МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ
УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ

КАФЕДРА КІБЕРБЕЗПЕКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МАТЕРІАЛИ

XVII-ої МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ
«FREE AND OPEN SOURCE SOFTWARE»

Дякуємо за підтримку

11-12 лютого 2026 р.
м. Харків

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

КАФЕДРА КІБЕРБЕЗПЕКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МАТЕРІАЛИ

XVII-ої МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

«FREE AND OPEN SOURCE SOFTWARE»

11-12 лютого 2026 р.

ХАРКІВ 2026

УДК 004
БК 32.973.202

Матеріали XVII-ої Міжнародної науково-практичної конференції «Free and Open Source Software», Харків, 11-12 лютого 2026 р. – Харків: Харківський національний економічний університет імені Семена Кузнеця, 2026. – 180 с.

Представлено матеріали пленарних та секційних засідань XVII-ої Міжнародної науково-практичної конференції «Free and Open Source Software». Обговорено основні проблеми, науково-технічні досягнення, впровадження і досвід використання сучасних технологій в області безкоштовних програмних продуктів, а також з відкритим вихідним кодом. Спеціальна секція присвячена публікаціям в рамках проєкту ERASMUS+ Jean Monnet EU-cyberconnect-UA "Стратегія кіберстандартизації ЄС для ефективного поєднання та цифрової інфраструктури: досвід для України". Для фахівців науково-дослідних, комерційних організацій, аспірантів та студентів.

Матеріали публікуються в авторській редакції.

Materials of the 17th International Scientific and Practical Conference "Free and Open Source Software", Kharkiv, February 11-12, 2026 - Kharkiv: Simon Kuznets Kharkiv National University of Economics, 2026. - 180 p.

The theses of the plenary and sectional meetings of the 17th International Scientific and Practical Conference "Free and Open Source Software" are presented. The main problems, scientific and technical achievements, implementation and experience of using modern technologies in the field of free software products, as well as open source, are discussed. A special section is devoted to publications within the framework of the ERASMUS+ Jean Monnet EU-cyberconnect-UA project "EU Cyber Standardization Strategy for Connectivity and Digital Infrastructure: Experience for Ukraine ". For specialists of research, commercial organizations, postgraduate students and students.

Materials are published in the author's editorial office.

Електронний варіант матеріалів конференції доступний на сайті конференції:
https://foss.kn-it.info/

ЗМІСТ

СЕКЦІЯ 1

СЕКЦІЯ 2

СЕКЦІЯ 3

СЕКЦІЯ 4

Секція 1

БЕЗКОШТОВНІ СЕРВІСИ ТА УТИЛІТИ

БЕЗКОШТОВНІ УТИЛІТИ КІБЕРБЕЗПЕКИ ТА ОПТИМІЗАЦІЇ ЯК ІНСТРУМЕНТ ФОРМУВАННЯ ЦИФРОВИХ КОМПЕТЕНТНОСТЕЙ СТУДЕНТІВ ІТ-СПЕЦІАЛЬНОСТЕЙ

Балим Г.В.
E-mail: balym@hrtt.kh.ua
Харків, Харківський радіотехнічний фаховий коледж

У процесі викладання ІТ-дисциплін дедалі більшої ваги набуває не лише передача теоретичних знань, а й формування практичних цифрових умінь здобувачів освіти. Відповідно до вимог стандартів фахової передвищої освіти у галузі інформаційних технологій, підготовка майбутніх фахівців має ґрунтуватися на компетентнісному та практикоорієнтованому підходах [1], [2], що відображає сучасні тенденції цифровізації освіти в Україні [3].

Інформаційна компетентність розглядається як інтегративна характеристика підготовки здобувача освіти та включає:

– цифрову грамотність і культуру роботи з інформацією;
– здатність використовувати системне та прикладне програмне забезпечення;
– розуміння принципів функціонування апаратного й програмного забезпечення;
– базові навички кібербезпеки та захисту даних;
– уміння аналізувати та оптимізувати інформаційні процеси;
– готовність до самостійного професійного розвитку.

Формування інформаційної культури здобувачів освіти розглядається як один із ключових напрямів модернізації освітнього середовища [4].

Ефективна робота з інформаційними системами потребує системного мислення та практичних навичок оптимізації цифрового середовища [5], а впровадження цифрових технологій у навчальний процес підсилює розвиток інформаційної культури студентів [3].

Дисципліна «Операційні системи» є однією з базових у підготовці ІТ-фахівців, оскільки формує розуміння логіки роботи комп’ютера, взаємодії програмного й апаратного забезпечення та основ адміністрування користувацького середовища. Операційна система виступає посередником між користувачем і ресурсами комп’ютера [6].

У педагогічній практиці можна спостерігати, що студенти значно краще засвоюють матеріал тоді, коли мають можливість не лише ознайомитися з теоретичними положеннями, а й безпосередньо працювати з інструментами аналізу системи, моніторингу ресурсів і захисту даних. Саме тому доцільним є поєднання теорії з використанням доступних безкоштовних програмних утиліт.

Під час практичних занять доцільно передбачати:

– моніторинг використання процесора та оперативної пам’яті;
– аналіз системних процесів і служб;
– керування автозавантаженням програм;
– аналіз структури дискового простору;
– резервне копіювання даних;
– ознайомлення з інструментами інформаційної безпеки.

Використання безкоштовних міжнародних програмних інструментів дозволяє організувати практичні заняття без додаткових фінансових витрат та забезпечує доступність навчального процесу.

Основні напрями практичних робіт:

– очищення кешу та тимчасових файлів;
– аналіз дискового простору;
– моніторинг навантаження процесора та пам’яті;
– перевірка швидкості інтернет-з’єднання;
– керування паролями та шифрування даних.

Далі представлено порівняльний аналіз безкоштовних програмних інструментів, що є найбільш доцільними для використання у навчальному процесі під час виконання практичних та лабораторних робіт, із зазначенням сформованих умінь та навичок (табл. 1.).

Таблиця 1 – Очищення кешу та тимчасових файлів.

Критерій	BleachBit	CCleaner Free
Тип ліцензії	Open-source	Безкоштовна версія
Країна походження	США / міжнародна спільнота	Велика Британія / ЄС
Платформи	Windows, Linux	Windows, macOS, Android
Інтерфейс	Мінімалістичний	Візуальний
Реклама	Відсутня	Присутня
Освітня доцільність	Поглиблене розуміння структури ОС	Швидка демонстрація результату

Сформовані уміння та навички: робота з файловою системою, аналіз структури даних та оптимізація ресурсів операційної системи (табл. 2).

Таблиця 2 – Аналіз дискового простору.

Критерій	WinDirStat	TreeSize Free
Тип ліцензії	Open-source	Безкоштовна версія
Країна походження	Німеччина	Німеччина
Візуалізація	Treemap-карти	Таблична структура
Швидкість	Середня	Висока
Освітня доцільність	Візуальний системний аналіз	Адміністративна оптимізація

Сформовані уміння та навички: аналіз структури даних, управління пам’яттю пристрою (табл. 3).

Таблиця 3 – Моніторинг навантаження процесора та пам’яті.

Критерій	Process Explorer	htop	Activity Monitor
Платформа	Windows	Linux	macOS
Розробник	Microsoft (США)	Open-source	Apple (США)
Інтерфейс	Графічний	Консольний	Графічний
Глибина аналізу	Дуже висока	Висока	Середня
Освітня доцільність	Поглиблене адміністрування	Формування системного мислення	Базовий рівень

Сформовані уміння та навички: аналіз процесів, керування службами (табл. 4).

Таблиця 4 – Перевірка швидкості інтернет-з’єднання

Критерій	Speedtest by Ookla	Fast.com
Платформа	Web, Android, iOS, Windows	Web, Android, iOS
Розробник	Ookla (США)	Netflix (США)
Інтерфейс	Деталізований	Мінімалістичний
Глибина аналізу	Висока	Базова
Освітня доцільність	Повний мережевий аналіз	Експрес-діагностика

Сформовані уміння та навички: мережева діагностика, інтерпретація параметрів з’єднання (табл. 5).

Таблиця 5 – Менеджери паролів та інструменти шифрування даних.

Критерій	Bitwarden	KeePass
Тип ліцензії	Open-source	Open-source
Країна походження	США	Німеччина
Тип зберігання	Хмарне + локальне	Локальне
Синхронізація	Автоматична	Ручна
Освітня доцільність	Сучасні хмарні рішення	Поглиблене локальне шифрування

Сформовані уміння та навички: цифрова гігієна, захист облікових даних, розуміння принципів шифрування. Застосування менеджерів паролів зменшує ризик несанкціонованого доступу до облікових записів.

Усі зазначені утиліти мають міжнародне або європейсько-американське походження, підтримуються відкритими спільнотами або відомими IT-компаніями. Це забезпечує прозорість розробки, регулярні оновлення безпеки та можливість легального використання в освітньому процесі.

Інтеграція безкоштовних цифрових інструментів у навчальний процес сприяє:

– розвитку інформаційних і цифрових компетентностей;

– формуванню навичок системного адміністрування;

– підвищенню обізнаності у сфері кібербезпеки;

– розвитку аналітичного та критичного мислення;

– підвищенню зацікавленості студентів у вивченні IT-дисциплін.

Такий підхід відповідає нормативним засадам розвитку фахової передвищої освіти в Україні [7].

Практичне впровадження безкоштовних утиліт оптимізації та сервісів захисту даних у межах дисципліни «Операційні системи» демонструє позитивний вплив на рівень засвоєння матеріалу здобувачами освіти. Поєднання теорії з реальними цифровими інструментами формує стійкі інформаційні компетентності та підвищує якість підготовки майбутніх IT-фахівців.

Література

[1] Стандарт фахової передвищої освіти України за спеціальністю 121 «Інженерія програмного забезпечення». Освітньо-професійний ступінь — фаховий молодший бакалавр. Міністерство освіти і науки України, 2021.

[2] Стандарт фахової передвищої освіти України за спеціальністю 123 «Комп’ютерна інженерія». Освітньо-професійний ступінь — фаховий молодший бакалавр. Міністерство освіти і науки України, 2022.

[3] Морзе Н. В., Барна О. В. Інформаційно-комунікаційні технології в освіті. — Київ: ВНV, 2016.

[4] Биков В. Ю. Моделі організаційних систем відкритої освіти. — Київ: Атіка, 2009.

[5] Tanenbaum A., Bos H. Modern Operating Systems. — Pearson Education Limited, 2015.

[6] Silberschatz A., Galvin P., Gagne G. Operating System Concepts. — Wiley, 2019.

[7] Закон України № 2745-VIII “Про фахову передвищу освіту”

ІНСТРУМЕНТИ КЕРУВАННЯ ПАРОЛЯМИ ДЛЯ КОРИСТУВАЧІВ

Загнібеда А.О.
Керівник: Міхєєв І.А.
E-Mail: anastasiia.zagnibeda@icloud.com
Харків, Харківський національний економічний університет імені Семена Кузнеця

На сучасному етапі розвитку цифрових технологій питання безпеки паролів залишається актуальним. Значна частина користувачів застосовує паролі, не приділяючи належної уваги способам їх захисту та зберігання, а також можливим наслідкам недотримання базових правил кібербезпеки. За даними статті Forbes [1], у відкритий доступ потрапило близько 16 мільярдів паролів, пов'язаних з обліковими записами таких сервісів, як Apple, Facebook, Google та інших компаній. Це свідчить про низький рівень обізнаності користувачів щодо захисту персональних даних і підтверджує актуальність проблеми безпеки паролів.

Одним із ключових аспектів дослідження стало вивчення частоти зміни паролів користувачами, оскільки регулярне оновлення паролів суттєво знижує ризик компрометації облікових записів та підвищує рівень захисту персональних даних [2]. Для аналізу зазначеної проблеми було проведено власне опитування серед 53 респондентів різних вікових категорій та з різним рівнем обізнаності у сфері кібербезпеки (рис. 1).

Рисунок 1 – Результати опитування щодо регулярності зміни паролів

Отримані результати свідчать про те, що значна частина користувачів не приділяє достатньої уваги регулярній зміні паролів, що може негативно впливати на рівень захисту їхніх персональних даних. У зв'язку з цим виникає питання, яким чином користувачі зберігають свої паролі. Для цього було проаналізовано способи збереження паролів, зокрема зберігання у браузері, використання менеджерів паролів, запам'ятовування або записування паролів у файлах чи на папері (рис. 2) [2].

За результатами аналізу було встановлено, що більшість користувачів надають перевагу запам'ятовуванню паролів. Такий підхід не є оптимальним з точки зору цифрової безпеки, оскільки змушує користувачів використовувати прості або повторювані паролі, що підвищує ризик несанкціонованого доступу до облікових записів. Альтернативним та більш безпечним рішенням є використання менеджерів паролів, які дозволяють генерувати складні паролі та зберігати їх у захищеному вигляді.

Рисунок 2 – Результати опитування щодо способів зберігання паролів

Наступним кроком є дослідження існуючих інструментів для зберігання, генерації та керування паролями, зокрема рішення хмарного та локального типу, а також проведення порівняльного аналізу їх сильних та слабких сторін.

Першою групою є менеджери паролів хмарного типу [4-9]. Такі сервіси як Google Password Manager, Dashlane, Bitwarden, Proton, 1Password та LastPass допомагають зберігати та керувати паролями, генерувати надійні паролі та автоматично заповнювати форми. Вони забезпечують безпечне зберігання паролів, синхронізацію між пристроями та підтримку принципу zero-knowledge. Однак деякі з них мають додаткові функції:

– Dashlane – зручний інтерфейс, підтримка декількох пристроїв та автоматична синхронізація .

– 1Password – простий у використанні, підтримка командної роботи та генерації складних паролів

Другою групою є локальні менеджери паролів [10, 11]. Такі сервіси як KeePass та Password Safe дозволяють зберігати паролі локально на пристрої користувача, створювати складні паролі та керувати ними без підключення до хмари. Вони забезпечують високий рівень шифрування та принцип zero-knowledge. Деякі їх функціональні можливості:

– KeePass – відкритий код, можливість використання плагінів та висока надійність шифрування.

– Password Safe – простий та надійний інтерфейс, локальне зберігання паролів та автоматична генерація.

На основі описаних функцій та особливостей хмарних і локальних менеджерів паролів було сформовано таблицю порівняння, яка дозволяє наочно оцінити підтримувані сервіси функції, тип рішення та принципи зберігання паролів. Це дає змогу зрозуміти сильні та слабкі сторони кожного менеджера та обрати оптимальний інструмент для безпечного зберігання та керування паролями користувачів (табл. 1).

Таблиця 1 – Порівняння менеджерів паролейза ключовими критеріями

	Google	Dashlane	Bitwarden	Proton	1Password	LastPass	KeePass	Password Safe
Генератор паролів	✅	✅	✅	✅	✅	✅	✅	✅
Автозаповнення	✅	✅	✅	✅	✅	✅	✅	✅
Зберігання локально	✅	✅	✅	✅	✅	✅	✅	✅
Синхронізація / облако	✅	✅	✅	✅	✅	✅	❌	❌
Zero-knowledge	❌	✅	✅	✅	✅	✅	✅	✅
Тип рішення	хмарний	хмарний	хмарний	хмарний	хмарний	хмарний	локальний	локальний
Мова інтерфейсу	ENG	ENG	ENG	ENG	ENG	ENG	ENG	ENG

Аналіз розглянутих менеджерів паролів показав, що кожен сервіс має свої переваги та обмеження, які визначають ефективність його використання залежно від потреб користувача. Хмарні рішення забезпечують синхронізацію та доступ з різних пристроїв, тоді як локальні сервіси надають повний контроль над даними. Для підвищення безпеки доцільно комбінувати різні сервіси, використовуючи їхні можливості для генерації та збереження паролів, що дозволяє користувачам надійно захищати свої дані. Результати дослідження будуть використані для побудови веб-сервісу із відповідним функціоналом.

Література

[1] Офіційний сайт Forbes [Електронний ресурс]. – Режим доступу: https://www.forbes.com/sites/daveywinder/2025/06/20/16-billion-apple-facebook-google-passwords-leaked---change-yours-now/

[2] Звички користувачів у роботі з паролями [Електронний ресурс]. – Режим доступу: https://docs.google.com/forms/d/1rkW4BBQVS1EHmQKKuTYx8PgszVUL9yERzgcUvX2DI3k/viewanalytics

[3] Electro IQ [Електронний ресурс]. – Режим доступу: https://electroiq.com/stats/password-manager-statistics/#Top_10_Most_Common_Passwords_Statistics

[4] Google [Електронний ресурс]. – Режим доступу: https://passwords.google.com/

[5] Dashlane [Електронний ресурс]. – Режим доступу: https://www.dashlane.com/

[6] Bitwarden [Електронний ресурс]. – Режим доступу: https://bitwarden.com/

[7] Proton [Електронний ресурс]. – Режим доступу: https://proton.me/pass

[8] 1Password [Електронний ресурс]. – Режим доступу: https://1password.com/

[9] Lastpass [Електронний ресурс]. – Режим доступу: https://www.lastpass.com/password-manager

[10] KeePass [Електронний ресурс]. – Режим доступу: https://keepass.info/

[11] Password Safe [Електронний ресурс]. – Режим доступу: https://pwsafe.org/

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ РЕКОМЕНДУВАННЯ ІНДИВІДУАЛЬНИХ КОНТАКТІВ КОРИСТУВАЧІВ СОЦІАЛЬНОЇ МЕРЕЖІ

Мацура М.А., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

У сучасних соціальних мережах спостерігається постійне зростання кількості користувачів та обсягу публікацій, що ускладнює процес пошуку релевантного та цікавого контенту. Користувачі часто не мають змоги самостійно знаходити сторінки, які відповідають їхнім інтересам, що знижує загальну ефективність взаємодії з платформою. У зв'язку з цим актуальним є завдання розробки програмного забезпечення рекомендування, яке дозволяє автоматично формувати персоналізовані пропозиції для підписки на основі аналізу поведінки користувачів та структури соціальної мережі. Такі результати можуть використовуватися в програмних засобах, які взаємодіють з самою соціальною мережею для створення рекомендацій у них замість самої соціальної мережі.

У даній роботі розроблено програмне забезпечення для рекомендації користувачів у соціальній мережі X/Twitter з використанням графового підходу. Мережа представлена у вигляді орієнтованого графа, де вершини відповідають користувачам, а ребра – фактам підписки між ними. Для реалізації та тестування алгоритму використано відкриті вибырки даних, отримані з платформи Kaggle, що містять інформацію про користувачів та їх взаємозв'язки. Для подальшої обробки граф було перетворено у матрицю суміжності, яка дозволяє компактно описати структуру підписок та застосовувати методи машинного навчання для аналізу схожості користувачів.

Кожен користувач у системі представлений у вигляді бінарного вектора підписок, де кожна компонента відображає наявність або відсутність підписки на іншого користувача. Такий підхід дозволяє розглядати задачу рекомендацій як задачу пошуку найближчих сусідів у високорозмірному просторі. Для визначення ступеня подібності між користувачами застосовано алгоритм Nearest Neighbors з косинусною метрикою відстані [1]. Косинусна метрика є доцільною для аналізу розріджених даних соціальних мереж, оскільки вона оцінює схожість структури інтересів незалежно від загальної кількості підписок користувача [2].

Формування рекомендацій базується на принципах колаборативної фільтрації. На першому етапі для заданого користувача визначається множина найбільш схожих користувачів. Далі аналізуються підписки цих користувачів з метою виявлення акаунтів, які можуть бути потенційно цікавими. Для кожного кандидата обчислюється індекс інтересу, який формується з урахуванням кількості схожих користувачів, що підписані на відповідний акаунт, а також ступеня їх подібності до цільового користувача. Такий підхід дозволяє надавати перевагу тим рекомендаціям, які підтримуються більшою кількістю користувачів зі схожими інтересами.

Реалізація програмного забезпечення виконана з використанням мови програмування Python та спеціалізованих бібліотек для аналізу даних і машинного навчання, зокрема pandas, NumPy, SciPy та scikit-learn. Такий підхід забезпечує гнучкість і масштабованість системи рекомендацій та може бути адаптований для інших соціальних мереж або сервісів, що використовують графові структури даних. Отримані результати підтверджують доцільність використання методів аналізу графів та алгоритмів пошуку схожих користувачів для побудови персоналізованих рекомендацій у соціальних інформаційних системах.

Література

[1] Scikit-learn: Nearest Neighbors [Electronic resource]. – Access mode: https://scikit-learn.org/stable/modules/neighbors.html

[2] Machine Learning: Cosine Similarity for Vector Space Models (Part III) [Electronic resource]. – Access mode: https://blog.christianperone.com/2013/09/machine-learning-cosine-similarity-for-vector-space-models-part-iii/

СЕРВІСИ ПЕРЕВІРКИ БЕЗПЕЧНОСТІ ПАРОЛІВ ТА ВИЯВЛЕННЯ КОМПРОМЕТАЦІЇ

Міхєєв Є. А.
Керівник: Долгова Н.Г.
Харків, Харківський національний економічний університет імені Семена Кузнеця

Сучасна кіберполіція наголошує, що надійний пароль має складатися щонайменше з 8 символів, містити великі та малі літери, цифри та спеціальні знаки, а його головною характеристикою є відсутність логічних зв'язків із персональними даними користувача. Використання менеджерів паролів та двофакторної автентифікації є базовим рівнем захисту, який рекомендується для запобігання зламу облікових записів [1].

Для перевірки вже існуючих паролів на компрометацію використовується база Pwned Passwords від сервісу *Have I Been Pwned*. Цей інструмент містить сотні мільйонів паролів, що були оприлюднені внаслідок витоків даних, і дозволяє перевірити комбінацію за допомогою хешування SHA-1. Такий підхід гарантує безпеку, оскільки сам пароль не передається в повному обсязі, а перевіряється лише частина його анонімного хешу [2].

Оцінка стійкості пароля в реальному часі ефективно реалізується за допомогою алгоритму zxcvbn. На відміну від застарілих методів, він використовує підрахунок ентропії та зіставлення з широкими списками, що включають прізвища, популярні англійські слова та цифрові послідовності. Це дозволяє сервісу оцінити, скільки саме спроб знадобиться зловмиснику для зламу конкретної комбінації, виходячи з її структури [3].

Компанія Google інтегрувала систему «Перевірка паролів» безпосередньо у свій менеджер паролів. Система автоматично аналізує збережені дані на предмет їхньої надійності, повторного використання на різних сайтах та наявності у базах відомих витоків. Якщо пароль визнано вразливим, користувач отримує миттєве сповіщення з рекомендацією змінити його для захисту облікового запису [4].

На рівні розробки стандарт OWASP визначає слабку автентифікацію як одну з ключових вразливостей. Для захисту систем рекомендується впроваджувати обмеження на кількість спроб входу та використовувати перевірку за списками найпопулярніших слабких паролів. Це мінімізує ризик автоматизованих атак, спрямованих на підбір сесійних ідентифікаторів [5].

На мою думку, попри стрімкий розвиток біометрії та технологій Passkeys, паролі залишатимуться основним методом ідентифікації ще тривалий час. Проблема безпеки сьогодні змістилася з «складності» пароля на його «унікальність». Навіть дуже складний пароль стає марним, якщо він скомпрометований на одному з десятків сайтів, де зареєстрований користувач. Тому сервіси моніторингу витоків та інтегровані аудитори безпеки є не просто допоміжними інструментами, а критично важливою частиною сучасної системи захисту, що дозволяє вчасно реагувати на загрози.

Література

[1] Кіберполіція України: Як створити надійний пароль та захистити свої аккаунти [Електронний ресурс]. – Режим доступу: https://cyberpolice.gov.ua/article/yak-stvoryty-nadijnyj-parol-7451/

[2] Have I Been Pwned: Pwned Passwords [Електронний ресурс]. – Режим доступу: https://haveibeenpwned.com/Passwords

[3] Dropbox Tech Blog. zxcvbn: realistic password strength estimation [Електронний ресурс]. – Режим доступу: https://dropbox.tech/security/zxcvbn-realistic-password-strength-estimation

[4] Google Safety Center. Як працює перевірка паролів у вашому обліковому записі [Електронний ресурс]. – Режим доступу: https://safety.google/intl/uk/authentication/password-manager/

[5] OWASP Top 10: Broken Authentication and Session Management [Електронний ресурс]. – Режим доступу: https://owasp.org/www-project-top-ten/

AI-ІНСТРУМЕНТИ ДЛЯ КООРДИНАЦІЇ АСИНХРОННОЇ ВЗАЄМОДІЇ УЧАСНИКІВ ГІБРИДНИХ IT-КОМАНД

Слісаренко М.В.
Керівник: Назарова С.О.
E-mail: slisarenko.mykola@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

Гібридна форма організації праці, за якої поєднуються традиційна та дистанційна робота, стала домінуючою в IT-секторі. За даними Gallup, 52% компаній впровадили такий формат, а працівники в середньому працюють дистанційно 1–2 дні на тиждень [1]. Це призводить до формування гібридних команд — команд, учасники яких мають можливість працювати як у спільному офісному просторі, так і дистанційно, при цьому конфігурація присутніх учасників команди змінюється залежно від дня [2].

Водночас така форма організації праці створює значні виклики для командної координації, які не вирішуються традиційними методами управління персоналом. Несинхронність робочих графіків суттєво обмежує можливості синхронної взаємодії: учасники проєкту можуть ніколи не працювати в офісі одночасно, що призводить до затримок у прийнятті рішень і отриманні зворотного зв'язку [3]. Саме тому зростає попит на AI-інструменти, здатні автоматизувати рутинну комунікацію та підтримувати асинхронну взаємодію гібридних команд. У цій роботі розглядаються виключно безкоштовні рішення або рішення з відкритим кодом — вибір, який відповідає потребами невеликих команд та стартапів, де бюджет на програмне забезпечення обмежений.

Наразі доступні AI-інструменти допомагають вирішувати такі задачі координації: втрата контексту учасниками, які були відсутніми на командних зустрічах; неефективність синхронних «stand-up» зустрічей через асинхронні графіки; потреба у швидкому уточненні інформації без організації дзвінків; необхідність централізованої бази знань для онбордингу та документації. Наразі доступні різні AI-інструменти, що вирішують кожну з цих задач: асистенти для транскрипції зустрічей, асинхронні «stand-up» боти, платформи відеоповідомлень та локальні AI-чат-боти.

Meetily — AI-асистент з відкритим кодом із повністю локальною обробкою даних — забезпечує автоматичну транскрипцію та генерацію підсумків зустрічі без залежності від хмарних сервісів [3]. Платформа підтримує основні відеоплатформи (Zoom, Google Meet, Microsoft Teams) та працює на базі локальних моделей, що гарантує конфіденційність корпоративних даних. Інший AI-асистент – Tactiq пропонує безкоштовний план із транскрипцією в реальному часі та AI-згенерованими підсумками зустрічі, що дозволяє відсутнім учасникам швидко ознайомитися з ключовими рішеннями [4]. Варто зауважити, що обидва інструменти найкраще працюють з англомовним контентом, а для української мови точність транскрипції нижча, що обмежує їх застосування в україномовних командах.

Geekbot — безкоштовний бот для команд до 10 осіб, що автоматизує асинхронні «stand-up» звіти в Slack та Microsoft Teams [5]. Бот надсилає стандартні питання у зручний для кожного учасника час з урахуванням часових зон та публікує відповіді в спільному каналі. Типовий набір питань: «Що зроблено вчора?», «Що заплановано на сьогодні?», «Які є перешкоди?». Це забезпечує прозорість прогресу без необхідності синхронних зустрічей. Додатково платформа підтримує проведення ретроспектив, опитування та моніторинг настрою команди. Kyber пропонує безкоштовний план для команд до 5 осіб із функціями планування завдань та Kanban-дошок. На практиці впровадження таких ботів потребує дисципліни: якщо учасники ігнорують питання або відповідають формально, інструмент швидко втрачає цінність.

Cap — альтернатива комерційному AI-асистенту Loom з відкритим кодом, що дозволяє записувати екран із вебкамерою та миттєво ділитися посиланням [7]. AI-функції платформи автоматично генерують субтитри, заголовки та підсумки відео. Такий підхід особливо ефективний для процедури перевірки коду, технічних пояснень та онбордингу,

коли текстова комунікація недостатня, а синхронний дзвінок неможливий через різницю в робочих графіках. Наприклад, замість 30-хвилинного дзвінка для демонстрації нової функціональності розробник може записати 3-хвилинне відео та надіслати посилання команді. Водночас відеоформат має обмеження: перегляд займає більше часу порівняно з текстом, тому оптимальною вважається тривалість до 3 хвилин.

Команди, що працюють під NDA або з проектами, що підпадають під вимоги GDPR, не завжди можуть використовувати хмарні сервіси для обробки внутрішньої інформації. Для них існує можливість розгорнути власного AI-помічника. Ollama у поєднанні з Open WebUI дозволяє створити корпоративний чат-бот на базі open-source моделей без передачі даних на зовнішні сервери [6]. Такий асистент може відповідати на типові питання про проєкт, допомагати з документацією та підтримувати онбординг нових учасників команди. Інтеграція з open-source месенджерами (Mattermost, Rocket.Chat) забезпечує єдину екосистему для комунікацій та AI-підтримки учасників гібридної команди. Головний недолік — необхідність технічної експертизи для початкового налаштування та подальшої підтримки, що робить це рішення непридатним для команд без DevOps-спеціаліста.

Аналіз функціоналу існуючих безкоштовних AI-інструментів для координації асинхронної взаємодії учасників гібридних IT-команд дозволяє визначити чотири сценарії впровадження AI-інструментів залежно від організаційного контексту. За критерієм інфраструктури: команди, що використовують хмарні платформи (Slack, Microsoft Teams) можуть використовувати комбінацію Geekbot, Tactiq та Cap, що не потребує технічної підготовки. За критерієм конфіденційності: проєкти з вимогами до приватності даних потребують інтеграції Meetily та Ollama з локальним розгортанням. За критерієм бюджету: стартапам достатньо Kyber та вбудованих AI-функцій Zoom чи Google Meet. За критерієм специфіки роботи: технічні команди з інтенсивною процедурою перевірки коду можуть використовувати Cap як основний канал асинхронної комунікації.

Підсумовуючи, результати аналізу існуючих безкоштовних AI-інструменти для координації асинхронної взаємодії учасників гібридних команд слід констатувати, що вони уже здатні суттєво зменшити втрати від асинхронності графіків. Головні переваги — автоматизація рутини, прозорість для відсутніх учасників, можливість локального розгортання та заміна частини синхронних дзвінків відеоповідомленнями. Разом із тим, жоден інструмент не є універсальним: мовні обмеження (більшість оптимізовано під англійську), потреба в технічній підтримці та залежність від дисципліни команди — фактори, які слід враховувати при впровадженні. Подальші дослідження варто спрямувати на емпіричну перевірку ефективності цих інструментів у реальних проєктних командах українського IT-сектору.

Література

[1] Gallup Inc. Indicator: Hybrid Work. 2022. URL: https://www.gallup.com/401384/indicator-hybrid-work.aspx (дата звернення: 28.01.2026).

[2] Handke L., Aldana A., Costa P. L., O'Neill T. A. Hybrid teamwork: What we know and where we can go from here. Small Group Research. 2024. Vol. 55, No. 6. P. 789–832.

[3] Meetily: Open Source AI Meeting Assistant. URL: https://meetily.ai (дата звернення: 28.01.2026).

[4] Tactiq: AI Meeting Transcripts. URL: https://tactiq.io (дата звернення: 28.01.2026).

[5] Geekbot: Async Standups in Slack. URL: https://geekbot.com (дата звернення: 28.01.2026).

[6] Ollama GitHub Repository. URL: https://github.com/ollama/ollama (дата звернення: 28.01.2026).

[7] Cap: Open Source Loom Alternative. URL: https://cap.so (дата звернення: 28.01.2026).

ВИКОРИСТАННЯ KALIGPT ДЛЯ ВИВЧЕННЯ ОСНОВ КІБЕРГІГІЄНИ, КІБЕРБЕЗПЕКИ, ЕТИЧНОГО ХАКІНГУ ТА ПЕНТЕСТИНГУ

Шапо В.Ф., Олексюк Д.І., Миндру А.М.
E-mail: vladlen.shapo@gmail.com, danaoleksuk191@gmail.com, andreymyndru@icloud.com
Одеса, Інститут Військово-Морських Сил

Кібергігієна та кібербезпека – напрям, який вимушує постійно отримувати нові знання і з точки зору розширення світогляду, і з точки зору глибокого проникнення в численні деталі. Одним з програмних продуктів, відомих в цій області на протязі тривалого часу, є операційна система Kali Linux [1], яка має величезну кількість вбудованих програмних інструментів в області кібербезпеки. Очевидно, що при виникненні потенційних або реальних проблем з кібербезпекою можна використовувати загальний пошук в Інтернеті, використати відомий ресурс Stack Overflow [2], працюючий в стилі питання-відповідь і т.д. Але нещодавно з'явився KaliGPT, – безкоштовний спеціалізований штучний інтелект, який може в значній частині замінити звичні інструменти.

KaliGPT – спеціальна версія ChatGPT, яка створена для спеціалістів та ентузіастів інформаційної безпеки та взагалі усіх, хто цікавиться напрямом кібербезпеки в широкому сенсі. Головна ціль проекту – зробити інструменти Kali Linux максимально зрозумілими, зручними і доступними і для новачків, і для досвідчених спеціалістів. KaliGPT може допомогти розібратися з командами, підібрати потрібні утиліти, пояснити логіку роботи скриптів и підказати, як вирішувати несподівані задачі в умовах відсутності часу та знань.

KaliGPT може бути дуже корисним наступним основним групам користувачів.

1. Тим, хто лише починає працювати з Kali Linux.
2. Професіональним пентестерам, в яких завжди не вистачає часу на глибоке заглиблення в проблему.
3. Етичним хакерам, які не хочуть порушувати закони та етичні аспекти.
4. IT-спеціалістам, які повинні швидко засвоювати нові методи захисту і атак.
5. Студентам та викладачам, які планують навчатися в області кібербезпеки взагалі та шукають приклади задач та рішень.

Нема сенсу штучно обмежувати вказаний список, оскільки KaliGPT – універсальний навігатор в величезному світі програмних інструментів, скриптів, підходів та методів.

KaliGPT доступний як окремий чат-бот, повністю орієнтований на тематику Kali Linux та інформаційної безпеки. Тому користувач не буде вимушений пояснювати ШІ, що таке metasploit [3] чи що таке nmap [4], оскільки KaliGPT розуміє контекст та «підтягує» відповідні дані.

Можна питати про команди Kali Linux: що робить відповідна команда, як її коректно запускати, які вона має додаткові параметри. KaliGPT може допомогти підібрати інструменти для вирішення конкретної задачі (наприклад, для проведення фішинг-тесту чи тесту на стійкість до SQL-ін'єкцій). Він також може пояснювати помилки, підказувати, як їх виправляти, розшифровувати складні повідомлення терміналу. Можна також просити KaliGPT пояснювати можливі сценарії: як виконати конкретний тип атаки, що зробити для захисту. Питання по операційній системі, конфігураціям, bash-скриптам, коду на Python чи PowerShell і т.д. вже інтегровано в модель. Діалог з KaliGPT може бути, наприклад, таким.

Питання: як правильно сканувати мережу за допомогою nmap, щоб це не відобразилось в лог-файлі?

Відповідь: використати параметри тихого сканування (-sS), підбір часу (-T0/1), використання параметрів --source-port.

Питання: що робити, якщо ettercap не запускається з-за помилки libpcap?

Відповідь: виконати послідовність кроків з діагностики (перевірити наявність бібліотеки, перезапустити з правами суперкористувача.

KaliGPT не надасть готових експлойтів, але дозволить зекономити багато часу на розуміння суті інструментів та методик. Його основні можливості описано нижче.

1. Пояснення принципів роботи інструментів Kali Linux від простого ping до екзотичних snmpwalk та hydra.

2. Поміч у створенні та оптимізації скриптів Bash, Python, PowerShell.

3. Діагностика помилок: чому не запускається сервіс, не працює маршрутизація, що робити після отримання повідомлення permission denied і т.ін.

4. Порівняння програмних інструментів для вирішення задачі: який обрати для сканування, а який для зламу перебором (брутфорса).

5. Пояснення понять: що таке sqlmap, paywall, ARP spoofing, DNS poisoning, чому sslstrip втрачає популярність, а mitmproxy її набирає.

6. Поради по досвіду: як сховати сліди сканування, що робити при блокуванні міжмережевим екраном, як мінімізувати запис власних дій в лог-файлах.

7. Рекомендації по навчальним ресурсам, документації, корисним спільнотам (HackerOne, Hack The Box, TryHackMe і т.д.).

KaliGPT добре справляється і з нестандартними задачами, наприклад, пояснюючи різницю між реальною атакою та лабораторною вправою, наводить приклади типових помилок новачків, ділиться порадами.

З KaliGPT не треба витрачати час на фільтрацію застарілих або невідповідних порад, бо відповіді надаються відразу в зрозумілому вигляді. У разі необхідності можна уточнити деталі на кшталт «Поясни, чому цей скрипт не працює саме на моїй версії Kali Linux», при цьому ШІ ніколи не втомлюється відповідати. Можна отримати структуровані пояснення та отримати коротку інформацію або докладну інструкцію. Очевидно, що це не відміняє необхідності самому невпинно вчитися, але дозволяє зберегти дорогоцінний час та зберігає нерви.

Можливі реальні сценарії застосування KaliGPT представлені нижче.

1. Навчання та виконання лабораторних та практичних робіт: можна просити пояснити завдання, розібрати типову лабораторну чи практичну роботу або крок за кроком розписати етапи тесту на проникнення.

2. Робота в режимі реального часу: як тільки появляється помилка в терміналі, можна відразу ж спитати, що вона означає і як її обійти.

3. Підготовка до змагань CTF (Capture The Flag, захоплення прапора): KaliGPT може допомогти швидко згадати синтаксис команд, пояснить логіку рішення, підкаже приклади зі схожими задачами.

4. Розбір методик Red Team та Blue Team: можна вивчати методи атак та захисту, отримуючи структуровані інструкції та контрзаходи.

5. Автоматизація задач: KaliGPT може допомогти створити найпростіший bash-скрипт для автоматизації сканування або підготовки звіту.

6. Підбір програмних інструментів для роботи з конкретною платформою або задачею: наприклад, який програмний засіб обрати для аналізу трафіку на Windows, а який для Android.

В кожному конкретному випадку користувач отримає не просто формальні результати, а індивідуальну рекомендацію, в якій взято до уваги контекст конкретної задачі.

Очевидно також, що ШІ не є чародієм і не може замінити реального професійного досвіду. KaliGPT також не ідеальний і не може робити, наприклад, перераховане нижче.

1. Не генерує зловмисний код, експлойти та не підтримує нелегальну активність.

2. Орієнтований на легальне, етичне використання програмних інструментів.

3. У відповідях можуть бути неактуальні рекомендації, тому вкрай бажано завжди перевіряти офіціальну документацію.

4. Може помилитися в складних сценаріях, особливо в нестандартних збірках Kali Linux або складних користувальницьких конфігураціях.

5. Погано справляється з задачами, які потребують глибокого ручного налаштування чи прямого доступу до системи користувача.

Все це є особливістю будь-якої мовної моделі, яка вчить, пояснює, але не замінює практику.

Щоб почати використовувати KaliGPT, треба перейти на його офіційну сторінку [5], увійти в аккаунт або спочатку зареєструватись. Далі можна починати діалог: відразу писати питання, просити пояснень, обговорювати реальні задачі. При необхідності можна уточнювати деталі, бо чим детальніше формулюється запит, тим кориснішою буде відповідь.

Інтерфейс є максимально простим: все виконується в звичному форматі чату, питання та відповіді поруч.

Користувач може використовувати такі ресурси та сервіси для практики: Hack The Box [6] – симуляція пентесту, тренувальні віртуальні машини; TryHackMe [7] – інтерактивне навчання з покроковими завданнями; GitHub [8] – множина готових скриптів, програмних інструментів та фреймворків; HackerOne [9] – платформа для пошуку вразливостей та баг баунті; Exploit Database [10] – база даних публічних експлойтів; офіційна документація Kali Linux [11].

В часи, коли знання кібербезпеки стало необхідним майже для всіх IT-спеціалістів, поява такого помічника, як KaliGPT, – суттєвий крок вперед. Він зберігає час, прискорює навчання, допомагає структурувати знання і знижує поріг входу в складну, динамічну, лякаючу, але вкрай цікаву область інформаційної безпеки. Очевидно при цьому, що персональний професійний досвід вкрай потрібен. Але KaliGPT – універсальний радник та помічник, який не буде сміятися над недолугим питанням, не втомить пустими фразами і не пошле «гуглити». В даному випадку ШІ працює на користувача і для нього, а не навпаки.

KaliGPT може бути корисним і для тих, хто ознайомлюється з основами кібергігієни на рівні бакалавра (зазвичай 1-й, 2-й, 3-й курс університетів), і для тих, хто вивчає кібербезпеку на старших курсах бакалаврату або в магістратурі.

Література

[1] Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.kali.org/

[2] Stack Overflow. [Електронний ресурс]. – Режим доступу до ресурсу: https://stackoverflow.com/questions

[3] Metasploit | Penetration Testing Software, Pen Testing Security. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.metasploit.com

[4] Nmap: the Network Mapper - Free Security Scanner. [Електронний ресурс]. – Режим доступу до ресурсу: https://nmap.org/

[5] ChatGPT - KaliGPT. [Електронний ресурс]. – Режим доступу до ресурсу: https://chatgpt.com/g/g-xouSQobsE-KaliGPT

[6] Hack The Box. Cyber Mastery: Community Inspired. Enterprise Trusted. [Електронний ресурс]. – Режим доступу до ресурсу: https://hs.hackthebox.com/

[7] TryHackMe. Anyone can learn cyber security with TryHackMe. [Електронний ресурс]. – Режим доступу до ресурсу: https://tryhackme.com/

[8] GitHub · Change is constant. GitHub keeps you ahead. [Електронний ресурс]. – Режим доступу до ресурсу: https://github.com/

[9] Hacker One | Global leader in offensive security | Security for AI | Crowdsourced Security. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.hackerone.com/

[10] Exploit Database - Exploits for Penetration Testers, Researchers, and Ethical Hackers. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.exploit-db.com/

[11] Kali Docs | Kali Linux Documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://www.kali.org/docs/

Секція 2

БЕЗКОШТОВНІ СЕРВІСИ, ФРЕЙМВОРКИ, СЕРЕДОВИЩА ТА ІНСТРУМЕНТИ ДЛЯ РОЗРОБНИКІВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ANALYSIS OF OPEN SOURCE FRAMEWORKS FOR DEPLOYING LARGE LANGUAGE MODELS ON EDGE NODES

Orel R.L.
Supervisor: Rozlomii I.O.
E-mail: r.l.orel.asp25@chdtu.edu.ua
Cherkasy, Cherkasy state technological university

The rapid evolution of Large Language Models (LLMs), such as GPT-4 and Llama 3, has traditionally relied on high-performance cloud infrastructure due to the immense computational resources required. However, issues related to latency, network reliability, bandwidth costs, and data privacy have catalyzed a shift towards "Edge AI" is deploying models directly on resource-constrained devices like Single Board Computers (SBCs) and mobile SoCs. This transition presents a fundamental architectural challenge: the "memory wall", where consumer-grade memory bandwidth significantly limits token generation speeds.

Running a standard 7-billion parameter model in half-precision (FP16) requires approximately 14 GB of RAM, exceeding the capacity of most edge devices. Furthermore, the discrepancy between the compute-bound "prefill" phase and the memory-bound "decode" phase of LLM inference necessitates highly optimized software stacks capable of aggressive model compression without catastrophic accuracy loss.

There are three primary open-source ecosystems facilitating LLM deployment on the edge:

Llama.cpp and the GGUF Standard. Llama.cpp has emerged as the most versatile runtime, favoring raw C/C++ implementation over complex dependencies. Its key innovation is the GGUF file format, which supports memory mapping (mmap), allowing the operating system to dynamically manage memory pressure. The framework utilizes "K-Quants" (block-wise quantization), enabling a 7B model to be compressed to 4-bit precision (Q4_K_M) with less than 1% perplexity degradation. On CPU-centric devices like the Raspberry Pi 5, Llama.cpp's hand-written NEON kernels provide optimal performance, achieving 2–4 tokens per second [1].

MLC LLM (Machine Learning Compilation). Unlike the runtime approach of Llama.cpp, MLC LLM leverages the Apache TVM compiler stack to treat neural networks as computational graphs compiled into machine code. By utilizing the Vulkan API, MLC LLM can deploy models across a wide range of heterogeneous GPUs (including integrated graphics), often achieving 2–3x faster decoding speeds than Llama.cpp on platforms like NVIDIA Jetson, though sometimes lagging in prefill latency due to static graph constraints [2].

ExecuTorch. Represents the next generation of PyTorch for mobile devices. It employs Ahead-Of-Time (AOT) compilation and native delegates for Neural Processing Units (NPUs), such as the Qualcomm Hexagon. Benchmarks indicate that ExecuTorch significantly outperforms CPU-based inference on mobile devices by offloading operations to the NPU, offering superior energy efficiency [3].

There is no "one-size-fits-all" solution for edge LLMs. Llama.cpp serves as a universal tool for CPU-based inference and rapid prototyping due to its portability. Conversely, compiler-based approaches like MLC LLM and ExecuTorch are essential for maximizing throughput on specialized hardware (GPUs and NPUs). The viability of edge AI ultimately depends on the synergy between these open-source software optimizations and hardware-aware quantization techniques.

References

[1] Gerganov, G. (2023). Llama.cpp: Port of Facebook's LLaMA model in C/C++. GitHub

[2] Chen, T. et al. (2018). TVM: An Automated End-to-End Optimizing Compiler for Deep Learning. OSDI

[3] Meta AI. (2023). ExecuTorch: Enabling on-device AI across mobile and edge

THE HUGGING FACE PLATFORM AS AN ENVIRONMENT FOR DEVELOPING AND TRAINING ARTIFICIAL INTELLIGENCE MODELS

Blyndaruk A., Dolgova N.
E-mail: natalya.dolgova@hneu.net
Kharkiv, Simon Kuznets Kharkiv National University of Economics

The fast development of artificial intelligence technologies has led to the appearance of universal platforms that support the development, training, and integration of machine learning models. One of the most widely popular open-source platforms is Hugging Face, which provides a broad set of tools for working with large language models, natural language processing, and the creation of intelligent applications.

The growing volume of data and the need to automate information processing contribute to the widespread adoption of artificial intelligence models. At the same time, ensuring the accessibility of development tools is important, as it allows researchers and developers to create intelligent systems without significant financial costs. The Hugging Face platform meets these requirements due to its open architecture, extensive model library, and support for integration with various software environments.

Let us focus on the functionality of the Hugging Face platform and its role in developing intelligent systems based on large language models. The main components of the Hugging Face platform include models, datasets, spaces, and the community (Fig. 1).

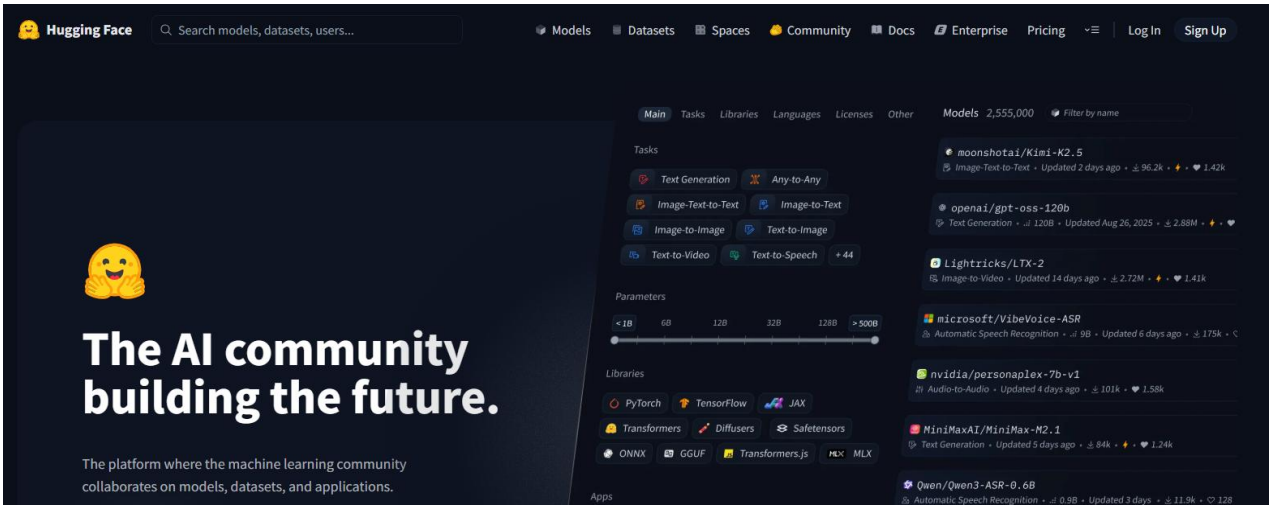


Figure 1. Structure of the Hugging Face platform main page

The Hugging Face model library provides access to a large number of pre-trained models for natural language processing, computer vision, and audio analysis. It supports popular model architectures, including BERT, GPT, T5, LLaMA, and others. Hugging Face Datasets provides access to open datasets and tools for their processing, which significantly simplifies the preparation of training samples required for training neural networks of various types.

Hugging Face Spaces (Fig. 2) is a cloud service of the platform designed for deployment, demonstration, and testing of artificial intelligence applications. It provides developers, researchers, and students with the ability to create interactive web interfaces for machine learning models without the need for complex server infrastructure configuration. As part of the ecosystem, Spaces facilitate rapid prototyping of AI applications. The platform ensures integration with machine learning libraries and supports publishing ready-made applications in open access. Among the interactive AI-based web applications available on Hugging Face Spaces, there are numerous chatbots, text analysis systems, computer vision models, audio processing systems, analytical decision-support systems, etc.

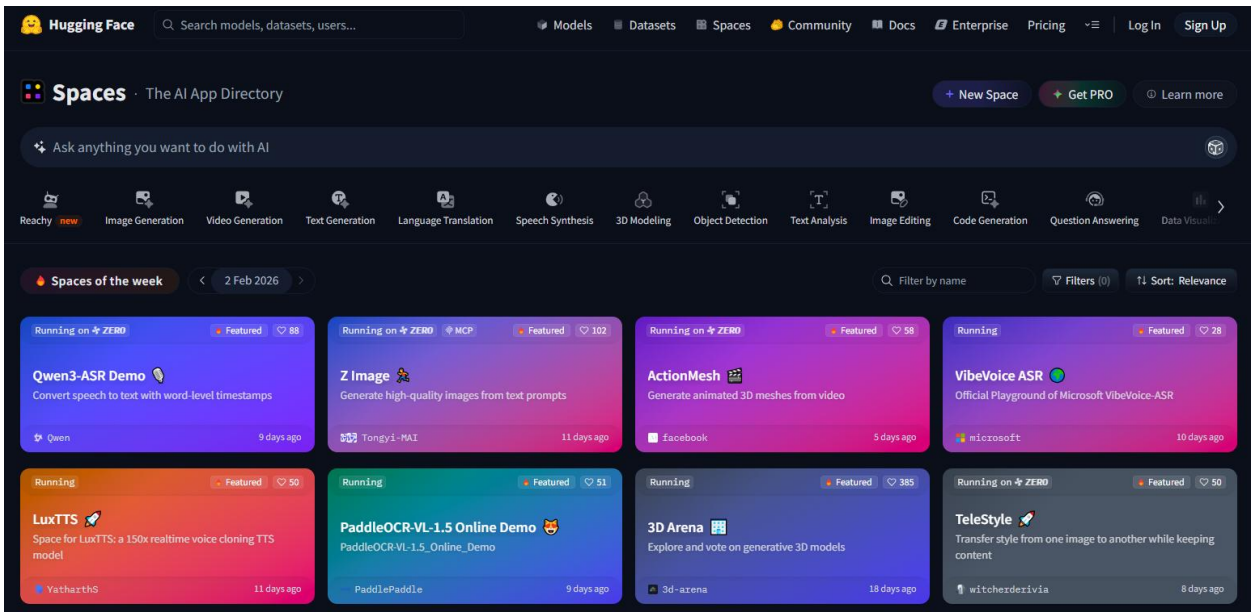


Figure 2. General view of the Hugging Face Spaces page

The platform supports several development environments, including Gradio for web applications, Streamlit for creating analytical web dashboards, and Docker for complex customized development. Hugging Face tools are actively used in education and scientific research, financial analytics, cybersecurity, document processing automation, intelligent information retrieval systems, and more.

One example of using the platform could be developing a cyber incident analysis system. Using natural language processing models is advisable for automatic analysis of information security event logs. Using the Transformers library, cyber incident classification can be implemented using the following code fragment:

```
from transformers import pipeline
classifier = pipeline( "text-classification",
    model="distilbert-base-uncased-finetuned-sst-2-english")
log_event = "Multiple failed login attempts detected from external IP"
result = classifier(log_event)
print(result)
```

This type of model allows classifying security events and identify potential threats. The obtained results can be used in security monitoring systems and incident response centers.

The Hugging Face platform also can be helpful for developing decision support systems (DSS) in cybersecurity by integrating artificial intelligence models with knowledge bases and analytical modules. The main application areas include cyber risk analysis, support of SOC center operations, generation of incident response recommendations, automation of regulatory document analysis, and cyber threat forecasting. Integration of models with RAG technology enables the use of knowledge bases, cybersecurity standards, and analytical reports to generate well-grounded managerial decisions.

The Hugging Face platform is a universal environment for developing artificial intelligence systems, combining tools for working with language models, datasets, and interactive applications. The use of open language models and data analysis tools allows automation of information processing and improves the efficiency of cyber threat response. Further research may focus on the development of multi-agent cybersecurity systems and integration of artificial intelligence models into cyber resilience systems of critical infrastructure.

CREATING GPT AGENTS USING OPEN-SOURCE SOFTWARE

Shapovalova O.O., Solodovnyk H.V.
E-mail: olena.shapovalova@hneu.net, ganna.solodovnyk@hneu.net
Kharkiv, Simon Kuznets Kharkiv National University of Economics

The development of generative artificial intelligence has led to the emergence of intelligent software agents capable of automating data processing, supporting decision-making, and interacting with users. This paper examines the methodology for creating GPT agents using free and open-source tools that ensure accessibility of artificial intelligence technologies for educational, scientific, and applied tasks.

The active implementation of large language models in various fields stimulates the development of accessible tools for designing intelligent agents capable of operating with limited amounts of information and providing the most accurate responses within the available dataset. The use of FOSS solutions reduces financial costs of AI implementation, enables adaptation of models to specific tasks, and promotes the development of open scientific research.

The paper analyzes the possibilities of creating GPT agents using free software platforms and proposes a generalized architecture for their implementation. The results of the review of FOSS resources for creating GPT agents are presented in Table 1.

Table 1 – Resources for Creating and Operating GPT Agents

Large Language Models	Agent Development Frameworks	Free Development Environments	Data Processing Tools	Vector Databases
LLaMA (Meta) – a family of models with local deployment capability	LangChain – a library for creating agent systems using LLM	Google Colab (Free Tier) – cloud environment for model training and testing	Hugging Face Transformers – a library for working with language models	Provide efficient information retrieval and implementation of RAG (Retrieval-Augmented Generation)
Mistral / Mixtral – efficient models with an open license	LlamaIndex – a tool for integrating models with knowledge bases	Jupyter Notebook – interactive environment for algorithm development	Datasets (Hugging Face) – collection of open datasets	FAISS
Falcon LLM – high-performance model for research tasks	Haystack – platform for building search and conversational systems	VS Code + Open Source Extensions – universal programming environment	spaCy – natural language processing tool	Weaviate (Open Source)
GPT4All – local models for personal agents	AutoGen – framework for multi-agent systems			ChromaDB

GPT agents can be created in two ways:

– through programming using open-source libraries;

– through ready-made interfaces (low-code / no-code platforms).

Both options are available in the free FOSS environment.

In the case of the classical FOSS approach, i.e., creating an agent using coding, a program is developed that implements agent logic (usually Python) using specialized open-source framework libraries (LangChain, LlamaIndex, GPT4All SDK, Transformers, FAISS / ChromaDB, AutoGen). After that, the model (Table 1) is connected, a knowledge base is added (Table 1), and the resulting system enables the agent to operate.

When choosing a no-code approach, it is advisable to use specialized platforms with user-friendly interfaces such as Flowise, LibreChat, Open WebUI, and Haystack UI (Table 2).

Table 2 – Low-Code / No-Code Platforms for Agent Development

Name	Flowise	LibreChat	Open WebUI	Haystack UI
Purpose	Visual agent builder	Open-source ChatGPT alternative	Interface for local LLMs	Platform for building search, analytical, DSS systems based on LLM
Compatibility	GPT4All, Ollama, Falcon, Llama	OpenAI, Ollama, Azure OpenAI, HuggingFace, Local LLM	Ollama, GPT4All, Local LLM, HuggingFace models	Open-source LLM, OpenAI, HuggingFace, Ollama, Falcon, Mistral
Capabilities	RAG without programming, local model connection, document integration, chat interface, API generation	Web interface for LLM, local model support, multi-agent mode, document integration	Simple interface, work with local LLMs, RAG support, minimal configuration	Powerful RAG pipelines, flexible orchestration logic, enterprise-level solutions, knowledge graph support
Application Areas	Education, scientific research, SOC agent prototyping	Corporate AI assistants, educational platforms, knowledge assistants, internal chat systems, research experiments	Local AI assistants, laboratory research, private corporate systems, offline cybersecurity systems, model testing	Corporate systems, DSS systems, threat analytics, intelligent document analysis, search systems, knowledge assistants

The typical architecture of a GPT agent consists of a user interaction interface, a query processing module, a language model, a knowledge base, and an integration module for external services. When creating an agent for monitoring and ensuring the cybersecurity of critical infrastructure facilities, it is advisable to use a combined approach, where the prototype is created using low-code / no-code platforms with interfaces and subsequently developed into an industrial system through coding.

For example, a prototype SOC agent can be created using Flowise, which allows integration of MITRE ATT&CK, adding incident logs, and testing agent logic. At the coding or Python implementation stage, integration with Wazuh, a free open-source cybersecurity platform combining SIEM (Security Information and Event Management) and XDR (Extended Detection and Response) capabilities, is advisable. Coding also enables automation of incident response and development of DSS systems.

The use of free and open-source software enables the creation of effective GPT agents without significant financial costs. The combination of open language models, agent frameworks, and vector databases ensures the development of intelligent systems for a wide range of applied tasks. Further research may focus on improving agent autonomy and optimizing their performance.

OPEN-SOURCE ENVIRONMENTS FOR EXPERIMENTAL STUDY OF TASK SCHEDULING IN HETEROGENEOUS DISTRIBUTED SYSTEMS

Yenhalychev S.O., Leunenko O.V.
E-mail: engalichev.sergiy@hneu.net, oleksii.leunenko@hneu.net
Kharkiv, Simon Kuznets Kharkiv National University of Economics

Abstract

The scheduling of tasks remains a core challenge within heterogeneous distributed systems, particularly as computational resources diverge in both performance and availability. As infrastructure evolves across cloud, edge, and hybrid environments, scheduling algorithms increasingly face challenges related to uncertainty caused by incomplete and volatile metadata concerning resource dependencies and execution timelines. Because traditional deterministic models often falter under these conditions, there is a critical need for frameworks that can account for inherent system uncertainty.

We explore how open-source ecosystems can transform the way researchers analyze task scheduling when metadata is uncertain. The approach uses open and well-documented frameworks to ensure that experiments can be reproduced and extended as the system scales. Our methodology brings together three powerful elements open-source simulation, containerized execution environments, and comprehensive monitoring into a single cohesive framework. Think of it as a controlled sandbox where you can rigorously test different scheduling strategies, no matter how much variability you throw at them.

In addition, this study illustrates the formal incorporation of uncertainty into experimental design via the use of probabilistic task modeling and the application of dynamic workload injections. By analyzing key performance metrics, including system throughput, resource utilization, and scheduling consistency, the proposed approach establishes a clear connection between theoretical frameworks and the practical operation of distributed systems. Our results indicate that open-source platforms provide the crucial transparency and flexibility required to promote the advancement of adaptive and intelligent scheduling solutions.

Keywords: task scheduling; heterogeneous distributed systems; metadata uncertainty; open-source environments; experimental evaluation; cloud and edge computing

Introduction

Today’s information systems that we all use are growing faster than ever before, leading to important reliance on connected computers that are spread across cloud data centers, nearby locations, and unique hardware. The problem is obvious: if we don’t manage tasks well, these complex setups won’t be able to achieve what they are supposed to do. Decisions regarding scheduling need to consider variations in processing power, network conditions, and the nature of workloads.

In practice, task scheduling is rarely performed with complete and precise information. Task metadata, such as execution time, memory usage, input data size, and communication requirements, may be unknown, partially available, or subject to change during execution. This phenomenon, referred to as metadata uncertainty, is especially pronounced in dynamic and large-scale distributed systems. As a result, scheduling methods must operate under uncertainty and adapt to evolving system states.

Theoretical models for task scheduling under uncertainty are only as valuable as their real-world performance. Without experimental evaluation in authentic environments, we cannot confidently assess which Open-source software allows anyone to use the tools they need for trying out ideas, simulating situations, and creating prototypes to see if their theories really hold up in real life.

This paper focuses on open-source environments that support the experimental study of task scheduling in heterogeneous distributed systems under metadata uncertainty. The goal is to identify suitable tools, propose an experimental workflow, and outline evaluation approaches that can be used by researchers and software developers.

Background and Related Work

Task scheduling in distributed systems has been extensively studied, and classical approaches typically assume deterministic task parameters and stable execution environments [3, 9]. Nevertheless, the assumptions underlying these systems are often compromised in real-world scenarios due to factors such as fluctuating workloads and a lack of complete information.

Recent research has explored probabilistic and adaptive scheduling methods that explicitly account for uncertainty in task metadata [6, 7]. In parallel, the rise of cloud-native technologies and container orchestration platforms has created new opportunities for an experimental type of research using open-source tools [1, 11]. Simulation frameworks enable controlled modeling of distributed systems [2, 5, 8], while container-based environments support realistic execution scenarios [1, 11, 13].

Open-source platforms are widely used in a number of related studies for evaluating scheduling strategies, resource management policies, and system scalability. Their advantages include transparency of implementation, reproducibility of experiments, and the ability to extend or modify system components. Despite these benefits, there is a need for structured experimental methodologies that integrate multiple open-source tools into a coherent environment for studying scheduling under uncertainty.

Problem Statement: Task Scheduling under Metadata Uncertainty

In heterogeneous distributed systems, tasks are characterized by a set of metadata attributes that influence scheduling decisions. These attributes may include expected execution time, required computational resources, data locality constraints, and inter-task dependencies. Metadata uncertainty arises when certain characteristics are either unrecognized, estimated imprecisely, or subject to fluctuation throughout the process. The difficulty of planning in light of this uncertainty can be seen as the task of assigning resources to specific tasks in a manner that optimizes particular performance objectives, while also taking into account data that is either incomplete or based on probabilities. If not managed effectively, such uncertainty may result in less-than-ideal scheduling choices, longer execution durations, or ineffective use of resources.

From an experimental perspective, it is necessary to model uncertainty explicitly and to evaluate scheduling methods under different uncertainty scenarios. This requires environments that support flexible task descriptions, dynamic system behavior, and detailed monitoring of execution outcomes.

Open-Source Experimental Environments

Open-source environments for studying task scheduling typically consist of three main components: modeling and simulation tools, execution environments, and monitoring frameworks [2, 5, 8, 10]. Simulation tools allow researchers to define heterogeneous system configurations and to generate workloads with controlled uncertainty [2, 5, 8, 10]. Execution environments, such as container-based platforms, enable the deployment of scheduling logic and task execution in realistic conditions [1, 11]. Monitoring tools provide visibility into system behavior and performance metrics [4, 12].

In practice, open-source solutions make it possible for experimental environments to move beyond rigid frameworks and adopt more flexible structures. This flexibility is reflected in the way tools can be adapted, integrated across platforms, and configured to meet specific experimental requirements. As a result, developers obtain finer control over uncertainty modeling, scheduling execution, and data collection, facilitating the construction of experimental environments that are suitable for practical research scenarios.

Such environments also facilitate collaboration and reproducibility, as experimental setups can be shared and extended by other researchers. This aligns with the principles of open science and accelerates the development of robust scheduling methods.

Recommended Open-Source Tooling Stack. To make experimentation reproducible and easy to extend, researchers can assemble a “tooling stack” that covers (i) modeling, (ii) execution, and (iii) observability. Table 1 summarizes commonly used open-source options.

Table 1 – Compact open-source stack for experimental studies of scheduling in heterogeneous distributed systems

Layer	Candidate open-source solutions
Simulation / modeling	CloudSim Plus [8]; iFogSim [5]; EdgeCloudSim [10]; SimGrid [2]
Execution / orchestration	Docker [11]; Kubernetes [1]
Monitoring / dashboards	Prometheus [12]; Grafana [4]

Experimental Workflow and Evaluation Methodology

The outlined experimental process starts by establishing a model of a heterogeneous system that encompasses the characteristics of resources and assumptions about the network. Tasks are created with uncertain metadata, which is depicted through probabilistic distributions or adjustments that occur while they are being executed. Following this, scheduling techniques are utilized to allocate tasks to available resources.

As tasks are executed, monitoring instruments gather performance data that indicate the behavior at both the system level and in relation to scheduling. The evaluation primarily concentrates on metrics including the time taken to complete tasks, overall system throughput, utilization of resources, and the reliability of scheduling choices amid different levels of uncertainty.

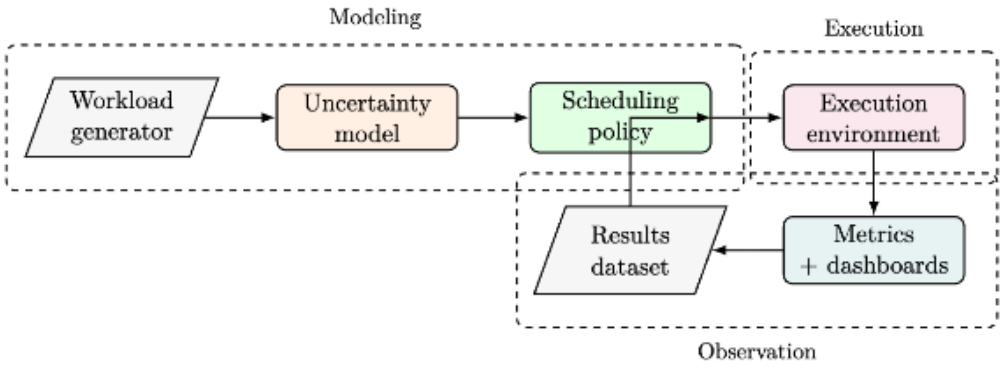


Figure 1: Reference open-source experimental pipeline (with feedback loop) for studying scheduling under metadata uncertainty.

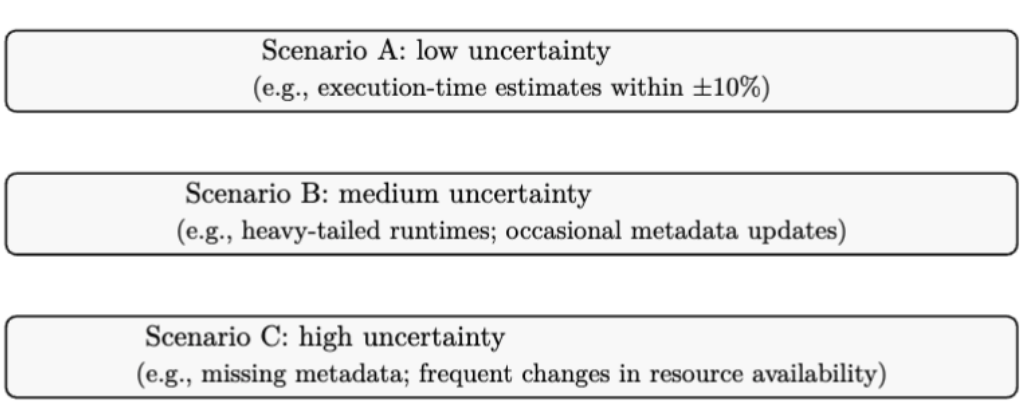


Figure 2. Example uncertainty profiles for structuring experimental campaigns.

Researchers have the ability to compare scheduling techniques and examine their responsiveness to insufficient data by conducting experiments in various configurations and under different uncertainty scenarios. Utilizing open-source tools facilitates the replication and expansion of these experiments.

Discussion

Task scheduling under metadata uncertainty requires flexible, transparent research environments. Open-source software enables faster development and testing of scheduling methods, easing challenges for researchers and developers alike.

At the same time, challenges remain in accurately modeling uncertainty and in scaling experiments to large and highly dynamic systems. Combining simulation-based approaches with containerized execution environments offers a promising direction for addressing these challenges.

Conclusions and Future Work

This paper has examined open-source environments for the experimental study of task scheduling in heterogeneous distributed systems under metadata uncertainty. A structured approach to integrating simulation, execution, and monitoring tools has been presented, along with evaluation considerations for uncertain environments. Upcoming research will aim to enhance the experimental framework to accommodate adaptive and intelligent scheduling techniques, particularly those utilizing machine learning.

References

[1] Brendan Burns, Joe Beda, and Kelsey Hightower. Kubernetes: Up and Running. O'Reilly Media, 3 edition, 2024.

[2] Henri Casanova, Arnaud Giersch, Arnaud Legrand, Martin Quinson, and Frédéric Suter. Lowering entry barriers to developing custom simulators of distributed applications and platforms with simgrid. Parallel Computing, 123:103125, 2025.

[3] Mahendra Bhatu Gawali and Subhash K. Shinde. Task scheduling and resource allocation in cloud computing using a heuristic approach. Journal of Cloud Computing, 7(4), 2018.

[4] Grafana Labs. Grafana documentation. Project documentation, 2025. Accessed 2026-02-01

[5] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K. Ghosh, and Rajkumar Buyya. ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. Software: Practice and Experience, 47(9):1275–1296, 2017.

[6] Khaled Matrouk, Khaleel Alatoun, Ali Sfar, Ahmad Khatib, and Ahmad Fayad. Scheduling algorithms in fog computing: A survey. International Journal of Networked and Distributed Computing, 9(1):1–22, 2021.

[7] Subham Kumar Sahoo and Sambit Kumar Mishra. A survey on task scheduling in edge-cloud. SN Computer Science, 2025.

[8] Manoel C. Silva Filho, Raysa L. Oliveira, Claudio C. Monteiro, Pedro R. M. Inácio, and Mário M. Freire. Cloudsim plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In 2017 IFIP/IEEE International Symposium on Integrated Network and Service Management (IM), pages 400–406, 2017.

[9] Shailender Singh, Inderveer Chana, and Rajkumar Buyya. A survey on scheduling techniques

in cloud computing: Issues and challenges. Journal of Network and Computer Applications, 134:40–65, 2019.

[10] Caglar Sonmez, Atay Ozgovde, and Cem Ersoy. Edgecloudsim: An environment for performance evaluation of edge computing systems. Transactions on Emerging Telecommunications Technologies, 29(11):e3493, 2018.

[11] The Docker Authors. Docker documentation. Project documentation, 2025. Accessed 2026-02-01.

[12] The Prometheus Authors. Prometheus documentation. Project documentation, 2025. Accessed

2026-02-01.

[13] Lavish Varshney and Yogesh Simmhan. Characterizing application scheduling on edge, fog and cloud computing resources. Future Generation Computer Systems, 99:228–244, 2019.

АРХІТЕКТУРА ВІДКРИТОЇ ПЛАТФОРМИ ДИНАМІЧНОЇ МАРШРУТИЗАЦІЇ ВЕБПОСИЛАНЬ ДЛЯ ЦИФРОВОЇ ІДЕНТИФІКАЦІЇ РЕСУРСІВ

Алексієв В.О.
E-mail: vlax@hneu.edu.ua
Харків, Харківський національний економічний університет імені Семена Кузнеця

Сучасний вебсайт компанії є ефективним інструментом бізнесу. Веб-ресурс або онлайн сервіс – це зручний актив компанії, який надає можливості зосередитись на рішенні поточних завдань та перенести до інформаційного середовища рішення ключових запитів та реалізацію бізнес процесів. Відповідно до концепції Індустрії 4.0 такий онлайн інструмент стає платформою для побудови цифрового двійника компанії. У свою чергу, значною мірою зростає складність побудови визначених цифрових активів комерційних компаній. Оскільки, зараз переважно у архітектурі веб-рішення застосовується патерн мікросервісів. Це певного часу, може створити виклик до формулювання завдання розробки, наприклад, з побудови зручного API для комунікації усіх складових онлайн проєктів та здійснення підтримки актуального стану URL-посилань на всі сторінки сайту.

Таким чином, коли вебсайт стає комунікаційною платформою для запуску мікросервісів, як його складових, а також розвиток самого сайту призводить до додавання та видалення безлічі сторінок та API-інтерфейсів й актуальним стає завдання побудови архітектури відкритої платформи динамічної маршрутизації вебпосилань для цифрової ідентифікації ресурсів. Фактично, для рішення цього завдання треба ефективно управляти адресацією сторінок на веб-ресурсі. Це можна досягти деяким способами. Один з них – побудова системи всередині вебсайту, де виконуються завдання переадресації сторінок. Це ефективно для SEO-міркувань та взагалі, для зручності роботи користувача.

Можна запропонувати застосувати ієрархічну структуру, де буде залучено трирівнева система перенаправлення ресурсів. На верхньому рівні доцільним є застосування DNS маршрутизації. Найчастіше – це DNS-записи (CNAME, ALIAS), які забезпечують інфраструктурну абстракцію доменів. Такий механізм має дуже високі вимоги щодо швидкості набуття змін, однак, на глобальному рівні вебсайту такий підхід є обов'язковим, звичайно, у поєднанні з іншими рівнями внутрішньої структури платформи динамічної маршрутизації вебпосилань [1, 2].

На другому рівні можна відзначити залучення засобів перенаправлення (redirect) як механізму реалізованого на базі протоколу HTTP, коли сервер повідомляє браузеру або пошуковому роботу, що запитуваний ресурс доступний за іншою адресою. Для цього слід налаштувати веб-сервер, який обслуговує сайт. Наприклад, веб-сервер Apache дозволяє робити перенаправлення через конфігурування .htaccess (локально для каталогу), через основний конфігураційний-файл (httpd.conf, apache2.conf) або VirtualHost. Веб-сервер Nginx слід налаштувати у конфігураційному файлі: /etc/nginx/sites-available/. Найпоширена практикою є застосування станів (відповіді сервера) протоколу HTTP: 301 Redirect – Moved Permanently (означає, що ресурс остаточно перенесений на нову адресу) та 302 Redirect – Found / Temporary Redirect (означає, що ресурс тимчасово переміщено).

Слід виділити третій рівень – застосунку, який виконується вже на рівні CMS (Content Management System), наприклад, WordPress. На цьому рівні користувач/адміністратор сайту отримує продуктивні та зручні інструменти налаштування HTTP-перенаправлень у зрозумілому веб-інтерфейсі. Наприклад, плагіни:

– Redirection (https://wordpress.org/plugins/redirection/),
– Simple 301 Redirects (https://wordpress.org/plugins/simple-301-redirects/),
– Easy HTTPS Redirection (SSL) (https://wordpress.org/plugins/https-redirection/),
– WP Post Redirect (https://wordpress.org/plugins/wp-post-redirect/),
– Safe Redirect Manager (https://wordpress.org/plugins/safe-redirect-manager/) та ін.

Розглянута архітектура пропонується як внутрішнє рішення на рівні вебсайту, однак слід враховувати й можливість та ефективність залучення зовнішнього сервісу, що буде функціонувати на базі окремого VPS (Virtual Private Server) або ін. Пропонується впровадження сервісу упорядкування роботи з посиланнями, які надаються на веб-ресурси компанії. У рамках цього сервісу можлива реалізація створення QR-кодів (quick-response code) для публікації на сайтах та офлайн розміщення. Впровадження веб-сервісу скорочення посилань дозволить ефективно керувати посиланнями на ресурси та послуги компанії, які надаються. Створене посилання можна опублікувати в соціальних мережах або надсилати у електронних листах, наприклад, протягом проведення маркетингових компаній або у спілкуванні з клієнтами. На цій основі можливим стає відстеження наявності переходів по посиланню та базове уявлення про інтерес до певного напряму або послуги. Звичайно, слід враховувати потенційну небезпеку компрометації QR-кодів.

Для реалізації цього проєкту можна застосувати рішення на основі комерційного веб-сервісу. Наприклад, одним з найвідоміших комерційних сервісів зі скорочення URL-адрес є Bitly (https://bitly.com/). Цей сервіс має безкоштовний план. Однак, ексклюзивне розміщення сервісу на ресурсах компанії дає можливість повністю керувати відповідним сервісом та даними, які на ньому оброблюються.

Для впровадження сервісу скорочення посилань зараз є доцільним використання рішень, які засновані на вільному програмному забезпеченні, наприклад:

– Snapp (https://snapp.li/), який має сучасний дизайн та інтерфейс системи, наявність аналітики, має розгортання у контейнері Docker;

– Shlink (https://shlink.io/) з динамічним інтерфейсом та підтримкою для управління декількома доменами, має інструменти статистичного аналізу, API взаємодії з іншими сервісами, розгортається у контейнері Docker та забезпечує генерацію QR-кодів;

– YOURLS (https://yourls.org/) є достатньо сталим проєктом, який розвивається з 2009 року. Інтерфейс системи можна кастомізувати завдяки зміни теми веб-інтерфейсу. Має численні плагіни, які розширюють функціонал системи;

– Kutt (https://kutt.it/) має легкий сучасний інтерфейс, має розширення для популярних браузерів. Побудований на основі сервера з Node.js, має розгортання у контейнері.

У результаті проведеного аналізу веб-ресурсів та демонстраційних сайтів (у разі їх наявності) виконано вибір системи для тестового впровадження сервісу скорочення URL-адрес. Доцільним, стабільним та тим, що має розширення можливостей на базі плагінів є YOURLS. Недоліком цього рішення є порівняно застарілий інтерфейс системи, однак його можна змінити сторонньою темою, а також цей інтерфейс системи буде доступний тільки співробітникам компанії які будуть його обслуговувати. Користувачі ресурсу отримують посилання та QR-код у вигляді графічного файлу та зручний сервіс перенаправлення.

YOURLS має повністю безкоштовною ліцензією на використання (розповсюджується за ліцензією MIT), проєкт поєднав велику спільноту розробників і ентузіастів, які використовують цей продукт. Перевага сервісу полягає в тому, що базова функціональність дозволяє скорочувати посилання, редагувати їх, а завдяки багатьом плагінам на вибір можна реалізувати потрібний для визначеного рішення функціонал створення QR-кодів та багато ін.

References

[1] LaCroix, J. Mastering Ubuntu Server / Jay LaCroix. – Birmingham : Packt Publishing, 2022. – 4-те вид. – 584 с.

[2] Ouiran, G. NGINX HTTP Server: Harness the power of NGINX with a series of detailed tutorials and real-life examples / Gabriel Ouiran. – Birmingham : Packt Publishing, 2024. – 5-те вид. – 348 с.

[3] 4 reasons I host my own URL shortener – XDA [Electronic resource]. – Resource access mode: https://www.xda-developers.com/reasons-host-own-url-shortener/

[4] How to Install YOURLS on Ubuntu with Nginx and Let’s Encrypt – Linux Stans [Electronic resource]. – Resource access mode: https://linuxstans.com/how-to-install-yourls/

ВИКОРИСТАННЯ СИСТЕМ АВТОМАТИЧНОЇ ПЕРЕВІРКИ КОДУ ДЛЯ РОЗВИТКУ НАВИЧОК АЛГОРИТМІЗАЦІЇ ЗДОБУВАЧІВ ОСВІТИ

Березенська С. М.
E-mail: berezsvet@gmail.com
Харків, Харківський радіотехнічний фаховий коледж

У сучасній ІТ-освіті ми все більше стикаємося з проблемою розвитку практичних навичок програмування, яка полягає в тому, що здобувачі освіти формують програмні рішення без усвідомлення загальної задачі, без врахування критичних значень вхідних даних, без визначення області видимості змінних та констант в межах програмних блоків тощо. Одну з причин цієї проблеми ми бачимо у відсутності або у спрощенні методичних підходів до формування алгоритмічного мислення здобувачів освіти [1].

Однією з ефективних технологій в питаннях розвитку алгоритмічного мислення можуть виступати системи автоматичної перевірки коду, більшість з яких базуються на принципах Free and Open Source Software або мають відкриті освітні можливості. Такі системи забезпечують оперативний зворотний зв'язок, сприяють самостійному навчанню та дозволяють масштабувати процес оцінювання. Відкритість таких рішень забезпечує можливість їх інтеграції в освітні середовища закладів освіти, а також гнучкість та прозорість алгоритмів оцінювання. Крім того, застосування відкритих платформ в освітньому процесі створює середовище, наближене до реальних умов роботи розробника – здобувачі освіти засвоюють принципи тестування, стандарти оформлення коду та алгоритмічну культуру.

Серед платформ, які найчастіше застосовуються для автоматичної перевірки коду: DOMjudge – відкрита система для проведення змагань з програмування та автоматичного тестування розв'язків; ejudge – система автоматичного тестування, що широко використовується у навчальних олімпіадах; Mooshak – вебсистема автоматичної перевірки рішень для навчальних курсів і конкурсів; Jutge.org – освітнє середовище з відкритими елементами автоматизованого оцінювання та інші. Ці системи мають зрозумілий інтерфейс, є інтерактивним, дозволяють проведення змагань з спортивного (олімпіадного) програмування, сумісні з різними мовами програмування, дозволяють багаторазове виконання завдань. А використання рейтингових систем та гейміфікації на цих платформах сприяє підвищенню навчальної мотивації та розвитку навичок самоконтролю [2].

Методичні підходи до використання систем автоматичної перевірки для розвитку алгоритмічного мислення полягають у виконанні індивідуальних алгоритмічних задач, організації навчальних змагань з програмування, а також у забезпеченні формувального оцінювання через поступове ускладнення задач. У процесі навчання алгоритмізації доцільно використовувати задачі, які легко формалізуються та можуть перевірятися автоматично за набором тестів. Прикладами таких задач є логічний ланцюжок від базових алгоритмічних конструкцій (перевірка числа на простоту; обчислення значення функції за заданими умовами; пошук суми цифр числа; генерація таблиць значень функцій тощо) до задач підвищеної складності (аналіз складності алгоритмів; задачі на жадібні алгоритми; базові задачі на графи (пошук у ширину/глибину)). Тобто для розвитку алгоритмічного мислення використовуються задачі різного рівня складності, які перевіряються автоматично за допомогою наборів тестів, що враховують коректність розв'язку, обробку граничних випадків та ефективність алгоритму.

Але розвиток алгоритмічних навичок відбувається не завдяки самому тестуванню, а завдяки циклу алгоритмічного мислення, який запускає автоматична перевірка.

Щоб пройти автоматичні тести, здобувач освіти має:

– виконати декомпозицію задачі: чітко визначити вхідні та вихідні дані, формалізувати умову, виділити кроки алгоритму;

– опрацювати граничні випадки: порожні масиви, мінімальні/максимальні значення, нестандартні входи;

– виконати покрокове (ітераційне) вдосконалення алгоритму: ідея алгоритму → реалізація → тестування → аналіз помилки → модифікація алгоритму, що фактично є алгоритмічним експериментом та розвиває навички оптимізації;

– проаналізувати складність та ефективність алгоритму, адже багато систем обмежують час виконання алгоритму та контролюють пам’ять;

– відлагодити програмний код: проаналізувати логіку алгоритму, перевірити умови циклів, знайти помилки у структурі рішення.

Таким чином, автоматична перевірка коду виконує не лише функцію оцінювання, а й виступає інструментом розвитку алгоритмічного мислення. Завдяки миттєвому зворотному зв’язку здобувачі освіти проходять ітераційний цикл побудови алгоритму: формалізація задачі, реалізація рішення, аналіз результатів тестування та оптимізація алгоритму. Наявність граничних тестів стимулює узагальнення алгоритмів, врахування нестандартних ситуацій та аналіз ефективності рішень, що сприяє формуванню стійких алгоритмічних навичок.

Практика використання автоматичних систем перевірки демонструє зростання рівня самостійності здобувачів освіти, підвищення швидкості засвоєння алгоритмічних конструкцій та покращення якості програмного коду. Значно зменшується навантаження на викладача щодо перевірки великої кількості однотипних завдань, що дозволяє зосередитися на методичному супроводі навчання.

Разом з тим, незважаючи на значні педагогічні переваги, системи автоматичної перевірки коду мають певні обмеження. Зокрема, існує ризик орієнтації здлбувачів освіти на проходження тестів без глибокого розуміння алгоритмів, недостатня оцінка процесу побудови рішення та зниження рівня рефлексії щодо алгоритмічних підходів. Крім того у здобувачів першого року навчання може існувати певний технічний бар’єр застосування систем через нерозуміння форматів введення/виведення або невміння аналізувати системні повідомлення про помилки. Також великим викликом на сьогодні є ризик академічної недоброчесності через копіювання рішень, генерацію кодів за допомогою штучного інтелекту, використання готових алгоритмів без розуміння. Тож ефективність використання таких систем значною мірою залежить від якості тестових наборів і методичного супроводу викладача. Саме тому автоматичну перевірку доцільно поєднувати з обговоренням алгоритмів, поясненням рішень та аналізом альтернативних підходів.

Враховуючи всі переваги та застереження доцільно зробити висновок, що системи автоматичної перевірки коду є ефективним інструментом розвитку алгоритмічного мислення здобувачів освіти та можуть бути успішно інтегровані у курси програмування в рамках концепції відкритого програмного забезпечення. Застосування відкритих платформ дозволяє створити навчальне середовище, наближене до реальної практики розробки програмного забезпечення, забезпечити індивідуалізацію темпу навчання, формувальне оцінювання та розвиток навичок самоконтролю й алгоритмічної рефлексії. Водночас ефективність використання систем значною мірою залежить від методичного супроводу викладача.

Інтеграція відкритих платформ в освітні курси створює передумови для модернізації ІТ-освіти, підвищення об’єктивності оцінювання та формування професійних компетентностей майбутніх фахівців. Подальші дослідження доцільно спрямувати на розробку методичних моделей збалансованого використання автоматичної перевірки, впровадження механізмів запобігання академічній недоброчесності та розширення практик відкритого освітнього програмного забезпечення у професійній підготовці ІТ-фахівців.

Література

[1] Вітковська, І. Крамар, Ю. (2025). Формування алгоритмічного мислення студентів ІТ-спеціальностей // Адаптивні системи автоматичного управління. Том 2. № 47, 77-86.

[2] Волошина, Т., Глазунова, О., Гуржій, А., Пархоменко, О., Корольчук, В. (2020). Платформи та системи автоматизованої перевірки завдань з програмування: аналіз, критерії добору та приклад використання // Електронне наукове фахове видання «Відкрите освітнє е-середовище сучасного університету», (8), 154-164.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ФОРМУВАННЯ ПЕРСОНАЛІЗОВАНИХ РЕКОМЕНДАЦІЙ ТОВАРІВ

Бойченко А.Г., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

Активний розвиток електронної комерції та постійне зростання кількості товарів в онлайн-магазинах призводять до актуалізації задачі автоматизованого формування персоналізованих рекомендацій товарів для користувачів. Значний обсяг інформації, велика кількість категорій і текстових описів ускладнюють процес пошуку релевантних товарів, що зумовлює необхідність використання програмних засобів, здатних аналізувати дані та визначати ступінь подібності між об'єктами. Тому розробка рекомендаційних систем, які базуються на методах аналізу тексту та машинного навчання, є важливою задачею у сфері інформаційних технологій.

У даній роботі розроблено програмне забезпечення для рекомендації товарів на основі їх текстових характеристик та історії взаємодії користувача з системою. Для тестування роботи програми використано набір даних у форматі CSV, що було отримано з відкритого репозиторію GitHub і який містить інформацію про товари, зокрема їх назви, описи, категорії та вартість [1]. Використання реальних відкритих даних дозволило наблизити результати роботи програми до умов практичного застосування.

Основою роботи програми є контентно-орієнтований підхід до формування рекомендацій, доповнений аналізом попередніх виборів користувача. На початковому етапі здійснюється завантаження CSV-файлу та попередня обробка даних: усуваються пропущені значення, числові характеристики приводяться до відповідного формату, а для кожного товару формується сукупний текст шляхом об'єднання його назви та опису. Отриманий текст використовується як основне джерело інформації для подальшого аналізу.

Для перетворення текстових даних у числовий вигляд застосовується метод TF-IDF (Term Frequency – Inverse Document Frequency), у результаті чого кожен товар подається у вигляді вектора ознак. Для визначення ступеня схожості між товарами використовується косинусна подібність, яка дозволяє ефективно визначати семантичну близькість текстових описів незалежно від їхньої довжини.

Алгоритм формування рекомендацій полягає у порівнянні товарів, які були раніше обрані користувачем, з іншими товарами з набору даних. Для кожного кандидата обчислюється значення косинусної подібності, а також враховується перетин категорій товарів, що дозволяє підвищити релевантність рекомендацій. На основі отриманих значень формується підсумковий рейтинг, за яким відбираються товари з найбільшим рівнем подібності.

Програмне забезпечення реалізовано мовою програмування Python, яка є зручною для роботи з даними та реалізації алгоритмів аналізу тексту. У процесі розробки використано бібліотеку pandas для зчитування та обробки даних у форматі CSV [2], а також бібліотеку sklearn для реалізації TF-IDF векторизації та обчислення косинусної подібності [3]. Розроблена програма демонструє практичне застосування методів аналізу текстових даних і може бути використана як основа для подальшого розвитку рекомендаційних систем у сфері електронної комерції.

Література

[1] Amazon dataset samples [Electronic resource]. – Access mode: https://github.com/luminati-io/Amazon-dataset-samples

[2] Pandas documentation [Electronic resource]. – Access mode: https://pandas.pydata.org/docs/

[3] sklearn API [Electronic resource]. – Access mode: https://scikit-learn.org/stable/modules/classes.html

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНОГО ФОРМУВАННЯ РОЗКЛАДУ ПОДІЙ

Болохнов А.А., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

Стрімкий розвиток соціальних мереж відзначається інформаційним перенасиченням, проте одночасно значна кількість важливих анонсів подій публікується виключно у стрічках соціальних мереж, що ускладнює доступ до них. Окрім того через динамічність оновлення інформації користувачі часто пропускають важливі заходи, маючи певну зашумленість у стрічці новин. Тому виникає необхідність у розробці спеціалізованих алгоритмів, які здатні автоматизувати процес моніторингу, виокремлювати релевантні анонси з неструктурованого тексту та формувати зручний для користувача розклад. Вирішення завдань, пов'язаних із семантичним аналізом постів та визначенням часу і місця подій, є актуальним для ефективного тайм-менеджменту.

Існує ряд методів для вирішення задачі аналізу тексту, включаючи глибоке навчання, використання великих мовних моделей або регулярних виразів. Ідея, взята за основу в даній роботі, полягає у гібридному підході, а саме використанні лінгвістичного аналізу (NLP) для визначення частин мови та евристичних правил для пошуку ключових слів. Головна перевага цього підходу порівняно зі складними нейромережами полягає у швидкодії та відсутності необхідності у тривалому навчанні моделі. Алгоритм базується на фільтрації за словником ключових слів (наприклад, «meetup», «conference») у поєднанні з аналізом контексту, що дозволяє точно ідентифікувати подію та відсіювати нерелевантний контент.

Алгоритм, що використовується в системі автоматичного формування розкладу подій, має наступну послідовність: отримання вебсторінки користувача, парсинг HTML-структури для виокремлення блоків повідомлень, токенізація тексту та визначення частин мови (POS-tagging). Для ідентифікації події перевіряється наявність іменників, що відповідають заданим тематичним маркерам. Важливим етапом є здобуття часових міток (дати та часу) як у формалізованому (ISO), так і у вільному текстовому форматі, а також визначення локації за допомогою контекстних прийменників. Завершується процес хронологічним впорядкуванням знайдених подій та фільтрацією за заданим часовим вікном (наприклад, найближчі 7 днів).

Програмне забезпечення розроблювалось з використанням мови програмування та пакетів requests для виконання HTTP-запитів до веб-ресурсів [1], bs4 (BeautifulSoup) для парсингу DOM-дерева HTML-сторінок [2], nltk для морфологічного аналізу тексту (токенізація, тегування частин мови) [3] та dateparser для інтелектуального розпізнавання дат у природній мові (наприклад, «next Friday») [4].

Робота програми розпочинається з ітеративного опитування списку заданих URL-адрес профілів, після чого сирі дані трансформуються у структуровані об'єкти. Для кожного повідомлення виконується лінгвістичний аналіз: якщо повідомлення класифікується як подія, з нього витягуються атрибути часу та місця. Результатом роботи є виведення повного хронологічного списку всіх знайдених подій, а також окремий блок нагадувань про події, що відбудуться у найближчому майбутньому, із зазначенням часу, що залишився до початку.

Література

[1] requests documentation [Electronic resource]. – Access mode: https://docs.python-requests.org/en/v2.0.0/

[2] bs4 documentation [Electronic resource]. – Access mode: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[3] nltk documentation [Electronic resource]. – Access mode: https://www.nltk.org/

[4] dateparser documentation [Electronic resource]. – Access mode: https://dateparser.readthedocs.io/en/latest/

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВІЗУАЛІЗАЦІЇ ТА СИНХРОНІЗАЦІЇ ІСТОРИЧНИХ ПОДІЙ

Бусол Д.М., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

В інформаційному просторі, що на даний момент свого розвитку характеризується експоненційним зростанням обсягів неструктурованих текстових даних, завдання автоматизованого аналізу набувають критичного значення. Особливу актуальність становить проблема виділення хронологічної інформації та її подальшої візуалізації. Це дозволяє виявляти приховані кореляції між різними сферами суспільного життя, наприклад, досліджувати вплив політичних подій на розвиток культури та мистецтва.

У даній роботі розроблено спеціалізоване програмне забезпечення, яке автоматизує повний цикл роботи з даними: від збору історичної інформації з відкритих енциклопедичних джерел до побудови синхронізованої часової шкали, дозволяючи не тільки систематизувати таку шкалу, але і відображати її безпосередньо.

Програмний продукт реалізовано з використанням мови програмування Python, яка є високорівневим інструментом з розвиненою екосистемою бібліотек для наукових обчислень та обробки тексту. В якості основного джерела даних обрано вільну енциклопедію Вікіпедія. Для взаємодії з її API використано бібліотеку wikipedia [1], що дозволяє автоматично отримувати текстовий контент статей та метадані за заданими ключовими словами без необхідності парсингу HTML-коду.

Для лінгвістичної обробки отриманого тексту застосовано інструментарій бібліотеки nltk [2]. Критично важливим етапом роботи алгоритму є сегментація тексту. Замість стандартних методів розділення рядків, використано інтелектуальний токенізатор sent_tokenize. Це рішення забезпечує коректну обробку речень, що містять скорочення, наприклад, «vol.», «pp.» тощо та складну пунктуацію, що значно підвищує якість подальшого аналізу.

Алгоритм роботи програми базується на послідовній обробці даних у функціях get_wiki_text та extract_events. Процес розпочинається із завантаження контенту та його попередньої очистки. Наступним кроком реалізовано механізм фільтрації шуму: алгоритм відсіює технічну інформацію, таку як ISBN, посилання на джерела та занадто короткі фрагменти тексту. Ключовим етапом є виділення хронологічних міток за допомогою модуля re. Використання оптимізованих регулярних виразів дозволяє точно ідентифікувати роки в тексті та сформувати структурований масив подій, де кожному запису присвоюється категорія: наприклад, історія або музика.

Візуалізація результатів реалізована засобами бібліотеки matplotlib.pyplot [3]. Графічне представлення даних виконано у вигляді діаграми розсіювання. Для наочного розмежування тематичних потоків події рознесені на різні рівні по вертикальній осі при збереженні єдиної хронологічної шкали абсцис. Для покращення сприйняття щільності подій у насичені історичні періоди використано налаштування прозорості маркерів, а також додано допоміжну сітку та легенду.

Отриманий програмний продукт дозволяє користувачеві швидко оцінювати історичний контекст та динаміку розвитку мистецьких явищ.

Література

[1] Wikipedia API documentation [Electronic resource]. – Access mode: https://pypi.org/project/wikipedia/

[2] NLTK 3.8.1 documentation [Electronic resource]. – Access mode: https://www.nltk.org/

[3] Matplotlib visualization library [Electronic resource]. – Access mode: https://matplotlib.org/

АНАЛІЗ МЕТРИКИ ПРОДУКТИВНОСТІ ТА ПРОГНОЗУВАННЯ ЕФЕКТИВНОСТІ РОЗПОДІЛЕНИХ КРОС-КУЛЬТУРНИХ КОМАНД ІТ- ПРОЄКТІВ

Вальчук Д. В.
Керівник: Назарова С. О.
E-mail: svitlana.nazarova@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

Глобалізація ІТ-галузі призводить до масового формування розподілених крос-культурних команд (далі – РККК) – команд, що складаються з учасників різних національних культур, які працюють у різних географічних локаціях. За даними McKinsey [3], команди з членами з різних культур на 35% частіше перевершують результати однорідних команд. Однак ця перевага реалізується лише при правильному урахуванні культурних відмінностей в процесі управлінні РККК. Дослідження показують [5], що культурні фактори впливають на ефективність метрик продуктивності на 25–35%. Мета даного дослідження полягає у проведенні порівняльного аналізу існуючих метрик продуктивності (DORA, SPACE, Velocity-based, Team Effectiveness, 7 Hidden Indicators) з фокусом на їхню придатність для оцінки РККК.

Однією з поширених метрик продуктивності команд ІТ-проєктів (з розробки ПЗ) є DORA Metrics, розроблена Google у 2018 р. [2], що включає п'ять показників для оцінювання ефективності доставки ПЗ: частота розгортання (Deployment Frequency) показує, як часто організація розгортає код у production, варіюючись від кількох разів на день для провідних команд до раз на місяць для низько-результативних; час виконання змін (Lead Time for Changes) вимірює тривалість від коміту до production, де провідні команди досягають менше 1 год., а відстаючі потребують 1–6 місяців; коефіцієнт відмов при змінах (Change Failure Rate) відстежує відсоток розгортань, що спричиняють відмови, з діапазоном від 0–15% для провідних до 46–60% для низько-результативних; середній час відновлення (Mean Time to Recovery) фіксує швидкість відновлення після відмови (від менше 1 год. до тиждень / місяць); та надійність (Reliability), додана у 2024 р., оцінює доступність системи від 99,95%+ до менше 95%. Попри те, що показники DORA Metrics є об'єктивними, легко автоматизованими через Git та CI/CD pipelines, та стали індустрійним стандартом для 60%+ ІТ-компаній, вони мають критичні обмеження для РККК: не враховують командну динаміку (психологічну безпеку, співпрацю), є культурно-нейтральними без адаптації до різних культурних цінностей, не враховують асинхронність роботи розподілених команд, та фокусуються виключно на доставці, ігноруючи якість комунікації та інші показники продуктивності ІТ-команди. Чутливість до культурних факторів цієї метрики становить лише 20% (низька), а прогнозна цінність відсутня, оскільки всі метрики є реактивними, вимірюючи фактичну продуктивність замість прогнозованої.

На відміну від DORA Metrics, SPACE Framework, створений GitHub, Microsoft та University of Victoria у 2021 р. [1], пропонує комплексний підхід до вимірювання продуктивності ІТ-фахівців через п'ять взаємопов'язаних вимірів з більш ніж 20 показниками на індивідуальному, командному та системному рівнях: задоволеність та благополуччя (Satisfaction & Well-being) охоплює задоволеність роботою, баланс роботи/життя та індикатори вигорання через опитування по шкалі 1–10, відстеження лікарняних та плинності кадрів; продуктивність (Performance) оцінює якість коду, завершення функціональності та надійність через аналіз якості при перегляді коду, частоти багів та дотримання SLA; активність (Activity) вимірює обсяг виходу та частоту комітів, включаючи кількість комітів, злитих PR та завершених story points; комунікація та співпраця (Communication & Collaboration) відстежує міжкомандну взаємодію та обмін знаннями через участь у PR review, повідомлення в Slack та створені документи; ефективність і послідовність (Efficiency & Flow) аналізує час до отримання цінності та переключення контексту, включаючи час циклу, ефективність потоку та частоту перебивань. Цей фреймворк демонструє значні переваги: комплексний підхід враховує людський фактор, гнучкість дозволяє адаптацію до різних

команд, фокус на благополуччя включає показники вигорання та баланс між роботою та особистим життям, а дослідження McKinsey показують, що в результаті його застосування 20–30% зменшились дефекти та 20% покращтвся досвід працівників [4]. Однак для РККК виявлено критичні обмеження: суб'єктивна інтерпретація метрик задоволеності, які розуміються по-різному в різних культурах; комунікаційна упередженість, де активність у Slack не еквівалентна якості комунікації у культурах Азії; та різні стандарти балансу між роботою та особистим життям між Європою (35 год./тиждень) та Силіконовою Долиною (США) (60 год./тиждень). Чутливість до культурних факторів становить 45% (середня), але прогнозна цінність відсутня, оскільки метрика є реактивною.

Velocity-based Metrics, що походять з Agile Manifesto та Scrum Framework (2001 р.) [6], фокусуються на вимірюванні швидкості доставки функціональності в ітераціях через чотири ключові показники: швидкість (Velocity) вимірює story points, завершені за спринт; темп вигорання роботи (Burndown Rate) показує швидкість "вигорання" роботи у спринті; передбачуваність спринту (Sprint Predictability) відстежує відсоток story points, завершених за планом; час циклу (Cycle Time) вимірює тривалість від "In Progress" до "Done". Ці метрики мають привабливу простоту завдяки легкому вимірюванню в ПЗ Jira або Azure DevOps, надають передбачуваність для планування спринтів, та забезпечують власне команди, коли команда сама оцінює story points, однак при цьому для РККК виявлені серйозні обмеження: інфляція story point виникає через різні культурні підходи до оцінювання, де німці схильні до консервативних оцінок, а американці до агресивних; 40% шуму через суб'єктивні оцінки робить метрики ненадійними; культурні відмінності в оцінці призводять до несумісності даних між командами з різних країн; а залежність від часових поясів створює зміщення вигоряння через асинхронну роботу розподілених команд, де частина команди може завершити роботу, поки інша ще спить. Чутливість до культурних факторів становить лише 25% (низька), а прогнозна цінність відсутня, оскільки метрики є винятково реактивними, відображаючи минулу продуктивність без здатності передбачати майбутні проблеми.

Team Effectiveness Frameworks, що розвинулися з досліджень Harvard Business Review у 2020-х р. [6], пропонують холістичний підхід до оцінювання командної динаміки через чотири основні напрями та шість додаткових ключових індикаторів: інноваційність (Innovation) вимірює кількість впроваджених ідей та рівень експериментування, демонструючи високу культурну чутливість через різні культурні установки до ризиків – від консервативних підходів у культурах з високим рівнем уникнення невизначеності до агресивного експериментування в Силіконовій Долині (США); прийняття рішень (Decision-Making) оцінює швидкість, якість рішень та рівень консенсусу, також з високою чутливістю, оскільки високо-контекстні культури (Азія) вимагають більше часу на досягнення консенсусу порівняно з низько-контекстними (США, Німеччина); гнучкість (Agility) відстежує час відповіді на зміни та швидкість адаптації з середньою чутливістю, залежною від рівня уникнення невизначеності у культурі; співпраця (Collaboration) аналізує міжфункціональну взаємодію та обмін знаннями з високою чутливістю через фундаментальні відмінності між індивідуалістичними (США, Західна Європа) та колективістськими (Азія, Латинська Америка) культурами. Шість ключових індикаторів додають глибини аналізу: ясність (Clarity) оцінює чіткість бачення, цінностей, ролей та відповідальності; синергія (Synergy) перевіряє, чи командна продуктивність перевищує суму індивідуальних частин; спроможність (Capacity) вимірює професійний розвиток через competency matrix; залученість (Commitment) відстежує емоційну прихильність через аналіз організаційної культури; узгодженість (Alignment) перевіряє відповідність командних цілей стратегічним; результати (Results) оцінюють доставку відповідно до бюджету та стандартів якості. Цей фреймворк демонструє найвищу чутливість до культурних факторів серед усіх аналізованих – 65% (висока) та має короткострокову прогнозну цінність 1–2 місяці, однак незручний через складність вимірювання (забагато суб'єктивних індикаторів), відсутність стандартних метрик (кожна організація визначає свої), та культурну упередженість у розумінні ефективної співпраці представниками різних культур.

7 Hidden Indicators, розроблені FullScale.io у 2025 р. [3], представляють революційний підхід до вимірювання прогнозованої ефективності команди, здатний прогнозувати проблеми за 4–6 місяців до їхнього виникнення з точністю 92–96% та ROI 300–600% через запобігання вигорання, перепрацювання та затримки [3]. До складі цих метрик входять: індекс розподілу когнітивного навантаження (CLDI) – вимірює рівномірність розподілу складності завдань; індекс психологічної безпеки (PSI) – відстежує частоту конструктивних технічних дискусій на тиждень; коефіцієнт швидкості нвчання (LVC) – аналізує діяльність з обміну знаннями на особу чи квартал; патерни латентності рішень (DLP) – вимірюють час від проблеми до архітектурного рішення; частота переключення контексту (CSF) – відстежує кількість переключень між завданнями на день; індекс асиметрії співпраці (CAI) – обчислює співвідношення допомоги наданої до отриманої; емоційне навантаження технічного боргу (TDEL) – використовує аналіз настроїв коміт-повідомлень для виявлення проблемних модулів. Ці індикатори покривають чотири з п'яти виявлених прогалин в сучаснтх дослідженнях РККК: «прогнозна здатність»: має раннє попередження через 4–6 місяців, «врахування людських факторів»: через фокус на динаміці команди (PSI, CLDI, LVC), «врахування розподіленої специфіки»: через врахування часових поясів у DLP, та частково «недоліки впровадження» через середню складність впровадження з 2–3 місяців часу для оцінки. Чутливість до культурних факторів становить 55% (середньо-висока), оскільки метрики явно враховують динаміку команд, але потребують культурної адаптації: наприклад, PSI інтерпретується по-різному в культурах з прямим зворотнім зв'язком (Північна Європа, Україна) порівняно з культурами з непрямим зворотним зв'язком (Азія), TDEL залежить від культурних норм вираження емоцій у професійному контексті, а CAI вимагає коригування на колективістські або індивідуалістичні культурні цінності. Це єдина метрика з високою прогнозною цінністю, що робить її найефективнішим інструментом для РККК, де раннє виявлення проблем критичне через складність координації та комунікації, що обумовлені часовими зонами та культурними барєрами.

Аналіз п'яти основних метрик виявив критичне розшарування в підходах до вимірювання та прогнозування продуктивності РККК. Так, DORA Metrics залишаються індустрійним стандартом завдяки об'єктивності та автоматизації, однак нечутливість до культури та відсутня прогнозна цінність роблять їх недостатніми для оцінки роботи РККК. SPACE Framework покращує ситуацію культурною чутливістю через включення людського фактору, однак залишається реактивним. Velocity-based Metrics підтримують передбачуваність спринтів, але значний шум від суб'єктивної оцінки має деструктивний вплив на обєктивність показників розподіленої команди, знижуючи її достовірність. Team Effectiveness Frameworks демонструють найвищу культурну чутливість, однак складність та незначний період прогнозування обмежують їх впровадження. 7 Hidden Indicators репрезентують парадигмальний зсув до прогнозного вимірювання, досягаючи 92–96% точності та покривають 4 з 5 прогалин, що найбільше за усіх інших метрик. Отже, результати проведеного аналізу показали, що жодна з поширених метрик повністю не враховує крос-культурні особливості розподілених команд, що обумовлює необхідність у розробці гібридної метрики для оцінки та прогнозування продуктивності РККК.

Література

[1] Forsgren N., Storey M.-A., Maddila C., et al. The SPACE of Developer Productivity: There's More to It than You Think. Queue. 2021. Vol. 19, No. 1. P. 20–48.

[2] Atlassian. DORA Metrics: How to measure Open DevOps Success. 2025. URL: https://www.atlassian.com/devops/frameworks/dora-metrics (дата звернення: 25.01.2026).

[3] Software Development Team Metrics That Predict Performance. FullScale. 2025. URL: https://fullscale.io/blog/software-development-team-metrics/ (дата звернення: 25.01.2026).

[4] Hunt V., Prince S., Dixon-Fyle S., Yee L. Delivering through Diversity. McKinsey & Company. 2018. URL: https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/delivering-through-diversity (дата звернення: 25.01.2026).

[5] Hofstede G., Hofstede G. J., Minkov M. Cultures and Organizations: Software of the Mind. 3rd ed. New York : McGraw-Hill, 2010. 576 p.

[6] Ukrainian software developers in cross-cultural communication. N-iX. 2019. URL: https://www.n-ix.com/ukrainian-software-developers-culture-communication-outsourcing/ (дата звернення: 25.01.2026).

ОПТИМАЛЬНЕ РОЗМІЩЕННЯ БАГАТОВИМІРНИХ КУЛЬ ДЛЯ КОДУВАННЯ МЕДИЧНИХ ДАНИХ: МАТЕМАТИЧНА МОДЕЛЬ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ

Веретельник К.О.[1], Чугай А.М.[1], Яськова Є.Г.[2]
E-mail: kostiantyn.veretelnyk@hneu.net
[1]Харків, Харківський національний економічний університет імені Семена Кузнеця
[2]Харків, Харківський національний університет імені В.Н. Каразіна

Обсяг медичних даних, що генеруються сучасними методами візуалізації, зокрема 3D КТ, МРТ та 4D динамічних досліджень, стрімко зростає, створюючи серйозні виклики для їх зберігання та особливо для передачі в телемедичних системах [1]. Актуальні дослідження підкреслюють, що це створює суттєві навантаження на інфраструктури зберігання та передачі, зокрема PACS-системи та телемедичні мережі [2]. Тому розробка ефективних методыв компресії э актуальною для сучасної медицини.

У сучасних методах стиснення медичних даних (зокрема у deep-learning підходах типу UniCompress) початкові об’єми КТ/МРТ розбиваються на багатовимірні блоки, які після перетворення (DWT, CNN-encoder) представлено векторами високої розмірності. Подальше кодування таких векторів ґрунтується на виборі одного елемента з кінцевої кодової книжки. Кодова книжка, у свою чергу, є множиною точок у багатовимірному евклідовому просторі. Кожна точка відповідає можливому відновленому блоку даних [3]. Щоб мінімізувати помилки передачі, відстань між кодовими точками має бути максимально великою. Це призводить до класичної геометричної задачі – оптимального розміщення точок у високовимірному просторі, яка еквівалентна задачі пакування багатовимірних куль. Такий зв’язок між побудовою кодових книжок та сферичними пакуваннями викладено в сучасних роботах з геометрії кодів і решіток [4,5]. Таким чином, задача оптимального кодування медичних даних природно зводиться до задачі оптимізації конфігурації куль у просторі, де розміщення центрів куль визначає якість стиснення та стійкість до шумів у процесі передачі .

Нехай $x_i \in \mathbb{R}^N$, $i = 1,...,M$, центри куль, тобто кодові точки кодової книжки. Кожний вектор x_i відповідає можливому відновленому фрагменту медичних даних, який декодер може обрати після квантованого або векторного кодування. Розмір кодової книжки M – це кількість кодових слів (центральних точок). Збільшення M потенційно підвищує точність відновлення, але зменшує можливу мінімальну відстань між точками.

Передача по каналу і квантування вимагають, щоб коди не виходили за межі допустимої потужності: $\|x_i\|^2 \le R^2,\ i = 1,...,M$. Тут R – максимально допустимий радіус N-вимірної кулі, що відповідає обмеженню на енергію сигналу або величину можливого вектора після перетворення.

Позначимо через $d > 0$ мінімальну відстань між будь-якими двома кодовими точками: $\|x_i - x_j\| \ge d,\ i, j = 1,...,M,\ i \ne j$. Чим більше d , тим легше декодер розрізняє кодові слова. Мінімізація помилок декодування безпосередньо залежить від геометричного рознесення точок. Мета – максимізувати мінімальну відстань d між центрами куль за фіксованого M та обмежень $\|x_i\| \le R,\ i = 1,...,M$.

Враховуючи, що задача є NP-складною, створюється множина початкових точок $\{x_i^{(k)}\}$, які належать кулі радіуса R . Для цього використовується рівномірна випадкова ініціалізація. Далі для пошуку локального максимума використовується безкоштовний

солвер задач нелінійного програмування IPOPT [6]. Серед отриманих результатів вибираємо конфігурацію розміщених куль з максимальним значенням d .

Для реалізації метода створено програмний комплекс. Він складається з трьох основних модулей: оркестратор (Experiment Manager), генератор початкових конфігурацій та інтерфейс з модулем IPOPT.

Оркестратор централізовано керує всім робочим циклом: створенням множини стартів, їхньою підготовкою, паралельним запуском локальних оптимізацій у IPOPT, моніторингом, логуванням та збереженням найліпшого результату.

Генератор початкових конфігурацій створює стартові набори координат центрів куль у гіперсферичній системі координат з урахуванням амплітудних обмежень, забезпечує різноманітність початкових станів, щоб уникати «пасток» локальних екстремумів при подальшій локальній оптимізації.

Інтерфейс із IPOPT виконує запуск локальної оптимізації для кожного стартового набору, контроль критеріїв зупинки, ведення журналу обчислювального процесу. Він також керує параметрами оптимізації: точністю, максимальною кількістю ітерацій, обчисленнями нев'язок обмежень, градієнтів, гесіанів, значеннями цільової функції, збереженням отриманих конфігурацій, керуванням параметрів IPOPT.

Для експериментальної оцінки метод було протестовано на конфігураціях із 50–100 куль у просторах розмірності від 2 до 64, що дало змогу перевірити стабільність алгоритму та його масштабованість у різних геометричних умовах. Отримані результати демонструють, що запропонований геометрично орієнтований підхід забезпечує формування кодових книжок із максимально можливою роздільною здатністю у високих розмірностях, що підвищує стійкість до шумів і точність відновлення блоків медичних даних.

Крім того, запропонована методологія має потенціал застосування не лише в телемедичних системах та медичних архівах, а й у будь-яких задачах, що вимагають оптимального розміщення кодових точок у багатовимірному просторі, зокрема в бездротових комунікаціях, оптичних системах зв'язку, квантових каналах, задачах кластеризації та побудові стійких високорозмірних репрезентацій у машинному навчанні.

Література

[1] Chitra P. 4D Image Compression in Healthcare Using Convolutional Neural Networks (CNNs). 2024 2nd International Conference on Computing and Data Analytics (ICCDA), Shinas, Oman. 2024. P. 1–7. https://doi.org/10.1109/ICCDA64887.2024.10867362

[2] Yang R., Chen Y., Zhang Z. et al. UniCompress: Enhancing Multi-Data Medical Image Compression with Knowledge Distillation. arXiv:2405.16850. 2024. [Electronic resource]. – Resource access mode: https://doi.org/10.48550/arXiv.2405.16850

[3] Gümüş K., Chen B., Bradley T., Okonkwo C. A Simplified Method for Optimising Geometrically Shaped Constellations of Higher Dimensionality. arXiv:2307.05179. 2023. [Electronic resource]. – Resource access mode: https://doi.org/10.48550/arXiv.2307.05179

[4] D. Radchenko, Sphere Packings, Lattices and Codes, ETH Zürich, 2021. [Electronic resource]. – Resource access mode:
https://metaphor.ethz.ch/x/2021/fs/401-3520-21L/sc/topics_splac.pdf

[5] Chen B. et al. Geometrically-Shaped Multi-Dimensional Modulation Formats in Coherent Optical Transmission Systems. Journal of Lightwave Technology. Vol. 41, no. 3. 2023. P. 897–910. https://doi.org/10.1109/JLT.2022.3204101

[6] Wächter A., Biegler L. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. Vol. 106. 2006. P. 25–57. https://doi.org/ 10.1007/s10107-004-0559-y

РОЗРОБЛЕННЯ ВЕБДОДАТКІВ НА ОСНОВІ ВЕБФРЕЙМВОРКУ ДЛЯ РЕАЛІЗАЦІЇ ДОСТУПУ ДО СИСТЕМ КЕРУВАННЯ БАЗАМИ ДАНИХ

Водоп'янов М.О., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

В сучасних реаліях ефективна обробка інформації грає велику роль у житті як звичайних користувачів, так і великих бізнесів. Фільтрація та пошук у величезних масивах даних може займати недоступну на практиці кількість часу та ресурсів. Інформація потребує обробки та структуризації для оптимізації запитів до неї, і для цього серверні частини успішних онлайн-проєктів використовують бази даних для збереження, індексації та обробки інформації.

Мова структурних запитів дозволяє розробникам зберігати дані в табличному вигляді, виконувати складні запити до створених таблиць та контролювати доступ до них, встановлювати правила їх обробки. Одним з найпопулярніших виборів розробників серед систем керування баз даних є MySQL [1], яку використовують в своїй основі такі популярні застосунки як Uber, Twitter, Slack, Netflix та Pinterest, де вона використовується самостійно, в поєднанні з іншими системами або з власними інструментами. Вибір зумовлений її швидкодією, нативністю до операційних систем з сімейства Linux, безкоштовною ліцензією та широким інструментарем.

В межах програмного проєкту було реалізовано вебдодаток для підтримки роботи ресторану, що використовує базу даних MySQL для збереження даних меню, столів, користувачів та замовлень, отриманих з вебсторінки. Реалізовано проєкт було на мові програмування Python, відомою своєю універсальністю завдяки великій кількості доступних пакетів та модулів. Зокрема, в роботі використовувався пакет вебфреймворку Django [2], що дозволяє створювати комплексні вебзастосунки на основі готових структурних рішень та інструментів, з використанням системи керування базою даних MySQL.

Для організації взаємодії між базою даних та поданнями фреймворку були розроблені моделі, включаючи модель столу, резервації столу, елемента меню, замовлення та частин замовлення. Вони містять в собі основні атрибути предметів, якими можуть бути числові дані, текстові дані, дати, посилання на інші моделі, тощо.

Під час будь-яких взаємодій користувача з інтерактивними елементами вебсторінки функція JavaScript, що підписана на взаємодію з цим елементом, передає необхідні дані за визначеною в файлі маршрутизації адресою та очікує на відповідь. Там ці дані оброблюються за відповідною логікою, описаною в файлі подань. Завдяки декораторам, які надає фреймворк, перед обробкою даних є можливість перевірки валідності запиту, стану авторизації користувача та його належності до певного класу користувачів, що спрощує процес розробки. Можливими результатами обробки може бути завантаження певної сторінки, перенаправлення користувача за певною адресою, повернення інформації у формі JSON-файлу або негативна відповідь з кодом помилки, що може бути використана JavaScript для подальшої обробки функції. Для реалізації базової роботи з вебдодатком підтримки роботи ресторану було створено подання, які забезпечують базові функції користувачів відповідних ролей.

У підсумку в роботі було виконано розробку вебдодатку, який може використовуватись для підтримки роботи ресторану, є придатним для подальшого розширення, застосовує принципи реалізації доступу до систем керування базами даних на основі моделей даних, організованих засобами вебфреймворку Django.

Література

[1] MySQL 8.4 Reference Manual [Electronic resource]. – Access mode: https://dev.mysql.com/doc/refman/8.4/en/

[2] Django Documentation – Django Web Framework [Electronic resource]. – Access mode: https://docs.djangoproject.com/

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПЕРСОНАЛІЗОВАНОГО НАПОВНЕННЯ НОВИННОЇ СТРІЧКИ

Гершиков В.І., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

Своєчасне отримання актуальної та релевантної інформації серед сучасних інформаційних потоків є важливою складовою повсякденного життя. Обсяг новинних матеріалів постійно зростає, а тематика інформаційних ресурсів є надзвичайно різноманітною, у зв'язку з чим не всі новини однаково важливі для кожного користувача. Велика кількість джерел та безперервний потік інформації ускладнюють процес ручного відбору новин, що зумовлює потребу у створенні персоналізованих інформаційних стрічок, здатних автоматично аналізувати та відбирати контент відповідно до інтересів користувача.

Для розв'язання описаної проблеми розроблено програмне забезпечення персоналізованого наповнення новинної стрічки на основі RSS-стрічок новинних сайтів із використанням методів машинного навчання. Запропоноване програмне забезпечення автоматично здійснює збір новин, аналізує заголовки та короткі описи статей, класифікує їх за тематичними категоріями та формує рекомендації для різних груп користувачів, що дозволяє значно зменшити обсяг нерелевантної інформації.

Алгоритм, покладений в основу реалізації програмного забезпечення, має таку послідовність: завантаження RSS-стрічок новинних ресурсів, парсинг RSS-даних, попередня обробка текстової інформації, перетворення текстових даних у векторне представлення, навчання моделі класифікації за відсутності навченої раніше моделі та формування рекомендацій на основі результатів класифікації. На етапі попередньої обробки здійснюється очищення тексту від HTML-тегів, нормалізація регістру символів та усунення зайвих службових елементів, що підвищує якість подальшого аналізу.

Ключовим елементом програмного забезпечення є модель машинного навчання, яка використовується для класифікації новин. Для реалізації моделі застосовано бібліотеку scikit-learn [1], що є однією з найпоширеніших бібліотек машинного навчання для мови програмування Python. Зокрема, для перетворення тексту у числове представлення використовується метод векторизації TF-IDF (Term Frequency – Inverse Document Frequency), який дозволяє відобразити важливість слів у тексті відносно всієї множини новин.

Після векторизації текстових даних формується модель класифікації, яка навчається на підготовлених текстових прикладах і відповідних тематичних категоріях. У процесі навчання модель аналізує статистичні залежності між словами та тематиками, після чого може автоматично визначати, до якої категорії належить нова, раніше невідома новина. Таким чином, класифікація новин здійснюється без жорстко заданих правил, а на основі результатів роботи навченої моделі.

На основі результатів класифікації формується рекомендація. Значення впевненості моделі використовується як внутрішня характеристика результату класифікації та дозволяє впорядковувати новини за рівнем їх відповідності інтересам користувачів.

Для уникнення дублювання інформації у програмі реалізовано механізм виявлення схожих новин, що базується на порівнянні заголовків із використанням метрик текстової подібності. Це дозволяє агрегувати новини з різних джерел, які описують одну й ту саму подію, та відображати їх як єдиний інформаційний елемент із зазначенням кількох джерел.

Таким чином, розроблене програмне забезпечення поєднує використання RSS-технологій, методів обробки природної мови та машинного навчання для створення інтелектуальної системи персоналізованого наповнення новинної стрічки.

Література

[1] scikit-learn: Machine Learning in Python [Electronic resource]. – Access mode: https://scikit-learn.org/stable/

ОСОБЛИВОСТІ ЧИСЕЛЬНИХ РОЗРАХУНКІВ БУДІВЕЛЬНИХ КОНСТРУКЦІЙ ЗА ГРАНИЧНИМИ СТАНАМИ

Дагіль В.Г., Кучер Г.І.
dahil_viktoriia@nuczu.edu.ua
Черкаси, Національний університет цивільного захисту України

Сучасний розвиток комп'ютерних технологій дозволяє перейти від експериментальних досліджень і аналітичних методів розрахунків будівельних конструкцій [3], до комп'ютерного моделювання та використання чисельних методів розрахунків. Не будемо зупинятися на всіх доступних програмних комплексах, а приділимо увагу засобу вирішення крайових задач при розрахунках будівельних конструкцій за граничними станами. Це мова програмування Python, яку використовують при написанні високоструктурованих програм і веб-застосунків, які інтерпретовані з платформою Microsoft.NET Framework. Python зручний для розв'язання математичних проблем, при рішенні крайових задач має обширну кількість бібліотек, але має недолік щодо швидкості при роботі в машинному навчанні з великою кількістю даних.

Рішення крайової задачі зводиться до наступних етапів:

1. Аналіз та постановка задачі: Визначення диференціального рівняння та крайових (граничних) умов.

2. Створення розрахункової схеми. Розрахункова схема призначена для визначення напружено-деформованого стану. Для реалізації цього етапу розв'язання задачі у процесорі комплексу (або в окремих CAD-системах) будують геометричну модель об'єкта.

3. Створення дискретної моделі: Заміна диференціального рівняння (метод скінченних різниць) або апроксимація функцій (метод скінченних елементів). Застосовуючи чисельний спосіб, складаємо модель, утворену з розрахункової схеми яка має кінцеве число ступенів свободи.

4. Розв'язання системи рівнянь: Обчислення факторів попередньо-напруженого стану у довільних точках кінцево-елементної моделі, на підставі відомих з теорії пружності та пластичності (внутрішніх зусиль, напруг, переміщень довільних точок) з побудовою їх епюр.

5. Аналіз результатів: Перевірка виконання умов та оцінка точності, зазначають матеріали. Оптимізація функцій.

6. Візуалізація та збереження отриманих результатів

Розглянемо саме питання аналізу результатів, їх оптимізацію та візуалізацію. В зв'язку з тим, що крайова задача – це задача визначення розв'язку диференціального рівняння, який задовольняє умови на границі області (крайові умови), то виникає необхідність оптимізації функцій у розрахунках будівельних конструкцій. Тобто пошук найкращих геометричних чи параметричних характеристик будівельних конструкції (мінімальна вага, вартість, максимальна жорсткість, вогнестійкість) за допомогою методів математичного програмування. Оптимізація базується на створені математичної моделі, де функція (поперечна сила, деформація, згинальний момент) залежить від керованих змінних.

Оберемо для тестування оптимізаційних функцій [1] в Python, одну з класичних функцій Хіммельблау [2] і виконаємо порівняння градієнтних методів оптимізації функцій кількох змінних в наступній послідовності:

In [1]: Визначення часткових похідних за допомогою символьних обчислень Sympy

```
import sympy as sp
from IPython.display import display


print('Обрана функція: (x**2 + y - 11)**2 + (x + y**2
- 7)**2', '\n') x, y = sp.symbols('x y')

print('Часткова похідна функції по x:', sp.diff((x**2 + y - 11)**2 + (x + y**2 - 7)**2,
x)) print('Часткова похідна функції по y:', sp.diff((x**2 + y - 11)**2 + (x + y**2 -
```

In [2]: Визначення градієнтних спусків

```
import
plotly.graph_objects as
go import
plotly.figure_factory
as ff import numpy as
np
import plotly.io as pio
pio.renderers.default='notebook'

def f(x:float, y:float):

    return (x**2 + y - 11)**2 + (x + y**2 - 7)**2
def df_dx(x:float, y:float):

    return 4*x*(x**2 + y - 11) + 2*x + 2*y**2 - 14
```

```
# plot_3d_surface() start
Contour=False
x_plt = np.arange(-4, 4, 0.05)

y_plt = np.arange(-4, 4, 0.05)

Z_plt = np.array([[f(x, y) for x in x_plt] for y in y_plt]) trace1 = go.Contour(x=x_plt, y=y_plt,

z=Z_plt)


#   створення даних для побудування фігури;   оптимальна палітра для конкретного графіку
trace2 = go.Surface(x=x_plt, y=y_plt, z=Z_plt, colorbar_x=-0.5, opacity= 0.83)


data1 = [trace1] data2 = [trace2]
```

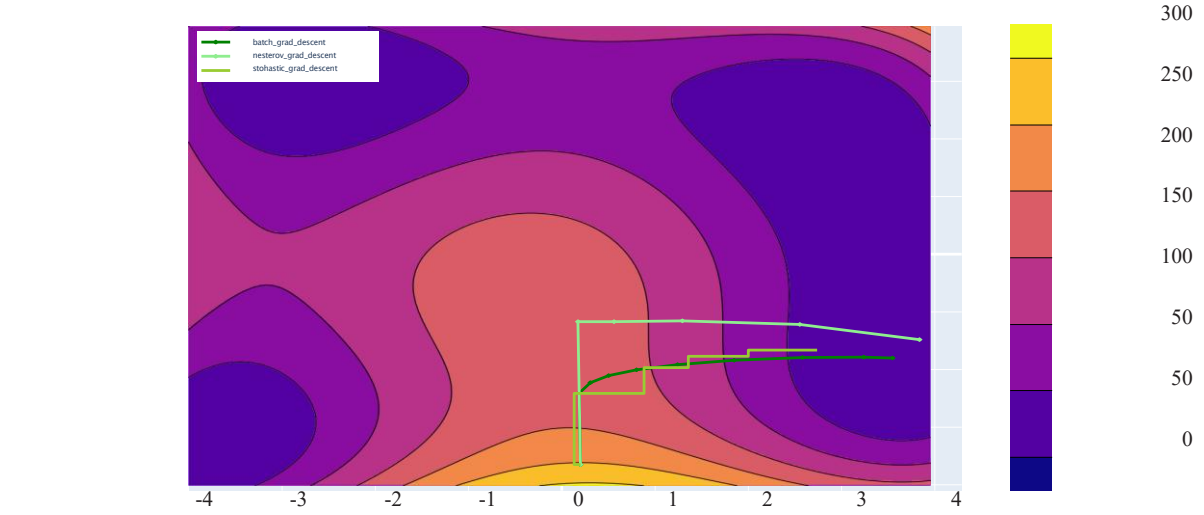
In [3]: Методи оптимізації

```
def batch_grad_descent(x, y, lmd_x, lmd_y, number_of_iterations):

    """Классичний градієнтний спуск з константними заданними кроками та наперед заданною кількістю
    ітерацій""" dots = np.array([x, y], dtype=float).reshape(-1, 2)

    for n in
        range(number_of_it
        erations): x = x -
        lmd_x * df_dx(x,
        y)

        y = y - lmd_y * df_dy(x, y)
        dots = np.append(dots, [[x, y]], axis=0)
```

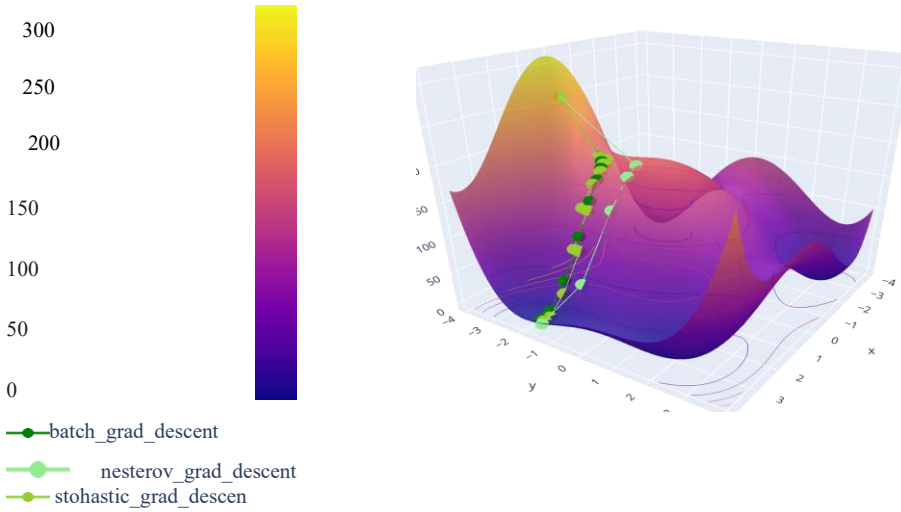
```
def nesterov_grad_descent(x, y, alpha, beta,
    number_of_iterations): """Accelerated gradient
    descent"""

    dots = np.array([x, y], dtype=float).reshape(-1, 2)

    x = x -
    alpha *
    df_dx(x, y)
    y = y -
    alpha *
    df_dy(x, y)
    dots = np.append(dots, [[x, y]], axis=0)

    for k in range(2, number_of_iterations):

        x = x - alpha * df_dx(x + beta*(x - dots[k-1, 0]), y) +
```

```
def stohastic_grad_descent(x, y, lmd, number_of_iterations):

    """Стохастичний градієнтний спуск з константними заданними кроками та наперед заданною
    кількістю ітерацій""" dots = np.array([x, y], dtype=float).reshape(-1, 2)

    for n in range(number_of_iterations):

        if
            np.random.ra
            ndint(0,2)
            == 0: x = x
            - lmd *
            df_dx(x, y)
        else:
            y = y - lmd * df_dy(x, y)
```

Level lines



Function



Висновок дослідив градієнтні методи оптимізації функцій кількох змінних, визначили, що результат роботи кожного з алгоритмів залежить від обраного початкового наближення та гіперпараметрів: стохастичному градієнтному спуску потрібно більше ітерацій, проте він має вдвічі менше обчислень, ніж у класичному; завдяки коефіцієнтам та і використанням попередніх результатів значно менше відхиляється від своєї траєкторії та швидше сходиться до мінімуму. Використання таких методів дозволяє знайти найкращі рішення на етапі проектування, підвищуючи ефективність конструкцій.

Література

[1] https://en.wikipedia.org/wiki/Test_functions_for_optimization

[2] https://en.wikipedia.org/wiki/Himmelblau%27s_function

[3] Дагіль В.Г., Хаткова Л.В. Розробка методики розрахунку показників надійності будівель з використанням теорії імовірностей та математичної статистики. Збірник наукових праць Черкаського інституту пожежної безпеки імені Героїв Чорнобиля Національного університету цивільного захисту України «Надзвичайні ситуації: попередження та ліквідація» Том 7 №1 (2023) с.15-22 https://fire-journal.ck.ua/index.php/fire/article/view/161/138

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ РЕКОМЕНДУВАННЯ КНИГ НА ОСНОВІ ІНТЕРЕСІВ КОРИСТУВАЧА

Єфремов А.Д., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

Обсяг доступної літератури у світі, у відповідних програмних системах постійно зростає. Користувачам стає складно обирати книги, які відповідають їхнім інтересам та вподобанням. Потреба у персоналізованих рекомендаціях літератури зростає, оскільки читач не може ознайомитися зі всіма наявними виданнями. В таких умовах розробка програмного забезпечення, яке автоматично аналізує характеристики книг та формує індивідуальні рекомендації, набуває великого практичного значення.

Існують різні системи рекомендацій літератури, які ґрунтуються на популярності книг, рейтингах користувачів або колективному досвіді. У цій роботі пропонується контентно-орієнтована система, яка аналізує зміст книг і формує рекомендації без потреби в попередньо розмічених даних. На відміну від систем на основі оцінок користувачів, цей підхід дозволяє рекомендувати книги новим користувачам або менш популярні видання.

Програмне забезпечення реалізоване мовою Python, яка є високорівневою та дозволяє ефективно працювати з даними завдяки великій кількості готових бібліотек. Для обробки табличних даних використано бібліотеку pandas [1], що дозволяє зручно зберігати та опрацьовувати інформацію про книги у форматі DataFrame. Для машинного навчання та аналізу тексту застосовано бібліотеку scikit-learn, зокрема клас TfidfVectorizer для перетворення текстових описів книг у числові вектори та функцію cosine_similarity для визначення схожості між книгами [2].

Алгоритм, покладений в основу системи рекомендацій, має наступну послідовність.

Спочатку виконується завантаження даних про книги із ресурсу OpenLibrary, включаючи назву, авторів, жанр, тематичні мітки, пов’язаних персонажів, місця дії, часові періоди та рік першої публікації [3]. Після отримання цих даних виконується формування табличної структури даних за допомогою пакету pandas, де кожна книга зберігається як окремий рядок DataFrame. Далі виконується створення текстового профілю книги, що об’єднує усі характеристики книги в один рядок тексту. На наступному кроці відбувається перетворення текстового профілю книги в числовий вектор з використанням TF-IDF, що дозволяє оцінити значущість слів у контексті всієї колекції книг. Далі виконується обчислення косинусної схожості між книгами, що показує, наскільки книги подібні між собою за змістом. На наступному кроці враховуються інтереси користувача, тобто книги, які він уже прочитав. Для кожної непрочитаної книги обчислюється середня схожість із прочитаними. Далі забезпечується формування рекомендацій, шляхом сортування книг за ступенем схожості та відбору ТОП-10 найрелевантніших для користувача.

Розроблене програмне забезпечення дозволяє автоматично формувати персоналізовані рекомендації книг на основі їхніх характеристик та вподобань користувача. Система працює без потреби у розмічених даних, на основі навчання без учителя, та ефективно виділяє найбільш релевантні книги з великої колекції. Застосування бібліотек pandas та scikit-learn забезпечує зручну обробку даних та надійні обчислення схожості, що робить підхід гнучким та придатним для подальшого розширення на інші типи контенту.

Література

[1] Pandas documentation [Electronic resource]. – Access mode: https://pandas.pydata.org/docs/index.html

[2] scikit-learn API [Electronic resource]. – Access mode: https://scikit-learn.org/stable/modules/classes.html

[3] OpenLibrary API documentation [Electronic resource]. – Access mode: https://openlibrary.org/developers/api

ЗАСТОСУВАННЯ GIT ТА GITHUB ЯК ІНСТРУМЕНТІВ СПІЛЬНОЇ РОЗРОБКИ ПЗ

Кузьменко Ю.Є.
E-mail: Kuzmenko.yuliia@student.karazin.ua
Харків, Навчально-науковий інститут «Каразінський банківський інститут»
Харківського національного університету імені В.Н. Каразіна

Розробка програмного забезпечення дедалі частіше відбувається в командах, учасники яких працюють у різних містах, країнах і часових поясах. За таких умов ключовим викликом стає не написання коду як такого, а організація спільної роботи, узгодження змін і збереження цілісності проєкту. Спільна розробка програмного забезпечення передбачає постійну взаємодію між учасниками команди, узгодження змін у коді та контроль цілісності програмного продукту. За умови, коли над одним проєктом одночасно працюють декілька розробників, особливо у форматі віддаленої або розподіленої роботи, зростає ризик втрати даних, конфліктів версій і неузгоджених змін. Саме тому постає потреба у використанні спеціалізованих інструментів, здатних забезпечити контроль версій, прозорість процесу розробки та ефективну командну взаємодію.

Однією з найпоширеніших систем контролю версій є Git, а одним із найпопулярніших сервісів для організації спільної роботи на його основі – GitHub. Усім учасникам командної розробки важливо розуміти відмінність між Git як системою контролю версій і GitHub як сервісом для спільної роботи над Git-репозиторіями. Git – це розподілена система контролю версій, яку можна використовувати локально чи разом із будь-яким віддаленим сервером, що підтримує Git, без прив'язки до конкретної платформи. Він дозволяє фіксувати зміни, створювати гілки, зливати їх та переглядати історію змін, що є базовою складовою колективного керування програмним кодом.

GitHub побудований на основі Git і пропонує веб-інтерфейс із додатковими можливостями: відстеження проблем, механізм pull request для рецензування змін, систему керування проєктами та навіть можливість перегляду різниці між версіями коду. GitHub є однією з найпопулярніших платформ для хостингу відкритих проєктів і колаборативної роботи – щонайменше 100 мільйонів проєктів розміщено на ній у всьому світі.

Централізація GitHub як сервісу має як переваги, так і обмеження. З одного боку, платформа забезпечує стандартний набір інструментів для організації командної роботи, включно з вбудованою CI/CD-автоматизацією через GitHub Actions, IDE у браузері через GitHub Codespaces та інші сервіси, що заощаджують час розробників. З іншого – частина розширених функцій доступна лише у платних підписках (наприклад, поглиблена аналітика безпеки чи деякі AI-інструменти у GitHub Copilot), які можуть бути дорогими для окремих розробників чи некомерційних проєктів. Це типова модель сучасних SaaS-платформ: базовий функціонал безкоштовний, але просунуті можливості – за гроші.

Серед основних недоліків GitHub експерти та практики називають:

– Складність для новачків – Git та GitHub мають інтерфейс і робочі процеси, що можуть бути важкими для розуміння без практики.

– Залежність від зовнішньої платформи – дані та інструменти розміщені на серверах GitHub, і повна втрата доступу або зміна політик може призвести до проблем.

– Частина потужних розширень, таких як повноцінні CI/CD або розширений пошук, може вимагати окремих платних рішень або інтеграції з іншими сервісами.

Це спонукає частину команд обирати альтернативні рішення або доповнювати GitHub іншими інструментами. Наприклад:

– GitLab – комплексна платформа для розміщення Git-репозиторіїв із вбудованими CI/CD та можливістю самостійного розгортання.

– Gitea – легка open-source альтернатива, яку можна самостійно розмістити на власному сервері, забезпечивши повний контроль над даними.

– SourceForge, Launchpad та інші сервіси теж пропонують Git-хостинг, але частіше з меншим набором сучасних можливостей, ніж GitHub.

Такі альтернативи часто підбираються командами, які хочуть уникнути залежності від централізованого сервісу, мають суворі вимоги до безпеки даних або потребують повністю безкоштовного рішення з відкритою архітектурою.

Література

[1] Що таке GitHub і як з ним працювати [Електронний ресурс] – Режим доступу до ресурсу: https://goit.global/ua/articles/shcho-take-github-i-iak-z-nym-pratsiuvaty/

[2] GitHub як платформа для спільної розробки та контролю версій [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/GitHub

[3] Git та GitHub: базові налаштування і принципи роботи [Електронний ресурс] – Режим доступу до ресурсу: https://dan-it.com.ua/uk/blog/shho-take-git-ta-github-bazovi-nalashtuvannya-i-yak-z-nymy-praczyuvaty/

[4] Основи Git і GitHub: система контролю версій та робота з репозиторіями [Електронний ресурс] – Режим доступу до ресурсу: https://dmytro.github.io/startpack/git_github/

[5] GitHub Codespaces — хмарне середовище розробки [Електронний ресурс] – Режим доступу до ресурсу: https://github.com/features/codespaces

[6] Gitea — open-source платформа для Git-хостингу [Електронний ресурс] – Режим доступу до ресурсу: https://about.gitea.com/

[7] Альтернативи GitHub для розміщення Git-репозиторіїв [Електронний ресурс] – Режим доступу до ресурсу: https://www.wearedevelopers.com/en/magazine/298/top-github-alternatives

ОГЛЯД РОЗВИТКУ НЕЙРОМЕРЕЖ ТА СТВОРЕННЯ СПЕЦІАЛІЗОВАНИХ БІБЛІОТЕК З ВІЛЬНИМ ДОСТУПОМ

Мартинова А.А.
Керівник: Шаповалова О.О.
Email: anastasiia.martynova@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

Як і з багатьма іншими відкриттями, створенню та розвитку штучних нейронних мереж сприяло дослідження принципів роботи цілком природних речей, а саме мозку, вивчення якого й дало поштовх для спроби створення його цифрового аналогу. Здавалося б, нейрофізіологія та математика лежать на різних краях дослідницьких інтересів, але занурення в одну область спонукало залучення іншої для глибшого розуміння процесів. Тож у середині минулого сторіччя було вперше застосовано поняття “штучна нейронна мережа” та навіть запропоновано перший алгоритм навчання. Отже, тема не зовсім нова, але з розвитком технологій дослідження в цій галузі набули іншого звучання та успішно продовжують розвиватися [1].

На початкових етапах розвитку нейронних мереж головною проблемою була підготовка даних для навчання нейромереж, зокрема структурування інформації в базах даних та їх попередня обробка. За відсутності Інтернету та більш потужної обчислювальної техніки вченим доводилося обробляти великі обсяги даних вручну, що вимагало значних витрат часу та супроводжувалось технічними помилками через участь людини у цьому процесі. Зараз через значний прогрес у розвитку потужности засобів обробки та автоматизацію самого процесу передпідготовки даних часові витрати на це значно скоротилися, хоча у разі роботи з дійсно великими масивами процес все ще вимагає значних ресурсів.

Вже в нашому столітті відбулось формування алгоритмів глибинного навчання, яке знайшло своє місце у багатьох системах. Формування багатошарових нейромереж та прорив

в обробці зображень дозволив науковцям використовувати нейромережі для розпізнавання та генерації образів.

Сьогодні сучасні системи, які використовують штучний інтелект, можуть вводити в оману навіть досвідчених людей та конкурувати з ними у різних нішах, як-от література, мистецтво тощо, але важливим фактом залишається відсутність унікальності таких творів та критичного мислення у машини. Дійсно, що не так давно, новела, написана штучним інтелектом (йому було задано і дійових осіб, і навіть речення), потрапила до фіналу конкурсу, в якому брали участь твори, написані людьми. Як виявилось, члени журі навіть не здогадувались, що новела була створена машиною, а дехто навіть відзначив чітку структуру [2].

Для спрощення створення та роботи з нейронними мережами були розроблені відповідні бібліотеки, які дозволи дослідникам не описувати всі процеси з нуля, а спиратись на здобутки попередників і просуватись вперед. Звичайно усе починалося з досить простих варіантів, але з часом вони поповнювались та оновлювались відповідно з отриманими результатами, що дозволило прискорити просування в цьому напрямку.

Однією з перших бібліотек стала Scikit-learn, яка була реалізована для мови програмування Python і надавала функціональність для створення та тренування різноманітних алгоритмів класифікації, регресії та кластеризації, таких як лінійна регресія, random forest, градієнтний бустинг. На початку над її створенням працював Девід Корнапе, який і створив першу версію, пізніше над нею працював Метью Брюхер, який зробив це частиною своєї дипломної роботи, а потім над цією бібліотекою почали працювати цілі команди спеціалістів [3]. Завдяки їх роботі протягом багатьох років було отримано Scikit-learn у тому вигляді, в якому її можна побачити зараз, і яка користується популярністю серед користувачів при реалізації машинного навчання.

Можна знайти багато прикладів використання бібліотеки Scikit-learn, що розміщені на офіційному сайті, де надається не тільки опис роботи і фінальний результат, а й доступні для усіх користувачів коди, за якими вони можуть розгорнути відповідні приклади на власному пристрої та протестувати їхню роботу. В одному з розділів наведено використання функції класифікації, а саме продемонстровано можливість розпізнавання рукописних чисел від 0 до 9. Результат виявляється позитивним і програма розпізнає усі числа, які попередньо задаються в форматі зображень з розміром 8x8 пікселів (рис. 1) [4].

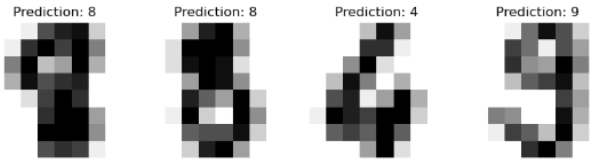


Рисунок 1. Результат класифікації зображень реалізований за допомогою бібліотеки Scikit-learn

Трохи пізніше з’явилася бібліотека TensorFlow, яка стала корисним інструментом у процесі створення та навчання різних моделей, серед яких знаходяться які прості лінійні регресії, так і складні нейронні мережі [5]. Вона, як і Scikit-learn, написана мовою програмування Python, але також має інтерфейс та реалізації для інших мов, таких як C++, Java, JavaScript, R тощо. Головними перевагами у використанні саме цієї бібліотеки у порівнянні з іншими є надійна обчислювальна потужність, а також наявність великої спільноти. Це привертає увагу багатьох людей перед початком роботи над власним проєктом, оскільки з’являється можливість отримати поради від більш досвідчених користувачів.

Бібліотека TensorFlow широко використовується для машинного навчання та є необхідною при побудові нейронної мережі, яка має шукати аномалії в журналах подій. Це

важлива складова саме під час процесу навчання моделі, щоб після завершення цього процесу з високою вірогідністю можна було виявляти аномалії в наданих їй журналах. В залежності від кількості даних, на основі яких відбувається навчання, процес може займати різний час, при невеликих об'ємах даних процес займає 39 секунд (рис. 2) [6].

```
[10]
✓ 39 с
    model.fit(x_train, y_train, epochs=5)

Epoch 1/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 8s 4ms/step - accuracy: 0.8618 - loss: 0.4853
Epoch 2/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 8s 4ms/step - accuracy: 0.9550 - loss: 0.1523
Epoch 3/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 8s 4ms/step - accuracy: 0.9676 - loss: 0.1041
Epoch 4/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 7s 4ms/step - accuracy: 0.9736 - loss: 0.0869
Epoch 5/5
1875/1875 ━━━━━━━━━━━━━━━━━━━━ 8s 4ms/step - accuracy: 0.9762 - loss: 0.0751
<keras.src.callbacks.history.History at 0x79bc74a33680>
```

Рисунок 2. Приклад процесу навчання моделі

Ще однією цікавою бібліотекою є PyTorch, яка була розроблена на рік пізніше за TensorFlow, а саме у 2016 році компанією Meta. Це чудовий приклад того, що можна взяти іншу бібліотеку та зробити її значно кращою. У цьому випадку PyTorch була побудована на основі Torch, яка була бібліотекою для машинного навчання, а внаслідок було отримано прискорення процесів створення прототипів дослідження до їх розгортання у виробничому середовищі [7]. Таким чином вдалося реалізувати бібліотеку з корисними інструментами для налагодження машинного навчання (рис. 3), а також перевагою стала наявність великої кількості вже попередньо навчених моделей нейронних мереж.

```
import os
import torch
from torch import nn
from torch.utils.data import DataLoader


1 usage
class NeuralNetwork(nn.Module):
    def __init__(self):
        super().__init__()
        self.flatten = nn.Flatten()
        self.linear_relu_stack = nn.Sequential(
            nn.Linear(28*28, out_features: 512),
            nn.ReLU(),
            nn.Linear( in_features: 512, out_features: 512),
            nn.ReLU(),
            nn.Linear( in_features: 512, out_features: 10),
        )


    def forward(self, x):
        x = self.flatten(x)
        logits = self.linear_relu_stack(x)
        return logits
```

Рисунок 3. Приклад побудови нейронної мережі за допомогою бібліотеки PyTorch

Головною перевагою усіх цих бібліотек є те, що вони мають відкритий код, тобто будь-хто може його переглядати та змінювати у відповідності до своїх потреб. Це зокрема дуже зручно як для студентів та науковців при проведенні досліджень та написанні наукових робіт.

Існує ще багато інших бібліотек, які також поліпшили процес створення та розгортання нейронних мереж, деякі з них отримали покращення та досі використовуються,

а деякі стали основою для формування вже нових та більш сучасних інструментів. Самі бібліотеки, як і нейронні мережі, розвивалися не один рік та продовжать свій розвиток з урахуванням вже нових здобутків. Надалі перед спеціалістами будуть ставати вже нові завдання з удосконалення цих систем, оскільки під час їх роботи стає зрозуміло, що не усі нюанси було враховано і виявилась необхідність у їх усуненні.

Література

[1] Нейромережі: що це таке і де застосовується [Електронний ресурс]. – Режим доступу до ресурсу: https://maxnet.ua/blog/neyroseti-chto-eto-takoye-i-gde-primenyayetsya/

[2] День, коли комп'ютер напише роман [Електронний ресурс]. – Режим доступу до ресурсу: https://www.golos.com.ua/article/266908

[3] Scikit-learn [Електронний ресурс]. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Scikit-learn

[4] Recognizing hand-written digits [Електронний ресурс]. – Режим доступу до ресурсу: https://scikit-learn.org/1.5/auto_examples/classification/plot_digits_classification.html

[5] TensorFlow що це і які основні можливості застосування [Електронний ресурс]. – Режим доступу до ресурсу: https://foxminded.ua/tensorflow-shcho-tse/

[6] TensorFlow 2 quickstart for beginners [Електронний ресурс]. – Режим доступу до ресурсу: https://www.tensorflow.org/tutorials/quickstart/beginner

[7] Open Source Neural Network Libraries [Електронний ресурс]. – Режим доступу до ресурсу: https://www.baeldung.com/cs/ml-open-source-libraries

РОЗРОБКА КЛІЄНТ-СЕРВЕРНИХ ЗАСТОСУНКІВ НА TYPESCRIPT ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ

Матієнко А.П.
Керівник: Латанська Л.О.
E-mail: liudmyla.latanska@nuos.edu.ua
Миколаїв, Національний університет кораблебудування імені адмірала Макарова

Сучасні програмні системи дедалі частіше реалізуються у вигляді клієнт-серверних застосунків, що функціонують у розподіленому мережевому середовищі та обслуговують велику кількість користувачів, працюючи з конфіденційними даними, що зумовлює високі вимоги до якості, надійності та безпеки програмного забезпечення.

Однією з ключових мов програмування для створення сучасних клієнт-серверних веб-застосунків є TypeScript – надмножина JavaScript зі статичною типізацією та механізмами перевірки типів на етапі компіляції. Згідно з дослідженням [1], використання TypeScript дозволяє зменшити кількість помилок, пов'язаних із невідповідністю типів, та покращити підтримуваність коду у великих проєктах.

TypeScript використовується як у клієнтській частині застосунків, так і на серверній стороні на платформі Node.js, що дає змогу створювати єдиний технологічний стек. Аналіз екосистеми TypeScript виявляє наявність специфічних дефектів у типових конфігураціях, асинхронній обробці даних та взаємодії з нетипізованими бібліотеками JavaScript [2].

Розробка сучасних клієнт-серверних застосунків на TypeScript характеризується необхідністю забезпечення безпеки типів на всіх рівнях архітектури – від взаємодії з базами даних до API-інтерфейсів та клієнтської логіки. Особливу увагу слід приділяти правильній типізації асинхронних операцій, обробці помилок та валідації даних, що надходять від зовнішніх джерел. Дослідження показують, що найбільша кількість дефектів виникає саме в місцях інтеграції TypeScript-коду з нетипізованими бібліотеками та зовнішніми сервісами [2].

Процес розробки типових клієнт-серверних застосунків включає проєктування REST або GraphQL API, реалізацію бізнес-логіки на серверній стороні, створення інтерактивних користувацьких інтерфейсів та забезпечення надійного обміну даними між компонентами системи. При цьому використання єдиної мови програмування для frontend та backend

компонентів сприяє підвищенню продуктивності команди та можливості повторного використання коду, зокрема типових визначень та бізнес-логіки валідації.

В останні роки суттєвого поширення набули інструменти розробки програмного забезпечення на основі штучного інтелекту, здатні автоматично генерувати фрагменти програмного коду, виконувати рефакторинг та пропонувати рішення для типових задач. Емпіричні дослідження свідчать про підвищення продуктивності розробників при використанні AI-асистентів [3, 4].

Водночас дослідження pull request'ів, створених за участю AI-агентів, демонструють проблеми зі складністю інтеграції змін, зростанням кількості правок під час код-рев'ю та появою шаблонного коду [5, 6], що свідчить про необхідність додаткового контролю якості.

Аналіз якості та безпеки коду, згенерованого AI-інструментами, показує, що такий код може містити більше вразливостей, пов'язаних з некоректною валідацією вхідних даних та помилками доступу до ресурсів [7, 8]. Незважаючи на переваги статичної типізації TypeScript, значна частина проблем виникає на межі типізованого та нетипізованого коду [9].

Практичний досвід показує, що найбільші переваги AI-інструментів досягаються при рефакторингу, написанні тестів та документації. При цьому понад 45% AI-генерованих pull request'ів потребують додаткових правок [10]. Актуальним є вдосконалення методик контролю метрик програмного коду з урахуванням особливостей AI-генерації, поєднання статичного аналізу, безпекових перевірок та обов'язкового ручного код-рев'ю.

Використання технологій штучного інтелекту в значній мірі змінює підходи до створення програмного коду і продовжує набирати популярність. Але доля рядків, що свідомо створені людиною-програмістом і належним чином оцінені автором, як особою, що несе професійну відповідальність, має тенденцію до зниження. Зокрема, поєднання TypeScript та інструментів штучного інтелекту відкриває нові можливості для підвищення ефективності розробки клієнт-серверних застосунків, однак вимагає підвищеної уваги до контролю якості, метрик коду та інформаційної безпеки програмних систем. Перспективним напрямком є розробка спеціалізованих методик верифікації AI-генерованого коду з урахуванням особливостей статичної типізації TypeScript.

Література

[1] Understanding TypeScript [Electronic resource]. – Resource access mode: https://files.sdiarticle5.com/wp-content/uploads/2025/01/Revised-ms_AJRCOS_129430_v1.pdf

[2] From Logic to Toolchains: An Empirical Study of Bugs in the TypeScript Ecosystem [Electronic resource]. – Resource access mode: https://arxiv.org/abs/2601.21186

[3] Intuition to Evidence: Measuring AI's True Impact on Developer Productivity [Electronic resource]. – Resource access mode: https://arxiv.org/pdf/2509.19708

[4] Developer Productivity With and Without GitHub Copilot [Electronic resource]. – Resource access mode: https://arxiv.org/pdf/2509.20353

[5] Evolving with AI: A Longitudinal Analysis of Developer Logs [Electronic resource]. – Resource access mode: https://arxiv.org/pdf/2601.10258

[6] Echoes of AI: Investigating the Downstream Effects of AI Assistants on Software Maintainability [Electronic resource]. – Resource access mode: https://arxiv.org/abs/2507.00788

[7] Assessing the Quality and Security of AI-Generated Code [Electronic resource]. – Resource access mode: https://arxiv.org/pdf/2508.14727

[8] Human-Written vs. AI-Generated Code: A Large-Scale Study of Defects, Vulnerabilities, and Complexity [Electronic resource]. – Resource access mode: https://arxiv.org/abs/2508.21634

[9] Security Vulnerabilities in AI-Generated Code: A Large-Scale Analysis of Public GitHub Repositories [Electronic resource]. – Resource access mode: https://arxiv.org/abs/2510.26103

[10] On the Use of Agentic Coding: An Empirical Study of Pull Requests on GitHub [Electronic resource]. – Resource access mode: https://arxiv.org/abs/2509.14745

ПОРІВНЯЛЬНА ХАРАКТЕРИСТИКА ФРЕЙМВОРКІВ ASP.NET CORE ТА SPRING BOOT ДЛЯ РОЗРОБКИ ВЕБЗАСТОСУНКІВ

Мінаєв А.І., Латанська Л.О.
E-mail: minaiev.andrii@gmail.com, liudmyla.latanska@nuos.edu.ua
Миколаїв, Національний університет кораблебудування імені адмірала Макарова

Сучасна веб-розробка потребує використання ефективних інструментів для швидкої та якісної реалізації програмного забезпечення, здатного відповідати вимогам продуктивності, безпеки та масштабованості. У цьому контексті важливим є обгрунтований вибір фреймворку, який спростить процес розробки, розгортання та супроводу програмних систем. Метою даної роботи є порівняльний аналіз фреймворків ASP.NET Core та Spring Boot і визначення особливостей їх використання для створення сучасних вебзастосунків.

Протягом тривалого часу одним із провідних рішень у веб-розробці в Java-екосистемі залишається фреймворк Spring, що значною мірою пояснюється кросплатформеністю мови Java завдяки використанню JVM. Проте дедалі частіше розробники обирають його розширення – Spring Boot, яке суттєво спрощує налаштування та пришвидшує процес розробки. Водночас класична платформа .NET була орієнтована на операційну систему Windows. Ситуація змінилася з виходом .NET Core у 2016 році, коли платформа стала open-source та кросплатформеною. Разом із нею з'явився фреймворк ASP.NET Core, що характеризується високою продуктивністю, зручною підтримкою API та тісною інтеграцією з інструментами Microsoft. Також слід відзначити стрімкий розвиток мови C#, яка є основною мовою програмування для ASP.NET Core та надає сучасні синтаксичні та функціональні можливості [1, 2].

Для узагальненого порівняння ключових характеристик фреймворків ASP.NET Core та Spring Boot у таблиці 1 представлено їх основні параметри [3].

Таблиця 1 – Основні параметри ASP.NET Core та Spring Boot

Параметри	ASP.NET Core	Spring Boot
Основна мова програмування	C#	Java
Додаткові мови	F#, VB.NET	Kotlin, Groovy
Екосистема доступу до БД	Entity Framework Core, Dapper	Spring Data
Основний інструмент збірки	MSBuild, dotnet CLI	Maven, Gradle
Кросплатформеність	Так (Windows, Linux, macOS)	Так (через JVM)
Open-source	Так	Так
Основне призначення	Вебзастосунки, Web API та мікросервіси	Вебзастосунки, REST API та мікросервіси

Проаналізувавши наведені дані, можна зробити висновок, що ASP.NET Core та Spring Boot є ефективними фреймворками для розробки вебзастосунків різної архітектурної складності. Вибір фреймворку значною мірою залежить від обраної технологічної платформи, досвіду команди розробників та вже наявної інфраструктури [3].

Література

[1] Microsoft. ASP.NET Documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://learn.microsoft.com/en-us/aspnet/core/

[2] Oracle. Java Documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://docs.oracle.com/en/java/

[3] ASP.NET Core vs. Java Spring Boot: A Comprehensive Comparison [Електронний ресурс]. – Режим доступу до ресурсу: https://www.linkedin.com/pulse/aspnet-core-vs-java-spring-boot-comprehensive-george-lebbos/

PYTHON ТА SQL ЯК УНІВЕРСАЛЬНІ ІНСТРУМЕНТИ ДЛЯ АНАЛІТИКИ ДАНИХ

Міхєєв І.А., Столяренко Т.Л.
E-mail: ivan.mikhieiev@hneu.net, tetiana.stoliarenko@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

У сучасному бізнес-середовищі прийняття ефективних рішень неможливе без якісної аналітики даних. Зростання обсягів інформації, яку генерують компанії в ході своєї діяльності, вимагає використання потужних та гнучких інструментів для її обробки, аналізу та візуалізації. Серед таких інструментів особливе місце займають мова програмування Python та мова структурованих запитів SQL. Їх поєднання дозволяє вирішувати широкий спектр завдань у сфері бізнес-аналітики, забезпечуючи глибоке розуміння процесів, тенденцій і можливостей для розвитку компанії.

Python – це високорівнева мова програмування, яка відзначається простотою синтаксису, широкою екосистемою бібліотек та активною спільнотою. Для бізнес-аналітики Python пропонує потужні інструменти для роботи з даними:

– бібліотеки pandas [1] та numpy [2] дозволяють ефективно обробляти та аналізувати великі масиви інформації;

– matplotlib [3] та seaborn [4] – створювати наочні графіки та діаграми;

– scikit-learn – застосовувати алгоритми машинного навчання для прогнозування та класифікації.

Python також ідеально підходить для автоматизації рутинних завдань, створення скриптів для збору, очищення та трансформації даних, що значно підвищує продуктивність аналітиків.

SQL (Structured Query Language) – це стандартна мова для роботи з реляційними базами даних, які є основним джерелом зберігання бізнес-інформації. За допомогою SQL можна швидко отримувати потрібні дані, виконувати складні вибірки, об’єднувати таблиці, агрегувати показники та будувати звіти. SQL дозволяє працювати з великими обсягами інформації, забезпечуючи високу швидкість та точність обробки запитів. Для аналітиків даних знання SQL є обов’язковим, оскільки більшість корпоративних даних зберігається саме у реляційних базах.

Поєднання Python та SQL відкриває нові можливості для аналітики даних. Аналітик може використовувати SQL для отримання необхідних даних із бази, а потім обробляти, аналізувати та візуалізувати їх у Python. Такий підхід дозволяє будувати комплексні аналітичні рішення, створювати інтерактивні дашборди, автоматизувати звітність та впроваджувати моделі прогнозування. Наприклад, у сфері продажів можна автоматично збирати дані про транзакції, аналізувати динаміку попиту, виявляти ключові фактори впливу на виручку та прогнозувати майбутні результати.

Використання Python та SQL у бізнес-аналітиці сприяє підвищенню якості прийняття рішень, оптимізації процесів та виявленню нових можливостей для розвитку компанії. Ці інструменти є універсальними, масштабованими та затребуваними на ринку праці, що робить їх вивчення важливим етапом підготовки сучасних фахівців з аналітики даних.

Література

[1] Pandas documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://pandas.pydata.org/docs/index.html

[2] NumPy documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://numpy.org/doc/stable/

[3] Matplotlib documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://matplotlib.org/stable/

[4] Seaborn: statistical data visualization [Електронний ресурс]. – Режим доступу до ресурсу: https://seaborn.pydata.org/

[5] Scikit-learn: Machine Learning in Python [Електронний ресурс]. – Режим доступу до https://scikit-learn.org/stable/

CKAN ЯК ІНФРАСТРУКТУРНА ПЛАТФОРМА ВІДКРИТИХ ДАНИХ

Моторнюк С.О.
Керівник: Старкова О.В.
E-mail: tonnydexter@gmail.com
Харків, Харківський національний економічний університет імені Семена Кузнеця

Відкриті дані стали базовою складовою цифрової трансформації держави, міських сервісів, науки та бізнесу. Ключові виклики їх системного використання - це каталогізація й пошук, забезпечення інтероперабельності, керування якістю, дотримання політик ліцензування та приватності, а також сталі процеси публікації й оновлення. CKAN - зріла відкрита платформа (AGPLv3) для побудови каталогів даних, яка виступає «хребтом» порталу відкритих даних: уніфікує метадані, надає API для інтеграцій та забезпечує масштабовану індексацію й доступ до ресурсів [1].

Ядро CKAN оперує базовими сутностями: набір даних (dataset), ресурс (файл чи API), організації, групи, теги та словники термінів. Платформа надає JSON API для створення, оновлення, пошуку та експорту метаданих і ресурсів [2]. Індексація та пошук реалізовані через Apache Solr із фасетною навігацією; існують інтеграції з Elasticsearch через розширення. Модель аутентифікації/авторизації підтримує ролі в межах організацій, приватні набори та запрошення користувачів; для корпоративних і державних інсталяцій доступні плагіни SSO на базі SAML/OIDC [1].

Для табличних даних CKAN має DataStore (PostgreSQL), який дозволяє робити фільтрацію та запити до даних через API без завантаження файлів. Масове завантаження з CSV/XLS(X) у DataStore забезпечують інструменти DataPusher (базовий) та XLoader (для великих файлів і надійніших пайплайнів) [1; 4]. Зберігання файлів (FileStore) може бути локальним чи у сумісних об’єктних сховищах (S3/MinIO) через відповідні плагіни. Підтримується Harvesting - агрегація метаданих з інших каталогів CKAN, джерел DCAT/CSW/INSPIRE тощо [5]. Локалізація, темізація та можливість «headless»-підходу (власний фронтенд поверх API) роблять платформу придатною як для швидких пілотів, так і для кастомних вітрин [1].

Типові сценарії використання:

– державні портали відкритих даних. Централізований каталог, керування правами публікації для розпорядників, експорт/імпорт за стандартами DCAT/DCAT-AP, планове оновлення даних, звітність щодо якості та доступності ресурсів [6; 7].

– міські портали (Smart City). Публікація транспортних, екологічних, бюджетних даних; інтеграція геосервісів (WMS/WFS), попередній перегляд карт і геопоиск (через ckanext-spatial та GeoServer) [8].

– наукові/університетські каталоги. Розширені схеми метаданих, валідація якості (Frictionless), зв’язок із ідентифікаторами (DOI/ORCID), різні ліцензії повторного використання [9; 10].

– корпоративні каталоги даних. Внутрішня інвентаризація джерел для аналітики/ML, контроль доступу, аудит публікацій, інтеграція з пайплайнами ETL/ELT.

– мета-портали/агрегатори. Harvesting із десятків джерел (CKAN, DCAT, CSW), уніфікація метаданих, нормалізація словників і якісні карти відповідності.

Відкритість архітектури CKAN - один із ключових факторів її поширення. Найуживаніші категорії модифікації вбудованого функціоналу:

– метадані та інтероперабельність: ckanext-scheming для опису та валідацій кастомних форм метаданих; ckanext-dcat для імпорту/експорту DCAT/DCAT-AP; ckanext-spatial - підтримка ISO 19115/INSPIRE, геоіндексація, попередній перегляд просторових даних [7–9].

– завантаження/якість: ckanext-xloader (масові імпорти у DataStore), ckanext-validation (перевірка схем Frictionless), ckanext-archiver і ckanext-qa (перевірка доступності ресурсів, оцінка якості) [4; 10].

– ідентифікація/SSO: ckanext-saml2, ckanext-oidc-pkce - інтеграція з Keycloak, Azure AD та іншими постачальниками ідентичностей.

– зберігання: ckanext-files/ckanext-cloudstorage/ckanext-s3filestore - робота з S3-сумісними сховищами, політики зберігання великих файлів.

– контент і UX: ckanext-pages (статичні сторінки), ckanext-showcase (вітрини кейсів повторного використання), ckanext-hierarchy (ієрархії організацій/категорій).

– аналітика/моніторинг: інтеграції з Matomo/GA, збір статистики API/завантажень.

Harvesting: ckanext-harvest із численними конекторами (CKAN, DCAT, CSW, INSPIRE) [5]. Кожне розширення має власну зрілість та активність мейнтейнерів, тому під продакшн-інсталяції доцільно проводити аудит сумісності з версією CKAN і планом оновлень [3].

Платформа CKAN використовується численними національними та муніципальними порталами у світі. До відомих прикладів належить Портал відкритих даних України (data.gov.ua), а також низка європейських і міських ініціатив, описаних у докладних кейсах/шоукейсах спільноти CKAN [11]. Окремі наукові та галузеві проєкти (екологічний моніторинг, транспортні дані) використовують розширені схеми метаданих і геопросторові модулі для забезпечення сумісності з OGC/INSPIRE [8; 11]. Практика показує, що успішні інсталяції поєднують базове ядро CKAN з 5–10 критичними розширеннями, налаштованими під доменні потреби.

Серед переваг платформи:

– зріле ядро, активна спільнота, розширюваність і багата екосистема плагінів [1; 3];

– відповідність стандартам інтероперабельності (DCAT/DCAT-AP, ISO/INSPIRE через розширення) [6-7; 9];

– прозора архітектура та відсутність vendor lock-in завдяки ліцензії AGPLv3 [12];

– можливість масштабування: окремі сервіси для БД (PostgreSQL/DataStore), пошуку (Solr), кешів (Redis), сховищ об’єктів, із гнучкою контейнеризацією. Обмеження: потреба в DevOps-компетенціях (налаштування Solr, постгрес-реплікацій/беккапів, політик зберігання), базова тема інтерфейсу вимагає кастомізації для сучасного UX, а також те, що CKAN не є сховищем потокових/стрімінгових даних - для цього потрібні окремі сервіси з інтеграцією через API.

Під час вибору платформи доцільно оцінювати: архітектуру розгортання, функціональність каталогу, інтероперабельність, розширюваність, безпеку/IAM, UX/візуалізації, продуктивність/масштабування, TCO та ризики lock-in. Порівняльний огляд альтернатив:

– uData. Легша у старті FOSS-платформа з акцентом на DCAT-сумісність, просте ядро та зрозумілий UX. Має плагіни й можливості темізації, але екосистема розширень та покриття специфічних сценаріїв (наприклад, розгалужені схеми метаданих чи складний harvesting) зазвичай поступається CKAN за різноманіттям [13]. Рекомендована для невеликих або середніх каталогів, де критична швидкість запуску та стандартизований DCAT-обмін.

– DKAN. Рішення на базі Drupal, що привабливе в екосистемах, де вже є компетенції з Drupal (контент-менеджмент, темізація, модулі). Сильні сторони - інтеграція з наявними процесами CMS і розширеннями Drupal; слабкі - можлива залежність від дистрибутиву, інша модель розробки плагінів порівняно з CKAN і специфічні витрати на інтеграцію з DataStore-подібною аналітикою [14].

– Комерційні SaaS (Socrata/Tyler, ArcGIS Hub, OpenDataSoft). Переваги: швидкий старт, керована інфраструктура, вбудовані візуалізації/дашборди, SLA-підтримка. Обмеження: вищий TCO у довгостроковій перспективі, ризики vendor lock-in (експорт не завжди покриває всі розширені артефакти), обмежена гнучкість у нестандартних сценаріях чи специфічних стандартах метаданих [14]. Рекомендовано, коли пріоритет - time-to-value та гарантована підтримка, а бюджет/політики дозволяють залежність від вендора.

– CKAN у цьому контексті. Доцільний вибір, коли потрібна гнучкість, контроль над інфраструктурою, широка екосистема розширень, підтримка стандартів і активна спільнота. У великих організаціях дає змогу поступово нарощувати функціональність (SSO, harvesting, геомодулі, валідація) без зміни платформи [1; 3–9].

CKAN - зрілий та гнучкий фундамент для порталів відкритих даних, який поєднує потужні можливості каталогу, масштабований пошук, табличне сховище з API-доступом і широку екосистему розширень. Саме ця розширюваність дозволяє адаптувати платформу під державні, міські, наукові й корпоративні сценарії, зберігаючи відповідність ключовим відкритим стандартам. Порівняльний аналіз свідчить: за наявності DevOps-компетенцій CKAN забезпечує найкращий баланс між функціональністю, контролем над інфраструктурою та TCO; альтернативи (uData, DKAN, SaaS) доречні в нішах швидкого старту, інтеграції з наявними CMS або за умов пріоритету керованих сервісів. Запропонована методика пілотування допомагає прийняти обґрунтоване рішення, мінімізуючи ризики міграції в майбутньому.

Література

[1] CKAN Documentation [Електронний ресурс]. – Режим доступу: https://docs.ckan.org/

[2] CKAN API Guide [Електронний ресурс]. – Режим доступу: https://docs.ckan.org/en/latest/api/

[3] CKAN Extensions (ckanext) registry and GitHub organization [Електронний ресурс]. – Режим доступу: https://extensions.ckan.org/ ; https://github.com/ckan

[4] ckanext-xloader and DataPusher [Електронний ресурс]. – Режим доступу: https://github.com/ckan/ckanext-xloader ; https://github.com/ckan/datapusher

[5] ckanext-harvest documentation [Електронний ресурс]. – Режим доступу: https://github.com/ckan/ckanext-harvest

[6] W3C DCAT Recommendation [Електронний ресурс]. – Режим доступу: https://www.w3.org/TR/vocab-dcat-3/

[7] ckanext-dcat documentation [Електронний ресурс]. – Режим доступу: https://github.com/ckan/ckanext-dcat

[8] ckanext-spatial (including ISO/INSPIRE) documentation [Електронний ресурс]. – Режим доступу: https://github.com/ckan/ckanext-spatial

[9] ckanext-scheming documentation [Електронний ресурс]. – Режим доступу: https://github.com/ckan/ckanext-scheming

[10] Frictionless Data specs and ckanext-validation [Електронний ресурс]. – Режим доступу: https://specs.frictionlessdata.io/ ; https://github.com/frictionlessdata/ckanext-validation

[11] CKAN showcase and case studies (examples of national/municipal portals) [Електронний ресурс]. – Режим доступу: https://ckan.org/showcase/

[12] GNU Affero General Public License v3 (AGPLv3) [Електронний ресурс]. – Режим доступу: https://www.gnu.org/licenses/agpl-3.0.en.html

[13] uData documentation. URL: https://udata.readthedocs.io/; Project site [Електронний ресурс]. – Режим доступу: https://www.data.gouv.fr/en/udata/

[14] DKAN documentation and vendor materials; SaaS platforms (Socrata/Tyler, ArcGIS Hub, OpenDataSoft) [Електронний ресурс]. – Режим доступу: https://getdkan.org/ ; https://www.tylertech.com/solutions/transformative-technology/socrata ; https://hub.arcgis.com/ ; https://www.opendatasoft.com/

МОДЕЛЬ ОЦІНКИ КОМПЕТЕНТНОСТЕЙ УЧАСНИКІВ РОЗПОДІЛЕНИХ КОМАНД ІТ-ПРОЄКТІВ

Назаров Д. Л., Старкова О. В.
E-mail: olha.starkova@hneu.net
Харків, Харківський національний економічний університету імені Семена Кузнеця

У сучасній науковій літературі представлено низку досліджень, присвячених компетентностям IT_фахівців, де структуруються технічні (hard), соціальні (soft) та персональні навички залежно від ролей у команді, а також аналізуються особливості компетенцій у глобальних та розподілених програмних командах [1; 2; 4; 6]. Окремі роботи підкреслюють зростання ролі soft skills для Agile- і Scrum-команд, зокрема комунікації, співпраці, адаптивності та емоційного інтелекту в умовах географічної розподіленості та культурного різноманіття [3; 5; 11]. Водночас більшість існуючих моделей або зосереджуються переважно на описі загального переліку компетентностей, або не враховують у повному обсязі специфіку віддалених та гібридних форматів роботи, а також не інтегрують в єдину систему оцінку hard skills, soft skills, психометричні показники та AI/ML-індикатори [1; 2; 7]. На тлі швидкого поширення розподілених та гібридних IT-команд це зумовлює потребу переходу від інтуїтивного до структурованого, компетентністно-базованого підходу до відбору й оцінювання учасників команд IT-проєктів, орієнтованого на прогнозування ефективності взаємодії в розподіленому середовищі [2; 4]. Метою доповіді є обґрунтування інтегрованої моделі оцінки компетентностей учасників розподілених команд IT-проєктів, яка поєднує шкали hard/soft skills, психометричні інструменти та AI/ML-методи аналізу [1; 7].

Авторами дослідження пропонується трирівнева модель оцінки компетентностей: блок hard skills (технічна компетентність за роллю у команді IT-проєкту); блок soft skills (collaborative index, що відображає здатність до співпраці в розподіленому середовищі); блок цифрових індикаторів, отриманих із використанням AI/ML-алгоритмів. Такий підхід узгоджується з сучасними моделями компетентностей для IT-фахівців, де технічні, соціальні та персональні компетентності виділяються як окремі, але взаємопов'язані групи [1; 2; 4; 6].

Оцінювання технічних компетентностей пропонується здійснювати за трьома підвимірами: загальний досвід у розробці ПЗ та досвід участі в розподілених IT-проєктах; володіння ключовими технологіями (мови програмування, фреймворки, хмарні сервіси, DevOps-інструменти) з урахуванням ролі в проєкті; предметно-орієнтована експертиза (fintech, e-commerce, healthcare тощо). Для кожного підвиміру використовується уніфікована порядкова шкала, наприклад 0–4: від відсутності практики до експертного рівня, що дозволяє формувати агрегований показник HS_score для конкретної ролі в команді. Дослідження компетенцій програмістів підтверджують ефективність таких структурованих профілів компетентностей для узгодження вимог ролі та наявних навичок фахівця [1; 2; 6].

У розподілених IT-командах soft skills безпосередньо впливають на якість комунікації, швидкість узгодження рішень і стійкість до стресу та невизначеності. Пропонується виділяти щонайменше чотири кластери: комунікація (ясність письмової та усної мови, вміння ставити уточнювальні питання), співпраця (командна робота, конструктивний фідбек, спільне вирішення проблем), адаптивність (готовність до змін, робота в різних часових поясах, самоменеджмент в умовах асинхронності), емоційний інтелект (емпатія, регуляція емоцій, підтримка психологічної безпеки). Для кожного кластера застосовується шкала 1–5 з описаними поведінковими індикаторами. На цій основі розраховується інтегральний показник collaborative index (CI_score) як зважена сума нормованих оцінок за кластерами, причому ваги можуть відрізнятися залежно від ролі (наприклад, вищі для тімлідів) [3; 4; 5; 11].

Комплексну оцінку компетентностей учасника розподіленої команди IT-проєкту пропонується розраховувати у вигляді інтегрального показника за виразом (1):

$$Total\_Competency\_Score = \alpha \cdot HS\_score + \beta \cdot CI\_score + \gamma \cdot AI\_score, \quad (1)$$

де α, β, γ – вагові коефіцієнти, які визначають відносну важливість технічних (HS_score), соціально-комунікативних (CI_score) та аналітично-цифрових індикаторів (AI_score).

Для суто технічних ролей доцільно встановлювати $\alpha > \beta \geq \gamma$, тоді як для керівних та координаційних ролей β може наближатися до α [2; 4]. Компонента AI_score формується на основі автоматизованого аналізу комунікації (NLP-оцінка відповідей, тону, послідовності аргументації), автоматизованої перевірки коду (code quality, тестування, безпека, стиль) та поведінкових моделей (результати сценарних симуляцій та ігрових завдань) [7; 9]. Гіпотеза дослідження може формулюватися так: інтегрована модель оцінки компетенцій, яка поєднує шкали hard skills (технічна експертиза), soft skills (collaborative index) та AI-базовані індикатори, забезпечує підвищення точності відбору учасників розподілених IT-команд щонайменше на 25–30% порівняно з традиційними методами відбору, що базуються переважно на оцінці резюме та неструктурованому інтерв'ю [7; 9].

Для підвищення об'єктивності оцінювання soft skills IT-фахівців доцільно інтегрувати валідовані психометричні тести (когнітивні тести, моделі особистості, мотиваційні опитувальники), адаптовані під специфіку віддаленої та гібридної роботи [6; 8]. Окремим перспективним напрямом є gamified assessments – ігрові та симуляційні завдання, які моделюють типові ситуації у розподілених командах (робота в умовах часових обмежень, конфлікт пріоритетів, асинхронна комунікація тощо). Дослідження показують, що gamified-підходи підвищують залученість кандидатів, скорочують час відбору та сприяють кращому виявленню поведінкових патернів, релевантних роботі в розподіленій команді [8].

Структуровані competency-based frameworks, доповнені психометричними й AI/ML-інструментами, дозволяють зменшити ризики помилкових рішень найму (mis-hires), підвищити відповідність кандидата вимогам розподіленої команди та скоротити час адаптації завдяки кращому попередньому узгодженню профілю компетенцій із завданнями проєкту [7; 9]. Практичне впровадження таких моделей у IT-компаніях створює основу для формування прозорих критеріїв відбору, планування індивідуального розвитку (upskilling та reskilling) і побудови предиктивної аналітики ефективності роботи розподілених команд [2; 4; 7]. Подальші дослідження доцільно спрямувати на емпіричну перевірку запропонованої гіпотези щодо зростання точності відбору, оптимізацію вагових коефіцієнтів запропонованої автором моделі для різних ролей та інтеграцію її показників у математичні моделі формування розподілених команд IT-проєктів [1; 2; 4; 6].

Література

[1] Assyne N. The essential competencies of software professionals // Information and Software Technology. – 2022.

[2] Hidayati A., Budiardjo E. Software Engineer Competencies in Global Software Development. – 2022.

[3] Hidayati A., Budiardjo E. Hard and Soft Skills for Scrum Global Software Development Teams. – 2019.

[4] Scrum Team Competence Based on Knowledge, Skills, Attitude in Global Software Development. – 2023.

[5] Soft Skills vs. Hard Skills and Their Importance to a Scrum Master // Big-Agile : вебсайт. – Режим доступу: https://big-agile.com (дата звернення: 27.01.2026).

[6] Professional competencies of future software engineers in the conditions of a mobile-oriented environment // Journal of Physics: Conference Series. – 2022.

[7] AI Talent Assessment: Evaluating Skills Accurately at Scale // InterWiz AI Blog : вебсайт. – 2025. – Режим доступу: https://interwiz.ai (дата звернення: 27.01.2026).

[8] Gamification in Psychometric Testing: Engaging Candidates and Improving Results // Psico-Smart Blog. – 2024. – Режим доступу: https://psico-smart.com (дата звернення: 27.01.2026).

[9] Ultimate Guide – The Best Talent Assessment Tools of 2025 // MokaHR Blog : вебсайт. – 2024. – Режим доступу: https://mokahr.io (дата звернення: 27.01.2026).

[10] Scrum/Agile Teams, Hard Skills and Results // ProcessGroup : вебсайт. – Режим доступу: https://processgroup.com (дата звернення: 27.01.2026).

COPERNICUS BROWSER ЯК ВЕБ-ІНСТРУМЕНТ ДЛЯ ВІЗУАЛІЗАЦІЇ ТА АНАЛІЗУ ДАНИХ ДИСТАНЦІЙНОГО ЗОНДУВАННЯ ЗЕМЛІ

Петриляк О. Р.
Керівник: Костенко С. Б.
E-mail: oleh.petryliak@gmail.com
Львів, Львівський національний університет імені Івана Франка

Європейська програма спостереження за Землею (Earth Observation Programme) Copernocus забезпечує відкритий доступ до великих обсягів супутникових зображень, однак практичне використання цих даних часто ускладнюється обсягом архівів, вимогами до інфраструктури, а також потребою у спеціалізованому ПЗ для пошуку та попередньої обробки. Copernicus Browser[1] – це веб-застосунком екосистеми Copernicus Data Space Ecosystem[2], який забезпечує зручний доступ до пошуку, перегляду, порівняння, аналізу та завантаження супутникових продуктів. Інструмент побудований на базі підходів Sentinel Hub EO Browser і орієнтований як на початківців, так і на користувачів з досвідом у дистанційному зондуванні.

Copernicus Browser надає доступ до даних супутників місій Sentinel, зокрема Sentinel-1 (радарні дані), Sentinel-2 (оптичні мультиспектральні дані), Sentinel-3 (океанографічні та атмосферні спостереження), Sentinel-5P (атмосферні гази та аерозолі). Типовий сценарій роботи включає вибір області дослідження на карті, встановлення часових меж, обмеження на хмарність (для оптичних даних) та вибір відповідної колекції. За цими параметрами виконується пошук доступних продуктів (наборів даних), після чого користувач може перейти до їх візуалізації (формування зображень з прив’язкою до географічних карт) та аналізу.

Рис. 1 – Приклад роботи в Copernicus Browser: вибір області на карті, часових меж і фільтрів для пошуку даних Sentinel

Ключовою перевагою Copernicus Browser є поєднання інструментів візуальної інтерпретації та елементарної аналітики без необхідності встановлення локального ПЗ. Користувач може застосовувати готові візуалізації (композити каналів, тематичні шари), створювати власні комбінації каналів, а також використовувати користувацькі скрипти на JavaScript для побудови індексів і тематичних візуалізацій. Підтримуються інструменти вимірювання відстаней і площ, завантаження зображень, формування таймлапсів, статистичний аналіз для точки або полігона, а також побудова часових рядів для обраної території. Практично це дозволяє швидко отримувати узагальнені характеристики,

наприклад середні значення індексу для ділянки, динаміку зміни показників за сезон або порівняння ситуації до та після певної події.

Як приклад використання Copernicus Browser розглянемо моніторинг стану рослинності за даними супутників місії Sentinel-2 L2A. Після вибору території, необхідного періоду та ввімкнути відображення шару NDVI (Normalized Difference Vegetation Index), що розраховується як певне співвідношенні інтенсивності відбиття сонячних променів від земної поверхні у червоному та ближньому інфрачервоному діапазонах, отримаємо карту, де відтінками зеленого будуть позначені ділянки з різним станом/видом рослинності. Якщо скористатися модулем Histogram, то Copernicus Browser обчислить середні значення та розподіл NDVI, а модулем Time Series – побудує графік зміни індексу в часі. Отримані дані використовують для попередньої оцінки фенологічних фаз (етапів розвитку рослинності), виявлення стресових зон та порівняння сезонів.

Аналогічно можна візуалізувати вологість ґрунту та оцінити зміни у ньому. Для цього використовують дані супутників місії Sentinel-1, що є радарними зображеннями, де кожен піксель показує інтенсивність відбитих від Землі електромагнітних хвиль, які були надіслані із супутників. Вологі та сухі грунти відбивають хвилі з різною інтенсивністю.

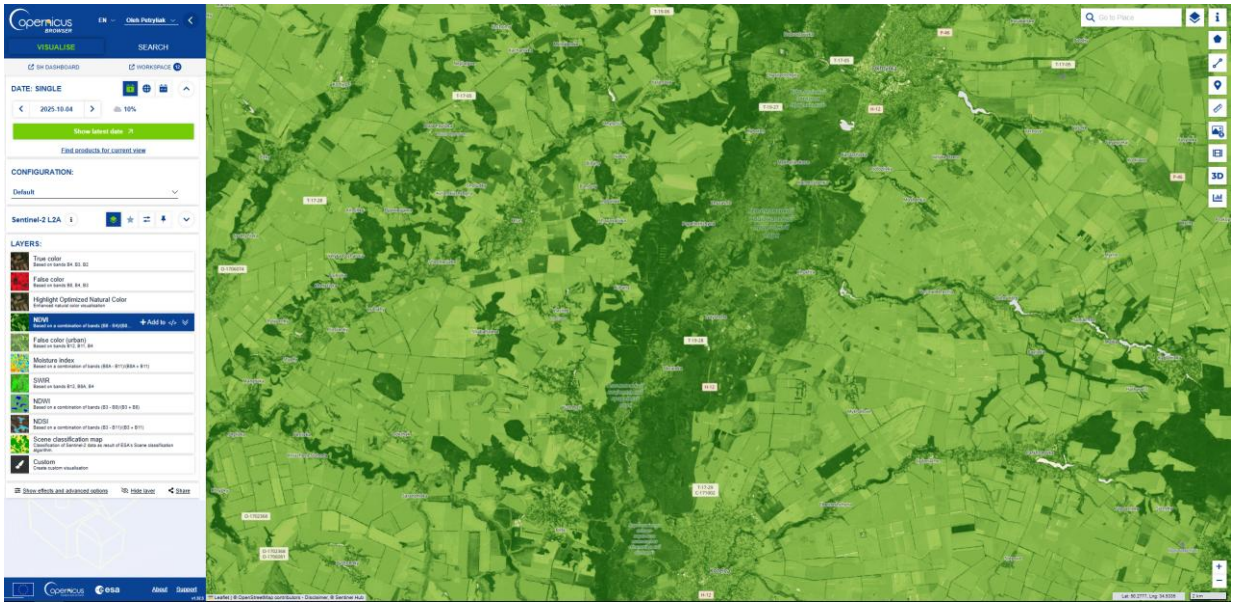


Рис. 2 – Результат відображення в Copernicus Browser шару NDVI за даними супутників місії Sentinel-2 для моніторингу рослинності

Copernicus Browser є безкоштовним для використання, проте доступ до всього функціоналу потребує реєстрації власного облікового запис. Вихідний код клієнтської частини Copernicus Browser є відкритим [3] (ліцензія MIT), що робить можливим його аудит, адаптацію та використання як навчальний приклад для веб-розробки геоінформаційних сервісів.

Таким чином, Copernicus Browser – це ефективний безкоштовний інструмент швидкої візуалізації супутникових даних поверхні Землі. Підтримка часових рядів, статистичних розрахунків і користувацьких скриптів забезпечує поєднання візуальної інтерпретації з кількісними оцінками та відтворюваними процедурами обробки даних.

Література

[1] Copernicus Data Space Ecosystem. About the Browser [Електронний ресурс]. – Режим доступу: https://documentation.dataspace.copernicus.eu/Applications/Browser.html

[2] Copernicus Data Space Ecosystem [Електронний ресурс]. – Режим доступу: https://dataspace.copernicus.eu/

[3] eu-cdse/copernicus-browser [Електронний ресурс]. – Режим доступу: https://github.com/eu-cdse/copernicus-browser

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ВИДІЛЕННЯ ПОВІДОМЛЕНЬ ПРО КАТАСТРОФИ

Піддубний Д.С., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

У цифровому світі соціальні мережі стали критично важливим каналом зв'язку під час надзвичайних ситуацій та катастроф. Мільйони користувачів у реальному часі повідомляють про події, що відбуваються, що створює безцінне джерело інформації для рятувальних служб, засобів масової інформації та благодійних організацій. Однак автоматичне визначення того, чи стосується повідомлення реальної катастрофи, чи є лише метафоричним висловлюванням, є складною задачею обробки природної мови. Основний виклик полягає в неоднозначності людської мови: слова, що описують катастрофу, часто використовуються у переносному значенні. Неправильна класифікація може призвести до фальшивих тривог або ігнорування реальної загрози.

У цій роботі представлено рішення задачі бінарної класифікації повідомлень: визначення, чи описує коротке текстове повідомлення реальну катастрофу (цільова мітка 1), чи ні (цільова мітка 0). Для цього розроблено комплексний конвеєр обробки тексту та класифікації на основі ансамблевої моделі Random Forest. Обраний підхід поєднує переваги TF-IDF векторизації для представлення тексту та стійкості Random Forest до перенавчання та шуму в даних.

Загальний алгоритм роботи програми складається з наступних ключових кроків:

– завантаження та первинний аналіз навчальних даних;

– попередня обробка тексту (лематизація, видалення стоп-слів);

– об'єднання текстових ознак (основний текст, ключове слово, локація) в єдине поле для аналізу;

– відділення навчальної та тестової вибірок;

– побудова та навчання конвеєра, що включає векторизацію TF-IDF та класифікатор;

– оцінка якості моделі на тестовій вибірці за допомогою метрик точності.

Основним елементом архітектури створеної моделі є конвеєр з двох кроків.

Перший крок – TfidfVectorizer, що перетворює текст (поле combined_features) у числову матрицю. TF-IDF (Term Frequency-Inverse Document Frequency) – це статистична міра, що оцінює важливість слова для документа в колекції. Використання біграм дозволяє моделі враховувати не лише окремі слова, а й поєднання (наприклад, "forest fire"), що може бути ключовим для виявлення контексту катастрофи.

Другий крок – RandomForestClassifier. Це ансамблевий алгоритм, який приймає рішення на основі множини дерев рішень.

Після навчання конвеєра на 80 % даних (X_train, y_train) його ефективність була перевірена на тестовій вибірці (20 % даних). Показник точності моделі склав 0.79.

Розроблено програмне забезпечення виділення повідомлень про катастрофи на основі застосування python-пакетів pandas [1] та scikit-learn [2]. Використання Random Forest у поєднанні з ретельною попередньою обробкою тексту (лематизація, видалення стоп-слів) та TF-IDF векторизацією біграм дозволяє створити модель, здатну розрізняти буквальний та метафоричний контекст слів з достатньою для практичного застосування точністю.

Таким чином, представлене програмне забезпечення служить основою для створення автоматизованих інструментів моніторингу соціальних мереж, спрямованих на оперативне виявлення надзвичайних ситуацій.

Література

[1] Pandas documentation [Electronic resource]. – Access mode: https://pandas.pydata.org/docs/index.html

[2] sklearn API [Electronic resource]. – Access mode: https://scikit-learn.org/stable/modules/classes.html

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ АВТОМАТИЧНОГО ВИЗНАЧЕННЯ АКОРДІВ З АУДІОФАЙЛІВ

Сазонова Н.О., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

Автоматичне визначення акордів із аудіосигналів має велику практичну користь зокрема для індексації музичних колекцій, автоматичного створення партитур, навчальних застосунків та систем рекомендацій. Ручна анотація великих масивів музичних записів є трудомісткою й дороговартісною, тому розробка програмних засобів, які дозволяють автоматично витягувати акордову інформацію з аудіосигналу, є актуальним завданням як у дослідницькому, так і прикладному контексті.

У роботі виконувалось створення програми для побудови класифікатора акордів, що включає завантаження аудіофайлів, виконання сегментації, визначення акустичних ознак, виконання класифікації. Такий підхід покликаний надати простий і відтворюваний шлях від сирих даних до моделі, придатної для подальшого тестування та інтеграції.

У реалізації програми автоматичного визначення акордів з аудіофайлів використані бібліотеки Python:

– librosa [1] для зчитування аудіоданих й обчислення ознак;

– numpy [2] для обробки масивів;

– xml.etree.ElementTree [3] для парсингу XML-анотацій;

– sklearn [4] для підготовки даних і навчання моделі.

Конфігураційні параметри створеного програмного прототипу включають частоту дискретизації SAMPLE_RATE у 22050 Гц, тривалість сегмента SEGMENT_DURATION у 2 секунди та кількість MFCC (параметр N_MFCC) у 20, що забезпечує баланс між інформативністю ознак і обчислювальною ефективністю.

Безпосередньо робота програми з визначення акордів з аудіофайлів починається з навчання моделі шляхом читання аудіосигналу та відповідного XML-файла, у якому послідовно витягуються акордові мітки шляхом обходу вузлів з подальшим формуванням текстових позначень у форматі «Нота Тип» (наприклад, «C major»). Аудіодані розбиваються на сегменти фіксованої довжини, для кожного сегмента обчислюються MFCC і їх агреговані статистики (середнє та дисперсія), що дозволяє представити фрагмент у вигляді вектора ознак. Отримані ознаки стандартизуються, мітки кодуються, а тоді виконується навчання класифікатора One-vs-Rest на основі логістичної регресії для багатокласового розпізнавання акордів.

Розроблена програма придатна для розпізнавання акордів і може бути використана як базова складова більшого процесу автоматичного аранжування або анотації музичних колекцій. Використання перевірених статистичних методів обробки ознак у поєднанні з лінійною класифікацією забезпечує високу швидкість обробки даних та стабільність результатів на чистих аудіозаписах. Однак у поточному вигляді реалізація має обмеження: нечітке часове вирівнювання між сегментами та XML-мітками, що може призводити до помилок у відповідності ознак і міток.

Література

[1] Tutorial – librosa 0.11.0 documentation [Electronic resource]. – Access mode: https://librosa.org/doc/latest/tutorial.html

[2] NumPy documentation [Electronic resource]. – Access mode: https://numpy.org/doc/stable/

[3] The ElementTree XML API [Electronic resource]. – Access mode: https://docs.python.org/3/library/xml.etree.elementtree.html

[4] sklearn API [Electronic resource]. – Access mode: https://scikit-learn.org/stable/index.html

РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ ПОШУКУ ОПТИМАЛЬНИХ ТРАНСПОРТНИХ МАРШРУТІВ

Третяк О.О., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

За поточного стану розвитку транспортної інфраструктури важливим завданням є швидке та обґрунтоване визначення оптимальних маршрутів між містами. Користувачам необхідно враховувати різні критерії вибору поїздки, зокрема вартість перевезення, тривалість перебування в дорозі та доступні види транспорту. Додатково складність задачі зростає через можливість комбінування різних транспортних засобів та необхідність формування маршрутів із пересадками. У ручному режимі такий аналіз є трудомістким і схильним до помилок, тому виникає потреба у створенні програмного забезпечення, яке автоматизує процес пошуку, розрахунку та порівняння маршрутів за різними критеріями.

У даній роботі розроблено програмну систему пошуку оптимальних транспортних маршрутів між містами на основі графової моделі. Транспортна мережа подається у вигляді орієнтованого графа, де вершинами виступають міста, а ребрами – можливі переміщення між ними. Кожне ребро містить набір параметрів, зокрема вартість поїздки, тривалість маршруту та тип транспорту. Такий підхід дозволяє формалізувати транспортну систему та застосовувати до неї класичні алгоритми оптимізації. Для побудови та обробки графової структури використовується бібліотека NetworkX [1], яка надає ефективні інструменти для створення графів, додавання атрибутів ребер та виконання алгоритмів пошуку шляхів.

Алгоритм роботи пошуку оптимальних транспортних маршрутів передбачає поетапну обробку даних: отримання інформації про маршрути з бази даних, формування на її основі графа транспортної мережі та подальше застосування алгоритму Дейкстри для пошуку оптимальних шляхів. Обчислення виконується окремо для різних вагових параметрів, що дозволяє знаходити як найдешевший, так і найшвидший маршрут між заданими містами. Підтримується побудова складених маршрутів із пересадками та використанням різних типів транспорту в межах одного шляху.

Програмне забезпечення реалізовано мовою програмування Python із використанням сучасних бібліотек обробки даних та серверної логіки. Для створення веб-інтерфейсу застосовано фреймворк Django [2], який забезпечує маршрутизацію HTTP-запитів, обробку форм введення, роботу з шаблонами сторінок та інтеграцію бізнес-логіки додатку. Обрана архітектура спрощує супровід і розширення системи.

Для зберігання інформації про міста, маршрути, транспорт та проміжні точки використовується реляційна база даних MySQL [3]. Використання структурованої схеми даних та зв’язків між таблицями дозволяє ефективно виконувати вибірки та формувати набір сегментів маршруту для подальшого графового аналізу.

Веб-інтерфейс системи дозволяє користувачу обирати початкове та кінцеве місто, після чого виконується автоматичний розрахунок маршрутів. Результати відображаються у зручному структурованому вигляді як послідовність міст із зазначенням виду транспорту, часу та вартості для кожного сегмента, а також загальних підсумкових характеристик маршруту. Передбачено можливість подальшого розширення функціональності, зокрема додавання нових критеріїв оптимізації.

Література

[1] NetworkX documentation [Electronic resource]. – Access mode: https://networkx.org/documentation/stable/tutorial.html

[2] Django documentation [Electronic resource]. – Access mode: https://docs.djangoproject.com/

[3] MySQL documentation [Electronic resource]. – Access mode: https://dev.mysql.com/doc/refman/8.4/en/

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ ОБМІНУ РЕЧАМИ МІЖ ВЛАСНИКАМИ

Ушаков М.О., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

У сучасному суспільстві, де ідеї сталого споживання та раціонального використання ресурсів набувають особливої актуальності, розвиток платформ для прямого обміну речами між власниками стає важливим економічним та соціальним завданням. Традиційні моделі купівлі-продажу не завжди задовольняють потреби користувачів, які прагнуть позбутися непотрібних речей, що знаходяться у їх власності, отримавши натомість щось корисне без залучення грошових коштів.

Проблема розробки ефективних систем обміну полягає у необхідності створення зручного механізму зіставлення пропозицій та управління статусами об'єктів у реальному часі. Сучасні користувачі потребують інструментів, які дозволяють не лише виставляти власні пропозиції з детальним описом, а й гнучко керувати процесом узгодження обміну.

У даній роботі розроблено програмне забезпечення у вигляді вебдодатка, що дозволяє користувачам публікувати оголошення про наявні речі та пропонувати варіанти для обміну. Головна ідея полягає в автоматизації процесу відстеження пропозицій та сповіщенні сторін про успішне завершення транзакції. Алгоритм функціонування системи включає наступні етапи: додавання предметів, внесення опису речі для обміну, вибір предмета іншого користувача та створення запиту на обмін, можливість прийняття або скасування пропозиції власником, автоматична зміна статусу предметів на «обміняно» у разі успіху та розсилка повідомлень користувачам.

Важливою основою реалізованого алгоритму є автоматичне скасування конкуруючих запитів: якщо один обмін для предмета підтверджується, всі інші активні пропозиції, пов'язані з цими предметами, автоматично набувають статусу «скасовано». Це гарантує цілісність даних та запобігає подвійному обміну однієї і тієї ж речі.

У базі даних для вебдодатка було створено такі сутності:

– користувачі (users), що містить ідентифікатори та унікальні імена учасників системи;

– предмети (items), що зберігає назву, опис, посилання на власника та статус обміну кожної речі;

– пропозиції (swap_proposals), що реєструє запити на обмін, пов’язуючи запропоновану та бажану речі зі статусами;

– сповіщення (notifications), що накопичує повідомлення для користувачів про результати обміну із зазначенням тексту та часу створення.

Програмне забезпечення розроблено мовою Python, яка забезпечує високу швидкість розробки та ефективну роботу з базами даних, з використанням мікрофреймворку Flask [1] для створення вебінтерфейсу та системи керування базами даних PostgreSQL [2] для зберігання даних. Для взаємодії з базою даних застосовано пакет psycopg2 [3]. Обробка HTTP-запитів і навігація між сторінками реалізовані за допомогою механізму Django routing, а відображення динамічного контенту – з використанням шаблонів Jinja2 [4].

Література

[1] Flask Documentation [Electronic resource]. – Access mode: https://flask.palletsprojects.com/

[2] PostgreSQL Tutorial [Electronic resource]. – Access mode: https://www.postgresqltutorial.com/

[3] Psycopg2 Documentation [Electronic resource]. – Access mode: https://www.psycopg.org/docs/

[4] Pallets Projects: Jinja2 [Electronic resource]. – Access mode: https://github.com/pallets/jinja

ПРОГРАМНА СИСТЕМА УПРАВЛІННЯ ВЕЛОСЕРВІСОМ

Філоненко Р.В., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

Екологічна свідомість та здоровий спосіб життя набувають дедалі більшого значення в сучасному світі. Разом з тим велосипедний транспорт стає популярним вибором для міських мешканців. Системи прокату велосипедів активно розвиваються в багатьох містах, пропонуючи зручну альтернативу традиційному транспорту. Водночас ефективне управління такими системами потребує комплексного програмного забезпечення, здатного забезпечити облік велосипедів, обробку оренд, автоматичний розрахунок вартості та формування статистики, що відповідає рівню програмної системи. У зв'язку з цим розробка програмної системи управління велосервісом є актуальним завданням у контексті цифровізації міських сервісів.

Для реалізації подібних систем можуть застосовуватись різні вебфреймворки та системи керування базами даних. У даній роботі в основі обрано фреймворк Django [1] у поєднанні з системою MySQL [2], що дозволяє створити повнофункціональний вебдодаток з мінімальними витратами на налаштування. Окрім того в процесі розробки використано мову програмування Python, фронтенд-фреймворк Bootstrap 5 для створення адаптивного інтерфейсу користувача [3], а також модуль django.utils для коректної роботи з часовими мітками та часовими зонами [4].

Функціонування вебдодатку включає такі основні етапи: ініціалізацію бази даних та створення моделей даних, реалізацію представлень для обробки запитів користувачів, створення форм для валідації вхідних даних, розробку шаблонів інтерфейсу користувача, інтеграцію механізму автоматичного розрахунку вартості оренди, а також реалізацію модуля статистики та звітності. Для роботи з категоріальними даними у моделях Django застосовано механізм choices, що забезпечує цілісність даних і спрощує перевірку коректності введеної інформації.

Функціональні можливості розробленої системи охоплюють повний цикл управління велосипедами, велостанціями та орендами, автоматичний розрахунок вартості, облік оплати, а також формування детальної статистики використання. Процес оренди включає перевірку доступності велосипеда, зміну його статусу та збереження інформації про клієнта. Після повернення велосипеда система автоматично обчислює вартість оренди на основі її тривалості та погодинного тарифу з округленням часу до повної години в більшу сторону. Інтерфейс користувача побудовано з використанням Bootstrap 5, що забезпечує адаптивність та коректне відображення на різних пристроях. Для інформування користувача про результати виконаних операцій використовується механізм повідомлень Django.

Результатом роботи є повнофункціональна програмна система, що складається з основних програмних модулів Django та набору HTML-шаблонів. Система забезпечує ефективне управління велосервісом і має потенціал для подальшого масштабування та розширення функціональності.

Література

[1] Django Documentation – Django Web Framework [Electronic resource]. – Access mode: https://docs.djangoproject.com/

[2] MySQL Documentation – Database Management System [Electronic resource]. – Access mode: https://dev.mysql.com/doc/

[3] Bootstrap 5 Documentation – Front-end Framework [Electronic resource]. – Access mode: https://getbootstrap.com/docs/5.3/

[4] Django Utils Documentation – Django Utilities [Electronic resource]. – Access mode: https://docs.djangoproject.com/en/stable/ref/utils/

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНОГО ВИЗНАЧЕННЯ ЖАНРУ КІНОСТРІЧКИ

Шевченко А.С., Льовкін В.М.
E-mail: vliovkin@gmail.com
Запоріжжя, Національний університет «Запорізька політехніка»

У сучасних умовах стрімкого розвитку індустрії розваг та постійного зростання обсягів медіаконтенту особливого значення набуває розробка програмного забезпечення, яке дозволяє у автоматизованому режимі класифікувати та аналізувати текстові дані. З огляду на те, що кожного року створюються сотні кінострічок, виникає гостра потреба у алгоритмах, які здатні ефективно виявляти жанрову приналежність фільму на основі його короткого опису (синопсису). Це завдання є важливим тому, що завдяки цьому можна оптимізувати роботу стримінгових сервісів, рекомендаційних систем та цифрових архівів.

У цій роботі розроблено програмне забезпечення, яке в якості синопсису використовує дані з сайту imdb.com. Алгоритм, покладений в основу реалізації програмного забезпечення, має певну послідовність: отримання назви кінострічки, пошук її за назвою на сайті imdb.com у пошуковій системі google, отримання id знайденого фільму, отримання синопсису за отриманим id, аналіз короткого опису, виведення результатів.

Для створення програмного забезпечення була використана мова Python – високорівнева, об'єктно-орієнтована мова програмування, яка націлена на ефективне вирішення різноманітних завдань з обмеженим використанням ресурсів завдяки широкому переліку пакетів, розроблених спільнотою ентузіастів. Було використано бібліотеку imdbpy підпакет Cinemagoer [1] для найефективнішої роботи з сайтом imdb.com, пакет sklearn для машинного навчання та аналізу даних [2], пакет joblib для збереження та завантаження моделі [3], підпакет pd бібліотеки pandas для обробки csv-файлів та іншої обробки тексту [4].

Алгоритм роботи програми базується на обробці текстових даних та використанні моделі машинного навчання для класифікації. Процес реалізовано наступним чином:

– програма приймає від користувача назву фільму, після цього за допомогою Google Custom Search API та бібліотеки Cinemagoer виконує пошук ідентифікатора imdb для отримання повного синопсису кінострічки;

– для підготовки моделі у разі її поточної відсутності завантажується набір даних із csv-файлу, де назви жанрів очищуються від зайвих пробілів та приводяться до строкового типу за допомогою бібліотеки pandas;

– створюється конвеєр, що складається з TfidVectorizer для перетворення тексту в числові вектори (векторизація), завдяки чому оцінюється важливість кожного слова, та класифікатора LogicRegression, що є основою моделі, оскільки саме тут відбувається її навчання;

– сформована модель зберігається у файлі model.pkl за допомогою бібліотеки joblib для повторного використання без необхідності перенавчання;

– отриманий синопсис передається до навченої моделі, яка на основі виявлених лінгвістичних закономірностей визначає найбільш імовірний жанр стрічки;

– кінцевий результат виводиться користувачеві.

Література

[1] Cinemagoer documentation [Electronic resource]. – Access mode: https://cinemagoer.readthedocs.io/en/latest/

[2] sklearn API [Electronic resource]. – Access mode: https://scikit-learn.org/stable/api/index.html

[3] Joblib: running Python functions as pipeline jobs [Electronic resource]. – Access mode: https://joblib.readthedocs.io/en/stable/

[4] pandas User guide [Electronic resource]. Access mode - https://pandas.pydata.org/docs/user_guide/index.html

Секція 3

ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ: ОФІСНІ ТА СПЕЦІАЛІЗОВАНІ ПАКЕТИ

MATHEMATICAL MODEL AND SOFTWARE FOR SIZE PREDICTION JAVA WEB APPLICATIONS WITH SPRING FRAMEWORK

Dzhurynskyi M.O.
Supervisor: Makarova L.M.
E-mail: 251185@nuos.edu.ua, lidiia.makarova@nuos.edu.ua
Mykolaiv, Admiral Makarov National University of Shipbuilding

Predicting the size of Java web applications with Spring framework is a critical task in software engineering. It allows for the effective estimation of development and maintenance costs, project budget planning, and the determination of necessary timeframes for functional implementation. The relevance of this work lies in the necessity of constructing a mathematical model for predicting the size of Java web applications using the Spring framework, and creating a corresponding software application.

This work is devoted to solving the problem of size predicting of web applications in Java using the Spring framework using regression analysis, which is one of the methods of data mining. The independent variable is NOC – Number of Classes (variable X), and the dependent variable is kLOC – thousands of Lines of Code (variable Y).

For the correct application of linear regression analysis, the data must meet a number of statistical requirements, specifically that model variables must have a distribution close to normal [1]. Since software metrics often have "heavy tails," normalization is required. The work uses the decimal logarithmic transformation for this purpose ($Z_x = lg(X);\ Z_y = lg(Y)$), which allows normalizing the dependency, reducing the task of building a non-linear regression to a linear one.

To ensure model stability, the sample must be cleaned of anomalous observations (outliers). For multidimensional data, this was done using the squared Mahalanobis distance (d_i^2) and the Fisher statistic (T_{S_i}), utilizing the methodology described in [2]. If $T_{S_i} > F_{crit}$, the observation is considered an outlier and removed from the sample.

For this work, 36 projects were selected from the GitHub repository [3]. All projects are written in the Java programming language using the Spring framework and have open-source code. The CK tool was used to calculate metric values [4]. Upon full execution of the algorithm, seven projects were excluded.

Based on the justified mathematical apparatus, an algorithm for predicting the software metric kLOC was developed. The algorithm implements an iterative process of building a non-linear regression model with preliminary data cleaning. The developed program allows for the automation of building univariate non-linear regression models using normalizing transformations.

The console-based software application developed during the work, using IntelliJ IDEA [5], performs the following functions:

– import of application metrics;
– data normalization;
– elimination of outliers;
– construction of a linear regression model;
– construction of a non-linear regression model;
– calculation of confidence and prediction interval boundaries;
– calculation of non-linear regression model quality parameters;
– plotting of regression equation graphs.

To obtain the non-linear regression model equation, an inverse transformation is used ($\hat{Y} = 10^{b_0} X^{b_1}$). The resulting non-linear regression equation for the initial data takes the form:

$$\hat{Y} = 10^{-1.4463} X^{1.0485}$$

For final model verification, quality metrics such as the Coefficient of Determination (R^2), Mean Magnitude of Relative Error (MMRE), and Prediction Level (PRED) were used. The

obtained model quality indicators are R^2=0.9248, MMRE=0.1704, PRED(0.25)=0.7931, indicating its adequacy.

The results of the program execution, including the graphical representation of regression equations and their intervals, are displayed at the Fig. 1 - 3.

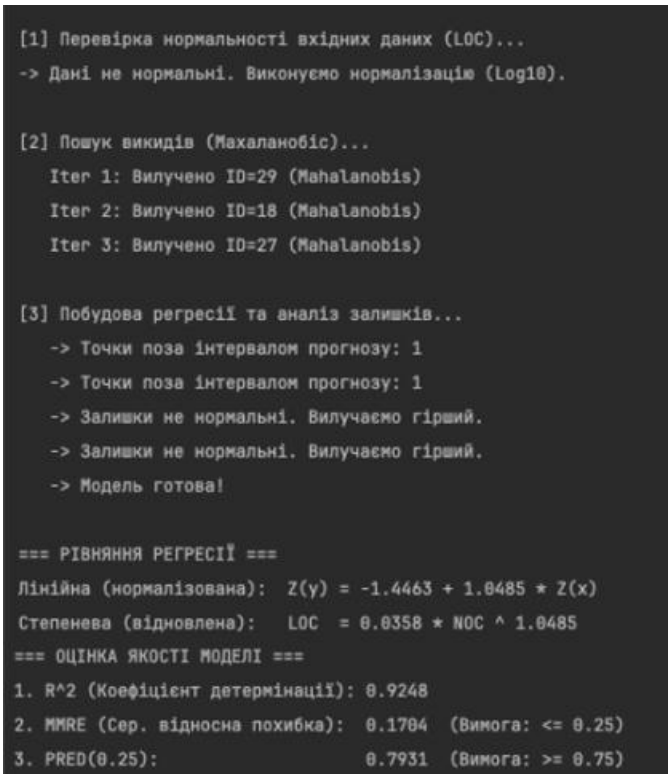


Fig. 1. Console screenshot showing the model construction result

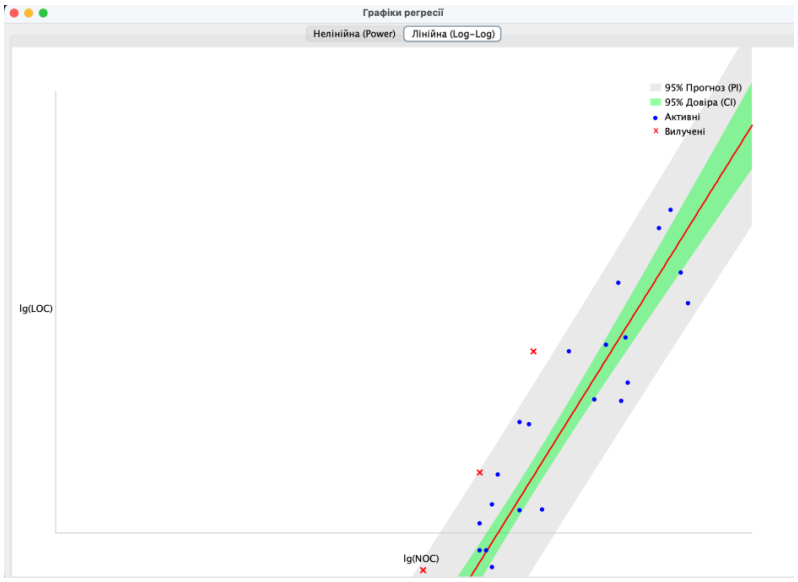


Fig. 2. Graphical representation of linear regression

In conclusion, a program was developed for the regression analysis of software metrics, which allows for constructing non-linear regression models with normalizing transformations and outlier detections.

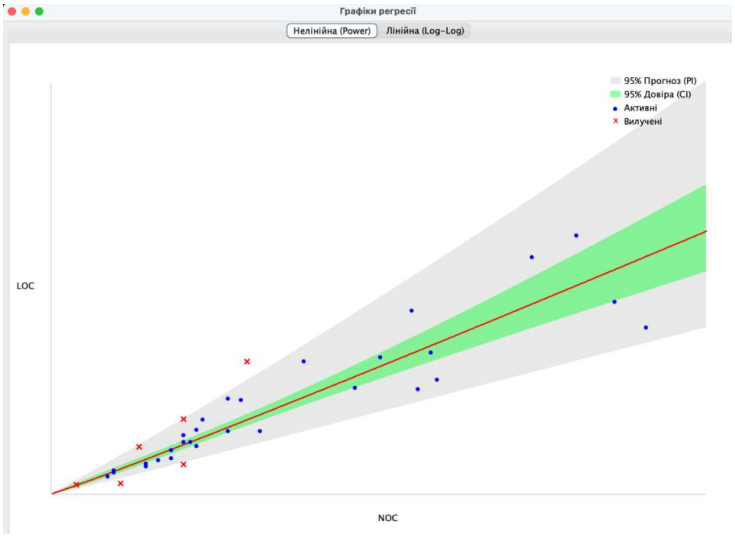


Fig. 3. Graphical representation of non-linear regression

References

[1] Chatterjee S., Hadi A.S. Regression analysis by example. Fifth Edition. New Jersey: A John Wiley & sons, Inc., Publication, 2012. 403 p.

[2] Prykhodko S., Prykhodko N., Makarova L., Pukhalevych A. Application of the Squared Mahalanobis Distance for Detecting Outliers in Multivariate Non-Gaussian Data. Proceedings of 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET). Lviv-Slavske, 2018. P. 962-965

[3] GitHub. https://github.com

[4] Java code metrics calculator (CK). https://github.com/mauricioaniche/ck

[5] IntelliJ IDEA. The Leading IDE for Professional Development in Java and Kotlin. https://www.jetbrains.com/idea/

OPEN-SOURCE TOOLS FOR 3D GAUSSIAN SPLATTING

Fadieiev P.V., Latanska L.O.
E-mail: pavelf200205@gmail.com, liudmyla.latanska@nuos.edu.ua
Mykolaiv, Admiral Makarov National University of Shipbuilding

3D Gaussian Splatting (3DGS) is a method for photorealistic novel-view synthesis that represents a 3D scene as millions of semi-translucent ellipsoid primitives [1]. The models are created using a multi-step optimization process from a set of multiple images of the scene taken from various known viewpoints. Unlike previous methods based on Neural Radiance Fields (NeRF), this method allows for real-time rendering at hundreds of frames per second. This makes it appealing for a wide range of applications, such as virtual production and VFX, virtual reality experiences, robotics, real estate, e-commerce, cultural heritage preservation, etc.

The original implementation of 3D Gaussian Splatting was published on GitHub under a non-permissive license that only allows non-commercial research and evaluation use. These limitations restrict the potential widespread adoption of the new technology, so the community developed a number of free and open-source tools for the creation of 3DGS models, including the following: Gsplat [2], LichtFeld Studio [3], and Brush [4].

Gsplat is an implementation of 3D Gaussian Splatting from the Nerfstudio project. Compared to the original implementation, it offers a significantly reduced memory footprint during training, as well as improved performance and quality thanks to upgrades based on subsequent research papers.

LichtFeld Studio is a fully featured suite for 3DGS that supports training, editing and rendering. A significant achievement of the LichtFeld Studio project is that it was rewritten without using the PyTorch framework, but with a custom CUDA implementation instead. Thanks to this,

the training speed was improved and the entire application is now available as a small, self-contained binary. With the support of sponsors, the LichtFeld Studio project holds competitions to improve model training speed, rendering quality and so on. LichtFeld Studio currently provides the fastest training speed of all 3DGS implementations, and integrates the most recent improvements to the training process based on the latest research papers.

Brush is a 3DGS training tool developed using the Rust programming language. It utilizes the WebGPU API and the Burn machine learning framework. Unlike other implementations, which require modern Nvidia graphics cards (RTX 2000 series or newer), Brush supports a much wider variety of hardware thanks to its WebGPU implementation. This includes desktop and mobile GPUs from AMD, Intel, Apple (M1-series), and Nvidia – including older pre-RTX models.

As 3D Gaussian Splatting is a recent development, most traditional 3D software does not yet support it, especially for editing purposes. This is where open source comes in to fill the gap.

SuperSplat is a powerful tool for 3DGS models, powered by the PlayCanvas engine [5]. It supports viewing, editing, and rendering the models as images or videos with camera animation.

The KIRI Engine team has developed a plugin for the popular Blender 3D software that enables the rendering and editing of 3DGS models [6]. With the help of this addon, 3DGS models can be rendered alongside traditional mesh-based models, enabling many powerful hybrid workflows.

References

[1] Kerbl, B., Kopanas, G., Leimkuhler, T., & Drettakis, G. (2023). 3D Gaussian Splatting for Real-Time Radiance Field Rendering. ACM Transactions on Graphics, 42(4). https://doi.org/10.1145/3592433

[2] Gsplat [Electronic resource]. – Resource access mode: http://www.gsplat.studio/

[3] LichtFeld Studio [Electronic resource]. – Resource access mode: https://lichtfeld.io/

[4] Brush [Electronic resource]. – Resource access mode: https://github.com/ArthurBrussee/brush

[5] SuperSplat Editor [Electronic resource]. – Resource access mode: https://github.com/playcanvas/supersplat

[6] 3DGS Render Blender Addon by KIRI Engine [Electronic resource]. – Resource access mode: https://github.com/Kiri-Innovation/3dgs-render-blender-addon

DEFORMATION-AWARE APPROXIMATION IN ARCHITECTURAL SCAN-TO-CAD PIPELINES

Toots R.
Supervisor: Shapovalova O.
E-mail: robtoots88@gmail.com
Kharkiv, Simon Kuznets Kharkiv National University of Economics

Automated generation of architectural drawings from point clouds is a rapidly developing field due to its clear potential benefits. However, many scan-to-CAD/scan-to-BIM pipelines idealize geometry, imposing planar, orthogonal, or template-based reconstructions. This can be a problem for buildings, particularly those of historic significance, where geometric deformations (tilt, convexity, non-orthogonality, settlement) are not noise but part of the building's actual condition. In this article, we argue that deformation preservation should be considered a requirement in automated design. We describe a concise, deformation-aware approximation strategy for point cloud-to-drawing workflows. We summarize recent advances in the automated generation of drawings and reconstructions in non-orthogonal interiors and the creation of linear drawings for heritage-focused environments. Building on these insights, we propose a hybrid approximation principle: using PCA-style orthogonal fitting for stable orientation estimation and applying limited refinement only where strict planarity is justified. The proposed framework supports open and reproducible workflows for automated architectural drawing production while preserving measurable deformations.

The availability of LiDAR systems has made documenting the actual condition of buildings based on dense point clouds increasingly practical. Automated pipelines can now generate floor plans, elevations, and sections, reduce manual drafting time and improve consistency. However, the modeling objective varies depending on the context. For typical facility management or renovation planning, simplified geometry may be acceptable; for historic architecture, idealization is inappropriate, so different approaches and solutions are needed.

Recent work demonstrates that modern CAD scanning pipelines are no longer limited to strict, grid-based building plans. These methods enable the processing of interiors and exteriors where walls are not perfectly straight, angles do not reach 90 degrees, and where "non-traditional" architectural forms are present. Floor plan extraction methods can now adapt to irregular layouts while still restoring accurate geometry. In buildings of historical significance, automated tools are also capable of creating clear and standardized line drawings directly from point clouds, capturing details that would otherwise require significant manual labor. Other advances focus on identifying and modeling complex architectural components even with insufficient data. Taking it together, these developments point to the existence of an automated workflow that accelerates the production of drawings and documentation, supports reliable digital preservation, and does not force objects into idealized geometric frameworks.

The next step is to ensure that pipelines do not "enhance" real buildings but rather preserve measurable deformations as is.

Preservation of Deformations is a Requirement

The Idealization Problem

Many reconstruction workflows, implicitly for the end user, optimize primitives (planes/lines) to ensure regularity (orthogonality/parallelism). This is effective, but can lead to systematic loss of information, such as:

Global deformations: general wall slope, floor slope, curvature over long distances.

Local deformations: bulges, uneven joints, hand-molding, war damage (holes, potholes).

Non-orthogonality: intended or accumulated deviations from right angles.

For the documentation of cultural heritage sites, such deviations can be the primary data of interest. Automatic drawing generation can capture these deformations, which may be missed or normalized in manual drawing [3]. This suggests that deformation-aware automation can reduce subjectivity and increase completeness.

Reconstruction is not the same as documentation

Automated pipelines must clearly distinguish between:

– Reconstruction – creating an idealized model close to the design

– Documentation – creating an accurate representation of the measured geometry.

For deformation-sensitive objects, the goal should be documentation, with the ability to abstract as a controlled post-process, not as an implicit output of the pipeline.

Method Overview: Deformation-Aware Approximation Principle

Below, a minimal and practical strategy is outlined, consistent with modern pipelines.

Pipeline Context

A typical workflow for converting a point cloud to a drawing includes:

– Preprocessing (noise reduction/downscaling),

– Segmentation (separation of structural surfaces/components),

– Fitting/approximation (planes/lines/curves),

– Vectorization and drawing rules (layers, symbols, dimensions),

– Comparison with the point cloud.

This process is suitable for end-to-end drawing generation and also extracts plans even in the presence of non-orthogonality [1]-[3].

Approximation Selection

A key step is the fitting of planes, lines for walls, floors, and other major surfaces. For deformed buildings, the fit should not assume a preferred axis (vertical distance regression). Instead, orientation estimation should be based on orthogonal residual minimization, which is

naturally consistent with PCA-style formulations, which in turn support arbitrary rotations. This helps in cases of tilted or deformed surfaces, reducing the risk of axis shift when the building is not perfectly aligned with the scanner or world coordinate system.

Hybrid Principle

A simple principle is proposed where deformation preservation remains a central element:

Global orientation estimation: Estimate dominant surface directions and orientations using an orthogonal, rotation-invariant fit (PCA-style) on segmented patches.

Conditional refinement: Refinement with tighter flatness and straightness constraints only when supported by the data (when the patch is clearly flat within a known tolerance or when design standards require strict correction for legibility). 3. Deformation reporting: Maintain a deformation signal (distance-to-match statistics or deviation fields) that can be visualized or attached to drawing metadata.

This principle complements workflows focused on cultural heritage preservation, where geometry and semantic structure are important [4], [5]. It also aligns with the development of automated drafting systems, which already incorporate multiple processing steps and domain-specific rules [1], [3].

Implementation Fragment

A normal vector can be obtained from a point fragment using eigenvalue covariance decomposition (an orientation-independent plane estimate). It is included only to clarify the mechanism, not as a complete algorithm.

```
import numpy as np
centroid = points.mean(axis=0)
X = points - centroid
cov = np.cov(X.T)
eigvals, eigvecs = np.linalg.eig(cov)
normal = eigvecs[:, np.argmin(eigvals)]
normal = normal / np.linalg.norm(normal)
```

The following points aim to find a balance between current automation capabilities and documentation needs:

– The principal fits into existing automated workflows

– The HBIM perspective is considered

– A fully trained system is not required; it can be integrated into open-source toolchains gradually (segmentation > fitting > vector export).

– Preserve deviation metrics so that the pipeline can support verification and interpretability.

Next:

– Define clear criteria for refinement and preservation,

– Test the tradeoff between drawing readability and the accuracy of measured deformations on representative datasets.

References

[1] Zhang, F., Kong, Q., Yuan, C., & Li, P. (2024). Automatic generation of architecture drawings from point clouds. Computer-Aided Civil and Infrastructure Engineering, 39(22), 3477–3488.

[2] Gao, X., Yang, R., Tan, J., & Liu, Y. (2024). Floor plan reconstruction from indoor 3D point clouds using iterative RANSAC line segmentation. Journal of Building Engineering, 89, 109238.

[3] Dong, S., Wu, D., Kong, W., Liu, W., et al. (2025). Research on intelligent generation of line drawings from point clouds for ancient architectural heritage. Buildings, 15(18), 3341–3364.

[4] Pang, B., Yang, J., Xia, T., Zhang, A., Zhang, K., Xu, Q., & Wang, F. (2025). Automated heritage building component recognition and modelling based on local features. Journal of Cultural Heritage, 71, 252–264.

[5] Escudero, P. A. (2023). Scan-to-HBIM: Automated transformation of point clouds into 3D BIM models for the digitization and preservation of historic buildings. VITRUVIO – International Journal of Architectural Technology and Sustainability, 8(2), 52–63.

OPEN-SOURCE TOOLS FOR RAPID LECTURE PRESENTATION DEVELOPMENT

Venhrina O.S.
E-mail: olena.venhrina@hneu.net
Kharkiv, Simon Kuznets Kharkiv National University of Economics

In the contemporary online educational and scientific environment, presentation development is an essential component of knowledge communication. Traditional tools, such as PowerPoint or Keynote, often require significant time investment for formatting and design adjustments. The transition to open-source solutions enables the optimization of the presentation creation process by shifting the focus toward content rather than technical execution.

The objective of this work is to investigate the efficacy of combining Logseq and Google Antigravity (or VS Code) with the Quarto extension and the reveal.js library for the rapid development of HTML-based presentations.

The presentation preparation process consists of three primary stages:

Stage 1. Content Structuring in Logseq. Logseq is utilized to organize knowledge into multi-level lists, ensuring a logical material structure [1]. Its seamless content transclusion features allow for effective material reuse. Upon completion, the page is exported in Markdown (.md) format [2].

Stage 2. Processing in Google Antigravity or VS Code. The exported .md file is converted into the .qmd format, enabling advanced markup options, including slide headers, body text, and speaker notes. Integrating Quarto with reveal.js automates the presentation's visual styling and layout [3-6].

Stage 3. HTML Presentation Generation. Export parameters are defined via a YAML block at the beginning of the file [7]. The process results in an interactive HTML presentation, compatible with any modern web browser and supporting a dedicated speaker mode.

The proposed workflow offers several key advantages:

– minimization of design and formatting time while maintaining cross-browser compatibility;

– high flexibility of configuration through YAML metadata;

– availability of a professional speaker mode comparable to PowerPoint.

Utilizing the Logseq, Quarto, and reveal.js stack enables a transition away from the manual slide formatting inherent in traditional office suites. By adopting the 'presentation-as-code' concept, the content creation process is automated: the instructor focuses on organizing material within a personal knowledge base, while technical visualization is handled by the software tools. This approach ensures full cross-browser compatibility and simplifies subsequent editing. Implementing such open-source solutions in the educational process reduces reliance on proprietary software and serves as a practical step toward the digitalization of academic content.

References

[1] Logseq [Electronic resource]. – Resource access mode: https://logseq.com/

[2] Markdown Guide [Electronic resource]. – Resource access mode: https://www.markdownguide.org/

[3] Google Antigravity [Electronic resource]. – Resource access mode: https://antigravity.google/

[4] Visual Studio Code [Electronic resource]. – Resource access mode: https://code.visualstudio.com/

[5] Welcome Quarto [Electronic resource]. – Resource access mode: https://quarto.org/

[6] Reveal.js [Electronic resource]. – Resource access mode: https://revealjs.com/

[7] YAML [Electronic resource]. – Resource access mode: https://yaml.org/

DAVINCI RESOLVE: ЦИФРОВІ ТЕХНОЛОГІЇ АУДІОВІЗУАЛЬНИХ МИСТЕЦТВ

Бондаренко Ю.В., Попов І.М.
E-mail: bondarenko.yuliia@ksada.org
Харків, Харківська державна академія дизайну і мистецтв

Сучасний стан аудіовізуальних мистецтв характеризується безпрецедентним рівнем технологічної інтеграції, де межі між знімальним процесом, монтажем, створенням візуальних ефектів та фіналізацією зображення стають дедалі прозорішими. У центрі цієї цифрової революції знаходиться програмний комплекс DaVinci Resolve — унікальний феномен в історії медіатехнологій, який пройшов шлях від елітарної апаратної системи вартістю в сотні тисяч доларів до загальнодоступного інструменту, що демократизував професійний постпродакшн.

Дослідження феномену DaVinci Resolve вимагає виходу за межі простого огляду функціональних можливостей програмного забезпечення. Це, насамперед, аналіз зміни виробничих парадигм, переосмислення ролі творця в цифрову епоху та дослідження впливу математичних алгоритмів на естетику кінозображення. Традиційний кінематографічний процес ("pipeline") десятиліттями будувався на сегрегації етапів: монтаж здійснювався в одній системі (offline), візуальні ефекти — в іншій, а кольорокорекція та мастеринг — у третій (online). Такий підхід, продиктований технологічними обмеженнями минулого, створював суттєві бар'єри для творчої гнучкості, призводив до помилок при конвертації даних (conforming) та значних фінансових витрат.

DaVinci Resolve, під керівництвом компанії Blackmagic Design, фактично зруйнував цю модель, запропонувавши концепцію "в усьому" (all-in-one). Це не просто технічна оптимізація; це зміна філософії візуального мистецтва. Завдяки архітектурі, що об'єднує редагування, композитинг, роботу зі звуком та грейдинг у єдиному середовищі зі спільним доступом до даних, стало можливим нелінійне співробітництво. Колорист може починати роботу над сценою ще до завершення фінального монтажу, а звукорежисер — паралельно накладати ефекти, що було немислимим у класичних стрічкових або ранніх цифрових процесах [1].

Розуміння сучасної архітектури та філософії DaVinci Resolve неможливе без занурення в його історію, яка є відображенням еволюції всієї індустрії цифрового відео.

Історія бренду бере початок у 1984 році в Корал-Спрінгс, штат Флорида, де була заснована компанія da Vinci Systems. У той час кольорокорекція була вузькоспеціалізованим процесом, тісно пов'язаним з перенесенням зображення з кіноплівки на відеоплівку (телесіне). Перші системи, такі як "The Wiz" (1982), а згодом легендарні DaVinci Classic та DaVinci Renaissance 888, були виключно апаратно-програмними комплексами [2].

Ці системи базувалися на кастомному апаратному забезпеченні. Обробка відеосигналу в реальному часі, особливо у роздільній здатності 2K (яка наприкінці 90-х вважалася високою), вимагала обчислювальних потужностей, недоступних для звичайних комп'ютерів. Система DaVinci 2K, випущена у 1998 році, стала золотим стандартом галузі. Вона використовувала масивні мейнфрейми та спеціалізовані плати обробки, а вартість повного комплексу могла сягати від $200,000 до $800,000 [2].

Архітектура цих ранніх систем використовувала паралельну обробку даних на базі топології InfiniBand, що дозволяло працювати з величезними потоками даних некомпресованого відео. Це створило навколо DaVinci ореол елітарності. Колористи, які працювали на цих системах, були "вищою кастою" постпродакшну, а самі кімнати кольорокорекції (color suites) нагадували центри управління польотами з масивними панелями керування, оснащеними трекболами та кільцями.

Саме в цей період сформувався фізичний інтерфейс взаємодії з кольором, який використовується досі. Три трекболи (Lift, Gamma, Gain) дозволяли колористу одночасно керувати тінями, середніми тонами та світлими ділянками зображення. Цей тактильний

підхід був не просто питанням зручності, а необхідністю: у режимі реального часу, під час прогону плівки, миша та клавіатура були занадто повільними інструментами [3].

Поворотний момент в історії технології настав у 2009 році, коли австралійська компанія Blackmagic Design придбала активи збанкрутілої da Vinci Systems [2]. Генеральний директор Blackmagic, Грант Петті (Grant Petty), мав радикально іншу візію розвитку продукту. Він прагнув зруйнувати високий поріг входу в індустрію, вважаючи, що творчість не повинна обмежуватися бюджетом на обладнання.

Статистика яскраво ілюструє цей зсув. У 2009 році, на момент придбання, у світі налічувалося близько 100 активних користувачів систем DaVinci. До січня 2019 року база користувачів перевищила 2 мільйони, зрівнявшись з Final Cut Pro X. Це перетворило Resolve з нішового інструменту на індустріальний стандарт, доступний кожному, хто має достатньо потужний комп'ютер.

Унікальність DaVinci Resolve як цифрової технології полягає не лише в наборі функцій, але й у фундаментальних принципах обробки даних, які відрізняють її від конкурентів на кшталт Adobe Premiere Pro чи Avid Media Composer.

Фундаментом, на якому будується якість зображення в DaVinci Resolve, є використання 32-бітної математики з плаваючою комою (32-bit float) для всіх операцій з кольором та ефектами [4]. Це критично важлива відмінність від систем, що працюють у цілочисельних форматах (8-bit або 10-bit integer).

У традиційному 8-бітному відео кожен колірний канал (червоний, зелений, синій) має 256 значень (від 0 до 255). Значення 0 — це абсолютний чорний, 255 — абсолютний білий. Якщо в процесі корекції яскравість пікселя перевищує 255, інформація обрізається (clipping) і втрачається назавжди.

У 32-бітному середовищі Resolve з плаваючою комою діапазон значень є віртуально нескінченним.

Однією з найбільших концептуальних переваг Resolve, особливо у сторінках Color та Fusion, є нодова архітектура. Більшість графічних редакторів (Photoshop, After Effects, Premiere Pro) використовують систему шарів (layers), де кожен новий елемент кладеться поверх попереднього [5].

Інтеграція Fusion зробила Resolve найпотужнішою системою "все в одному". На відміну від After Effects, який базується на шарах і є псевдо-3D (2.5D), Fusion має повноцінний 3D-простір (True 3D workspace) [6].

Починаючи з 16-ї версії і досягнувши піку у версії 20, Blackmagic Design активно впроваджує технології машинного навчання під назвою DaVinci Neural Engine. Цей рушій використовує глибокі нейронні мережі для вирішення рутинних, але трудомістких задач, звільняючи час для творчості [7].

У Resolve 20 можливості ШІ у сфері звуку (Fairlight) досягли революційного рівня:

Voice Isolation (Ізоляція голосу): Ця функція стала "рятівником" для багатьох проектів. Нейромережа відокремлює людську мову від будь-якого фонового шуму (шум вулиці, вітер, гул кондиціонера, музика) з неймовірною якістю. Це дозволяє використовувати "брудні" записи з петличок або накамерних мікрофонів у фінальному міксі, уникаючи дорогого переозвучення (ADR) [8].

Dialogue Matcher: ШІ аналізує спектральні характеристики еталонного діалогу і автоматично підганяє під нього інші кліпи (еквалізацію, рівень, атмосферу кімнати). Це вирішує проблему різного звучання дублів, записаних у різні дні або на різні мікрофони [7].

IntelliTrack AI: Використовує відеотрекінг для керування аудіо. Наприклад, якщо автомобіль проїжджає через кадр зліва направо, IntelliTrack автоматично панорамує звук синхронно з рухом об'єкта, створюючи імерсивну звукову картину [7].

DaVinci Resolve також починає інтегрувати генеративні функції. Music Remixer дозволяє розділяти готову музичну композицію на окремі стем-файли (вокал, ударні, бас, інструменти) для створення реміксів або підгонки музики під монтаж. Функція AI Music

Editor може автоматично перемонтувати музичний трек під задану тривалість сцени, зберігаючи музичну структуру та ритм [9].

У сучасному кіновиробництві, де на одному майданчику можуть працювати камери ARRI, RED, Sony та дрони DJI, кожна з яких має свою кольорову науку, питання уніфікації кольору (Color Management) є критичним. Resolve Color Management (RCM) це власна розробка Blackmagic Design, яка пропонує автоматизований підхід. RCM автоматично розпізнає метадані камери (наприклад, Blackmagic RAW або ARRI RAW) і конвертує всі вхідні сигнали у єдиний величезний колірний простір — DaVinci Wide Gamut Intermediate.

Одним із найважливіших факторів успіху DaVinci Resolve є унікальна бізнес-модель Blackmagic Design, яка протистоїть тренду SaaS (Software as a Service), домінуючому в індустрії (Adobe, Avid).

Adobe Creative Cloud вимагає щомісячної оплати (~$60/міс). Якщо користувач перестає платити, він втрачає доступ до своїх проектів. Це викликає значне невдоволення, відоме як "subscription fatigue" (втома від підписок) [10].

Blackmagic Design пропонує дві версії:

DaVinci Resolve (Free): Безкоштовна версія, яка є повнофункціональною для 95% користувачів. Вона не має водяних знаків, обмежень на експорт до 4K (UHD) та включає більшість професійних інструментів кольору та монтажу.

DaVinci Resolve Studio ($295): Платна версія, яка купується один раз назавжди. Вона включає Neural Engine, підтримку декількох GPU, експорт вище 4K, інструменти 3D, шумоподавлення та ефекти плівкового зерна. Усі майбутні оновлення (наприклад, з версії 16 до 20) є безкоштовними.

DaVinci Resolve є унікальним прикладом технологічної еволюції, яка переросла своє початкове призначення і змінила ландшафт цілої індустрії.

Література

[1] DaVinci Resolve – Fusion [Електронний ресурс]. – Режим доступу до ресурса: https://www.blackmagicdesign.com/products/davinciresolve/fusion

[2] DaVinci Resolve [Електронний ресурс]. – Режим доступу до ресурса: https://en.wikipedia.org/wiki/DaVinci_Resolve

[3] Layers VS Nodes [Електронний ресурс]. – Режим доступу до ресурса: https://www.youtube.com/watch?v=SLoP0lMsGAE

[4] Understanding Integer vs. Float [Електронний ресурс]. – Режим доступу до ресурса: https://www.steakunderwater.com/VFXPedia/__man/Resolve18-6/DaVinciResolve18_Manual_files/part1904.htm

[5] ClickActionFilms. Layers vs. Nodes - what are the differences for compositing? [Електронний ресурс]. – Режим доступу до ресурса: https://www.reddit.com/r/vfx/comments/13sdldo/layers_vs_nodes_what_are_the_differences_for/

[6] Schmierer S. Nodes vs. Layers [Електронний ресурс]. – Режим доступу до ресурса: https://www.videomaker.com/article/c3/17836-nodes-vs-layers/

[7] DaVinci Resolve – What's New [Електронний ресурс]. – Режим доступу до ресурса: https://www.blackmagicdesign.com/products/davinciresolve/whatsnew

[8] Hermans J. 7 must-know AI & NEURAL ENGINE Features in DaVinci Resolve STUDIO 18 [Електронний ресурс]. – Режим доступу до ресурса: https://www.youtube.com/watch?v=CK4Dv5acRY8

[9] Blackmagic Design Announces DaVinci Resolve 20: press release [Електронний ресурс]. – Режим доступу до ресурса: https://www.blackmagicdesign.com/media/release/20250404-02

[10] Zhou Z. Free feels wrong: a text analysis of consumer preferences on davinci resolve's one-time payment model. International Journal of Business Management and Economic Review. 2025. Vol. 8, No. 05. Pp. 74–83. ISSN 2581-4664.

СИСТЕМА МОНІТОРИНГУ ВНУТРІШНІХ ВРАЗЛИВОСТЕЙ ЛОКАЛЬНОЇ МЕРЕЖІ НА ОСНОВІ АНАЛІЗУ ШАБЛОННОЇ АКТИВНОСТІ ЗЛОВМИСНИКІВ

Волков В.В.
E-mail: vikvolk99@gmail.com
Миколаїв, Чорноморський національний університет імені Петра Могили

У сучасних умовах значна кількість компаній експлуатує великі парки персональних комп'ютерів, серверів та різноманітних кінцевих пристроїв, які мають бути налаштовані з урахуванням різних сценаріїв використання та підвищених вимог до інформаційної безпеки. [1] Паралельно з цим багато сервісів використовують розгалужену інфраструктуру віртуальних машин, зокрема в демілітаризованих зонах (DMZ), що суттєво ускладнює контроль стану мережі.)[2] [3]

Попри критичну важливість таких сегментів, [4] системам моніторингу внутрішніх мереж та виявленню потенційних вразливостей часто приділяється недостатньо уваги. [5] Водночас інженеру, який відповідає за експлуатацію та безпеку інфраструктури, необхідно мати актуальну та цілісну картину стану мережі для своєчасного виявлення інцидентів і реагування на них.

У межах даної роботи розглядається підхід до побудови системи моніторингу та виявлення внутрішніх вразливостей локальних мереж, що базується на використанні шаблонів поведінкової активності потенційного зловмисника. [6] [7]Запропонований підхід дозволяє оперативно ідентифікувати підозрілу активність, швидко знаходити скомпрометовані або потенційно небезпечні хости та ізолювати їх без суттєвого впливу на роботу всієї мережі.

Наукова новизна роботи полягає у:

– застосуванні поведінкових шаблонів для виявлення внутрішніх вразливостей локальної мережі без використання глибокого аналізу пакетів (DPI); [8]

– поєднанні централізованої аналітики з децентралізованим виконанням політик реагування;

– адаптації механізмів реагування відповідно до ролі та контексту кінцевого хоста в мережі.

Практична цінність роботи:

– можливість впровадження системи у гетерогенних середовищах (Linux, Windows, FreeBSD);

– низькі вимоги до обчислювальних ресурсів агентів;

– інтеграція з існуючими системами моніторингу та візуалізації, зокрема Grafana;

– можливість автоматичного створення правил міжмережевих екранів для швидкого реагування, а також потенційне централізоване керування blacklist-ами на мережевому маршрутизаторі.

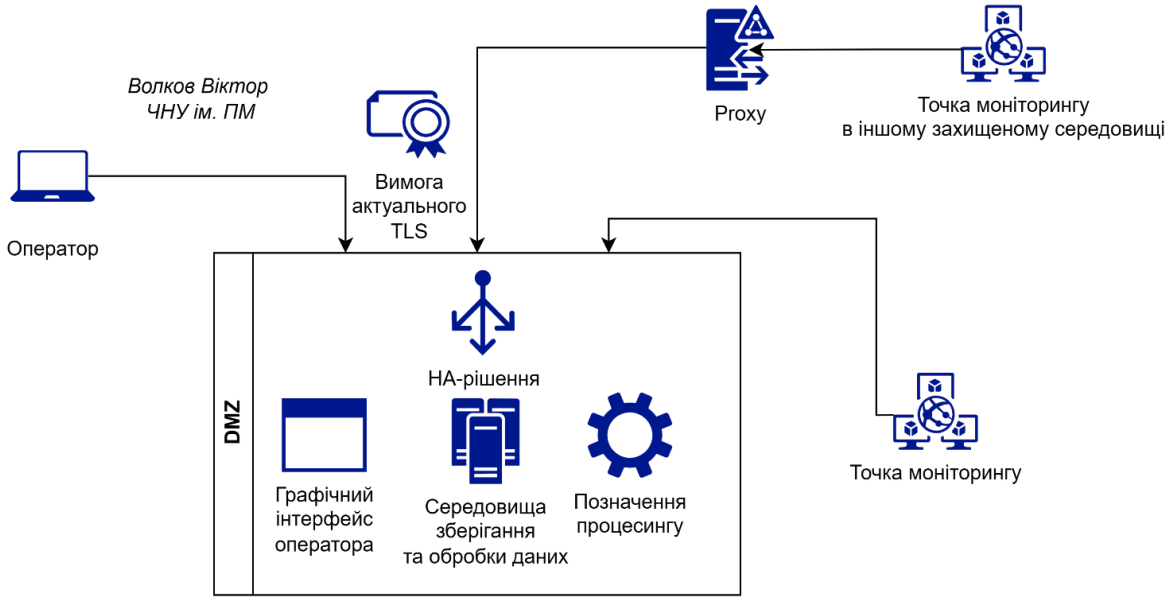
На представленій схемі (рис. 1) зображено узагальнену архітектуру системи моніторингу та виявлення внутрішніх вразливостей локальної мережі з акцентом на безпечну взаємодію компонентів і розмежування зон доступу.

Взаємодія з системою здійснюється оператором, який підключається до графічного інтерфейсу управління. Для забезпечення конфіденційності та цілісності переданих даних доступ оператора можливий виключно з використанням актуальних версій протоколу TLS, що знижує ризик перехоплення або підміни трафіку під час керування системою та перегляду результатів моніторингу.

На представленій схемі зображено узагальнену архітектуру системи моніторингу та виявлення внутрішніх вразливостей локальної мережі з акцентом на безпечну взаємодію компонентів і розмежування зон доступу.

Взаємодія з системою здійснюється оператором, який підключається до графічного інтерфейсу управління. Для забезпечення конфіденційності та цілісності переданих даних доступ оператора можливий виключно з використанням актуальних версій протоколу TLS,

що знижує ризик перехоплення або підміни трафіку під час керування системою та перегляду результатів моніторингу.



Рисноук 1 – Архітектура системи

Що стосується взаємодії системи, вона представлена як подієво-орієнтований процес між децентралізованими точками моніторингу та центральним ядром аналізу. У процесі роботи агенти фіксують мережеву активність хостів і передають узагальнені метадані до Core-компонента, де відбувається кореляція подій та ідентифікація поведінкових шаблонів. У разі виявлення активності, що відповідає формалізованим ознакам атаки, система виконує оцінку рівня ризику з урахуванням контексту та історії взаємодій. На основі результатів оцінювання формується керуюче рішення, яке може передбачати обмеження або блокування мережевої взаємодії. Прийняті рішення централізовано зберігаються та використовуються для подальшої кореляції подій іншими компонентами системи. Реалізація реагування здійснюється на рівні операційної системи хоста шляхом автоматизованого застосування правил безпеки.

Таким чином, взаємодія компонентів утворює замкнений цикл, що охоплює збір даних, аналіз, оцінку ризиків та автоматизоване реагування, забезпечуючи своєчасне виявлення та нейтралізацію внутрішніх загроз у локальній мережі.

Розроблена система моніторингу та виявлення внутрішніх вразливостей локальної мережі побудована за модульним принципом і складається з логічно відокремлених компонентів, що взаємодіють між собою через захищені програмні інтерфейси. Такий підхід забезпечує масштабованість, гнучкість і можливість адаптації системи до різних інфраструктурних середовищ.

Frontend призначений для взаємодії оператора з системою та реалізує функції керування, контролю й візуалізації. Через веб-інтерфейс оператор отримує доступ до налаштування конфігурацій, політик безпеки та сценаріїв реагування, а також до перегляду подій безпеки, результатів аналізу та історії дій системи. Взаємодія з ядром здійснюється виключно через захищені канали з використанням HTTPS і REST API. Окремо реалізовано інтеграцію з Grafana для моніторингу ключових метрик і механізми аудиту дій оператора.

Core-компонент є центральним логічним елементом системи та відповідає за обробку даних, аналітику й прийняття керуючих рішень. До його складу входить API Gateway, який забезпечує уніфікований доступ до функціональності системи з використанням токен-орієнтованої автентифікації. Аналіз зібраних даних виконується модулем Pattern Engine на основі поведінкових шаблонів і виявлення аномалій, після чого Scoring Engine здійснює

оцінку рівня ризику для кожного хоста. На основі отриманих оцінок Policy Engine ініціює відповідні сценарії реагування з урахуванням визначених політик безпеки та бази знань загроз.

End Protect Point являє собою компонент захисту кінцевої точки, що розгортається безпосередньо на хості. Основним елементом є агент, який здійснює збір локальних подій і первинний аналіз активності. До його складу входять модулі збору подій операційної системи, аналізу мережевого трафіку на рівні метаданих та локального аналізу поведінки. Для реалізації активних дій реагування використовується адаптер міжмережевого екрана, який взаємодіє з ОС через відповідні API. Обмін даними з Core-компонентом здійснюється через захищені канали з використанням токенів доступу.

Grafana використовується для візуалізації стану системи та відображення ключових показників її роботи. Компонент отримує метрики з Core-компонента через API з доступом у режимі «тільки читання», що зменшує ризики несанкціонованого впливу. Інформаційні панелі дозволяють оператору аналізувати динаміку подій безпеки, навантаження на систему та ефективність застосованих політик.

PostgreSQL виконує роль централізованого сховища даних системи та використовується для збереження облікових даних операторів, токенів агентів, журналів подій і результатів аналізу. Також у базі зберігаються службові структури, зокрема списки блокування. Взаємодія з Core-компонентом здійснюється через захищене TCP-з'єднання, що забезпечує цілісність і надійність зберігання інформації.

Що стосується взаємодії системи, вона представлена як подієво-орієнтований процес між децентралізованими точками моніторингу та центральним ядром аналізу. У процесі роботи агенти фіксують мережеву активність хостів і передають узагальнені метадані до Core-компонента, де відбувається кореляція подій та ідентифікація поведінкових шаблонів. У разі виявлення активності, що відповідає формалізованим ознакам атаки, система виконує оцінку рівня ризику з урахуванням контексту та історії взаємодій.

На основі результатів оцінювання формується керуюче рішення, яке може передбачати обмеження або блокування мережевої взаємодії. Прийняті рішення централізовано зберігаються та використовуються для подальшої кореляції подій іншими компонентами системи. Реалізація реагування здійснюється на рівні операційної системи хоста шляхом автоматизованого застосування правил безпеки.

Таким чином, взаємодія компонентів утворює замкнений цикл, що охоплює збір даних, аналіз, оцінку ризиків та автоматизоване реагування, забезпечуючи своєчасне виявлення та нейтралізацію внутрішніх загроз у локальній мережі.

Література

[1] Stallings W. Network Security Essentials: Applications and Standards. 6th Edition. Pearson. 2020. P. 1–450. ISBN: 978-0134605382.

[2] NIST SP 800-41 Rev. 1. Guidelines on Firewalls and Firewall Policy. National Institute of Standards and Technology. 2009. P. 1–70. DOI: 10.6028/NIST.SP.800-41r1.

[3] Cisco Systems. Designing Secure Network Architectures. Cisco Press. 2017. P. 1–320. ISBN: 978-1587144212.

[4] NIST SP 800-92. Guide to Computer Security Log Management. National Institute of Standards and Technology. 2006. P. 1–36. DOI: 10.6028/NIST.SP.800-92.

[5] MITRE ATT&CK® Framework. MITRE Corporation. 2024. URL: https://attack.mitre.org

[6] Sommer R., Paxson V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. Proc. of the IEEE Symposium on Security and Privacy. 2010. P. 305–316. DOI: 10.1109/SP.2010.33.

[7] Behl A., Behl K. Cyberwar: The Next Threat to National Security. Springer. 2016. P. 1–180. ISBN: 978-3319404217.

[8] Cisco NetFlow White Papers. Cisco Systems. 2023. URL: https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html

РОЗРОБКА ЧАТ-БОТА ДЛЯ АВТОМАТИЗАЦІЇ ПЕРЕВІРКИ УЧНІВСЬКИХ РОБІТ

Волкотрубенко Є.О., Козакевич М.С.
Керівник: Гусєва-Божаткіна В.А.
E-mail: jenya2010v@gmail.com, maxim.kozakevych@gmail.com gusevabozh@meta.ua,
Миколаїв, Національний університет кораблебудування імені адмірала Макарова

В сучасних умовах проведення навчально процесу за умов застосуванні діджитал-технологій зросло навантаження на шкільних вчителів. Під час виконання роботи було розроблено програмне забезпечення у вигляді Telegram-бота «Помічник вчителя», який автоматизує перевірку учнівських робіт. Метою роботи є спрощення перевірки знань учнів за допомогою діджитал-технологій та зменшення навантаження на вчителів.

Запропонований Telegram-бот має 4 ключові кабінети: «Кабінет голови циклової комісії», «Кабінет вчителя», «Кабінет учня», «Кабінет батьків», що покривають всі основні сценарії використання: від створення завдань і здачі робіт до контролю відвідуваності та перегляду статистики. Розроблений дизайн декількох кабінетів наведено на рисунках 1-2.

Рисунок 1 - Профіль учня

Рисунок 2 - Головне меню учня

Під час проєктування та реалізації програмного забезпечення застосовано шарову архітектуру та принципи SOLID, що забезпечують масштабованість, надійність та зручність супроводу. Основна увага приділялась асинхронній обробці запитів та інтеграції інтелектуальних сервісів. Як базовий інструмент для реалізації серверної частини чат-бота було обрано фреймворк aiogram — асинхронну бібліотеку для створення Telegram-ботів мовою програмування Python. Для роботи з базою даних було використано бібліотеку SQLModel [1], яка поєднує можливості ORM SQLAlchemy [2] та систему валідації даних Pydantic [3].Застосування SQLModel сприяє підвищенню надійності зберігання даних та спрощує подальше розширення структури бази без суттєвих змін у бізнес-логіці застосунку. Архітектура проєкту побудована з урахуванням перспектив подальшого розвитку. У межах поточної реалізації чат-боту додано функціонал автоматичної перевірки робіт за допомогою ШІ, обрано бібліотеку google-genai [4], що дозволяє оцінювати завдання з відкритою відповіддю, а також систему контролю наукової доброчесності (плагіат) між роботами учнів. Передбачена можливість подання учнями апеляції на оцінку, якщо ШІ помилився. У майбутньому можливе розширення палітри інструментів штучного інтелекту, зокрема шляхом підключення хмарних рішень та інших спеціалізованих сервісів, що дозволить нарощувати функціональність без порушення цілісності та стабільності. Для підвищення якості отримування знань введено розрахунок середнього балу та відхилення оцінок. Це дає можливість швидко аналізувати рівень засвоєння матеріалу та виявляти слабкі сторони.

Таким чином, розроблений чат-бот «Помічник вчителя» є функціонально насиченим інструментом, який оптимізує роботу вчителя, підвищує об'єктивність оцінювання та сприяє покращенню якості навчального процесу в цілому.

Література

[1] Aiogram. (n.d.). Asynchronous framework for Telegram bots in Python [Electronic resource]. – Resource access mode: https://docs.aiogram.dev

[2] SQLAlchemy. (n.d.). Python SQL toolkit and object relational mapper [Electronic resource]. – Resource access mode: https://www.sqlalchemy.org

[3] SQLModel. (n.d.). SQL databases with Python and Pydantic [Electronic resource]. – Resource access mode: https://sqlmodel.tiangolo.com

[4] Google. (n.d.). Gemini API documentation [Electronic resource]. – Resource access mode: https://ai.google.dev

ПРИКЛАДНІ ПРОГРАМНІ ЗАСОБИ ДЛЯ ПРОЄКТУВАННЯ, МОДЕЛЮВАННЯ Й СУПРОВОДУ МЕХАТРОННИХ ТА РОБОТОТЕХНІЧНИХ СИСТЕМ

Любименко О.М, Штепа О.А.
E-mail: e.n.lyubimenko@gmail.com
Дрогобич, Донецький національний технічний університет
Луцьк, Луцький національний технічний університет

У роботі розглянуто роль прикладного програмного забезпечення в повному життєвому циклі мехатронних і робототехнічних систем – від формування вимог та конструкторського проєктування до моделювання, програмування, випробувань і експлуатаційного супроводу. Показано, що стійкий результат забезпечує не “один універсальний пакет”, а узгоджений інструментальний ланцюг, у якому офісні застосунки підтримують управління вимогами, планування, документацію та аналітику, а спеціалізовані пакети — CAD, CAE, схемотехнічне проєктування, середовища розробки ПЗ, вбудованих систем, симулятори, засоби інтеграції та експериментальної верифікації. Наведено класифікацію програмних засобів за етапами життєвого циклу, критерії вибору інструментів (інтегрованість, відтворюваність, підтримка стандартів обміну, вартість, ліцензування, вимоги до компетенцій) і практичну модель інтеграції “документи → модель → симуляція → прототип → дані”. Окремо розглянуто використання відкритих екосистем для робототехніки, зокрема ROS 2 як набору бібліотек і інструментів для побудови роботизованих застосунків.

Мехатроніка описує інтеграцію механіки, електроніки та комп’ютерного керування в єдині технічні системи. Робототехнічні системи є одним із найбільш наукоємних прикладів такого синтезу: вони включають механічну конструкцію, приводи, сенсори, електронні вузли, алгоритми керування та програмне забезпечення високого рівня (навігація, планування, комп’ютерний зір тощо).

У практиці розроблення роботів зростає значення інструментальної узгодженості: команда працює паралельно над механікою, електронікою й кодом, а результат повинен бути керованим за змінами (версійність), перевірюваним (симуляція, тести) і відтворюваним (повторна збірка та налаштування). Саме тому прикладне ПЗ доцільно розглядати як єдину екосистему, де офісні пакети забезпечують управлінсько-документаційний контур, а спеціалізовані — інженерний контур (проєктування, моделювання, верифікація, розгортання).

У профільних публікаціях та інженерній практиці підкреслюється роль симуляції як способу зменшити вартість і ризики прототипування: моделі роботів і середовища дозволяють відпрацьовувати конструктивні рішення та алгоритми керування без негайного виготовлення фізичного прототипу. Окремий напрям – порівняльні аналізи симуляторів (наприклад, Webots, CoppeliaSim, Gazebo) з точки зору інтеграції, підтримки датчиків, фізичних рушіїв та придатності до розроблення в ROS-екосистемі.

Паралельно розвиваються відкриті програмні екосистеми для робототехніки. ROS 2 позиціонується як набір бібліотек та інструментів для побудови роботизованих застосунків – від драйверів до алгоритмів і засобів розробника. У результаті найбільш уживаною стає гібридна методологія, де проєктна документація і планування ведуться в офісних інструментах;інженерні моделі розробляються в спеціалізованих пакетах; інтеграція і тестування виконуються в симуляції та/або стендових умовах з аналізом даних.

Ефективність розроблення мехатронних та робототехнічних систем значною мірою залежить від правильно підібраного ланцюга прикладного ПЗ, а не від окремого “найкращого” пакета. Для навчальних і прикладних проєктів доцільно застосовувати єдину систему критеріїв, а саме інтегрованість і формати обміну: STEP/IGES (механіка), Gerber/ODB++ (плати), CSV/JSON (дані), URDF/SDF (робот-моделі), FMI/FMUs (моделі), імпорт-експорт між CAD і симуляторами; відтворюваність; підтримка симуляції та тестування; ліцензія і доступність; поріг входження та сумісність із цільовим “залізом”.

ЦИФРОВІ КАРТИ УКРИТТІВ ЯК ІНСТРУМЕНТ ПОЖЕЖНОЇ ПРОФІЛАКТИКИ В МІСЬКОМУ СЕРЕДОВИЩІ

Мельник І. В.
E-mail: melnyk_iryna@nuczu.edu.ua
Черкаси, Національний університет цивільного захисту України

У сучасних умовах щільної міської забудови пожежна профілактика дедалі більше спирається на інформаційні технології, які забезпечують просторовий аналіз ризиків і підтримку управлінських рішень. Цифрові карти укриттів є спеціалізованим інформаційним продуктом, що поєднує дані про розташування захисних споруд, їх доступність, технічний стан і маршрути підходу/під’їзду. В Україні відповідні набори даних формуються у цифрових реєстрах та відображаються у відкритих онлайн-сервісах [1,2].

Для задач профілактики карта укриттів корисна не лише під час надзвичайної ситуації, а й у режимі планування: вона дає змогу виявляти «білі плями» покриття територій, ділянки зі складною доступністю та типові перешкоди в міському середовищі (вузькі проїзди, тупикові під’їзди, бар’єри). Поєднання шару укриттів із даними про забудову та дорожню мережу дозволяє ранжувати мікрорайони за пріоритетністю профілактичних заходів і підготовки маршрутів інформування населення.

Сучасні дослідження підтверджують ефективність GIS-підходів для оцінювання пожежних ризиків у містах (аналіз доступності, критичних зон, пріоритизація територій) [3]. Практична реалізація таких підходів на базі відкритого ПЗ (QGIS) та відкритих геоданих (OpenStreetMap, відкриті міські реєстри) підсилює відтворюваність результатів, знижує вартість впровадження та спрощує інтеграцію з іншими цифровими сервісами.

У контексті секції «Прикладне програмне забезпечення» доцільно акцентувати типовий робочий процес: підготовка та валідація шарів у QGIS; обмін у GeoPackage/GeoJSON/CSV; підключення до PostgreSQL/PostGIS; застосування плагінів для мережевого аналізу, контролю атрибутів і топології. Офісні пакети (табличні процесори, форми збору даних) використовуються для стандартизації полів, контролю заповнення та формування зведень/звітів; далі дані можуть публікуватися як веб-карти через WMS/WMTS або у вигляді інтерактивних дашбордів [1,2].

Таким чином, цифрові карти укриттів у поєднанні з прикладним ГІС-інструментарієм забезпечують перехід до даноорієнтованої пожежної профілактики у міському середовищі, підвищуючи якість планування, актуальність даних і готовність населення та служб реагування.

Література

[1] ДСНС України. Інформаційна система «Облік та візуалізація фонду захисних споруд цивільного захисту» [Електронний ресурс]. – Режим доступу: https://shelters.dsns.gov.ua.

[2] Дія. Нова послуга в Дії: знаходьте найближче укриття в кілька кліків [Електронний ресурс]. – 2024. – Режим доступу: https://diia.gov.ua.

[3] Thakare K. V., Tajne K. M. A comprehensive review of GIS-based methodologies for urban fire risk assessment // Cureus Journal of Engineering. – 2025. – DOI: 10.7759/s44388-024-02916-y.

KRITA – ЦИФРОВИЙ ЖИВОПИС ТА ІНТЕРАКТИВНЕ МИСТЕЦТВО

Носкова В.В.
E-mail: noskova.viktoriia@ksada.org
Харків, Харківська державна академія дизайну і мистецтв

Еволюція інструментарію для цифрового мистецтва протягом останніх десятиліть демонструє чіткий вектор від загальних графічних редакторів до вузькоспеціалізованих екосистем, що фокусуються на потребах конкретних творчих дисциплін. У цьому контексті Krita посідає унікальне місце як вільне програмне забезпечення з відкритим вихідним кодом (FOSS), яке трансформувалося з амбітного клону растрових редакторів у домінуючу платформу для цифрового живопису, концепт-арту та 2D-анімації [1].

Розуміння сучасного стану Krita неможливе без аналізу її тривалої та подекуди драматичної історії. Проект розпочався в 1998 році в межах спільноти KDE, коли Маттіас Етріх продемонстрував можливість створення графічного інтерфейсу навколо існуючих додатків, обравши об'єктом для демонстрації GIMP [1]. Попри те, що початкові ініціативи не призвели до прямої співпраці з командою GIMP, вони заклали фундамент для створення власного редактора зображень у складі пакету KOffice. Процес ідентифікації проекту пройшов через низку юридичних та бренд-трансформацій: від KImage та KImageShop до Krayon, поки в 2002 році не було обрано назву Krita, що шведською мовою означає «крейда» або «малювати» [2].

Ключовим моментом у розвитку стало стратегічне рішення 2009 року змінити фокус розробки. Спільнота вирішила відмовитися від спроб конкурувати з універсальними редакторами на кшталт Photoshop у сегменті маніпуляції фотографіями, натомість зосередивши всі ресурси на створенні найкращого інструменту для художників, ілюстраторів та концепт-артистів. Це рішення дозволило Krita знайти власну ідентичність та сформувати унікальний набір функцій, які сьогодні вважаються галузевими стандартами в цифровому живописі.

Створення Krita Foundation у 2012 році стало актом інституціоналізації проекту, що дозволило залучати професійних розробників на повну зайнятість. Фундація функціонує як незалежна некомерційна організація, метою якої є надання художникам та студіям усього необхідного для створення цифрового мистецтва. Одним із яскравих прикладів успішності такої моделі є фінансування Дмитра Казакова, чия робота над продуктивністю полотна та двигунами пензлів вивела Krita на рівень готовності до професійного використання [1].

Технічна архітектура Krita базується на використанні мови C++ та фреймворку Qt, що забезпечує високу швидкість виконання операцій та кросплатформність. Програма оптимізована для багатоядерних процесорів та використовує OpenGL або Direct3D 11 для апаратного прискорення полотна [3]. Це критично важливо при роботі з великими полотнами та складними пензлями, де затримка введення повинна бути мінімальною.

На відміну від багатьох безкоштовних інструментів, Krita з самого початку проектувалася з урахуванням професійного управління кольором. Програма підтримує повноцінний колірний менеджмент через бібліотеку LCMS для профілів ICC та OpenColor IO (OCIO) для файлів EXR [4]. Це дозволяє художникам інтегрувати Krita в існуючі студійні пайплайни, забезпечуючи точність передачі кольору між різними пристроями та програмами.

Krita є піонером серед растрових редакторів у підтримці HDR-живопису. Завдяки підтримці колірних просторів з високою бітовою глибиною (до 32-біт на канал), художники можуть працювати з зображеннями, де значення яскравості виходять за межі стандартного динамічного діапазону.

Трансформація Krita в анімаційну студію відбулася завдяки інтеграції таймлайну та інструментів керування ключовими кадрами. Krita фокусується на традиційній покадровій 2D-анімації, надаючи інструменти, які раніше були доступні лише в дорогому спеціалізованому ПЗ [4].

Найбільш революційним доповненням до екосистеми останнього часу став плагін Krita AI Diffusion, який інтегрує потужність Stable Diffusion безпосередньо в робочий процес художника [5]. Цей інструмент не замінює художника, а пропонує нові форми інтерактивної взаємодії з ШІ.

Використання Krita в професійному середовищі підтверджується її впровадженням у студіях рівня Disney через підрядників, таких як Icon Creative, для створення концепт-арту [6]. Особливо показовим є досвід Blender Studio, де Krita використовується як основний інструмент для малювання [7].

Дослідження показують, що Krita демонструє найкращу продуктивність на Linux. Це зумовлено тим, що Krita є «рідним» додатком для середовища KDE, а драйвери планшетів у ядрі Linux часто забезпечують меншу затримку, ніж їхні аналоги на Windows [8]. На macOS ситуація складніша через застарілість підтримки OpenGL з боку Apple, що змушує розробників шукати шляхи переходу на Metal через посередницькі бібліотеки [6].

Krita сьогодні є зрілим інструментом, який успішно поєднує філософію вільного ПЗ із вимогами високобюджетного виробництва. Її здатність до швидкої адаптації, як це видно на прикладі ШІ-плагінів, та відкритість до автоматизації через Python API роблять її не просто графічним редактором, а інтерактивним середовищем, що визначає майбутні стандарти цифрової візуальної культури. Дослідження підкреслює, що для професійного успіху з Krita художнику необхідно виходити за межі простого володіння інструментами малювання, освоюючи навички управління кольором, розуміння архітектури шарів та можливостей скриптингу для кастомізації власного робочого простору [3].

Література

[1] Krita Foundation [Електронний ресурс]. – Режим доступу до ресурса: https://krita.org/en/about/krita-foundation/

[2] History | Krita [Електронний ресурс]. – Режим доступу до ресурса: https://krita.org/en/about/history/

[3] XPPen. Drawing in Krita with a Drawing Tablet: Choosing the Best Tablet for Krita [Електронний ресурс]. – Режим доступу до ресурса: https://www.xp-pen.com/blog/drawing-tablet-for-painting-sketching-krita.html

[4] Features | Krita [Електронний ресурс]. – Режим доступу до ресурса: https://krita.org/en/features/

[5] Krita AI Diffusion - Generative AI For Krita [Електронний ресурс]. – Режим доступу до ресурса: https://kritaaidiffusion.com/

[6] Professional work? [Електронний ресурс]. – Режим доступу до ресурса: https://www.reddit.com/r/krita/comments/1dcrokb/professional_work/

[7] Siddi F. Custom digital painting workflow in Krita [Електронний ресурс]. – Режим доступу до ресурса: https://studio.blender.org/blog/custom-digital-painting-workflow-in-krita/

[8] IncarnaTFs. Which version is faster? [Електронний ресурс]. – Режим доступу до ресурса: https://www.reddit.com/r/krita/comments/9nfzl5/which_version_is_faster_ windows_ or_linux/

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРИ ПРОВЕДЕННІ ВИПРОБУВАНЬ НА ВОДОВІДДАЧУ ВОДОПРОВІДНИХ МЕРЕЖ

Петухова О.А., Трипольська К.С.
E-mail: petukhova olena@nuczu.edu.ua
Черкаси, Національний університет цивільного захисту України

Проведення випробувань водопровідних мереж на водовіддачу є важливою та обов’язковою складовою утримання їх в працездатному стані. За вимогами нормативних документів випробування проводять при прийнятті до експлуатації будь-якого об’єкта, а для зовнішніх мереж ще і один раз кожного року. За результатами випробувань складається акт, в якому зазначаються нормативні витрати води на пожежогасіння для об’єктів, які приєднані

до цієї ділянки водопровідної мережі, спосіб проведення випробувань, значення фактичних витрат води, що були одержані з мережі, висновок про можливість мережі забезпечити подачу необхідних витрат води на пожежогасіння.

Для одержання результатів, на які можна розраховувати для використання при роботі мережі в будь-яку годину доби будь-якої пори року, важливим є правильне виконання кожного етапу проведення випробувань: підготовчого (ознайомлення з результатами попередніх випробувань, вибір часу та місця для проведення випробувань, визначення нормативних витрат води на пожежогасіння, кількості пожежних гідрантів та пожежно-технічного обладнання); проведення випробувань (правильне використання обладнання, зняття показань приладів та перерахунок, у разі необхідності, виміряних величин у водовіддачу); оцінка результатів (порівняння фактичної водовіддачі з нормативними витратами води на пожежогасіння, формулювання висновку про можливість мережі подати необхідну кількість воли на пожежогасіння, а у випадку необхідності - формулювання пропозицій з покращення ситуації, складання акту) [1].

Сучасні інформаційні технології можуть допомогти спростити підготовку до проведення випробувань, оцінку та оформлення результатів. Для цього в Національному університеті цивільного захисту разом з викладанням теоретичних матеріалів з цієї теми частина навчального матеріалу відпрацьовується на лабораторних та контрольних роботах, на яких використовуються тестування за темою, відпрацювання практичної складової за допомогою навчально-тестового симулятора, виконання контрольної роботи у гугл-формах [2, 3].

В порівнянні з діями, які можна відпрацювати підключаючи рукави до пожежних гідрантів або кран-комплектів, пускаючи воду та роблячи вимірювання приладами, робота з тестами або симулятором може надавати менш практичних навичок, але саме така робота забезпечує відпрацювання всіх етапів проведення випробувань та виключення багатьох помилок.

Так, проводячи тестування, можливо сконцентрувати увагу здобувачів на необхідність уважної підготовки до проведення випробувань, що включає вивчення нормативних витрат води на пожежогасіння, що в свою чергу впливає на кількість та характеристики пожежно-технічного обладнання, на необхідність правильного вибору часу та місця проведення випробувань, на різноманітність існуючого обладнання та особливості його використання. Важливим є те, що на сьогодні створення тестів можливо у гугл-формах, які є безкоштовним онлайн-інструментом, що дає можливість формувати запитання у вигляді від простих текстових полів до складних шкал, сіток і випадаючих списків, з персоналізацією за допомогою тем, кольорів, власних зображень або готових шаблонів Google, з миттєвим збиранням відповідей у вкладці «Відповіді» та можливістю їх експортування в Google Таблиці для подальшого аналізу, до того ж є можливість призначити бали за правильні відповіді та налаштувати автоматичну перевірку знань, а також редагування форми кількома користувачами одночасно в режимі реального часу (рис.1).

Виконання контрольної роботи в формі тестів дозволяє з успіхом використовувати всі можливості гугл-форм та при цьому відпрацювати роботу з таблицями нормативних документів та реалізувати всі три етапи проведення випробувань, використовуючи різні прилади (рис.2).

Наступним кроком у вивченні теми випробувань на водовіддачу водопровідних мереж є виконання лабораторної роботи за допомогою навчально-тестового симулятора “Водовіддача”. Метою цієї роботи є навчити здобувачів виконувати всі етапи випробувань, при цьому задача виконується чотири рази для різних типів будівель, біля яких прокладений водопровід, та з використанням різних приладів. Кожна дія здобувача оцінюється та у разі помилки надається можливість змінити прийняте рішення, але остаточна оцінка при цьому зменшується. Таким чином, відпрацьовуючи на симуляторі випробування, здобувач має можливість урізноманітнити свої навички не лише з роботи з різними приладами, а ще і звертаючи увагу на визначення нормативних витрат для будівель різного призначення, а

також навички з особливостей перерахунку виміряних величин у витрати води. Якщо симулятор буде використовувати практичний працівник, він може не лише заздалегідь якісно підготуватись до проведення випробувань, віртуально відпрацьовуючи різні прилади, а і використовувати симулятор на підготовчому та завершальному етапах випробувань.

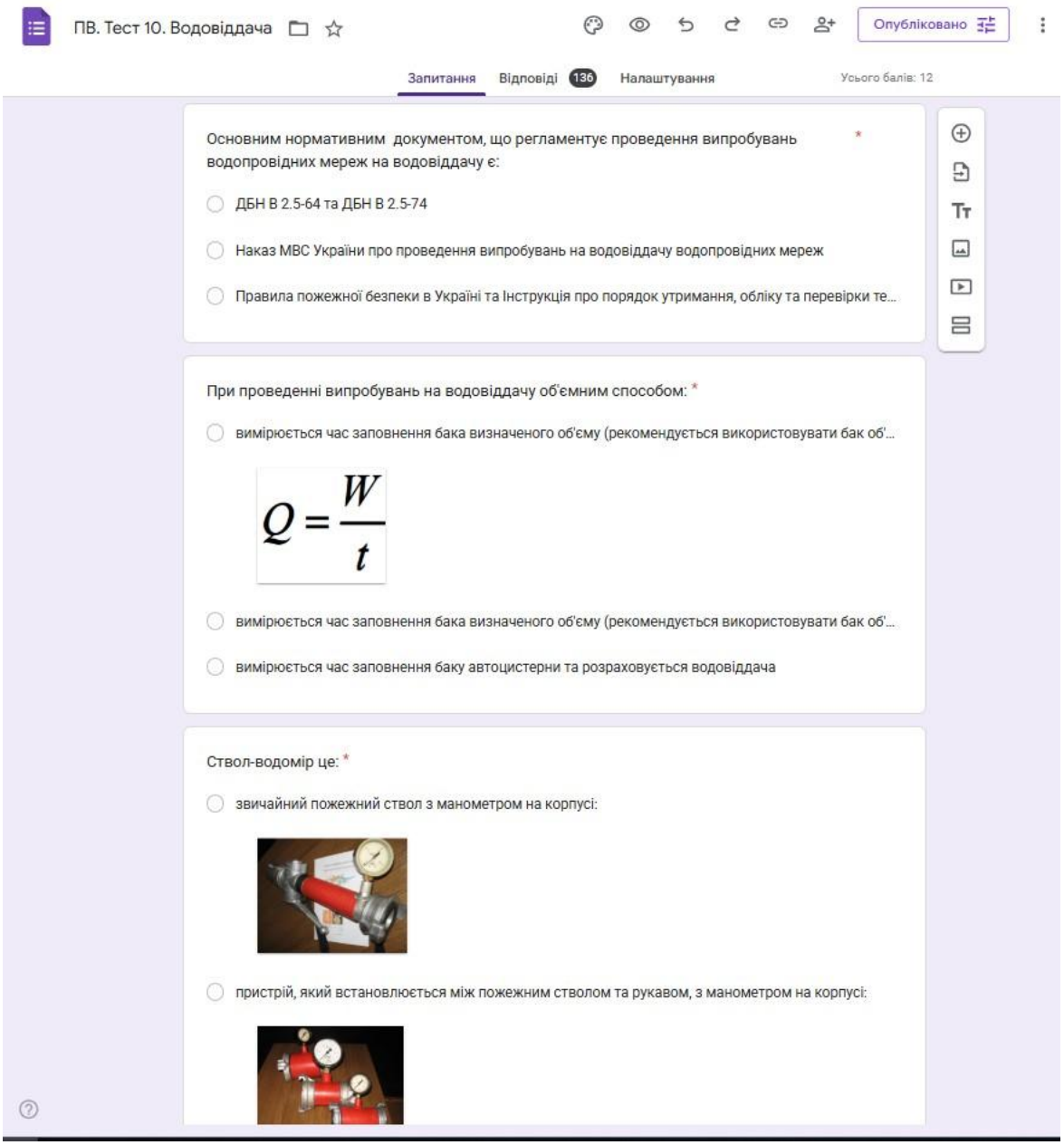


Рис.1 - Приклад гугл-тесту для перевірки знань з теми “Випробування на водовіддачу водопровідних мереж”

Проведення випробувань водопровідних мереж на водовіддачу виконуються працівниками практичних підрозділів ДСНС України і їх якісна підготовка до проведення таких дій є запорукою того, що будуть визначені само фактичні витрати води, які дійсно можливо буде використовувати для гасіння пожежі. Запропоновані тестування, контрольна робота та робота з навчально-тестовим симулятором можуть бути використані для підготовки практичних працівників, а також все може бути доповнено презентаціями з теоретичними основами за темою.

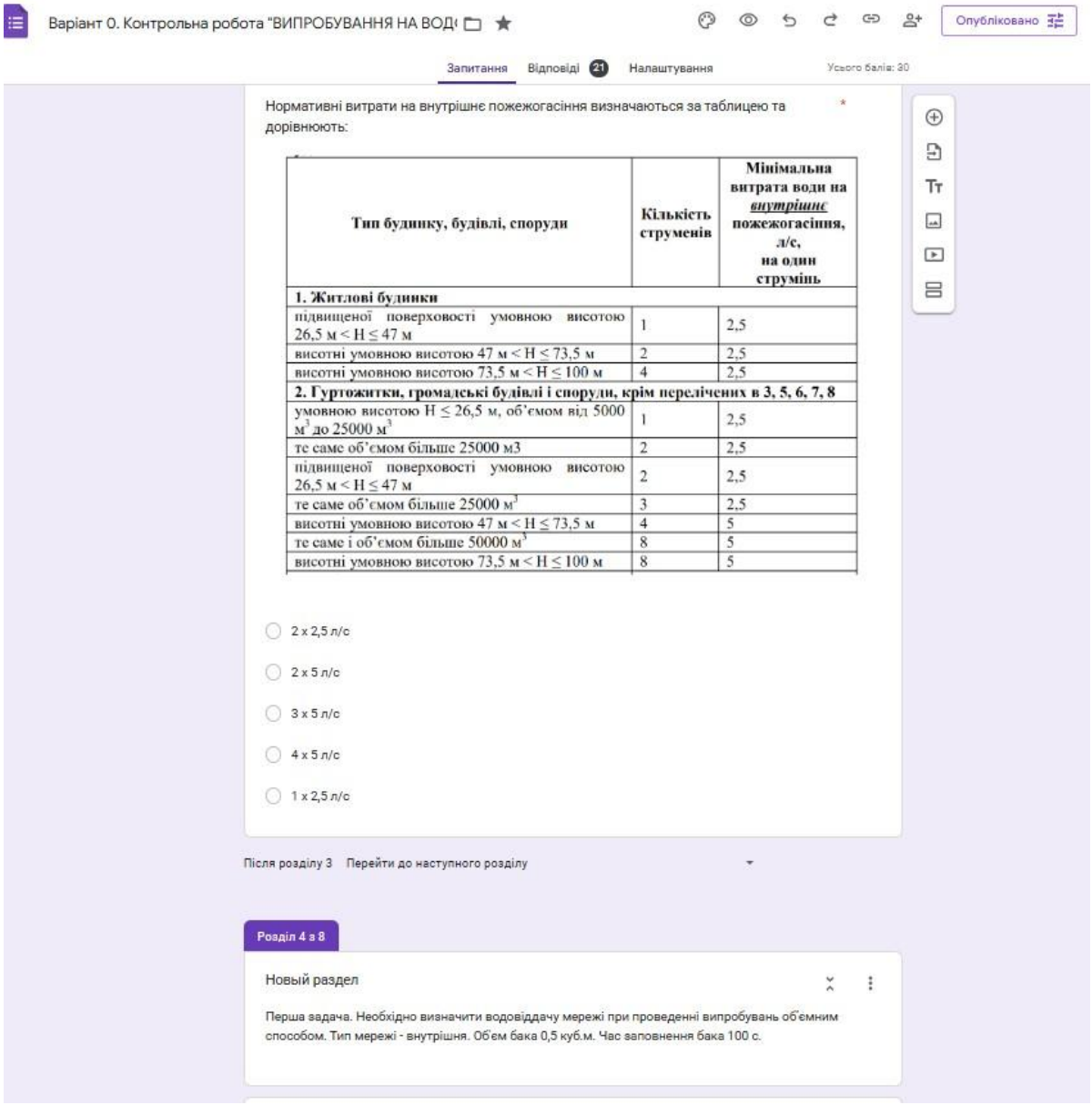


Рис.2 - Приклад реалізації контрольної роботи в гугл-формах

Таким чином, використання інформаційних технологій при проведенні випробувань водопровідних мереж на водовіддачу є доцільним та ефективним при навчанні для здобувачів вищої освіти та в практичній діяльності для підрозділів ДСНС України.

Література

[1] Протипожежне водопостачання : Підручник / О.А. Петухова, В.А. Андронов, С.А. Горносталь, Р.Е. Черепаха. - Х.: Друкарня Мадрид, 2022 . – 280 с.

[2] Петухова О.А., Добринська В.Є., Кулеш Д.П. Способи підвищення ефективності навчання з наукового напрямку цивільна безпека // Безпека людини у сучасних умовах: матеріали IX Міжнародної науково-методичної конференції, Міжнародної наукової конференції EAS – Харків: НТУ «ХПІ», 2022. (http://repositsc.nuczu.edu.ua/handle/123456789/16422)

[3] Петухова О.А. Шляхи інтеграції професійної освітньої компоненти до європейського освітнього простору. // Методологія сучасних наукових досліджень: матеріали Ювілейної XX Міжнародної науково-практичної конференції. – Харків: ХНПУ імені Г.С. Сковороди, 2024. – С. 88-91 http://repositsc.nuczu.edu.ua/handle/123456789/19904

ОНЛАЙН-СИМУЛЯТОРИ PHET INTERACTIVE SIMULATIONS ТА TINKERCAD CIRCUITS У ВИКЛАДАННІ ТЕХНІЧНИХ ДИСЦИПЛІН

Печеневська О.О.
E-mail: olganevsk2404@gmail.com
Харків, Харківський радіотехнічний фаховий коледж

Невід'ємною складовою сучасного освітнього процесу є використання цифрових технологій, які забезпечують інтерактивність, наочність і доступність навчання. В умовах дистанційного навчання для формування практичних навичок при вивченні технічних дисциплін незамінними стають спеціалізовані пакети прикладного програмного забезпечення. Саме ці цифрові інструментарії дозволяють проводити віртуальні лабораторні роботи, сприяють активній пізнавальній діяльності здобувачів освіти, формують дослідницькі та цифрові компетентності. Серед великої кількості застосунків та онлайн-сервісів для симуляції різноманітних процесів в пристроях аналогової та цифрової техніки, все частіше викладачі надають перевагу саме онлайн-симуляторам. Вони не потребують встановлення спеціального програмного забезпечення, доступні з будь-якого пристрою і не залежать від операційної системи (Windows, Linux, macOS), що усуває проблеми сумісності.

Розглянемо два популярні сервіси - PhET Interactive Simulations (phet.colorado.edu) та Tinkercad Circuits, що широко використовуються при вивченні таких дисциплін, як теорія кіл, електроніка, мікросхемотехніка тощо [1].

PhET Interactive Simulations — це освітній проєкт Університету Колорадо (США), який пропонує велику бібліотеку безкоштовних інтерактивних симуляцій, що дозволяють в ігровій формі засвоїти фундаментальні закони [2]. Tinkercad Circuits є частиною платформи Autodesk Tinkercad, що дозволяє проєктувати, моделювати та програмувати електричні схеми, включаючи Arduino.

Обидва сервіси безкоштовні, мають простий та інтуїтивно зрозумілий інтерфейс, дозволяють спостерігати за процесами в схемах в режимі реального часу, перевіряти й налагоджувати їх працездатність. Але PhET має досить обмежений набір інструментів і більш орієнтований на концептуальне розуміння фізичних явищ, тоді як Tinkercad орієнтований на практичне застосування знань, має велику бібліотеку електронних компонентів, вбудований редактор Arduino, можливість хмарного збереження власних проєктів, тому надає широкі можливості для проєктної та дослідницької діяльності [3].

Онлайн-симулятори PhET Interactive Simulations і Tinkercad Circuits є ефективними цифровими інструментами, які доповнюють традиційне навчання. Їх використання сприяє підвищенню якості освітнього процесу, розвитку критичного мислення та формуванню цифрових та професійних компетентностей.

Враховуючи освітню спрямованість, функціональність, складність та обмеження при використанні PhET більш ефективний для пояснення та закріплення теоретичного матеріалу,

Tinkercad – для виконання лабораторних робіт та створення власних проєктів.

Саме комплексне застосування обох платформ дозволить створити сучасне освітнє середовище, орієнтоване на активне навчання, дослідження та практичну діяльність.

Література

[1] Mashami, R. A., Ahmadi, Kurniasih, Y., & Khery, Y. (2023). Use of PhET Simulations as a Virtual Laboratory to Improve Students' Problem Solving Skills. Jurnal Penelitian Pendidikan IPA,9(12), 11455–11465. https://doi.org/10.29303/jppipa.v9i12.6549

[2] Pranata, O. D. (2024). Physics education technology (PhET) as a game-based learning tool: A quasi-experimental study. Pedagogical Research, 9(4), em0221. https://doi.org/10.29333/pr/15154

[3] Golubev, Leonty P; Tkach, Mikhailo M; Makatora, Dmytro A. (2023) Using Tinkercad to support online the laboratory work on the design of microprocessor systems at technical university. Information Technologies and Learning Tools, Vol 93, №1

ADOBE PREMIERE PRO ЯК СПЕЦІАЛІЗОВАНИЙ ПАКЕТ ДЛЯ ОБРОБКИ ТА МОНТАЖУ ВІДЕО

Птухін М.Ю.
Керівник: Чайка А.В.
E-mail: mikita.ptu@gmail.com
Харків, Комунальний заклад «Харківський ліцей № 153 Харківської міської ради»

У сучасному цифровому середовищі відеоконтент став одним із провідних способів передачі інформації, навчання, реклами та розваг. Соціальні мережі, відеоплатформи, освітні курси та медіапроєкти активно використовують відеоформат для ефективної комунікації з аудиторією. У зв'язку з цим зростає потреба у використанні професійних прикладних програмних засобів, що дозволяють здійснювати якісний монтаж, обробку та створення сучасного відеоконтенту. Одним із найпопулярніших спеціалізованих програмних продуктів у цій сфері є Adobe Premiere Pro.

Adobe Premiere Pro — це професійне програмне забезпечення для нелінійного монтажу відео, розроблене компанією Adobe. Програма широко використовується у кіноіндустрії, телебаченні, створенні рекламних роликів, освітніх відео та контенту для соціальних мереж і відеохостингів. Завдяки широкому функціоналу та інтеграції з іншими продуктами Adobe, такими як After Effects, Photoshop та Audition, користувач отримує комплексне середовище для роботи з мультимедійними матеріалами.

Однією з ключових переваг Adobe Premiere Pro є професійний, але логічно структурований інтерфейс. Робоче середовище складається з кількох основних панелей: таймлайну для монтажу відео, вікна попереднього перегляду, панелі інструментів та медіабібліотеки. Користувач може налаштовувати розташування елементів інтерфейсу відповідно до власних потреб, що значно підвищує зручність роботи та продуктивність монтажу [1].

Функціональні можливості програми охоплюють широкий спектр операцій із відео та аудіо. Користувач може здійснювати обрізку та склеювання відеофрагментів, додавати переходи, відеоефекти та текстові титри, виконувати корекцію кольору за допомогою інструментів Lumetri Color, а також синхронізувати аудіо та відео. Програма підтримує роботу з багатошаровими відеодоріжками, що дозволяє створювати складні монтажні проєкти різного рівня.

Важливою особливістю Adobe Premiere Pro є підтримка великої кількості відео- та аудіоформатів без необхідності попередньої конвертації файлів. Програма дозволяє працювати з матеріалами високої якості, від Full HD та вище, що робить її універсальним інструментом для професійного відеовиробництва. Крім того, завдяки апаратному прискоренню та оптимізації роботи з відеокартою забезпечується ефективна обробка великих проєктів, хоча для стабільної роботи потрібні достатньо потужні технічні ресурси комп'ютера.

Adobe Premiere Pro має низку переваг, серед яких професійний рівень монтажу, гнучкі налаштування інструментів, підтримка сучасних стандартів відео та інтеграція з екосистемою Adobe. Водночас програма має і певні обмеження, зокрема платну ліцензію, високі системні вимоги та складність освоєння для початківців без попереднього навчання [2].

Також основні можливості програми:

1. Робота з багатокамерним монтажем.

Adobe Premiere Pro підтримує функцію багатокамерного монтажу, що дозволяє працювати з відео, записаним одночасно з кількох камер. Це особливо корисно під час створення інтерв'ю, концертних виступів, освітніх лекцій або подій. Користувач може синхронізувати відеодоріжки за звуком або часовим кодом та швидко перемикатися між ракурсами без необхідності складного ручного редагування.

2. Робота з графікою та анімацією.

Програма дозволяє створювати анімовані титри, інтерактивні текстові елементи та графічні вставки. Завдяки інтеграції з Adobe After Effects користувач може створювати складні візуальні ефекти та імпортувати їх безпосередньо в монтажний проєкт. Це значно розширює можливості відеомонтажу та дозволяє створювати професійний мультимедійний контент.

3. Автоматизація процесів за допомогою штучного інтелекту.

Сучасні версії Adobe Premiere Pro містять інструменти на основі штучного інтелекту Adobe Sensei. Вони дозволяють автоматично виконувати розпізнавання мови, створювати субтитри, покращувати звук та оптимізувати колір відео. Такі функції значно скорочують час обробки матеріалів та роблять монтаж більш ефективним навіть для користувачів із невеликим досвідом.

4. Використання у навчальному процесі.

Adobe Premiere Pro може ефективно застосовуватися в освітній діяльності для створення навчальних відео, презентацій, соціальних роликів та творчих проєктів. Робота з відеомонтажем сприяє розвитку цифрової грамотності, креативності, медіаграмотності та навичок командної роботи. Здобувачі освіти можуть створювати власні відеопроєкти, що підвищує їх мотивацію до навчання.

5. Співпраця та командна робота над проєктами.

Програма підтримує спільну роботу над відеопроєктами через хмарні сервіси Adobe Creative Cloud. Кілька користувачів можуть працювати над різними частинами одного проєкту, обмінюватися файлами та швидко вносити зміни. Це робить Adobe Premiere Pro ефективним інструментом для студійної роботи, освітніх команд та медіапроєктів [3].

Отже, Adobe Premiere Pro є сучасним спеціалізованим прикладним програмним забезпеченням для професійного монтажу та обробки відео, яке поєднує широкий функціонал із гнучкістю налаштування робочого середовища. Програма дозволяє створювати відеоконтент різної складності — від коротких роликів для соціальних мереж до повноцінних медіапроєктів і навчальних відеоматеріалів. Завдяки підтримці сучасних форматів, інтеграції з іншими програмами Adobe та використанню технологій штучного інтелекту значно підвищується ефективність роботи користувачів.

Важливою перевагою Adobe Premiere Pro є можливість застосування як у професійній діяльності, так і в освітньому середовищі. Робота з відеомонтажем сприяє розвитку творчого мислення, цифрової грамотності, навичок аналізу інформації та створення медіаконтенту. Незважаючи на певні складнощі освоєння та високі технічні вимоги, програма залишається одним із найпотужніших інструментів для сучасного відеовиробництва.

Таким чином, Adobe Premiere Pro виступає універсальним середовищем для реалізації творчих і професійних завдань, розвитку цифрових компетентностей та формування практичних навичок роботи з мультимедійними технологіями. Використання цього програмного забезпечення відкриває широкі можливості для створення якісного відеоконтенту та самовираження у сучасному інформаційному просторі.

Література

[1] Adobe Premiere Pro – офіційна сторінка продукту [Електронний ресурс]. – Режим доступу: https://www.adobe.com/ua/products/premiere.html

[2] Adobe Premiere Pro User Guide. Basic Workflow [Електронний ресурс]. – Режим доступу: https://helpx.adobe.com/ua/premiere-pro/using/basic-workflow.html

[3] Основні можливості Adobe Premiere Pro [Електронний ресурс]. – Режим доступу: https://www.adobe.com/ua/products/premiere/features.html

АНАЛІЗ МОЖЛИВОСТЕЙ ТА СТРАТЕГІЧНИХ ПЕРЕВАГ ВИКОРИСТАННЯ СИСТЕМИ МОНІТОРИНГУ MIKROTIK THE DUDE В СУЧАСНИХ МЕРЕЖЕВИХ ІНФРАСТРУКТУРАХ

Свинаренко М.С., Литвиненко Є.М.
E-mail: svynarenko.maksym@ksada.org
Харків, Харківська державна академія дизайну і мистецтв

Система моніторингу та управління мережею The Dude, розроблена компанією MikroTik, представляє собою спеціалізоване програмне рішення, орієнтоване на автоматизацію процесів візуалізації, діагностики та контролю мережевих середовищ різного масштабу [1]. У сучасних умовах, коли складність корпоративних мереж постійно зростає, інтеграція подібних інструментів стає не просто допоміжною функцією, а критичною необхідністю для забезпечення стабільності бізнес-процесів [2]. Основна архітектурна перевага The Dude полягає в її глибокій інтеграції з операційною системою RouterOS, що дозволяє адміністраторам використовувати ресурси маршрутизаторів безпосередньо для збору та зберігання аналітичних даних, мінімізуючи потребу в зовнішніх обчислювальних потужностях [3].

Однією з найбільш помітних переваг The Dude є її здатність до автоматичної побудови топологічних карт мережі. Процес автоматичного виявлення (Discovery) використовує комбінацію протоколів для ідентифікації пристроїв, їхніх характеристик та зв'язків між ними [4]. Система здатна сканувати задані діапазони IP-адрес, визначати типи сервісів, що запущені на пристроях, та автоматично розміщувати їх на графічній схемі.

Візуалізація в The Dude не обмежується простим відображенням іконок [5]. Карти мережі є динамічними об'єктами, які в реальному часі відображають стан кожного пристрою та каналу зв'язку. Використання кольорової індикації (зелений — норма, оранжевий — нестабільність, червоний — збій) дозволяє персоналу оперативного центру моніторингу (NOC) миттєво реагувати на аномалії.

Ефективна організація великих мереж досягається за допомогою вкладених карт (Submaps). Це дозволяє створювати глобальну карту країни або міста, де кожен об'єкт є переходом до детальної схеми конкретного вузла або будівлі [5]. Такий підхід значно знижує когнітивне навантаження на адміністратора та дозволяє структурувати моніторинг відповідно до фізичної або логічної топології мережі.

Важливим аспектом налаштування карт є встановлення залежностей між пристроями (Dependencies). У типовій мережевій топології вихід з ладу магістрального комутатора призводить до втрати зв'язку з усіма пристроями, що підключені через нього. Без налаштування залежностей система генеруватиме сотні сповіщень про збій кожного окремого пристрою. При встановленні залежності "батьківський пристрій — дочірні пристрої", The Dude інтелектуально пригнічує другорядні сповіщення, вказуючи лише на першоджерело проблеми.

Графічне представлення каналів зв'язку (Links) дозволяє в реальному часі спостерігати за завантаженням інтерфейсів. Адміністратори можуть налаштовувати підписи до ліній зв'язку, відображаючи швидкість трафіку, відсоток використання смуги пропускання або кількість помилок на портах. Підтримка SVG-іконок дозволяє створювати професійні схеми, що відповідають корпоративним стандартам візуалізації.

The Dude не обмежується перевіркою доступності пристроїв через ICMP-запити (Ping). Основна аналітична потужність системи реалізується через підтримку протоколу SNMP (Simple Network Management Protocol) версій v1, v2c та v3 [2]. Це дозволяє зчитувати практично будь-який параметр з мережевого обладнання, серверів під управлінням Windows або Linux, принтерів та систем безперебійного живлення.

Для вирішення специфічних завдань моніторингу The Dude надає інструментарій створення власних функцій та проб. Функція — це програмний код, який отримує дані

(зазвичай через SNMP OID), обробляє їх і повертає значення. Проба використовує ці значення для прийняття рішення про статус об'єкта [6].

Для пристроїв виробництва MikroTik система The Dude пропонує розширені можливості моніторингу без використання SNMP [6]. Використовуючи нативний протокол управління, система може отримувати детальну інформацію про реєстрацію бездротових клієнтів, стан тунелів VPN, навантаження на окремі ядра процесора та версії встановлених пакетів прошивки. Це забезпечує безпрецедентний рівень контролю в однорідних мережах MikroTik, дозволяючи адміністратору виконувати оновлення ПЗ на сотнях пристроїв безпосередньо з консолі моніторингу.

Ефективність системи моніторингу визначається не лише її здатністю виявляти проблеми, але й швидкістю інформування відповідального персоналу [2]. The Dude пропонує гнучку систему налаштування повідомлень (Notifications), яка підтримує різноманітні канали комунікації.

Одним із найбільш затребуваних сценаріїв у сучасних IT-відділах є інтеграція моніторингу з месенджерами. Завдяки можливості виконання скриптів RouterOS у відповідь на події, The Dude може надсилати деталізовані звіти в Telegram або Slack [7].

The Dude дозволяє реалізувати елементи самокерованої мережі. Наприклад, при виявленні втрати пакетів до віддаленої точки доступу, система може автоматично виконати скрипт для перезавантаження порту живлення PoE на комутаторі. Це значно скорочує показник середнього часу відновлення (MTTR) та звільняє персонал від виконання рутинних операцій з виправлення дрібних збоїв. Крім того, система може бути налаштована на зміну конфігурації маршрутизації (наприклад, перемикання на резервного провайдера), якщо проба зафіксувала деградацію каналу за межі допустимих значень затримки (Latency) або джиттера (Jitter).

The Dude від MikroTik залишається потужним, гнучким та, що не менш важливо, економічно вигідним рішенням для моніторингу сучасної IT-інфраструктури. Її переваги в автоматизації візуалізації та нативній інтеграції з RouterOS роблять її ідеальним вибором для мережевих інженерів та провайдерів.

Незважаючи на конкуренцію з боку великих Enterprise-платформ, The Dude продовжує утримувати свою нішу завдяки унікальному поєднанню простоти та глибини налаштування, залишаючись "швейцарським ножем" у руках досвідченого мережевого адміністратора. Враховуючи розвиток RouterOS v7, можна з упевненістю стверджувати, що актуальність цього інструменту в найближчі роки лише зростатиме.

Література

[1] Mikrotik Network Management Guide [Електронний ресурс]. – Режим доступу до ресурса: https://www.scribd.com/doc/309191466/The-Dude-Manual-En-ingles.

[2] Monitoring Large Scale Network by The Dude [Електронний ресурс]. – Режим доступу до ресурса: https://mum.mikrotik.com/presentations/BD18/presentation_5311_1526347573.pdf.

[3] How to Install DUDE Server on MikroTik [Електронний ресурс]. – Режим доступу до ресурса: https://monovm.com/blog/how-to-install-dude-on-mikrotik/.

[4] MikroTik CHR: How to set-up The Dude Monitoring [Електронний ресурс]. – Режим доступу до ресурса: https://www.bgocloud.com/knowledgebase/36/mikrotik-chr-how-to-setup-the-dude-monitoring.html.

[5] The Dude Device Map Overview [Електронний ресурс]. – Режим доступу до ресурса: https://www.scribd.com/document/120950416/Manual-the-dude.

[6] The functions in the Mikrotik Dude [Електронний ресурс]. – Режим доступу до ресурса: https://mivilisnet.wordpress.com/2016/07/15/the-functions-in-the-mikrotik-dude/.

[7] Manual - The Dude v6 - Dude Telegram Example [Електронний ресурс]. – Режим доступу до ресурса: https://www.scribd.com/document/548720034/Manual-The-Dude-v6-Dude-Telegram-Example-MikroTik-Wiki.

ВИКОРИСТАННЯ ВІЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА GOOGLE MAPS API ДЛЯ СТВОРЕННЯ ІНТЕРАКТИВНИХ КАРТ ЗОВНІШНЬОГО ПРОТИПОЖЕЖНОГО ВОДОПОСТАЧАННЯ

Сіпко О.В., Тищенко Б.М.
e-mail:sipko_oleksandr@nuczu.edu.ua
Черкаси, Національний університет цивільного захисту України

Актуальність. Забезпечення оперативного доступу до актуальних даних про джерела зовнішнього протипожежного водопостачання є важливою передумовою ефективного реагування підрозділів ДСНС на пожежі. У сучасних умовах щільної міської забудови вирішальне значення мають точність просторової прив'язки пожежних гідрантів, наявність достовірної інформації про їх технічний стан та можливість швидкого визначення найближчих вододжерел з урахуванням транспортної доступності. Традиційні паперові схеми та розрізнені відомчі переліки не забезпечують необхідної оперативності та актуальності даних.

Метою роботи є обґрунтування архітектури створення інтерактивних карт зовнішнього протипожежного водопостачання на основі вільного програмного забезпечення з можливістю інтеграції сервісів Google Maps через Google Maps API. Для досягнення мети поставлено завдання визначити мінімальний набір атрибутивних даних пожежних гідрантів, описати програмний стек для підготовки, зберігання та публікації геоданих, а також проаналізувати практичні можливості використання таких карт у діяльності підрозділів ДСНС.

Матеріали та методи. У роботі використано підхід, що ґрунтується на застосуванні відкритих геоінформаційних технологій. Як настільну ГІС використано QGIS для створення та редагування шарів пожежних гідрантів і контролю якості даних [1,2]. Централізоване зберігання та оброблення просторової інформації здійснюється за допомогою СУБД PostgreSQL з розширенням PostGIS [3]. Для публікації вебкарти застосовано бібліотеку Leaflet, що забезпечує відображення шарів, фільтрацію та взаємодію з атрибутами об'єктів [4].

Модель даних пожежного гідранта включає такі основні атрибути: унікальний ідентифікатор, координати розташування, тип гідранта, технічний стан, діаметр умовного проходу, наявність та стан покажчика, доступність для під'їзду пожежної техніки, дату останнього огляду та балансоутримувача. Така структура дозволяє уніфікувати облік гідрантів і створює передумови для автоматизованого аналізу стану системи водопостачання.

Збір та актуалізація даних передбачають поєднання відомчої інформації з результатами польових обстежень. Для цього можуть використовуватися мобільні інструменти, сумісні з QGIS, які забезпечують офлайн-збір даних, фотофіксацію та подальшу синхронізацію з центральною базою [5]. Регулярне оновлення відомостей є необхідною умовою підтримання достовірності інтерактивної карти.

Інтеграція з Google Maps API використовується як допоміжний інструмент для навігації та орієнтування. Зокрема, Maps JavaScript API дозволяє реалізувати побудову маршрутів до обраних пожежних гідрантів і візуалізацію дорожньої мережі. При цьому тематичні шари з даними про вододжерела зберігаються у власній інфраструктурі, що відповідає вимогам безпеки та контролю доступу.

Результати та обговорення. Практичне застосування інтерактивних карт у діяльності підрозділів ДСНС показує доцільність їх використання на етапах підготовки, оперативного реагування та контролю технічного стану гідрантів. Карта дозволяє швидко визначати найближчі справні вододжерела, зменшувати втрати часу на водозабір і підвищувати обґрунтованість управлінських рішень щодо ремонту та утримання елементів системи зовнішнього протипожежного водопостачання.

Висновки. Запропонований підхід демонструє, що використання вільного програмного забезпечення у поєднанні з сервісами Google Maps забезпечує ефективне

створення та експлуатацію інтерактивних карт пожежних гідрантів. Такі рішення є економічно доцільними, масштабованими та придатними для практичного використання підрозділами ДСНС з метою підвищення оперативності реагування на пожежі.

Література

[1] ДБН В.2.5-64:2012. Внутрішній водопровід та каналізація. Основні положення.
[2] ДСТУ 2272-06:2010. Пожежна техніка. Терміни та визначення основних понять.
[3] Кодекс цивільного захисту України.
[4] Офіційний сайт ДСНС України. https://dsns.gov.ua (дата звернення: 05.02.2026).
[5] QGIS Україна. Спільнота користувачів геоінформаційних систем. https://qgis.org/uk/site/ (дата звернення: 05.02.2026).

РОЗРОБКА ПРОГРАМИ ДЛЯ ПОБУДОВИ РЕГРЕСІЙНИХ МОДЕЛЕЙ З МЕТРИК ОБ'ЄКТНО-ОРІЄНТОВАНИХ ПРОГРАМНИХ ПРОЄКТІВ

Татаренко М.А.
Керівник: Макарова Л.М.
E-mail: multi3volume@gmail.com, lidiia.makarova@nuos.edu.ua
Миколаїв, Національний університет кораблебудування імені адмірала Макарова

Побудова регресійних моделей займає важливе місце в статистичному аналізі даних. Одне із застосувань в інженерії програмного забезпечення (ПЗ) полягає в оцінюванні його розміру та визначення якості [1-3]. Для цього застосовуються кількісні показники ПЗ - метрики ПЗ. Для об'єктно орієнтовних програмних проєктів - застосовується набір метрик Чідамбера та Кемерера (Chidamber & Kemerer, CK Metrics) [4-5].

Отримання числових значень вибраних метрик і є основою для збирання їх числових значень в набір даних перед побудовою регресійної моделі. Для отримання числових значень вище зазначених метрик існують різні інструменти. Наприклад для ПЗ написаного на мові програмування Java є достатньо великий вибір інструментів [6].

Зазвичай дані з метрик ПЗ не розподілені за нормальним законом, що робить дуже обмежену можливість використання лінійних регресійних моделей. В такому випадку переходять до застосування нелінійних регресійних моделей.

Процес побудови нелінійних регресійних моделей часто здійснюється за допомогою табличного процесора MS Excel з пакету MS Office. Це може бути оптимальним варіантом для побудови однієї, або декількох моделей на невеликому наборі даних (приблизно п'ятдесяти точок даних) із застосуванням відносно простих нормалізуючих перетворень. Під точкою даних мається на увазі зібрані числові значення метрик по одному, окремому програмному проєкту. Але якщо іде мова про побудову великого числа нелінійних регресійних моделей, до того ж на наборах даних, де їх розмір перевищує сто і більше точок даних, в рамках окремого дослідження використання подібних рішень на базі MS Excel є достатньо ресурсовитратною задачею з точки зору прикладаних зусиль. Основні недоліки при використанні MS Excel: висока вірогідність помилок при ручних розрахунках та високі затрати часу.

Метою роботи є розробка програми для автоматизації побудови нелінійних регресійних моделей з метрик об'єктно орієнтованих програмних проєктів з використанням нормалізуючих перетворень, знаходження та відкидання викидів.

Запропоновано програмне рішення для автоматизації побудови ряду нелінійних регресійних моделей на основі вибраного набору даних будь якого розміру - Non Linear Regression Model Builder - nlrmb. Дана програма написана мовою програмування R. Дозволяє виконувати в автоматичному режимі побудову трьох-факторних нелінійних регресійних моделей на основі наступних нормалізуючих перетворень: десяткового логарифма, перетворення Бокса-Кокса, перетворення Джонсона.

Після побудови кожної моделі виводяться: коефіцієнти лінійної моделі, оцінки нормалізуючого перетворення, параметри якості моделі. Також надаються результати

прогнозування моделі, границі довірчих інтервалів та інтервалів прогнозування в табличній формі.

Особливості програмного рішення nlrmb:

– Побудова трьох-факторних нелінійних регресійних моделей;

– Підтримка нормалізуючих перетворень:

– десяткового логарифма;

– перетворення Бокса-Кокса;

– перетворення Джонсона сімейства S_B;

– Робота з одним набором даних;

– Розділення одного набору даних на тренувальну (train) та тестову (test) вибірки та робота з ними;

– Отримання основних характеристик набору даних в табличному та графічному вигляді (мінімум, максимум, середньоквадратичне відхилення, асиметрія, ексцес);

– Перевірка на мультиколінеарність;

– Перевірка на багатовимірну нормальність за тестом Мардіа;

– Побудова моделей;

– Табличне представлення результатів прогнозування, границь довірчих інтервалів та інтервалів прогнозування (див. рис. 1);

– Представлення результатів розрахунків:

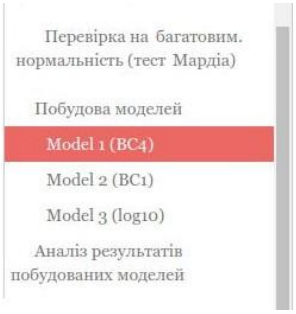
– Кількість викидів для кожної моделі (див. рис. 2);

– Висновки по якості побудованих моделей (див. рис. 3);

– Висновки по результатам MRE та MMRE побудованих моделей;

– Висновки по результатам довірчих інтервалів та інтервалів прогнозування.

Результати розрахунків генеруються у вигляді звіту у форматі html, який можна переглядати в будь якому веб-браузері.



Результати прогнозування (Y_pred), границі інтервалів прогнозування (pi) та довірчих інтервалів (ci)

	n	Y	Y_pred	ci_lb	ci_ub	pi_lb	pi_ub
1	1	17.836	13.2849	12.7164	13.8787	9.7921	18.0174
2	2	187.777	157.4204	145.3062	170.5405	115.4917	214.4959
4	4	1.515	1.2910	1.2113	1.3760	0.9459	1.7615
5	5	10.849	10.0997	9.2636	11.0110	7.3756	13.8249
6	6	7.006	5.4031	5.1634	5.6540	3.9777	7.3369
7	7	0.287	0.2368	0.2188	0.2562	0.1726	0.3248
9	9	0.916	1.2029	1.1310	1.2794	0.8816	1.6408
10	10	21.854	20.3141	19.3887	21.2835	14.9738	27.5497
11	11	52.230	59.9475	56.2090	63.9336	44.0972	81.4669

Рисунок 1 - Табличне представлення результатів прогнозування, границь довірчих інтервалів та інтервалів прогнозування

Кількість викидів для кожної моделі

Type	M1_BC4	M2_BC1	M3_log10	Common
n_start	137	137	137	137
SMD	44, 133, 73, 122, 134, 70, 29, 105	44, 133, 73, 122, 134, 70, 29, 105	44, 133, 134, 73, 70, 29, 96, 122	44, 133, 73, 122, 134, 70, 29
Lin	60, 48, 56, 112, 96, 108, 46, 42, 104, 89, 41, 84, 3, 135, 100, 8, 61, 27, 30, 59	39, 60, 56, 48, 112, 61, 96, 8, 46, 30, 27, 42, 104, 2, 84, 107, 41, 89, 3, 100, 76, 9, 77, 1, 108	56, 60, 104, 89, 46, 41, 42, 3, 84	60, 56, 46, 42, 104, 89, 41, 84, 3
PI	39, 71, 93	71, 93	39, 71, 93, 105	71, 93
Total	31	35	21	•
Remain pr	106/137 (77%)	102/137 (74%)	116/137 (85%)	•

Рисунок 2 - Висновки по кількості викидів за трьома тестами

Рисунок 3 - Висновки по якості нелінійних моделей

Висновки. Представлене програмне рішення для автоматизації побудови нелінійних регресійних моделей може бути корисним в дослідницькій діяльності у випадках побудови набору моделей на основі різних наборів даних. Це особливо корисно, при дослідженнях моделей на основі великої кількості наборів даних, а також коли їх розміри перевищують сто точок.

У випадках роботи над невеликими наборами даних та побудови однієї або декількох моделей, поставлені задачі можна вирішити за допомогою табличного процесора Excel з пакету MS Office.

Наступним кроком вдосконалення запропонованого програмного рішення можна вважати додавання функціоналу для побудови k-факторних моделей (k - кількість факторів), а також розширення звітів.

Література

[1] Prykhodko, S. B., Prykhodko, N. V., & Koltsov, A. V. (2024). A nonlinear regression model for early LOC estimation of open-source Kotlin-based applications. Radio Electronics, Computer Science, Control, (1), 85. https://doi.org/10.15588/1607-3274-2024-1-8

[2] Prykhodko, N. V., & Prykhodko, S. B. (2018). The non-linear regression model to estimate the software size of open source Java-based systems. Radio Electronics, Computer Science, Control, (3). https://doi.org/10.15588/1607-3274-2018-3-17

[3] Prykhodko, A. S., & Malakhov, E. V. (2024). Determining Object-Oriented design complexity due to the identification of classes of open-source Web applications created using Php frameworks. Radio Electronics, Computer Science, Control, (2), 160. https://doi.org/10.15588/1607-3274-2024-2-16

[4] Chidamber, S. R., & Kemerer, C. F. (1991). Towards a metrics suite for object oriented design. ACM SIGPLAN Notices, 26(11), 197–211. https://doi.org/10.1145/118014.117970

[5] Chidamber & Kemerer Object-Oriented Metrics Suite. 2021. [Електронний ресурс] – Режим доступу до ресурсу: https://www.aivosto.com/project/help/pm-oo-ck.html

[6] Макарова, Л. М., & Татаренко, М. А. (2025). Аналіз інструментальних засобів для отримання метрик Чидамбера і Кемерера з програмних проектів, розроблених мовою Java. Вісник Херсонського національного технічного університету, 2(2(93)), 218–227. https://doi.org/10.35546/kntu2078-4481.2025.2.2.26

АВТОМАТИЗАЦІЯ РЕПЕТИТОРСЬКОЇ ДІЯЛЬНОСТІ З ВИКОРИСТАННЯМ ЗАСОБІВ ВЕБТЕХНОЛОГІЙ

Тімченко Е.О.
Керівник: Макарова Л.М.
E-mail: elinatimchenco@gmail.com, lidiia.makarova@nuos.edu.ua
Миколаїв, Національний університет кораблебудування імені адмірала Макарова

У сучасних умовах активного впровадження цифрових технологій в освітню сферу особливої актуальності набуває проблема автоматизації професійної діяльності викладачів та репетиторів. Після пандемії COVID-19 дистанційне та індивідуальне навчання стало звичним форматом освітнього процесу, що спричинило зростання попиту на послуги приватних репетиторів. У зв'язку з цим виникла потреба у зручних та універсальних інструментах для організації їхньої роботи [1].

Діяльність репетитора не обмежується лише проведенням занять, а включає планування розкладу, ведення обліку учнів, контроль оплат, а також постійну комунікацію з батьками. Використання паперових носіїв або кількох окремих сервісів ускладнює робочий процес, призводить до втрати інформації та підвищує ймовірність помилок. Тому доцільним є створення єдиного цифрового середовища, яке об'єднує всі необхідні функції в одному ресурсі. Вибір формату вебзастосунку зумовлений його універсальністю та зручністю використання. Вебрішення не потребує встановлення додаткового програмного забезпечення на комп'ютер або мобільний пристрій, що є важливим з огляду на різні технічні можливості користувачів. Адаптивна мобільна версія вебсайту забезпечує повноцінну роботу зі смартфона, який завжди знаходиться під рукою, при цьому не займаючи пам'ять пристрою. Доступ до вебзастосунку здійснюється лише за потреби, що є оптимальним для освітніх ресурсів.

Аналіз існуючих програмних рішень показав, що більшість аналогів або не охоплюють весь необхідний функціонал для репетиторської діяльності, або є платними. Для репетиторів, які не мають стабільного та прогнозованого доходу, використання дорогих підписок часто є економічно недоцільним [2-5]. Це підкреслює актуальність розробки доступного вебзастосунку з базовим набором необхідних інструментів.

Можна визначити такі ключові функції вебзастосунку:

– система керування розкладом занять з можливістю додавання, редагування та перегляду уроків;

– облік учнів із збереженням інформації про рівень знань, прогрес та навчальні досягнення;

– окремий доступ для батьків із можливістю перегляду розкладу, коментарів репетитора та інформації про оплату;

– модуль фінансового обліку для контролю оплат за заняття;

– збереження нотаток репетитора щодо учнів та особливостей навчального процесу.

Таким чином, у даній роботі розглянуто концепцію вебзастосунку для автоматизації діяльності репетитора, визначено його основні функції та переваги використання. Отримані результати створюють підґрунтя для подальшої реалізації та впровадження описаного вебзастосунку в репетиторській діяльності.

Література

[1] Wikipedia. Online tutoring [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Online_tutoring

[2] Google Classroom. Online learning platform [Електронний ресурс] – Режим доступу: https://edu.google.com/intl/uk/products/classroom/

[3] TutorBird. Tutoring management software [Електронний ресурс] – Режим доступу: https://www.tutorbird.com

[4] Teachworks. Tutoring management system [Електронний ресурс] – Режим доступу: https://www.teachworks.com

[5] SkillzRun. Мобільний додаток для навчання [Електронний ресурс] – Режим доступу: https://www.skillzrun.com

МОЖЛИВОСТІ ПРИКЛАДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ FLIPACLIP ДЛЯ СТВОРЕННЯ АНІМАЦІЙ

Чайка А.В.
E-mail: wowalinachaika@gmail.com
Харків, Комунальний заклад «Харківський ліцей № 153 Харківської міської ради»

У сучасному цифровому світі анімація відіграє важливу роль у сфері освіти, розваг, реклами та мистецтва. Для створення анімацій широко використовуються прикладні програмні засоби, які дозволяють користувачам реалізувати творчі ідеї без необхідності глибоких знань програмування. Одним із таких популярних інструментів є програмне забезпечення FlipaClip, орієнтоване на створення покадрової 2D-анімації.

FlipaClip - це прикладне програмне забезпечення, яке працює на мобільних пристроях (Android, iOS), а також на комп'ютерах за допомогою емуляторів. Основною перевагою програми є простий та інтуїтивно зрозумілий інтерфейс, що робить її доступною для початківців, школярів та студентів. Разом з тим, FlipaClip має достатньо функціональних можливостей для використання у напівпрофесійній анімаційній діяльності.

Також це найпростіший спосіб втілити в життя малюнки, мультфільми, аніме та історії. В програмі можна створювати будь-які види анімації, незалежно від того, чи це аніме, чи меми, чи анімація з паличок чи навіть мультсеріал.

Дана програма для 2D-анімації поєднує простоту анімації фліпбуку з професійними інструментами редактора анімації. Можна малювати покадрово, редагувати кожну деталь та експортувати свою анімацію як відео або GIF.

Ключові можливості FlipaClip:

1. Створення покадрової анімації. Користувач може малювати кожен кадр окремо, що дозволяє детально контролювати рух об'єктів і персонажів. Програма підтримує різну частоту кадрів, завдяки чому можна регулювати плавність анімації та швидкість рухів. Це є важливим аспектом при створенні якісних анімаційних роликів.

2. Широкий набір інструментів для малювання. Зокрема олівець, пензель, маркер, ластик та інструменти для заливки. Користувач може змінювати товщину ліній, прозорість та колір, що дає змогу створювати різні художні стилі. Додатково підтримується робота зі стилусом, що особливо зручно для точного малювання на планшетах.

3. Робота з шарами (layers). Шари дозволяють розділяти об'єкти фону та переднього плану, а також спрощують редагування окремих елементів анімації без впливу на інші частини зображення. Це значно підвищує ефективність роботи та зменшує кількість помилок під час створення анімації.

4. Підтримка функції onion skin (цибулинної анімації). Саме вона дозволяє переглядати попередні та наступні кадри напівпрозоро. Це допомагає аніматору краще контролювати рухи об'єктів і забезпечує плавні переходи між кадрами. Дана можливість є стандартом у професійній анімації та значно полегшує процес створення руху.

5. Можливість додавання звуку. У FlipaClip користувач може імпортувати аудіофайли або записувати звук безпосередньо в програмі. Це дозволяє створювати анімації з озвученням, музичним супроводом або звуковими ефектами, що підвищує виразність та емоційність готового продукту.

6. Підтримка експорту анімацій у різних форматах. Зокрема MP4 та GIF. Це дає змогу публікувати створені роботи у соціальних мережах, на відеохостингах або використовувати їх у презентаціях та навчальних матеріалах. Таким чином, FlipaClip сприяє поширенню творчих проєктів і розвитку цифрової креативності.

7. Освітній потенціал FlipaClip. Програма активно використовується у навчальному процесі для розвитку творчого мислення, візуального сприйняття та навичок цифрового малювання. Вона допомагає здобувачам освіти зрозуміти базові принципи анімації, такі як таймінг, ритм і послідовність рухів [1].

Також додаткові інструменти, які певним чином допомагають якнайкраще працювати з програмою:

– Часова шкала покадрової анімації для повного контролю;

– Ефект світіння та режими змішування;

– Імпорт фотографій або відео для створення ротоскопічної анімації;

– Експорт у послідовності MP4, GIF або PNG з прозорістю;

– Magic Cut, інструмент на базі штучного інтелекту, який миттєво вирізає зображення та об'єкти з кадрів [2].

Кожна функція цього засобу для створення анімації створена для того, щоб допомогти користувачеві швидко навчитися анімувати. Незалежно від того, чи вміє він малювати аніме, мультфільми, меми чи історії з життя, FlipaClip — це саме той додаток для 2D-анімації, який швидко та просто допоможе створити анімацію.

Однією з переваг FlipaClip є доступність для різних вікових категорій користувачів. Завдяки простому інтерфейсу та зрозумілій логіці роботи програму можуть ефективно використовувати як молодші школярі, так і старшокласники чи студенти. Це робить її зручним інструментом для впровадження елементів цифрової творчості на уроках інформатики, мистецтва або медіаграмотності.

FlipaClip також сприяє розвитку проєктної діяльності здобувачів освіти. Користувачі можуть створювати власні анімаційні історії, соціальні ролики, освітні мінівідео або інтерактивні презентації. Така діяльність формує навички планування, командної роботи, критичного мислення та цифрової комунікації, що відповідає сучасним освітнім стандартам.

Крім того, використання FlipaClip допомагає інтегрувати різні навчальні предмети через міждисциплінарні проєкти. Наприклад, на уроках історії можна створювати анімовані реконструкції подій, на уроках літератури — екранізації уривків творів, а на заняттях з інформатики — вивчати основи алгоритмічного мислення через покадрову логіку створення руху. Це робить навчальний процес більш інтерактивним та мотивуючим для здобувачів освіти [3].

Отже, FlipaClip є ефективним прикладним програмним забезпеченням для створення 2D-анімації, яке поєднує доступність використання з широким набором функціональних можливостей. Завдяки інтуїтивному інтерфейсу та різноманітним інструментам програма підходить як для початківців, так і для користувачів із базовим досвідом у сфері цифрового малювання та анімації. Її функціонал дозволяє створювати якісні анімаційні роботи, розвивати творче мислення, візуальну грамотність та навички роботи з цифровими технологіями. Особливо цінним є освітній потенціал FlipaClip, адже застосунок можна ефективно використовувати під час навчального процесу для реалізації проєктної діяльності, розвитку креативності та формування сучасних цифрових компетентностей здобувачів освіти. Таким чином, використання FlipaClip сприяє не лише створенню анімаційного контенту, а й формуванню практичних навичок, необхідних у сучасному інформаційному суспільстві, відкриваючи широкі можливості для самовираження та навчання у цифровому середовищі.

Література

[1] FlipaClip у Google Play [Електронний ресурс]. – Режим доступу: https://play.google.com/store/apps/details?id=com.vblast.flipaclip

[2] Офіційний сайт FlipaClip [Електронний ресурс]. – Режим доступу: https://flipaclip.com/

[3] Офіційний YouTube-канал FlipaClip [Електронний ресурс]. – Режим доступу: https://youtube.com/@flipaclip

Секція 4

КІБЕРБЕЗПЕКА, СТАНДАРТИЗАЦІЯ ТА ЦИФРОВІ ТЕХНОЛОГІЇ

(СЕКЦІЯ ЗА МАТЕРІАЛАМИ ПРОЄКТУ 101176904 – EU-CYBERCONNECT-UA – ERASMUS-JMO-2024-MODULE)

METHOD FOR SELECTING IDP PROVIDER FOR INTEGRATION WITH DOCKER

Darienko D.H., Kohut N.Yu.
Supervisor: Parkhuts L.T.
E-mail: liubomyr.t.parkhuts@lpnu.ua
Lviv, Lviv Polytechnic National University

The paper investigates the problem of ensuring the security of the container build process in the Docker environment by selecting the optimal identity provider (IdP). The main goal is to form a method for selecting an IdP that is capable of preventing unauthorized operations during the build and deployment of containers. The Security Threat Oriented Requirements Engineering (STORE) approach was used to identify key risks, which made it possible to identify threats caused by developer errors, the use of unreliable or compromised build agents, as well as the possibility of unauthorized build initiation. Based on the analysis of these threats, the following requirements for IdPs were formulated: support for multi-factor authentication, the use of short-term and revocable tokens, flexible access control models, integration with CI/CD processes and Kubernetes, as well as centralized auditing and mapping of claims to authorization policies. A comparative analysis of identity providers identified the most suitable IDP providers that provide the necessary flexibility and level of security.

The active development of containerization and continuous integration and delivery (CI/CD) tools has led to the need for increased control over the software build process. Docker, which is the basis of most DevOps pipelines, does not provide a full-fledged mechanism for distinguishing between user rights and build agents. As a result, any user with access to the environment can perform actions beyond their authority, creating the risk of unauthorized changes to the code, leakage of confidential data, or compromise of the deployment process. This problem is especially acute in distributed development teams, where the lack of a centralized authentication and authorization system makes it difficult to control access to critical components of the CI/CD infrastructure.

The method proposed in the paper is based on a combination of the principles of Security Threat Oriented Requirements Engineering (STORE) [1] and classic access control models, in particular Role-Based Access Control (RBAC) [2].

The method is aimed at eliminating threats that arise during the container build process through integration with an external identity provider. The theoretical basis for the development of the method was the work in the field of identification and access management (IAM), which emphasizes the importance of centralized identity models [3], as well as research into modern practices of Single Sign-On and multi-factor authentication [4, 5].

The method involves a sequence of stages that transform threat analysis into formalized criteria for selecting an IdP.

1. The first stage identifies threats specific to the Docker image assembly process. This corresponds to the Threat Identification phase in STORE. Objects (Docker builder, agents, Dockerfile, secrets), potential attack subjects (external attacker, developer), consequences (embedding malicious code, data leakage), and levels of probability and impact are determined.

2. The second stage is to transform threats into requirements for the IdP (Security Requirements Elicitation). A countermeasure is defined for each threat: for example, unauthorized assembly launch is overcome through short-lived tokens, agent verification through SCIM and API lifecycle management, and secret leakage through auditing and claims-to-policy restrictions.

All these requirements are grouped into five categories:

– SEC (Security) – authentication, MFA, tokens, auditing;
– INT (Integration) – SCIM, API, lifecycle and claims mapping;
– OPS (Operations) – administration, updates, support;
– REL (Reliability) – SLA, performance, architectural stability;
– TCO (Total Cost of Ownership) – licensing, implementation and maintenance costs.

3. The third stage is the quantitative assessment of providers using criteria assessment:

$$S = \sum_{i=0}^{n} (w_i \times s_i)$$

where n – is the number of evaluation criteria -1, is the weight of the -th criterion, and is the normalized score (from 0 to 1) obtained as a result of the analysis of a specific parameter. This approach allows for an objective comparison of different solutions, taking into account not only functionality, but also compliance with the practical requirements of the container build environment.

4. At the fourth stage, a scorecard is formed for each IdP, which allows for a quantitative comparison of their compliance with DevSecOps requirements. The method is completed by analyzing the results, where providers are ranked by the final score.

Conclusion

As a result of the research, a method for selecting an identity management system (IdP) provider for integration into the Docker builder environment was developed to prevent unauthorized actions during software build.

Based on the analysis of threats identified within the Security Threat Oriented Requirements Engineering (STORE) approach, a system of criteria and weighting factors was formed that reflect the impact of each aspect of the IdP on reducing risks associated with uncontrolled actions of developers, assembly agents and the possibility of compromising secrets.

The results of evaluating providers according to this model showed that the highest indicators were demonstrated by Okta, Azure AD and Auth0, which provide extended support for OIDC-hardening, multi-factor authentication, flexible RBAC/ABAC policies and mature SCIM and API integration mechanisms. Their architectural capabilities ensure the effective implementation of the principle of least privilege and centralized access management in the CI/CD infrastructure.

At the same time, open source solutions such as Keycloak and Ory are characterized by high flexibility and cost-effectiveness, but require additional customization to achieve the level of security and reliability inherent in commercial solutions.

References

[1] T. Jamal Ansari, D. Pandey and M. Alenezi. STORE: Security Threat Oriented Requirements Engineering. Journal of King Saud University – Computer and Information Sciences, pp. 1210-1211, - 2020.

[2] R.S. Sandhu, E.J. Coyne, F.L. Hall, and C.E. Youmank. Role-Based Access Control Models. IEEE Computer, pp. 38-74, - 1996.

[3] J. Glöckler, J. Sedlmeir, M. Frank and G. Fridgen. A Systematic Review of Identity and Access Management Requirements in Enterprises and Potential Contributions. Business & Information Systems Engineering, pp. 421-440, - 2023.

[4] V. Radha and D. Sahitha. A Survey on Single Sign-On Techniques. Procedia Technology, pp. 134-139, - 2012.

[5] A.Zineddine, Y.Belfaik, A.Rehaimi, Y.Sadqi and S.Safi. Single Sign-On Security and Privacy: A Systematic Literature Review. Computers, Materials & Continua, pp. 4019-4054, - 2025.

ANALYSIS OF APPLICATION-LAYER VIDEO DATA TRANSMISSION WITH ADAPTIVE CONTROL IN UAV NETWORKS

Jiang He[1], Jian Yu[2],
Supervisor: Semenov S.[3]
E-mail: s_semenov@ukr.net, jianyu220272@gmail.com, hjamxm@163.com
[1]*China, Zhongke Shuguang,*
[2]*China, CNOOC Financial Shared Service Center PRD Branch,*
[3]*Kharkiv, Simon Kuznets Kharkiv National University of Economics*

The rapid development of UAV-based systems has led to a growing demand for reliable real-time video transmission in highly dynamic and resource-constrained network environments.

UAV communication channels are characterized by variable latency, intermittent connectivity, limited bandwidth, and frequent topology changes. In such conditions, conventional video transmission techniques designed for stable terrestrial networks often fail to meet the temporal requirements of UAV applications (1-4).

Most existing solutions address video transmission challenges by adapting physical, MAC, or transport layer parameters. However, these approaches are frequently limited by hardware constraints, protocol rigidity, or the lack of semantic awareness of transmitted video content. As a result, application-layer control has emerged as a promising alternative, enabling decision-making based on the actual usefulness and timeliness of video data.

This paper provides an analytical examination of application-layer video transmission with adaptive control in UAV networks, focusing on the role of frame deadlines, strategy selection, and stochastic delay modeling.

Video transmission in UAV networks differs fundamentally from classical multimedia streaming scenarios. The following characteristics are particularly critical:

– Temporal relevance of video frames. In many UAV applications, video frames lose their value if delivered too late. For example, outdated visual information may be useless for navigation or target tracking.

– Stochastic nature of transmission delays. Due to mobility, interference, and multi-hop wireless links, frame delivery times exhibit significant randomness that cannot be captured by deterministic models.

– Resource constraints. UAVs operate under strict limitations on energy, processing power, and communication bandwidth, which restrict the complexity of adaptation mechanisms.

– Limited protocol flexibility. Lower-layer protocols are often standardized or hardware-bound, making application-layer solutions more practical for adaptive control.

These factors motivate a shift from reliability-oriented or throughput-oriented transmission toward deadline-aware and context-sensitive data delivery.

Application-layer adaptive control treats video transmission as a sequence of decision-making steps performed for individual video frames. Each frame is associated with a deadline that defines the maximum acceptable delivery time.

Instead of attempting to deliver all frames reliably, the application layer dynamically selects a transmission strategy based on the estimated probability of meeting the deadline. Such strategies may differ in terms of redundancy, segmentation, or tolerance to packet loss.

The core idea is to maximize the probability of timely delivery rather than minimize average delay or packet loss. This paradigm allows the system to discard frames that are unlikely to arrive on time, thereby conserving network resources and reducing congestion.

Importantly, this approach does not require modifications to existing video transport protocols (e.g., RTP/RTCP), as deadlines and decision logic are handled internally at the application layer.

Analytical and simulation-based studies indicate that application-layer adaptive control can significantly improve the probability of timely frame delivery compared to non-adaptive baseline approaches. The most pronounced gains are observed under strict deadline constraints, which are typical for real-time UAV applications.

While the adaptive method may tolerate a slightly higher frame loss rate, the overall fraction of late or useless frames is reduced. This trade-off is acceptable and even desirable in scenarios where outdated information is more harmful than missing information.

Furthermore, separating probabilistic metrics from time-based metrics provides clearer insight into system behavior and avoids misleading interpretations caused by mixed-scale evaluations.

Conclusions.

This paper analyzed the process of application-layer video data transmission with adaptive control in UAV networks. The analysis highlights the advantages of deadline-oriented decision-making and probabilistic modeling of transmission delays.

The results demonstrate that application-layer adaptive control offers a flexible and effective means of improving video transmission timeliness without modifying lower-layer protocols. This approach is particularly well suited for UAV networks, where temporal relevance and adaptability are more critical than traditional quality-of-service metrics.

Future research directions include integrating learning-based delay estimation, extending the model to multi-UAV cooperative scenarios, and validating the approach in real-world flight experiments.

References

[1] M. K. Sharma, C.-F. Liu, I. Farhat, N. Sehad, W. Hamidouche, and M. Debbah, "UAV Immersive Video Streaming: A Comprehensive Survey, Benchmarking, and Open Challenges," arXiv preprint arXiv:2311.00082, 2023.

[2] A. Baltaci, V. Bajpai, D. Schupke, et al., "Analyzing Real-time Video Delivery over Cellular Networks for Remote Piloting Operations," in Proc. ACM Internet Measurement Conference (IMC), 2022.

[3] R. Muzaffar, A. Zubow, and A. Wolisz, "Live Multicast Video Streaming from Drones: An Experimental Study," (conference paper / technical report), 2019.

[4] M. Nagy, V. Singh, J. Ott, and L. Eggert, "Congestion Control Using FEC for Conversational Multimedia Communication," in Proc. ACM NOSSDAV, 2014.

COPYRIGHT IN THE CONTEXT OF CYBERSECURITY

Khoroshko H.O.; Rovda V.V., Brailovskyi M.M.
E-mail: bk1972@ukr.net
Kyiv, Taras Shevchenko National University of Kyiv
Kyiv, State University of Information and Communication Technologies

In the digital environment, where information technologies permeate all spheres of life, the protection of intellectual property acquires particular relevance. Copyright, as a legal mechanism for safeguarding creative achievements, constitutes an integral component of cybersecurity systems. It not only regulates the legitimate use of software, databases, multimedia materials, and other objects of digital content but also serves as one of the key factors ensuring the integrity, confidentiality, and availability of information resources. Consequently, analyzing the interplay between copyright and cybersecurity represents a scientifically substantiated research direction aimed at identifying legal, technical, and organizational risks associated with the unauthorized use or dissemination of protected objects.

Regarding the artificial intelligence factor within the European Union's cybersecurity framework, the forthcoming EU AI Act – which is scheduled to enter into force in August 2026— cannot be overlooked, as it is expected to introduce radical changes affecting information security professionals and software developers alike.

Developers of AI models, particularly those employed for automated vulnerability detection, will be obligated to publish detailed summaries regarding the use of copyright-protected content for training their systems. Furthermore, rights holders will possess the unequivocal right to prohibit the use of their code or data for AI training purposes. This will compel cybersecurity companies to conduct more rigorous verification of the legitimacy of their datasets. Additionally, any content generated by AI will be required to carry digital watermarking, thereby integrating copyright protection with safeguards against social engineering and disinformation campaigns.

At present, when a Python program is created by artificial intelligence, a legal dilemma regarding authorship arises. AI-generated source code is often fully functional and may effectively substitute for human programmers. Figure 1 illustrates an example of Python code autonomously generated by an AI system. The program is entirely operational and requires no further modification.

```python
# ----------------------------
# 1. Визначення критеріальних функцій
# ----------------------------
def R(x):
    """Ризик: зменшується зі зростанням x"""
    return 100.0 / (1.0 + x)

def C(x):
    """Витрати (млн грн): зростають зі зростанням x"""
    return 0.8 * x ** 1.5

def T(x):
    """Час реагування (сек): зменшується зі зростанням x"""
    return 20.0 / (1.0 + 0.3 * x)
```

Figure 1. Example of Python code generated by an artificial intelligence system

Ukrainian legislation, specifically the Law of Ukraine "On Copyright and Related Rights," currently recognizes only natural persons as subjects of copyright. The legal status of software created with AI assistance has gained particular urgency due to the accelerated development of generative models and automated code-generation systems. According to current Ukrainian legislation-namely, the Law of Ukraine "On Copyright and Related Rights" (hereinafter referred to as "the Law") – copyright arises in connection with the creative activity of a natural person as the sole subject of authorship (Article 7 of the Law). This implies that objects of copyright must result from human intellectual creative activity, thereby explicitly precluding the attribution of legal subjectivity to AI systems.

Consequently, source code autonomously generated by an AI system without substantial creative human input formally falls outside the scope of legal protection as a copyrightable object. However, in practice, the process of code generation with AI assistance almost invariably involves some degree of human participation – be it prompt formulation, selection of architectural solutions, parameter tuning, post-processing, or verification of the output. Within this context, authorship may potentially be attributed to:

– The developer of the AI system, if it functions merely as a tool analogous to a compiler or text editor, and code generation represents a mechanical output of an algorithm without creative influence from the user;

– The AI user, provided their actions demonstrate a sufficient level of creativity – for instance, through complex iterative interaction with the model, original problem formulation, or critical analysis and modification of the generated output;

– A legal entity, in cases where the program is developed within the scope of employment duties, with copyright transferring to the employer pursuant to contractual arrangements (Article 430 of the Civil Code of Ukraine [14]).

From the perspective of scientific activity, a Python program created with AI assistance may be recognized as a scientific work provided it:

– Demonstrates novelty in methodological, algorithmic, or applied aspects;

– Constitutes the outcome of independent scientific research;

Is properly documented in the form of a scientific publication, technical report, or dissertation;

Exhibits proven scientific value – for example, by implementing a novel approach to cryptographic protocol modeling, network security analysis, or numerical simulation of dynamic systems.

It is essential to emphasize that even when AI serves as an auxiliary instrument, scientific work must demonstrate the author's personal intellectual contribution in formulating hypotheses, designing experiments, interpreting results, and other core research activities. The mere utilization of AI does not disqualify a work as scientific but necessitates clear demarcation between automated generation and genuine creative scholarly input.

In conclusion, effective protection of copyright for software products developed with AI involvement requires an integrated approach combining:

– Legal formalization of authorship through comprehensive documentation of the development process;

– Technical protection mechanisms (e.g., digital watermarks, version control systems, logging of AI interactions);

– Adherence to academic integrity through proper citation of utilized tools and explicit disclosure of AI's role in the creation process.

Such a comprehensive approach ensures not only legal but also scientific legitimacy for these developments within the framework of Ukraine's contemporary information and education-science policy.

Thus, copyright constitutes not merely a legal category but also a critical component of cybersecurity architecture. Its effective implementation demands the integration of legal norms with technical information protection mechanisms. As artificial intelligence, blockchain technologies, and cloud services continue to evolve, this interdependence will deepen further, necessitating the development of novel protection models that harmoniously reconcile the interests of authors, users, and information system operators. Accordingly, future scientific research in this domain should focus on formulating a unified methodology capable of ensuring both legal and technical resilience of digital objects within cyberspace.

Copyright constitutes an essential structural element of cybersecurity architecture, ensuring integrity, confidentiality, and availability of digital assets. Under Ukrainian legislation, only natural persons qualify as copyright holders, creating legal ambiguity for AI-generated software. Nevertheless, practical AI-assisted development invariably involves human creative input—through prompt engineering, architectural decisions, or critical validation—enabling authorship attribution to the AI user (demonstrating sufficient creativity), the AI developer (if functioning as a mere tool), or a legal entity under employment arrangements. For scientific recognition, AI-assisted code must exhibit methodological novelty, result from independent research, undergo proper documentation, and demonstrate verifiable scholarly value, with the researcher's intellectual contribution clearly distinguishable from automated generation. Effective protection demands an integrated triad: legal formalization via development logs, technical safeguards (digital watermarking, version control, interaction logging), and academic integrity through transparent AI disclosure. As AI, blockchain, and cloud technologies converge, harmonizing legal frameworks with technical protection mechanisms becomes imperative. Future research must prioritize unified methodologies ensuring both legal enforceability and technical resilience of digital objects, thereby reconciling authors', users', and operators' interests within evolving cyberspace.

References

[1] Guadamuz, A. (2021). Artificial Intelligence and Copyright. Journal of Intellectual Property Law & Practice, 16(10), pp. 1034–1042. DOI: 10.1093/jiplp/jpab112

[2] WIPO (2022). WIPO Technology Trends 2022: Artificial Intelligence and Intellectual Property. World Intellectual Property Organization, Geneva. DOI: 10.34667/tind.1011

[3] Khoroshko, V., Laptiev, O., Brailovskyi, M., Laptieva, T., & Laptiev, S. (2025). Effectiveness of software for state cyber protection. Cybersecurity: Education, Science, Technology, 2(30), 1–19. https://doi.org/10.28925/2663-4023.2025.30.922

THE ROLE OF WIRESHARK IN NETWORK TRAFFIC ANALYSIS

Kyselova Y.O.
Supervisor: Starkova O.V.
E-mail: yanakis005@gmail.com
Kharkiv, Simon Kuznets Kharkiv National University of Economics

Wireshark is a powerful network traffic analyser widely used by information security specialists to examine network traffic, protocols, and packets in order to detect signs of cyberattacks. The tool can be used for both professional activities and educational purposes.

The history of Wireshark began in 1997, when Gerald Combs created the open-source project Ethereal to address network issues. The project developed rapidly and, in 2006, was renamed Wireshark. In 2008, the company released version 1.0, updated the user interface in 2015, and by 2023, the project went under the management of the Wireshark Foundation [1].

The programme gained popularity due to its ability to collect and analyse network packets in real-time [2].

Wireshark's technical characteristics allow for in-depth analysis of traffic at all levels of the OSI reference model. The tool supports hundreds of network protocols, including TCP, UDP, HTTP, HTTPS, DNS, ICMP, and many others. Each intercepted packet can be disassembled into parts, allowing detailed examination of the whole structure to perform further examination of its headers, content, and transmission parameters [3].

Wireshark works by capturing network packets using special libraries (libpcap/WinPcap) and then decoding them. The traffic can be gathered from various network interfaces, save it in pcap format files [4], and perform further offline analysis. Thanks to a system of filters, the programme allows quick identification of relevant packets and examine only relevant data [5].

Wireshark's capabilities do not end with traffic interception. Detecting suspicious network activity, analysing unauthorised access attempts, investigating malware behaviour, and testing the effectiveness of attack detection systems are just a few of the services available to the user. It is this rich functionality that gives every reason to consider the tool under review as an essential tool for investigating cybersecurity incidents and digital forensics [3].

The advantages of Wireshark include openness, cross-platform compatibility, support for large number of protocols, and the availability of advanced filtering and search mechanisms. Thanks to its graphical interface, the tool is accessible even to beginners, i.e., while still offering extensive capabilities for professional analysis.

All of this makes Wireshark one of the key open-source tools in the field of cybersecurity, providing opportunities for detailed analysis of network packets, diagnosis of network problems, and detection of potential cyber threats.

The use of Wireshark enhances the security of information systems and contributes to the practical training of future cybersecurity professionals.

References

[1] A Brief History of Wireshark [Electronic resource]. – Resource access mode: https://www.wireshark.org/docs/wsug_html_chunked/ChIntroHistory.html

[2] Wireshark Official Documentation [Electronic resource]. – Resource access mode: https://www.wireshark.org/docs/

[3] Wireshark User Guide [Electronic resource]. – Resource access mode: https://www.wireshark.org/docs/wsug_html_chunked/

[4] Wireshark Capture Filters Documentation [Electronic resource]. – Resource access mode: https://wiki.wireshark.org/CaptureFilters

[5] The “Filter” Toolbar [Electronic resource]. – Resource access mode: https://www.wireshark.org/docs/wsug_html_chunked/ChUseFilterToolbarSection.html

OVERVIEW OF FREE SOFTWARE TOOLS FOR SPAM FILTERING

Lichman V. O.
Supervisor: Pochanskiy O.M.
E-mail: vladislav.lichman@hneu.net
Kharkiv, Simon Kuznets Kharkiv National University of Economics

Despite advances in email security, spam remains a persistent problem for many mail systems. For most users, it is mainly an annoyance, but in reality, unsolicited messages often carry much greater risks. Spam is frequently used for phishing, distributing malware, and manipulating recipients through deceptive tactics. Filtering unwanted email traffic is therefore still an important task, especially when commercial solutions are not feasible and open-source tools are preferred.

Apache SpamAssassin is one of the oldest and most widely used tools in this area. It evaluates emails by scoring them based on heuristics, signature checks, DNS blacklists, and Bayesian analysis. This setup is flexible and can integrate with popular mail servers like Postfix, Sendmail, and Exim. However, when processing large volumes of incoming messages, SpamAssassin can consume substantial computing resources, which may reduce its efficiency in large installations [1].

Rspamd has become a popular alternative in recent years. It is designed for speed and performance, combining statistical analysis, machine learning, reputation checks, and attachment inspection. Rspamd also offers centralized configuration and a web-based monitoring interface, making it easier to manage and suitable for institutions and corporate networks [2].

Bogofilter represents a different approach and is typically used on the client side. It relies on Bayesian classification and adapts gradually to the user's email patterns through continuous learning. While lightweight and easy to deploy, its accuracy depends on the quality and representativeness of the training data [3].

In practice, spam filters are rarely used alone. Additional services often improve overall reliability. For example, Spamhaus DNSBL and SURBL help evaluate the reputation of sending servers and URLs in messages, while ClamAV detects known malware in attachments. Combining these tools with primary filters improves protection and reduces false positives [4, 5].

In summary, open-source spam filtering tools can provide a reliable level of protection when used as part of a multi-layered security approach. For educational institutions, small and medium-sized organizations, and research environments, they offer a practical and flexible alternative to proprietary email filtering systems.

References

[1] Apache SpamAssassin Project. Apache SpamAssassin Documentation [Online]. Available at: https://spamassassin.apache.org (accessed February 6, 2026).

[2] Rspamd Project. Rspamd: Fast, Free and Open-Source Spam Filtering System [Online]. Available at: https://rspamd.com (accessed February 6, 2026).

[3] Bogofilter Project. Bogofilter: A Bayesian Spam Filter [Online]. Available at: https://bogofilter.sourceforge.net (accessed February 6, 2026).

[4] Spamhaus Project. The Spamhaus Project: DNS-Based Block Lists [Online]. Available at: https://www.spamhaus.org (accessed February 6, 2026).

[5] ClamAV Team. Clam AntiVirus Documentation [Online]. Available at: https://www.clamav.net (accessed February 6, 2026).

SUSTAINABLE DEVELOPMENT ISSUES OF UNDERGROUND CRITICAL INFRASTRUCTURE FACILITIES

Liubynskyi P.L.
Supervisor: Shapovalova O.O.
E-mail: petro.liubynskyi@gmail.com
Kharkiv, Simon Kuznets Kharkiv National University of Economics

Underground engineering structures are an essential part of urban life-support systems. They play an important role in maintaining environmental, sanitary, and socio-economic safety. Their reliable operation contributes directly to urban resilience and sustainability, which justifies considering these facilities as elements of critical infrastructure [1].

The importance of treating underground engineering networks as critical infrastructure is linked to several challenges, including aging systems, harsh operating conditions, limited funding, and growing exposure to natural and technological threats. Failures in these networks can trigger serious social, economic, and environmental impacts. Therefore, their sustainable operation requires systematic and scientifically based management approaches [1].

Modern information technologies significantly strengthen the resilience of critical infrastructure facilities. Digital monitoring tools, geographic information systems, automated data collection platforms, and integrated information environments make it possible to evaluate the technical condition of underground structures in real time. These technologies also help combine technical, financial, and environmental data within a unified analytical framework, improving the accuracy and speed of decision-making [1, 2].

Intelligent data analysis methods, including artificial intelligence and expert systems, support predictive maintenance, risk evaluation, and scenario-based infrastructure planning. This makes it possible to shift from responding to failures after they occur toward preventing them through proactive management [2, 3].

An indicator-based approach to sustainability assessment, supported by information technologies, helps structure monitoring processes, define threshold values, and detect critical conditions at early stages. Decision-support systems also assist in allocating resources more effectively, improving rehabilitation planning, and reducing environmental and technological risks [1].

In conclusion, integrating information technologies into the management of underground engineering structures enhances their safety, reliability, and sustainability. It also provides a methodological basis for developing digitally oriented systems for modern urban infrastructure management.

References

[1] Aleinikova, A., Bondarenko, D., Goncharenko, D., Starkova, O. (2022). Methodological principles for informational and technological monitoring of the stable operation of the sewerage networks. Kharkiv, 272. Available at: http://repository.hneu.edu.ua/handle/123456789/29772

[2] Hojjati, A., Jefferson, I., Metje, N., Rogers, C. D. F. (2018). Sustainability assessment for urban underground utility infrastructure projects. Proceedings of the Institution of Civil Engineers – Engineering Sustainability, 171 (2), 68–80. https://doi.org/10.1680/jensu.16.00050

[3] Sakai, H. (2024). Review of research on performance indicators for water utilities. AQUA – Water Infrastructure, Ecosystems and Society, 73 (2), 167–182. https://doi.org/10.2166/aqua.2024.224

IMAGE CODEC LIBRARIES AS A BASELINE FOR STEGANOGRAPHY USING SUBOPTIMAL DECISION ENCODING: PNG 3 EXAMPLE

Ponomarenko Y.V.
E-mail: allagrir@gmail.com
Kyiv, Taras Shevchenko National University of Kyiv

The two primary ways of storing image information in digital form are raster and vector. Put simply, raster stores the color information recorded, captured, or stored at a particular point in space, called a pixel. Those pixels are then arranged at a rectangular array and are thought to be squares of equal size tightly packed together. Vectors, instead, record visual information using mathematical equations and transformations. Those allow the end user to solve those mathematical equations for any value within the image, allowing theoretically infinite precision and infinitely sharp and well-defined features.

Steganography is “the art or practice of concealing a message, image, or file within another message, image, or file” [1]. It could be thought of as storing information “in plain sight”. The goal is to conceal the existence of a second stream of information. Usually it is performed through changing the source data in a way that embeds a second, hidden stream within the first.

A new model of steganographic encoding has been developed recently, named Suboptimal Decision Encoding (SDE) [2]. Instead of modifying the source image (or sound, or video, or any other media file), it changes the way it is encoded. Digital files that store analogue media require the use of particular storage formats and corresponding codecs to map intensity data to bits and bytes. One of the properties that most storage formats possess is the ability to represent the same exact input with an arbitrarily large amount of variations of representations. Commonly, the goal of many storage formats and codecs is to reduce the amount of storage space required to represent the same input (or as close to it as possible). Instead, SDE changes the goal of an encoder to instead create a representation that, apart from taking up the least amount of space, also happens to encode a secret message within the output stream.

Within PNG, an open-source, royalty-free image format [3], the main targets for SDE embedding are IDAT chunks, which store the actual pixel values for a given image, as well as the fdAT chunks that store frames of an animated sequence. PNG employs different filters and zlib compression [4] to reduce the output filesize. The official implementation of PNG codec, libpng [5], can be used as a starting point to analyze the encoding implementation and create a new encoder that uses SDE to additionally embed a new, secret stream of data within the PNG image. Additionally, another SDE embedding created for the zlib compression algorithm can be used to further encode an additional secret stream, or to extend the possible length of a total stream of data, allowing for better overall efficiency of the system.

References

[1] Merriam-Webster, “Steganography,” Merriam-Webster.com Dictionary. [Online]. Available: https://www.merriam-webster.com/dictionary/steganography. [Accessed: Feb. 9, 2026].

[2] Y. Ponomarenko and O. Laptiev, “Mathematical model of steganography using suboptimal decisions in data compression algorithms,” Information systems and technologies security, vol. 1, no. 9, pp. 54-60, Aug. 2025, doi: 10.17721/ISTS.2025.9.54-60.

[3] World Wide Web Consortium, "Portable Network Graphics (PNG) Specification," W3C Technical Reports. [Online]. Available: https://www.w3.org/TR/png-3/. [Accessed: Feb. 9, 2026].

[4] P. Deutsch and J-L. Gailly, "ZLIB Compressed Data Format Specification version 3.3," RFC 1950, Network Working Group, May 1996. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1950. [Accessed: Feb. 9, 2026].

[5] "Libpng Home Page," The libpng Group. [Online]. Available: https://www.libpng.org/pub/png/libpng.html. [Accessed: Feb. 9, 2026].

REVIEW OF READY-MADE SOLUTIONS FOR MONITORING AND ANOMALY DETECTION

Serdiuk I. O.
Supervisor: Pochanskiy O.M.
E-mail: ihor.serdiuk@hneu.net
Kharkiv, Simon Kuznets Kharkiv National University of Economics

Modern approaches to ensuring the security of information systems involve the adoption of ready-made software solutions for the centralized collection and analysis of security events, as well as the detection of anomalous activities in the network. Such solutions help increase the overall level of protection, reduce incident response time, and automate threat handling.

One of the key free solutions is Wazuh – an open-source SIEM platform (Security Information and Event Management) that provides security monitoring, log analysis, incident management, and event correlation from multiple sources. Wazuh is capable of collecting data from endpoints, servers, and cloud services, performing file integrity monitoring, detecting suspicious activities, and responding to incidents. It is available as open-source software and is supported by extensive documentation and an active user community [1].

To enhance Wazuh's capabilities in detecting network-based attacks, Suricata is commonly used – a modern Intrusion Detection and Prevention System (IDS/IPS). Suricata analyzes network traffic in real time, applies signature-based rules to identify known attacks, and enables deep inspection of HTTP, DNS, and other protocols. Logs generated by Suricata can be forwarded to a SIEM system for further correlation analysis and alert generation [2].

The integration of Wazuh with Suricata provides a comprehensive approach: the former performs centralized collection and analysis of security events, while the latter ensures deep network visibility. This architecture allows for more effective detection of complex attacks that may be missed when using a single monitoring tool. Practical studies demonstrate that such an integrated system significantly improves attack detection capabilities, including DoS attacks, SQL injections, and brute-force scanning [3].

In addition to these tools, other open-source solutions for network and security monitoring exist – for example, Security Onion, which combines multiple IDS/NSM tools along with log management and visualization systems [4].

Predefined integration models and laboratory examples, such as SOC labs based on Wazuh, Suricata, and log collectors, demonstrate practical steps for deploying and configuring a comprehensive monitoring and protection system [5].

Thus, in addressing the task of improving the security of an academic scientific and technical library's information system, it is advisable to apply ready-made open-source solutions such as Wazuh in combination with Suricata. This approach ensures centralized state monitoring, anomaly detection, and network threat identification, with the possibility of scalability and further adaptation to evolving security requirements. The use of these solutions is justified by strong community support, extensive documentation, and numerous practical implementation cases.

References

[1] Wazuh Documentation. Official website of the Wazuh security monitoring and SIEM platform. URL: https://wazuh.com/ (accessed: 05.02.2026)

[2] Dedy Unpak. Analysis of Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) Using Suricata. Medium, 2023. URL: https://medium.com/@dedy.unpak06/analysis-of-intrusion-detection-system-ids-intrusion-prevention-system-ips-using-suricata-a07c55d18025 (accessed: 05.02.2026)

[3] Integration of Wazuh and Suricata for Attack Detection. IPB University Repository, 2024. URL: https://repository.ipb.ac.id/handle/123456789/136766 (accessed: 05.02.2026)

[4] Security Onion. Wikipedia – description of the network monitoring and IDS platform. URL: https://en.wikipedia.org/wiki/Security_Onion (accessed: 05.02.2026)

[5] Marxgoo. Wazuh SOC Lab (Wazuh + Suricata practical laboratory). GitHub Repository, 2023. URL: https://github.com/marxgoo/Wazuh-SOC-Lab (accessed: 05.02.2026)

OVERVIEW OF KEY CYBERSECURITY STANDARDS AND REGULATIONS IN THE EUROPEAN UNION

Starkova O.V.
E-mail: olha.starkova@hneu.net
Kharkiv, Simon Kuznets Kharkiv National University of Economics

The European Union has developed a coherent cybersecurity framework aimed at protecting information systems, digital services, and critical infrastructure. This framework integrates binding legal acts, international technical standards, and institutional mechanisms to harmonize cybersecurity requirements across Member States.

A central component of this framework is the Network and Information Security Directive (Directive (EU) 2016/1148) and its revised version, NIS2 (Directive (EU) 2022/2555), which establish mandatory cybersecurity obligations for essential and important entities [1, 2]. NIS2 expands the range of covered sectors, strengthens risk management and incident reporting requirements, and increases the responsibility of organizational management for cybersecurity governance [2].

The protection of personal data is regulated by the General Data Protection Regulation (GDPR), which requires organizations to implement appropriate technical and organizational measures to ensure the confidentiality, integrity, and availability of personal data, as well as to report security breaches in a timely manner [3]. These requirements directly contribute to the overall cybersecurity posture of information systems.

From a technical perspective, organizations within the EU widely apply the ISO/IEC 27000 family of standards, particularly ISO/IEC 27001, as a reference model for building and maintaining an Information Security Management System (ISMS) [4]. Compliance with this standard supports structured risk management and facilitates alignment with European regulatory requirements.

An additional element of the EU cybersecurity architecture is the EU Cybersecurity Act, which establishes a European cybersecurity certification framework for ICT products, services, and processes and strengthens the role of ENISA in coordinating cybersecurity activities at the EU level [5]. Certification mechanisms contribute to a consistent assessment of cybersecurity assurance across the internal market.

In summary, the EU cybersecurity framework combines regulatory instruments, technical standards, and certification mechanisms into an integrated system that supports the sustainable protection of digital infrastructure and information systems within the European Union.

References

[1] Directive (EU) 2016/1148 (NIS Directive). Official Journal of the European Union. URL: https://eur-lex.europa.eu/

[2] Directive (EU) 2022/2555 (NIS2 Directive). Official Journal of the European Union.URL: https://eur-lex.europa.eu/

[3] Regulation (EU) 2016/679 (GDPR). General Data Protection Regulation. URL: https://eur-lex.europa.eu/

[4] ISO/IEC 27001:2022. Information Security Management Systems – Requirements. URL: https://www.iso.org/

[5] Regulation (EU) 2019/881 (EU Cybersecurity Act). ENISA and the EU Cybersecurity Certification Framework. URL: https://www.enisa.europa.eu/

ANALYSIS OF METHODS AND MEANS OF PROTECTION OF UAV COMMUNICATION CHANNELS IN THE CONDITIONS OF APPLICATION OF RADIO ELECTRONIC WARFARE EQUIPMENT

Syniavskyi O.Yu.
Supervisor: Kostyak M.Yu.
E-mail: maryna.y.kostiak@lpnu.ua
Lviv, Lviv Polytechnic National University

Modern wireless communication systems increasingly operate in a complex electromagnetic environment, characterized by a low signal-to-noise ratio, the presence of intentional interference and active means of radio electronic suppression. In such conditions, ensuring stable, interference-resistant and protected radio communication becomes one of the key technical challenges, especially for critical applications.

The problem of protecting radio channels is of particular importance in the field of application of unmanned aerial vehicles (UAVs), which are widely used for reconnaissance, surveillance, fire control, logistics and performing special tasks. The effectiveness of the functioning of UAVs directly depends on the reliability of the communication channel between the aircraft and the control point. The loss or degradation of this channel due to the action of electronic warfare (EW) can lead to a complete loss of control or the performance of a combat mission.

Relevance of the problem

In this regard, it is relevant to analyze existing methods and means of protecting radio communication channels with UAVs, assess their effectiveness in conditions of active electronic warfare and determine appropriate directions for their integrated application. The growing role of information and electronic warfare in modern military conflicts necessitates the development and implementation of jamming-resistant and low-visibility communication systems. Electronic warfare (EW) means are capable of implementing both broadband and selective suppression of radio channels, intercepting, analyzing and distorting transmitted signals. In such conditions, traditional approaches to building communication channels are not effective enough. For UAVs, the problem is complicated by limited energy and computing resources of on-board equipment, high mobility, variable radio wave propagation conditions and the need to maintain communication over significant distances. Therefore, it is relevant to search for such protection methods that provide not only cryptographic stability, but also physical noise immunity and radio transmission concealment.

Problem statement

The purpose of the research is to generalize and analyze modern methods and means of protecting radio communication channels from UAVs in the conditions of using electronic warfare (EW) means, as well as to determine the feasibility of their use in a complex.

To achieve the set goal, the following tasks were solved in the work:

– to analyze the main approaches to protecting radio channels used in modern communication systems;

– assess the advantages and limitations of individual protection methods in conditions of active air noise;

– identify promising directions for combining different methods to increase the stability of UAV communication channels.

Ways to solve the problem

Analysis of literature sources [1-5] shows that the most common methods of protecting radio channels are information encryption, frequency scrambling, the use of directional antennas, the construction of distributed radio networks, masking of information transmission and the use of high-order noise-resistant coding.

Encryption provides protection of information from unauthorized access in the event of signal interception, but is not able to counteract intentional radio-electronic suppression or complete destruction of the communication channel. Therefore, this method should be considered as a necessary but insufficient element of the protection system.

Frequency scrambling methods and the use of noise-like signals can significantly increase the noise immunity and concealment of radio communications. Pseudo-random change of the operating frequency complicates the process of interception, detection and suppression of the signal, which makes such methods especially effective in conditions of active action of electronic warfare means.

The use of directional antennas allows you to concentrate the radiation energy in a given direction, reducing the likelihood of interception and interference. Despite the requirements for precise orientation and the complexity of operation in field conditions, this approach is promising for stationary or semi-stationary UAV control points.

Distributed radio networks provide increased fault tolerance and flexibility of communication systems by using a set of nodes capable of dynamically changing data transmission routes. This approach reduces the likelihood of complete loss of communication even in the event of suppression of individual network elements.

Methods of masking information transmission, in particular the use of chaotic signals and spectral masking, allow you to hide the very fact of information transmission, which significantly complicates the operation of enemy radio reconnaissance systems.

High-order noise-resistant coding is an important element of ensuring communication reliability in conditions of significant interference. Adaptive use of code structures depending on the current signal-to-noise ratio allows you to achieve an optimal balance between transmission speed and information reliability.

Conclusion

Based on the analysis of the main methods and means of protecting UAV radio communication channels in the conditions of using electronic warfare means, it is shown that none of the considered methods provides complete protection when used in isolation.

The most appropriate is a comprehensive approach that involves a combination of cryptographic protection, frequency scrambling, masking the fact of transmission, the use of noise-like signals and noise-resistant coding. Under favorable conditions, the efficiency of the system can be further increased by using directional antennas and distributed radio networks.

The results obtained will be used as a theoretical basis for further research and development of practical solutions in the field of ensuring the stability and security of UAV communication channels in a complex electromagnetic environment.

References

[1] Yudin O.K., Korchenko O.G., Konakhovich G.F. Information protection in data transmission networks. – K.: Publishing House LLC "NVP INTERSERVIS", – 2009. – 716p.

[2] Hryb D.A.. Principles, methods and technologies of conducting armed struggle, control of forces and means in conditions of active information confrontation of the conflicting parties / D.A. Hryb, B.O. Demidov, Yu.F. Kucherenko, A.M. Tkachev, T.V. Kuleshova // Science and Technology of the Air Force of the Armed Forces of Ukraine. 2019. Volume 1, No. 43. – P. 12-22.

[3] Shyshatsky A.V. Improving the characteristics of radio communication channels with frequency multiplexing / A.V. Shyshatsky, K.N. Gritsenok, V.K. Chumak, A.A. Zavada // Control, navigation and communication systems. Collection of scientific papers. 2017. Vol.3, No. 43. P. 5-8.

[4] Belokurskyi Yu.P.. Principles of building a system of electronic defense of units of the National Guard of Ukraine during the performance of assigned tasks / Yu.P. Belokurskyi, O.Yu. Iokhov, V.E. Kozlov, O.O. Shcherbina // Armament Systems and Military Equipment. – 2017. – Volume 4, No. 52. – P. 73-80.

[5] Gorlynskyi B.V. Methods of ensuring the reliability of information in wireless data transmission means due to adaptive coding: a-ref. Ph.D.: Kyiv, – 2019. – 20 p.

ПЕРЕВАГИ ТА РИЗИКИ ВПРОВАДЖЕННЯ ЦИФРОВИХ ТЕХНОЛОГІЙ У НАВЧАННЯ

Андрєєва Л.І.
E-mail: andrieieva_liudmyla@nuczu.edu.ua
Черкаси, Національний університет цивільного захисту України

Актуальність теми. Цифровізація освіти є однією з провідних тенденцій розвитку сучасного суспільства. Активне впровадження інформаційно-комунікаційних технологій, онлайн-платформ, хмарних сервісів та інструментів дистанційного навчання змінює традиційні підходи до організації освітнього процесу, забезпечуючи його гнучкість, доступність та орієнтацію на потреби здобувачів освіти [3].

Однією з ключових переваг цифрових технологій є розширення доступу до освітніх ресурсів. Електронні курси, відеолекції, цифрові бібліотеки та відкриті освітні платформи дають можливість навчатися незалежно від місця проживання та часових обмежень, що особливо важливо для реалізації концепції навчання впродовж всього життя [3]. Важливою перевагою є можливість персоналізації навчання. Сучасні освітні платформи дозволяють адаптувати темп, зміст і рівень складності матеріалу відповідно до індивідуальних особливостей і рівня підготовки здобувачів освіти. Це сприяє підвищенню ефективності засвоєння знань та формуванню індивідуальної освітньої траєкторії [2]. Використання мультимедійних матеріалів, інтерактивних моделей, симуляцій і тестових систем підвищує мотивацію до навчання та активізує пізнавальну діяльність. Цифрове освітнє середовище сприяє розвитку критичного мислення, креативності, навичок комунікації та цифрової грамотності, що є важливими компетентностями сучасного фахівця [1].

Водночас цифровізація освітнього процесу супроводжується певними ризиками. Одним із основних викликів є цифрова нерівність, яка проявляється у відсутності рівного доступу до технічних засобів і якісного інтернет-з'єднання. Це може призводити до зниження якості освіти для окремих категорій здобувачів [1]. Суттєвими є також ризики, пов'язані з інформаційною безпекою та захистом персональних даних. Використання онлайн-сервісів потребує дотримання правил кібербезпеки та формування відповідальної цифрової поведінки учасників освітнього процесу [5]. Надмірне використання цифрових пристроїв може негативно впливати на стан здоров'я, рівень концентрації уваги та стан здобувачів освіти. Крім того, зменшення безпосереднього спілкування може ускладнювати розвиток соціальних і комунікативних навичок[6].Окремою проблемою є недостатній рівень цифрової компетентності педагогічних працівників. Ефективне використання сучасних технологій потребує постійного підвищення кваліфікації, методичної підтримки та впровадження інноваційних педагогічних підходів [1]. Таким чином, цифрові технології значно розширюють можливості організації навчального процесу, підвищують його ефективність і доступність. Водночас їх використання потребує зваженого підходу, поєднання традиційних і цифрових методів навчання, забезпечення інформаційної безпеки та розвитку цифрової компетентності всіх учасників освітнього процесу.

Література

[1] Биков В. Ю. Цифрова трансформація освіти і науки: теорія та практика. – К.: Інститут цифровізації освіти НАПН України, 2024.

[2] Морзе Н. В., Співаковський О. В. Інформаційно-комунікаційні технології в освіті: навчально-методичний посібник. – Київ: Видавництво «Університет», 2023.

[3] Забіяка І. М. Цифровізація освіти: сучасні тенденції та перспективи розвитку. – Луцьк: Волинський національний університет, 2024.

[4] Кремень В. Г. (ред.) Освіта і наука України в умовах цифрової трансформації. – Київ: НАПН України, 2023.

[5] Петренко С. М., Коваленко О. І. Основи кібербезпеки в освітньому середовищі. – Харків: Основа, 2024.

[6] Савченко О. Я., Пометун О. І. Психолого-педагогічні аспекти використання цифрових технологій у навчанні. – Київ: Педагогічна думка, 2024.

КРИПТОВАЛЮТА ЯК ОБ'ЄКТ КІБЕРАТАК

Балюк С.І.
Керівник: Міскевич О.І.
E-mail: miskevich87@gmail.com
Луцьк, Луцький національний технічний університет

Активний розвиток цифрових технологій сприяв формуванню нових фінансових інструментів, серед яких особливе місце займають криптовалюти. Вони використовуються не лише як платіжний засіб, а й як об'єкт інвестування та елемент фінансової інфраструктури. Разом із поширенням криптовалют зростає і кількість загроз, пов'язаних з їх несанкціонованим використанням. Збільшення кількості користувачів та обсягів операцій із цифровими активами супроводжується інтенсифікацією кібератак, спрямованих на незаконне заволодіння криптовалютними ресурсами. Криптовалюти являють собою цифрові активи, функціонування яких забезпечується технологією блокчейн із застосуванням криптографічних алгоритмів. Відсутність централізованого управління, відкритість розподіленого реєстру та автономність учасників мережі створюють високий рівень стійкості системи, проте водночас формують сприятливі умови для реалізації кіберзагроз.

Найпоширенішими є атаки, спрямовані на компрометацію криптовалютних гаманців користувачів. Для цього застосовуються фішингові вебресурси, шкідливі програми, а також методи психологічного впливу. Метою таких дій є отримання доступу до приватних ключів або seed-фраз, втрата яких фактично унеможливлює відновлення контролю над цифровими активами через незворотність транзакцій у блокчейн-мережах [1].

Не менш небезпечними є атаки на криптовалютні біржі, які виконують роль посередників між користувачами та акумулюють значні фінансові ресурси. Недоліки в реалізації механізмів безпеки, помилки адміністрування або недостатній контроль доступу можуть призвести до масштабних інцидентів і значних фінансових втрат, що негативно позначається на репутації всієї криптовалютної галузі [2].

Окрему групу загроз становлять атаки на рівні блокчейн-мережі, зокрема сценарії отримання контролю над більшістю обчислювальної потужності мережі. У такому випадку зловмисник може впливати на процес підтвердження транзакцій або здійснювати подвійне витрачання коштів. Подібні атаки є найбільш ймовірними для мереж з низьким ступенем децентралізації та обмеженою кількістю учасників [3].

З метою зменшення рівня кіберризиків у криптовалютних системах доцільним є впровадження комплексу технічних і організаційних заходів захисту. Серед них варто виокремити використання апаратних засобів зберігання ключів, багатофакторну автентифікацію, регулярне оновлення програмного забезпечення та дотримання міжнародних криптографічних стандартів [4].

Таким чином, криптовалюти залишаються одним із пріоритетних об'єктів кібератак. Аналіз проблематики свідчить про те, що подальший розвиток криптовалютної інфраструктури має супроводжуватися системним підходом до забезпечення кібербезпеки.

Література

[1] Antonopoulos A. M. Mastering Bitcoin: Programming the Open Blockchain. https://github.com/bitcoinbook/bitcoinbook

[2] Conti M., Kumar E. S., Lal C., Ruj S. A Survey on Security and Privacy Issues of Bitcoin. https://arxiv.org/abs/1706.00916

[3] Narayanan A., Bonneau J., Felten E., Miller A., Goldfeder S. Bitcoin and Cryptocurrency Technologies. https://bitcoinbook.cs.princeton.edu

[4] NIST, FIPS PUB 186-4. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

ЗАСТОСУВАННЯ ЗАЛИШКОВИХ НЕЙРОННИХ МЕРЕЖ ДЛЯ АВТОМАТИЧНОЇ КЛАСИФІКАЦІЇ МОДУЛЯЦІЇ СИГНАЛІВ У СИСТЕМАХ ЦИФРОВОГО РАДІОМОНІТОРИНГУ

Бобров С.І.[1], Німич О.В.[2], Якимчук Н.М.[3]
E-mail: sergey.bobrov@snt.ua, 5356349@stud.kai.edu.ua, n.yakymchuk@lntu.edu.ua
[1,2]Київ, Національний університет «Київський авіаційний інститут»
[3]Луцьк, Луцький національний технічний університет

Сучасні цифрові системи радіозв'язку, телеметрії та дистанційного керування дедалі частіше працюють у складній радіоелектронній обстановці, де параметри сигналів і каналу є апріорно невідомими, а доступ до передавача відсутній. Це означає необхідність швидкого виявлення та ідентифікації типів сигналів у спектрі з метою контролю легітимності передавання, виявлення потенційно небезпечних або несанкціонованих випромінювань, а також оцінювання ризиків впливу завад і імітаційних дій. У таких умовах ключовим елементом є автоматична класифікація модуляції (AMC), оскільки саме вона забезпечує перехід від факту наявності сигналу до інтерпретації його структури, режиму роботи та можливого призначення [1].

Класичні підходи до AMC, що базуються на ручному конструюванні ознак та статистичних критеріях, демонструють обмежену придатність у практичних сценаріях: вони потребують стабільних умов приймання, є чутливими до шумів, частотних і фазових зсувів, а також погіршують точність при аналізі коротких фрагментів сигналу, характерних для моніторингу спектра та радіоконтролю. Це ускладнює їх використання в прикладних задачах цифрових технологій кіберзахисту, де рішення мають прийматися оперативно та з гарантованою достовірністю за дефіциту апріорної інформації.

Перспективним напрямом є застосування методів глибокого навчання, зокрема згорткових нейронних мереж, здатних автоматично формувати інформативні ознаки без залучення експертних правил [2-3]. Водночас для задач AMC у некооперативних умовах актуальною проблемою залишається стабільність навчання та точність класифікації за низьких відношень сигнал/шум і наявності апаратно-канальних спотворень. У цьому контексті особливу увагу привертають залишкові нейронні мережі (ResNet), що завдяки механізму залишкового навчання забезпечують ефективніше навчання глибоких моделей і потенційно кращу узагальнювальну здатність у складних умовах приймання.

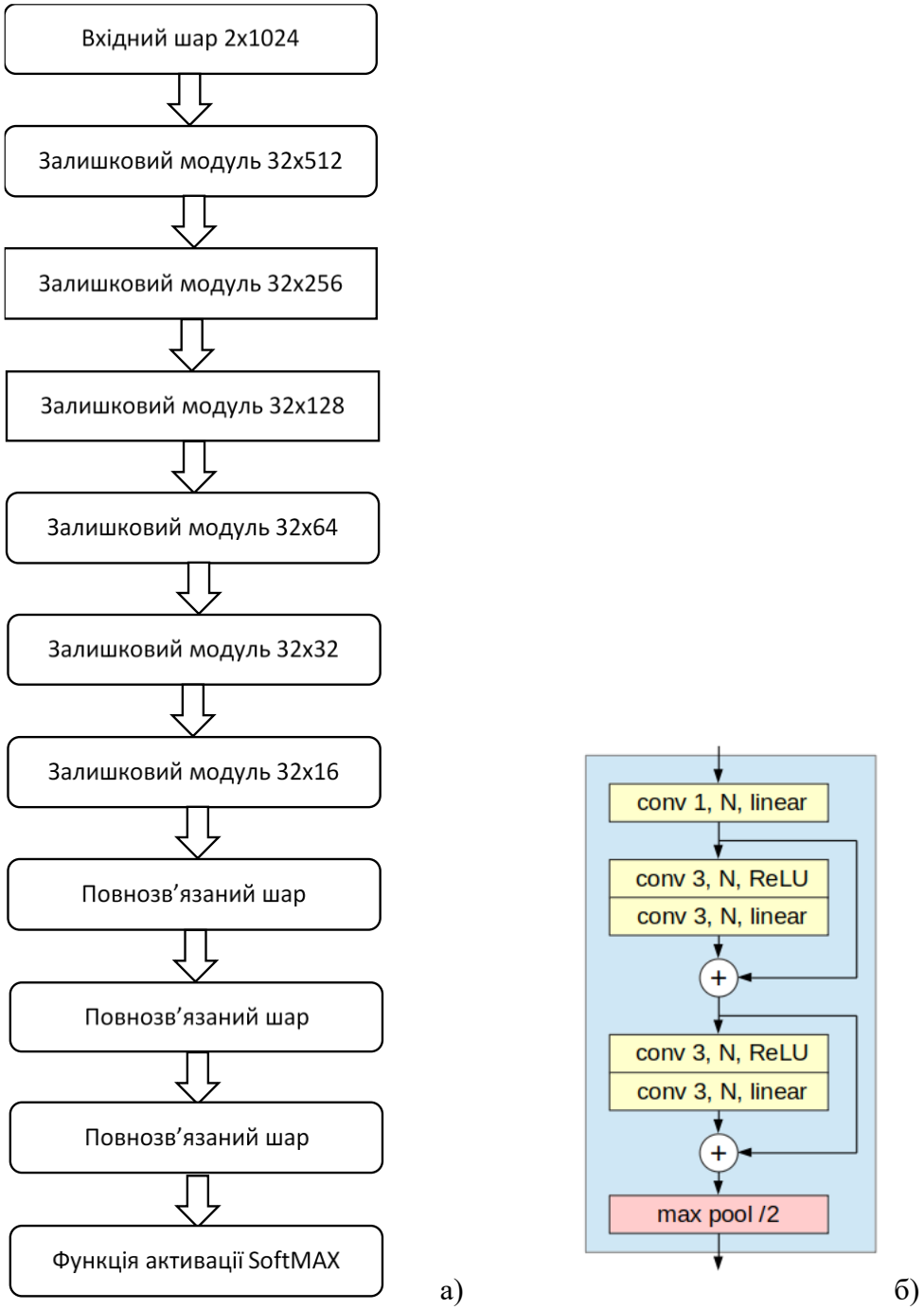
Розглядається задача автоматичної класифікації виду модуляції сигналів, що приймаються в некооперативних умовах, характерних для систем цифрового радіомоніторингу та кіберзахисту. Вихідні дані формуються у вигляді коротких фрагментів комплексного IQ-подання сигналу, отриманих у присутності адитивного шуму, частотних і фазових зсувів, а також за непостійних параметрів каналу поширення.

Формально задача зводиться до багатокласової класифікації, де за заданим вектором відліків необхідно визначити клас модуляції з наперед заданого набору (PSK, QAM, FSK тощо) без використання апріорної інформації про параметри сигналу, структуру передавача або умови поширення. Ключовими вимогами до методу є стійка робота в широкому діапазоні відношення сигнал/шум; збереження точності для коротких фрагментів сигналів; можливість застосування в задачах кібербезпеки та технічного контролю цифрових радіосистем.

Для розв'язання поставленої задачі запропоновано метод автоматичної класифікації модуляції на основі залишкової згорткової нейронної мережі (ResNet), адаптованої до оброблення IQ-подання сигналів. На відміну від класичних згорткових архітектур, у запропонованій мережі використовується механізм залишкового навчання, що забезпечує ефективніше поширення градієнтів і стабільність навчання при збільшенні глибини моделі [4].

Вхідними даними мережі є двоканальне представлення сигналу, сформоване з I- та Q-компонент, що дозволяє безпосередньо враховувати амплітудно-фазову структуру модуляції.

Архітектура ResNet містить послідовність згорткових блоків із залишковими зв'язками, що дає змогу автоматично формувати інформативні ознаки модуляції та зменшувати чутливість до шумових і частотних спотворень (рис. 1).



а) б)

Рисунок 1 – а) Структура залишкової нейронної мережі ResNet автоматичного розпізнання модуляції б) Модуль залишкової нейронної мережі (ResNet)

Запропонований підхід орієнтований на використання в цифрових системах моніторингу та кіберзахисту радіоканалів, де важливими є швидка обробка даних, узагальнювальна здатність моделі та можливість роботи без жорсткої прив'язки до конкретних стандартів передавання.

Ефективність запропонованого підходу оцінювалася на наборах IQ-фрагментів різних видів модуляції в умовах, характерних для цифрового радіомоніторингу: наявність адитивного шуму (AWGN), зміна відношення сигнал/шум (SNR) у широкому діапазоні та вплив типових апаратно-канальних спотворень (зокрема частотних зсувів). Для порівняння

використовувалась базова згорткова нейронна мережа (CNN) [5] зі співставною кількістю параметрів.

Запропонована ResNet-архітектура демонструє вищу середню точність автоматичної класифікації модуляції у широкому діапазоні умов приймання, ніж базові CNN, особливо за низьких SNR та при наявності спотворень [6]. За результатами дисертаційних експериментів середня точність класифікації для запропонованої ResNet досягає 94-95% у діапазоні SNR від −10 до +25 дБ, що дає приріст 3-8% порівняно з класичними згортковими архітектурами за близької кількості параметрів моделі.

Поліпшення найбільш помітне для близьких за структурою видів модуляції, де помилки взаємного змішування класів є критичними для прикладних сценаріїв кібербезпеки (ідентифікація / атрибуція сигналів у спектрі).

Отримані результати підтверджують доцільність застосування залишкових нейронних мереж у задачах цифрового радіомоніторингу та кіберзахисту радіоканалів, де потрібне надійне розпізнавання типів сигналів за обмежених фрагментів даних і без апріорної інформації про передавач. Підвищення точності та стабільності класифікації створює основу для подальших процедур технічного контролю режимів випромінювання, виявлення відхилень від очікуваних профілів сигналів та підтримки рішень у системах спектрального нагляду.

Література

[1] Abbas A., Pano V., Mainland G., Dandekar K. Radio modulation classification using deep residual neural networks. MILCOM 2022 - 2022 IEEE Military Communications Conference. 2022. PP. 311-317. DOI:10.1109/MILCOM55135.2022.10017640.

[2] Wang D., Lin M., Zhang X., Huang Y. Automatic modulation classification based on CNN-Transformer graph neural network. Sensors (Basel). 23 (16): 7281, 2023. DOI:10.3390/s23167281.

[3] Harper C. A., Thornton M. A., Larson E. C. Automatic modulation classification with deep neural networks. Electronics. 12 (18): 3962, 2023. DOI:10.3390/electronics12183962.

[4] Zheng Q., Tian X., Yu Z., Wang H., Elhanashi A., Saponara S. DL-PR: generalized automatic modulation classification method based on deep learning with priori regularization. Engineering Applications of Artificial Intelligence. 121: 106082, 2023. DOI: 10.1016/j.engappai.2023.106082.

[5] Nasir S. Sheikh S. A., Mumtaz F. Automatic modulation classification using convolutional neural network and support vector machine. SSRN. 2024. 22 p. DOI:10.2139/ssrn.4939905.

[6] Tian X., et al. A survey on deep learning enabled automatic modulation classification methods: Data representations, model structures, and regularization techniques. Signal Processing. Vol 242, 1104444, 2025. DOI: 10.1016/j.sigpro.2025.110444.

МОДЕЛЬ ЦИФРОВОГО ПОРТРЕТА СУБ'ЄКТА ЯК РОЗШИРЕННЯ РІШЕННЯ UEBA

Божаткін С.М.[1], Гусєва-Божаткіна В.А.[2], Пасюк Б.Б.[3]
E-mail: sergii.bozhatkin@nuos.edu.ua, gusevabozh@meta.ua, b.pasiuk@ukma.edu.ua
[1,2]Миколаїв, Національний університет імені адмірала Макарова,
[3]Київ, Національний університет «Києво-Могилянська академія»

Сьогодні фахівці з безпеки переорієнтувалися з захисту комп'ютерних систем на захист та аналіз цифрових ідентифікаційних даних. Останні дослідження показують, що традиційні методи перевірки ідентичності та контролю доступу стали менш ефективними, оскільки зловмисники застосовують способи маніпулювання людськими помилками не лише з використанням методів соціальної інженерії.

Дослідження є актуальним, оскільки існує нагальна потреба в розробці математичних моделей, які можуть пояснити поведінку користувачів, їхню схильність до соціальної інженерії та рівень компрометації облікових даних. Цифровий портрет суб'єкта пропонується розглядати не як статичну колекцію атрибутів, а як динамічний багатовимірний вектор у просторі ознак, що базується на даних соціальних мереж (SOCMINT), відкритих джерел (OSINT) та DarkNet (тіньового сегмента мережі Інтернет).

Для захисту організацій цей інструмент допомагає нам краще використовувати аналітику поведінки користувачів та об'єктів (UEBA), щоб виявляти внутрішні загрози та незвичайну активність, перш ніж вони завдадуть шкоди активам компанії. Щоб визначити загрозу для компанії, математичне моделювання дозволяє автоматизувати процес розвідки та виявляти найслабші цілі шляхом аналізу соціальних зв'язків та аналізу контенту. [1]

Протягом останніх кількох років цифровий слід перетворився не просто на запис подій, а на складну мережу показників. Згідно з науковими дослідженнями, цифровий слід має два сегменти: активний та пасивний. Користувач свідомо створює активний цифровий слід, публікуючи контент, відповідаючи на коментарі та здійснюючи транзакції, щоб контролювати свій онлайн-імідж. Пасивний слід включає в себе мовчазний збір інформації про телеметрію пристрою, метадані з'єднання, журнали DNS та активність P2P-мережі, що зазвичай дає чіткіше уявлення про справжню поведінку та наміри людини. [2, 3]

Це вказує на те, що цифровий двійник людини (HDT) – це повний набір цифрових слідів, які можна використовувати для розробки імітаційної моделі реакції користувача на певні подразники. За допомогою великих даних та когнітивного моделювання за допомогою складних математичних інструментів можна передбачити реакцію співробітника на фішингове електронне повідомлення. [4, 5]

Необхідно чітко розмежовувати дані, доступні у відкритому доступі (Clear Web), та дані, що циркулюють у закритих спільнотах та на чорних ринках (Dark Web). Інтеграція цих розрізнених джерел у єдину модель є ключовим викликом, який ми намагаємося вирішити. Векторизація даних з форумів кіберзлочинців та зіставлення їх з профілями у LinkedIn або Facebook дозволяє виявляти приховані зв'язки та потенційні вектори атак, що раніше залишалися непоміченими. [6]

Для побудови надійної математичної моделі необхідно визначити базові примітиви та простір, у якому вони існують. Ми пропонуємо використовувати векторний простір станів, де кожен користувач U представлений вектором v розмірності n.

Якщо ми говоримо про розширення UEBA, то у традиційних системах ідентифікація є бінарною: користувач або аутентифікований, або ні. У запропонованій моделі ідентичність є неперервною величиною. Відповідно до останніх досліджень цифрова ідентичність може бути описана як набір атрибутів A_d у домені d. Проте для задач кібербезпеки цього недостатньо. Ми розширюємо це визначення, пропонуючи введення понять «вектора поведінки» та «вектора компрометації».

Нехай F - це простір ознак, що включає:

– лексичні ознаки (стилометрія, словниковий запас, частота використання специфічних термінів);

– графові ознаки (центральність у соціальному графі, щільність зв'язків, приналежність до спільнот);

– поведінкові ознаки (часові патерни активності, динаміка кліків, клавіатурний почерк, навігаційні патерни);

– технічні ознаки (IP-адреси, User-Agent, fingerprint браузера та інші цифрові сліди);

Кожен користувач у момент часу t описується вектором стану $S_t \in F$. Еволюція цього вектора:

$$\Delta S = S_{t+1} - S_t \quad (1)$$

відображає динаміку цифрового портрета суб'єкта. Значні відхилення ΔS можуть свідчити про аномалію (наприклад, злом або втрату облікового запису) або зміну поведінкової моделі (insider threat).

Чим більше вхідних даних (кращої якості та більшої кількості) подаються в модель цифрового портрету, тим кращий її вихідний результат. Ми визначаємо три рівні збору інформації, які необхідно обробляти по-різному:

– поверхневий рівень включає векторизацію текстових даних (NLP)

– глибокий рівень містить метадані торрент-мереж, іншу телеметрію за поведінковими ознаками

– тіньовий рівень містить інформацію про суб'єкта з мережі DarkNet.

Дослідження доводять, що пасивний цифровий слід здатний ідентифікувати особу набагато краще, ніж паспортні дані. Збір та обробка таких даних підпадають під дію, з одного боку, Загального регламенту про захист даних (GDPR) в ЄС, а з іншого - законодавство України щодо захисту персональних даних.

Надалі планується використовувати гібридну модель, що складається з детермінованих матричних методів та ймовірнісних алгоритмів машинного навчання.

Вкрай важливо впроваджувати методи профілювання, що зберігають конфіденційність. Вектори ознак слід зберігати в зашифрованому вигляді або використовувати методи диференціальної конфіденційності, щоб унеможливити зворотне проектування вихідних даних. Аналітики повинні працювати з агрегованими оцінками ризику, а не з необробленими даними про особисте життя співробітників.

Використання моделей машинного навчання (ML) в оцінці довіри поширюватиме та навіть посилюватиме дискримінацію, наприклад, шляхом неправильного позначення відмінностей у звичайних моделях спілкування як аномалій і тим самим незаконного приписування поганих намірів користувачам з певної культурної групи. У зв'язку з цим моделі повинні постійно піддаватися аудиту упередженості (аудиту справедливості). Прийняття рішень має бути прозорим. [7]

Для команди аналітиків та захисту компанії від Insider Threat цифровий портрет співробітника є, по суті, еталоном його нормальної поведінки. Таким чином, система (на базі розроблюваної математичної моделі) має постійно оновлювати матрицю взаємодій суб’єкта в мережі підприємства та поза нею. Нові ненульові елементи матриці, що означають доступ до нових серверів, файлів або різку зміну ваги критеріїв, що означає збільшення обсягу трафіку, наприклад, призведуть до збільшення відхилення ΔS. Якщо значення перевищує певний поріг, то це аномальна поведінка — команді захисту миттєво необхідно реагувати згідно з політиками та SLA.

Одним із ключових факторів є моніторинг DarkNet-мережі на наявність витоків корпоративних даних (особливо документів та облікових даних співробітників). Завдяки використанню баз даних витоків облікових даних (Threat Intelligence системи) стає можливим швидше реагувати на проблеми безпеки. Такі послуги надають всесвітньо відомі

вендори: Flare, SOCRadar, DeHashed, StealSeek, LeakBase та ін. Люди досить часто повторно використовують паролі, що робить цей вектор атаки (Credential Stuffing) дуже ефективним. Математично це можна зобразити у вигляді графа, де вершини — це ідентифікатори (електронна пошта, логін), а ребра — спільні облікові дані (паролі, хеші). Пошук компонентів зв'язку такого графа забезпечить альтернативні шляхи входу в систему, які можна виявити під час тестування на проникнення. [8]

Таким чином, можливим стає проєктування адаптивної аутентифікації. Наприклад, інтеграція відповідних метрик у системи керування ідентифікацією та доступом (IAM). Якщо відповідна метрика у користувача зростає (наприклад, його пароль було знайдено в останньому дампі та це відображає Threat Intelligence система), тоді відбувається автоматичне підвищення вимог до автентифікації користувача. Наприклад, відбувається перехід на апаратні ключі FIDO2 або відбувається негайне відключення користувача від критичних систем з подальшим розслідуванням інциденту з боку команди захисту. [9]

Література

[1] Beyond Firewall: Leveraging Machine Learning for Real-Time Insider Threats Identification and User Profiling. Saif Al-Dean Qawasmeh, Ali Abdullah S. AlQahtani. 18.02.2025 року. [Electronic resource]. – Resource access mode: https://www.mdpi.com/1999-5903/17/2/93

[2] Assessment of the Digital Footprint of a Participant in Internet Interactions Using Mathematical Statistics Methods. E.A. Ostanina, E.V. Pokolodina. 08.2025 [Electronic resource]. – Resource access mode: https://www.researchgate.net/publication/395010377_Assessment_of_the_Digital_Footprint_of_a_Participant_in_Internet_Interactions_Using_Mathematical_Statistics_Methods

[3] Breadcrumbs in the Digital Forest: Tracing Criminals through Torrent Metadata with OSINT. Annelies de Jong, Giuseppe Cascavilla, Jessica De Pascale. 04.01.2026 [Electronic resource]. – Resource access mode: https://arxiv.org/html/2601.01492v1

[4] Toward Human Digital Twins for Cybersecurity Simulations on the Metaverse: Ontological and Network Science Approach. Tam N Nguyen. 30.04.2022 [Electronic resource]. – Resource access mode: https://pmc.ncbi.nlm.nih.gov/articles/PMC10414334/

[5] Human Digital Twin: The Future of Personalized AI and Virtual Identity.Tam N Nguyen, Tam N Nguyen. 2025 [Electronic resource]. – Resource access mode: https://www.infosys.com/services/incubating-emerging-technologies/documents/future-of-personalized-ai.pdf

[6] RLINK: Deep reinforcement learning for user identity linkage. Xiaoxue Li, Yanan Cao, Qian Li, Yanmin Shang, Yangxi Li, Yanbing Liu, Guandong Xu. 28.11.2019 [Electronic resource]. – Resource access mode: https://www.researchgate.net/publication/343519397_RLINK_Deep_reinforcement_learning_for_user_identity_linkage

[7] LLMs for Cybersecurity in the Big Data Era: A Comprehensive Review of Applications, Challenges, and Future Directions. Aristeidis Karras, Leonidas Theodorakopoulos, Christos Karras, Alexandra Theodoropoulou, Ioanna Kalliampakou, Gerasimos Kalogeratos. 4.11.2025 [Electronic resource]. – Resource access mode: https://www.mdpi.com/2078-2489/16/11/957

[8] OSINT-Based Threat Intelligence: Investigating Leaked Data on the Dark Web. Dr. Mukesh Patidar. 14.03.2025. [Electronic resource]. – Resource access mode: https://ijsrem.com/download/osint-based-threat-intelligence-investigating-leaked-data-on-the-dark-web/

[9] France – Sovereign Intelligence Assessment FR-2026-INT: Systemic Failure of Identity Governance and Multi-Vector Exfiltration of Law Enforcement Repositories by ShinyHunters-Linked Actors . 18.01.2026. [Electronic resource]. – Resource access mode: https://debuglies.com/2026/01/18/france-sovereign-intelligence-assessment-fr-2026-int-systemic-failure-of-identity-governance-and-multi-vector-exfiltration-of-law-enforcement-repositories-by-shinyhunters-linked-actors/

ІНТЕГРАЦІЯ PENETRATION TESTING У ЖИТТЄВИЙ ЦИКЛ РОЗРОБКИ БЕЗПЕЧНИХ ВЕБЗАСТОСУНКІВ

Волошенюк В.О.
Керівник: Старкова О.В.
E-mail: Viktoriia.Volosheniuk@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

Стрімкий розвиток інформаційних технологій та масове впровадження веб-додатків у сферах фінансів, електронної комерції, електронного урядування й освіти зумовили зростання кількості атак, спрямованих на порушення конфіденційності, цілісності та доступності інформаційних ресурсів. Значна частина інцидентів інформаційної безпеки пов'язана не з унікальними експлойтами, а з повторюваними вразливостями програмного забезпечення, що виникають на етапах проєктування, реалізації та тестування вебзастосунків.

У роботі розглядається комп'ютерна складова забезпечення безпеки вебзастосунків шляхом поєднання практик penetration testing та концепції Secure Software Development Life Cycle (Secure SDLC). Основна увага приділяється програмній архітектурі веб-додатку, механізмам автентифікації й авторизації, обробці користувацького вводу, а також типовим помилкам реалізації серверної та клієнтської логіки, що призводять до виникнення вразливостей.

У межах огляду сучасних підходів до забезпечення безпеки вебзастосунків проаналізовано наукові публікації, галузеві стандарти та технічну документацію, що описують методи виявлення та усунення програмних вразливостей. Особливу увагу приділено рекомендаціям OWASP, зокрема OWASP Top 10 та OWASP Web Security Testing Guide, які визначають найбільш критичні ризики для веб-додатків і практичні методи їх тестування [1, 2]. Також розглянуто концепцію Secure SDLC як системний підхід до інтеграції механізмів безпеки на всіх етапах життєвого циклу програмного забезпечення – від аналізу вимог і проєктування архітектури до тестування та експлуатації [3]. Аналіз існуючих програмних інструментів динамічного та статичного аналізу коду дозволив обґрунтувати доцільність поєднання автоматизованих засобів тестування з ручним аналізом логіки веб-додатків для досягнення більш повного виявлення вразливостей.

У продовженні дослідження передбачається проєктування та реалізація навчального веб-додатку зі штучно інтегрованими вразливостями. Такий підхід дозволяє на практичному рівні продемонструвати процес виявлення дефектів безпеки за допомогою інструментів тестування на проникнення, зокрема динамічного аналізу вебзастосунків (DAST), а також ручних методів аналізу HTTP-запитів і логіки роботи застосунку [1].

Особливу увагу приділено найбільш поширеним класам вразливостей веб-додатків, серед яких SQL-ін'єкції, міжсайтовий скриптинг (XSS), міжсайтові запити підробки (CSRF) та порушення контролю доступу. Для кожного типу вразливостей аналізується причина їх виникнення на рівні програмного коду та конфігурації, а також пропонуються методи усунення відповідно до рекомендацій Secure SDLC та галузевих стандартів безпеки [2, 3].

Результати тестування використовуються для коригування архітектурних рішень і внесення змін у програмну реалізацію вебзастосунку. Це дозволяє оцінити ефективність впровадження безпечних практик розробки, а також продемонструвати зменшення площини атаки після усунення вразливостей. Отримані результати можуть бути використані як навчальна та методична база для підготовки фахівців у галузі кібербезпеки та для підвищення рівня захищеності реальних веб-додатків.

Література

[1] OWASP Foundation. OWASP Web Security Testing Guide [Електронний ресурс]. URL: OWASP Web Security Testing Guide | OWASP Foundation

[2] OWASP Foundation. OWASP Top 10: Web Application Security Risks [Електронний ресурс]. URL: https://owasp.org/www-project-top-ten/

[3] McGraw G. Software Security: Building Security In. – Addison-Wesley, 2006.

МЕТОДОЛОГІЯ ВИЗНАЧЕННЯ ШЛЯХІВ ЗБЕРІГАННЯ ЦИФРОВИХ ДОКАЗІВ ДЛЯ FORENSICS-АНАЛІЗУ ПІСЛЯ ВИДАЛЕННЯ ВІДОМИХ ANDROID-ДОДАТКІВ

Гапоненко Є.А.
E-mail: ye.haponenko@ukma.edu.ua
Київ, Національний університет «Києво-Могилянська академія»

Цифровізація перетворила мобільні пристрої на основні інструменти роботи з конфіденційними даними. Крім того, доступність Інтернету, яку пропонують оператори мобільного зв'язку, також робить мобільні пристрої зручними для виконання багатьох завдань, які раніше переважно виконували за допомогою ПК. Це призвело до накопичення значних обсягів цифрових артефактів, які мають критичне значення для розслідування інцидентів інформаційної безпеки.

Через таку поширеність смартфонів мобільна криміналістика (Digital Forensics) стає дедалі актуальнішою та набуває статусу фундаментальної дисципліни у сферах виявлення шахрайства, протидії кіберзлочинності та захисту персональних даних. Однак складність сучасних мобільних операційних систем та постійне оновлення протоколів безпеки створюють додаткові виклики для спеціалістів у сфері цифрової криміналістики. [1]

Цифрова форензика у контексті мобільних пристроїв - це процес науково обґрунтованого пошуку, вилучення та аналізу доказів, що зберігаються у цифровій формі. Таким чином, наше дослідження спрямоване на розробку методології, яка дозволить ідентифікувати приховані шляхи зберігання таких доказів, що є надзвичайно важливим для розслідування кіберзлочинів та захисту персональних даних. В даному випадку, ми говоримо про відомі додатки - месенджери. [2]

Стандартна процедура деінсталяції додатка в ОС Android фокусується на очищенні директорії */data/data/<package_name>*, проте цифрові артефакти додатків зберігаються не лише за цим шляхом. Процес видалення відбувається через системну службу “Package Manager Service” (PMS), яка працює за наступним алгоритмом:

– зупинка всіх активних процесів та служб, що належать до конкретного ID додатка.
– видалення даних із внутрішнього приватного сховища (*/data/data/*).
– видалення APK-файлу із директорії */data/app/*.
– видалення записів про дозволи та конфігурації із системних файлів.

Видалення додатку - це лише видалення його виконуваних файлів і локальних файлів баз даних (БД), але не повне очищення слідів його перебування на пристрої. Описаний вище алгоритм PMS ігнорує дані, що були делеговані системним сервісам або були збережені у спільних директоріях. Як наслідок, артефакти та залишкові дані часто можна знайти в інших системних файлах та зовнішніх директоріях, що створює додаткові проблеми безпеки та можливість вилучення такої інформації поза “пісочницею” самого додатку. [2]

Месенджери не є ізольованими від інших файлів операційної системи. Для свого функціонування вони використовують інші системні служби (буфер обміну, клавіатуру, медіа, мікрофон тощо), які мають власну логіку роботи та логування. Дослідження доводить, що навіть після деінсталяції застосунку в системних файлах залишаються записи взаємодії таких служб із додатком. Вказані дані не підпадають під дію алгоритмів очищення, оскільки вони є незалежними від файлів застосунку.

Відсутність механізмів шифрування у системному буфері обміну ОС Android може призводити до витоку конфіденційної інформації (sensitive information). Оскільки об’єкти буферу обміну доступні для зчитування будь-якому процесу з відповідним дозволом, а також зберігаються у вигляді відкритого тексту (plain text), копіювання чутливих даних створює умови для їхнього перехоплення стороннім ПЗ. [3] Для мінімізації ризиків витоку інформації через буфер обміну, слід вжити відповідних заходів:

– для критичних полів (логіни, паролі, CVV-коди) розробникам слід вимикати контекстне меню копіювання.

– впровадження програмного таймера, який автоматично очищує вміст буфера обміну через 30–60 секунд після копіювання текстових даних.

– якщо дані копіюються для перенесення до іншого застосунку того самого розробника, дані мають бути зашифровані всередині додатка-джерела. Додаток-отримувач дешифрує його лише після внутрішньої перевірки цілісності та походження. [4]

Література

[1] Aguirregomezcorta L. and García del Pozo J. M., «Mobile forensics for Android devices» Jönköping University [Електронний ресурс] – Режим доступу: https://www.diva-portal.org/smash/get/diva2:1977641/FULLTEXT01.pdf

[2] Medium. Explore the Android File System Hierarchy In-Depth [Електронний ресурс] – Режим доступу: https://medium.com/thesecmaster/explore-the-android-file-system-hierarchy-in-depth-3cdc67234e7b#39b5

[3] Wikipedia. Data remanence [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/Data_remanence

[4] Wikipedia. End-to-end encryption [Електронний ресурс] – Режим доступу: https://en.wikipedia.org/wiki/End-to-end_encryption

КРИПТОГРАФІЧНО ВЕРИФІКОВАНИЙ ЗАХИЩЕНИЙ ДОКУМЕНТООБІГ У СЕРЕДОВИЩАХ З ОБМЕЖЕНИМ ДОСТУПОМ НА ОСНОВІ DLT

Долгова Н.Г.
E-mail: natalya.dolgova@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

Актуальність досліджень у напрямі захищеного електронного документообігу зумовлена тим, що цифрові документи набули статусу юридично значущих артефактів, а процеси їх створення, узгодження, підписання та архівування дедалі частіше виконуються у розподілених організаційних середовищах із залученням багатьох учасників і зовнішніх контрагентів. За таких умов критично важливими стають не лише конфіденційність і доступність, а й підзвітність та відтворюваність подій: необхідність достовірно встановити, хто, коли та на якій підставі виконував операції над документом, а також довести незмінність історії версій. Практичну значущість цієї проблематики підсилює зростання інцидентів, що впливають на корпоративні сховища та сервіси доступу до даних (зокрема, інцидентів із застосуванням шкідливого ПЗ і ransomware), які створюють ризики блокування доступу, втрати контрольованості над версіями та підриву доказової цілісності процесів [1].

У сучасній літературі простежується стійка тенденція застосування технологій розподілених реєстрів (DLT/блокчейн) як інструменту формування незмінного журналу подій, фіксації часових міток і забезпечення трасованості документних операцій. Показано, що використання timestamps та смарт-контрактів дає змогу реєструвати дії над документами, контролювати їхню цілісність та забезпечувати перевірюваність історії змін у межах офісного документообігу [2]. У прикладних сценаріях управління документацією ефективність підходів на основі блокчейну пов’язується з незворотною логікою погодження, незмінною реєстрацією змін та підтриманням цілісності історії версій через реєстрові структури даних і смарт-контракти [3]. Водночас для середовищ із обмеженим доступом принциповим є підхід, за якого DLT використовується не як сховище контенту, а як доказовий шар: у реєстрі зберігаються лише хеш-значення, часові мітки, посилання на підписи та події, тоді як самі документи розміщуються поза ланцюгом (off-chain) у зашифрованому вигляді з керованим доступом.

Архітектурний підхід SDMS (Secure Document Management System) реалізує розділення на контентний і доказовий контури та орієнтований на розподілене, верифіковане й криптографічно захищене опрацювання документів у середовищах із підвищеними вимогами до безпеки. Контентний контур забезпечує зберігання документів лише у зашифрованому вигляді у зовнішньому сховищі, тоді як доказовий контур на основі DLT

фіксує події життєвого циклу документа (створення, редагування, погодження, підписання, архівування, зміна політик доступу), їхні часові мітки, криптографічні відбитки (хеші) та атрибути підписів. Практики децентралізованого управління документами додатково підкреслюють доцільність відокремлення контенту від механізмів перевірки та використання криптографічних примітивів (зокрема схем розподілу секрету) для підвищення стійкості до компрометації окремих компонентів інфраструктури [4].

Забезпечення незмінності історії змін у SDMS досягається формалізацією життєвого циклу документа як послідовності криптографічно зв'язаних станів (Document State Chain). Кожна значуща операція породжує новий стан документа, що містить хеш версії, метадані (ідентифікатор, версія, статус, політика доступу, учасники маршруту), підписи та часову мітку. Новий хеш визначається як функція попереднього стану та поточних атрибутів, завдяки чому забезпечується перевірювана зв'язність: ретроспективна підміна ранньої версії або її атрибутів призводить до невідповідності хеш-ланцюга і виявляється процедурою верифікації. Такий підхід розширює подієве журналювання до формально перевірюваної моделі історії документа і підтримує доказовість не лише окремого стану, але й цілісності всієї послідовності станів.

Література

[1] Verizon. 2024 Data Breach Investigations Report (DBIR): Executive Summary [Електронний ресурс]. – 2024. – Режим доступу: https://www.verizon.com/business/resources/reports/2024-dbir-executive-summary.pdf

[2] Zhai X., Pang S., Wang M., Qiao S., Lv Z. TVS: a trusted verification scheme for office documents based on blockchain // Complex & Intelligent Systems. – 2023. – Vol. 9. – P. 2865–2877. – DOI: 10.1007/s40747-021-00617-1. – Режим доступу: https://doi.org/10.1007/s40747-021-00617-1

[3] Das M., Tao X., Liu Y., Cheng J. C. P. A blockchain-based integrated document management framework for construction applications // Automation in Construction. – 2022. – Vol. 133. – Art. 104001. – DOI: 10.1016/j.autcon.2021.104001. – Режим доступу: https://doi.org/10.1016/j.autcon.2021.104001

[4] Han J., Kim H., Eom H., Son Y. A decentralized document management system using blockchain and secret sharing // Proceedings of the 36th Annual ACM Symposium on Applied Computing (SAC 2021). – Association for Computing Machinery, 2021. – P. 305–308. – DOI: 10.1145/3412841.3442077. – Режим доступу: https://doi.org/10.1145/3412841.3442077

ПАТЕРНИ ОРКЕСТРУВАННЯ У МУЛЬТИАГЕНТНИХ СИСТЕМАХ АНАЛІЗУ МЕРЕЖЕВОГО ТРАФІКУ

Євлампієв В.Ю.
Supervisor: Бурлаченко І.С.
E-mail: gekko.vladyslav@gmail.com, ivan.burlachenko@chmnu.edu.ua
Миколаїв, Чорноморський національний університет імені Петра Могили

Сьогодення неможливо уявити без глобальних мереж. Інтернет речей, хмарні обчислення, банківські операції – все це генерує просто колосальні об'єми трафіку щосекунди. І там, де є трафік, завжди є ті, хто хоче використати його для атак. Кіберзагрози еволюціонують навіть швидше, ніж засоби захисту: від примітивних DDoS-атак до хитрих сканувань портів, які намагаються знайти вразливість в системі непомітно для адміністратора. Класичний підхід, коли системний адміністратор вручну переглядає логи або налаштовує прості статичні правила на фаєрволі, вже не працює ефективно. По-перше, даних занадто багато. По-друге, реакція має бути миттєвою. Якщо система впаде від перевантаження трафіком, компанія втратить гроші ще до того, як адміністратор отримає повідомлення про збій.

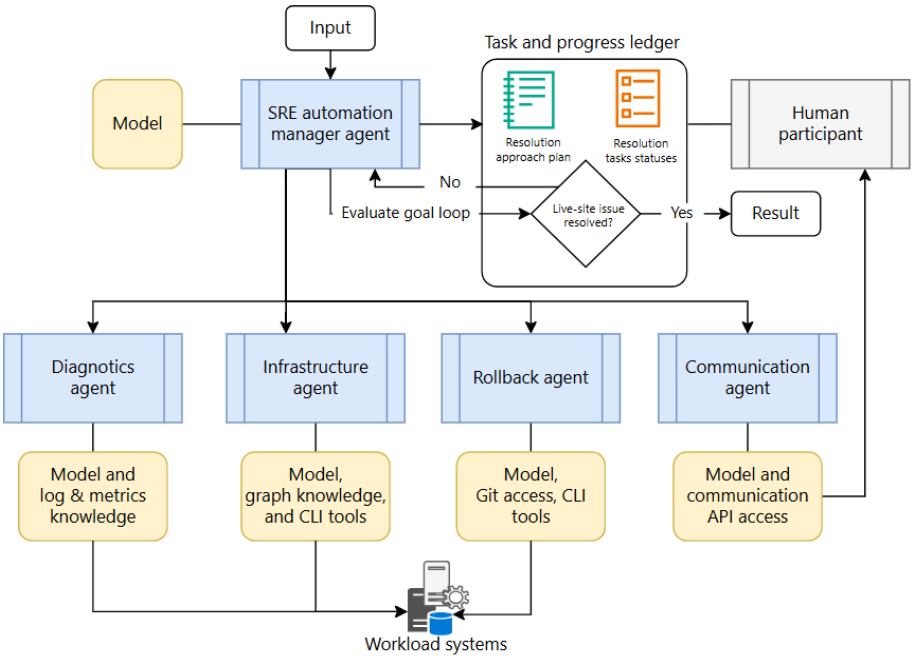


Рис. 1 – Архітектура МАС з використанням патерна оркестрування Magentic.

Тому зараз на перше місце виходять інтелектуальні автоматизовані системи, здатні аналізувати потоки даних у реальному часі. Але тут виникає інша проблема – архітектурна. Більшість старих систем моніторингу це моноліти. Вони важкі, їх складно масштабувати. Якщо модуль аналізу не витримає навантаження, впаде і модуль веб-інтерфейсу, і модуль сповіщень. Це неприпустимо для систем безпеки. Саме тому розробка сучасних архітектур рухається в бік мультиагентних систем (МАС), де агенти можуть бути представленими у вигляді спеціалізованих мікросервісів [1]. Розділення на незалежні агентні сервіси [2] дозволяє системі функціонувати, навіть якщо один з компонентів тимчасово вийшов з ладу, а також легко додавати нові ефективні функції (наприклад, ще один екземпляр агента аналізатора) без зупинки всього програмного комплексу.

Для відділу забезпечення надійності вебсайту (SRE) було спроєктовано автоматизацію, яка використовує оркестрацію за допомогою патерну Magentic для обробки сценаріїв реагування на інциденти з низьким рівнем ризику. Коли в рамках автоматизації відбувається збій у роботі сервісу, МАС повинна динамічно створювати та впроваджувати план усунення несправностей. МАС робить це, не знаючи заздалегідь конкретних необхідних кроків. Коли автоматизація (рис. 1) виявляє кваліфікаційний інцидент, агент-менеджер починає роботу зі створення початкового журналу завдань із високорівневими цілями, такими як відновлення доступності послуг та визначення першопричини. Потім агент-менеджер консультується зі спеціалізованими агентами для збору інформації та уточнення плану усунення несправностей. Агент діагностики аналізує системні журнали, показники продуктивності та шаблони помилок, щоб виявити потенційні причини. Він повідомляє про результати агенту-менеджеру. На основі результатів діагностики агент-менеджер оновлює реєстр завдань, вказуючи конкретні кроки розслідування, та консультується з агентом-інфраструктури, щоб зрозуміти поточний стан системи та доступні варіанти відновлення.

Комунікаційний агент забезпечує можливості сповіщення зацікавлених сторін, а керуючий агент включає контрольні точки комунікації та шлюзи затвердження в план, що розвивається, відповідно до процедур ескалації команди SRE. Коли сценарій стає зрозумілішим, агент-менеджер може додати агента-відкату до плану, якщо потрібне повернення до попереднього стану розгортання, або передати інцидент інженерам SRE, якщо інцидент виходить за межі автоматизації.

Протягом цього процесу агент-менеджер постійно уточнює завдання в журналі на основі нової інформації. Агент-менеджер додає, видаляє або змінює порядок завдань у міру

розвитку інциденту. Наприклад, якщо агент діагностики виявляє проблему з підключенням до бази даних, агент-менеджер може переключити весь план зі стратегії відкату розгортання на план, який зосереджений на відновленні підключення до бази даних. Агент-менеджер стежить за надмірними зупинками у відновленні обслуговування та захищає від нескінченних циклів виправлення. Він веде повний журнал аудиту плану, що розвивається, та етапів впровадження, що забезпечує прозорість для перевірки після інциденту. Дана прозорість гарантує, що команда SRE може покращити як робоче навантаження, так і автоматизацію на основі отриманого досвіду.

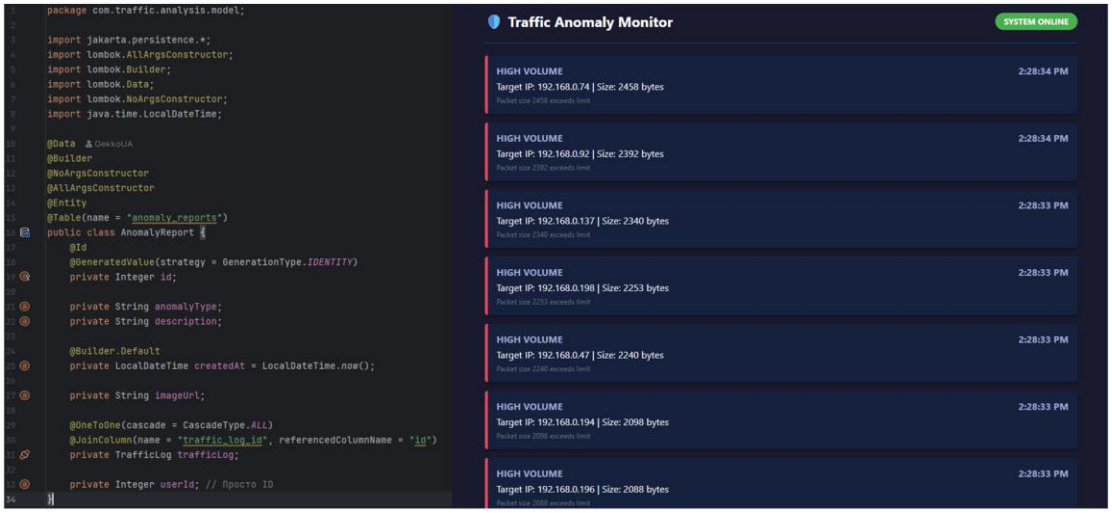


Рис. 2 – а) Сутність AnomalyReport (JPA); б) Web UI для real-time моніторингу аномалій.

Згідно з вимогами до оперативності реагування на інциденти (рис. 2-а), система повинна мати графічний інтерфейс для відображення звітів аномалій у реальному часі. Оскільки класичний підхід із періодичним оновленням сторінки Polling створює зайве навантаження на сервер і має затримки, було обрано технологію WebSockets. Клієнтська частина (рис. 2-б) реалізована як Single Page Application, що складається з HTML5-структури та JavaScript-скриптів для обробки потоку даних. На розробленій панелі керування MAC з оркестрацією на основі патерну Magentic присутня статистика. Індикатор статусу з'єднання, що візуалізує стан підключення до сервера WebSockets. Стрічка подій (Event Feed) містить динамічний список карток, що відображають деталі виявлених аномалій (тип атаки, IP-адресу зловмисника, обсяг даних). Взаємодія реалізована за протоколом STOMP (Simple Text Oriented Messaging Protocol) поверх WebSockets. При завантаженні сторінки JS-клієнт ініціює з'єднання з точкою входу /ws на сервісі сповіщень. Після успішного рукостискання (Handshake) клієнт підписується на топік /topic/alerts. Коли NotificationService отримує повідомлення з черги RabbitMQ, він пересилає його в цей топік. JS-скрипт отримує JSON-об'єкт, парсить його і динамічно додає новий DOM-елемент у верхню частину списку.

Література

[1] Burlachenko I., & Zavorotnii, D. (2025). Microservice Architecture of an Adaptive Authentication System for Users of Financial Institutions. Computer-Integrated Technologies: Education, Science, Production, (60), 429-440. https://doi.org/10.36910/6775-2524-0560-2025-60-46

[2] AI agent orchestration patterns [Electronic resource]. – Resource access mode: https://learn.microsoft.com/en-us/azure/architecture/ai-ml/guide/ai-agent-design-patterns

МЕТОДИ АНАЛІЗУ МЕТАДАНИХ PDF ТА ГРАФІЧНИХ ФАЙЛІВ ДЛЯ ВИЯВЛЕННЯ ЦИФРОВОЇ ФАЛЬСИФІКАЦІЇ ДОКУМЕНТІВ

Ємцова О.А.
Керівник: Лимаренко В.В.
E-mail: sasaemtsova@gmail.com
Харків, Харківський національний економічний університет імені Семена Кузнеця

У сучасному світі, де перехід документів у цифровий формат швидко набирає обертів, збільшується ризик підробки документів через загальновживані програмні забезпечення. Оскільки PDF-файли та зображення документів легко підробити, перевірити їхню достовірність складно, що створює додаткові ризики з безпекою для організацій та агентств.

Підробка цифрових паперів може бути здійснена шляхом створення нового фіктивного документу на основі шаблона, видаляючи або редагуючи частини зображень та змінюючи текст. Часто такі редагування залишають за собою цифрові сліди, які можна виявити за допомогою аналізу структури файлів і метаданих.

Метадані – це «інформація про дані», яка генерується під час редагування та створення файлу програмним забезпеченням. Метадані включають властивості програмного забезпечення документа, такі як його формат, вміст, а також дату та час редагування.

PDF-файли здатні зберігати вбудовані метадані: автора, дати редагувань та створення і структуру документа. Файли зображень (JPEG, RAW, PNG, TIFF) містять метадані стандарту EXIF, що зберігають параметри зйомки, модель пристрою, програмне забезпечення для відновлення та часові мітки редагування.

Розбіжність між цими параметрами на момент створення файлу є важливим показником ризику шахрайства. Наприклад, використання редакторів зображення для створення підробленого документа або розбіжності між датами створення та публікацією.

У роботі розглядаються методи аналізу метаданих за допомогою програмного забезпечення з відкритим кодом. У тому числі, бібліотеки PyPDF2 та pdfminer.six надають можливість отримати інформацію про послуги з PDF-документів і аналізувати їх структуру. Було здійснено дослідження з метою порівняння оригінальних документів та їх змінених копій у текстових та графічних редакторах. Результат підтвердив, що навіть незначні модифікації, внесені до файлу, спричиняють зміни в метаданих, які можна використовувати для виявлення несанкціонованих перетворень.

Поданий метод може бути використаний у цифрових криміналістичних розслідуваннях, у військових, освітніх та урядових організаціях для первинної перевірки достовірності електронних документів. Завдяки інструментів з відкритим кодом такі дослідження можна проводити без витрат на спеціалізоване програмне забезпечення.

Отже, аналіз метаданих PDF-файлів та графічних зображень є ефективним інструментом для виявлення підроблених цифрових документів і невід'ємною частиною кібербезпеки в сучасному світі.

Література

[1] Wikipedia. Метадані [Електронний ресурс]. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Метадані

[2] ExifTool. Інструмент для аналізу метаданих файлів [Електронний ресурс]. – Режим доступу до ресурсу: https://exiftool.org/

[3] PyPDF2 Documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://pypdf2.readthedocs.io/

[4] pdfminer.six Documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://pdfminersix.readthedocs.io/

[5] ExifRead Documentation [Електронний ресурс]. – Режим доступу до ресурсу: https://pypi.org/project/ExifRead/

ПРОТИДІЯ DOS ТА DDOS АТАКАМ: ВИКЛИКИ ТА ІНСТРУМЕНТИ ВІДКРИТОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Журавка А.В., Галань В.Я.
E-mail: andrii.v.zhuravka@lpnu.ua
Львів, Національний університет «Львівська політехніка»

DDoS (Distributed Denial of Service) атаки залишаються одними з найпоширеніших і найнебезпечніших загроз у сучасному інтернет середовищі. Їх мета полягає у виведенні з ладу сервісів шляхом перевантаження ресурсів, до яких можна віднести процесорні потужності, пам'ять чи пропускної здатності мережі. У випадку DDoS атаки використовується велика кількість заражених пристроїв (ботнетів), що робить її особливо складними для виявлення та блокування. Масштабні атаки можуть паралізувати роботу банківських систем, державних порталів чи комерційних сервісів, завдаючи значних економічних і репутаційних збитків [1].

Сучасні DDoS атаки значно відрізняються від класичних SYN flood чи UDP flood. Як показує Wang та співавтори [2], вони поширюються на нові протоколи та сервіси, включаючи IoT-системи, хмарні платформи та навіть блокчейн-мережі. Зловмисники використовують розподілені ботнети, які здатні генерувати трафік у терабітному діапазоні, а також застосовують техніки обходу традиційних систем захисту, наприклад, шифрування трафіку чи використання легітимних сервісів для маскування атаки.

Серед основних причин складності протидії можна виділити наступні [2]:

– архітектурні обмеження інтернету (TCP/IP створювався з акцентом на доступність, а не на захист від перевантажень);

– IoT-пристрої (мільйони слабко захищених пристроїв легко стають частиною ботнетів);

– шифрування (використання HTTPS та VPN ускладнює аналіз трафіку);

– адаптивність атак (зловмисники змінюють вектори в реальному часі, щоб обійти фільтри).

Глибинна причина проблеми полягає у відкритості інтернет-інфраструктури. Протоколи TCP/IP створювалися з акцентом на доступність і сумісність, а не на захист від зловмисних перевантажень [3]. Це робить можливим використання таких технік, як SYN flood, UDP flood чи HTTP flood. Ситуацію ускладнює стрімке зростання кількості IoT-пристроїв, які часто мають слабкі механізми безпеки і легко стають частиною ботнетів. Одним з найбільш відомих прикладів атаки ботнету є Mirai, що у 2016 році вивели з ладу значну частину інтернет-сервісів у США та Європі.

Протидія DoS/DDoS атакам потребує багаторівневого підходу. На мережевому рівні застосовуються системи фільтрації трафіку, які здатні відсікати підозрілі пакети ще до того, як вони досягнуть сервера. На рівні додатків важливим є використання механізмів rate limiting та CAPTCHA, що ускладнюють масові запити. На інфраструктурному рівні ефективним є розподіл навантаження через балансувальники та використання CDN-мереж, які здатні поглинати великі обсяги трафіку. Важливим напрямом є також застосування систем раннього виявлення, що використовують алгоритми машинного навчання для аналізу поведінки трафіку та прогнозування потенційних атак.

Відкриті інструменти відіграють ключову роль у цьому процесі. Такі системи як Snort чи Suricata забезпечують глибокий аналіз пакетів і дозволяють налаштовувати правила для виявлення аномалій. Fail2Ban допомагає блокувати IP-адреси, що здійснюють підозрілу активність. Wireshark використовується для детального аналізу трафіку, а Bro/Zeek використовується для моніторингу мережевої поведінки. Завдяки відкритому коду ці інструменти постійно вдосконалюються спільнотою, що забезпечує швидку реакцію на нові загрози. Важливо й те, що вони доступні для освітніх цілей, дозволяючи студентам і дослідникам вивчати реальні механізми протидії атакам.

Практичні дослідження показують, що ефективність протидії залежить від комплексності підходу. Використання лише одного інструмента не гарантує захисту: наприклад, фільтрація на рівні мережі може бути обійдена складними HTTP flood атаками, тоді як механізми CAPTCHA не здатні протистояти масовим запитам із ботнетів. Тому сучасні системи безпеки поєднують кілька рівнів захисту, інтегруючи відкриті рішення з комерційними сервісами. Це дозволяє створювати багатошарову оборону, яка значно ускладнює роботу зловмисників.

Перспективи розвитку полягають у подальшій автоматизації процесів. Алгоритми машинного навчання здатні аналізувати поведінку трафіку в реальному часі та відрізняти легітимні запити від атак. Хмарні сервіси безпеки пропонують централізоване управління політиками та здатні масштабувати ресурси під час атаки. Важливим напрямом є також розвиток освітніх програм, адже без належної обізнаності адміністраторів навіть найсучасніші технології залишаються неефективними. Лише поєднання технічних рішень, відкритих інструментів та освітніх ініціатив може забезпечити реальний захист від DoS/DDoS атак.

Таким чином, протидія DoS та DDoS атакам є багатогранним завданням, що потребує системного підходу. Використання відкритого програмного забезпечення дозволяє не лише ефективно виявляти та блокувати атаки, а й формувати культуру безпечного користування інтернет-ресурсами. Подальші дослідження мають бути спрямовані на інтеграцію інтелектуальних систем, розвиток хмарних сервісів та підвищення рівня обізнаності користувачів і адміністраторів. Лише комплексна стратегія може забезпечити стійкість цифрової інфраструктури у світі, де DoS та DDoS атаки залишаються одним із головних викликів.

Література

[1] Adedeji, K.B.; Abu-Mahfouz, A.M.; Kurien, A.M. DDoS Attack and Detection Methods in Internet-Enabled Networks: Concept, Research Perspectives, and Challenges. J. Sens. Actuator Netw. 2023, 12, 51. https://doi.org/10.3390/jsan12040051.

[2] Wang J., Yu L., Lui J.C.S., Luo X. Modern DDoS Threats and Countermeasures: Insights into Emerging Attacks and Detection Strategies. arXiv:2502.19996, 2025. https://doi.org/10.48550/arXiv.2502.19996.

[3] Hill, W., Acquaah, Y.T., Mason, J. et al. DDoS in SDN: a review of open datasets, attack vectors and mitigation strategies. Discov Appl Sci 6, 472 (2024). https://doi.org/10.1007/s42452-024-06172-x.

ПОШУК ВРАЗЛИВОСТЕЙ WI-FI: АНАЛІЗ, ІНСТРУМЕНТИ ТА ПЕРСПЕКТИВИ

Журавка А.В., Мазур М.О.
E-mail: andrii.v.zhuravka@lpnu.ua
Львів, Національний університет «Львівська політехніка»

Wi-Fi став основою сучасної цифрової екосистеми, адже саме він забезпечує мобільність, швидкий доступ до ресурсів та інтеграцію користувачів у глобальний інформаційний простір. Проте разом із поширенням бездротових мереж зростає і кількість атак, що спрямовані на їх слабкі місця. Вразливості можуть призвести до витоку конфіденційних даних, компрометації систем чи навіть масштабних кібератак, що робить питання їхнього пошуку та усунення надзвичайно актуальним. Як показали дослідження Vanhoef і Piessens [1], навіть широко використовуваний протокол WPA2 містив критичну помилку, що дозволяла здійснювати так звану KRACK-атаку. Це доводить, що безпека Wi-Fi не може розглядатися як остаточно вирішене питання, а потребує постійного моніторингу та вдосконалення.

Глибинні причини вразливостей. Проблема безпеки Wi-Fi має багатошарову природу [2]. На технічному рівні вона пов’язана з недоліками протоколів, які часто створюються як

компроміс між продуктивністю, сумісністю та безпекою. Наприклад, WEP був розроблений як простий механізм шифрування, але його ключі легко підбираються, що робить протокол фактично непридатним. WPA2 довгий час вважався надійним, однак KRACK показав, що навіть добре перевірені стандарти можуть містити фундаментальні помилки. WPA3, який мав стати вирішенням проблеми, виявився вразливим до Dragonblood атак, що свідчить про складність створення абсолютно захищеного протоколу.

На організаційному рівні вразливості виникають через неправильні налаштування обладнання [3]. Використання слабких паролів, незмінені параметри за замовчуванням, активований WPS усе це створює додаткові ризики. Виробники часто не забезпечують регулярних оновлень прошивки, залишаючи користувачів беззахисними перед новими загрозами. На рівні користувачів проблема полягає у низькій обізнаності: люди підключаються до відкритих мереж без VPN, не змінюють стандартні налаштування, стають жертвами фішингових точок доступу. Таким чином, технічні, організаційні та соціальні фактори взаємодіють, створюючи комплексну проблему.

Методи пошуку вразливостей. Пасивний аналіз трафіку дозволяє виявляти підозрілі пакети та приховані точки доступу. Такі інструменти Wireshark чи Kismet є стандартом у цій сфері, проте їхня ефективність залежить від кваліфікації дослідника: неправильна інтерпретація даних може призвести до хибних висновків.

Активне тестування проникнення (Aircrack-ng, Reaver) дає змогу перевірити стійкість мережі до атак, але воно може створювати ризики для продуктивності системи, якщо проводиться у робочому середовищі.

Слід відмітити, що сканування конфігурацій (Nmap, OpenVAS) допомагає знайти відкриті порти та відомі вразливості, проте не завжди враховує нові загрози, які ще не внесені до баз даних. Моделювання атак у лабораторних умовах є найбільш безпечним методом, але воно потребує значних ресурсів і не завжди відображає реальні сценарії.

Дослідження на понад п’ятдесяти точках доступу різних виробників показало [3], що сорок відсотків пристроїв використовують паролі за замовчуванням, чверть не підтримують WPA3, а п’ятнадцять відсотків мають активований WPS. Ці дані свідчать про системну проблему, що полягає в тому що, навіть якщо протоколи вдосконалюються, користувачі та організації часто не впроваджують базові заходи безпеки. Це означає, що технічні рішення мають супроводжуватися освітніми програмами та політиками безпеки. Інакше нові стандарти залишатимуться лише теоретично захищеними.

Роль відкритого програмного забезпечення. Відкритий софт є ключовим у виявленні вразливостей на сьогодняшній день, оскільки забезпечує прозорість алгоритмів, можливість незалежної перевірки та швидкої реакції спільноти на нові загрози. Такі сучасні проекти, як Aircrack-ng та Wireshark, стали стандартами у галузі безпеки завдяки їх саме відкритості. Через це вони стимулюють дослідників і мають освітню цінність, і використовуються у навчальних програмах. Крім того, відкритий код дозволяє адаптувати інструменти для конкретних цілей, роблячи їх більш універсальними.

Майбутнє пошуку вразливостей Wi-Fi пов’язане з автоматизацією процесів. Інтеграція методів машинного навчання дозволить прогнозувати атаки на основі поведінкових моделей. Хмарні сервіси безпеки забезпечать централізоване управління політиками Wi-Fi, що особливо важливо для великих організацій. Освітні ініціативи мають стати невід’ємною частиною стратегії: без підвищення рівня обізнаності користувачів навіть найсучасніші технології залишатимуться неефективними. Окрему увагу слід приділити IoT-пристроям, які часто мають слабкі механізми захисту і можуть стати новим полем для атак.

Пошук вразливостей Wi-Fi є комплексною проблемою, яка потребує поєднання технічних інструментів, відкритих рішень та підвищення рівня обізнаності користувачів. Використання відкритого програмного забезпечення дозволяє не лише ефективно виявляти недоліки, а й формувати культуру безпечного користування бездротовими мережами. Подальші дослідження мають бути спрямовані на автоматизацію процесів, інтеграцію

інтелектуальних систем та розвиток освітніх програм. Лише комплексний підхід від технологій до освіти може забезпечити реальний захист у світі, де Wi-Fi є основою сучасних комунікацій.

Перспективи розвитку полягають у створенні автоматизованих систем пошуку вразливостей, інтеграції методів машинного навчання для прогнозування атак та використанні хмарних сервісів безпеки для централізованого управління політиками Wi-Fi. Не менш важливим напрямом є освітні ініціативи, адже саме знання користувачів і адміністраторів часто стають вирішальним фактором у запобіганні інцидентам. Подальші дослідження мають бути спрямовані на автоматизацію процесів, інтеграцію інтелектуальних систем та розвиток освітніх програм, що у підсумку сприятиме підвищенню загального рівня кібербезпеки.

Література

[1] N. H. N. Zulkipli and M. I. B. Khusairi, "An Experimental Analysis for Public Wi-Fi Attacks," 2024 IEEE 6th Symposium on Computers & Informatics (ISCI), Kuala Lumpur, Malaysia, 2024, pp. 247-252, https://doi.org/10.1109/ISCI62787.2024.10667813

[2] Alamleh, H.; Estremera, L.; Arnob, SS.; AlQahtani, A.A.S. Advanced Persistent Threats and Wireless Local Area Network Security: An In-Depth Exploration of Attack Surfaces and Mitigation Techniques. J. Cybersecur. Priv. 2025, 5, 27. https://doi.org/10.3390/jcp5020027

[3] Ullah, A., Sakib, M.N., Rahman, M.H., Shahin, M.S.K., Hossain, F., Hossain, M.A. (2024). A Comparative Study on Vulnerabilities, Challenges, and Security Measures in Wireless Network Security. In: Abraham, A., Pllana, S., Hanne, T., Siarry, P. (eds) Intelligent Systems Design and Applications. ISDA 2023. Lecture Notes in Networks and Systems, vol 1048. Springer, Cham. https://doi.org/10.1007/978-3-031-64650-8_28

ОГЛЯД МЕТОДІВ АВТОМАТИЧНОЇ СТРУКТУРИЗАЦІЇ ЛОГІВ ТА ВИЯВЛЕННЯ АНОМАЛІЙ

Звягінцев Я. В.
Керівник: Долгова Н.Г.
E-mail: yarichek.zviagintsev@gmail.com
Харків, Харківський національний економічний університет імені Семена Кузнеця

Автоматична структуризація логів і виявлення аномалій є важливою складовою сучасних систем моніторингу, експлуатації та інформаційної безпеки. Через великі обсяги неструктурованих лог-даних ручний аналіз є практично неможливим, тому застосовуються алгоритмічні та машинні методи обробки [1][5].

Нижче розглянуто п’ять найбільш поширених підходів.

Перший підхід ґрунтується на автоматичному виділенні шаблонів логів із подальшим застосуванням статистичних методів. На початковому етапі лог-повідомлення групуються за структурною подібністю, в результаті чого формуються шаблони подій. Подальший аналіз базується на відстеженні частоти появи цих шаблонів, їх змін у часі та відхилень від історичних значень. Аномалії визначаються як статистично значущі відхилення [2][5]. Даний підхід характеризується простотою реалізації та високою інтерпретованістю результатів, однак він малоефективний для виявлення складних або контекстних аномалій.

Другий клас методів передбачає використання алгоритмів класичного машинного навчання. Після етапу структуризації логів формується набір ознак, наприклад, частоти подій, n-грамні послідовності або агреговані показники в часових вікнах. На основі цих ознак навчаються моделі для виявлення відхилень від нормальної поведінки [4][6]. Порівняно зі статистичними методами, цей підхід дозволяє виявляти більш складні аномалії, проте потребує ретельного підбору ознак і має нижчий рівень інтерпретованості.

Третій підхід базується на використанні послідовних нейронних мереж, зокрема рекурентних моделей. У цьому випадку логи розглядаються як часові послідовності подій, а модель навчається прогнозувати наступну подію або оцінювати ймовірність певної

послідовності. Аномаліями вважаються події або ланцюжки подій, які істотно відрізняються від типових сценаріїв роботи системи [3][9]. Такі моделі добре виявляють причинно-наслідкові та часові залежності, однак потребують значних обчислювальних ресурсів і складніші в налаштуванні.

Четвертий підхід використовує трансформерні архітектури та методи самонавчання. Лог-повідомлення обробляються як текстові або токенізовані послідовності, а модель навчається відновлювати або прогнозувати їх частини без використання розмічених даних. Аномалії визначаються на основі помилки відновлення або низької ймовірності спостережуваних подій [7][8]. Даний підхід демонструє високу точність для складних і семантично насичених логів, але має високі вимоги до обсягу даних і обчислювальних ресурсів.

П’ятий підхід пов’язаний із використанням автоенкодерів. Модель навчається стискати та відновлювати нормальні лог-патерни, мінімізуючи помилку реконструкції. Під час аналізу логів аномальні записи характеризуються значно більшою помилкою відновлення [5][6]. Перевагою цього методу є можливість повністю ненаглядного навчання, проте його ефективність сильно залежить від якості навчальної вибірки.

Порівнюючи наведені методи, можна зробити висновок, що шаблонні та статистичні підходи є найпростішими у впровадженні, але мають обмежені можливості. Класичні методи машинного навчання забезпечують кращий баланс між складністю та ефективністю. Послідовні нейронні мережі й трансформери дозволяють виявляти складні аномалії, проте потребують значних ресурсів і експертного налаштування. Автоенкодери займають проміжне положення та є доцільними у випадках відсутності розмічених даних.

Література

[1] He J., Zhu J., Zheng Z., Lyu M.R. Loghub: A Large Collection of System Log Datasets towards Automated Log Analytics. https://arxiv.org/abs/2008.06448

[2] Xu W., Huang L., Fox A., Patterson D., Jordan M.I. Detecting Large-Scale System Problems by Mining Console Logs. ACM SOSP ’09, DOI:10.1145/1629575.1629587. https://doi.org/10.1145/1629575.1629587

[3] Du M., Li F., Zheng G., Srikumar V. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. CCS ’17 (ACM). https://dl.acm.org/doi/10.1145/3133956.3134012

[4] Zhang Y., Roughan M., Duncan A. Machine Learning for Anomaly Detection in Network and System Logs. Journal of Network and Systems Management (2019).

[5] Chandola V., Banerjee A., Kumar V. Anomaly Detection: A Survey. ACM Computing Surveys, 2009. DOI:10.1145/1541880.1541882. https://dl.acm.org/doi/10.1145/1541880.1541882

[6] Ahmed M., Mahmood A.N., Hu J. A Survey of Network Anomaly Detection Techniques. Journal of Network and Computer Applications (2016).

[7] Jurafsky D., Martin J.H. Speech and Language Processing (3rd ed.) – Draft online chapters. https://web.stanford.edu/~jurafsky/slp3/

[8] Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. https://www.deeplearningbook.org/

[9] Graves A. Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence (2012). https://link.springer.com/book/10.1007/978-3-642-24797-2

КІБЕРБЕЗПЕКА ТА СТАНДАРТИЗАЦІЯ В IOT-СИСТЕМАХ МОНІТОРИНГУ ТВАРИН НА ОСНОВІ ПРОТОКОЛУ LORAWAN

Карлов Д.С.[1], Семенов С.Г.[2]
E-mail: dmytro.karlov@nure.com, serhii.semenov@uken.krakow.pl
[1]Харків, Харківський національний університет радіоелектроніки
[2]Краків, Університету комісії національної освіти

Стрімке впровадження технологій Інтернету речей (IoT) у сільському господарстві створює значні ризики для безпеки, особливо щодо цілісності даних та доступності мережі. Загалом ефективність інтелектуальних систем моніторингу в цій галузі критично залежить від надійності каналів передачі даних. У цій роботі розглядається метод побудови захищеної інфраструктури моніторингу здоров'я тварин з використанням протоколу LoRaWAN, що базується на відкритих стандартах та спеціалізованих алгоритмах стійкості.

Для запобігання несанкціонованому доступу пристроїв система використовує метод активації по ефіру (Over-the-Air Activation - OTAA) замість активації через персоналізацію (ABP). У цьому процесі запит на приєднання (Join Request) криптографічно підписується за допомогою унікального 64-бітного ідентифікатора пристрою DevEUI та ключа AppKey [1]. Після успішної автентифікації мережевим сервером генеруються динамічні сесійні ключі (NwkSKey для цілісності мережі та AppSKey для шифрування даних). Це гарантує, що навіть у разі фізичної компрометації одного трекера загроза буде локалізована в межах однієї сесії та не розкриє майстер-ключі всієї мережі ферми.

Критичною вразливістю бездротової телеметрії є «атака повторного відтворення», коли зловмисник записує валідні дані (наприклад, «тварина здорова») і транслює їх пізніше, щоб приховати спалах захворювання. Запропонований метод нейтралізує цю загрозу, використовуючи стандартні для LoRaWAN лічильники кадрів (fCnt) та коди цілісності повідомлень (MIC). Згідно з протоколами системи, кожен пакет uplink містить унікальний fCnt (наприклад, 244) та 4-байтний підпис MIC (наприклад, e20ee105) [2]. Сервер відхиляє будь-який пакет із порушеною послідовністю лічильника або недійсним підписом, гарантуючи, що дані не можуть бути підроблені, повторно відтворені або змінені під час передачі. Для додаткового захисту на прикладному рівні корисне навантаження використовує суворий стандарт Type-Length-Value (TLV). Завдяки використанню чітких схем на рівні байтів (наприклад, Тип 0xCC для станів тварини), система запобігає вразливостям, пов'язаним із переповненням буфера, які часто виникають через некоректно сформовані нестандартні пакети даних. Для забезпечення такої складової кібербезпеки як «доступність», система реалізує механізм захисту від масового збою [3]. У випадку відновлення електропостачання тисячі пристроїв можуть одночасно спробувати перепідключитися, створюючи умови для відмови в обслуговуванні (DoS). Система запобігає цьому, встановлюючи ліміт черги (1 875 пакетів на хвилину) та поріг максимального часу перебування в черзі (10 хвилин). Ця логіка захищає радіоінтерфейс від перевантаження, гарантуючи, що критичні сповіщення про здоров'я не будуть втрачені під час відновлення мережі. Поєднання стандартних механізмів безпеки LoRaWAN (OTAA), структурованого кодування корисного навантаження (TLV) та розробленої логіки стійкості до збоїв дозволяє створити надійну та захищену систему. Забезпечення цілісності телеметрії та експлуатаційної доступності, реалізоване у запропонованому підході, є критично важливим для кіберзахисту промислових IoT-рішень.

Література

[1] L. Butun, N. Pereira, and M. Gidlund, "Security in LoRaWAN: A Comprehensive Survey," in IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2282–2311, Fourthquarter 2022.

[2] M. A. Ferrag et al., "Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges," in IEEE Access, vol. 8, pp. 32031–32053, 2023.

[3] W. A. Jabbar et al., "LoRaWAN-Based IoT System Implementation for Smart Livestock Monitoring," in IEEE Access, vol. 9, pp. 60520-60538, 2021.

РОЗРОБЛЕННЯ СИСТЕМИ АВТОМАТИЧНОГО АНАЛІЗУ СКЛАДУ КОМПОНЕНТІВ (SBOM) ДЛЯ ПІДВИЩЕННЯ БЕЗПЕКИ РЕЛІЗУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кахутов Ю.Д.
Керівник: Алексієв В.О.
E-mail: yuriikakhutov@gmail.com
Харків, Харківський національний економічний університет імені Семена Кузнеця

На сьогодні поширеною практикою є те, що велика кількість програмних продуктів складається з десятків або навіть сотень зовнішніх бібліотек, образів контейнеризованих застосунків, SDK та зовнішніх сервісів [1]. З огляду на це важко керувати процесом доставки програмного забезпечення: будь-які проблеми з прямими або транзитивними залежностями, зміни в репозиторіях або зміни артефактів/ліцензій можуть зробити реліз неробочим або створювати ризики вже після введення системи в експлуатацію.

Передбачається, що однією з можливостей організації процесу збірки програмного забезпечення є так званий «Software Bill of Materials» (SBOM). SBOM містить інформацію про всі компоненти збірки та їх метадані, тому забезпечує можливість їх автоматичної перевірки та управління залежностями. Найпоширеніші формати SBOM – CycloneDX та SPDX [1, 2]. У випадку перевірки SBOM під час CI/CD-процесу збірки команди будуть повідомлені про критичні проблеми ще до введення версії в промислову експлуатацію.

У дослідженні пропонується розробка прототипу автоматичної системи управління SBOM під час збірки програмного забезпечення, який може виконуватися в якості вхідного контролю під час релізу застосунків. Прототип може зчитувати або створювати SBOM під час збірки застосунків та контейнеризованих застосунків, виконувати перевірку даних SBOM, нормалізувати Package URL – purl, CPE, контрольні суми та будувати граф залежностей із розрізненням прямих і транзитивних зв’язків.

Система порівнює компоненти SBOM з відкритими базами даних відомостей про уразливості (OSV і NVD) [3, 4] і оцінює критичні уразливості за шкалою CVSS [5] з урахуванням контексту залежності (наприклад, глибини в графі). Кожен окремий модуль здійснює окремий аналіз ліцензій: SPDX-ідентифікатори, встановлює невизначені значення і перевіряє відповідність політиці організації [6]. Нарешті, модуль release gate підсумовує ризики й приймає рішення щодо дозволу виходу версії, видання попередження або блокування виходу версії відповідно до встановлених ризикових порогових значень і політики ліцензування.

Результати прототип подає за варіантами: звіт для людини (HTML/PDF) і машинозчитані дані для автоматизації (JSON/SARIF) [7].

Література

[1] OWASP Foundation. CycloneDX Specification [Електронний ресурс]. – Режим доступу: https://cyclonedx.org/guides/

[2] The Linux Foundation. SPDX Specification [Електронний ресурс]. – Режим доступу: https://spdx.dev/specifications/

[3] Open Source Vulnerabilities (OSV) [Електронний ресурс]. – Режим доступу: https://osv.dev/

[4] NIST. National Vulnerability Database (NVD) [Електронний ресурс]. – Режим доступу: https://nvd.nist.gov/

[5] FIRST. Common Vulnerability Scoring System (CVSS) v3.1 [Електронний ресурс]. – Режим доступу: https://www.first.org/cvss/specification-document

[6] SPDX License List [Електронний ресурс]. – Режим доступу: https://spdx.org/licenses/

[7] OASIS. Static Analysis Results Interchange Format (SARIF) 2.1.0 [Електронний ресурс]. – Режим доступу: https://docs.oasis-open.org/sarif/sarif/v2.1.0/sarif-v2.1.0.html

ЗАСТОСУВАННЯ ВІЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ У ОЦІНЮВАННІ РИЗИКІВ

Кравченко В.Р.
Керівник: Солодовник Г.В.
E-mail: ganna.solodovnyk@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

Вступ. Процес управління сучасними соціально-економічними системами вимагає обробки та налізу великої кількості інформації, тому є неможливим без застосування інформаційних технологій. Засоби програмного забезпечення, що розповсюджуються за вільними ліцензіями є економічно обґрунтованим інструментарієм, який дозволяє легко модифікувати та масштабувати реалізовані моделі.

Постійне підвищення рівня динамічності об’єктів управління та зовнішнього середовища, яке обумовлює ефективність їх функціонування, висуває питання оцінювання ризиків на провідне місце. Тому наразі є актуальною задача врахування ризиків та невизначеності в управлінських процесах за використання засобів вільного програмного забезпечення.

Мета роботи полягає у розробці програми для визначення послідовності раціональних рішень щодо розвитку фірми з урахуванням можливості додаткового дослідження ринку. Об’єктом роботи є процес прийняття управлінських рішень щодо розвитку фірми в умовах невизначеності. Предметом роботи є методи та програмні засоби формування послідовності раціональних управлінських рішень.

Невизначеність інформації в процесі прийняття управлінських рішень пов’язана з віддаленістю у часі момента їх ухвалення від періоду реалізації цих рішень. Крім того на момент ухвалення кожного рішення слід врахувати можливий влив результатів його реалізації на наступні рішення. Таким чином, виникає послідовність взаємопов’язаних рішень, кожне з який приймається в умовах ризику та невизначеності. Математичним апаратом підтримки прийняття таких рішень є багатоетапні ігри з природою. Застосування методу багатоетапних ігор з природою передбачає: формулювання умов задачі прийняття рішення завдяки визначенню можливостей збору інформації; складання переліку подій, які з певною ймовірністю можуть відбутися протягом реалізації рішень; визначення порядку розташування цих подій у часі; оцінку ймовірностей здійснення кожної з конкретних подій. Для наочності розв’язання задачі наведені дані відображаються графічно у вигляді дерева рішень. Листям такого дерева приписується виграш (або програш як виграш з від’ємним знаком) першого гравця, тобто суб’єкта прийняття рішень, у випадку реалізації їм послідовності рішень та настання послідовності станів середовища, які складають певну гілку дерева рішень.

Дану роботу присвячено створенню програмної реалізації розв’язання задачі формування послідовності раціональних рішень в рамках теорії ігор.

Функціонування програми передбачає введення користувачем початкових даних; побудову дерева рішень; обчислення очікуваної грошової оцінки кожного рішення, які складають відповідні вузли дерева; формування послідовності раціональних рішень за порівнянням значень їх очікуваних грошових оцінок; вивід результатів у текстовому та графічному вигляді. Очікувана грошова оцінка розраховується як сума добутків виграшів від реалізації рішень за різних станів зовнішнього середовища на ймовірність настання даного стану.

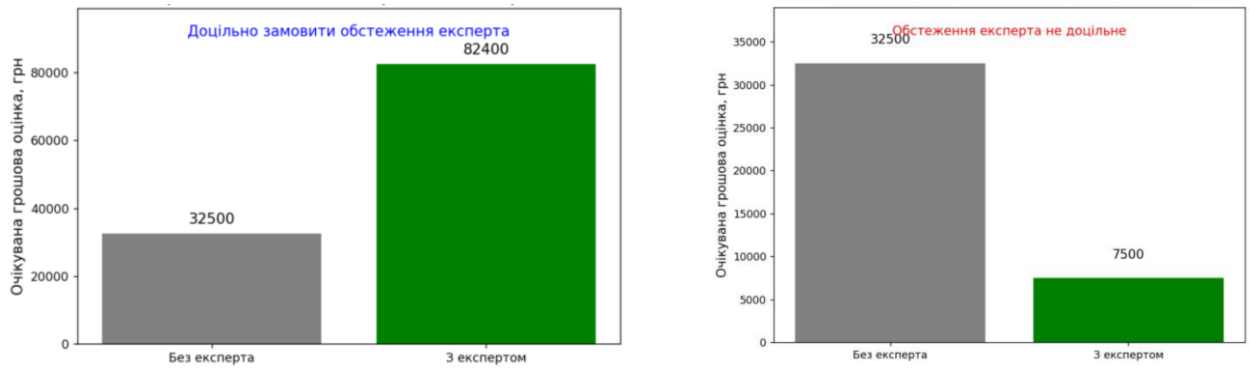
Початкові дані програми складають значення прибутків та збитків у випадку різних стратегій розвитку фірми за умов виникнення сприятливої та несприятливої ринкової обстановки, а також вартість додаткових обстежень ринку.

Обчислення очікуваних грошових оцінок стратегій розвитку фірми відбувається для гілки, яка передбачає додаткове обстежень ринку та гілки без обстеження. Додаткове обстеження дозволяє уточнити апріорні ймовірності сприятливої та несприятливої ринкової

ситуації, проте не надає гарантії настання того або іншого стану ринку. Точність результатів досліджень залежить від компетентності фірми, яка проводить дослідження. Прийняття рішення про доцільність обстеження ринку визначається порівнянням розрахованих значень очікуваної грошової оцінки відповідних рішень.

Проміжними результатами роботи програми є очікувані грошові оцінки рішень кожної стратегії розвитку. Кінцеві результати складають фінальні рекомендації щодо доцільності обстеження ринка та щодо найкращої стратегії розвитку. Завершальний етап роботи програми передбачає графічну візуалізацію порівняння очікуваних грошових оцінок з обстеженням ринку та без нього, що дозволяє наочно оцінити ефективність інвестицій у дослідження ринку та обрати найвигіднішу стратегію розвитку компанії.

На рисунку 1 наведено графіки, які були отримані за допомогою розробленої програми. Тестування роботи програми відбувалося для набору вхідних даних, за яких доцільним є додаткове обстеження ринку (рис. 1 а)), та для набору вхідних даних, за яких таке обстеження є недоцільним (рис. 1 б)).



а) *б)*

Рисунок 1 – Результати роботи програми

За інструментальні засоби розробки було обрано мову програмування Python. Програмну реалізацію створено у середовищі Visual Studio Code. Такий вибір інструментарію обумовлений тим, що застосування мови Python та середовища Visual Studio Code є доцільним у створенні відкритих науково обґрунтованих програмних рішень з можливостю подальшого масштабування, адже використання Python у поєднанні з Visual Studio Code дозволяє швидко створювати та модифікувати програмні прототипи.

Вибір мови програмування Python зумовлений її відкритістю та наявністю широкої системи відкритих бібліотек, що забезпечує ефективну реалізацію алгоритмів аналізу, моделювання та підтримки прийняття рішень. Python підтримує кросплатформність, що дозволяє використовувати розроблене програмне забезпечення в різних операційних системах без змін у коді.

Висновок. Практичне значення створеної програмної реалізації полягає у підвищенні наукової обґрунтованості управлінських рішень та скорочені часу на підготовку та оцінювання різних стратегій розвитку компанії за різних можливих станів зовнішнього середовища. Використання в ході розробки програми вільного програмного забезпечення робить її більш економічно доступною, що є важливим фактором для компаній малого та середнього секторів бізнесу. Подальший розвиток програми полягає у доробці користувацького інтерфейсу за рахунок створення інтуїтивно зрозумілих форм введення початкових даних.

СУЧАСНІ ПІДХОДИ ДО АВТЕНТИФІКАЦІЇ ТА АВТОРИЗАЦІЇ В REST API ЗА ДОПОМОГОЮ OAUTH 2.0 ТА JWT

Кунах.І.А.
Керівник: Коробейнікова.Т.І.
E-mail: ihor.kunakh.kb.2022@lpnu.ua
Львів, Національний Університет «Львівська політехніка»

Робота присвячена практичному використанню протоколу OAuth 2.0 у поєднанні з форматом токенів JWT для організації безпечної, зручної та масштабованої системи автентифікації й авторизації в сучасних веб- та мобільних додатках. Основний акцент зроблено на тому, як ці технології дозволяють відмовитися від передачі логінів і паролів третім сторонам, забезпечити делеговану авторизацію, підтримувати різні типи клієнтів і при цьому зберігати високу безпеку та простоту масштабування REST API.

Постановка задачі :Сьогодні практично всі сучасні веб-сервіси, мобільні програми та односторінкові додатки взаємодіють із сервером через REST API. Це робить питання безпечної автентифікації та чіткого розмежування прав доступу одним із найважливіших аспектів розробки. Традиційні підходи на основі сесійних cookie та зберігання стану на сервері стають незручними при роботі з мікросервісами, розподіленими системами, мобільними клієнтами та високонавантаженими додатками. Саме тому більшість сучасних проєктів переходять на токенну авторизацію, де OAuth 2.0 виступає стандартом делегованої авторизації, а JWT - зручним і широко поширеним форматом самих токенів. Проте неправильна реалізація цих механізмів призводить до серйозних вразливостей: витік токенів, підміна прав доступу, повторне використання перехоплених токенів, обхід захисту в публічних клієнтах тощо. Метою роботи є показати, як на практиці правильно застосовувати ці технології, щоб отримати всі їхні переваги й водночас уникнути типових помилок.

Практичний підхід :OAuth 2.0 дозволяє користувачеві надати певному додатку обмежений доступ до своїх даних (наприклад, профілю, фото, пошти) без того, щоб передавати свій пароль цьому додатку. У процесі авторизації користувач взаємодіє з сервером авторизації, а потім отримує токен доступу, який додаток надсилає до API для виконання запитів від імені користувача [3, 4]. У більшості сучасних реалізацій цей токен доступу оформлюється саме у форматі JWT. Такий підхід дуже зручний, тому що JWT — це самодостатній токен: він уже містить усю необхідну інформацію про користувача, його права (scopes), термін дії, видавача тощо. Завдяки цьому серверу ресурсів (тобто самому API) не потрібно щоразу звертатися до сервера авторизації для перевірки токена — достатньо перевірити підпис і актуальність claims [2].

Найчастіше для веб- та мобільних додатків використовується потік Authorization Code Flow з обов'язковим застосуванням PKCE (Proof Key for Code Exchange). Це захищає від перехоплення авторизаційного коду в публічних клієнтах, якими є браузери та мобільні програми [3, 8]. Для сервер-серверної взаємодії (наприклад, між мікросервісами) частіше застосовується Client Credentials Flow. А щоб користувачу не доводилося постійно вводити пароль, використовуються refresh-токени, які дозволяють отримувати нові access-токени без повторної авторизації. На практиці безпека значно залежить від кількох ключових налаштувань. Access-токени повинні мати короткий термін дії — зазвичай від 5 до 60 хвилин. Refresh-токени живуть довше, але їх необхідно зберігати максимально захищено: найкраще в HttpOnly-куках з атрибутами Secure та SameSite=Lax або Strict, щоб уникнути витоку через XSS-атаки. Важливо також використовувати тільки асиметричні алгоритми підпису (RS256, ES256), а не HS256 у розподілених системах, де секрет важко синхронізувати між серверами.

Кожен JWT, який приходить у запиті, повинен проходити повну перевірку: видавець (iss), аудиторія (aud), термін дії (exp), час початку дії (nbf), час видачі (iat). Якщо хоч один із цих параметрів не відповідає очікуваному — запит відхиляється без обговорень [2, 6].

Окремо варто подбати про можливість відкликання токенів (revocation) — наприклад, через чорний список або через інтроспекцію на сервері авторизації.

Практичне значення: Правильне використання OAuth 2.0 разом із JWT дає низку суттєвих переваг. По-перше, значно підвищується безпека порівняно з класичними сесіями чи передачею логінів/паролів. По-друге, система стає stateless — серверу не потрібно зберігати інформацію про активні сесії, що спрощує горизонтальне масштабування та роботу з мікросервісами. По-третє, з'являється можливість легко підключати зовнішніх провайдерів авторизації (Google, Facebook, корпоративний Keycloak, Auth0 тощо), що дуже зручно для B2C-продуктів. Для українських розробників це особливо актуально, оскільки більшість сучасних проєктів орієнтовані на міжнародний ринок і повинні відповідати високим стандартам безпеки та захисту персональних даних. Крім того, така архітектура добре поєднується з популярними фронтенд-фреймворками (React, Vue, Angular) та мобільними платформами [4, 5].

На сьогодні поєднання OAuth 2.0 та JWT є одним із найпоширеніших і найнадійніших способів побудови автентифікації та авторизації в REST API. Якщо правильно налаштувати потоки авторизації, обов'язково застосовувати PKCE для публічних клієнтів, використовувати короткий термін дії токенів, асиметричний підпис, захищене зберігання refresh-токенів і повну перевірку всіх claims — вдається створити систему, яка одночасно безпечна, зручна для користувачів і легка в масштабуванні. Саме такий підхід став де-факто промисловим стандартом для більшості сучасних веб- та мобільних застосунків.

Література

[1] OWASP Foundation. OWASP API Security Top 10 (2023) [Електронний ресурс]. – Режим доступу: https://owasp.org/www-project-api-security/.

[2] Jones M., Bradley J., Sakimura N. JSON Web Token (JWT) // RFC 7519. – IETF, 2015. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc7519.

[3] OAuth 2.0 та JWT: сучасна автентифікація в веб-додатках [Електронний ресурс] // Hostragons. – Режим доступу: https://www.hostragons.com/uk/блог/сучасна-автентифікація-з-oauth-2-0-i-jwt/.

[4] Що таке OAuth 2.0 і як працює авторизація через токени [Електронний ресурс] // Wezom. – Режим доступу: https://wezom.com.ua/ua/blog/scho-take-oauth-20-i-yak-pratsyuje-avtorizatsiya-cherez-tokeni.

[5] OAuth: що це та як працює [Електронний ресурс] // Foxminded. – Режим доступу: https://foxminded.ua/oauth-shcho-tse/.

[6] Практичні рекомендації з використання OAuth 2.0 та JWT в REST API [Електронний ресурс] // ITSSI Journal. – Режим доступу: https://www.itssi-journal.com/index.php/ittsi/article/view/509/462.

АКТУАЛЬНІ ПРОБЛЕМИ ЗБЕРЕЖЕННЯ ТА ВІДНОВЛЕННЯ ДАНИХ З ВИКОРИСТАННЯМ КІБЕРСХОВИЩ

Лубенець С.В., Шелестова А.М., Губін В.О.
E-mail: s.lubenec@karazin.ua, anna.shelestova@karazin.ua, vadim.gubin@nure.ua
Харків, Харківський національний університет імені В.Н. Каразіна
Харків, Харківський національний університет радіоелектроніки

Цифровізація економіки й управління вимагає від компаній та організацій значних зусиль та ресурсів задля забезпечення ефективної цифрової інформаційної безпеки [1]. Резервування даних та конфіденційної інформації в управлінських інформаційних системах – один з ключових інструментів кіберзахисту. Надійне збереження даних є основою будь-якого їх відновлення. Кіберзлочинці намагаються знищити дані, але кіберсховища можуть допомогти у їх збереженні та відновленні. Практично щодня організації виконують відновлення даних зі своїх систем резервного копіювання. Більшість із них є звичайними відновленнями, можливо, через просту людську помилку або інші незначні події. Інші

відновлення є складнішими, в результаті збоїв обладнання (серверів, сховищ або збоїв у роботі мережі), помилок програмного забезпечення тощо. Відновлення може містити як один файл, так і цілі системи з сотнями терабайтів.

Дослідження IDC [2] показало, що лише менше третини організацій (31%) здатні повністю відновитися після кібератаки без втрати даних та без сплати викупу. Більше того, приблизно половина атак передбачає спробу видалити, пошкодити або іншим чином скомпрометувати системи резервного копіювання. У разі успіху цих атак на резервні копії, зловмисник практично гарантовано отримує викуп, коли у жертв немає інших варіантів. На жаль, ця тактика успішна приблизно в половині випадків. Це означає, що кожна четверта жертва атаки не має можливості відновлення за допомогою власних систем. При цьому, за умови правильного налаштування, зовнішні кіберсховища є найкращим способом забезпечити збереження даних як для аварійного відновлення, так і для кібервідновлення.

Кіберсховища стали цінним інструментом для забезпечення збереження даних у разі будь-якої атаки чи сценарію втрати даних. Завдяки типовому розгортанню поза межами офісу, вони захищають дані від стихійних лих, пожеж, повеней або інших подій, які унеможливлюють доступ до центру обробки даних. Вони також дають будь-якій організації значну перевагу в боротьбі з викупниками.

За шкалою складності аварійне відновлення (disaster recovery DR) та кібервідновлення (cyber-recovery CR) часто є найскладнішими сценаріями відновлення, з якими стикаються IT-організації. Обсяг відновлення може бути значним, наприклад, залучати більше одного центру обробки даних, локально або в хмарі. Усі такі ситуації відновлення відбуваються в умовах обмеженого часу та адміністративного тиску. Наявність доступу до відомих, чистих резервних копій може зменшити це навантаження та дозволити організаціям швидко відновитися.

Три основні принципи кібервідновлення наступні: забезпечення збереження даних, забезпечення цілісності даних, забезпечення швидкого відновлення. Кіберсховища вирішують перші два та сприяють третьому. Завдяки кіберсховищу операції з відновлення можуть розпочатися швидше, коли є гарантовано безпечна, чиста та точна копія даних для відновлення.

На жаль, занадто багато IT-команд не знайомі з перевагами кіберсховищ і тому не використовують їх. Фактично, деякі не усвідомлюють необхідності зберігання даних поза межами офісу. Тому важливим є дослідження й аналіз переваг розгортання та використання кіберсховищ.

Згідно визначення, кіберсховище – це сховище для резервних копій даних, яке включає їх шифрування та незмінність, що робить знищення або компрометацію даних практично неможливим.

Організації можуть впроваджувати кіберсховища за допомогою апаратно-специфічних систем, таких як спеціально розроблені пристрої локального резервного копіювання. Як альтернатива, їх можна впроваджувати в хмарі, використовуючи ресурси гіпермасштабування. До основних особливостей впровадження кіберсховищ можна віднести наступне:

Впровадження локальних систем може вимагати складних конфігурацій. Якщо їх впроваджувати без реплікації між центрами обробки даних, це може не відповідати вимогам до віддаленого сховища. Однак локальне впровадження може бути необхідним, коли дані неможливо перемістити за межі офісу.

Дослідження IDC показує, що понад 90% організацій використовують гібридну хмарну (з локальної до хмарної) архітектуру для захисту даних. Таким чином, впровадження хмарного кіберсховища зазвичай є відносно безболісним розширенням поточних можливостей.

Кіберсховище існує в середовищі, повністю відокремленому від виробничого середовища та самої організації, додаючи ще один рівень захисту.

Багато організацій знайомі зі стратегією резервного копіювання 3-2-1, яка передбачає три копії даних на двох різних типах носіїв з однією офлайн-копією. Це була найкраща практична стратегія протягом понад двох десятиліть. Однак, з поширенням використання хмарних технологій та зростанням кіберзагроз, новіша та актуальніша стратегія 3-2-1-1-0 покращує стару стратегію.

Кіберсховища враховують нові компоненти стратегії резервного копіювання: друга «1» – це віддалене сховище для забезпечення збереження даних, причому «0» означає незмінність даних у цьому сховищі. У поєднанні ці два додаткових елементи гарантують, що резервні дані в кіберсховищі можна відновити, і що вони є автентичними.

Ключові функції кіберсховища включають наступне:

Незмінність. Вона гарантує, що дані у сховищі не можуть бути змінені або видалені. Це важливо для забезпечення цілісності та збереження даних.

Шифрування. Шифрування даних гарантує, що з них не буде користі, навіть якщо їх викрадено. Це особливо важливо, оскільки дослідження IDC виявило збільшення випадків витоку даних шляхом застосування програм-вимагачів [3]. Шифрування також запобігає загрозам із внутрішніх джерел.

Двофакторна/двоособова автентифікація. Це допомагає зменшити ризик використання індивідуальних облікових даних для отримання доступу до системи. Двоособова автентифікація вимагає, щоб дві особи змовилися для незаконного доступу до систем, що зменшує частоту інсайдерських атак.

Віддалене розташування. Якщо це не передбачено вимогами керування даними, організації повинні використовувати віддалене розташування для сховища. Віддалене сховище особливо важливе для сценаріїв аварійного відновлення.

Підтримка вимог до рівня обслуговування. Кіберсховища, завдяки цілісності та доступності даних, можуть сприяти підтримці вимог до рівня обслуговування цільової точки відновлення (recovery point objective RPO) та цільового часу відновлення (recovery time objective RTO) організації.

Управління даними та суверенітет. Правильно налаштовані сховища даних підтримують та дотримуються вимог до управління даними та суверенітету.

Прикладом сучасного ефективного кіберсховища є Veeam Data Cloud Vault [4] – це повністю керований, безпечний хмарний ресурс сховища компанії. Він включає довговічне сховище, яке використовує об'єктне сховище й забезпечує синхронну реплікацію для резервного копіювання традиційних та сучасних робочих навантажень як локально, так і в хмарі. Дане кіберсховище базується на архітектурі нульової довіри (Zero Trust), щоб обмежити можливості зловмисників отримувати доступ до конфіденційних систем. За замовчуванням воно є незмінний, щоб запобігти будь-яким спробам зміни або видалення резервних копій, як зловмисним, так і випадковим. Воно також шифрує дані для захисту конфіденційної інформації та запобігання витоку даних.

За своєю природою кіберсховище Veeam Vault також повністю відповідає стратегії 3-2-1-1-0, оскільки резервні копії даних знаходяться поза межами офісу, а його архітектура з нульовою довірою та незмінне зберігання дозволяють перевіряти сховища даних без помилок.

Однак IT-команди повинні усвідомлювати те, що якщо дані копіюються в хмару, це не означає, що вони ізольовані. Організації повинні забезпечити, щоб їхні процеси підтримували фізичне розділення шляху передачі даних та керування даними за допомогою окремих облікових даних. Також доцільно захистити копії даних за допомогою двофакторної автентифікації та/або автентифікації двох осіб. Організаціям також слід розглянути можливість реплікації хмарних репозиторіїв в інші зони або регіони у разі збою в хмарі. Ці репліки повинні бути налаштовані відповідно до вимог суверенітету та управління даними.

Таким чином, збереження даних є основою будь-якого відновлення даних. Кіберсховища стали важливим захистом від атак на дані, незалежно від того, чи спрямована ця атака на самі дані, чи на витік даних. Хоча організації повинні продовжувати

дотримуватися звичайних найкращих практик захисту даних, кіберсховища надають додатковий рівень захисту, який є останнім, найкращим шансом на збереження та відновлення даних. Організації без кіберсховища наражають себе на значний бізнес-ризик. Хмарні, повністю керовані кіберсховища можуть зменшити цей ризик за допомогою хмарної економіки.

Література

[1] Сергій Лубенець, Ігор Харченко та Євген Павленко. (2023). Актуальні проблеми міжнародної інформаційної безпеки. Вісник Харківського національного університету імені В.Н. Каразіна. Серія: Міжнародні відносини. Економіка. Країнознавство. Туризм, (17), 42-48 [Електронний ресурс]. – Режим доступу до ресурсу: https://doi.org/10.26565/2310-9513-2023-17-04

[2] IDC. Get the latest market data and insights [Electronic resource]. – Resource access mode: https://www.idc.com/

[3] Лубенець С.В., Шелестова А.М., Павленко Є.П. (2025). Актуальні задачі міжнародної цифрової інформаційної безпеки у цифровізації світової економіки. Актуальні проблеми світового господарства і міжнародних економічних відносин: матеріали XX всеукраїнської науково-практичної конференції, 28 лютого 2025 р. Харків: ХНУ ім. В.Н. Каразіна, 32-35 [Електронний ресурс]. – Режим доступу до ресурсу: https://ekhnuir.karazin.ua/handle/123456789/20986

[4] Veeam. Veeam data cloud vault [Electronic resource]. – Resource access mode: https://www.veeam.com/products/veeam-data-cloud/cloud-storage-vault.html

МЕТОДОЛОГІЧНІ ПІДХОДИ ДО УПРАВЛІННЯ РИЗИКАМИ В КІБЕРБЕЗПЕЦІ

Любименко О.М., Штепа О.А.
E-mail: e.n.lyubimenko@gmail.com
Дрогобич, Донецький національний технічний університет

Кіберінциденти сьогодні є не лише “технічною проблемою”, а й фактором операційної стійкості, фінансових втрат, правової відповідальності та репутаційних наслідків. Через це організації переходять до управління кібербезпекою як портфелем ризиків: визначають прийнятний рівень ризику, порівнюють варіанти оброблення ризиків та інвестують у заходи захисту на підставі очікуваного ефекту.

Водночас кіберризики мають специфіку: високу динаміку загроз, залежність від ланцюгів постачання та хмарних сервісів, складність вимірювання ймовірностей, “каскадні” впливи на бізнес-процеси. Тому методологія управління ризиками в кібербезпеці має поєднувати: управлінський контур, дисциплінований процес оцінювання ризику, практику підбору й перевірки контролів, безперервний моніторинг та перегляд.

Мета статті — систематизувати методологічні підходи до управління кіберризиками та запропонувати практичну інтегровану схему їх застосування в організації.

Найпоширеніші “опорні” джерела методології можна умовно поділити на три групи:

Міжнародні стандарти менеджменту ризиків та інформаційної безпеки: ISO/IEC 27001 задає вимоги до системи менеджменту інформаційної безпеки (ISMS); ISO/IEC 27005 надає настанови щодо управління ризиками інформаційної безпеки як підтримки ISMS; ISO 31000 формулює універсальні принципи й процес управління ризиками на рівні організації.

Національні/галузеві рамки та методичні документи: NIST пропонує низку публікацій, що задають процес управління ризиками (RMF) і методику оцінювання ризиків: NIST SP 800-37 Rev.2 (RMF) та NIST SP 800-30 Rev.1; окремо важливим “містком” між технікою і управлінням став NIST Cybersecurity Framework 2.0, який фіксує функції кібербезпеки та робить акцент на управлінні (зокрема через виділення функції Govern у CSF 2.0).

Регуляторні вимоги та наглядова практика: у ЄС директива NIS2 закріплює посилений підхід до кіберризик-менеджменту та встановлює ключові дедлайни

імплементації в національне законодавство; для фінансового сектору ЄС регламент DORA визначає рамку цифрової операційної стійкості та застосовується з 17 січня 2025 року.

Запропоновано практичну модель для управління ризиками- гібридну: рамка + метод оцінювання + механізм реалізації контролів + моніторинг. Наведено один із робочих варіантів інтеграції:

– governance і контекст: визначити цілі, ризик-апетит, власників ризику, вимоги комплаєнсу (CSF 2.0 / ISO 31000);

– інвентарактивів і процесів: що захищаємо і чому це критично;

– побудова сценаріїв: джерела загроз → вектори → події → наслідки (OCTAVE/EBIOS);

– оцінювання ризику: якісне (шкали) або кількісне (FAIR), або комбіноване;

– вибір реакції на ризик: уникнення / зменшення / передача / прийняття (ISO 31000 логіка);

– добір та реалізація контролів (RMF-логіка життєвого циклу);

– оцінка ефективності: тестування, аудит, технічні перевірки;

– фіксація залишкового ризику та його прийняття (формалізація рішення);

– безперервний моніторинг і перегляд: метрики, інциденти, зміни в архітектурі/постачальниках, переоцінювання;

– комунікація ризиків для керівництва: топ-ризики, тренди, план оброблення, статус контролів.

Управління ризиками в кібербезпеці еволюціонувало від “контрольних списків” до системного, керованого ризиком підходу, який поєднує вимоги стандартів, очікування регуляторів і практику безперервного моніторингу. У статті узагальнено ключові методологічні підходи до кіберризик-менеджменту: стандарто-орієнтований (ISO/IEC 27001/27005, ISO 31000), рамковий процесний (NIST RMF, NIST SP 800-30), управлінсько-комунікаційний (NIST CSF 2.0), сценарно-активний (OCTAVE, EBIOS RM), кількісний (FAIR) та регуляторно-орієнтований (зокрема NIS2, DORA). Показано їх сильні сторони, обмеження та умови доцільного застосування. Запропоновано інтегровану логіку побудови програми управління кіберризиками, що забезпечує узгодження цілей бізнесу, технічних контролів та прийняття залишкового ризику на рівні керівництва.

Література

[1] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. ISO/IEC 27005:2022 Information security, cybersecurity and privacy protection — Guidance on managing information security risks [Electronic resource]. – Resource access mode: https://www.iso.org/standard/80585.html.

[2] JOINT TASK FORCE. Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy. NIST Special Publication 800-37 Revision 2. Gaithersburg, MD: National Institute of Standards and Technology, 2018. [Electronic resource]. – Resource access mode: https://csrc.nist.gov/pubs/sp/800/37/r2/final

ПІДХОДИ ЩОДО ПРОВЕДЕННЯ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ

Марченко Я.В., Якимчук Є.А.
E-mail: yaroslav.marchenko@npp.kai.edu.ua, yevhenii.iakymchuk@npp.kai.edu.ua
Київ, Державний університет «Київський авіаційний інститут»

А сьогодні питання забезпечення інформаційної безпеки є основним пріоритетом абсолютно для всіх організацій, малих чи великих бізнесів. Одним із ефективних методів первинної перевірки запроваджених методів інформаційної безпеки є проведення тесту на проникнення. В процесі тестування, шляхом реалізації атак на об’єкт дослідження, спеціаліст документує отримані результати. Наприкінці тестування, отримується звіт, по якому можна дослідити стійкість реалізований методів до певних сучасних атак, а також зрозуміти слабкі місця в об’єкті дослідження.

Кожний тест має певний набір пунктів за яким буде відбуватися дослідження об'єкта. Методи дослідження можуть відрізнятися в залежності від вхідних даних. Але навіть в такому випадку можна виділити основні пункти, які будуть не змінними.

Сам процес тестування, відноситься до етичного хакінгу, тобто всі дії проводяться виключно з метою знаходження вразливостей та усуненню небезпеки, не дивлячись на те, що методи при тестуванні не будуть відрізнятися від методів зловмисника [1]. Для збереження етичності потрібно дотримуватися наступних вимог:

– весь процес тестування відбувається виключно з дозволу власника об'єкта;

– тестування повинно відбуватися чітко за законодавчим регламентом та контрактом;

– конструктивність проведення тестування.

Після отримання дозволу на проведення тестування, узгодження часу та обсягу роботи, спеціаліст починає свою роботу з першого етапу, а саме отримання інформації про об'єкт дослідження. Отримані дані дозволять побудувати вектор подальшої роботи.

Отримання інформації поділяють на два типи. Активний збір даних потребує взаємодію з цілю тестування. При пасивному зборі інформації використовується відкриті джерела на яких зберігаються загальнодоступні записи [2].

Інформація, яку можна отримати після проведення даного етапу: діапазон IP-адрес, доменні імена, відкриті порти, інформацію про співробітників, корпоративні пошти тощо.

Додатково для отримання даних, спеціалісти використовують соціальну інженерію. Це дозволяє перевірити стійкість персоналу до маніпуляцій та дотримання встановленої політики безпеки, такий підхід в подальшому допоможе уникнути витоку конфіденційної інформації [3].

Наступним етапом дослідження є сканування об'єкта. Воно допомагає визначити відкриті порти та вказати, які сервіси на них розгорнуті. До прикладу, Якщо при скануванні отримується інформація про відкритий порт 22 з мережевим протоколом, який забезпечує безпечний віддалений доступ [4]. Маючи певний діапазон корпоративних пошт з попереднього етапу, можна реалізувати атаку грубої сили для підбору паролю до облікового запису користувача, використавши пошти, як логін. Даний приклад показує щільний зв'язок між етапами тестування, тобто вихідні дані попереднього етапу, будуть вхідними даними для наступного. Додатково виконується сканування відомих вразливостей CVE [5].

Завершальним етапом є спроба отримання доступу до об'єкту. На цьому проміжку тестування, вся теоретична інформація отримана з минулих етапів використовується на практиці, тобто перетворити знайдену потенційну вразливість на робочий експлойт.

Отже, тестування на проникнення є одним із основний елементів перевірки та підтримки інформаційної безпеки. Цей систематизований підхід дозволяє перевірити конфіденційність, цілісність та доступність, шляхом реалізації сучасних атак в режимі реального часу, для перевірки стійкості системи. Залучення соціальної інженерії допомагає перевірити знання та стійкість до впливу співробітників, що в деяких випадках може вказувати на необхідність проведення тренінгів зі своєю командою. Зручний формат подання результатів тестів залишається у власності компанії, що дозволяє при повторних тестуваннях дозволяє дозволяє порівнювати звіти та спостерігати покрашення чи погіршення інформаційної безпеки в організаціях.

Література

[1] What is Ethical Hacking [Електронний ресурс] - EC-Council Cybersecurity Exchange: https://www.eccouncil.org/cybersecurity-exchange/ethical-hacking/what-is-ethical-hacking/

[2] Understanding the Five Phases of the Penetration Testing Process [Електронний ресурс] - EC-Council Cybersecurity Exchange: https://www.eccouncil.org/cybersecurity-exchange/penetration-testing/penetration-testing-phases/

[3] Psychological Exploitation of Social Engineering Attacks [Електронний ресурс] - Cyber Risk GmbH: https://www.cyber-risk-gmbh.com/Psychological_Exploitation_of_Social_Engineering_Attacks.html

[4] What is Common Vulnerabilities and Exposures (CVE)? [Електронний ресурс] – IBM: https://www.ibm.com/think/topics/cve

[5] SSH Protocol – Secure Remote Login and File Transfer [Електронний ресурс] - SSH Communications: https://www.ssh.com/academy/ssh/protocol

КРИМІНАЛІСТИЧНИЙ АНАЛІЗ ВИТОКІВ ІНФОРМАЦІЇ: ІНТЕГРАЦІЯ OSINT У ПРОЦЕСИ ВИЯВЛЕННЯ ТА РЕКОНСТРУКЦІЇ КАНАЛІВ ВИТОКУ

Приходько Т.Ю.
E-mail: tetiana.prykhodko@npp.kai.edu.ua
Київ, Національний університет «Київський авіаційний інститут»

Актуальність. Сучасні кіберінциденти, пов’язані з витоками інформації, характеризуються використанням розподіленої мережевої інфраструктури та багаторівневих каналів передачі даних. Аналіз лише внутрішніх журналів подій і локальних артефактів системи не забезпечує повної реконструкції інциденту. Значна частина цифрових слідів існує у зовнішньому інформаційному середовищі: відкритих базах даних, мережевих сервісах, публічних ресурсах і спеціалізованих онлайн-спільнотах.

OSINT-системи, як структуровані комплекси збору, обробки та аналізу інформації з відкритих джерел, дозволяють систематизувати ці дані та інтегрувати їх у процес цифрової криміналістики. Архітектура таких систем передбачає наявність інформаційного фонду, інструментів автоматизованого збору даних, аналітичних модулів і механізмів взаємодії з іншими інформаційними системами. Включення OSINT у стандартизовані процеси кібербезпеки розширює можливості ідентифікації джерел витоку та сприяє повнішій реконструкції каналів несанкціонованої передачі інформації [1-2].

Метою роботи є дослідження методики криміналістичного аналізу витоків інформації з використанням OSINT як інтегрованого інструменту збору та кореляції цифрових доказів. Робота спрямована на формування підходу, у якому OSINT-системи використовуються як складова єдиного аналітичного середовища разом із журналами подій, мережевими даними та результатами технічної експертизи.

Комплексний криміналістичний аналіз витоків інформації передбачає поетапну інтеграцію OSINT у процес розслідування.

Першим етапом є превентивна ідентифікація вразливостей шляхом систематичного моніторингу відкритих джерел. Аналіз публічних витоків облікових даних, спеціалізованих форумів, репозиторіїв і відкритих баз даних дозволяє виявляти потенційні слабкі місця інфраструктури та ознаки підготовки атак. Такий моніторинг формує інформаційну основу для оцінки ризиків і підвищує готовність до реагування на інциденти.

Другим етапом є інтеграція зовнішніх даних у процес розслідування конкретного інциденту. Внутрішні цифрові артефакти — журнали подій, мережеві записи, результати аналізу пам’яті — корелюються з OSINT-даними для ідентифікації підозрілих доменів, IP-адрес та інфраструктурних зв’язків. Це дозволяє встановити приналежність ресурсів, виявити можливі інструменти зловмисника та уточнити контекст атаки.

Третім етапом є реконструкція каналу витоку інформації. За допомогою OSINT здійснюється відстеження подальшого руху даних за межами локальної мережі, що дає змогу об’єднати розрізнені технічні докази у єдину логічну модель. Ідентифікація кінцевих точок отримання інформації та шляхів несанкціонованого виведення інформації сприяє повнішому розумінню механізму інциденту та підвищує точність атрибуції.

Важливою складовою ідентифікації джерела є робота з інформаційним фондом — базою даних, що формується з відкритих джерел. Процес аналізу витоку має включати очищення даних від спаму та реклами для виявлення реальних закономірностей у діях зловмисників. Сучасна реконструкція каналу витоку неможлива без інтеграції з системами типу SIEM, де технічні алерти збагачуються даними OSINT про репутацію IP-адрес та інструментаріями хакерських груп. Додатково, використання генеративного штучного

інтелекту дозволяє автоматизувати аналіз неструктурованих звітів, значно скорочуючи час на атрибуцію атаки. [1]

Висновок. OSINT є важливим компонентом сучасної цифрової криміналістики та систем кібербезпеки. Його інтеграція в процеси розслідування витоків інформації забезпечує більш повну картину інциденту та підвищує точність ідентифікації його джерела. [3] Використання структурованих OSINT-систем, здатних взаємодіяти з іншими аналітичними платформами, сприяє переходу до комплексної моделі кібербезпеки, у якій внутрішній технічний аналіз поєднується з системним дослідженням зовнішнього інформаційного середовища. Такий підхід підвищує ефективність виявлення та розслідування витоків інформації в умовах зростаючої складності цифрових систем.

Література

[1] Д.В. Ланде OSINT у кібербезпеці : навч. пос. - Київ: ТОВ "Інжиніринг",2024 – 522 с.

[2] D. Lande, O. Puchkov, I. Subach, M. Boliukh, D. Nahornyi OSINT investigation to detect and prevent cyber attacks and cyber security incidents // Information Technology and Security, 2021. Vol 9 (2). – pp. 209-218.

[3] Yong-WoonHwang,Im-Yeong Lee, Hwankuk Kim, Hyejung Lee, and Donghyun Kim. Current Status and Security Trend of OSINT. Wireless Communications and Mobile Computing, vol. 2022, Article ID 1290129, 14 pages, 2022

ПЕРСПЕКТИВИ ЗАСТОСУВАННЯ МОДЕЛІ MAMBA ДЛЯ ВИЯВЛЕННЯ АНОМАЛІЙ МЕРЕЖЕВОГО ТРАФІКУ

Рихва В.
Керівник: Солодовник Г.В.
E-mail: ganna.solodovnyk@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

Зростання обсягів мережевого трафіку та ускладнення методів кібератак створюють серйозні виклики для сучасних систем виявлення вторгнень (IDS). За даними міжнародних звітів, кількість таких інцидентів щорічно зростає, а зловмисники дедалі частіше використовують складні багатоетапні атаки, що важко виявляються традиційними методами. Сигнатурний аналіз, що лежить в основі таких систем як Snort та Suricata, ефективний лише проти відомих загроз, тоді як нові типи атак залишаються невидимими для цих систем [1]. Це зумовлює необхідність дослідження і розробки нових методів виявлення аномалій.

Останні дослідження демонструють активне застосування архітектур глибокого навчання для задачі виявлення мережевих аномалій. Згорткові нейронні мережі (CNN) ефективно виявляють локальні шаблони у мережевому трафіку, рекурентні мережі (LSTM) моделюють часові залежності між пакетами, а Transformer-архітектури використовують механізм уваги для глобального аналізу послідовностей [2]. Проте кожна з цих архітектур має суттєві обмеження. CNN не здатні моделювати довгострокові залежності між віддаленими подіями у потоці трафіку. LSTM мають обмежену пропускну здатність через послідовний характер обчислень. Transformer-моделі мають квадратичну обчислювальну складність $O(n^2)$ відносно довжини послідовності, що робить їх непрактичними для обробки великих обсягів трафіку в реальному часі.

Архітектура Mamba, запропонована Gu та Dao [3], представляє принципово новий підхід до моделювання послідовностей на основі *селективних моделей простору станів (Selective State Space Models, SSM)*. На відміну від класичних SSM, які застосовують фіксовані параметри до всіх елементів послідовності, Mamba робить параметри функціями вхідних даних, що дозволяє моделі селективно зберігати або відкидати інформацію залежно від контексту. Ключовою перевагою Mamba є лінійна обчислювальна складність $O(n)$ відносно довжини послідовності – на порядок ефективніше за квадратичну складність Transformer. При цьому Mamba демонструє швидкість інференсу у 5 разів вищу за

Transformer аналогічного розміру та здатність ефективно обробляти послідовності довжиною до мільйона елементів [3].

Ці характеристики роблять Mamba особливо перспективною архітектурою для задачі виявлення аномалій мережевого трафіку з кількох причин. По-перше, мережевий трафік за своєю природою є послідовними даними з довгостроковими залежностями – підготовка до атаки може включати розвідувальні пакети за хвилини або години до основного вторгнення, і здатність моделювати такі зв'язки є критичною. По-друге, лінійна складність дозволяє обробляти потоки трафіку в реальному часі навіть на обмежених обчислювальних ресурсах, що є важливим для розгортання на периферійних пристроях IoT-мереж. По-третє, селективний механізм Mamba природно відповідає задачі виявлення аномалій: модель навчається фокусувати увагу на нетипових патернах трафіку, ігноруючи фоновий шум нормальної мережевої активності.

Перші дослідження підтверджують потенціал Mamba для кібербезпеки. Zhang та ін. [4] запропонували модель Mamba-ECANet для виявлення вторгнень, яка поєднує базову екстракцію ознак через Mamba з механізмом уваги ECANet для селекції найбільш інформативних характеристик. Експериментальна оцінка на датасеті CIC-IDS2017 продемонструвала покращення точності виявлення на 5% порівняно з традиційними методами. Alrubei та ін. [5] розробили гібридну архітектуру IDS-GraphMamba для виявлення вторгнень у мережах Інтернету медичних речей (IoMT), що поєднує графові нейронні мережі з блоками Mamba та ланцюгами Маркова, досягнувши точності 99.70% на спеціалізованому датасеті. Ці результати свідчать, що архітектура Mamba здатна ефективно обробляти різноманітні типи мережевих даних – від класичного потокового трафіку до складних графових структур IoT-мереж.

У дослідженні [6] мною було проведено порівняльний аналіз гібридних моделей, які суттєво перевершують базові: гібридна модель на основі CNN + Mamba була вперше застосована для аналізу аномалій мережевого трафіку і продемонструвала найвищу точність і найкращу ефективність на датасеті UNSW-NB15. Подальше дослідження буде спрямоване на поглиблене дослідження Mamba на інших датасетах.

Література

[1] Bace R., Mell P. Intrusion Detection Systems. NIST Special Publication 800-31. National Institute of Standards and Technology, 2001. 51 p.

[2] Kumar A., Pandey D. Enhancing intrusion detection with ML and deep learning: A survey of CICIDS 2017 and CSE-CIC-IDS2018 datasets. AIP Conference Proceedings. 2024. Vol. 3168, No. 1. P. 020003. DOI: 10.1063/5.0229391.

[3] Gu A., Dao T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. arXiv preprint arXiv:2312.00752. 2024.

[4] Zhang H., Zhu D., Gan Y., Xiong S. End-to-End Learning-Based Study on the Mamba-ECANet Model for Data Security Intrusion Detection. Journal of Information, Technology and Policy. 2024. Vol. 2, No. 1. P. 1–17. DOI: 10.62836/jitp.v1i1.219.

[5] Alrubei M. et al. IDS-GraphMamba: A Markov-enhanced graph Mamba framework for real-time intrusion detection in IoMT edge networks. Computer Networks. 2025. Vol. 265. P. 111289. DOI: 10.1016/j.comnet.2025.111289.

[6] Рихва В., Солодовник Г. Аналіз архітектур глибокого навчання для виявлення аномалій мережевого трафіку. Наука і техніка сьогодні. 2025. № 10(51). С. 1912–1922. DOI: 10.52058/2786-6025-2025-10(51)-1912-1922.

ЦИФРОВІ ТЕХНОЛОГІЇ В УМОВАХ СУЧАСНИХ ЗАГРОЗ

Рудешко І., Качура О.
E-mail: rudeshko_iryna@nuczu.edu.ua
Черкаси, Національний університет цивільного захисту України

Стрімка цифровізація всіх сфер суспільного життя, включаючи промисловість, енергетику, транспорт, державне управління та об'єкти критичної інфраструктури, зумовлює зростання ролі кібербезпеки як ключового елементу національної та міжнародної безпеки. Використання сучасних цифрових технологій (хмарних сервісів, штучного інтелекту, тощо) суттєво підвищує ефективність управління системами, однак водночас розширює поверхню кібератак. В умовах гібридних та воєнних загроз кіберпростір перетворюється на окремий театр протиборства, де атаки можуть призводити до масштабних техногенних, економічних і соціальних наслідків. Саме тому стандартизація у сфері кібербезпеки набуває критичного значення, оскільки дозволяє уніфікувати підходи до управління ризиками, захисту інформації та забезпечення стійкості цифрових систем [1–6].

Кібербезпека розглядається як сукупність організаційних, технічних та програмних заходів, спрямованих на захист інформаційних ресурсів, мереж та автоматизованих систем від несанкціонованого доступу, порушення цілісності, конфіденційності та доступності даних. Особливу увагу приділяють захисту критичної інформаційної інфраструктури, відмова або компрометація якої може спричинити каскадні наслідки для держави та суспільства. Сучасні кібератаки характеризуються високим рівнем складності, використанням шкідливого програмного забезпечення нового покоління, соціальної інженерії та цільових атак (APT). Це потребує переходу від реактивних до проактивних моделей кіберзахисту, заснованих на аналізі ризиків і постійному моніторингу загроз [3, 6].

Міжнародні та національні стандарти є основою формування єдиних вимог до систем управління інформаційною безпекою [1, 2, 4, 5]. Найбільш поширеними є стандарти серії ISO/IEC 27000, які визначають вимоги до політик безпеки, управління ризиками, контролю доступу та реагування на інциденти.

Важливе значення мають також стандарти NIST, рекомендації ENISA та нормативні документи Європейського Союзу, зокрема Директива NIS2. В Україні питання кібербезпеки регламентуються законами та підзаконними актами, що гармонізуються з міжнародними підходами.

Стандартизація дозволяє [1–5]:

– забезпечити сумісність і взаємодію цифрових систем;

– підвищити рівень довіри до інформаційних технологій;

– створити основу для аудиту та сертифікації систем;

– знизити ризики кібератак за рахунок уніфікованих процедур захисту.

Впровадження цифрових технологій, зокрема хмарних обчислень, призводить до децентралізації обробки даних і збільшення кількості точок доступу. Штучний інтелект використовується як для підвищення рівня кіберзахисту, так і для створення нових інструментів атак. У цьому контексті особливої актуальності набуває концепція «безпеки за проєктом» (security by design), яка передбачає інтеграцію вимог кібербезпеки на всіх етапах життєвого циклу цифрових систем. Стандарти та нормативні документи виступають інструментом формалізації таких вимог [1, 2, 6, 8].

Висновки. Кібербезпека, стандартизація та цифрові технології є взаємопов'язаними елементами сучасної системи безпеки. Ефективний захист цифрових систем неможливий без впровадження міжнародно визнаних стандартів та адаптації їх до національних умов [1–5, 7, 8]. Подальший розвиток цифрових технологій потребує випереджального вдосконалення нормативної бази та підготовки фахівців, здатних працювати в умовах постійно зростаючих кіберзагроз.

Література

[1] ISO/IEC 27001:2022. Information security, cybersecurity and privacy protection — Information security management systems — Requirements. — Geneva : ISO, 2022.

[2] ISO/IEC 27002:2022. Information security controls. — Geneva : ISO, 2022.

[3] NIST Cybersecurity Framework 2.0. — Gaithersburg : National Institute of Standards and Technology, 2024. — 63 p.

[4] Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union (NIS2) // Official Journal of the European Union. — 2022.

[5] Про основні засади забезпечення кібербезпеки України : Закон України від 05.10.2017 № 2163-VIII (зі змін. і допов.). — Режим доступу : https://zakon.rada.gov.ua.

[6] ENISA. Cybersecurity Threat Landscape : Annual Report. — Heraklion : European Union Agency for Cybersecurity, 2023.

[7] Шевченко В. Л. Кібербезпека критичної інфраструктури: виклики та підходи до захисту // Наукові праці. — 2022. — № 4. — С. 45–52.

[8] Козловський С. В. Стандартизація інформаційної безпеки в умовах цифрової трансформації // Вісник Національного технічного університету. — 2023. — № 2. — С. 61–68.

ПОРІВНЯЛЬНИЙ ОГЛЯД СТАНДАРТІВ КІБЕРБЕЗПЕКИ ЄВРОПЕЙСЬКОГО СОЮЗУ ТА НОРМАТИВНОЇ БАЗИ УКРАЇНИ

Старкова О.В., Почанський О.М.
E-mail: olha.starkova@hneu.net, oleh.pochanskiy@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

Забезпечення кібербезпеки є одним із ключових пріоритетів сучасних держав в умовах цифровізації та зростання кількості кіберзагроз. Європейський Союз сформував системну та багаторівневу нормативну базу у сфері кібербезпеки, яка слугує орієнтиром для країн-партнерів, зокрема України, що перебуває в процесі гармонізації законодавства з європейськими стандартами.

У ЄС базовим регуляторним документом є Директива NIS та її оновлена редакція NIS2, які встановлюють вимоги до кіберзахисту операторів критично важливих і важливих суб’єктів. Аналогічні підходи в Україні закріплені у Законі України «Про основні засади забезпечення кібербезпеки України», який визначає суб’єктів національної системи кібербезпеки, принципи захисту критичної інформаційної інфраструктури та координацію між державними органами. Водночас NIS2 містить жорсткіші вимоги до управління ризиками, відповідальності керівництва та обов’язкового повідомлення про кіберінциденти, що лише частково відображено в українському законодавстві [1, 2].

У сфері захисту даних у ЄС ключову роль відіграє GDPR, який встановлює обов’язкові технічні й організаційні заходи безпеки та суворі вимоги до повідомлення про порушення [3]. В Україні подібні положення закріплені у Законі України «Про захист персональних даних», однак він має менш деталізовані вимоги щодо управління ризиками та відповідальності операторів даних, що створює потребу в подальшому наближенні до європейських норм [4, 5].

Як у ЄС, так і в Україні важливим технічним орієнтиром є міжнародні стандарти серії ISO/IEC 27000, зокрема ISO/IEC 27001. В Україні ці стандарти впроваджуються як національні (ДСТУ ISO/IEC 27001) та застосовуються у державному й корпоративному секторах для побудови систем управління інформаційною безпекою, що забезпечує сумісність з європейськими підходами [6].

Європейський Cybersecurity Act запроваджує єдину систему сертифікації кібербезпеки та закріплює провідну роль агентства ENISA. В Україні функції координації та реагування

виконують Держспецзв'язку, CERT-UA та інші суб'єкти, однак єдина система сертифікації кібербезпеки за зразком ЄС перебуває на стадії розвитку [7].

Таким чином, українська нормативна база з кібербезпеки загалом відповідає базовим принципам європейських стандартів, однак потребує подальшої гармонізації з вимогами NIS2 та GDPR, зокрема в частині управління ризиками, відповідальності керівництва та сертифікації засобів кіберзахисту. Зближення підходів ЄС і України є важливим кроком для підвищення кіберстійкості держави та інтеграції у європейський цифровий простір.

Література

[1] Directive (EU) 2016/1148 (NIS Directive). Official Journal of the European Union. URL: https://eur-lex.europa.eu/

[2] Directive (EU) 2022/2555 (NIS2 Directive). Official Journal of the European Union. URL: https://eur-lex.europa.eu/

[3] Regulation (EU) 2016/679 (GDPR). General Data Protection Regulation. URL: https://eur-lex.europa.eu/

[4] Закон України «Про основні засади забезпечення кібербезпеки України». URL: https://zakon.rada.gov.ua/

[5] Закон України «Про захист персональних даних». URL: https://zakon.rada.gov.ua/

[6] ISO/IEC 27001:2022. Information Security Management Systems – Requirements. URL: https://www.iso.org/

[7] CERT-UA. Офіційний сайт урядової команди реагування на комп'ютерні інциденти України.URL: https://cert.gov.ua/

АВТОМАТИЗОВАНА СИСТЕМА РОЗВІДКИ ТА ASSET MANAGEMENT ДЛЯ ОРГАНІЗАЦІЙ НА ОСНОВІ ГРАФОВИХ БАЗ ДАНИХ

Тугай А.С., Пасюк Б.Б.
E-mail: a.tuhai@ukma.edu.ua, b.pasiuk@ukma.edu.ua
Київ, Національний університет «Києво-Могилянська академія»

Прийнято вважати, що OSINT з відкритих чи закритих джерел інформації - це діяльність розвідувальних служб держав світу. Наразі це є обов'язковим етапом будь-якого тестування на проникнення (пентестингу), аудиту або Red Teaming assessment. Записи DNS, дані SSL-сертифікатів, перевірка "Shadow IT", піддоменів, "злитої" інформації в мережах Інтернет та DarkNet, дані про співробітників та інші корпоративні дані надають змогу отримати повнішу картину про стан безпеки компанії, аніж внутрішні звіти про стан безпеки.

Опрацювання такого величезного обсягу даних вимагає нових підходів, у той час як наявні інструменти серед професіоналів інформаційної безпеки не встигають за змінами у світі. Основна проблема полягає у тому, щоб автоматизувати збір хаотичних цифрових "крихт" та об'єднати їх в єдину картину, перш ніж це зроблять зловмисники. Для глибшого розуміння ми маємо скористатися моделлю Cyber Kill Chain.

Фаза виявлення (Discovery) є першим і найбільш значущим етапом будь-якого технічного аудиту безпеки [1]. У термінології MITRE ATT&CK ця діяльність вказується як тактика "Reconnaissance" (TA0043), метою якої є збір даних для подальшої підготовки до атак [2]. На даному етапі варто виділити проблему: сучасні організації часто не володіють повною картою своїх активів, залишаючи "сліпі зони" у вигляді забутих піддоменів, тестових серверів або некоректно налаштованих DNS-записів, які суперечать специфікаціям RFC1035 [3]. Ручний або напівавтоматизований моніторинг є трудомістким та не завжди можливим процесом через масштаб інфраструктури, а існуючі засоби автоматизації часто фокусуються на точкових перевірках, ігноруючи непрямі зв'язки між активами. Це створює ситуацію, коли в компанії оперують застарілими (неактуальними) даними, в той час як зловмисники, використовуючи тактики активної розвідки (MITRE ATT&CK T1595), знаходять та експлуатують ці "сліпі зони" (Shadow IT) швидше, ніж організація встигає їх інвентаризувати [2].

Хоч дані легкодоступні, але їх ефективна обробка залишається викликом. Комерційні платформи, гарними представниками яких є Maltego, забезпечують якісну візуалізацію, але є фінансово недоступними для багатьох дослідників та мають закритий код [4]. З іншого боку, популярні open-source рішення (наприклад, SpiderFoot) часто мають істотні проблеми інформаційного перевантаження: вони генерують масиви розрізнених табличних даних, в яких важко виявити критичні зв'язки [5]. У результаті аналітики отримують плоскі списки хостів замість розуміння контексту їхніх залежностей. Традиційні реляційні бази даних, що лежать в основі більшості цих систем, неефективно обробляють рекурсивні запити, що унеможливлює швидкий пошук складних ланцюжків “атак у глибину”: реляційна модель створює надмірне навантаження при спробі моделювання високозв'язних даних, призводячи до проблеми “JOIN bombs” [6]. Крім цього, якщо потреба компанії полягає в постійному моніторингу мережі, то всі вищевказані інструменти не мають зручного функціонального стеку для цього: замість автоматичного періодичного сканування ми отримуємо обмеження та відсутність гнучкості. Відсутність доступного інструменту, який поєднував би автоматизацію збору real-time інформації за доменом компанії (OSINT) з графовою аналітикою, створює суттєву прогалину в арсеналі фахівців з кібербезпеки.

Для підвищення рівня, а також швидкості виявлення внутрішніх активів компанії (автоматизація процесу розвідки) пропонується інтеграція спеціалізованих open-source мікросервісів у поєднанні з графовими базами даних на противагу реляційним, що значно прискорить отримання актуальної інформації по активах компанії.

Запропонована методологія збору даних по активах компанії поєднує пасивні та активні методи розвідки для забезпечення повноти та достовірності даних. Це дозволяє виявляти “тіньові” активи набагато швидше. Активна верифікація знайдених активів включає аналіз банерів сервісів (Banner Grabbing) [9]. Це дозволяє ідентифікувати версії програмного забезпечення та автоматично співставляти їх з базами відомих вразливостей для оцінки потенційної поверхні атаки, що дозволяє не використовувати дороговартісні системи (наприклад, Nessus або Qualys) та застосовувати кастомні інтеграції.

Для перетворення зібраних даних на придатну для прийняття рішень розвідувальну інформацію важливим є впровадження метрик оцінки ризиків. Пропонується інтегрувати метрики кількісної оцінки ризиків, зокрема RAV (Risk Assessment Value) згідно з методологією OSSTMM 3 [10]. Це дозволить не просто надавати список активів за алфавітом, а й числове відображення рівня загрози для кожного активу, базуючись на його експозиції та наявності відомих вразливостей.

Важливим завершальним етапом буде правильне та зрозуміле представлення отриманих даних для аналітиків безпеки. Для візуалізації результатів доцільно використовувати бібліотеки інтерактивного моделювання графів безпосередньо у веб-інтерфейсі (наприклад, Cytoscape.js) [11]. У результаті замість аналізу статичних звітів аналітик отримає можливість динамічного дослідження зв'язків на єдиному дашборді, що пришвидшить швидкість реагування при виникненні реальних або потенційних загроз периметру та активів компанії.

Метод автоматизації OSINT, описаний вище, дозволяє вирішити проблему фрагментації даних, яка виникає під час інвентаризації активів. Інструменти з відкритим кодом забезпечують універсальність, зрозумілість алгоритмів та можливість налаштування системи відповідно до конкретних вимог управління активами. Це робить збір та автоматизований аналіз інформації про активи компанії проактивними інструментами кібербезпеки.

Література

[1] Scarfone K. [et al.]. Technical Guide to Information Security Testing and Assessment (NIST Special Publication 800-115) [Електронний ресурс]. – Режим доступу: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

[2] The MITRE Corporation. Reconnaissance, Tactic TA0043 - Enterprise | MITRE ATT&CK [Електронний ресурс]. – Режим доступу: https://attack.mitre.org/tactics/TA0043/

[3] Mockapetris P. RFC 1035: Domain names - implementation and specification [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc1035

[4] Maltego Technologies. Maltego: Cyber Security & Cyber Threat Intelligence [Електронний ресурс]. – Режим доступу: https://www.maltego.com/

[5] Micallef S. SpiderFoot: Automate OSINT [Електронний ресурс]. – Режим доступу: https://github.com/smicallef/spiderfoot

[6] Robinson I., Webber J., Eifrem E. Graph Databases: New Opportunities for Connected Data [Електронний ресурс]. – Режим доступу: https://www.oreilly.com/library/view/graph-databases-2nd/9781491930885/

[7] Ramírez S. FastAPI Documentation [Електронний ресурс]. – Режим доступу: https://fastapi.tiangolo.com/

[8] Laurie B., Langley A., Kasper E. RFC 6962: Certificate Transparency [Електронний ресурс]. – Режим доступу: https://datatracker.ietf.org/doc/html/rfc6962

[9] OWASP Foundation. OWASP Web Security Testing Guide (WSTG) v4.2 [Електронний ресурс]. – Режим доступу: https://owasp.org/www-project-web-security-testing-guide/

[10] Herzog P. OSSTMM 3: The Open Source Security Testing Methodology Manual [Електронний ресурс]. – Режим доступу: https://www.isecom.org/OSSTMM.3.pdf

[11] Cytoscape.js: Graph theory / network library for analysis and visualisation [Електронний ресурс]. – Режим доступу: https://js.cytoscape.org/

АВТОМАТИЗАЦІЯ СУБ'ЄКТИВНИХ МЕТОДІВ ОЦІНЮВАННЯ РИЗИКІВ ЗАСОБАМИ ВІЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Чуєва А.О.
Керівник: Солодовник Г.В.
E-mail: ganna.solodovnyk@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

Вступ. У багатьох сферах діяльності, для яких притаманні висока динамічність та відсутність статистичних даних, зокрема в IT-менеджменті, кібербезпеці, управлінні проєктами та критичною інфраструктурою, суб'єктивні методи оцінювання ризиків залишаються одним з основних інструментів прийняття управлінських рішень. Водночас такі методи значною мірою залежать від суджень експертів, що може призводити до непослідовності та упередженості результатів оцінювання. Нівелювати вказані недоліки можна завдяки провадженню автоматизованих засобів обробки експертних оцінок. Зручним та економічно доцільним є використання інструментарію програмного забезпечення з відкритим вихідним кодом.

Актуальність. Суб'єктивні методи оцінювання ризиків, зокрема експертні опитування, широко застосовуються у випадках обмеженості статистичних даних або високої складності об'єктів аналізу. Проте їх ефективність значною мірою залежить від прозорості процедур оцінювання та подальшої обробки експертної інформації. Тому актуальності набуває питання автоматизації суб'єктивних методів оцінювання ризиків з використанням вільного програмного забезпечення. Застосування відкритих програмних засобів дозволяє знизити фінансові витрати, підвищити доступність інструментів аналізу та забезпечити прозорість алгоритмів обробки даних. Крім того, відкритий код створює можливості для адаптації, розширення функціональності та незалежної верифікації методів оцінювання, що є важливим з наукової та практичної точок зору.

Автоматизація суб'єктивних методів забезпечує формалізацію процесів оцінювання ризиків та зменшенню впливу людського фактору на результати експертних оцінок. Використання програмних засобів дозволяє реалізувати багатокритеріальні моделі, проводити сценарний аналіз, виконувати агрегацію експертних думок та забезпечувати

збереження і повторне використання результатів оцінювання. Це особливо важливо для підтримки прийняття рішень у динамічних та ризик-орієнтованих середовищах.

Метою роботи є розробка програмної реалізації використання методів експертного оцінювання для визначення рівня ризику в прийнятті управлінських рішень. Об'єктом роботи є процес оцінювання ризиків, предметом – методи експертного оцінювання ризиків та їх програмна реалізація.

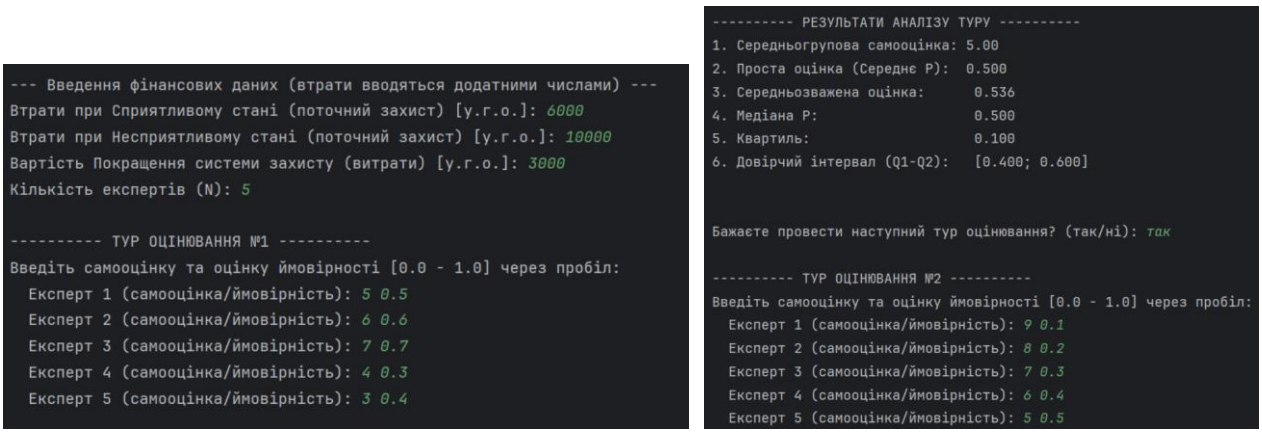
Сучасні дослідження зосереджуються на проактивному управлінні ризиками в складних організаційних та технічних середовищах, з можливістю створення інформаційних систем, в яких інтегруються методи експертних оцінок, багатокритеріального аналізу та автоматизованих розрахунків [1]. Водночас визначають проблему суб'єктивності експертних суджень, яка є характерною для багатьох галузей, зокрема й управління ризиками. У [2] зазначається, що неконтрольована суб'єктивність може призводити до суттєвих викривлень результатів оцінювання та, як наслідок, до хибності рішень, які були прийняті на їх основі. Результати таких досліджень демонструють підвищення точності та узгодженості оцінок, отриманих за застосування засобів автоматизації, порівняно з традиційними неавтоматизованими підходами.

Управлінські рішення приймаються на підставі інформації про стан об'єкта управління та про чинники, під впливом яких відбувається його функціонування, і які описують стан зовнішнього середовища. В загальному випадку динамічність та невизначеність чинників функціонування може бути виражена ймовірністю настання того або іншого стану зовнішнього середовища. Задача, яка розглядається в роботі, полягає у визначені ймовірності настання несприятливого стану функціонування, за якого загрози та атаки на інформаційні ресурси об'єкта автоматизації будуть значними та призведуть до значних втрат в разі поточного рівня захисту інформації компанії. Необхідно прийняти рішення щодо вкладання коштів у заходи з покращення системи захисту інформації.

Для визначення ймовірності несприятливого стану застосовано метод експертного оцінювання Делфі [3]. Основним недоліком експертних оцінок є їх високий рівень суб'єктивності, внаслідок чого відсутня впевненість у достовірності отриманих результатів. Крім того можливим є виникнення негативних процесів таких як упередженість експертів, вплив авторитету, значна зашумленість тривіальними ідеями, феномен зрушення оцінок в бік підвищення їх ризикованості.

Нівелювати зазначені негативні риси дозволяє метод Делфі, який передбачає проведення декількох турів опитувань за умови ознайомлення експертів з оцінками та коментарями їх колег, що дозволяє знизити такі недоліки експертних оцінювань як: суб'єктивність, тиск авторитету, забруднення зайвими ідеями та підвищення їх ризикованості. Процес аналізу даних передбачає обчислення: середньогрупової самооцінки, простої оцінки, середньозваженої оцінки, медіани та довірчого інтервалу, за розмірами якого визначається ступінь погодженості експертних оцінок поточного туру.

Екранні форми тестування роботи розробленої програми наведені на рисунку 1. Програмна реалізація дозволяє: ввести необхідні фінансові показники для визначення доцільності поліпшення системи захисту за різних значень ймовірності настання несприятливого стану (рис. 1 *а)*); надання експертами оцінок цієї ймовірності (рис. 1 *а)*); отримати аналітику погодженості думок експертів поточного туру опитування (рис. 1 *б)*); ініціалізувати наступний тур опитувань, у випадку незадовільного розміру довірчого інтервалу (рис. 1 *б)*).



а) *б)*

Рисунок 1 – Екранні форми проведення експертних оцінювань

Екранні форми результатів роботи програмної реалізації наведено на рисунку 2. Результати містять значення ймовірностей сприятливого та несприятливого станів за аналізу фінального туру експертних оцінювань та очікувані фінансові результати втрат за покращення системи захисту інформації та без нього. На підставі порівняння цих втрат формується та виводиться у текстовому вигляді висновок щодо доцільності вкладання коштів у покращення системи захисту інформаційних ресурсів компанії. Наочність результатів розрахунків підвищено за рахунок генерації графіка очікуваних втрат за поліпшення системи захисту та без нього, виражених в умовних грошових одиницях.

Вхідними даними програми є значення втрат, які понесе об’єкт автоматизації за настання сприятливого та несприятливого стану зовнішнього середовища в кожному з випадків: з покращеною системою захисту та з системою захисту у поточному стані, а також вартість витрат на модернізацію системи захисту інформації.

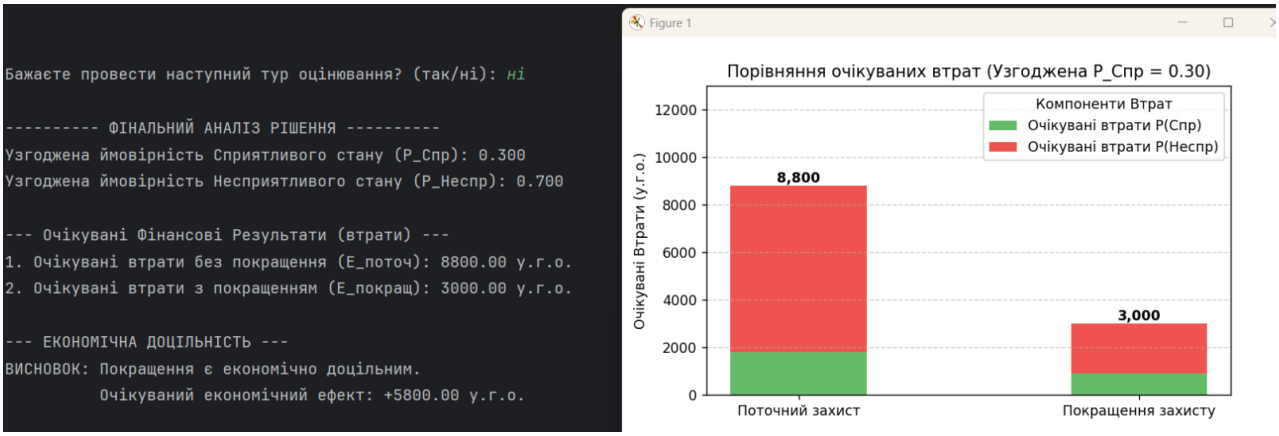


Рисунок 2 – Екранні форми результатів роботи програми

Визначення втрат від пошкодження інформаційних ресурсів теж є складною слабоформалізованою задачею. В загальному випадку такі втрати можна поділити на дві основні групи: неочікувані витрати на відновлення інформаційних ресурсів та втрати в основній діяльності об’єкта автоматизації, які пов’язані з пошкодженням інформаційних ресурсів. Окремо слід визначити можливі репутаційні втрати, які не можна визначити у грошовому еквіваленті, так само як загрозу здоров’ю та життю людей або екологічної катастрофи. З метою визначення таких втрат також доцільно використовувати експертні методи оцінювання.

Параметрами програми є кількість експертів та надані ними самооцінки, які визначають ступінь обізнаності експертів в предметній області.

Розробка програмної реалізації відбувалася мовою програмування Python. Розрахунок очікуваних фінансових втрат виконано стандартними можливостями Python. Для графічного представлення результатів застосовано бібліотеку Matplotlib.

Висновок. Розроблена у роботі програмна реалізація визначення доцільності вкладання коштів у покращення системи захисту інформаційних ресурсів може бути впроваджена як частина системи підтримки прийняття рішень в сфері управління фінансовими засобами. Подальший розвиток роботи можливий за рахунок вдосконалення інтерфейсу користувача. Функціонал програмної реалізації може бути доповнений наданням експертам можливості введення коментарів з обґрунтуванням їх оцінок. Окремим питанням модернізації є розробка функції оцінювання втрат від пошкодження інформаційних ресурсів.

Література

[1] Claycamp H. G. Elicitation of expert knowledge and probabilities for controlling subjectivity in risk-based decision making // Pharmaceutical Technology. – 2023. – URL: https://www.pharmtech.com/view/elicitation-of-expert-knowledge-and-probabilities-for-controlling-subjectivity-in-risk-based-decision-making

[2] En-Naaoui A., El Hami A., El Oualkadi A. A novel decision support system for proactive risk management // Decision Support Systems. – 2024. – Vol. 178. – Art. 114140. – DOI: https://doi.org/10.1016/j.dss.2024.114140

[3] Cabrera J.S., Reyes A.R.L., Lasco C.A. Multicriteria Decision Analysis on Information Security Policy: A Prioritization Approach // Advances in Technology Innovation. 2021. Т. 6, № 1. С. 31–38. URL: https://ojs.imeti.org/index.php/AITI/article/view/5476

ВИКОРИСТАННЯ KICAD ДЛЯ ПРОЕКТУВАННЯ ЕЛЕКТРОННИХ СХЕМ І ДРУКОВАНИХ ПЛАТ

Шапо В.Ф., Улізько Д.О.
E-mail: vladlen.shapo@gmail.com, ulizko1994@gmail.com
Одеса, Інститут Військово-Морських Сил

У теперішній час надзвичайно швидко розвиваються різноманітні комп’ютерні системи, які з апаратної точки зору базуються на множині аналогових та цифрових електронних компонентів та друкованих платах, на які ці компоненти встановлюються. Тому питання використання програмного забезпечення (ПЗ) для проектування електронних схем та друкованих плат в електроніці, робототехніці, системах зв’язку та передавання даних, віддаленому управлінні безліччю об’єктів у цивільному житті, промисловості, військовій справі набуло великої актуальності. Інженерні та навчальні процеси потребують доступу до ефективних інструментів, які дозволяють створювати, перевіряти за допомогою моделювання та реалізовувати електронні проекти на всіх етапах розробки. При цьому комерційне програмне забезпечення (наприклад, давно відомі PCAD, OrCAD) часто має високу складність, великі системні вимоги, високу вартість і обмежує можливості самостійної роботи, що не дає можливості навчатися та робити перші професійні кроки.

KiCad [1] – комплекс ПЗ з відкритим кодом для проектування електронних схем і друкованих плат, що може бути використаний для вирішення вказаних проблем. KiCad підтримує і інтегрований процес проектування, при якому схема та друкована плата розробляються одночасно, і їх редагування окремо для спеціальних випадків. Система також містить інструменти для спрощення розробки схем і плат. KiCad працює на всіх основних операційних системах і на комп’ютерах з різною конфігурацією, у т.ч. морально застарілих. Система підтримує розробку багатошарових друкованих плат (до 32 шарів) і може використовуватися в проєктах будь-якої складності. Остання версія KiCad 9.0.7 випущена 1.01.2026 і розробляється командою ентузіастів-програмістів та інженерів усього світу з метою створення якісного безкоштовного ПЗ з відкритим кодом, придатного для професійної розробки електронних схем і друкованих плат, а також в навчальному процесі в множині навчальних закладів.

Типовий робочий процес у KiCad складається зі створення принципової схеми та розведення друкованої плати на її базі. В комплексному проекті спочатку створюється принципова схема. Якщо потрібні схемотехнічні елементи відсутні у вбудованих стандартних бібліотеках, користувач може створити їх самостійно. На цьому етапі також обираються посадкові місця для кожного компонента і, за потреби, створюються власні посадкові місця (корпуси елементів). Після завершення розробки принципової схеми та проходження перевірки електричних правил Electrical Rule Check (ERC) дані про з'єднання елементів схеми переноситься до редактора друкованих плат.

Принципова схема показує, які компоненти входять до проєкту та як вони з'єднані; редактор друкованих плат використовує ці дані для автоматизації трасування доріжок та запобігання появі невідповідностей між схемою та платою. Розміщення компонентів вимагає уважного підбору місця для кожного посадкового місця (корпусу елементу) на платі. Після розміщення компонентів модуль трасування створює між ними доріжки відповідно до з'єднань у схемі, а також з урахуванням інших електричних вимог, таких як опір мідних доріжок, взаємні наводки та інше. Процеси моделювання працездатності принципових схем потребують окремого уважного розгляду.

Коли трасування плати завершене і вона пройшла перевірку на відповідність правилам проектування Design Rule Check (DRC), генеруються вихідні дані для виготовлення плати, після чого її можна виготовити на відповідному виробництві. Для ефективного виконання всього циклу проектування KiCAD надає набір спеціалізованих інструментів, коротко розглянутий нижче.

1. Schematic Editor (редактор схем) дозволяє створювати та редагувати електронні схеми (рис. 1), моделювати роботу схем за допомогою модуля SPICE, генерувати списки компонентів (Bill of Materials, BOM).

2. Symbol Editor (редактор компонентів) дозволяє створювати та редагувати символи компонентів (транзистори, резистори і т.д.), додавати їх до бібліотек та керувати бібліотеками символів. PCB Editor створює та редагує друковані плати, експортує 2D/3D файли, готує файли для виробництва. Тут формуються фізичні з'єднання компонентів на платі.

3. Footprint Editor (редактор посадочних місць) дозволяє створювати та редагувати посадкові місця компонентів та керувати їхніми бібліотеками. Посадкове місце – це фізичний контур для монтажу елемента на плату, який залежить від типу та розміру корпусу.

4. Модуль GerbView переглядає Gerber- та drill-файли (дані щодо опису шарів плати та подальшого свердлення відповідно), необхідні для виробництва плат (рис. 2). Використовуються стандарты Gerber-файлів RS-274-X або більш новий X2/X3.

5. Bitmap2Component конвертує растрові зображення у символи або посадкові місця для легкого додавання нестандартних елементів.

6. Printed Circuit Board (PCB) Calculator (модуль розрахунку параметрів плат) дозволяє розраховувати параметри компонентів, ширину доріжок, зазори і т. ін., допомагає дотримуватися електричних норм і уникнути помилок.

7. Page Layout Editor (редактор макетів сторінок) дозволяє створювати та редагувати робочі файли для документації і звітів.

Ці програмні модулі забезпечують повний цикл проектування схем і друкованих плат, що застосовується для розробки широкого спектру електронних пристроїв.

KiCAD дозволяє керувати усіма етапами розробки електронних принципових схем і друкованих плат без залежності від комерційного програмного забезпечення та дозволяє модифікувати, тестувати, редагувати та модернізувати конкретні проєкти. Це надає можливість зменшити ймовірність помилок завдяки високому рівню автоматизації і робить KiCAD ефективним інструментом для навчання та розробки реальних систем.

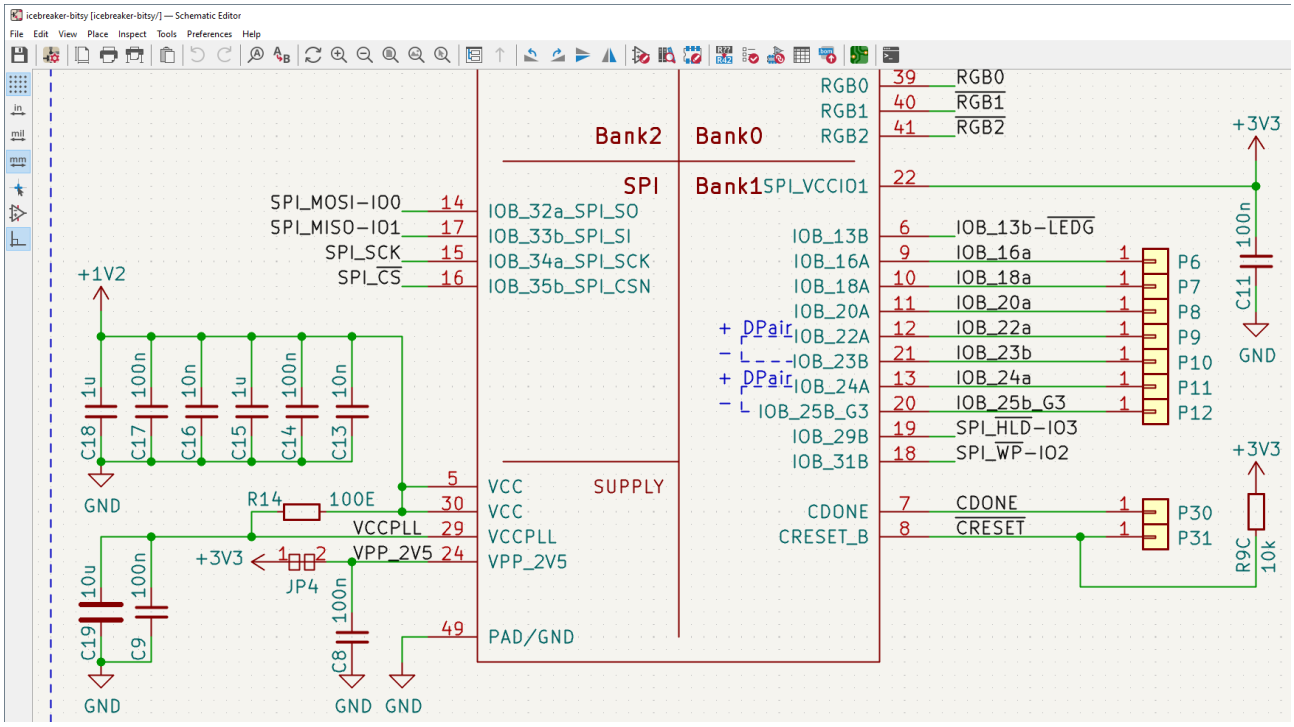


Рис. 1. Графічний редактор побудови принципових схем в KiCAD

Підготовлений проєкт плати може бути показаний у тривимірному вигляді (рис. 3).

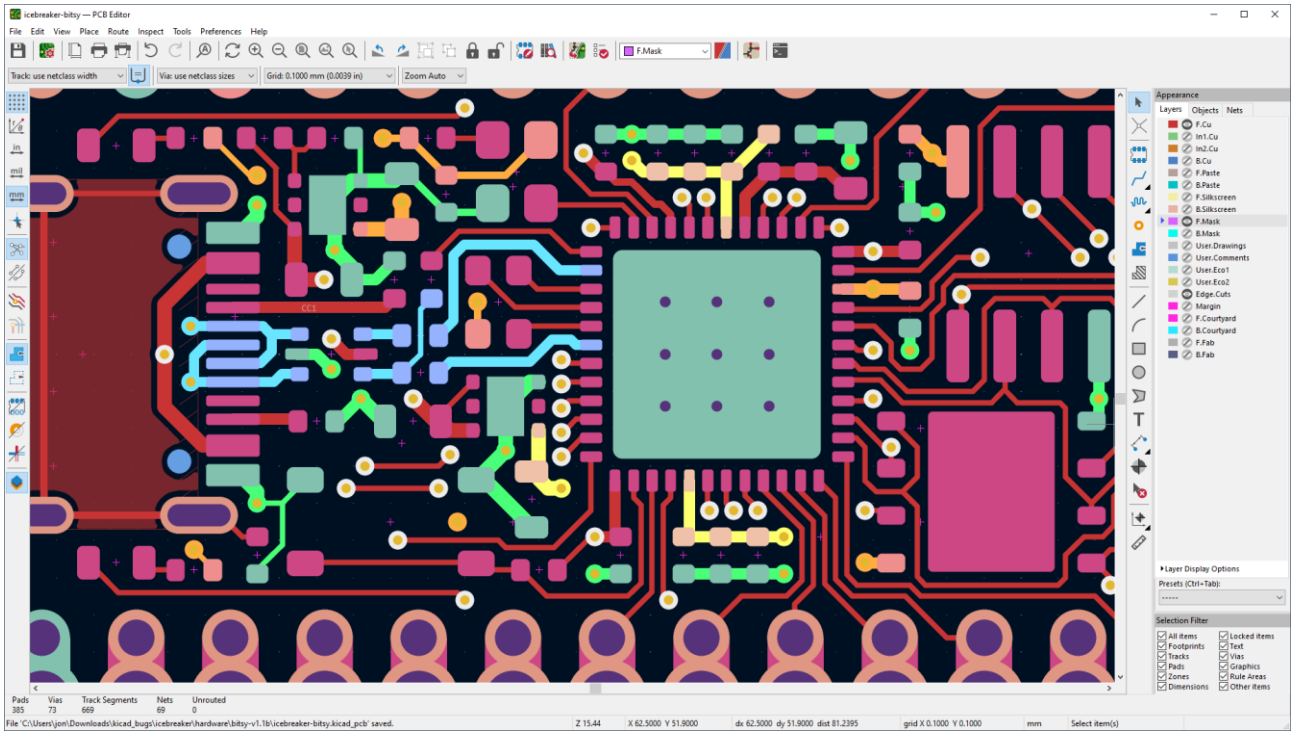


Рис. 2. Редактор проектування друкованих плат в KiCAD

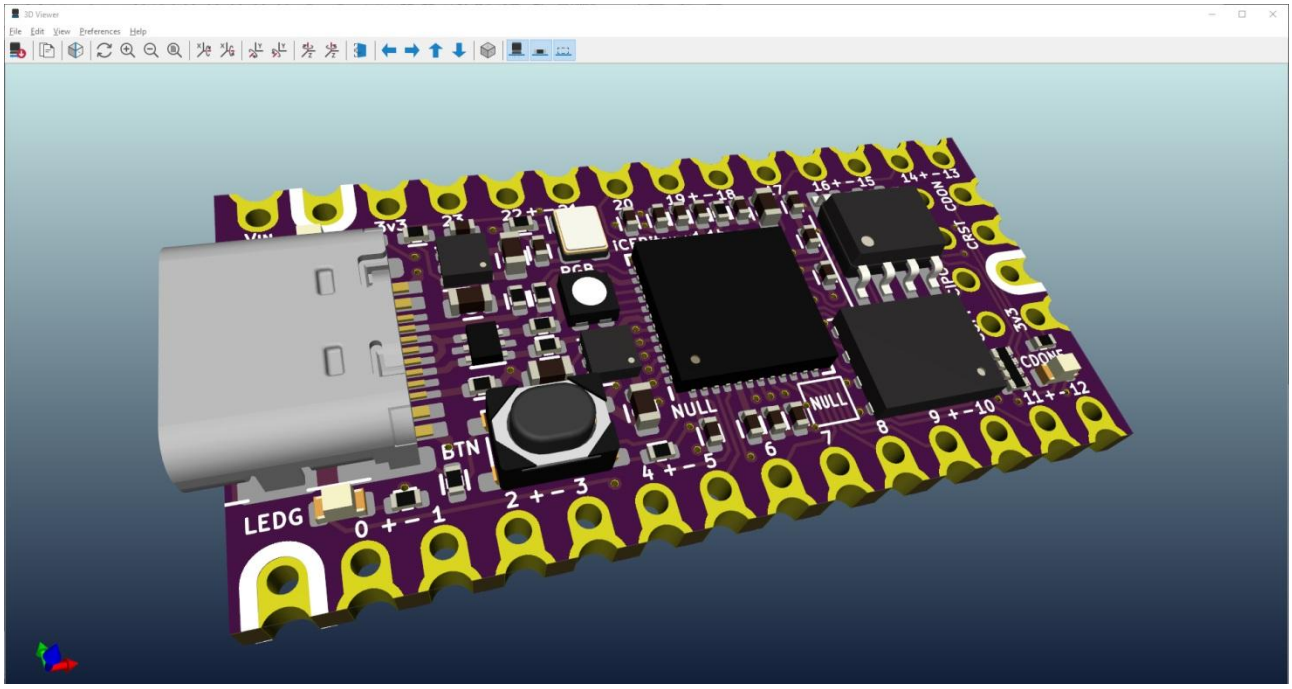


Рис. 3. Тривимірна візуалізація проекту KiCAD у вбудованому засобі перегляду

Використання KiCAD в першу чергу в навчальних цілях дозволить студентам та курсантам безлічі навчальних закладів отримати широкий спектр знань в області електроніки навіть на морально застарілій комп'ютерній техніці.

Література

[1] KiCad: A Cross Platform and Open Source PCB Design Suite. [Електронний ресурс]. – Режим доступу до ресурсу: https://www.kicad.org/

ВИКОРИСТАННЯ РЕДАКТОРУ РОЗДІЛІВ ДИСКІВ GPARTED ДЛЯ ВИРІШЕННЯ НАВЧАЛЬНИХ ТА ПРОФЕСІЙНИХ ЗАДАЧ

Шапо В.Ф., Шевченко А.О.
E-mail: vladlen.shapo@gmail.com, andrii1501@ukr.net
Одеса, Інститут Військово-Морських Сил

Задача редагування розділів жорстких дисків та твердотільних накопичувачів різних об'ємів і типів, з різними файловими системами є вельми актуальною для великої кількості професіоналів та просунутих користувачів в усьому світі. Кількість задач, для яких використовуються комп'ютерні системи, невпинно збільшується, їх складність швидко зростає, а з тим і об'єм даних, які треба зберігати, також дуже швидко збільшується. Останні 15-20 років існує жорстка необхідність створювати численні резервні копії даних, при чому вони можуть мати різний рівень актуальності та допускають різний час доступу до даних (умовно кажучи, «гарячі», «теплі», «холодні» резервні копії), і багато таких копій зберігається саме на жорстких дисках. У зв'язку з зазначеним вище відносно нещодавно з'явилися такі технології та пристрої, як RAID (Redundant Array of Independent Disks, надлишковий масив незалежних дисків) та NAS (Network Attached Storage, мережеве сховище даних), які потребують використання мінімум двох, а часто і суттєво більшої кількості жорстких дисків. Досить стандартною стала ситуація, коли на одному комп'ютері використовується і твердотільний накопичувач, що забезпечує малий час завантаження операційної системи та швидке завантаження найчастіше використовуваних даних та програмних продуктів, але коштує відносно дорого, і класичний механічний жорсткий диск, на якому знаходяться програмні продукти та дані, що використовуються на так часто, і який

має суттєво менші швидкості запису та читання даних, але і за суттєво меншу ціну (як абсолютну, так і за 1 Гбайт) у порівнянні з твердотільними накопичувачами. Взагалі жорсткі диски можна розділити на 4 класи.

1. Desktop (диски для звичайних комп'ютерів). Ці диски використовуються в комп'ютерах, ноутбуках та в якості зовнішніх, що підключаються за інтерфейсами USB, eSATA, FireWire (IEEE1394). Вони мають низьку ціну, призначені для відносно невеликого навантаження (робота 5 днів на тиждень по 8 годин для офісних комп'ютерів, 4-6 годин на день для домашніх комп'ютерів), мають низьку стійкість до вібрації.

2. NAS (Network Attached Storage, – мережеве сховище даних). Ці диски мають вищі показники з надійності та складності режиму роботи у порівнянні з дисками класу Desktop. Рекомендоване виробниками навантаження суттєво більше, ніж у дисків Desktop, що дозволяє використовувати такі диски в домашніх NAS з кількістю 5-8 шт. Обмеження обумовлено зростанням вібрації при збільшенні числа дисків в системі.

3. Surveillance. Ці жорсткі диски розроблені для систем відеоспостереження. При їх виробництві використовуються компоненти з найкращими показниками при роботі на запис. Основна задача такого диску, – записувати потокові дані з відеокамер. Співвідношення часу запису/читання приблизно 90/10 (90% часу – запис, 10% часу – читання). Дуже не бажано, щоб читання виконувалось при запису даних з камер. Клас Surveillance добре показує себе в невеликих (до 8 дисків) дискових масивах.

4. Enterprise. Це диски корпоративного класу для використання в серверах, що мають високі надійність, показник напрацювання на відмову і найбільше, у порівнянні з іншими дисками, річне навантаження. Такі диски зазвичай можуть роками записувати і зчитувати дані цілодобово з максимальною швидкістю.

Дуже сильно різняться групи користувачів і за тим, наскільки сучасні комп'ютерні системи вони використовують. Так, багато хто використовує застарілі комп'ютерні системи для роботи в якості «електронної печатної машинки», щоб вистачало швидкодії для роботи з текстовими редакторами, електронними таблицями та виконання задач доступу до мережі Інтернет. До другої групи можна віднести співробітників організацій, які не мають сучасного парку комп'ютерних систем, та не можуть забезпечити комп'ютером кожного співробітника (школи, коледжі, вищі навчальні заклади, центри підвищення кваліфікації, медичні заклади, банки тощо), або студентів, які вимушені по черзі використовувати один і той самий комп'ютер в комп'ютерних класах різноманітних навчальних закладів.

Під час війни в українських військових з'явилася необхідність використовувати будь-які морально та фізично застарілі комп'ютерні системи самостійно (головний критерій в таких випадках, – хоча б мінімальна працездатність), тому задача модернізації дискових підсистем комп'ютерів та управління дисками для різних користувачів при вирішенні різних задач є дуже важливою.

Можливим рішенням для управління розділами дисків є безкоштовне і вільно розповсюджуване програмне забезпечення GParted Live [1]. Його версія 1.8.0-2 – нове стабільне оновлення на базі Linux Debian. Воно включає нещодавно випущену версію GParted 1.8 і оновлені системні компоненти. Випуск сформовано з репозиторію Debian Sid станом на 27.01.26. Це забезпечує нові основні компоненти в порівнянні з попередніми образами. Ядро Linux оновлено до версії 6.18.5-1, що покращує підтримку апаратного забезпечення та забезпечує широку сумісність. Ця версія вводить новий механізм, який запобігає появі чорних екранів при завантаженні на системах з деякими графічними конфігураціями.

Головний екран GParted представлено на рисунку 1.

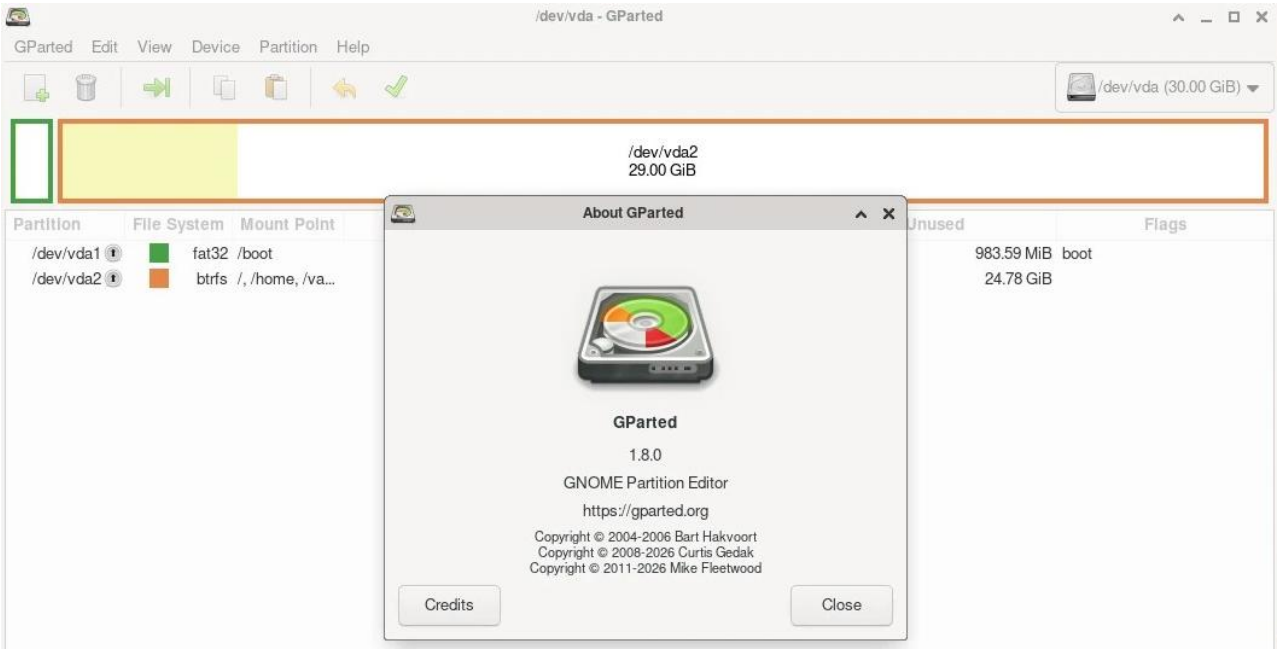


Рис. 1. Головне вікно редактора розділів дисків GParted

GParted Live – автономна завантажувальна система Linux, призначена для управління дисками та розділами. Вона підтримує файлові системи bcachefs, btrfs, exfat, ext2/ext3/ext4, fat16/fat32,hfs/hfs+, linux-swap, lvm2 pv, minix, nilfs2, ntfs, reiserfs/reiser4, udf, ufs, xfs. Систему можна запустити безпосередньо з зовнішніх носіїв. Це дозволяє створювати, змінювати розмір, переміщати, копіювати, перевіряти та видаляти розділи диска без необхідності встановлення операційної системи на комп'ютері.

На рисунку 2 показано перелік файлів GParted, доступних для завантаження.

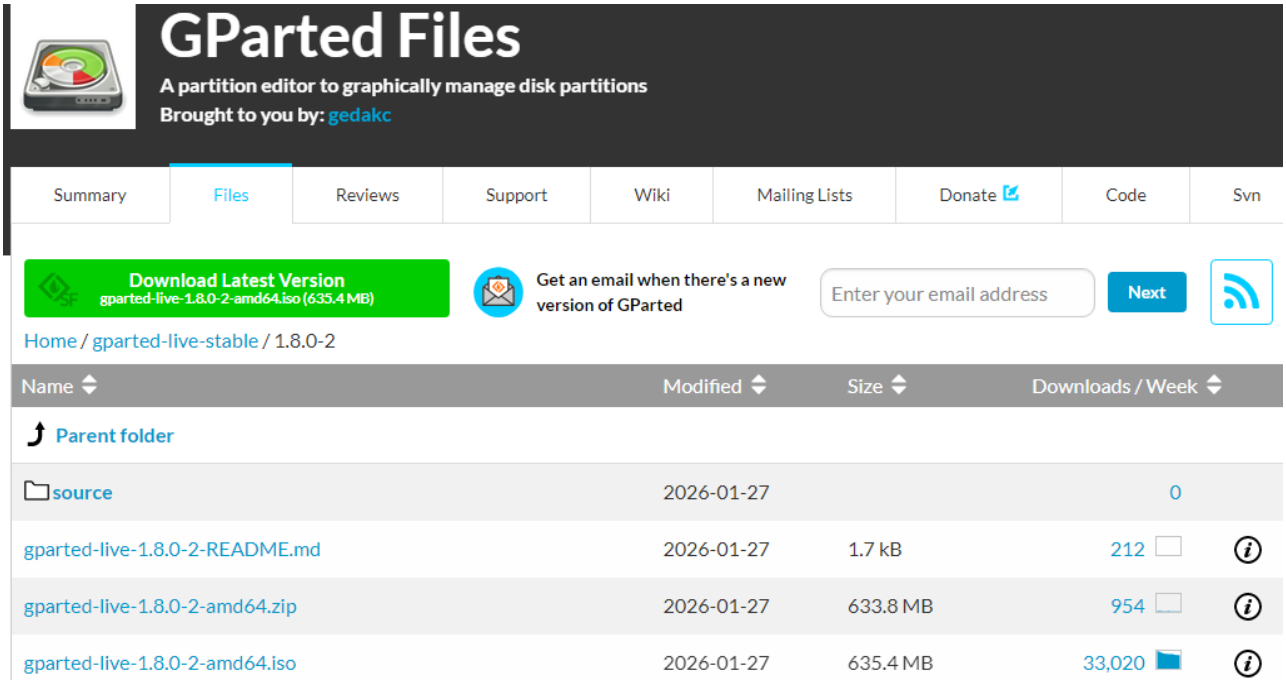


Рис. 2. Перелік файлів GParted, доступних для завантаження

Вказаний випуск GParted також дозволяє навести наглядні приклади для тих, хто вивчає області, пов'язані з кібербезпекою (наприклад, хеш-функції).

Алгоритм MD5: 943e4082a048d86792c4bcc2c50cba4f для файлу gparted-live-1.8.0-2-amd64.iso (32 16-річних символи, 128 біт); 7a2e865c55da5752c046d8c1fd66d39b для файлу gparted-live-1.8.0-2-amd64.zip (32 16-річних символи, 128 біт).

Алгоритм SHA1: cfa1f7d60b4a1e58f7e28009f12ff6c62d7df17d для файлу gparted-live-1.8.0-2-amd64.iso (40 16-річних символів, 160 біт); 59399e5b92ed98fad2a414499f315da85b733c7a для файлу gparted-live-1.8.0-2-amd64.zip (40 16-річних символів, 160 біт).

Алгоритм SHA256: 167a114b25b0cabb8ca921413b777d2693511bc18bc1625ae310b84597b79413 для файлу gparted-live-1.8.0-2-amd64.iso (64 16-річних символів, 256 біт); 6899a3a3a425f9f7bf63b8f3e35cbb3d382cc324e3ad2f888e01ffe2846fe1a0 для файлу gparted-live-1.8.0-2-amd64.zip (64 16-річних символів, 256 біт).

Алгоритм SHA512: 4535fba5d776daf2ce119dd0bbf15bcc9a503af6e1c1cd47a72fdc4d1d9a234c554416b9c4392a573a55c8aadf2b7e88a65f021628987c881238b0694a7ad983 для файлу gparted-live-1.8.0-2-amd64.iso; 089fababf421c306bba5f2b9eecbb85a96fae7cab6475c8a79a6f1b015db67df60aaf7753f353dd04bc7fc33f818776f3aa9ec7561676efe9b1f2baab27eab5e для файлу gparted-live-1.8.0-2-amd64.zip (128 шістнадцятирічних символів, 512 біт).

Алгоритм Blake2: d61f0a8175ce54d3878fff56e697829a6faee2f68690ccb1dcef5e1887084555fb512895251eb59929927b180a13a8001298791cbc674dc1b58a42f88cac360b для файлу gparted-live-1.8.0-2-amd64.iso; 337baa27c89221ac8642682eaeaa18aab29767c98918 16460c84e4e38ecc2e7878e963ba3423c11f79fae0cfd7c9c97d5d6ddc70634c7e50fd8c7c01d39abff0 для файлу gparted-live-1.8.0-2-amd64.zip (128 шістнадцятирічних символів, 512 біт).

Алгоритм Blake3: 9f21100ffe45f913de06d797ad760e165880c7a56aac81f093506a2588cb75fc для файлу gparted-live-1.8.0-2-amd64.iso; 38cb9b32d3b8aec3c5bb00f07eec69ddff07c5ed78a73bd260914349fa388ed2 для файлу gparted-live-1.8.0-2-amd64.zip (64 16-річних символів, 256 біт).

Представлені результати розрахунку хеш-функцій, виконані за різними алгоритмами для файлів, в яких містяться дистрибутиви GParted live для встановлення на комп'ютерні системи з архітектурою AMD64. Вони можуть також бути корисними для вивчаючих двійкову та шістнадцятирічну системи числення та основи кібергігієни.

Література

[1] GParted: A free application for graphically managing disk device partitions. [Електронний ресурс]. – Режим доступу до ресурсу: https://gparted.org/

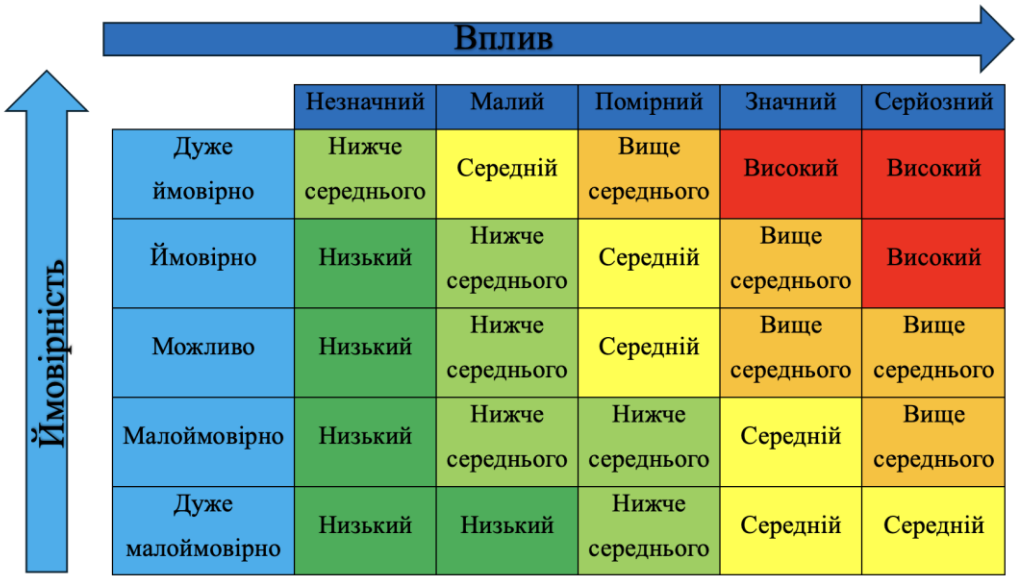
СТРАТЕГІЇ ПРІОРИТЕТИЗАЦІЇ КІБЕРРИЗИКІВ У СИСТЕМАХ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Шаповалов Б.Д., Коробейнікова Т.І.
E-mail: bohdan.shapovalov.mkbbi.2025@lpnu.ua, tetiana.i.korobeinikova@lpnu.ua
Львів, Національний університет «Львівська політехніка»

Управління кіберризиками є критично важливим у сучасних інформаційних системах, де загрози, такі як витоки даних, кібератаки та порушення конфіденційності, можуть мати катастрофічні наслідки. Пріоритизація кіберризиків дозволяє організаціям ефективно розподіляти ресурси для захисту від найбільш небезпечних загроз. У цій доповіді розглядаються стратегії пріоритизації кіберризиків, їх особливості та рекомендації щодо застосування в системах інформаційної безпеки [1].

Стратегії пріоритизації кіберризиків

Пріоритизація за важливістю. Цей метод використовує матриці ризиків для оцінки кіберризиків за ймовірністю їх настання та потенційним впливом на інформаційні системи. Наприклад, ризик фішингової атаки з високою ймовірністю та значним впливом (витік даних) матиме вищий пріоритет. Матриця ризиків будується з ймовірністю по осі Y та впливом по осі X (рис. 1).



	Незначний	Малий	Помірний	Значний	Серйозний
Дуже ймовірно	Нижче середнього	Середній	Вище середнього	Високий	Високий
Ймовірно	Низький	Нижче середнього	Середній	Вище середнього	Високий
Можливо	Низький	Нижче середнього	Середній	Вище середнього	Вище середнього
Малоймовірно	Низький	Нижче середнього	Нижче середнього	Середній	Вище середнього
Дуже малоймовірно	Низький	Низький	Нижче середнього	Середній	Середній

Рисунок 1 – Матриця кіберризиків за ймовірністю та впливом

Пріоритизація за вартістю. Організації часто оцінюють кіберризики на основі фінансових втрат, які вони можуть спричинити, наприклад, через відновлення після кібератаки або втрату клієнтської довіри. Альтернативно, пріоритетність може визначатися вартістю заходів із пом'якшення, таких як впровадження антивірусного програмного забезпечення. Однак ігнорування високовпливових ризиків заради економії може призвести до серйозних наслідків [2].

Пріоритизація за нормативно-правовим покаранням. Цей підхід фокусується на кіберризиках, пов'язаних із порушеннями регуляторних вимог, наприклад, GDPR чи локальних законів про захист даних. Організації можуть надавати пріоритет ризикам, які загрожують значними штрафами, навіть якщо їх ймовірність низька.

Пріоритизація за позицією. Пріоритизація за позицією враховує схильність організації до кіберризиків, її толерантність до них та поріг ризику. Наприклад, ризик, який може призвести до зупинки критичної інформаційної системи, перевищує поріг і потребує негайного реагування, тоді як менш критичні ризики можуть бути відкладені [3].

Пріоритизація за чутливістю. Чутливість кіберризиків визначається через аналіз їх впливу на операційну діяльність, наприклад, доступність серверів або цілісність даних. Діаграма невизначеності (рис. 2) ранжує ризики від найбільш до найменш невизначених, допомагаючи ідентифікувати критичні загрози, такі як DDoS-атаки.

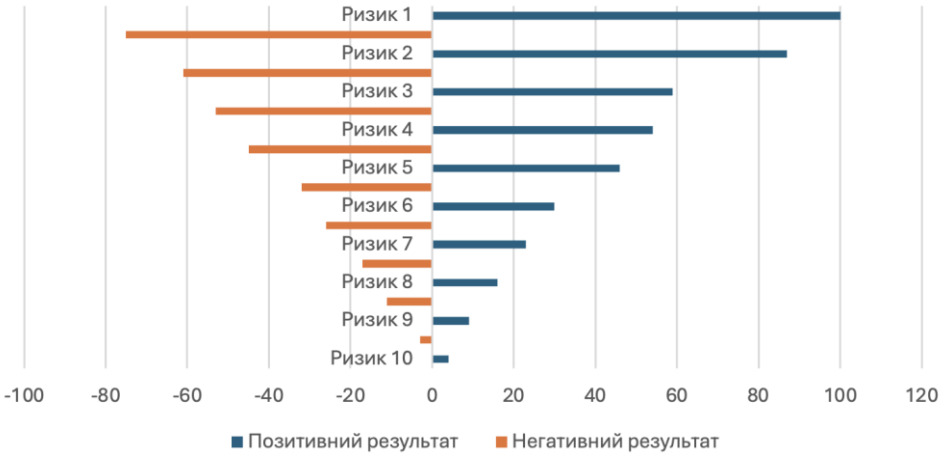
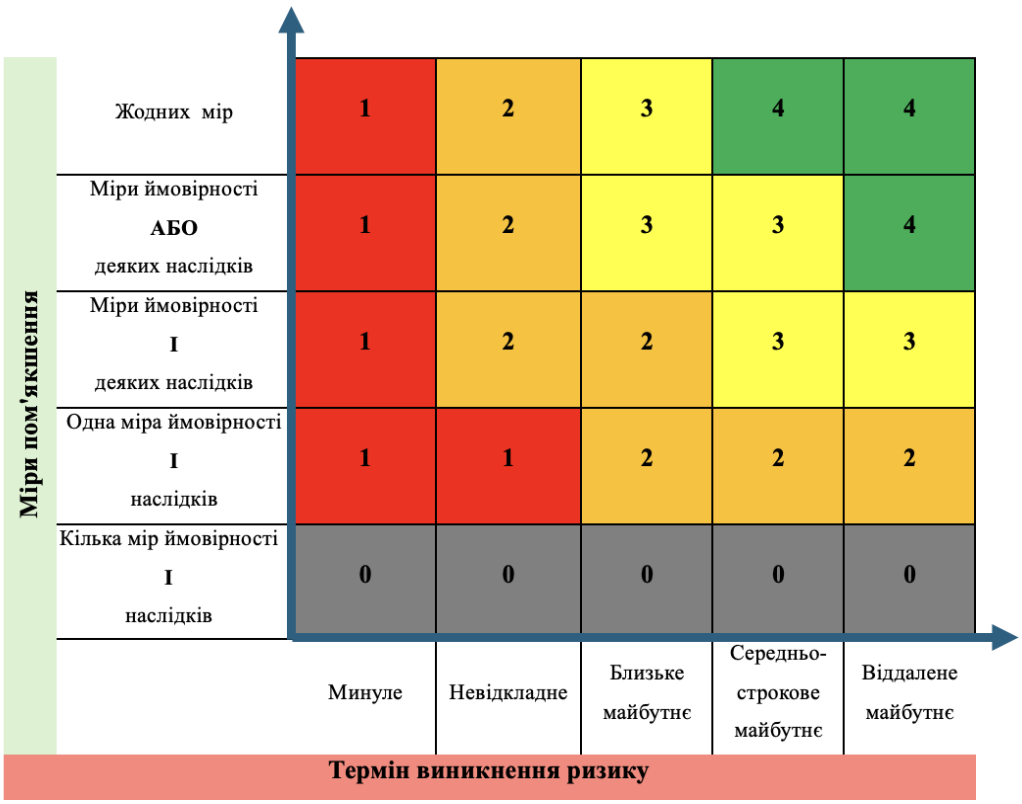


Рисунок 2 –Діаграма невизначеності кіберризиків

Пріоритизація за доступністю ресурсів. Коли ресурси для захисту від кіберризиків обмежені, пріоритет надається загрозам, які можна усунути з наявними засобами, наприклад, оновленням програмного забезпечення. Ризики, що потребують значних інвестицій, таких як впровадження SIEM-систем, можуть бути відкладені [4].

Пріоритизація за керованістю. Менш керовані кіберризики, такі як складні APT-атаки, потребують негайного реагування через їх високий вплив. Керовані ризики, наприклад, незначні вразливості в некритичних системах, можуть бути відкладені (рис. 3).



Керованість

0	1	2	3	4
Відсутня	Дуже низька	Низька	Помірна	Висока

Рисунок 3 – Матриця керування кіберризиками

Пріоритизація кіберризиків є ключовим елементом ефективного управління інформаційною безпекою в сучасних організаціях, оскільки дозволяє зосередити обмежені ресурси на найбільш критичних загрозах, що можуть завдати значної шкоди бізнес-процесам, репутації чи фінансовому стану. У сучасному цифровому середовищі, де кіберзагрози, такі як фішинг, програми-вимагачі, DDoS-атаки та витоки даних, постійно еволюціонують, правильне визначення пріоритетів допомагає організаціям не лише мінімізувати потенційні збитки, але й відповідати регуляторним вимогам, таким як GDPR, ISO/IEC 27001 чи NIST, а також підтримувати довіру клієнтів і партнерів [5].

Комбінований підхід до пріоритизації кіберризиків, який поєднує кілька стратегій, є оптимальним рішенням для комплексного управління. Використання матриць ризиків дозволяє візуалізувати ймовірність настання загрози та її потенційний вплив, що допомагає ідентифікувати критичні ризики, такі як атаки на критичну інфраструктуру, які потребують негайного реагування. Аналіз чутливості дає змогу оцінити, як зміна певних параметрів ризику (наприклад, рівня невизначеності чи вразливості системи) впливає на операційну

діяльність, що особливо важливо для виявлення ризиків, пов'язаних із новими технологіями, такими як хмарні сервіси чи Інтернет речей (IoT). Оцінка вартості враховує фінансові аспекти як потенційних збитків від кібератак, так і витрат на їх пом'якшення, дозволяючи організаціям ефективно розподіляти бюджет, наприклад, інвестуючи в антивірусне програмне забезпечення чи навчання персоналу. Оцінка доступності ресурсів забезпечує реалістичний підхід до управління ризиками, визначаючи, які заходи можна впровадити негайно, а які потребують додаткових інвестицій чи залучення спеціалізованих фахівців, наприклад, для впровадження систем SIEM.

Регулярне оновлення стратегій пріоритизації є критично важливим з огляду на швидку еволюцію кіберзагроз і зміну ландшафту інформаційної безпеки [6]. Нові види атак, такі як атаки на основі штучного інтелекту чи експлуатація вразливостей нульового дня, вимагають від організацій постійного моніторингу та адаптації.

Таким чином, комбінований підхід, який інтегрує різноманітні методи пріоритизації, разом із регулярним переглядом і оновленням стратегій, дозволяє організаціям не лише ефективно протистояти сучасним кіберзагрозам, але й підтримувати стійкість інформаційних систем, забезпечуючи їхню відповідність бізнес-цілям і регуляторним стандартам.

Література

[1] Information Security Investments: How to Prioritize? Proceedings of the 20th Brazilian Symposium on Information Systems. [Електронний ресурс] – Режим доступу до ресурсу: https://dl.acm.org/doi/abs/10.1145/3658321.3658363

[2] Costa I., Guarda T. Information System Security Risk Priority Number: A New Method for Evaluating and Prioritization Security Risk. [Електронний ресурс] – Режим доступу до ресурсу: https://link.springer.com/chapter/10.1007/978-981-16-7618-5_49

[3] Cybersecurity Risk Management: Frameworks, Plans, and Best Practices. Hyperproof resource guide. [Електронний ресурс] – Режим доступу до ресурсу: https://hyperproof.io/resource/cybersecurity-risk-management-process

[4] Шаповалов Б. Д. Управління ризиками: процес та стратегії реагування. Матеріали конференції «Кібербезпека в сучасному світі: актуальні виклики». [Електронний ресурс] – Режим доступу до ресурсу: https://hdl.handle.net/11300/28986

[5] Коробейнікова Т. І., Шаповалов Б. Д. Управління кіберризиками підприємства: класифікація та технологічні аспекти. Збірник тез доповідей МНПК (Рівне). [Електронний ресурс] – Режим доступу до ресурсу: https://www.economics.in.ua/2025/03/29-2.html

[6] Шаповалов Б. Д., Коробейнікова Т. І. Порівняльний аналіз міжнародних стандартів управління кіберризиками в корпоративному секторі. [Електронний ресурс] – Режим доступу до ресурсу: https://e-u.edu.ua/userfiles/files/135/2025-zbirnik_aktualni_pitannya_zabezpechennya_kiberbezpeki_ta_zahistu_informacii--2025.pdf

СИСТЕМА REAL-TIME МОНІТОРИНГУ КОРПОРАТИВНИХ ВИТОКІВ ДАНИХ ТА ВРАЗЛИВОСТЕЙ ПРОГРАМНИХ КОМПОНЕНТІВ

Швачка Д.І.
E-mail: d.shvachka@ukma.edu.ua
Київ, Національний університет «Києво-Могилянська академія»

У теперішні часи стрімка цифровізація охоплює бізнес-процеси усіх сфер діяльності, де дані тепер можна вважати найціннішими активами. В умовах сучасної кібервійни захист корпоративної інформації набуває критичного значення, оскільки периметр безпеки стає дедалі розмитішим. Стрімка інтеграція хмарних рішень та віддалених режимів роботи призвела до того, що традиційні методи захисту мережевого периметра поступово втрачають ефективність, оскільки вектори атак зміщуються в бік компрометації облікових записів користувачів шляхом, наприклад, фішингу або недбалим ставленням працівників компанії до кібергігієни. Це, своєю чергою, призводить до завантаження зараженого програмного забезпечення інфостилерами та "дроперами".

Багато компаній готуються до потенційних небезпек майбутнього, плануючи закупівлю різного роду антивірусних рішень чи EDR. В той час коли першим ешелоном захисту має виступати безпека всіх асетів (Asset Management & Security) та захист облікових даних співробітників.. Враховуючи статистику від IBM за 2025 рік, то найбільшу частку “зламів” (53%) [1] складають саме використання викрадених облікових даних та токенів з комп’ютерів співробітників організацій через фішинг та вбудоване вірусне ПЗ до завантажених пакетів програм не з офіційних джерел (наприклад, веб-сайту вендора). Викрадені дані одного з необачних співробітників відразу опиняються у зловмисників, які досить часто без розбору вивантажують величезну кількість викрадених облікових даних у так званий “combolist” та розміщують у вигляді архіву на хакерському форумі або Telegram-каналі. Реалії сьогодення такі, що час реагування на такі інциденти з боку команди безпеки і реагування має бути мінімальним, а то й миттєвим.

Аналогічне питання постає при необхідності реагування на проблеми безпеки supply chain розроблених програмних застосунків, де у використаних бібліотеках можуть бути виявлені критичні вразливості, а компанія про це не дізнається ще довгий час (що може стати критичним фактором). Це створює нагальну потребу в інструментах, здатних не просто виявляти атаки, а проактивно реагувати на появу нових вразливостей (CVE) шляхом розвідки загроз за допомогою впровадження Threat Intelligence систем, які працюють вкрай повільно (повідомлення про появу нової вразливості для пристроїв мережевого периметру, наприклад, може займати більше однієї доби. За цей час зловмисник може скористатися експлоїтом, в результаті якого отримає віддалений доступ або здампить/відредагує файли конфігурацій.

Так, дослідження, проведені у 2024 році на базі аналізу 27 мільярдів записів, доводять, що час між публікацією дампу даних та початком їх активного використання хакерами (Time-to-Exploit) стрімко скорочується [2]. Це вимагає від систем захисту реакції в режимі, наближеному до реального часу. Основним джерелом такої оперативної інформації, окрім Darknet, стають OSINT-канали: наприклад, Telegram (який перетворився на центр спілкування зловмисників та мережею поширення злитих облікових даних користувачів з усього світу) [3].

Існуючі Open Source рішення класу Threat Intelligence Platform (такі як MISP), хоч і є стандартом де-факто, але часто критикуються за монолітну архітектуру та складність інтерфейсу, що ускладнює їх розгортання для вузькоспеціалізованих задач моніторингу [4]. Більше того, такі системи не наповнюються інформацією автоматично, а лише вручну спеціалістами команди вендора (надавача послуг). Інші платформи, наприклад, OpenCTI, є простими прикладами візуалізацій на основі вхідних даних з TI-платформ [5]. Тобто, компанія втрачає досить багато часу на реагування в найпотрібніший момент, ще на етапі розвідки чи розгортання кампанії проти її активів з боку зловмисника. Найчастіше реакція вже відбувається пост-фактум, коли всередині периметра вже було виявлено інцидент безпеки.

Для вирішення проблеми проактивного реагування команди захисту пропонується застосування Event-Driven Architecture та ймовірнісних структур даних, що дозволяє обійти обмеження класичних монолітних TIP-систем.

По-перше, замість монолітної системи пропонується використання мікросервісної декомпозиції та асинхронності системи. Наприклад, завдяки патерну Database per Service можна ізолювати процеси збору та аналізу даних [6], а використання асинхронних I/O операцій та патерну Publish/Subscribe для маршрутизації між модулями забезпечує високу пропускну здатність при обробці великої кількості concurrent-з'єднань та знижує зв’язність системи [7].

По-друге, великою проблемою при обробці даних з різних джерел (форуми в DarkNet, окремі канали, чати чи діалоги в месенджерах, знайдені посилання на архіви з обліковими даними і т.д.) є неструктурованість і відсутня будь-яка шаблонність даних. Для приведення їх до єдиного структурованого вигляду пропонується двоетапний алгоритм обробки:

первинна фільтрація “шуму” за допомогою NLP-інструментів та використання регулярних виразів для видобутку записів по типу “email:password” [3].

Для вирішення питання інформативності та достовірної інформації щодо CVE у стеку програмних компонентів будь-якої розробки пропонується використання стандарту OSV (Open Source Vulnerability). Це JSON-схема, що містить уніфіковані дані про рівень небезпеки (Severity), рейтинг за вектором CVSS 2.0/3.0/3.1/4.0, а також інформацію про екосистему та пакет, в якому знайдена вразливість, діапазони вразливих версій. [8]. Використання формату JSON для парсингу даних з ресурсу OSV дозволяє додавати нові поля, наприклад - метрику EPSS, тощо.

Варто також враховувати факт, що реєстр вразливостей CVSS ведеться з 1999 року і наразі налічує майже 313 тисяч записів, та кількість викрадених облікових даних щороку збільшується кратно [9]. Для збільшення швидкості реагування запропонована система використовує інвертовані індекси [10]. Інформаційна система також враховує можливе дублювання даних. Цю проблему ефективно вирішують імовірнісні структури даних, наприклад, фільтр Блума (Bloom Filter), забезпечуючи перевірку наявності запису з мінімальними витратами пам’яті [11].

Застосування вищезгаданих підходів дозволить досягти зменшення часу виявлення загроз та швидшої можливості реагування на них, не навантажуючи інфраструктуру моніторингу так, як роблять це “важкі” аналоги. Запропонований підхід зосереджується на динамічній обробці неструктурованих OSINT-даних у режимі, близькому до реального часу, забезпечуючи швидкий Time-to-Detection.

Література

[1] IBM. Cost of a Data Breach Report 2025: The AI Oversight Gap [Електронний ресурс]. – Режим доступу: https://www.ibm.com/downloads/documents/us-en/131cf87b20b31c91

[2] Thomas K. [et al.]. Beyond the Leak: Analyzing the Real-World Exploitation of Stolen Credentials Using Honeypots [Електронний ресурс]. – Режим доступу: https://www.mdpi.com/1424-8220/24/12/3676

[3] Zhang Y. [et al.]. CTI Dataset Construction from Telegram [Електронний ресурс]. – Режим доступу: https://arxiv.org/abs/2512.21380

[4] Wagner C., Dulaunoy A., Wagener G. MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/309419134

[5] AI4COLLAB: An AI-based Threat Information Sharing Platform [Електронний ресурс]. – Режим доступу: https://www.researchgate.net/publication/384308471

[6] Richardson C. Microservices Patterns: With examples in Java [Електронний ресурс]. – Режим доступу: https://microservices.io/patterns

[7] Hohpe G., Woolf B. Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions [Електронний ресурс]. – Режим доступу: https://github.com/ivanarandac/Books/blob/master/Enterprise%20Integration%20Patterns%20-%20Designing%2C%20Building%20And%20Deploying%20Messaging.pdf

[8] OSV Schema: Open Source Vulnerability format specification [Електронний ресурс]. – Режим доступу: https://ossf.github.io/osv-schema/

[9] CVE: History of CVE Program [Електронний ресурс]. – Режим доступу: https://www.cve.org/about/history

[10] Gormley C., Tong Z. Elasticsearch: The Definitive Guide [Електронний ресурс]. – Режим доступу: https://www.elastic.co/guide/en/elasticsearch/guide/current/

[11] Carlson J. Redis in Action [Електронний ресурс]. – Режим доступу: https://redislabs.com/ebook/redis-in-action/

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АНАЛІЗУ ЛОГІВ ДЛЯ ВИЯВЛЕННЯ КІБЕРЗАГРОЗ

Шерстнюк А.В.
Керівник: Лимаренко В.В.
E-mail: sherstnuk1@gmail.com
Харків, Харківський національний економічний університет імені Семена Кузнеця

Умови сучасного кіберпростору характеризуються постійним зростанням кількості та складності атак на веб-ресурси. Більшість інцидентів безпеки залишають цифрові сліди у вигляді логів, проте їх ручний аналіз потребує значного часу та кваліфікації. Тому актуальним є створення програмних засобів, здатних здійснювати автоматизований моніторинг і інтелектуальний аналіз журналів подій у реальному часі з метою оперативного виявлення кіберзагроз[1].

Метою роботи є розробка прототипу системи аналізу логів, яка може бути інтегрована з будь-яким веб-застосунком та забезпечувати миттєве інформування відповідальних осіб про підозрілу активність.

Основні функціональні можливості розроблюваної системи:

– створення тестового веб-сайту з модулем авторизації та розподілом ролей (адміністратор, користувач) для моделювання реального середовища;

– реалізація окремого програмного модуля аналізу логів, що працює незалежно від веб-системи та може бути підключений до різних ресурсів;

– збір і обробка подій у режимі реального часу: спроби входу, помилки автентифікації, зміни привілеїв, запити до бази даних[2];

– автоматичне виявлення аномалій, зокрема атак грубої сили, SQL-ін’єкцій та інших підозрілих дій;

– надсилання push-повідомлень користувачу з найвищими привілеями з коротким описом інциденту та рівнем його критичності.

Технологічна реалізація. Веб-частина прототипу розробляється з використанням сучасних засобів побудови веб-застосунків (Flask/Django або Node.js). Модуль аналізу логів реалізується як окремий сервіс мовою Python із застосуванням бібліотек для обробки подій та регулярних виразів. Для взаємодії між сайтом і системою аналізу використовується API та механізми веб-сокетів. Під час тестування моделюються атаки грубого перебору паролів та SQL-ін’єкції, що дозволяє оцінити ефективність виявлення загроз[3].

Переваги запропонованого рішення:

– можливість оперативного реагування на інциденти завдяки миттєвим сповіщенням;

– скорочення часу на аналіз журналів подій;

– універсальність підключення до різних веб-систем;

– автоматизація рутинних процесів моніторингу безпеки.

Перспективи розвитку. У подальшому планується впровадження елементів машинного навчання для більш точного виявлення аномальної поведінки, розширення переліку підтримуваних типів логів, інтеграція з SIEM-системами, а також реалізація гнучкої системи правил для різних рівнів критичності інцидентів. Передбачається можливість використання розробки у корпоративних мережах як компонента комплексної системи кіберзахисту.

Таким чином, створення програмного забезпечення аналізу логів у реальному часі дозволяє підвищити рівень захищеності веб-ресурсів, забезпечити швидке виявлення кібератак та зменшити ризики для інформаційних систем.

Література

[1] Scarfone K., Mell P. Guide to Computer Security Log Management. – NIST Special Publication 800-92. – National Institute of Standards and Technology, 2006. – 72 p.

[2] Behl A., Behl K. Cybersecurity and Cyberwar: What Everyone Needs to Know. – Oxford University Press, 2020. – 304 p.

[3] Chuvakin A., Schmidt K., Phillips C. Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management. – Syngress, 2013. – 448 p.

АВТОМАТИЗОВАНЕ ПРОГНОЗУВАННЯ STORY POINTS НА ОСНОВІ СЕМАНТИЧНИХ ЕМБЕДІНГІВ ТЕКСТОВИХ ОПИСІВ ЗАДАЧ AGILE-ПРОЄКТІВ

Шкода В.М.
Керівник: Бондаренко Д.О.
E-mail: vladyslav.shkoda@hneu.net
Харків, Харківський національний економічний університет імені Семена Кузнеця

У сучасних Agile-проєктах оцінювання зусиль виконує ключову роль у процесах планування спринтів. Метрика Story Points (SP), що відображає відносну складність задач, традиційно визначається експертним шляхом під час командних обговорень. У віддалених командах цей процес ускладнюється – через асинхронність взаємодії, різний рівень залученості учасників та високі витрати часу на проведення зустрічей. Це зумовлює потребу в автоматизованих інструментах, які здатні підтримувати або частково замінювати ручне оцінювання складності задач [1].

Одним із перспективних напрямів є застосування методів обробки природної мови та машинного навчання для прогнозування SP на основі текстових описів задач. У більшості Agile-проєктів саме текстові атрибути – title та description – містять основну інформацію про сутність, складність та вимоги до задачі [2]. Тому доцільним є перетворення цих полів у щільні векторні представлення за допомогою попередньо навченої мовної моделі типу GPT [3]. Семантичні ембедінги забезпечують збереження контексту та значень, роблячи текст придатним для подальшої обробки.

У межах дослідження текстові описи задач формуються шляхом об’єднання title і description у єдину контекстну структуру, після чого перетворюються на ембедінги за допомогою мовної моделі типу GPT. Експерименти виконано на публічно доступному датасеті TAWOS, який містить задачі із відкритих Agile-проєктів (Jira). Датасет охоплює різні предметні області, характеризується високою варіативністю значень Story Points та наявністю шуму в експертних оцінках, що робить його репрезентативним для задач оцінювання зусиль [4]. Для кожної задачі доступні текстові поля title та description а також окремі структуровані атрибути задачі (priority, type, status). Попередній аналіз показав, що ці додаткові атрибути не підвищують точність прогнозування, тому у фінальну модель включаються лише семантичні векторні представлення тексту.

На основі отриманих ембедінгів виконано порівняння кількох класів моделей машинного навчання для задачі регресії, серед яких представлені як класичні методи, так і ансамблеві моделі. До аналізу включено ElasticNet, Random Forest, ExtraTrees, CatBoost, градієнтні бустинги (XGBoost, LightGBM), метод K-Nearest Neighbors (KNN), а також багатошаровий перцептрон (MLP).

Для оцінювання якості застосовуються стандартні метрики, рекомендовані в дослідженнях із прогнозування зусиль: MAE, MdAE та SA. Застосування цих метрик забезпечує стійку інтерпретацію результатів і дозволяє порівнювати алгоритми (табл.1) [1, 2].

Порівняльний аналіз показує, що найкращі результати забезпечують KNN, CatBoost та ExtraTrees, які демонструють вищу точність та стабільність оцінювання порівняно з іншими методами. Це підтверджує, що семантичні ембедінги, сформовані мовною моделлю GPT, можуть слугувати ефективною основою для автоматизованого прогнозування Story Points.

Застосування автоматизованих моделей оцінювання зусиль має практичну цінність для віддалених Agile-команд. Такий підхід дозволяє скоротити тривалість планувальних зустрічей, зменшити когнітивне навантаження на учасників та підвищити узгодженість оцінок.

Модель може виконувати роль інструменту попереднього прогнозування, допомагаючи виявляти задачі зі складною семантикою або потенційними ризиками ще до проведення експертного обговорення.

Таблиця 1 – Якість базових моделей прогнозування Story Points

Метод	MAE	MdAE	SA
Catboost	1.7305	1.3144	0.3102
ExtraTrees	1.7824	1.4695	0.2849
Random Forest	1.8597	1.4586	0.2555
KNN	1.7114	1.0363	0.3343
XGBoost	1.9926	1.4301	0.2159
LightGBM	2.3811	1.4886	0.0488
ElasticNet	2.0594	1.4776	0.1844
MLP	1.9324	1.2266	0.2678

Література

[1] Choetkiertikul M., Dam H. Khanh., Tran T., Pham T., Ghose A., Menzies T. A deep learning model for estimating story points // IEEE Transactions on Software Engineering. 2016. DOI: https://doi.org/10.1109/TSE.2018.2792473

[2] Yalçıner B., Dinçer K., Karaçor AG., Efe MÖ. Enhancing Agile Story Point Estimation: Integrating Deep Learning, Machine Learning, and Natural Language Processing with SBERT and Gradient Boosted Trees // Applied Sciences. 2024. DOI: https://doi.org/10.3390/app14167305

[3] OpenAI Platform. Vector embeddings. – [Електронний ресурс] – Режим доступу до ресурсу: https://platform.openai.com/docs/guides/embeddings.

[4] Tawosi V., Al-Subaihin A., Moussa R., Sarro F. A versatile dataset of agile open source software projects // Proceedings of the 19th International Conference on Mining Software Repositories (MSR '22). Association for Computing Machinery. 2022. P. 707–711. DOI: https://doi.org/10.1145/3524842.3528029

ЕВОЛЮЦІЯ МОДЕЛЕЙ ВЕБЗАГРОЗ В УМОВАХ ІНТЕГРАЦІЇ ІНТЕЛЕКТУАЛЬНИХ ПОМІЧНИКІВ

Якимчук Є.А., Марченко Я.В.
E-mail: yevhenii.iakymchuk@npp.kai.edu.ua, yaroslav.marchenko@npp.kai.edu.ua
Київ, Національний державний університет «Київський авіаційний інститут»

Більшість сучасних інформаційних систем працюють у вебсередовищі. На даний момент ці системи швидко розвиваються. Причиною є розвиток хмарних технологій, мікросервісної архітектури та широкої інтеграції систем штучного інтелекту. Одним із ключових елементів цієї трансформації є впровадження інтелектуальних помічників в ці системи. Вони використовуються для інтерпретації користувацьких запитів, автоматизації прийняття рішень та доступу до її внутрішніх ресурсів. Ці зміни серйозно впливають на кібербезпеку і змушують нас по-новому подивитися на традиційні вебзагрози

Традиційні вебзагрози з'явилися ще тоді, коли веб-додатки були орієнтовані на сервер. Користувач в таких системах безпосередньо взаємодіяв з серверною частиною за допомогою HTML-форм та HTTP-запитів. В основному ризиками таких систем є: некоректна валідація введених даних або її відсутність, порушення механізмів автентифікації та авторизації, а також відсутність належного контролю доступу. Ці вразливості представлені у вигляді загальноприйнятих стандартів і рекомендацій, зокрема запропонованих OWASP [1].

Головна риса класичних вебзагроз - вони зосереджені на низькорівневих помилках, які допускають при розробці. Їх можна виявити за допомогою автоматизованих статичних та

динамічних інструментів аналізу. З часом ефективність більшості традиційних атак зменшилася завдяки широкому впровадженню фреймворків. Вони мають вбудовані механізми безпеки, тестування та параметризованих запитів до бази даних [2]. У результаті класичні вебвразливості залишаються актуальними, однак вони перестають бути ключовим джерелом експлуатації зловмисниками.

Слід зауважити, що моделі веб-загроз змінюються через еволюцію архітектури вебдодатків. Сучасні системи все частіше будуються з розділенням на фронтенд та бекенд частини з використанням мікросервісної архітектури, зокрема відокремлюють клієнтські інтерфейси від серверної логіки. У таких умовах класичні точки входу атак, що пов'язані з вебінтерфейсами, поступово втрачають лідуючі позиції. Тоді ж як більше цінується впровадження контролю доступу на рівні бізнес-логіки та внутрішньої взаємодії сервісів [6].

Інтеграція інтелектуальних помічників дає новий поштовх в трансформації вебзагроз. У сучасних інформаційних системах інтелектуальні помічники виконують роль логічних посередників між користувачем та внутрішніми компонентами системи. Вони здійснюють модифікацію, інтерпретації введених даних користувачів, агрегують інформацію з різних джерел та ініціюють рішення щодо виконання бізнес-операцій.

Інтелектуальні помічники, на відміну від звичайних вебсервісів, аналізують не тільки структуру запиту, а й його контекст. Це призводить до зміни класичної моделі довіри в системі, оскільки рішення щодо доступу дедалі частіше приймаються на основі намірів користувача, які інтерпретовані інтелектуальним помічником [3].

Сучасні дослідження штучного інтелекту підтверджують, що вебсистеми, які використовують інтелектуальні помічники у своїй роботі, формують нові поверхні атаки, які не покриваються класичними моделями загроз [4, 5]. Основні ризики пов'язані з логічними помилками, галюцинаціями, надмірними повноваженнями та порушеннями контексту доступу. Такі загрози набагато важче виявити за допомогою традиційних інструментів безпеки, оскільки вони не беруть до уваги контекст виконання операції.

У результаті впровадження інтелектуальних помічників відбувається зміщення пріоритетів вебзагроз від технічних атак, наприклад ін'єкцій, до контекстних вразливостей. Це зумовлює необхідність адаптації існуючих підходів до моделювання загроз і управління ризиками з урахування специфіки роботи інформаційних вебсистем, які покладаються на штучний інтелект у своїх процесах [7]. Отже, класичні вебзагрози все ще є важливою частиною сучасної кібербезпеки. Інтелектуальні помічники змінюють моделі взаємодії користувача з вебсистемами та формують нові класи ризиків, орієнтовані на логіку та контекст, що змушує до перегляду традиційних моделей кіберзагроз.

Література

[1] OWASP Top 10 Web Application Security Risks [Електронний ресурс] - OWASP Foundation, 2023: https://owasp.org/API-Security/editions/2023/en/0x00-header/

[2] OWASP Testing Guide v4.2. [Електронний ресурс] - OWASP Foundation, 2022: https://owasp.org/www-project-web-security-testing-guide/v42/

[3] NIST. AI Risk Management Framework (AI RMF 1.0). [Електронний ресурс] - National Institute of Standards and Technology, 2023: https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf

[4] ENISA. Threat Landscape for Artificial Intelligence [Електронний ресурс] - European Union Agency for Cybersecurity, 2023: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

[5] MITRE. ATLAS: Adversarial Threat Landscape for Artificial Intelligence [Електронний ресурс]. - MITRE Corporation: https://atlas.mitre.org/

[6] OWASP. API Security Top 10. [Електронний ресурс]. - OWASP Foundation, 2023: https://owasp.org/www-chapter-bangkok/slides/2023/2023-03-31_OWASP-API.pdf

[7] ISO/IEC 23894:2023 [Електронний ресурс] - Artificial Intelligence: https://cdn.standards.iteh.ai/samples/77304/cb803ee4e9624430a5db177459158b24/ISO-IEC-23894-2023.pdf

Матеріали XVII-ої Міжнародної науково-практичної конференції
«FREE AND OPEN SOURCE SOFTWARE»

Харківський національний економічний університет імені Семена Кузнеця

Відповідальний за випуск: Старкова О.В.

Редактор: Міхєєв І.А.

Затверджено засіданням кафедри кібербезпеки та інформаційних технологій
ХНЕУ імені С. Кузнеця
протокол № 10 від «13» лютого 2026 р.

Видавець і виготовлювач – ХНЕУ імені С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А
Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.