпорально упорядкованих даних в задачі побудови пояснень в інтелектуальній системі / С.Ф. Чалий, Р. В. Кравченко. АСУ та прилади автоматики. 2025. № 185. С. 77-85.

Об'єктом дослідження є процес побудови пояснень в інтелектуальних інформаційних системах. Предметом дослідження є моделі та методи формування пояснень в інтелектуальних інформаційних системах. Метою роботи є розробка моделі графової нейронної мережі, яка використовує темпоральний порядок у вхідних даних для побудови пояснень щодо процесу формування рішення в інтелектуальній системі. Завданнями дослідження є розробка моделі графової нейронної мережі для темпорально упорядкованих даних; експериментальна перевірка графової мережі в задачі побудови пояснень для системи електронної комерції. Запропоновано модель графової нейронної мережі, яка включає функціональні блоки побудови векторних представлень, виявлення темпоральних патернів з використанням мережі LSTM, формування графа мережі на заданому інтервалі часу, підготовки пояснень, прогнозування, генерації пояснень щодо прогнозних рішень з використанням механізму уваги. Механізм уваги комбінує темпоральні патерни для окремих вершин графової мережі та для взаємопов'язаних вершин графа. Розроблена модель забезпечує циклічне формування графової мережі для визначених часових інтервалів з використанням загальних правил навчання, що дозволяє виявляти темпоральні патерни та порівнювати ці патерни на різних інтервалах часу. Експериментальна перевірка підтвердила здатність мережі враховувати як статичні властивості вхідних об'єктів, так і динамічні зміни цих властивостей у часі. Додаткова перевага розробленої мережі полягає у можливості послідовної деталізації факторів, що пояснюють вплив вхідних даних на прогнозні результати роботи моделюючої системи.

Ключові слова: інтелектуальна система, система штучного інтелекту, пояснення, самопояснювальний штучний інтелект, графова нейронна мережа, темпоральні дані, механізм уваги, векторні представлення, LSTM, темпоральні патерни.

Табл. 2. Іл. 1. Бібліогр.: 22 назв.

UDC 004.8:004.9

Graph Neural Network for Temporally Ordered Data in the Task of Explanation Generation in an Intelligent System / S.F. Chaly, R.V. Kravchenko. Management Information System and Devices. 2025. № 185. P. 77-85.

The object of the research is the process of building explanations in intelligent information systems. The subject of the research is models and methods of explanation formation in intelligent information systems. The aim of the work is to develop a graph neural network model that uses temporal order in input

data to build explanations regarding the decision-making process in an intelligent system. The research tasks are to develop a graph neural network model for temporally ordered data, experimental verification of the graph network in the task of building explanations for an e-commerce system. A graph neural network model is proposed that includes functional blocks for building vector representations, identifying temporal patterns using LSTM networks, forming a network graph at a given time interval, preparing explanations, forecasting, and generating explanations regarding predictive decisions using an attention mechanism. The attention mechanism combines temporal patterns for individual vertices of the graph network and for interconnected graph vertices. The developed model ensures cyclical formation of the graph network for defined time intervals using common learning rules, which allows identifying temporal patterns and comparing these patterns across different time intervals. Experimental verification confirmed the network's ability to consider both static properties of input objects and dynamic changes of these properties over time. An additional advantage of the developed network lies in the possibility of sequential detailing of factors that explain the influence of input data on the predictive results of the modeling system.

Keywords: intelligent system, artificial intelligence system, explanation, self-explainable artificial intelligence, graph neural network, temporal data, attention mechanism, vector representations, LSTM, temporal patterns.

Tab. 2. Fig. 1. Ref.: 22 items.

ПРАВИЛА ОФОРМЛЕННЯ СТАТЕЙ
У ВСЕУКРАЇНСЬКОМУ МІЖВІДОМЧОМУ НАУКОВО-ТЕХНІЧНОМУ ЗБІРНИКУ
«АВТОМАТИЗОВАНІ СИСТЕМИ УПРАВЛІННЯ ТА ПРИЛАДИ АВТОМАТИКИ»

1. Загальні вимоги

До розгляду приймаються раніше не опубліковані статті українською та англійською мовами. Статті англійською мовою подаються разом з українськомовним варіантом. Статті, перекладені англійською за допомогою комп'ютерного перекладача та не відредаговані належним чином, не розглядаються.

Наукова стаття, яка подається до розгляду, має бути структурована та містити всі основні частини, характерні для наукової статті:

– постановка проблеми у загальному вигляді та її зв'язок з важливими науковими та практичними задачами;

– аналіз останніх досліджень та публікацій, у яких розпочато вирішення даної проблеми та на які спирається автор, виділення невирішених раніше частин загальної проблеми;

– формулювання цілей статті (постановка задачі);

– подання основного матеріалу досліджень з повним обґрунтуванням отриманих результатів;

– висновки даного дослідження та перспективи подальших досліджень у даному напрямку;

– перелік посилань (References).

2. Вимоги до структури рукопису

Структурно матеріали статті поділяються на такі елементи:

– УДК;

– прізвища та ініціали авторів статті;

– заголовок статті;

– анотація до статті;

– основний текст статті;

– перелік посилань;

– дата надходження статті до редколегії збірника;

– відомості про авторів статті;

– реферати українською та англійською мовами.

Бажаний порядок та зміст розділів основного тексту статті:

а) розділ 1 «Вступ», в якому визначається проблема у загальному вигляді та її зв'язок з важливими науковими та практичними задачами;

б) розділ 2 «Аналіз літературних джерел та визначення проблеми дослідження», в якому наводяться результати аналізу останніх досліджень та публікацій, де розпочато вирішення даної проблеми та на які спирається автор, виділяються невирішені раніше частини загальної проблеми дослідження та конкретизується головна проблема дослідження у даній статті;

в) розділ 3 «Мета і задачі дослідження», в якому наводяться описи мети дослідження та задач дослідження, вирішення яких дозволяє досягти визначеної раніше мети дослідження;

г) розділ 4 «Матеріали і методи дослідження», в якому наводяться описи формального апарату та раніше проведених експериментальних досліджень, які будуть використані у

подальшому тексті статті;

д) розділ 5 «Результаті дослідження», в якому структуровано наводяться результати вирішення сформульованих у розділі 3 окремих задач дослідження (теоретичних та експериментальних);

е) розділ 6 «Обговорення результатів дослідження», в якому наводяться: опис особливостей отриманих результатів дослідження та їхньої відмінності від результатів попередніх досліджень у відповідній галузі; опис переваг отриманих результатів перед існуючими; опис недоліків і обмежень, які утруднюють використання отриманих результатів дослідження; опис подальших перспектив проведення досліджень за цим напрямом;

ж) розділ 7 «Висновки», в якому наводяться стислі описи отриманих результатів вирішення окремих задач дослідження та загальний висновок про досягнення поставленої у розділі 3 мети дослідження.

Заголовки окремих розділів основного тексту статті можуть змінюватися відповідно до змісту конкретної статті.

Розділи основного тексту статті, перелік посилань, дата надходження статті до редколегії збірника та відомості про авторів статті відокремлюються один від одного одним порожнім рядком.

3. Вимоги до оформлення рукопису

До розгляду приймаються матеріали статей обсягом не менше 5 повних сторінок (з урахуванням рисунків і таблиць).

Матеріали статті повинні бути набраними у редакторі MS Word. Припустимі формати файлу з матеріалами статті – .doc або .docx.

Формат сторінки – А4 (210х297 мм). Поля знизу, зверху, справа, зліва – 3 см.

Основний текст статті набирається шрифтом Times New Roman, кегль 11, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 0 мм, вирівнювання по ширині.

Для УДК – шрифт Times New Roman, кегль 11, інтервал – 1,1, абзацний відступ – 8 мм, інтервал перед – 0 мм, інтервал після – 6 мм, вирівнювання по ширині.

Для прізвищ та ініціалів авторів статті – шрифт Times New Roman, кегль 11, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 6 мм, вирівнювання по ширині.

Для заголовка статті – шрифт Times New Roman, кегль 11, напівжирний, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 6 мм, вирівнювання по ширині.

Для анотації – шрифт Times New Roman, кегль 10, інтервал – 1,1, відступ зліва – 0,8 см, абзацний відступ – 8 мм, інтервал перед – 6 мм, інтервал після – 0 мм, вирівнювання по ширині.

Для заголовків таблиць – шрифт Times New Roman, кегль 10, інтервал – 1,1, абзацного відступу немає, інтервали перед і після – 0 мм, слово «Таблиця» та її номер – з вирівнюванням вправо, назва таблиці (якщо вона є) – з вирівнюванням по центру.

Для підрисункових підписів – шрифт Times New Roman, кегль 10, інтервал – 1,1, абзацного відступу немає, інтервали перед і після – 0 мм, вирівнювання по центру.

Для переліку посилань та відомостей про авторів – шрифт Times New Roman, кегль 9, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 0 мм, вирівнювання по ширині.

Для рефератів – шрифт Times New Roman, кегль 10, інтервал – 1,1, абзацний відступ – 8 мм, інтервали перед і після – 0 мм, вирівнювання по ширині.

Формули набираються у редакторі формул Microsoft Equation або MathType, розташовуються у центрі робочого поля, нумерація – з правої сторони поля. Для цього

необхідно весь рядок розташувати справа, а потім вирівняти формулу табуляціями так, щоб вона розташовувалася по центру. Відступ зверху і знизу – по 6 пунктів. Нумерація формул усередині кожної статті наскрізна.

Формули, а також їхні складові, присутні у тексті, набираються з такими параметрами (див. рис. 1).

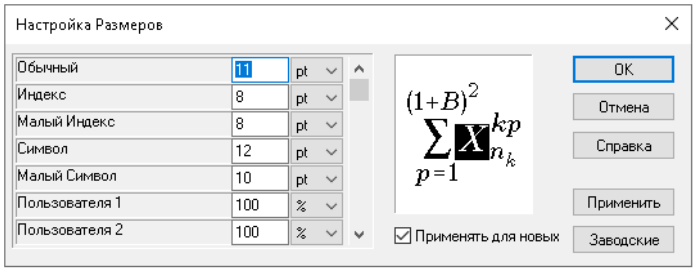


Рис. 1. Параметри настроювання розмірів редактора формул MathType

Кожна таблиця виконується та розташовується в тексті одразу після посилання на неї. Усі таблиці у статті обов'язково нумеруються, незважаючи на їх кількість. Таблиця відокремлюється від попереднього та наступного тексту (таблиці, рисунку тощо) одним порожнім рядком.

Дані всієї таблиці набираються шрифтом розміром 10 пунктів, розміщуються по центру; у випадках, коли необхідно показати розрядність, – вирівнювання за знаком. Товщина сітки таблиці – 1 пункт. Приклад оформлення таблиці наведено на рис. 2.

Таблиця 1

Множина описів сутностей функціональної задачі

ID	Найменування
1	Academic_load
2	Academic
3	Department
4	Individual_plan
5	Academic_section

Рис. 2. Приклад оформлення таблиці у тексті статті.

Бажано таблицю зі сторінки на сторінку не переносити. Якщо таблиця не може розміститися на сторінці, її поділяють на частини. У кожній частині таблиці повторюють її головку та боковик або замінюють їх відповідно номерами колонок або рядків, нумеруючи їх арабськими цифрами на першій частині таблиці. Слово «Таблиця» подається лише над першою її частиною. Над наступними її частинами праворуч друкується: «Продовження таблиці», а на останній – «Кінець таблиці», в усіх випадках вказується номер таблиці.

Кожен рисунок виконується та розташовується в тексті одразу після посилання на нього. Усі рисунки в статті обов'язково нумеруються, незважаючи на їх кількість. Необхідно вставляти рисунки у текст як графічні об'єкти (файли з розширенням .bmp, .jpg, .tiff чи .png, якість не менше 300 dpi), об'єкти MS Word або MS Visio.

Рисунок відокремлюється від попереднього та наступного тексту (таблиці, рисунку тощо) одним порожнім рядком.

Кожен рисунок повинен мати підрисунковий підпис, в якому вказується номер та, у

випадку необхідності, назва рисунку. Якщо рисунок займає менше 50 % ширини робочого поля, то можна зробити обтікання рисунку текстом, розташувавши його ліворуч або праворуч від робочого поля. Приклад рисунку з підписом наведений на рис. 3.

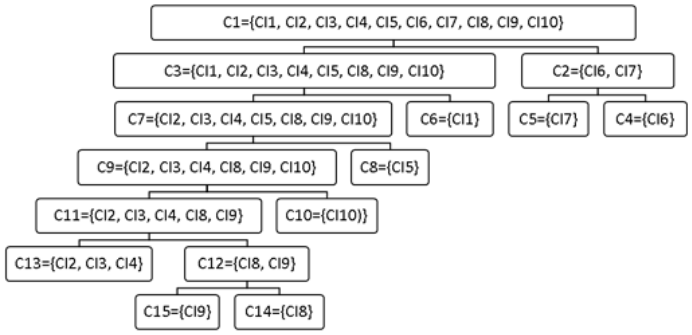


Рис. 3. Приклад виконання рисунку та підрисункового підпису

Посилання на літературні та електронні джерела у тексті статті позначаються у квадратних дужках [1]. До переліку посилань включаються тільки ті роботи, на які посилається автор статті. Посилання на неопубліковані роботи не допускаються.

Для оформлення переліку посилань слід використовувати один з таких шаблонів:

а) шаблон IEEE (автоматичне оформлення за шаблоном IEEE https://www.citethisforme.com/ieee/source-type);

б) положення ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання» та ДСТУ 3582:2013 «Інформація та документація. Бібліографічний опис. Скорочення слів і словосполучень українською мовою. Загальні вимоги та правила».

Кожен з цих шаблонів слід використовувати для оформлення усіх елементів переліку посилань. Використання двох шаблонів для оформлення одного й того ж переліку посилань неприпустимо.

Кожне посилання у переліку посилань наводиться за порядком появи цих посилань у тексті статті.

У переліку посилань бажано використовувати посилання на сучасні публікації, вік яких не перевищує п’яти років у момент подачі статті до редакції. Крім того, під час формування переліку посилань статті необхідно дотримуватися такого розподілу: самоцитування – до 20 %, цитування зарубіжних публікацій – не менше 50%.

Відомості про авторів слід наводити українською та англійською мовами. У відомості про авторів слід включати: повні прізвище, ім’я та по-батькові; вчений ступінь (за наявності); вчене звання (за наявності); посаду; країну, місто; e-mail (вкрай бажано вказувати корпоративний e-mail, можна вказувати кілька e-mail, на які ви бажаєте отримувати повідомлення від редакції та читачів, які можуть зацікавитися вашою статтею) ; ORCID.

Реферат набирається українською та англійською мовами. Реферат повинен бути змістовним, дотримуватися логіки опису результатів у статті та давати можливість встановити її основний зміст. Реферат не повинен містити формул та рисунків. Необхідні символи в рефераті необхідно додавати через функцію вставки символів.

Реферат містить: УДК, назву статті (напівжирним шрифтом), ініціали та прізвища авторів (курсивом), текст (не менше 1800 друкованих знаків з пробілами та ключовими словами), ключові слова, кількість таблиць, рисунків та посилань у статті.

Ключові слова повинні містити до 10 слів, а не словосполучень, без використання абревіатур, в іменному відмінку, розділятися крапкою з комою.

Реферати надаються до редколегії разом із статтею у вигляді окремого файлу.

4. Правила надсилання статей та подальшої взаємодії з редакційною колегією збірника

До редколегії збірника «АСУ та прилади автоматики» слід надсилати такі матеріали:

– файл у форматі .doc або .docx з текстом статті українською мовою;

- файл у форматі .doc або .docx з текстом статті англійською мовою (якщо автори бажають опублікувати статтю у збірнику англійською мовою);

– файл (у форматі .doc або .docx з текстами рефератів статті українською та англійською мовами;

– відскановану копію експертного висновку з дозволом опублікувати матеріали статті у відкритому друку. В разі потреби експертні висновки для авторів – співробітників (студентів, аспірантів тощо) ХНУРЕ можуть оформлюватися редколегією централізовано.

Матеріали статей надсилати електронною поштою – за адресою maksym.ievlanov@nure.ua.

Кожна надіслана в редакцію стаття після проходження рецензування і при позитивному рішенні редколегії буде надрукована в найближчому випуску збірника. Для цього авторам від імені редколегії надсилається ліцензійний договір, який закріплює право першої публікації статті у збірнику «АСУ та прилади автоматики». Автори статті повинні підписати цей ліцензійний договір та завірити свої підписи печаткою організації, в якій вони працюють. Підписаний ліцензійний договір автори статті надсилають на адресу редколегії збірника.

Відповідальний випусковий В.М. Левикін
Редактор О.Є. Неумивакіна
Комп'ютерна верстка М.В. Євланов, О.Є. Неумивакіна
Дизайн обкладинки номера за участю Є.Чех

Підп. до друку 27.06.2025. Формат 60x841/8. Умов. друк. арк.
Обл.-вид. арк. 11,8. Тираж 300 прим.
Зам. № 144. Ціна договірна.

Харківський національний університет радіоелектроніки (ХНУРЕ).
Україна, 61166, Харків, пр. Науки, 14

Оригінал-макет підготовлено у редакційно-видавничому відділі ХНУРЕ,
Україна, 61166, Харків, пр. Науки, 14

Збірник віддруковано в ТОВ «ДРУКАРНЯ МАДРИД»
61024, м. Харків, вул. Гуданова, 18
Тел.: +38(057)7565325
Свідоцтво суб'єкта видавничої справи
Серія ДК № 4399 від 27.08.2012 р.
www.madrid.in.ua e-mail:info@madrid.in.ua