МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені В.Н.КАРАЗІНА
V.N. KARAZIN KHARKIV NATIONAL UNIVERSITY

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА
COMPUTER SCIENCE AND CYBERSECURITY
(CS&CS)

№ 1(25) 2024
Issue 1(25) 2024

Заснований 2015 року

Міжнародний електронний науково-теоретичний журнал
International electronic scientific journal

В журналі публікуються наукові статті з теоретичних і науково-технічних проблем, що пов'язані зі створенням ефективних засобів комп'ютерних інформаційно-комунікаційних систем та питань захисту інформації, на основі передових математичних методів, інформаційних технологій і технічних засобів.

Журнал виходить кожні півроку.

Схвалено до розміщення в мережі Інтернет Вченою радою Харківського національного університету імені В.Н. Каразіна (29.08.2024 р., Протокол № 16).

DOI (Онлайн): **10.26565/2519-2310-2024-1**.

Редакція:
Харківський національний університет імені В.Н. Каразіна
майдан Свободи, 6, офіс 315а, Харків, 61022, Україна (*Північний корпус університету, 3 поверх*)

Електронна пошта: cscsjournal@karazin.ua
Веб-сторінка: http://periodicals.karazin.ua/cscs (Open Journal System)

Автори опублікованих матеріалів несуть повну відповідальність за достовірність наведених фактів, власних імен тощо. Опубліковані статті пройшли внутрішнє та зовнішнє рецензування.

The journal publishes research articles on theoretical, scientific and technical problems of effective facilities development for computer information-communication systems and information security questions based, on advanced mathematical methods, information technologies and technical means.

The journal is published every six months.

Approved for placement on the Internet by Academic Council of the Karazin Kharkiv National University (August 29, 2024, Protocol No.16).

The journal has Digital Object Identifier: **10.26565/2519-2310-2024-1** (Online).

Editorial office:
V.N. Karazin Kharkiv National University
Svobody Sq., 6, office 315a, Kharkiv, 61022, Ukraine (*North building of University, 3th floor*)

E-mail: cscsjournal@karazin.ua
Web-page: http://periodicals.karazin.ua/cscs (Open Journal System)

The authors of the published materials are solely responsible for the selection, accuracy of the facts, proper names, etc. Published articles have been internally and externally peer reviewed.

ЗМІСТ

CONTENTS

ISSN 2519-2310 (Online)

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА, 2024, № 1 (25)

COMPUTER SCIENCE AND CYBERSECURITY, 2024, Iss. 1 (25)

DOI: https://doi.org/10.26565/2519-2310-2024-1-01
УДК 004.056.5

УЗАГАЛЬНЕННЯ НАПРЯМІВ ФІЛЬТРАЦІЇ DNS ТРАФІКУ ЯК СКЛАДОВОЇ БЕЗПЕКИ СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

Данило Чепель[1], студент магістратури спеціальності «Комп'ютерні системи та мережі», кафедра захисту інформаційних систем та технологій, e-mail: dan4epel@gmail.com, ORCID: https://orcid.org/0000-0003-3308-424X

Сергій Малахов[1], доктор філософії, старший науковий співробітник, кафедра комп'ютерних наук, e-mail: malakhov@karazin.ua, ORCID: https://orcid.org/0000-0001-8826-1616

[1]*Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, 61022, Україна*

Рукопис надійшов 21 квітня 2024 р. Отримано після рецензування 21 травня 2024 р. Прийнято 23 червня 2024 р.

Анотація: В роботі проведено аналіз джерел, стосовно методів і технологій DNS (Domain Name System) фільтрації трафіку. Визначені п'ять основних напрямків, які активно використовуються для підвищення безпеки на рівні DNS. Усі розглянуті технології пропонують підвищення якості DNS фільтрації. Підкреслено, що одночасне комбінування різних підходів може підвищити загальний рівень безпеки. Узагальнення результатів досліджень з проблематики безпеки DNS трафіку, вказує на існування певних проблем у якості використовуваних каналів розвідки загроз. Саме тому впровадження AI та LM технологій, повинно посилити «глибину» екстракції корисної інформації про актуальні загрози. Звернено увагу на те, що розгляд питань ІБ, слід вести виключно у розрізі недопущення диспаритету можливостей штучного інтелекту (AI) на користь протиборчої сторони (тобто зловмисників). Практично це означає, що майбутні системи фільтрації DNS, повинні широко впроваджувати останні напрацювання на рівні стеку VR, AI, LM та DL технологій. Це особливо важливо в межах протидії алгоритмам генерації доменів (DGA - Domain Generation Algorithm) та поширення ботнетів. Наголошено на специфіку питань забезпечення консенсусу безпеки та продуктивності діючих інформаційно-комунікаційних систем (ІКС) при впровадженні в них інструментів шифрування DNS. В якості основної проблеми, пов'язаної з шифруванням DNS трафіку, підкреслена можливість його використання з боку зловмисників, як інструменту приховування їх деструктивної діяльності (фішинг, спам та інші).

Ключові слова: *DNS, DGA, RPZ, інформаційна безпека, загрози безпеки, фільтрація трафіку, ботнет*

Як цитувати: Чепель Д., Малахов С.. Узагальнення напрямів фільтрації DNS трафіку як складової безпеки сучасних інформаційних систем. *Комп'ютерні науки та кібербезпека.* 2024; № 1(25): С. 6–21. https://doi.org/10.26565/2519-2310-2024-1-01

In cites: Chepel D., Malakhov S. (2024). Summary of DNS traffic filtering trends as a component of modern information systems security. *Computer Science and Cybersecurity.* 1(25): 6–21. https://doi.org/10.26565/2519-2310-2024-1-01 (in Ukrainian)

1. Вступ

Технологія зіставлення доменних імен - *Domain Name System* (*DNS*) із їх числовими IP-адресами, є найважливішим механізмом сучасного Інтернет, виступаючи в якості умовного «посередника», який перетворює «зручні» для користувачів доменні імена в їх IP-адреси [1]. Система із розгалужених *DNS* серверів не тільки спрощує процес навігації в Інтернеті але й забезпечує ефективне і точне з'єднання, як між користувачами, так і «місцями призначення» їх пошукових запитів (інформаційними ресурсами) в Інтернеті. Нажаль, сучасні мережеві зловмисники знаходять способи використання *DNS* служб для реалізації різних варіантів атак, причому як окремо, так і в якості складового елементу при реалізації багатоходових - інтегрованих атак [2-3]. Тому, на поточний момент фільтрація *DNS* трафіку є невід'ємною складовою заходів з інформаційної безпеки (ІБ) для переважної більшості сучасних інформаційно-комунікаційних систем (ІКС) Інструменти управління й фільтрації *DNS* забезпечують контроль доступу до веб-ресурсів, захист від шкідливого програмного забезпечення (ПЗ) і надає можливість гнучкого впровадження потрібних політик безпеки на рівні мережі (*для моделі OSI*).

Метою роботи є стислий огляд відомих напрямів використання *DNS* технології, що відіграють важливу роль у забезпеченні ІБ сучасних ІКС.

Ключовими питаннями, що розглядаються, є: - канали розвідки загроз ІБ; - шифрування *DNS* трафіку; - синтез та управління зонами політик реагування на основі *DNS*; - особливості алгоритмів генерації фіктивних доменів та виявлення активності ботнетів (серверів управління ботнетами).

2. Основна частина

2.1. Основні вектори застосувань технології DNS при вирішенні питань ІБ

Служба/Система DNS. Ця служба відіграє ключову роль у функціонуванні та масштабованості мережі Інтернет, оскільки майже кожен інший протокол залежить від вирішення домену *DNS* для своєї коректної роботи. *DNS* є одним з небагатьох протоколів, що складають умовне «ядро» Інтернету. В загальному випадку, *DNS* використовується переважно для перетворення «зручних» для читання людиною доменних імен у їх цифрові IP-адреси. Для пошуку потрібного мережевого домену клієнт/користувач надсилає *DNS*-запит до відповідного рекурсивного *DNS* серверу, який, зазвичай, надається поточним інтернет провайдером та має можливості розпізнавання і кешування (тимчасового зберігання) доменних імен. У випадку, коли використовуваний сервер доменних імен не має у своєму кеші необхідних відомостей, то він звертається до кількох інших – «зовнішніх» *DNS* серверів, які зберігають розподілену базу даних доменних імен та їх відповідні IP-адреси. Таким чином, у пошуках потрібного мережевого домену, кожний умовний *DNS* запит транслюється через певні сегменти ієрархічної мережі із довірених серверів імен, доки не знайде потрібну відповідь (*зіставлення*) та не надішле її клієнту (*користувачеві*). Отримавши потрібну IP-адресу, шукач необхідного інформаційного ресурсу може використовувати її для підключення до хосту призначення [1].

DNS-фільтрація. *DNS*-фільтрація мережевого трафіку є елементом проактивної стратегії кіберзахисту, що діє на рівні системи доменних імен для контролю та управління доступом до Інтернету в мережі. Використовуючи *DNS*-фільтрацію, організації можуть впроваджувати потрібні політики безпеки, які обмежують доступ до певних веб-сайтів та/чи категорій контенту, які вважаються такими, що не відповідають корпоративним вимогам. Цей механізм фільтрації працює шляхом перехоплення *DNS*-запитів і їх наступного порівняння із попередньо визначеними реєстрами дозволених чи заблокованих доменних імен та IP-адрес. У разі наявності відповідного ресурсу в стоп листі, *DNS*-фільтр блокує таке з'єднання.

DNS-фільтрація надає організаціям і окремим мережевим користувачам кілька помітних переваг, зокрема підвищення поточного рівня кібербезпеки шляхом запобігання відвідуванню потенційно зловмисних та/чи небажаних інформаційних ресурсів (наприклад, як функція «батьківський контроль»). Така фільтрація джерел контенту допомагає забезпечити дотримання відомчих нормативних вимог, обмежуючі доступ до сайтів, які не відповідають політиці безпеки компанії/установи. Крім того, *DNS*-фільтрація сприяє підвищенню продуктивності персоналу, обмежуючи його доступ до інформаційних ресурсів, що не пов’язані з виконанням їх функціональних обов’язків, та зменшує навантаження на корпоративну мережу, шляхом адміністрування небажаного ресурсоємного трафіку [4]. Також слід підкреслити, що делегування функцій *DNS*-фільтрації на довірені зовнішні ресурси/сервіси, дозволяє в значній мірі позбутися необхідності регулярного перевірки й оновлення стану відповідних реєстрів доступу силами корпоративних фахівців та забезпечити більшу функціональну стійкість корпоративної інфраструктури в разі відповідних атак.

Канали розвідки загроз. Канали розвідки загроз є важливим компонентом для реалізації сучасних стратегій ІБ (*див. рис.1*). Їх використання, в режимі реального часу, забезпечує корпоративних фахівців з ІБ відомостями про нові загрози, вразливості і діяльність кіберзловмисників [2-3]. Вони функціонують шляхом постійного збору та узагальнення даних з різних джерел для виявлення потенційних кіберзагроз і вразливостей [5-6]. Зазвичай ці канали містять індикатори компрометації, наприклад, такі як: - «шкідливі» IP-адреси, доменні імена, хеші файлів та типові шаблони/сценарії підозрілої поведінки [3, 7].

Інтеграція каналів розвідки загроз в інструменти та корпоративні системи ІБ дозволяє організаціям автоматично блокувати відомі зловмисні чи скомпрометовані джерела та ранжувати пріоритетність сповіщень системи безпеки на основі відомостей стосовно актуальності інформації та серйозності наслідків. Це дає змогу приймати обґрунтовані рішення, покращувати час реагування на інциденти та покращити загальну систему ІБ. Використовуючи канали розвідки поточних загроз, сучасні організації можуть покращити показник поінформованості про стан актуальних загроз та завчасно вживати профілактичних заходів для усунення відповідних вразливостей власних ІКС ще до того, як ними спробують скористатися зловмисники.

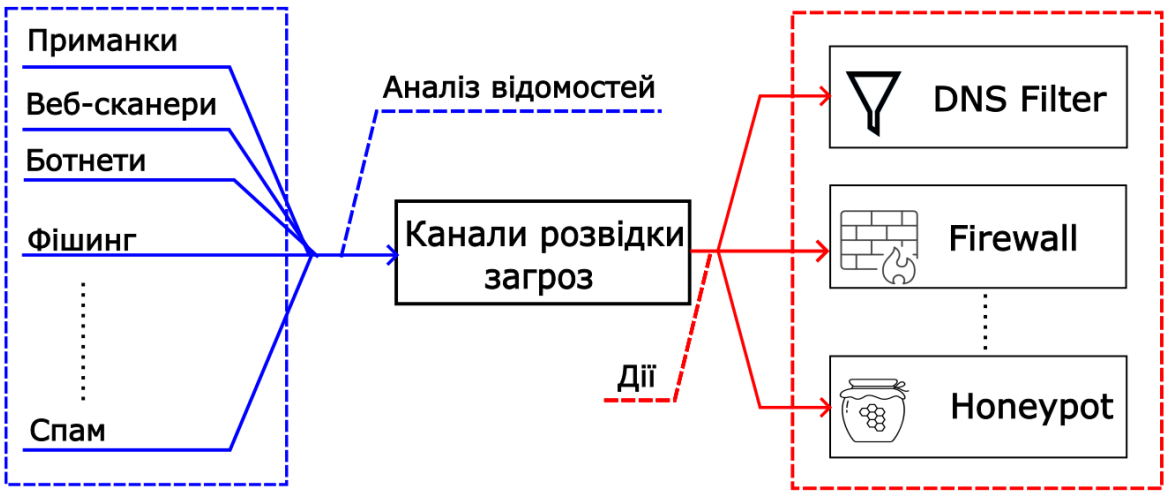


Рис. 1 - Сутність використання каналів розвідки загроз ІБ
Fig. 1 - The essence of using threat intelligence feeds

Постійно співвідносячи дані внутрішнього моніторингу безпеки із зовнішніми каналами розвідки загроз, сучасні організації можуть виявляти приховані загрози та превентивно їм протидіяти [7-9]. Такий проактивний підхід до кібербезпеки допомагає випереджати кіберзлочинців, мінімізувати вплив інцидентів безпеки та ефективно захищати конфіденційні дані й критично важливі активи [10].

Сервери управління ботнетом. Сервер управління ботнет системи являє собою централізований (*ведучій*) сервер, що використовується зловмисником для дистанційного управління скомпрометованими мережами та/чи пристроями, відомих, як боти або бот-мережі. Командно-контрольні сервери ботнетів (*див. рис.2*) відіграють ключову роль в організації зловмисних дій, таких як проведення розподілених атак на відмову в обслуговуванні (*DDoS-атак, в т.ч. DNS amplification attack*), *DNS* та *NTP Spoofing*, розсилання спаму, здійснення шахрайства, поширення зловмисного ПЗ тощо [3, 11-12].

Аналізуючи поточні властивості мережевого трафіку і таким чином ідентифікуючи роботу відповідних серверів, засоби безпеки можуть протидіяти впливу ботнетів, своєчасно парируючи їх зловмисну дію. Переваги виявлення такого трафіку передбачають зменшення ризику атак з боку ботнетів, захист мережевих пристроїв від їх компрометації з використанням експлойтів та підтримку безпечного корпоративного середовища [5-7, 13].

Шифрування DNS трафіку. Шифрування *DNS* трафіку, зокрема *DNS-over-TLS* (*DoT*), *DNS-over-HTTPS* (*DoH*) та *DNS-over-QUIC* (*DoQ*), відіграє важливу роль у підвищенні безпеки та ефективності контролю *DNS* запитів [14]. Цей процес перетворює інформацію *DNS* трафіку у зашифрований формат, що гарантує можливість декодування інформації лише довіреними сторонами, такими як *DNS*-клієнт і сервери *DNS* провайдера.

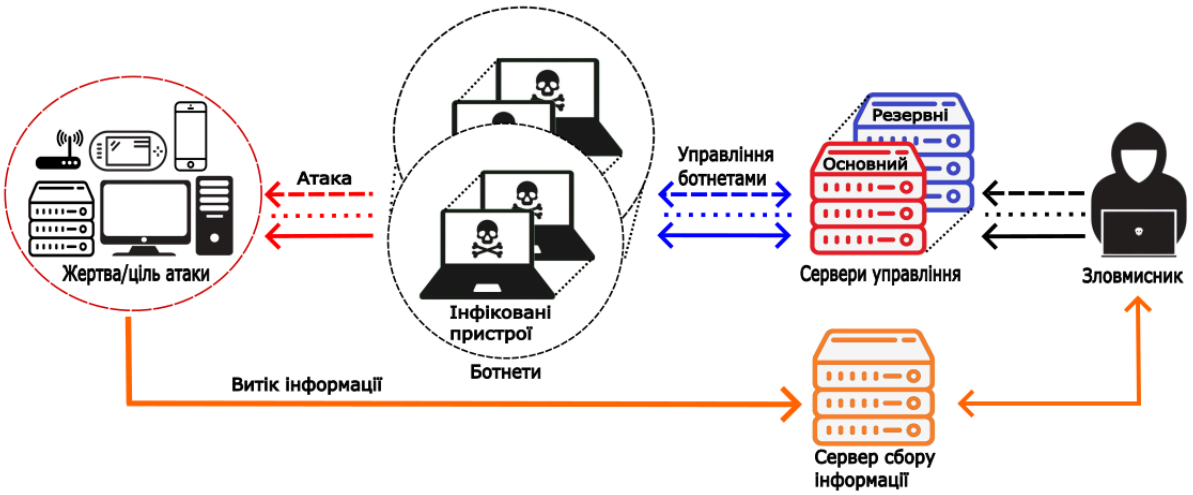


Рис. 2 - Архітектура ботнета (авторська розробка)
Fig. 2 - Architecture of a botnet (author's development)

Шифрування зменшує ризик атак на *DNS* трафік і маніпуляції з даними цих запитів, знижуючи ймовірність неавторизованих змін у *DNS* записах. Крім того, існуючі протоколи шифрування *DNS* підвищують рівень цілісності й конфіденційності відповідного трафіку, протидіючи несанкціонованому моніторингу та відстеженню відомостей *DNS*-запитів. Процес шифрування *DNS* трафіку, в цілому, сприяє підвищенню конфіденційності та анонімності онлайн дій користувачів, однак на цьому шляху є і певні складнощі. Наприклад, складність відстеження вмісту зашифрованого *DoH* трафіку, з боку адміну безпеки, свідчить про те, що

аналіз мережевих *DoH* з'єднань з метою виявлення потенційного шкідливого трафіку (*наприклад з боку зовнішніх командно-контрольних серверів умовних бот-систем [13]*), є актуальним напрямом для його подальшого аналізу [14].

Зони політики реагування. *Response Policy Zone* (*RPZ*) – це механізм, який використовується при фільтрації *DNS* для визначення діючих локальних політик у стандартизованому форматі та завантаження/оновлення політик із інших, «зовнішніх» джерел. Використовуючи *RPZ*, сучасні організації можуть оперативно контролювати, які запити можуть обробляти їхні сервери *DNS*, а які ні. Це дозволяє оперативно блокувати шкідливі домени й ресурси або виконувати інші дії на основі попередньо встановлених політик безпеки. Політики безпеки формалізуються у вигляді файлів складених для відповідних зон *DNS* (*див. рис.3*), котрі формуються за визначеними корпоративними критеріями та/чи діючими нормами мережевої поведінки [15] для основних груп користувачів інформаційних послуг компанії/установи. Цей процес в рівній мірі відноситься, як для власного персоналу, так і користувачів основних послуг компанії.

Впровадження концепції *RPZ* полегшує, масштабування, використання й оновлення актуальних зон безпеки. Парадигма *RPZ* передбачає їх спільне застосування між декількома *DNS*-серверами (через передачу зон), що дозволяє оперативно транслювати дані політики за допомогою звичайних протоколів *DNS* [14]. Політики *RPZ* складаються з відповідних поведінкових (логічних) тригерів та дій. Тригери визначають, коли потрібно застосувати ту чи іншу політику (*тобто декларують певні мережеві умови/обставини*), а дії, відповідно, вказують, які саме процедури слід виконати в даному разі.

Впровадження *RPZ* забезпечує уніфікацію, масштабованість і гнучкість управління політиками *DNS*. Це підвищує безпеку і цілісність налаштувань фільтрації *DNS* трафіку, а також дозволяє організаціям визначати й застосовувати власні політики для адміністрування сервісу *DNS* [16].

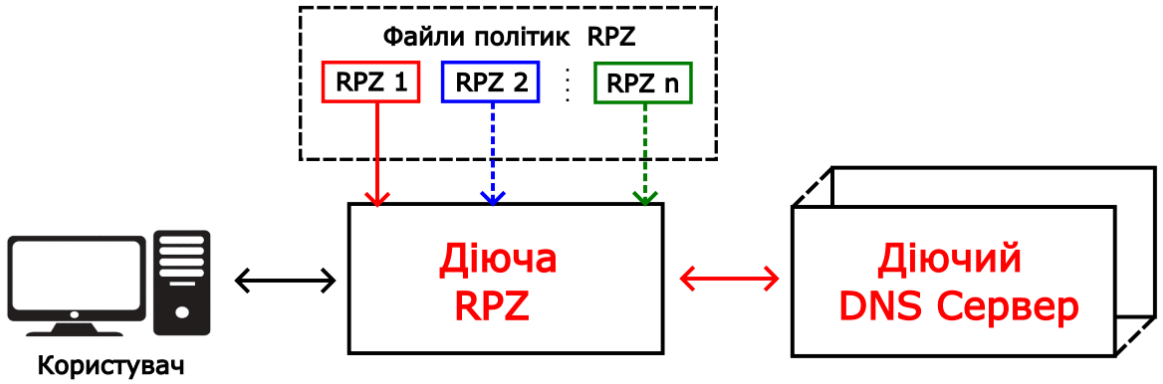


Рис. 3 - Сутність використання RPZ (авторська розробка)
Fig. 3 - The essence of using RPZ (author's development)

Алгоритми генерації доменів. *Domain Generation Algorithm* (*DGA*) – це алгоритми, які використовуються кіберзловмисниками для генерування великої кількості доменних імен, що слугують умовними точками «зустрічі» в мережі між скомпрометованими пристроями та серверами управління зловмисника. В цьому випадку (*див. рис.4*), скомпрометовані/атаковані комп'ютери й інше мережеве устаткування намагаються встановити зв'язок (*DNS spoofing*) з цими згенерованими доменними іменами, щоб отримати від них «оновлення» чи інші команди з

боку атакуючої сторони. Ці алгоритми створені для періодичного генерування значної кількості доменних імен, що помітно ускладнює боротьбу з ботнетами. Таким чином *DGA* використовуються для приховування фактичного розташування командно-контрольних серверів ботнетів серед великої кількості потенційно легітимних адрес, що значно ускладнює пошук та блокування (згодом і видалення) цих доменів. В широкому сенсі *DGA*, за своєю парадигмою, є своєрідним аналогом атак типу *DNS amplification* [12].

Виявлення злочинних доменних імен, що синтезовані *DGA*, є складної задачею через їх великі обсяги відповідної генерації та випадковий характер цих процесів. Однак, залучення технологій штучного інтелекту і машинного навчання (*AI/LM*), та впровадження методів глибокого навчання, потенційно демонструють високі показники в ідентифікації злочинних *DGA* [17].

2.2. Бліц-огляд робіт, щодо практичних реалізацій механізмів DNS-фільтрації

Спираючись на аналіз ряду відомих робіт, проведемо узагальнення основних відомостей, стосовно особливостей взаємозв'язку технології *DNS* та питань безпеки сучасних ІКС. Результати такого огляду систематизовано у відповідності до запропонованої в п.п. 2.1 послідовності розгляду для найбільш поширених напрямів застосувань інструментів та методів *DNS*-фільтрації.

Канали розвідки загроз. В роботі [18] розглянуто проблематику «пасивних» *DNS*-потоків для отримання актуальної інформації про потенційні загрози безпеки. Автор роботи фокусує увагу на ідентифікації підозрілих і зловмисних доменів, виявленню зловживань *DNS* та їх аномалій, підкреслюючи важливість перехресної перевірки даних з іншими джерелами інформації про кіберзагрози. Акцентовано необхідність ретельного аналізу *DNS*-трафіку для виявлення різноманітних кіберзлочинних дій, таких як ботнети, спам і фішинг. Крім того, дослідження автора зосереджене на розробці та впровадженні масштабованої методики «пасивного» аналізу *DNS*, яка забезпечує своєчасне виявлення актуальних загроз ІБ.

Рис. 4 - Інфографіка порядку дій зловмисника при використанні DGA (авторська розробка)
Fig. 4 - Infographic of the attacker's actions during the use of DGA (author's development)

Проблематика визначення «якості» каналів розвідки загроз безпеки є вкрай актуальним та доволі складним завданням, оскільки їх важко порівнювати між собою. Так наприклад, у роботі [19] її автором розглядається методологія вимірювання надійності та якості безкоштовних каналів розвідки про кіберзагрози з відкритим кодом. У цьому дослідженні дані аналізуються за допомогою різних метрик, наприклад: – обсягом даних, швидкістю змін, географічним розподілом та збігами між різними каналами.

Підкреслено, що різні канали зосереджені на різних питаннях, що ускладнює прямі порівняння. Автором стверджується, що остаточного методу визначення найкращого каналу поки не існує та можна лише зробити певні висновки, стосовно їх якості. Дослідження підкреслює складність оцінки безкоштовних каналів розвідки загроз та потребу в більш надійних та достовірних інструментах [19].

Робота [20] також розглядає проблематику оцінки якості каналів розвідки і пропонує всебічний аналіз каналів розвідки загроз. Автори цієї роботи звертають увагу на необхідність та важливість емпіричних оцінок у цій галузі. Дослідження представляє набір показників для оцінки каналів розвідки загроз, включаючи такі як: – обсяг, диференціальний внесок, унікальний внесок, затримку, точність, охоплення та ін.. Дослідивши 47 каналів із різними IP-адресами та 8 каналів із хешами файлів зловмисного ПЗ, автори роботи виявили значні обмеження в охопленні та точності наявних даних розвідки загроз. Це вказує на високий ступінь унікальності та помилкових узагальнень (прогнозів) для різних каналів. В цілому, ця робота надає обґрунтовану методологію для оцінки каналів розвідки загроз та підкреслює важливість постійного розвитку методів їх оцінки [20].

Виявлення активності ботнетів. Дослідницька група авторів роботи [21] пропонує огляд джерел, який включає в себе вичерпний аналіз методів виявлення ботнетів на основі аналізу *DNS* трафіку. Робота звертає увагу на проблему ботнетів та пов'язаних з ними загроз ІБ. Автори класифікують та порівнюють кілька підходів, що використовують різні властивості *DNS* трафіку, такі як: – зміна домену, лексичні характеристики доменних імен і шаблони в запитах та *DNS* відповідях. В роботі аналізуються сильні й слабкі сторони для відповідних підходів. Зазначено, що методи виявлення на основі *DNS* мають переваги перед методами на основі хосту, оскільки вони можуть надати більш широке уявлення про мережеву активність і уникнути деяких обмежень, пов'язаних з розгортанням систем моніторингу в мережі. У роботі також обговорюються обмеження існуючих підходів, такі як відсутність стандартних метрик і показників продуктивності та підкреслено потреба в нових методологіях виявлення для протидії поширенню загроз ботнетів [21].

В роботі [22] розглядаються різні методології аналізу поведінки ботів та ботнетів, включаючи статистичний аналіз і вимірювання трафіку та наголошується на важливість комплексного підходу до виявлення ботнетів. На думку авторів для ряду попередніх робіт була притаманна деяка невизначеність, насамперед в частині, що підкреслює складність пошуку всіх типів ботнетів через складність моделі їх поведінки. Саме тому дана робота пропонує новий механізм виявлення ботнетів, заснований на моніторингу трафіку *DNS* для виявлення групової активності розподілених ботів, усуваючи обмеження попередніх підходів і забезпечуючи більш надійний метод для виявлення різних варіацій ботнетів [22].

Робота [23] містить огляд існуючих методів виявлення ботнетів та пов’язаних з ними обмежень. Авторами роботи висвітлюються проблеми виявлення нових різновидів ботнетів та звертається увага на необхідності впровадження методів, що не базуються на перевірці корисного навантаження пакетів, оскільки вони могли би працювати із зашифрованими мережевими протоколами [14]. У роботі обговорюється обмеження існуючих методів виявлення при роботі з новими атаками ботнетів. Також підкреслено переваги аналізу поведінки трафіку порівняно з аналізом корисного навантаження пакетів, зокрема можливість працювати із

зашифрованим трафіком та менший вплив на продуктивність мережі. У дослідженні запропонована модель виявлення, окреслено параметри її експериментальної оцінки та надано результати для запропонованого детектора, котрі свідчать про його здатність виявляти ботнет-атаки з високою точністю [23].

Шифрування DNS. Робота [24] надає ґрунтовний огляд поточного спектру можливостей шифрування *DNS* [14], зосереджуючись на стандартних методах: – *DoT*, *DoH* та *DoQ*. Автори аналізують статус прийняття цих методів, їх продуктивність, переваги та проблеми безпеки. Основним фокусом статті є зловживання шифруванням *DNS* для командно-контрольних комунікацій (*див. рис.2*) і каналів викрадення/витоку даних, що створює певні труднощі у виявленні та боротьбі з відповідною діяльністю. Також обговорюються методи аналізу зашифрованого *DNS*-трафіку для профілювання дій користувачів з метою виявлення зловмисної та/чи нештатної мережевої активності. Авторами роботи сформульовано напрями майбутніх досліджень, стосовно підвищення продуктивності та безпеки шифрування *DNS* [24].

Автори роботи [25] провели детальний аналіз розгортання і використання протоколів *DNS* шифрування, зосереджуючись на *DoT* та *DoH*. Дослідження охоплює порівняльну оцінку різних протоколів *DNS* шифрування та проблеми безпеки, що пов’язані з цими протоколами, зокрема оцінку доступності та продуктивності, а також порівняння обсягів трафіку між традиційними й зашифрованими *DNS* запитами. Автори дослідження підкреслюють, що хоча якість послуг постачальників *DNS* загалом є задовільна, однак, деякі служби мають неправильні конфігурації, що потребує уваги. Висновки висвітлюють поточний стан галузі телекомунікацій до широкомасштабного впровадження зашифрованих *DNS* протоколів [25].

Стаття [26] присвячена дослідженню шифрування *DNS* запитів для захисту конфіденційності користувачів від атак, що передбачають аналіз трафіку. У роботі вивчаються питання ефективності аналізу *DNS* трафіку для виявлення моделей веб-активності користувачів через зашифрований *DNS*, з особливим акцентом на протокол *DoH* [14]. Пропонуючи новий набір функцій, розроблений спеціально для аналізу зашифрованого *DNS*-трафіку, автори демонструють «успішні» атаки з високою точністю, підкреслюючи існуючі обмеження поточних засобів захисту. Автори досліджень оцінюють рівень захисту, який забезпечують *DoH* і *DoT*, акцентуючи, що *DoT* демонструє кращій рівень безпеки. Підкреслено, що потенціал фільтрації трафіку на основі *DNS*, навіть у сценаріях із зашифрованим *DNS*, надає цінну інформацію про вразливості і засоби захисту в рамках зашифрованого *DNS* [26].

Зони політики реагування (*RPZ*). Робота [27] надає широкий аналіз проблематики зон політики реагування *DNS*. Автор детально розглядає історію, розвиток та особливості реалізації *RPZ* в ПЗ сервера імен *BIND*, а також приклади їх (політик) розгортання в реальному середовищі. Згідно з дослідженням, впровадження зон політики реагування ефективно блокує спроби *DNS* запитів з небезпечними доменами, захищаючи клієнтські системи без втрати їхньої продуктивності. Автором роботи представлено набір інструментів для аналізу журнальних даних *RPZ*, який сприяє завчасної ідентифікації потенційно скомпрометованих систем та небажаної мережевої поведінки користувачів. У роботі спрогнозовано подальші напрямки розвитку *RPZ*, включаючи залучення комерційних постачальників відповідних послуг та вдосконалення механізмів захисту від фішингових атак [3, 27-28].

Дослідження роботи [29] присвячено питанням виявлення і блокування аномальних вихідних *DNS*-запитів, що пов’язані із функціонуванням ботнетів. Автори пропонують систему на основі політик, яка використовує зони політики реагування *DNS* для вдосконалення свого попереднього підходу, котрий базувався на базі даних *MySQL*, основним недоліком якого була мережева затримка. Система що розглядається використовує програмно-визначену мережу для контролю мережевого трафіку та аналізу *DNS*-запитів на відповідність політикам, збереженим у відповідних *RPZ*, ефективно зменшуючи затримку і поліпшуючи виявлення, та блокування

шкідливого *DNS*-трафіку. Робота містить попередню оцінку продуктивності і функціональності системи, яка підтверджує її ефективність у локальному мережевому середовищі на основі програмно-визначеної мережі. Автори планують продовжити оцінку цієї системи в умовах роботи реальної мережі та вивчення методів збереження конфіденційності *DNS* трафіку [29].

Виявлення алгоритмів генерації доменів. Робота [30] пропонує всебічний огляд проблематики у напряму виявлення алгоритмів генерації доменів. Автори вказують на важливість тестів, стандартизованих метрик і методів виділення ознак для підвищення надійності та відтворюваності експериментів у дослідженнях *DGA*. Досліджуючи різні *DGA* та їх альтернативи проведено розгляд стратегій, котрі використовуються зловмисниками, та потребу в більш складних методах виявлення. Запропонована авторами методологія застосовує ймовірнісні підходи для ефективного виявлення *DGA* на основі списку слів. Робота пропонує аналіз поточного стану виявлення *DGA* та окреслює можливі майбутні напрями досліджень для протидії інструментам створення шкідливих доменів.

Авторський колектив роботи [31] представляє власний підхід до ідентифікації алгоритмів генерації доменів, шляхом використання методів глибокого навчання (*DL - Deep Learning*). Дослідження зосереджено на вирішенні труднощів виявлення *DGA*, які генерують домени шляхом псевдовипадкового об’єднання словникових термінів. Використовуючи контекстно-залежне вбудовування слів та «простий» повнозв’язний класифікатор, автори роботи демонструють ефективність свого підходу до класифікації доменів. Використання попередньо підготовлених слів, мінімальні обсяги вихідних даних та коротка тривалість термінів навчання, відрізняють цей підхід від існуючих методів. Дослідники підкреслюють оригінальність своєї методики, яка не потребує розробки функцій вручну та вивчення списків слів *DGA* [31], що підтверджує її потенціал для умов реального застосування.

У роботі [32] досліджуються особливості виявлення зловмисних доменних імен генерованих *DGA*, за допомогою реалізації різних архітектур *DL*. Дослідження охоплює використання згорткової нейронної мережі (*CNN*), рекурентної нейронної мережі (*RNN*), довготривалої короткочасної пам’яті (*LSTM*) та інших моделей. Автори досліджують ефективність різних підходів *DL* для точного визначення шкідливих доменів, порівнюючи показники продуктивності різних архітектур моделей глибокого навчання. Запропонований фреймворк [32] демонструє високу ефективність у виявленні доменних імен, що синтезовані *DGA* та може бути використаний для блокування ботів від зовнішніх зв’язків і порушення каналів управління з командно-контрольними серверами ботнетів (*див. рис.2*).

2.3. Узагальнення поточних застосувань та концептуальних напрацювань в рамках проблематики фільтрації DNS

Як вже було зазначено вище, канали розвідки загроз (*рис.1*) є важливими джерелами критично важливих даних про поточні кіберзагрози, які допомагають завчасно виявляти і оперативно реагувати на нові загрози ІБ. Однак при цьому, актуальною проблемою залишається оцінка якості цих каналів. Дослідження за цим напрямом пропонують нові метрики та методики для оцінювання каналів розвідки, в т.ч. на основі аналізу даних *DNS*-трафіку. Крім того, наголошується на високому ступені унікальності та надмірності помилкових спрацювань у різних каналах розвідки. Узагальнюючи стан напрацювань та поглядів дослідників на подальші можливості більш релевантної екстракції даних з каналів розвідки загроз, можна зробити висновок, про необхідність покращення ефективності механізмів оцінки наявних каналів розвідки та концентрації зусиль на зменшенні частки помилкових спрацювань (*прогнозів*).

Виявлення ознак мережевої активності командних серверів ботнетів (*рис.2*) може надати важливу інформацію для систем безпеки з фільтрацією *DNS*-трафіку. Впровадження такого

механізму протидії дозволяє відокремити інфіковані засоби у мережі від відповідних командних центрів або завчасно захистити мережу від атак ботнетів, наприклад за рахунок нав'язування мережевої «гри» з використанням мережевих пасток [6-7]. Поточний стан досліджень тримає у фокусі питання виявлення каналів управління ботнетів, що в свою чергу, зумовлює постійне вдосконалення методик аналізу їх поведінки шляхом комплексного моніторингу *DNS*-трафіку, включаючи «роботу» з зашифрованим трафіком [14]. Отримані напрацювання декларують досить високу точність виявлення для відомих різновидів ботнетів, однак дослідники звертають увагу на обмеження існуючих підходів [30-32].

Стосовно напряму шифрування *DNS*, слід чітко розділяти дві взаємопов'язані іпостасі подальшого розвитку подій. Так, з одного боку шифрування *DNS*, може сприяти підвищенню якості фільтрації *DNS*, теоретично забезпечуючи при цьому більш безпечний канал для *DNS*-запитів, запобігаючи підробці та/чи втручанню в трафік зловмисників. Однак, з іншого боку, із впровадженням шифрування *DNS* виникає проблема ускладнення процесу виявлення зловмисного трафіку [14], зокрема трафіку командно-контрольних центрів ботнетів (*див. рис.2*). Більш того, у аналізованих роботах є приклади «успішних» атак на шифрування *DNS* (наприклад у [26]), що ставить під сумнів ефективність шифрування, як інструменту із забезпечення конфіденційності мережевої активності користувачів. Автори робіт пропонують власні методи аналізу шифрованого трафіку для профілювання активності користувачів з метою виявлення зловмисних дій та вказують на перевагу *DNS-через-TLS* у ефективності захисту. В якості основної проблеми, у межах напряму шифрування *DNS*, слід зазначити можливість використання зловмисниками технологій шифрування цього трафіку з метою ускладнення виявлення їх деструктивної діяльності (*фішинг, спам, DDos, DNSA тощо*).

Практика використання зон політик реагування (*RPZ*) незмінно продовжує привертати увагу дослідників [27, 29] до цього напряму та є ефективним інструментом управління безпекою сучасних ІКС на рівні служби *DNS*, що забезпечує завчасне виявлення й блокування небезпечних доменів.

Виявлення ознак роботи алгоритмів генерації доменів (*DGA*), що виступають головним інструментом для створення сукупності умовних «точок зустрічі» кіберзловмисників з їх власними командно-контрольними серверами ботнетів (*див. рис.2*), залишається вкрай важливим завданням. Стрімке й одночасне поширення Інтернету речей (*IoT*), технологій віртуалізації (*VR*), штучного інтелекту і машинного навчання (*AI/LM*), лише додатково підкреслюють вектор на невідкладність розробок, що адекватні рівню та темпу появи нових загроз ІБ. Вочевидь, що проблематика оперативної ідентифікації, блокування генерації і поширення злочинних доменів [30-32] ще до того, як вони зможуть завдати шкоди, є безумовно пріоритетним напрямом для подальших досліджень фахівців з ІБ.

Останні дослідження [22-24, 30-32] в галузі протидії впливу *DGA* розглядають тести, метрики та нові методи виявлення відповідних алгоритмів, котрі базуються на ймовірнісному аналізі і техніках *LM/DL*. Узагальнюючи цей досвід можна стверджувати, що пріоритетним напрямом з протидії *DGA*, є інтеграція нових - більш вдосконалених методик у єдину інтегровану систему фільтрації *DNS*, що втілює парадигму проактивного захисту та базується на останніх напрацюваннях в галузі штучного інтелекту і машинного навчання. Впровадження цих технологій розширює можливості поведінкового аналізу мережевої активності [6-7] та вдосконалює евристичні алгоритми, що здатні завчасно й оперативно виявляти і класифікувати підозрілі домені, тим самим усуваючи основні передумови для поширення активності ботнетів. Тож найбільш ймовірним напрямом досліджень в галузі проблематики фільтрації *DNS*, можна вважати синтез нових структур та логіки роботи інтегрованих підсистем *DNS* безпеки (*див. рис.5*), що впроваджують останні напрацювання на рівні стеку *VR-AI-LM-DL* технологій. Вочевидь, що саме таке поєднання, може дати паритетну відповідь на загрозу масштабного

впровадження *AI* в алгоритми зловмисної генерації доменів. Іншими словами, де-факто потрібно вести розмову про необхідність ефективної протидії впливу *AI* як основного елемента протиборчої сторони '[33].

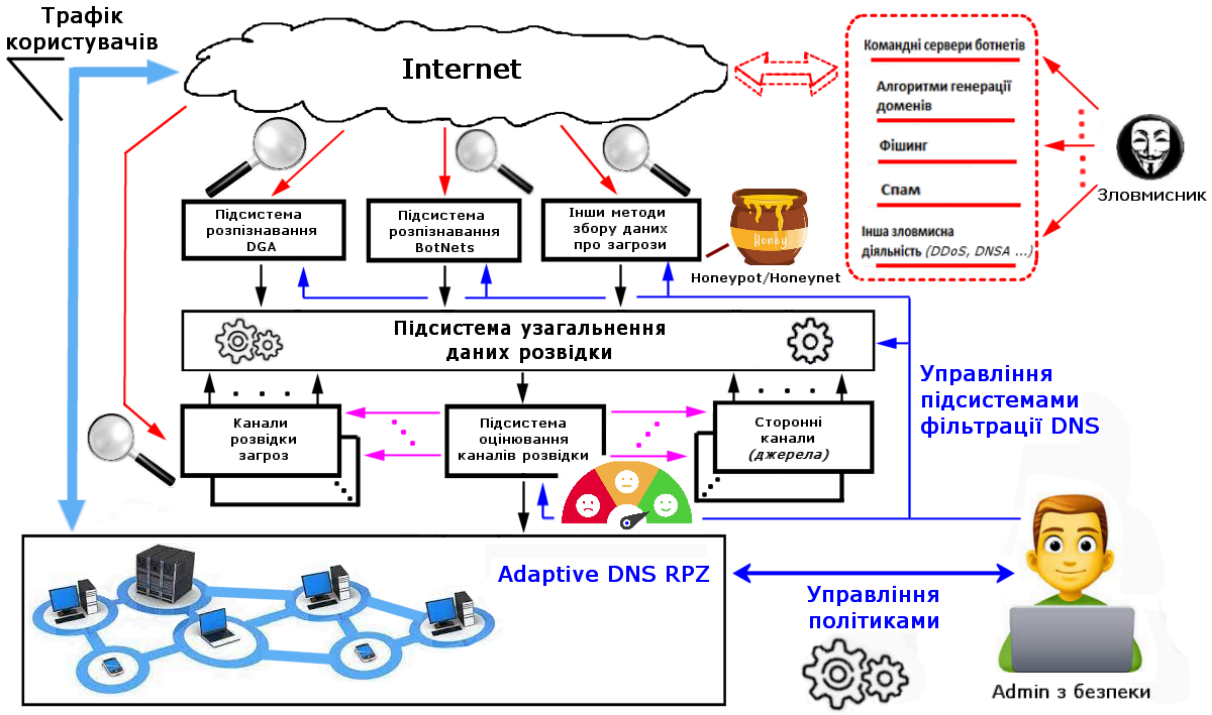


Рис. 5 - Структурний концепт інтегрованої системи DNS безпеки [33]
Fig.5 - Structural concept of an integrated DNS security system [33]

Для ефективної протидії новим загрозам таки системи мають впроваджувати механізми комплексної обробки та агрегування даних у канали розвідки загроз. Ці механізми повинні забезпечувати постійний моніторинг і оновлення інформації, надаючи актуальні дані для аналізу загроз та прийняття коригуючи рішень (*наприклад, стосовно зміни поточної RPZ на рис.5*). Дані із системи агрегування інформації (*узагальнення даних розвідки*) повинні оперативно використовуватися для покращення якості оцінки наявної сукупності основних та сторонніх каналів розвідки загроз. Відповідним чином, мають бути модифіковані адаптивні зони політик реагування, які використовують дані з каналів розвідки загроз для автоматичного застосування заходів захисту та/чи коригування налаштувань параметрів функціонування окремих підсистем інтегрованої системи *DNS* безпеки.

Узагальнюючи результати різних дослідницьких груп [21-27, 29-32], слід відзначити важливість робіт у напрямах оптимізації порядку взаємодії різних компонентів системи розпізнавання і збору даних та вдосконалення механізмів оновлення й активації політик реагування (*RPZ*). Зокрема це передбачає розробку інтерфейсів обміну даними, уніфікацію форматів даних та забезпечення безпеки передачі інформації (*особливо в разі реалізації розподіленої та/чи хмарної системи безпеки*).

3. Висновки

1. Новий, прийдешній постквантовий технологічний уклад, виводить довічний процес протистояння засобів атаки і захисту на принципово новий рівень, де можливості людини, як

адміністратора систем безпеки, повинні бути корінним образом переосмислені й модифіковані [33]. Причина такого стану справ очевидна: – масштаби, темп і сутність технологічного розвитку сучасного інформаційного суспільства, котрі перетнули межу фізіологічних можливостей сучасної людини. У цьому сенсі, сфера ІБ є одним із флагманських движків, де базовими аргументами назрілих змін є:

- масштабність наслідків;
- багатопотоковість даних;
- взаємопов'язаність процесів;
- складність формалізації завдань;
- висока швидкість перебігу подій і явищ;
- зростання обсягів оброблюваної інформації;
- розподіленість інфраструктур та основних акторів;
- необхідність протидіяти впливу AI як протиборчій стороні тощо.

2. Додатковими факторами, котрі виступають потужним прискорювачем процесів створення сучасних ІКС є стрімке поширення технологій *IoT* та *mesh*-мереж (*що вже вийшли за рівень Scatternet*). Конвергенція цих напрямів з можливостями *AI* та *LM*, створює нові горизонти задач особливо в частині протидії алгоритмам генерації доменів та ботнетів на стороні кіберзлочинців.

3. Розгляд питань з забезпечення ІБ слід вести виключно у розрізі недопущення диспаритету можливостей *AI* на користь протиборчої сторони. На практиці це означає, що майбутні системи захисту повинні широко впроваджувати останні новації на рівні стеку *VR*, *AI*, *LM* та *DL* технологій.

4. Особливої уваги потребують питання забезпечення балансу безпеки та продуктивності (швидкодії) перспективних ІКС при впроваджені в них інструментів шифрування *DNS*. Розробка нових методів шифрування, які одночасно забезпечують високий рівень захисту, мінімізують затримки у передачі даних і при цьому не ускладнюють процес виявлення зловмисного трафіку, є важливим напрямом для подальших досліджень.

5. Аналіз результатів досліджень з проблематики аналізу і фільтрації *DNS* трафіку, вказує на існування певних проблем у якості використовуваних каналів розвідки загроз безпеки, де впровадження технологій *AI* та *LM* повинно прискорити й одночасно посилити «глибину» екстракції потрібної, корисної інформації про актуальні загрози [33].

Конфлікт інтересів

Автори повідомляють про відсутність конфлікту інтересів.

Список літератури:

1. What is DNS? Вилучено з URL: https://www.cloudflare.com/learning/dns/what-is-dns/
2. Погоріла, К., Лєсная, Ю., Богданова, Є., & Малахов, С. (2022). Соціальний інжиніринг, як фактор реалізації інсайдерських загроз. *Scientific Collection «InterConf», (111): with the Proceedings of the 1st International Scientific and Practical Conference «Scientific Community: Interdisciplinary Research» (June 6-8, 2022)*. Boston, USA; pp. 494-501. Вилучено з https://archive.interconf.center/index.php/conference-proceeding/article/view/645/666
3. Лєсная, Ю., Малахов, С. Узагальнення основних передумов реалізації фішингових атак. *Proceedings of the XVII International Scientific and Practical Conference*. Ankara, Turkey. 2023. Pp.453-457. Available at: https://doi.org/10.46299/ISG.2023.1.17
4. What is DNS filtering? Вилучено з URL: https://www.cloudflare.com/learning/access-management/what-is-dns-filtering/

5. Яремчук, К., Воскобойников, Д., & Мелкозьорова, О. (2022). Сучасні загрози та способи забезпечення безпеки веб-застосунків. *Комп'ютерні науки та кібербезпека,* (2), 28-34. https://doi.org/10.26565/2519-2310-2022-2-03
6. Богданова, Є., Чорна, Т., & Малахов, С. (2022). Огляд поточного стану загроз, що обумовлені впливом експлойтів. *Комп'ютерні науки та кібербезпека,* (2), 35-40. https://doi.org/10.26565/2519-2310-2022-2-04
7. Кохановська, Т., Нарежний, О., & Дьяченко, О. (2020). Дослідження можливостей технології Honeypot. *Комп'ютерні науки та кібербезпека,* 1(1), 33-42. https://doi.org/10.26565/2519-2310-2020-1-03
8. Михайленко Д., Нємцев М. Особливості технології мережевих пасток як інструменту активного захисту та аналізу дій атакуючої сторони. *Proceedings of the XXI International Scientific and Practical Conference.* Melbourne, Australia. 2023. Pp. 483-487. Available at: https://doi.org/10.46299/ISG.2023.1.21
9. Січкар, М., & Малахов, С. (2024). Узагальнення особливостей відомих засобів міжмережевого екранування. *Proceedings of the XXI International Scientific and Practical Conference.* Sofia, Bulgaria. 2024. Pp. 370-376. Available at: https://doi.org/10.46299/ISG.2024.1.21
10. What is a Threat Intelligence Feed? Вилучено з URL: https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/threat-intelligence-feeds/
11. Guofei Gu, Junjie Zhang & Wenke Lee. BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. URL: https://people.engr.tamu.edu/guofei/paper/Gu_NDSS08_botSniffer.pdf
12. How Does a DNS Amplification Attack Work? (2024). Check Point. Вилучено з https://www.checkpoint.com/cyber-hub/network-security/what-is-dns-security/what-is-a-dns-amplification-attack/
13. Albulayhi, K., Smadi, A., Sheldon, F., & Abercrombie, R. (2021). IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. Sensors, (6432), 21. https://doi.org/10.3390/s21196432
14. Коробейнікова, Т., & Федчук, Т. (2024). Огляд протоколів DNS, DOH та DOT. *Collection of Scientific Papers «ΛΟΓΟΣ»*, (March 1, 2024; Paris, France), 253–256. https://doi.org/10.36074/logos-01.03.2024.056
15. Гайкова, В., & Малахов, С. (2021). Аналіз факторів і умов реалізації кібербулінгу з урахуванням можливостей сучасних інформаційних систем. *Комп'ютерні науки та кібербезпека,* (1), 50-59. https://doi.org/10.26565/2519-2310-2021-1-04
16. Hugo M. Connery. DNS Response Policy Zones History, Overview, Usage and Research. URL: https://www.dnsrpz.info/RPZ-History-Usage-Research.pdf
17. What Are Domain Generation Algorithms? Вилучено з URL: https://www.akamai.com/glossary/what-are-dgas
18. Anhar Haneef. On the Scalable Generation of Cyber Threat Intelligence from Passive DNS Streams URL: http://surl.li/phbham
19. Keijo Korte. Measuring the quality of Open Source Cyber Threat Intelligence Feeds. URL: http://surl.li/yhiqoe
20. Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, Stefan Savage, Kirill Levchenko. Reading the Tea Leaves: A Comparative Analysis of Threat Intelligence, URL: https://www.usenix.org/system/files/sec19-li-vector_guo.pdf
21. Constantinos Patsakis, Fran Casino. Exploiting Statistical and Structural Features for the Detection of Domain Generation Algorithms. URL: https://arxiv.org/pdf/1912.05849
22. Joewie J. Koh, Barton Rhodes. Inline Detection of Domain Generation Algorithms with Context-Sensitive Word Embeddings. URL: https://arxiv.org/pdf/1811.08705
23. Amara Dinesh Kumar, Harish Thodupunoori, R. Vinayakumar, K. P. Soman, Prabaharan Poornachandran, Mamoun Alazab, and Sitalakshmi Venkatraman. Enhanced Domain Generating Algorithm Detection Based on Deep Neural Networks. URL http://surl.li/sgufmu
24. Minzhao Lyu, Hassan Habibi Gharakheili, Vijay Sivaraman. A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques. URL: https://arxiv.org/pdf/2201.00900
25. Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang & Jianping Wu. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come? URL: http://surl.li/ebouep

26. Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, Carmela Troncoso. Encrypted DNS - Privacy? A Traffic Analysis Perspective. URL: https://arxiv.org/abs/1906.09682
27. Hugo M. Connery. DNS Response Policy Zones History, Overview, Usage and Research. URL: https://www.dnsrpz.info/RPZ-History-Usage-Research.pdf
28. Скибун, О. (2023). Фішинг та фішери в сучасному світі. *Grail of Science*, (23), 259–264. https://doi.org/10.36074/grail-of-science.23.12.2022.38
29. Hikaru Ichise, Yong Jin & Katsuyoshi Iida. Policy-based Detection and Blocking System for Abnormal Direct Outbound DNS Queries using RPZ. URL: https://eprints.lib.hokudai.ac.jp/dspace/handle/2115/86951
30. Kamal Alieyan, Ammar Almomani, Ahmad Manasrah, Mohammed M. Kadhum. A survey of botnet detection based on DNS. URL: http://surl.li/vqinqn
31. Hyunsang Choi, Hanwoo Lee, Heejo Lee, Hyogon Kim. Botnet Detection by Monitoring Group Activities in DNS Traffic. URL: http://surl.li/nbbypc
32. David Zhao, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, Dan Garant. Botnet detection based on traffic behavior analysis and flow intervals. URL: http://surl.li/ydwehw
33. Чепель, Д., Малахов, С. & Колованова, Є. (2024). Огляд можливостей фільтрації DNS, як інструмента безпеки сучасних інформаційних систем. *Grail of Science*, (42), 395–398. https://doi.org/10.36074/grail-of-science.02.08.2024

SUMMARY OF DNS TRAFFIC FILTERING TRENDS AS A COMPONENT OF MODERN INFORMATION SYSTEMS SECURITY

Danylo Chepel[1], CSD Student (master), Department of Security of Information Systems and Technologies; e-mail: dan4epel@gmail.com; ORCID: https://orcid.org/0000-0003-3308-424X
Serhii Malakhov[1], Ph.D., Senior Researcher, Computer Science Department;
e-mail: malakhov@karazin.ua; ORCID: https://orcid.org/0000-0001-8826-1616
[1] *V. N. Karazin Kharkiv National University, Ukraine*

Manuscript was received April 21, 2024; Received after review May 21, 2024; Accepted June 23, 2024

Abstract. The study analyzes sources related to methods and technologies for DNS (Domain Name System) traffic filtering. Five main directions are identified that are actively used to enhance security at the DNS level. All examined technologies offer improvements in the quality of DNS filtering. It is emphasized that combining different approaches simultaneously can enhance overall security. The summary of research results on DNS traffic security issues indicates certain problems in the quality of the threat intelligence channels used. Therefore, the implementation of AI and LM technologies should enhance the "depth" of extracting useful information about current threats. It is emphasized that the consideration of information security issues should be conducted exclusively in the context of preventing the disparity of artificial intelligence (AI) capabilities in favor of the adversary (i.e., cybercriminals). Practically, this means that future DNS filtering systems should widely implement the latest advancements in VR, AI, LM, and DL technologies. This is particularly important in countering Domain Generation Algorithm (DGA) mechanisms and the spread of botnets. The specific issues of ensuring a consensus on the security and performance of current information and communication systems when implementing DNS encryption tools are highlighted. The primary problem associated with DNS traffic encryption is the potential for its misuse by attackers to conceal their destructive activities (phishing, spam, etc.).

Keywords: *DNS, DGA, RPZ, information security, security threats, traffic filtering, botnet*

Conflicts of Interest: the authors declare no conflict of interest.

References

1. What is DNS? URL: https://www.cloudflare.com/learning/dns/what-is-dns/
2. Pohorila, K., Lesnaya, Yu., Bogdanova, E., & Malakhov, S. (2022). Social engineering as a factor in the implementation of insider threats. *Scientific Collection «InterConf», (111): with the Proceedings of the 1st International Scientific and Practical Conference «Scientific Community: Interdisciplinary Research» (June 6-8, 2022)*. Boston, USA; pp. 494-501. https://archive.interconf.center/in-dex.php/conference-proceeding/article/view/645/666 [In Ukrainian]
3. Lesnaya, Yu., Malakhov, S. Generalization of the main prerequisites for the implementation of phishing attacks. *Proceedings of the XVII International Scientific and Practical Conference.* Ankara, Turkey. 2023. Pp.453-457. Available at: https://doi.org/10.46299/ISG.2023.1.17 [In Ukrainian]
4. What is DNS filtering? URL: https://www.cloudflare.com/learning/access-management/what-is-dns-filtering/
5. Yaremchuk, K., Voskoboynikov, D., & Melkozyorova, O. (2022). Modern threats and ways to secure web applications. *Computer Science and Cybersecurity*, (2), 28-34. https://doi.org/10.26565/2519-2310-2022-2-03 [In Ukrainian]
6. Bohdanova, E., Chorna, T., & Malakhov, S. (2022). Overview of the current state of threats caused by the influence of exploits. *Computer Science and Cyber Security,* (2), 35-40. https://doi.org/10.26565/2519-2310-2022-2-04 [In Ukrainian]
7. Kokhanovska, T., Narezhny, O., & Dyachenko, O. (2020). Exploring the capabilities of Honeypot technology. *Computer Science and Cybersecurity,* 1(1), 33-42. https://doi.org/10.26565/2519-2310-2020-1-03 [In Ukrainian]
8. Mykhaylenko D., Nemtsev M. Peculiarities of the technology of network traps as a tool of active protection and analysis of the actions of the attacking party. *Proceedings of the XXI International Scientific and Practical Conference.* Melbourne, Australia. 2023. Pp. 483-487. Available at: https://doi.org/ 10.46299/ISG.2023.1.21 [In Ukrainian]
9. Sichkar, M., & Malakhov, S. (2024). Generalization of the features of known means of network shielding. *Proceedings of the XXI International Scientific and Practical Conference.* Sofia, Bulgaria. 2024. Pp. 370-376. Available at: https://doi.org/10.46299/ISG.2024.1.21 [In Ukrainian]
10. What is a Threat Intelligence Feed? URL: https://www.crowdstrike.com/cybersecurity-101/threat-intelligence/threat-intelligence-feeds/
11. Guofei Gu, Junjie Zhang & Wenke Lee. BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. URL: https://people.engr.tamu.edu/guofei/paper/Gu_NDSS08_botSniffer.pdf
12. How Does a DNS Amplification Attack Work? (2024). Check Point. URL: https://www.checkpoint.com/cyber-hub/network-security/what-is-dns-security/what-is-a-dns-amplification-attack/
13. Albulayhi, K., Smadi, A., Sheldon, F., & Abercrombie, R. (2021). IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. Sensors, (6432), 21. https://doi.org/10.3390/s21196432
14. Korobeynikova, T., & Fedchuk, T. (2024). Overview of protocols DNS, DOH, DOT. *Collection of Scientific Papers «ΛΟΓΟΣ»*, (March 1, 2024; Paris, France), 253–256. https://doi.org/10.36074/logos-01.03.2024.056 [In Ukrainian]
15. Haykova, V., & Malakhov, S. (2021). Analysis of factors and conditions for the implementation of cyberbullying, taking into account the capabilities of modern information systems. *Computer Science and Cybersecurity,* (1), 50-59. https://doi.org/10.26565/2519-2310-2021-1-04 [In Ukrainian]
16. Hugo M. Connery. DNS Response Policy Zones History, Overview, Usage and Research. URL: https://www.dnsrpz.info/RPZ-History-Usage-Research.pdf
17. What Are Domain Generation Algorithms? URL: https://www.akamai.com/glossary/ what-are-dgas
18. Anhar Haneef. On the Scalable Generation of Cyber Threat Intelligence from Passive DNS Streams URL: http://surl.li/phbham
19. Keijo Korte. Measuring the quality of Open Source Cyber Threat Intelligence Feeds. URL: http://surl.li/yhiqoe
20. Vector Guo Li, Matthew Dunn, Paul Pearce, Damon McCoy, Geoffrey M. Voelker, Stefan Savage, Kirill Levchenko. Reading the Tea Leaves: A Comparative Analysis of Threat Intelligence, URL: https://www.usenix.org/system/files/sec19-li-vector_guo.pdf

21. Constantinos Patsakis, Fran Casino. Exploiting Statistical and Structural Features for the Detection of Domain Generation Algorithms. URL: https://arxiv.org/pdf/1912.05849
22. Joewie J. Koh, Barton Rhodes. Inline Detection of Domain Generation Algorithms with Context-Sensitive Word Embeddings. URL: https://arxiv.org/pdf/1811.08705
23. Amara Dinesh Kumar, Harish Thodupunoori, R. Vinayakumar, K. P. Soman, Prabaharan Poornachandran, Mamoun Alazab, and Sitalakshmi Venkatraman. Enhanced Domain Generating Algorithm Detection Based on Deep Neural Networks. URL http://surl.li/sgufmu
24. Minzhao Lyu, Hassan Habibi Gharakheili, Vijay Sivaraman. A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques. URL: https://arxiv.org/pdf/2201.00900
25. Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Haixin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang & Jianping Wu. An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come? URL: http://surl.li/ebouep
26. Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, Carmela Troncoso. Encrypted DNS - Privacy? A Traffic Analysis Perspective. URL: https://arxiv.org/abs/1906.09682
27. Hugo M. Connery. DNS Response Policy Zones History, Overview, Usage and Research. URL: https://www.dnsrpz.info/RPZ-History-Usage-Research.pdf
28. Skibun, O. (2023). Phishing and phishers in the modern world. *Grail of Science*, (23), 259–264. https://doi.org/10.36074/grail-of-science.23.12.2022.38 [In Ukrainian]
29. Hikaru Ichise, Yong Jin & Katsuyoshi Iida. Policy-based Detection and Blocking System for Abnormal Direct Outbound DNS Queries using RPZ. URL: https://eprints.lib.hokudai.ac.jp/dspace/handle/2115/86951
30. Kamal Alieyan, Ammar Almomani, Ahmad Manasrah, Mohammed M. Kadhum. A survey of botnet detection based on DNS. URL: http://surl.li/vqinqn
31. Hyunsang Choi, Hanwoo Lee, Heejo Lee, Hyogon Kim. Botnet Detection by Monitoring Group Activities in DNS Traffic. URL: http://surl.li/nbbypc
32. David Zhao, Issa Traore, Bassam Sayed, Wei Lu, Sherif Saad, Ali Ghorbani, Dan Garant. Botnet detection based on traffic behavior analysis and flow intervals. URL: http://surl.li/ydwehw
33. Chepel, D., Malakhov, S. & Kolovanova, E. (2024). Overview of DNS filtering capabilities as a security tool for modern information systems. *Grail of Science*, (42), 395–398. https://doi.org/10.36074/grail-of-science.02.08.2024 [In Ukrainian]

ISSN 2519-2310 (Online)

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА, 2024, № 1 (25)

COMPUTER SCIENCE AND CYBERSECURITY, 2024, Iss. 1 (25)

DOI: https://doi.org/10.26565/2519-2310-2024-1-02
УДК 004.056.5

АНАЛІЗ ФАКТОРА ЕРМІТА АЛГОРИТМУ BKZ НА РЕШІТКАХ МАЛОЇ РОЗМІРНОСТІ

Іван Горбенко[1], доктор технічних наук, професор, e-mail: i.d.gorbenko@karazin.ua, ORCID: https://orcid.org/0000-0003-4616-3449
Сергій Кандій[1], аспірант кафедри захисту інформаційних систем та технологій, e-mail: sergeykandy@gmail.com, ORCID: https://orcid.org/0000-0003-4616-3449
[1]*Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, 61022, Україна*

Рукопис надійшов 1 квітня 2024 р. Отримано після рецензування 2 травня 2024 р. Прийнято 3 червня 2024 р.

Анотація: Криптографія на решітках є одним з перспективних напрямів досліджень у сучасній криптографії. Електронні підписи та механізми інкапсуляції ключів на решітках вже використовуються на практиці. У перспективі такі квантово-стійкі перетворення на решітках замінять усі стандарти, що не мають стійкості до атак на квантових комп'ютерах. Це робить аналіз їх безпеки надзвичайно актуальним. Аналіз безпеки криптографічних перетворень на решітках часто зводиться до оцінки мінімального розміру блоку у алгоритмах редукції решіток. Щоб визначити наскільки малі вектори може отримати алгоритм редукції для заданого розміру блоку часто використовується модель GSA, яка використовує так званий фактор Ерміта для передбачення розміру векторів, які може отримати алгоритм редукції решіток при заданих параметрах. Для його оцінки на практиці використовуються асимптотичні формули, проте питання їх точності на криптографічних решітках не до кінця досліджено. В роботі було отримано оцінки точності існуючих асимптотичних оцінок фактору Ерміта для решіток розмірностей 120, 145, 170 для класичного алгоритму BKZ. Дослідження проводились з використанням бібліотеки fpylll. Було показано, що існуючі оцінки з практичної точки зору є еквівалентними та мають достатньо мале середньоквадратичне відхилення від істинних значень. Було отримано формулу, що прив'язує середньоквадратичну похибку апроксимації фактору Ерміта до криптографічних параметрів решіток. Отримані результати є корисними для уточнення оцінок безпеки існуючих криптографічних перетворень.

Ключові слова: *квантово-стійка криптографія, криптографія на решітках, фактор Ерміта, BKZ, GSA*

Як цитувати: Горбенко І., Кандій С.. Аналіз фактора Ерміта алгоритму BKZ на решітках малої розмірності. *Комп'ютерні науки та кібербезпека.* 2024; № 1(25): С. 22–34. https://doi.org/10.26565/2519-2310-2024-1-02

In cites: Gorbenko I., Kandii S. (2024). The analysis of Hermite factor of BKZ algorithm on small lattices. *Computer Science and Cybersecurity.* 1(25): 22–34. https://doi.org/10.26565/2519-2310-2024-1-02 (in Ukrainian)

1. Вступ

Криптографія на решітках є перспективним напрямком досліджень, який активно розвивається в останні роки. Зокрема, фіналістами конкурсу NIST PQC стали криптографічні схеми на решітках [1]. У той же час, в Україні вже навіть є квантово-стійкі стандарти на решітках: стандарт квантово-стійкого асиметричного шифрування ДСТУ 8961:2019 [2] та стандарт квантово-стійкого електронного підпису ДСТУ 9212:2023 [3]. Іншим напрямком застосування криптографії на решітках є схеми гомоморфного шифрування [4], які вже активно використовуються в системах, що потребують роботи з конфіденційною інформацією без її розголошення. Це робить актуальними дослідження безпеки криптографічних перетворень на решітках.

Криптоаналіз сучасних криптографічних перетворень на решітках переважно зводиться до аналізу процесів редукції базису решіток. Алгебраїчні та комбінаторні техніки при цьому грають допоміжну роль [5, 6]. Для оцінки складності редукції решіток використовуються різні моделі, які дозволяють оцінити характеристики базису решітки після редукції. Фактор Ерміта є важливим показником якості редукції базису, який показує наскільки малі вектори здатен отримати заданий алгоритм редукції решіток [5]. Проте, сучасні оцінки фактору Ерміта є асимптотичними і на практиці можуть дещо відрізнятися від реальних значень фактору Ерміта. Оцінка, що була представлена в роботі [7], вважається стандартною оцінкою фактору Ерміта при криптоаналізі криптографічних перетворень на решітках. Проте, у роботі [8] була отримана більш точна асимптотична оцінка фактору Ерміта, хоча і не набула популярності серед дослідників.

Метою цієї роботи є експериментальне порівняння поведінки існуючих асимптотичних оцінок фактору Ерміта на решітках малої розмірності для алгоритму BKZ. У результаті проведеного аналізу були отримані оцінки середньоквадратичної похибки для відомих асимптотичних оцінок фактору Ерміта та було показано, що для криптографічних значень вплив помилки апроксимації фактору Ерміта є незначним.

2. Теоретичні відомості з теорії решіток

Для аналізу введемо основні теоретичні положення теорії решіток. Теоретичні відомості викладаються згідно [9]. Решітка Λ з базисом $B=(b_1,...,b_n)$ є множиною цілочисельних комбінацій лінійно незалежних векторів $b_1,...,b_n \in \mathbb{R}^n$:

$$\Lambda(b_1,...,b_n)=\left\{\sum_{i=1}^{n} x_i b_i, x_i \in \mathbb{Z}\right\} \quad (1)$$

Довжиною вектору v є стандартна евклідова норма $\|v\|=\sqrt{v\cdot v}$, де операція «$\cdot$» є скалярним добутком, який для двох векторів $v=(v_1,...,v_n)$, $w=(w_1,...,w_n)$ визначений як $v\cdot w=\sum_{i=1}^{n} v_i w_i$.

Для заданого базису $B=(b_1,...,b_n)$ для решітки, що задається формулою (1), ортогоналізований за Граммом-Шмідтом базис є $B^*=(b_1^*,...,b_n^*)$, де $b_i^*=b_i-\sum_{j=1}^{i-1}\mu_{ij}b_j^*$ для $1\le j<i\le n$, де $\mu_{ij}=(b_i\cdot b_j^*)/\|b_j^*\|^2$ – коефіцієнти Грамма-Шмідта, $\|b_j^*\|$ – довжини векторів Грамма-Шмідта. Профілем базису будемо називати вектор $(\|b_1^*\|,\|b_2^*\|,...,\|b_n^*\|)$.

Для решітки $\Lambda = \Lambda(B) \subseteq \mathbb{R}^n$ з базисом $B \in \mathbb{R}^{n \times k}$ фундаментальний паралелепіпед визначений як $P(B) = \{B \cdot x \mid x \in [0,1)^k\}$. Детермінант базису решітки є інваріантом і може бути обчислений як $\det(L) = \sqrt{\det(B^T B)} = \prod_{i=1}^{n} \left\| b_i^* \right\|$. При цьому, детермінант решітки дорівнює об'єму фундаментального паралелепіпеда $vol(\Lambda)$.

Ортогональна проекція є відображенням $\pi_i : R^n \mapsto span(b_i, ..., b_{i-1})^{\perp}$ для $i \in \{1, ...n\}$. Проективна решітка $\Lambda_{[i:j]}$ – решітка, яка задається наступним чином:

$$\Lambda_{[i:j]} = \Lambda(\pi_i(b_i), \pi_i(b+1), ..., \pi_i(b_j)) \tag{2}$$

Для $j \in \{i, i+1, ..., n\}$.

У кожній решітці Λ існує найменший ненульовий вектор. $\lambda_1(\Lambda)$ – норма найменшого ненульового вектору. Проблема пошуку найменшого вектору (SVP) полягає у пошуку вектору довжини $\lambda_1(\Lambda)$.

Важливим послабленням проблеми SVP є проблема апроксимації найменшого вектору – α-SVP, яка полягає у пошуку вектору, що має норму, меншу за $\alpha \cdot vol(\Lambda)^{1/n}$, де α – деяка константа, що залежить від розмірності решітки.

Константа Ерміта γ_n визначає обмеження на найменший вектор серед усіх решіток розмірності n і визначена як

$$\gamma_n = \sup_{\Lambda} \{\lambda_1^2(\Lambda) / vol(\Lambda)^{2/n}\} \tag{3}$$

Для константи Ерміта (3) відомі наступні оцінки [9]:

$$\frac{n}{2\pi e} + \frac{\log(\pi n)}{2\pi e} \le \gamma_n \le \frac{1.744n}{2\pi e} + o(n) \tag{4}$$

3. Алгоритми редукції решіток та фактор Ерміта

Алгоритм редукції LLL виконує над базисом дві операції: редукція за розміром $b_i \leftarrow b_i - round(\mu_{ij}) b_j$ для $j \in [i-1]$ та обмін місцями векторів b_i і b_{i+1}, якщо $(\eta - \mu_{i+1,i}^2) \left\| b_{i+1}^* \right\|^2 \le \left\| b_i^* \right\|^2$ для загальносистемного параметра $\eta \in (0.25, 1)$ поки відбуваються зміни у базисі.

Алгоритм BKZ є узагальненням LLL. У алгоритмі BKZ (та його варіаціях) фіксується розмір блоку β і відбувається пошук найменшого вектору на решітках $\Lambda_{[i,i+\beta'-1]}$ (формула (2)) для i від 1 до n–1, де $\beta' = \min(\beta, n-i+1)$. Пошук найменшого вектору відбувається окремою процедурою.

Для індексу i стандартна реалізація алгоритму BKZ викликає алгоритм пошуку найменшого вектору для решітки $\Lambda_{[i,i+\beta'-1]}$ і знаходить найкоротший вектор v на цій решітці. Далі BKZ вставляє v у старий базис між b_{i-1} та b_i. Для базису $(b_1, ..., b_{i-1}, v, b_i, ..., b_{min(i+\beta-1,n)})$ застосовується LLL для отримання нового базису з меншими векторами. Ці процедури складають один раунд алгоритму. У оригінальній версії BKZ алгоритм зупинявся, коли оновлень не відбувалось протягом n–1 раундів.

Для аналізу алгоритму BKZ зазвичай використовується евристика Гауса [5, 6, 9], сутність якої полягає у тому, що кількість $|\Lambda \cap \Omega|$ точок решітки у довільному вимірюваному тілі $\Omega \subset \mathbb{R}^n$ складає $vol(\Omega)/vol(\Lambda)$. Використовуючи d-вимірний шар у якості вимірюваного тіла, для випадкової решітки $\Lambda \subset \mathbb{R}^d$, очікуваний найменший вектор, згідно до евристики Гауса, можливо оцінити як:

$$GH(\Lambda) = \left(\frac{vol(\Lambda)}{vol(\Omega)}\right)^{1/d} = \frac{\Gamma\left(1+\frac{d}{2}\right)}{\sqrt{\pi}} \cdot vol(\Lambda)^{1/d} \approx \sqrt{\frac{d}{2\pi e}} \cdot vol(\Lambda)^{1/d} \quad (5)$$

Практичні експерименти з алгоритмами LLL та BKZ показують [10], що $\left\|b_i^*\right\| / \left\|b_{i+1}^*\right\| \approx const$, якщо $d >> \beta$. У якості ілюстрації цього твердження на рис. 1 наведено профілі для 230-мірної випадкової q-арної решітки для β = 2,10,30,40,50.

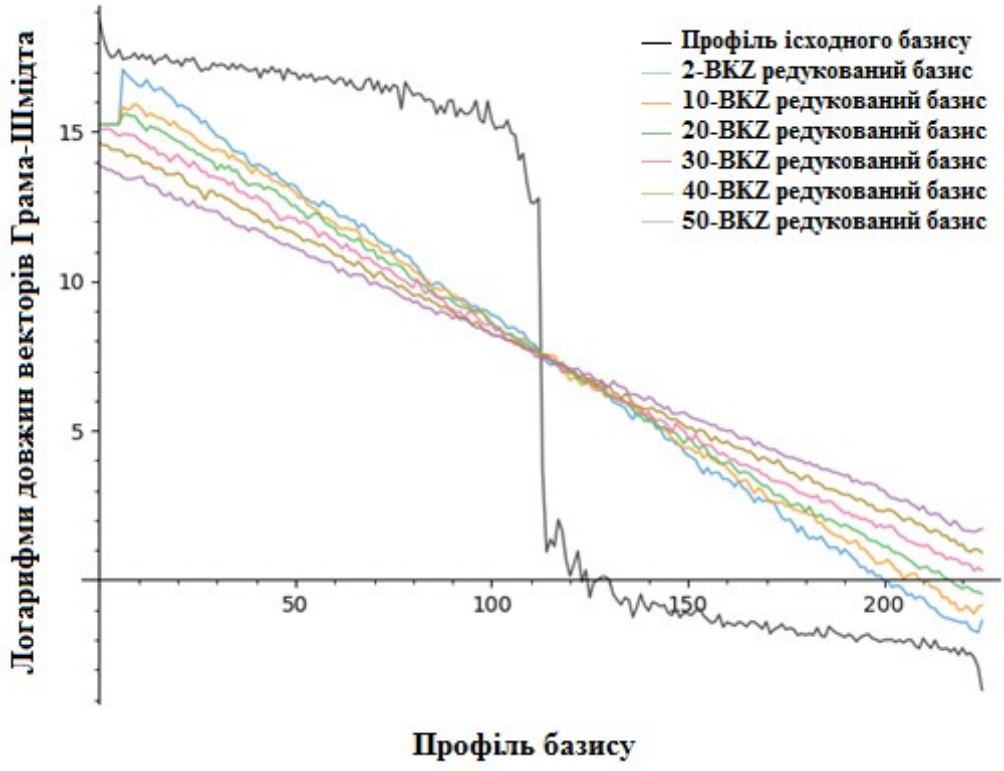


Рис. 1. – Профілі 230-мірної q-арної решітки для β = 2,10,30,40,50
Fig. 1 – Profiles of a 230-dimensional q-ary lattice for β = 2,10,30,40,50

Застосовуючи евристику Гауса (5) до BKZ-β редукованого базису $B = (b_1, ..., b_n)$ та враховуючи припущення $\left\|b_i^*\right\| / \left\|b_{i+1}^*\right\| \approx const$, маємо:

$$\log\left\|b_i^*\right\| = \frac{d-1-2i}{2} \cdot \log(\alpha_\beta) + \frac{1}{d}\log\left(vol(\Lambda)\right) \quad (6)$$

Для деякого α_β, що залежить від властивостей BKZ-β.

Рівняння (6) є моделлю редукції решіток GSA (англ. Geometric Series Assumption) [10]. Експериментальні дослідження у роботах [10, 11] показують доволі гарну точність моделі GSA для $50 < \beta << n$. На рис. 2 наведено приклад застосування моделі GSA.

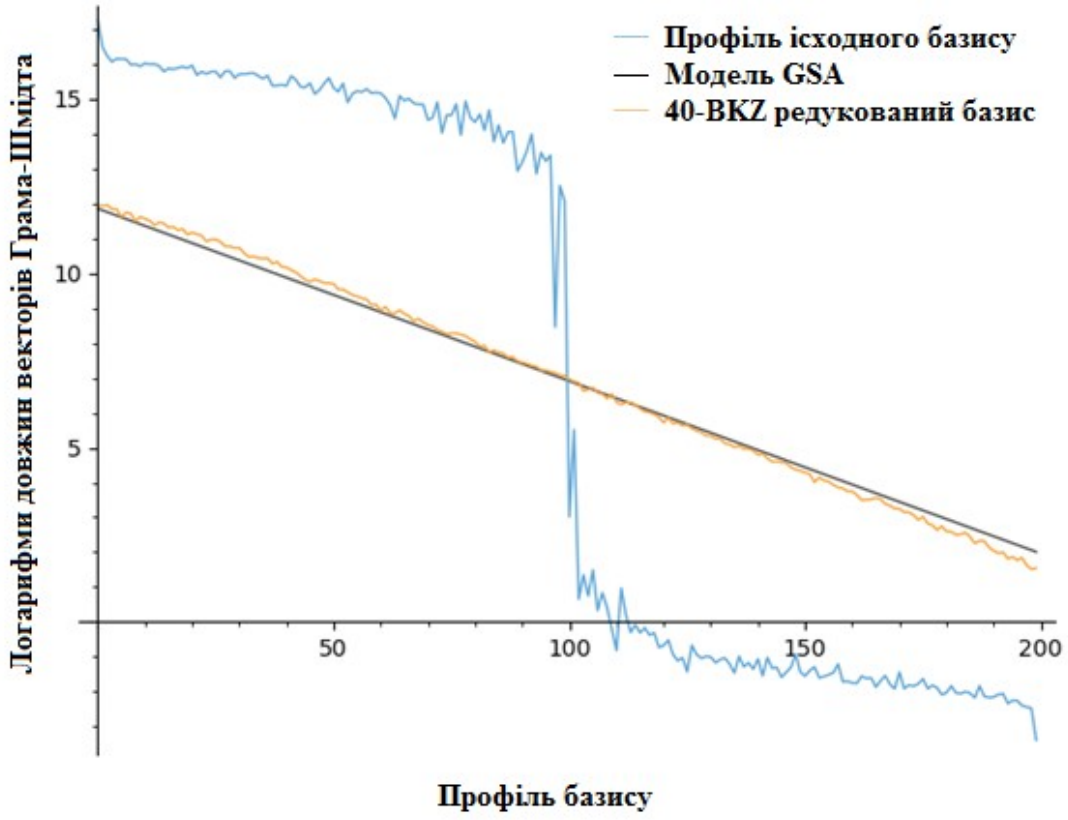


Рис. 2 – Приклад застосування моделі GSA для 200-мірної решітки
Fig. 2 – An example of applying the GSA model to a 230-dimensional lattice

Для алгоритму BKZ, з формули (6) випливає наступна оцінка:

$$\|b_0\| \le \sqrt{\alpha_\beta}^{d-1} \cdot vol(\Lambda(B))^{1/d} \tag{7}$$

Фактор Ерміта визначає найменше значення α, для якого алгоритм BKZ-β може вирішити задачу α-SVP. Він є аналогом константи Ерміта (3), тільки для конкретного базису і формально визначений як

$$\delta_\beta = \left(\|b_0\| / vol(\Lambda)^{1/d} \right)^{1/d} \tag{8}$$

З рівнянь (7) та (8) випливає, що $\delta_\beta = \sqrt{\alpha_\beta}^{1-1/d}$. Ця рівність поєднує формальне визначення фактору Ерміта з його практичним застосуванням для оцінки якості редукції.

У роботі [7] була запропонована асимптотична оцінка

$$\lim_{\beta \to \infty} \delta_\beta = \left(\frac{\beta}{2\pi e} \cdot (\pi\beta)^{1/\beta} \right)^{\frac{1}{2(\beta-1)}} \tag{9}$$

Ця оцінка використовується у всіх сучасних моделях безпеки. У роботі [8] було показано, що ця оцінка є лише першим наближенням і може бути уточнена наступним чином:

$$\lim_{\beta\to\infty}\delta_\beta=\left(\frac{\beta}{2\pi e}\cdot(\pi\beta)^{1/\beta}\right)^{\frac{1}{2(\beta-1)}+\frac{\beta}{2n^2}} \tag{10}$$

На рис. 3 зображено графік фактору Ерміта. З рисунку видно, що оцінка (9) наближається до нижньої теоретичної межі (константи Ерміта, формула (4)), у той час як оцінка (10) є більш помірною.

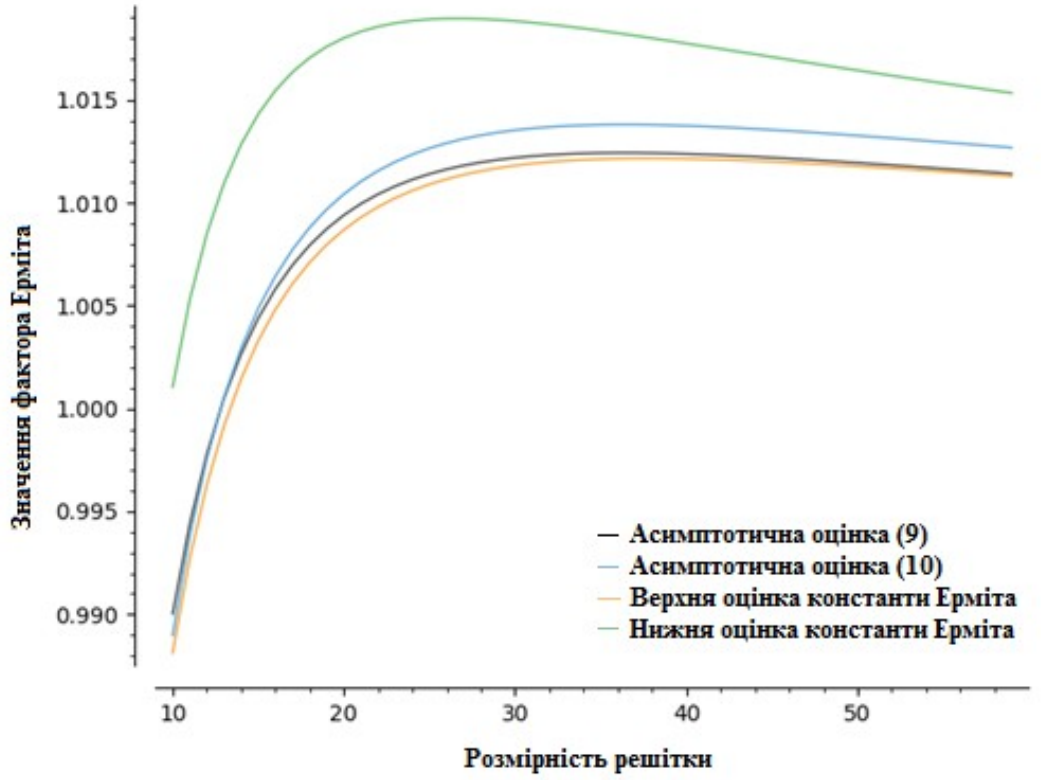


Рис. 3 – Значення фактору Ерміта
Fig. 3 – Asymptotic estimates of the Hermite factor

Фактично, фактор Ерміта у моделі GSA повністю визначає профіль для заданої решітки, тому важливо розуміти які він приймає значення на практиці. Для криптографічно значущих розмірностей, звісно, отримати значення неможливо, проте можливо протестувати на малих розмірностях.

4. Експериментальна оцінка фактору Ерміта

У загальному випадку на значення фактору Ерміта можуть впливати розмір блоку редукції, розмірність решітки та розмір в бітах кожного коефіцієнта векторів в базисі. Для виявлення впливу кожного з цих факторів були проведені експериментальні дослідження поведінки кореневого фактору Ерміта на випадкових решітках.

Дослідження проводилося на випадкових q-арних решітках, оскільки саме такі решітки використовуються в криптографії на решітках. Для простого $q \geq 2$ q-арна решітка визначається базисом

$$B = \begin{pmatrix} qI_m & A \\ 0 & \zeta I_n \end{pmatrix} \in \mathbb{Z}^{(m+n)\times(m+n)}$$

де ζ – деяка константа. Детермінант такої решітки відповідно має значення $q^m\zeta^n$. У проведених експериментах використовувалося значення $\zeta = 1$.

Варто зауважити, що в q-арних решітках у загальному випадку перші та останні вектори можуть не підпорядковуватися моделі GSA, проте при обраних для дослідження параметрах модель GSA працює достатньо гарно.

У таблицях 1-3 наведено результати виміру фактору Ерміта для випадкових q-арних решіток. Для досліджень використовувалися випадкові решітки розмірностей 120, 145, 170. Для кожної з цих розмірностей розглядалися значення $\log_2 q = 10, 20, 40$ для розмірів блоку від 3 до 60.

Для кожного набору параметрів виконувалася оцінка щонайменше на 200 випадкових решітках, генерація яких відбувалася за допомогою засобів бібліотеки fpylll, що реалізує алгоритми редукції решіток. Редукція решіток відбувалася з використанням алгоритму BKZ з стандартними налаштуваннями.

Таблиця 1 – Експериментальні оцінки фактору Ерміта для $\log_2 q = 10$
Table 1 – Experimental estimation of the Hermite factor for $\log_2 q = 10$

β	**Розмірність 120**	**Розмірність 145**	**Розмірність 170**
3	1.017378	1.017690	1.016867
5	1.017378	1.017690	1.016867
10	1.015669	1.016230	1.015883
15	1.014016	1.014413	1.014484
20	1.013434	1.013680	1.013671
25	1.012825	1.013024	1.013012
30	1.012559	1.012833	1.012833
35	1.012352	1.012610	1.012653
40	1.012187	1.012415	1.012458
45	1.011986	1.012163	1.012241
50	1.011449	1.011726	1.011718
55	1.011132	1.011373	1.011397
60	1.010854	1.011097	1.011049

З таблиці 1 видно, що фактор Ерміта залежить від розмірність решітки має вплив на значення фактору Ерміта.

Таблиця 2 – Експериментальні оцінки фактору Ерміта для $\log_2 q = 20$
Table 2 – Experimental estimation of the Hermite factor for $\log_2 q = 20$

β	**Розмірність 120**	**Розмірність 145**	**Розмірність 170**
3	1.017318	1.017987	1.018275

5	1.017318	1.017987	1.018275
10	1.015702	1.016254	1.016846
15	1.013923	1.014496	1.015003
20	1.013243	1.013654	1.013946
25	1.012820	1.012977	1.013223
30	1.012630	1.012746	1.013053
35	1.012296	1.012596	1.012792
40	1.012116	1.012393	1.012651
45	1.011863	1.012116	1.012360
50	1.011399	1.011695	1.011787
55	1.011200	1.011372	1.011453
60	1.010895	1.011065	1.011194

З таблиці 2 видно, що значення q впливає лише на малих значеннях β.

Таблиця 3 – Експериментальні оцінки фактору Ерміта для $\log_2 q = 40$
Table 3 – Experimental estimation of the Hermite factor for $\log_2 q = 40$

β	**Розмірність 120**	**Розмірність 145**	**Розмірність 170**
3	1.017322	1.017685	1.018242
5	1.017322	1.017685	1.018242
10	1.015547	1.016249	1.016789
15	1.013991	1.014420	1.014897
20	1.013351	1.013708	1.013915
25	1.012738	1.013018	1.013307
30	1.012563	1.012835	1.013085
35	1.012377	1.012607	1.012849
40	1.012162	1.012407	1.012661
45	1.012019	1.012167	1.012385
50	1.011495	1.011734	1.011870
55	1.011121	1.011356	1.011508
60	1.010811	1.011107	1.011172

Введемо функціонал середньоквадратичної похибки:

$$MSE(\delta_{etalon}, \delta_{experiment}) = \frac{1}{d}\sum_{i=0}^{d-1}\left(\delta_{etalon}[i] - \delta_{experiment}[i]\right)^2 \quad (11)$$

У таблицях 4-5 наведено значення середньоквадратичної похибки (11) для оцінок фактору Ерміта, починаючи від $\beta > 30$ для (9) та (10) відповідно.

Таблиця 4 – Значення MSE для оцінок фактору Ерміта (9)
Table 4 – MSE estimate for the Hermite factor estimation (9)

	Розмірність 120	**Розмірність 145**	**Розмірність 170**
$\log_2 q = 10$	0.0003728998	0.0002477775	0.0002575795
$\log_2 q = 20$	0.0003651147	0.0002409057	0.0003197217
$\log_2 q = 40$	0.0003566181	0.0002485526	0.0003261832

Таблиця 5 – Значення MSE для оцінок фактору Ерміта (10)
Table 5 – MSE estimate for the Hermite factor estimation (10)

	Розмірність 120	**Розмірність 145**	**Розмірність 170**
$\log_2 q = 10$	0.0021920808	0.0013799646	0.0010351942
$\log_2 q = 20$	0.0021856184	0.0014050912	0.0009452115
$\log_2 q = 40$	0.0021794041	0.0013804805	0.0009179317

З таблиць 4-5 випливає, що середньоквадратична помилка для оцінки (9) є меншою.

На рис. 4 наведено графіки δ_{etalon} та $\delta_{experiment}$, усереднені за параметром q для розмірності 170.

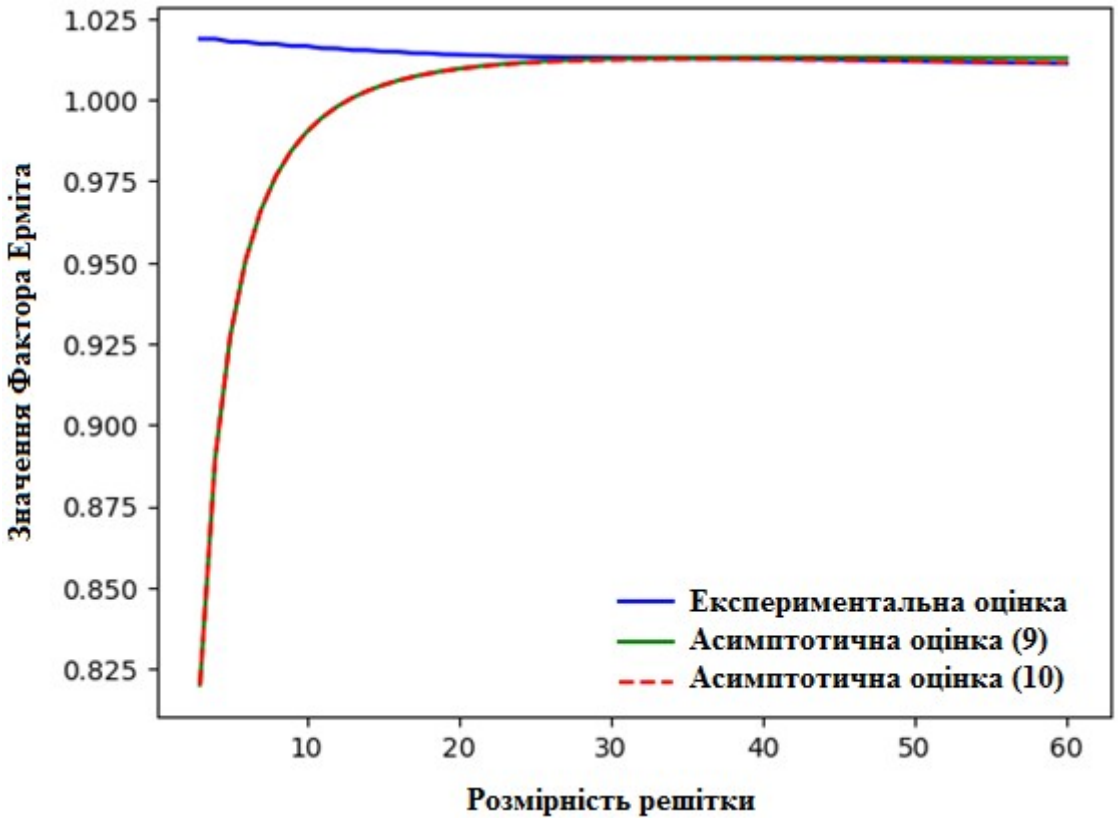


Рис. 4 – Експериментальна оцінка фактору Ерміта для розмірності 170
Fig. 4 – Experimental estimation of the Hermite factor for dimension 170

З рис. 4 видно, що на малих значеннях $\beta < 30$ оцінки (9) та (10) не працюють через свій асимптотичний характер. Далі експериментальна оцінка наближається до (9).

5. Обговорення результатів

На рис. 5 зображено фрагмент рис. 4 для значень $20 \le \beta \le 30$. З рис. 5 видно, що реальне експериментально обчислене значення фактору Ерміта на малих розмірностях навіть менше, ніж дає оцінка (9). При цьому з ростом розмірності реальні значення фактору Ерміта збільшуються, тому для оцінки фактору Ерміта можливо виділити 3 сценарії на решітках у криптографічно значимих розмірностях:

- Песимістичний сценарій. Значення фактору Ерміта не будуть досягати оцінки (9).
- Оптимістичний сценарій. Значення фактору Ерміта будуть близькі до оцінки (10).
- Реалістичний сценарій. Значення фактору Ерміта будуть більшими за (9), проте меншими за (10).

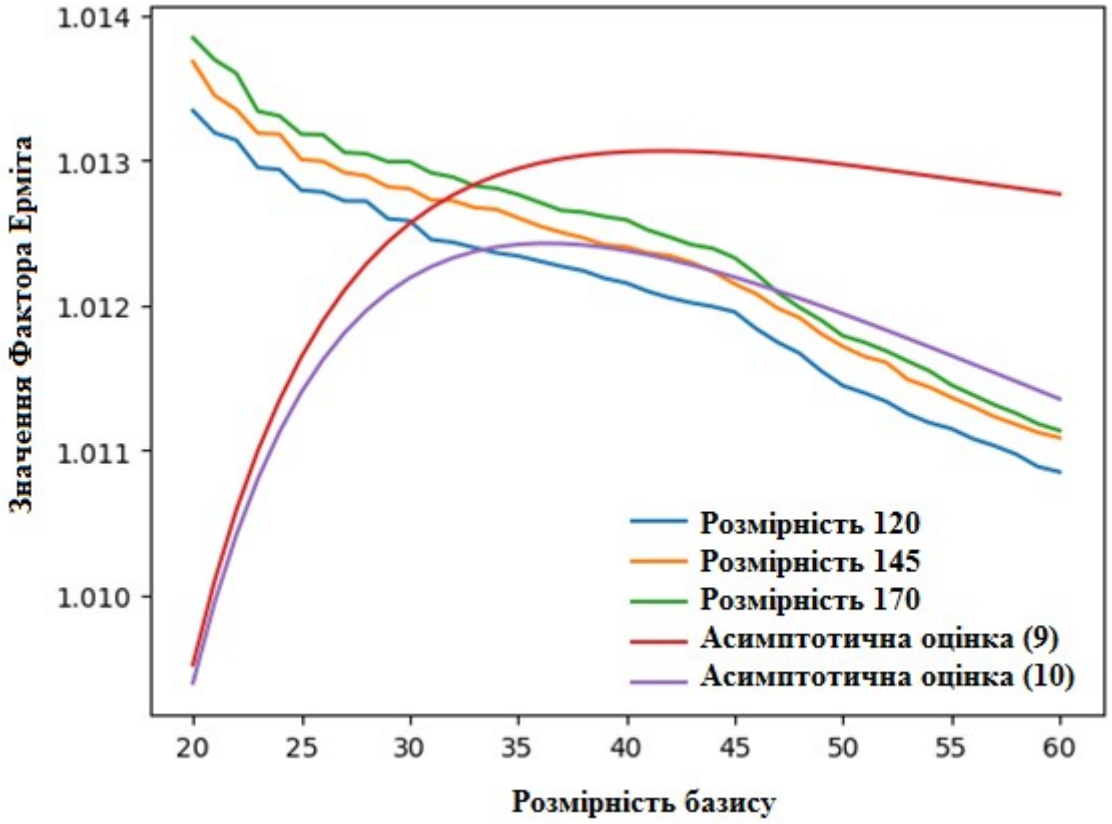


Рис. 5 – Експериментальна оцінка фактору Ерміта для розмірностей 120, 145, 170
Fig. 5 – Experimental estimation of the Hermite factor for dimensions 120,145,170

Для того, щоб знайти вплив похибки вимірювання τ_{mse} фактору Ерміта для великих розмірностей скористаємося біноміальною апроксимацією $(1+x)^{\alpha} \approx 1+\alpha x$ до формули (7):

$$\begin{aligned} \|b_0\| &= \delta^d \cdot vol(\Lambda)^{1/d} \approx (1-(\delta_{real}-1)+\tau_{mse})^d \cdot vol(\Lambda)^{1/d} \\ &\approx (1+d((\delta_{real}-1)+\tau_{mse}))) \cdot vol(\Lambda)^{1/d} = \|b_0\|_{etalon} + d\tau_{mse} \cdot vol(\Lambda)^{1/d} \end{aligned} \quad (12)$$

Оскільки для криптографічних випадків $vol(\Lambda)^{1/d} = q^m$, то з формули (12) маємо оцінку похибки

$$\tau_\delta = d\tau_{mse} \cdot q^{m/d} \approx d\tau_{mse} \cdot \sqrt{q} \quad (13)$$

Тож, для практичних обчислень можливо враховувати похибку апроксимації за формулою (13). Для типових криптографічних параметрів $d\sqrt{q} \approx 10^6$. Оскільки вже для малих розмірностей решіток значення $\tau_{mse} \approx 10^{-3} - 10^{-4}$ і має тенденцію до зменшення, то для криптографічних наборів параметрів похибка буде достатньо малою, щоб не впливати на оцінку безпеки.

6. Висновки

Існуючі асимптотичні оцінки дають гарну апроксимацію фактору Ерміта вже на малих розмірностях решіток. На решітках малої розмірності не було виявлено впливу розміру коефіцієнтів векторів в базисі на значення фактор Ерміта, проте розмірність решітки дійсно має вплив на фактор Ерміта. Хоча оцінка (9) не враховує цього, проте дає кращі результати у порівнянні з оцінкою (10), що враховує вплив розмірності. Проте, оскільки отримана середньоквадратична помилка має порядок 10^{-3}-10^{-4}, то можливо стверджувати, що помилка апроксимації фактору Ерміта не має впливу на оцінку складності редукції решіток для криптографічних параметрів, враховуючи асимптотичних характер формул.

Конфлікт інтересів

Автори повідомляють про відсутність конфлікту інтересів.

Список літератури:

1. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce. (n.d.). *Post-Quantum Cryptography | CSRC | CSRC*. Retrieved from https://csrc.nist.gov/projects/post-quantum-cryptography
2. Інформаційні технології. Криптографічний захист інформації. Алгоритми асиметричного шифрування та інкапсуляції ключів: ДСТУ 8961:2019. (2019). Київ: Держспоживстандарт України.
3. Інформаційні технології. Криптографічний захист інформації. Алгоритм електронного підпису на алгебраїчних решітках із відхилами: ДСТУ 9212:2023. (2023). Київ: Держспоживстандарт України.
4. Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes. *ACM Computing Surveys*, 51(4), 1–35. https://doi.org/10.1145/3214303
5. Albrecht, M. R., Göpfert, F., Virdia, F., & Wunderer, T. (2017). Revisiting the expected cost of solving USVP and applications to LWE. *In Lecture notes in computer science* (pp. 297–322). https://doi.org/10.1007/978-3-319-70694-8_11
6. Albrecht, M. R., Bai, S., Li, J., & Rowell, J. (2021). Lattice Reduction with Approximate Enumeration Oracles. *In Lecture notes in computer science* (pp. 732–759). https://doi.org/10.1007/978-3-030-84245-1_25
7. Chen, Y., & Nguyen, P. Q. (2011). BKZ 2.0: Better Lattice Security Estimates. *In Lecture notes in computer science* (pp. 1–20). https://doi.org/10.1007/978-3-642-25385-0_1
8. Li, J., & Nguyen, P. Q. (2022, March 5). A complete analysis of the BKZ Lattice Reduction algorithm. Retrieved from https://eprint.iacr.org/2020/1237
9. Nguyen, P. Q., & Valle, B. (2010). The LLL algorithm. *Information security and cryptography.* https://doi.org/10.1007/978-3-642-02295-1
10. Schnorr, C. P. (2003). Lattice reduction by random sampling and birthday methods. *In Lecture notes in computer science* (pp. 145–156). https://doi.org/10.1007/3-540-36494-3_14

11. Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. (2019, July 10). Post-quantum key exchange - a new hope. Retrieved from https://eprint.iacr.org/2015/1092

THE ANALYSIS OF HERMITE FACTOR OF BKZ ALGORITHM ON SMALL LATTICES

Ivan Gorbenko[1], Doctor of Sciences (Engineering), Full Prof.; e-mail: i.d.gorbenko@karazin.ua; ORCID: https://orcid.org/0000-0003-4616-3449
Serhii Kandii[1], Ph.D. Student, Department of Security of Information Systems and Technologies; e-mail: sergeykandy@gmail.com; ORCID: https://orcid.org/0000-0003-0552-8341
[1] *V. N. Karazin Kharkiv National University, Ukraine*
Manuscript was received April 1, 2024; Received after review May 2, 2024; Accepted June 3, 2024

Abstract. Lattice cryptography is one of the promising directions in modern cryptography research. Digital signatures and key encapsulation mechanisms on lattices have already been used in practice. In the future, such quantum-resistant transformations on lattices replace all standards that are not resistant to attacks on quantum computers. This makes the analysis of their security extremely relevant. Analysis of the security of cryptographic transformations on lattices is often reduced to the estimation of the minimum block size in the lattice reduction algorithm. For the expansion of small vectors, a reduction algorithm can be obtained for a given block size, the GSA model is often used, which uses the so-called Hermitian factor to predict the size of the vectors that the lattice reduction algorithm can obtain given the parameters. Asymptotic formulas have been developed to evaluate it in practice, but the question of their accuracy on cryptographic lattices has not been fully investigated. The work obtained estimates of the accuracy of the existing asymptotic estimates of the Hermite factor for lattices of sizes 120, 145, 170 for the classical BKZ algorithm. Research was conducted using the fpylll library. It was shown that the existing estimators are equivalent from a practical point of view and have a sufficiently small root mean square deviation from the true values. A formula was obtained that binds the root-mean-square error of approximation of the Hermit factor to the cryptographic parameters of lattices. The obtained results are useful for refining the security assessments of existing cryptographic transformations.

Keywords: *quantum-resistant cryptography; lattice cryptography; the Hermite factor, BKZ, GSA*

Conflicts of Interest: the authors declare no conflict of interest.

References

1. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce. (n.d.). Post-Quantum Cryptography | CSRC | CSRC. Retrieved from https://csrc.nist.gov/projects/post-quantum-cryptography
2. Information technologies. Cryptographic protection of information. Algorithms for asymmetric encryption and key encapsulation: DSTU 8961:2019. (2019). Kiev: State Committee for Standardization of Ukraine.
3. Information technologies. Cryptographic protection of information. Algorithm for electronic signature on algebraic lattices with wicks: DSTU 9212:2023. (2023). Kiev: State Committee for Standardization of Ukraine.
4. Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes. *ACM Computing Surveys*, 51(4), 1–35. https://doi.org/10.1145/3214303

5. Albrecht, M. R., Göpfert, F., Virdia, F., & Wunderer, T. (2017). Revisiting the expected cost of solving USVP and applications to LWE. In Lecture notes in computer science (pp. 297–322). https://doi.org/10.1007/978-3-319-70694-8_11
6. Albrecht, M. R., Bai, S., Li, J., & Rowell, J. (2021). Lattice Reduction with Approximate Enumeration Oracles. In Lecture notes in computer science (pp. 732–759). https://doi.org/10.1007/978-3-030- 84245-1_25
7. Chen, Y., & Nguyen, P. Q. (2011). BKZ 2.0: Better Lattice Security Estimates. In Lecture notes in computer science (pp. 1–20). https://doi.org/10.1007/978-3-642-25385-0_1
8. Li, J., & Nguyen, P. Q. (2022, March 5). A complete analysis of the BKZ Lattice Reduction algorithm. Retrieved from https://eprint.iacr.org/2020/1237
9. Nguyen, P. Q., & Valle, B. (2010). The LLL algorithm. Information security and cryptography. https://doi.org/10.1007/978-3-642-02295-1
10. Schnorr, C. P. (2003). Lattice reduction by random sampling and birthday methods. In Lecture notes in computer science (pp. 145–156). https://doi.org/10.1007/3-540-36494-3_14
11. Alkim, E., Ducas, L., Pöppelmann, T., & Schwabe, P. (2019, July 10). Post-quantum key exchange - a new hope. Retrieved from https://eprint.iacr.org/2015/1092

ISSN 2519-2310 (Online)

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА, 2024, № 1 (25)

COMPUTER SCIENCE AND CYBERSECURITY, 2024, Iss. 1 (25)

DOI: https://doi.org/10.26565/2519-2310-2024-1-03
УДК 004.056.5

SECURITY IN THE ERA OF WIRELESS INNOVATIONS: ANALYSIS OF POTENTIAL THREATS AND PROTECTIVE MEASURES

Yevheniia Matvieieva[1], bachelor's student at the Faculty of Computer Science,
e-mail: belka.j.0507@gmail.com, ORCID: https://orcid.org/0000-0001-8801-2185
Maryna Yesina[1,2], candidate of Technical Sciences, Associate Professor,
e-mail: m.v.yesina@karazin.ua, ORCID: https://orcid.org/0000-0002-1252-7606
Oleksandr Shumov[2], technical director of JSC «IIT», e-mail: alex.shumoff@gmail.com
[1]*V. N. Karazin Kharkiv National University, Svobody Square, 4, Kharkiv, 61022, Ukraine*
[2]*JSC «IIT», Kolomenska Street, 15, Kharkiv, 61166, Ukraine*
Manuscript was received April 2, 2024; Received after review May 2, 2024; Accepted June 3, 2024

Abstract: In today's interconnected world, wireless data transmission technologies have seamlessly integrated into the fabric of modern business operations. As reliance on these technologies grows, so does the imperative to ensure robust cyber security measures. Particularly in the age of wireless innovations, exemplified by the proliferation of the Internet of Things (IoT), the discourse surrounding the security of wireless technologies underscores the necessity of comprehending both established threats and the continuous emergence of new vulnerabilities. This underscores the urgent need for timely detection and mitigation strategies. While the convenience afforded by wireless data transmission technologies grants society unprecedented access to information and facilitates the management of diverse devices, processes, and systems, it also exposes users and modern information and communication systems (ICS) to significant cyber threats and vulnerabilities. Consequently, there arises a pressing need to address these challenges comprehensively. This research dissects contemporary methodologies aimed at restricting access to wireless networks, identifying potential vulnerabilities, and crafting effective responses to cyberattacks. It delves into various facets of cyber security, including data encryption, user authentication mechanisms, traffic monitoring protocols, and anomaly detection algorithms. Furthermore, it delves into the crucial aspect of educating personnel on wireless security practices, equipping them with threat awareness and incident response capabilities. Given the dynamic landscape of cybersecurity technologies and threats, this work seeks to establish a foundational understanding of the security landscape within wireless networks. By doing so, it aims to outline pragmatic strategies for effectively managing security risks, thereby fortifying the resilience of modern organizations and safeguarding critical information assets.

Keywords: *wireless technology, information security, Internet of Things (IoT), vulnerabilities, authentication, quantum cryptography*

In cites: Matvieieva Y., Yesina M., Shumov O. (2024). Security in the era of wireless innovations: analysis of potential threats and protective measures. *Computer Science and Cybersecurity.* 1(25): 35–41. https://doi.org/10.26565/2519-2310-2024-1-03

1. Introduction

One of the most widespread impacts of wireless technologies (*data transfer networks*) is a combination of factors, which are connected to the "weak" strength of the password and/or lack of reliable mechanisms of authentication and authorization. This may lead to unauthorized access to the network and/or confidential data. Some wireless devices use standard passwords or information exchange protocols that contain vulnerabilities caused by "weak" security/ It makes it difficult to ensure the required level of security of existing informational resources. It creates a conditional path for attackers to gain access to the configuration options of the affected network equipment and/or gain access to sensitive information circulating through the compromised device [1].

The presence of vulnerabilities in network protocols of wireless technologies, such as *Wi-Fi* or *Bluetooth*, is also a serious challenge. Unencrypted or poorly secured networks can be easily attacked. As a result, attackers will be allowed to intercept data and/or inject their software.

Security vulnerabilities in wireless networks put both business and personal interests at risk. Effective protection of wireless protection requires awareness of potential threats and the use of up-to-date measures to prevent possible attacks or data leakage. In a business environment, unauthorized access to confidential information may lead to leakage of valuable data, disclosure of commercial secrets, financial losses, or loss of customer trust [6]. Relevant threats may include attacks on remote access systems, attacks on connected IoT devices, etc. [4].

2. Challenges and Opportunities in Wireless Security

Based on the analysis of the latest trends in the development of IT technologies and summarizing the results of known security incidents, it is possible to identify several key challenges and new opportunities that are worth paying attention to:

1. Constant growth in the number of connected wireless devices: integration into average life IoT, increases the number of devices, which creates new attack vectors and increases IS threats.

2. Expansion of the used frequency band: the introduction of new IT technologies determines the need for a greater width of the frequency band, while at the same time complicating the principles of formation of the used signal-code structures and methods of compression of data transmission channels. The combination of these factors creates prerequisites for the emergence of new security challenges in the field of administration of existing channel resources and protection of information circulating (*stored*) in the respective networks [2].

3. Integration of wireless solutions into the information infrastructure of modern cities. This field of activity has a very high pace of implementation, and this may become the main prerequisite for the spread of cyberattacks, which require new, specific (*simplicity, expansion, decentralization of management, low-resource, etc.*) security measures [3].

4. Development of innovative cyber protection technologies. The emergence of new technologies and their mutual integration (*for example, bio- and information technologies*), provides opportunities for creating more effective, multi-level cluster systems for monitoring and protecting information resources and network environments (*for example, blockchain with elements of virtual/augmented reality (VR/AR) or the synthesis of both units and group emulation of their network behavior within created bot farms, etc.*) [8].

Data protection when implementing wireless technologies requires a comprehensive approach. The main protection strategies should include a wide range of technical and organizational measures. These measures, combined with proper access management, ongoing monitoring of current processes, and staff training, are the basic components of a successful wireless security strategy. In this context, it should be noted that with the widespread use of wireless technologies in finance, especially mobile banking [7], the issue of wireless communication security is becoming increasingly relevant.

3. Enhancing Security Measures in Wireless Technology

Network security measures include the use of encryption to protect data transmission over wireless networks, the implementation of appropriate security protocols (*for example, WPA3 in Wi-Fi networks*), and the control and monitoring of network traffic to detect anomalies or possible threats. Along with this, timely software updates, installation of security patches, and use of virtual private networks to protect data transmission are important aspects of network security in a wireless environment [11].

It should be emphasized that the use of wireless networks significantly increases the risk of access to personal (private) information. In this case, the use of data encryption on devices and during information transmission via wireless networks is also an integral part of personal data protection [10]. Encryption ensures the confidentiality and integrity of information when it is transmitted over networks. In addition, an additional step in the preservation of personal data is to limit access to sensitive information. In this sense, users should carefully monitor and control who and under what conditions they provide access to their private data.

From the point of view of further prospects for ensuring security in wireless technologies, the following areas should be highlighted:

1. Quantum cryptography. This direction can significantly change encryption methods (protocols) and provide proportional protection against quantum computers and new algorithms for relevant cyber-attacks [5].

2. Integration of artificial intelligence and machine learning (*AI/LM*) capabilities. The use of AI/LM capabilities to detect network anomalies (*including network behavior anomalies* [12, 13]) in wireless networks and user behavior analysis will allow prompt response to potential threats and/or minimize the consequences of their implementation.

3. Biometric methods of authentication. The simultaneous use of various biometric features (fingerprints, face recognition, retina, etc.) can become a standard for secure access to data devices and important/critical IC control functions).

4. Management of security incidents. The development and implementation of centralized monitoring systems (including based on the broad involvement of AI, ML, AR capabilities, etc.) and rapid response to security incidents [9] should provide opportunities for early detection of IS threats and effective countermeasures against new types of cyber-attacks.

5. Synthesis of new and modification of already existing security protocols, as well as the emergence of new types of electronic services and methods of interaction (*VR, AR, AI, etc.*) of users, both among themselves and when requesting/requesting the necessary information resources.

6. Gradual growth of end-user competencies. Training and increasing the level of competencies in IS issues among ordinary users should become the main component of basic security skills, helping to avoid social engineering attacks and phishing [14, 15].

In general, these directions of development in the field of wireless technologies are most likely to determine the future level of security in the world of wireless solutions, providing more effective and reliable tools and technologies for the protection of information and the functioning of the networks themselves.

Ensuring a high level of security in wireless technologies is an important factor in supporting digital transformation in various sectors of modern society. Security in wireless technology not only protects data and networks, but also influences innovation, and the development of new industries, and drives technological progress, making this aspect critical to today's digital world.

Conclusions

1. Ensuring a high level of IS of wireless technologies not only guarantees the protection of information and related networks. On the other hand, this also has a crucial importance for stimulating innovation and the development of new industries. This data protection activity is essential to the further development of various innovations where wireless technologies become an integral aspect of our daily lives.

2. The security of wireless technologies requires a combination of technical and organizational strategies to effectively prevent possible attacks and ensure the required level of integrity and confidentiality of user data.

3. The development of new information technologies, such as IoT, quantum cryptography, multi-factor biometric authentication systems, and wide integration of AI/ML and AR/VR solutions, indicate the need for continuous improvement of existing strategies and security measures in the field of development and implementation of new wireless technologies.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Kolovanova, E., Melkoziorova, O., & Malakhov, S. (2023). The specifics of exploits and the particularities of countering this threat. *Proceedings of the XXIX International Scientific and Practical Conference.* July 25-27 2023, Warsaw, Poland. 216-224. https://doi.org/10.46299/ISG.2023.1.29 [in Ukrainian]
2. Shi, Q. (2019). Edge computing-enabled internet of things: A review, challenges and open issues. *IEEE Internet of Things Journal*, 6(5), 1615-1630. https://doi.org/10.1109/jiot2019.2892052
3. Elkhodr, M., (2019). A systematic review of industrial wireless sensor networks applications in oil and gas, agriculture and water treatment. *IEEE Access*, (7), 116623-116634. https://doi.org/10.1016/j.csi.2011.03.004
4. Onishchenko, Y., Chukalov, K., Geldt, S., & Kalancha, A. (2023). Methodology of evil websites and add-ons using SQL-injection and countering it. *Proceedings of the XII International Scientific and Practical Conference.* March 28-31, 2023. Florence, Italy. 409-414. https://doi.org/10.46299/ISG.2023.1.12 [in Ukrainian]
5. Kunz, M., Puchta, A., Groll, S., Fuchs, L., & Pernul, G. (2019). Attribute quality management for dynamic identity and access management. *Journal of Information Security and Applications*, (44), 64–79. https://doi.org/10.1016/j.jisa.2018.11.004
6. Earle, A. E., Frost, R. D. (2012). Wireless Security Handbook. (2nd ed.). New York: Auerbach Publications.
7. Muhammad Ehsan Rana, Mohamed Abdulla, Kuruvikulam Arun. (2007). Common Security Protocols for Wireless Networks: A Comparative Analysis. *IEEE Communications Magazine*, 45(4), 143-149. https://doi.org/10.2991/ahis.k.210913.080
8. Lee, I., Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), 431–440. https://doi.org/10.1016/j.bushor.2015.03.008
9. Pogorila, K., Bogdanova, E., & Kolovanova, E. (2022). An overview of the possibilities and specifics of the implementation of XDR technology, as a means of comprehensively counteracting current threats to information security. *Technologies, tools and strategies for the implementation of scientific research: materials of the IV International Scientific Conference*. Zhovten 7, 2022. Vinnytsia: European Science Platform. https://doi.org/10.46299/ISG.2023.1.22 [in Ukrainian]

10. K. Ramesh Rao, Dr. S.N. Tirumala Rao, Prof. P. Chenna Reddy. (2017) Wireless Communication Security and Privacy issues and Challenges. *International Journal of Computer Science and Information Security (IJCSIS)*, 15(7), 202-209. https://www.academia.edu/34148630/Wireless_Communication_Security_and_Privacy_issues_and_Challenges, ISBN 1947-5500
11. IEEE. (2020). Recommendations for Wireless Network Security. *IEEE Standards Association*. https://standards.ieee.org/ieee/802.11/7028/
12. Arjona, G., Garcia, M. P., Gil, J. A., Gómez, J. A. (2018). Enhancing Network Security Using Software-Defined Networking (SDN). *Journal of Cybersecurity and Privacy*, 1(1), 45-53. https://doi.org/10.3390/electronics12143077
13. Gorbenko, I., Gorbenko, Y., Yesina, M., & Ponomar, V. (2017). Propositions from the new level analysis and acceptance during the competition are decided to overcome new asymmetric post-quantum cryptographic primitives. *Computer Science and Cybersecurity*, (1), 53-70. ISBN: 2519-23-10 https://periodicals.karazin.ua/cscs/issue/view/577/827 [in Ukrainian]
14. Pogorila, K., Lesnaya, Y., Bogdanova, E., & Malakhov, S. (2022). Social engineering as a factor in the implementation of insider threats. *Scientific Collection "InterConf"*, (111), 494-501. https://archive.interconf.center/files/journals/3/issues/11/public/11-12-PB.pdf#page=495.%20ISBN%20978-1-0747-2337-8 ISBN 978- 1-0747-2337-8 [in Ukrainian]
15. Lesnaya, Yu., & Malakhov, S. (2023). Understanding the main changes in the implementation of phishing attacks. Proceedings of the XVII International Scientific and Practical Conference, 453-457. https://doi.org/10.46299/ISG.2023.1.17 [in Ukrainian]

БЕЗПЕКА В ЕПОХУ БЕЗДРОТОВИХ ІННОВАЦІЙ: АНАЛІЗ ПОТЕНЦІАЛЬНИХ ЗАГРОЗ ТА ЗАХОДИ ЗАХИСТУ

Євгенія Матвєєва[1], студентка факультету комп'ютерних наук (бакалаврат);
e-mail: belka.j.0507@gmail.com; ORCID: https://orcid.org/0000-0001-8801-2185
Марина Єсіна[1,2], кандидат технічних наук, доцент; e-mail: m.v.yesina@karazin.ua;
ORCID: https://orcid.org/0000-0002-1252-7606
Олександр Шумов[2], технічний директор АТ «ІІТ»; e-mail: alex.shumoff@gmail.com
[1]*Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, 61022, Україна*
[2]*АТ «ІІТ», вул. Коломенська, 15, Харків, 61166, Україна*
Рукопис надійшов 2 квітня 2024 р. Отримано після рецензування 2 травня 2024 р.
Прийнято 3 червня 2024 р.

Анотація: У сучасному взаємопов'язаному світі технології бездротової передачі даних бездоганно інтегровані в структуру сучасних бізнес-операцій. Зі зростанням довіри до цих технологій зростає необхідність забезпечення надійних заходів кібербезпеки. Особливо в епоху бездротових інновацій, прикладом яких є поширення Інтернету речей (IoT), дискурс навколо безпеки бездротових технологій підкреслює необхідність розуміння як усталених загроз, так і постійної появи нових вразливостей. Це підкреслює нагальну необхідність своєчасного виявлення та стратегій пом'якшення. Хоча зручність, яку забезпечують бездротові технології передачі даних, надає суспільству безпрецедентний доступ до інформації та полегшує керування різноманітними пристроями, процесами та системами, вона також наражає користувачів і сучасні інформаційно-комунікаційні системи (ICS) на серйозні кіберзагрози та вразливості. Отже, виникає нагальна потреба у комплексному вирішенні цих викликів. У цьому дослідженні розглядаються сучасні методології, спрямовані на обмеження доступу до бездротових мереж, виявлення потенційних вразливостей і створення ефективної відповіді на

кібератаки. Дана робота розглядає різні аспекти кібербезпеки, включаючи шифрування даних, механізми автентифікації користувачів, протоколи моніторингу трафіку та алгоритми виявлення аномалій. Крім того, робота звертає увагу на найважливіший аспект навчання персоналу методам безпеки бездротового зв'язку, оснащення його засобами поінформованості про загрози та реагування на інциденти. Враховуючи динамічний ландшафт технологій і загроз кібербезпеки, ця робота спрямована на встановлення базового розуміння ландшафту безпеки в бездротових мережах. Окреслюються прагматичні стратегії для ефективного управління ризиками безпеки, тим самим зміцнюючи стійкість сучасних організацій і захищаючи критичні інформаційні активи.

Ключові слова: *бездротові технології, інформаційна безпека, Інтернет речей (IoT), вразливості, автентифікація, квантова криптографія*

Конфлікт інтересів: автори повідомляють про відсутність конфлікту інтересів.

Список літератури:

1. Колованова, Є., Мелкозьорова, О., & Малахов, С. (2023). Специфіка використання експлойтів та особливості протидії цій загрозі. *Proceedings of the XXIX International Scientific and Practical Conference*. July 25-27 2023, Warsaw, Poland. 216-224. https://doi.org/10.46299/ISG.2023.1.29
2. Shi, Q. (2019). Edge computing-enabled internet of things: A review, challenges and open issues. IEEE Internet of Things Journal, 6(5), 1615-1630. https://doi.org/10.1109/jiot2019.2892052
3. Elkhodr, M., (2019). A systematic review of industrial wireless sensor networks applications in oil and gas, agriculture and water treatment. *IEEE Access*, (7), 116623-116634. https://doi.org/10.1016/j.csi.2011.03.004
4. Онищенко, Ю., Чукалов, К., Гельдт, С., & Каланча, А. (2023). Методологія зломів вебсайтів й додатків за допомогою SQL-injection та протидія ним. *Proceedings of the XII International Scientific and Practical Conference*. March 28-31, 2023. Florence, Italy. 409-414. https://doi.org/10.46299/ISG.2023.1.12
5. Kunz, M., Puchta, A., Groll, S., Fuchs, L., & Pernul, G. (2019). Attribute quality management for dynamic identity and access management. *Journal of Information Security and Applications*, (44), 64–79. https://doi.org/10.1016/j.jisa.2018.11.004
6. Earle, A. E., Frost, R. D. (2012). Wireless Security Handbook. (2-nd ed.). New York: Auerbach Publications.
7. Muhammad Ehsan Rana, Mohamed Abdulla, Kuruvikulam Arun. (2007). Common Security Protocols for Wireless Networks: A Comparative Analysis. *IEEE Communications Magazine*, 45(4), 143-149. https://doi.org/10.2991/ahis.k.210913.080
8. Lee, I., Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), 431–440. https://doi.org/10.1016/j.bushor.2015.03.008
9. Погоріла, К., Богданова, Є., & Колованова, Є. (2022). Огляд можливостей та узагальнення специфіки реалізації XDR-технології, як засобу комплексної протидії актуальним загрозам інформаційної безпеки. *Технології, інструменти та стратегії реалізації наукових досліджень: матеріали IV Міжнародної наукової конференції*. Жовтень 7, 2022. Вінниця: Європейська наукова платформа. https://doi.org/10.46299/ISG.2023.1.22
10. K. Ramesh Rao, Dr. S.N. Tirumala Rao, Prof. P.Chenna Reddy. (2017) Wireless Communication Security and Privacy issues and Challenges. *International Journal of Computer Science and Information Security (IJCSIS)*, 15(7), 202-209. https://www.academia.edu/34148630/Wireless_Communication_Security_and_Privacy_issues_and_Challenges ISBN 1947-5500
11. IEEE. (2020). Recommendations for Wireless Network Security. *IEEE Standards Association*. https://standards.ieee.org/ieee/802.11/7028/
12. Arjona, G., Garcia, M. P., Gil, J. A., Gómez, J. A. (2018). Enhancing Network Security Using Software-Defined Networking (SDN). *Journal of Cybersecurity and Privacy,* 1(1), 45-53. https://doi.org/10.3390/electronics12143077

13. Горбенко, І., Горбенко, Ю., Єсіна, М., & Пономар, В. (2017). Пропозиції з виконання порівняльного аналізу та прийняття в процесі конкурсу рішень щодо переваг певних асиметричних постквантових криптографічних примітивів. *Комп'ютерні науки та кібербезпека*, (1), 53-70. https://periodicals.karazin.ua/cscs/issue/view/577/827 ISBN: 2519-23-10
14. Погоріла, К., Лєсная, Ю., Богданова, Є., & Малахов, С. (2022). Соціальний інжиніринг, як фактор реалізації інсайдерських загроз. *Scientific Collection «InterConf»*, (111), 494-501. https://archive.interconf.center/files/journals/3/issues/11/public/11-12-PB.pdf#page=495.%20ISBN%20978-1-0747-2337-8 ISBN 978-1-0747-2337-8
15. Лєсная, Ю., & Малахов, С. (2023). Узагальнення основних передумов реалізації фішингових атак. *Proceedings of the XVII International Scientific and Practical Conference*, 453-457. https://doi.org/10.46299/ISG.2023.1.17

ISSN 2519-2310 (Online)

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА, 2024, № 1 (25)

COMPUTER SCIENCE AND CYBERSECURITY, 2024, Iss. 1 (25)

DOI: https://doi.org/10.26565/2519-2310-2024-1-04
УДК 004.056.5

КЛАСТЕРИЗАЦІЯ ТА КЛАСИФІКАЦІЯ ЧАСОВИХ ЗВУКОВИХ РЯДІВ

Станіслав Качанов[1], аспірант, e-mail: staskachanov2000@gmail.com,
ORCID: https://orcid.org/0009-0002-6938-6717
Дмитро Власенко[1], старший викладач кафедри теоретичних та прикладних комп'ютерних наук, кандидат математичних наук, e-mail: vlasenkod@karazin.ua,
ORCID: https://orcid.org/0009-0006-8780-2066
[1]*Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, 61022, Україна*

Рукопис надійшов 17 березня 2024 р. Отримано після рецензування 19 квітня 2024 р.
Прийнято 20 травня 2024 р.

Анотація: Було розглянуто дві важливі задачі в аналізі даних – класифікація та кластеризація часових рядів на прикладі звукових записів серцебиття людей. Однією з основних проблем аналізу часових рядів є складність порівняння різних рядів через їх варіативність у довжині, формі та амплітуді коливань. Для вирішення цих задач були використані різні алгоритми, серед яких рекурентна нейронна мережа з довгою короткочасною пам'яттю (LSTM) і алгоритм k найближчих сусідів для класифікації, та метод k-середніх (K-means) і DBSCAN для кластеризації. Результати дослідження показали, що LSTM є потужним інструментом для класифікації часових рядів завдяки здатності зберігати інформацію про контекст у часі. KNN, з іншого боку, продемонстрував високу точність і швидкість класифікації, однак його обмеження проявилися в умовах великих наборів даних. Для задач кластеризації, метод K-means виявився більш ефективним у порівнянні з DBSCAN, демонструючи вищу якість кластеризації за метриками силуету, Rand Score та іншими. Дані для дослідження були отримані з архіву часових рядів UCR, що включає звукові записи серцебиття різних категорій. Аналіз результатів показав, що обрані методи класифікації та кластеризації можуть бути ефективно використані для діагностики серцевих захворювань. Крім того, це дослідження відкрило нові можливості для подальшого вдосконалення методів обробки та аналізу даних, зокрема, для розробки нових інструментів медичної діагностики. Таким чином, ця робота демонструє ефективність використання алгоритмів машинного навчання для аналізу часових рядів та їх значення для покращення діагностики серцево-судинних захворювань.

Ключові слова: *класифікація часових рядів, кластеризація часових рядів, рекурентна нейронна мережа, LSTM, KNN, K-means, DBSCAN, аналіз звукових даних, серцеві звуки, машинне навчання, діагностика серцевих захворювань*

Як цитувати: Качанов С., Власенко Д.. Кластеризація та класифікація часових звукових рядів. *Комп'ютерні науки та кібербезпека.* 2024; № 1(25): С. 42–52. https://doi.org/10.26565/2519-2310-2024-1-04

In cites: Kachanov S., Vlasenko D. (2024). Clustering and Classification of Time Series Sound Data. *Computer Science and Cybersecurity.* 1(25): 42–52. https://doi.org/10.26565/2519-2310-2024-1-04 (in Ukrainian)

1. Вступ

У статті буде досліджуються дві актуальні задачі: класифікація і кластеризація часових рядів, які є важливими завданнями в аналізі даних.

Класифікація та кластеризація часових рядів є складними завданнями, оскільки ряди можуть мати різну довжину, форму та амплітуду коливань, що робить непростим встановлення схожості між рядами. У зв'язку з цим, розроблено багато різних методів для вирішення цих завдань, включаючи методи на основі статистичних моделей, нейронних мереж та інші. Існує багато алгоритмів, призначених для класифікації часових рядів. В роботі досліджено основні з них, та обгрунтовано, чому саме ці алгоритми, моделі та методи є найкращими для вирішення цих задач.

2. Опис обраних алгоритмів та підходів для задач класифікації

У цьому дослідженні як основний підхід застосовується один із найефективніших методів для розв'язання поставленої задачі за різними оцінками — використання глибокого навчання, зокрема рекурентної нейронної мережі з довготривалою короткочасною пам'яттю (LSTM) [1].

LSTM є одним із найвідоміших типів рекурентних нейронних мереж (RNN) і використовується для обробки послідовних даних, таких як мовлення, текст та часові ряди. Головна проблема традиційних RNN полягає в їхній неспроможності зберігати інформацію на довгих часових відрізках у послідовності [10]. LSTM була розроблена для вирішення цієї проблеми; вона має спеціальну структуру, яка дозволяє зберігати та використовувати інформацію протягом тривалих періодів часу. У звичайних нейронних мережах для класифікації зазвичай використовується пряме поширення сигналу (feedforward), де кожен вхідний сигнал обробляється окремо. Такі моделі не враховують динаміку змін у часі, оскільки не мають пам'яті про попередні вхідні дані.

Натомість LSTM здатна зберігати та використовувати попередні стани внутрішніх блоків, що називається пам'яттю LSTM. Це дозволяє моделі зберігати інформацію про попередній контекст і розуміти часові залежності між даними. Таким чином, LSTM може робити передбачення на основі повної історії часового ряду, враховуючи всі попередні значення, що робить її ефективною в задачах класифікації часових рядів. Крім того, LSTM може автоматично визначати, яку інформацію слід забути, а яку зберегти в пам'яті, що дозволяє моделі відкинути зайві дані, які можуть заважати правильній класифікації.

Для класифікації часових рядів за допомогою LSTM необхідно спочатку створити модель LSTM з відповідними вхідними та вихідними шарами [2]. Вхідний шар повинен мати розмірність (кількість часових кроків, кількість ознак), де кількість кроків відповідає довжині часового ряду, а кількість ознак — кількості характеристик у кожному кроці часу. У цьому дослідженні використовується набір даних із 5 класами, тому вихідний шар представляє ймовірності належності до певного класу [10].

Для порівняння також обрано алгоритм k-найближчих сусідів (k-Nearest Neighbors, KNN) — метод машинного навчання без учителя, який використовується для класифікації та регресії. Для класифікації нового часового ряду KNN знаходить k найближчих часових рядів із навчального набору, використовуючи відстань між векторами, яку можна обчислити за допомогою різних метрик, таких як евклідова або манхеттенська відстані. Класифікація нового часового ряду здійснюється шляхом голосування найближчих k сусідів. Однією з головних переваг KNN у класифікації часових рядів є його простота та ефективність [3]. Він не вимагає великої кількості навчальних даних і може добре працювати із зашумленими даними. Крім того, KNN може ефективно працювати з великими наборами даних, оскільки не потребує часу на

тренування моделі.

Однак KNN має й деякі недоліки [4]. Він може бути чутливим до викидів у даних, оскільки не враховує структуру даних і може помилково класифікувати тестові зразки, якщо навчальний набір містить зміщені дані або викиди. Щодо застосування KNN для класифікації часових рядів, цей метод може бути особливо ефективним, оскільки допомагає знаходити схожість між часовими рядами та використовувати цю інформацію для класифікації нових даних.

Таким чином, вибір методу для класифікації часових рядів повинен базуватися на конкретній задачі та характеристиках даних, які потрібно аналізувати.

3. Опис обраних алгоритмів та підходів для задач кластеризації

Метод к-середніх (K-means) є одним з найпоширеніших методів кластеризації [5], який дозволяє розділити множину даних на кластери на основі схожості між їх елементами.

Використання методу к-середніх для кластеризації часових рядів полягає в тому, щоб розділити множину часових рядів на кластери на основі їх схожості. Для цього, зазвичай, використовують такі метрики, як Евклідова відстань або манхеттенська відстань, або косинусна відстань між векторами. Крім того, можна використовувати різні методи для побудови векторів-ознак для часових рядів, такі як метод головних компонент.

У задачах класифікації часових рядів метод к-середніх можна використовувати для попередньої кластеризації даних та подальшого застосування методів класифікації для кожного кластеру окремо. Це може допомогти поліпшити якість класифікації та знизити час роботи алгоритму [6].

Для порівняння було обрано інший алгоритм кластеризації - DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

Одна з головних переваг DBSCAN полягає в тому, що він може працювати з даними різної густини та форми кластерів, і він добре підходить для кластеризації часових рядів, де зазвичай зустрічаються складні форми та різні рівні густини даних. Крім того, DBSCAN може ідентифікувати шумові точки, які не належать до жодного кластера.

Однак, існує деяка кількість недоліків, пов'язаних з використанням DBSCAN. Наприклад, якщо розмір набору даних дуже великий, то алгоритм може стати надто повільним (обраний набір даних не є дуже об’ємним, тому алгоритм виконується досить швидко).

У порівнянні з методом k-середніх, DBSCAN має деякі переваги. Наприклад, DBSCAN може працювати з даними з різною густотою та формою, тоді як k-середніх передбачає, що кластери мають сферичну форму та рівномірну густину [7].

Можна заявити з упевненістю, що DBSCAN - це потужний метод кластеризації часових рядів, який дозволяє виявляти кластери будь-якої форми та не вимагає заздалегідь визначених кількості кластерів. Його недоліки полягають у потребі в налаштуванні гіперпараметрів та високих обчислювальних витратах порівняно з методом k-середніх, але варто зазначити, що і в методі k-середніх підбір гіперпараметрів теж відіграє важливу роль.

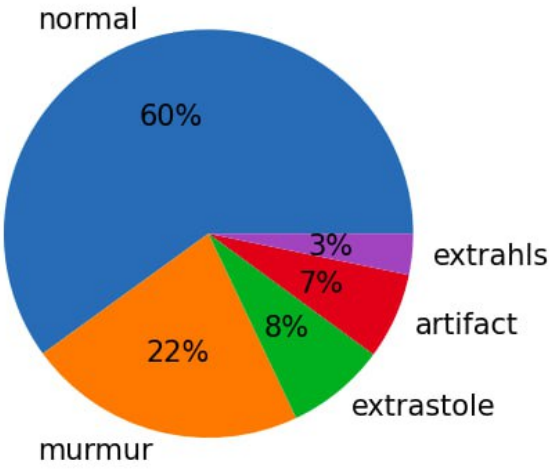
4. Актуальність обрання даних та вирішення цієї задачі, опис даних

Для обрання максимально релевантних даних, було вирішено обрати відомий архів, в якому зібрані дані за останні 20 років (синтетичні та натуральні), які найкраще підходять для наукових досліджень. Архів часових рядів UCR [8], запроваджений у 2002 році, став важливим ресурсом для спільноти з видобутку даних часових рядів.

Аудіофайли мають різну довжину від 1 до 30 секунд (деякі були обрізані, щоб зменшити надмірний шум та надати виокремлений фрагмент звуку). Більшість інформації у звуках серця

міститься у низькочастотних компонентах, а шум - у вищих частотах. Кожен файл відноситься до однієї із категорій даних:

- **Нормальний звук (normal).** У категорії «Нормальний звук» є нормальні, здорові звуки серця. Вони можуть містити шум у останній секунді запису, коли прилад віднімається від тіла. Вони можуть містити різноманітні фонові шуми (від транспорту до радіо). Вони також можуть містити випадковий шум, що відповідає диханню або торканню мікрофона одягом або шкірою.
- **Категорія бурчання (murmur).** Серцеві шуми звучать так, ніби є «свист, рев, гуркіт або бурхлива рідина» в одному з двох тимчасових місць: (1) між «луб» і «даб» або (2) між «даб» і «луб». Вони можуть бути симптомом багатьох захворювань серця, деякі серйозні.
- **Категорія додаткового серцевого звуку (extrahls).** Додатковий серцевий звук може не бути ознакою хвороби. Однак у деяких ситуаціях це важливий ознака хвороби, яку, якщо виявити рано, може допомогти людині. Додатковий серцевий звук важливо виявляти, оскільки його не можна добре виявити ультразвуком.
- **Категорія «Артефакт» (artifact).** В цій категорії є широкий спектр різних звуків, включаючи звук зворотнього зв'язку і ехо, мову, музику та шум. Зазвичай немає відчутних звуків серця і має мало або жодної тимчасової періодичності на частотах нижче 195 Гц. Ця категорія найбільш відрізняється від інших. Відрізняти цю категорію від чотирьох інших, дуже важливо щоб той, хто збирає дані, здійснив повторну спробу.
- **Категорія «Надзвичайний серцевий ритм» (extrastole).** Звуки цього ритму можуть з'являтися час від часу і можуть бути визначені тим, що серцевий ритм порушений через додаткові або пропущені серцеві скорочення (це не те саме, що додатковий серцевий звук, оскільки ця подія не відбувається регулярно). Надзвичайний серцевий ритм може не бути ознакою хвороби, однак у деяких ситуаціях надзвичайні ритми можуть бути спричинені серцевими захворюваннями.



Діаграма 1 – Відсотковий розподіл даних за категоріями
Diagram 1 – Percentage Distribution of Data by Categories

На круговій діаграмі вище (Діаграма 1) видно розподіл даних за категоріями: найбільше нормальних звичайних записів серцебиття (60%), далі йде категорія «бурчання» (22%), за нею записи з надзвичайним серцевим ритмом (8%), далі некоректні записи-артефакти (7%) і найменше з категорії додаткового серцевого звуку.

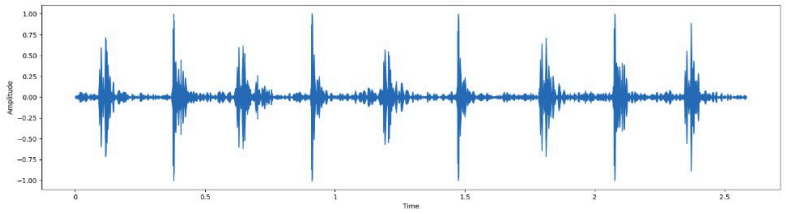


Рис. 1 – Звукова хвиля для нормального серцебиття (normal)
Fig. 1 – Sound Wave for Normal Heartbeat (normal)

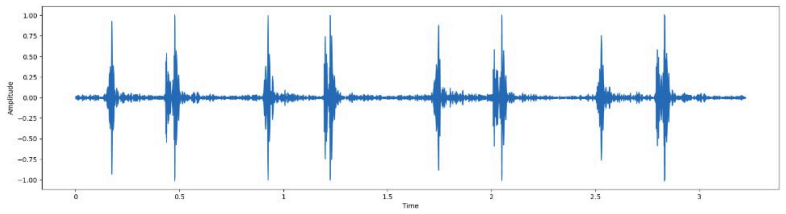


Рис. 2 – Звукова хвиля для серцебиття з бурчанням (murmur)
Fig. 2 – Sound Wave for Heartbeat with Murmur (murmur)

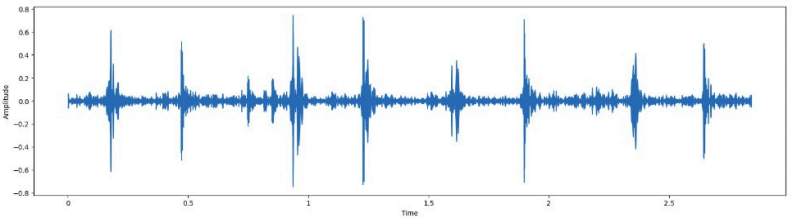


Рис. 3 – Звукова хвиля для серцебиття з надзвичайним серцевим ритмом (extrastole)
Fig. 3 – Sound Wave for Heartbeat with Extrastole (extrastole)

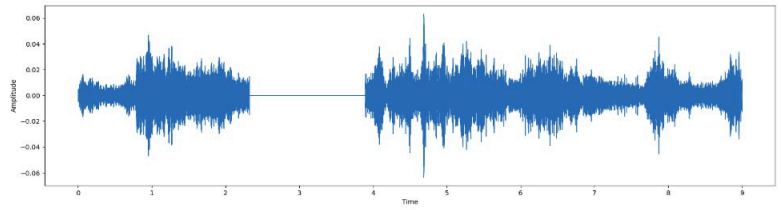


Рис. 4 – Звукова хвиля для неправильних записів (artifacts)
Fig. 4 – Sound Wave for Incorrect Recordings (artifacts)

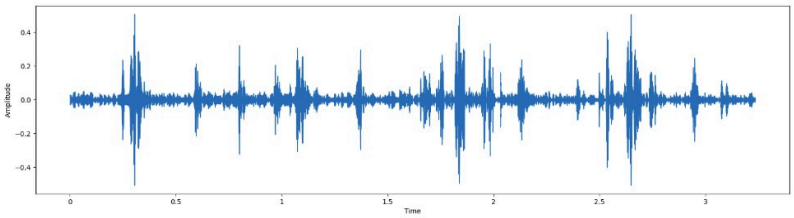


Рис. 5 – Звукова хвиля для серцебиття з додатковим серцевим звуком (extrahls)
Fig. 5 – Sound Wave for Heartbeat with Additional Heart Sound (extrahls)

На Рис. 1 добре видно яскравий цикл при нормальному серцебитті. Навіть візуально видно той самий нормальний звук серця, який має чіткий шаблон «луб даб, луб даб», це і є цикл нормального серцебиття.

На Рис. 2 теж видно те саме «бурчання», паузи між «луб» і «даб» нерівномірні, пікові значення та їх довжина різна.

На Рис. 3 теж приблизно проглядаються додаткові або пропущені серцеві скорочення, ті самі, «луб-луб-даб» або «луб-даб-даб». А на Рис. 4 явна демонстрація некоректних даних, які не відображають хоч якесь серцебиття.

На Рис. 5 більше пікових значень, тобто якраз і видно цей додатковий серцевий ритм.

5. Побудова моделей для класифікації і кластеризації часових рядів

Грунтуючись на багатьох дослідженнях було обрано по 2 [6], найкращих для цієї задачі, алгоритма кластеризації і класифікації, а саме методи K-means і DBSCAN для кластеризації часових рядів, а KNN і LSTM для класифікації.

Оскільки дані розмічені, а задачі кластеризації відносяться до так званого unsupervised learning (навчання без учителя), то відповідно видалено всі таргети для застосування кластеризації.

Для методу K-means основним параметром є кількість кластерів K. Для значення 5 результати були найкращі.

Для алгоритму DBSCAN аналогічно видалено таргети. Через не дуже великі об’єми даних DBSCAN навчається досить швидко, бо повільність навчання - є його відомим недоліком.

Алгоритм KNN ефективно відпрацював з гіперпараметрами за умовчанням, і як показав подальший аналіз для цих даних він є ефективним у співвідношенні якість/швидкості.

І в кінці проаналізовано найскладніший, але при цьому найефективніший підхід – рекурентну нейронну мережу LSTM.

Це багатошарова рекурентна мережа із функціями активації ReLU, а вже на вихідному шарі активація Softmax.

При використанні меншої кількості шарів модель мала меншу ефективність, але при цьому через велику загальну кількість параметрів LSTM навчалась дуже довго.

6. Аналіз отриманих результатів кластеризації часових рядів

Для усіх результатів ми виокремимо дві категорії: результати кластеризації і результати класифікації. Очевидно, що через різні підходи відповідно будуть різні метрики оцінювання якості алгоритмів і моделей.

Для алгоритмів DBSCAN і K-means обрано чотири метрики, які однаково можуть використовуватись як для K-середніх, так і для DBSCAN: silhouette_score, adjusted_rand_score, davies_bouldin_score та adjusted_mutual_info_score. Вони є широко використовуваними метриками для оцінки якості кластеризації в машинному навчанні.

- Silhouette Score - оцінює наскільки схожі між собою об'єкти в середині кластера та наскільки вони відрізняються від об'єктів в інших кластерах.
- Adjusted Rand Score - оцінює наскільки схожі кластеризації на істинні мітки. Значення 1 означає, що кластери повністю збігаються з істинними мітками.
- Davies-Bouldin Score - оцінює суміш внутрішньокластерної схожості та зовнішньокластерної відмінності для кожного кластера та робить узагальнення для всієї кластеризації. Чим нижче значення, тим краща кластеризація.
- Adjusted Mutual Information Score - оцінює наскільки взаємозв'язок між кластеризацією та істинними мітками відмінний від того, який очікується випадковим чином. Чим більше значення, тим краща кластеризація.

Одна з причин, чому метрики силуета, адаптована взаємна інформація, відстань Девіса-Болдуїна та адаптований рандомізований індекс чистоти підходять для кластеризації часових рядів, полягає в тому, що вони оцінюють якість кластерів, а не якість класифікації.

Ці метрики оцінюють, наскільки добре об'єкти в кожному кластері схожі між собою, і наскільки відмінні вони від об'єктів інших кластерів. Це дуже важливо для часових рядів, оскільки вони мають складну структуру та можуть мати різні форми. Метрики також дозволяють оцінити, чи є розділення на кластери зрозумілим та придатним для подальшого аналізу.

Таким чином, ці метрики допомагають зробити висновки про якість кластеризації та знайти найкращі параметри для алгоритмів кластеризації.

Проаналізуємо результати:

- Для K-середніх:
- Silhouette Score = 0.345
- Adjusted Rand Score = 0.006
- Davies-Bouldin Score = 0.876
- Adjusted Mutual Information Score = 0.049

Для DBSCAN:

- Silhouette Score = -0.6336115
- Adjusted Rand Score = 0.013803153550193674
- Davies-Bouldin Score = 1.339207713289357
- Adjusted Mutual Information Score = 0.037943281139379566

За результатами метрик можна сказати, що алгоритм K-means працює краще, ніж DBSCAN, для цього конкретного набору даних.

Він досягнув значень Silhouette Score близько до 0.34, що свідчить про те, що кластери добре відокремлені один від одного. Оцінка Adjusted Rand Score для K-means низька, а це може бути пов'язано з тим, що дані можуть містити шум, або кластеризація може бути досить складною.

Davies-Bouldin Score для K-means більше 0.87, що вказує на те, що кластери забезпечують добру відмінність один від одного, але можуть бути не такими оптимальними, як би ми хотіли. Adjusted Mutual Information Score для K-means також низький, що свідчить про низький рівень взаємозалежності між вхідними даними та отриманими кластерами.

З іншого боку, результати метрик для DBSCAN не такі високі [9]. Silhouette Score близький до -0.63, що показує, що кластери майже не відокремлені один від одного.

Оцінка Adjusted Rand Score для DBSCAN дещо вища, ніж для K-means, але все ще низька. Це означає, що кластери можуть мати різну кількість елементів і не відповідати оригінальним міткам. Davies-Bouldin Score для DBSCAN більше 1.33, що свідчить про те, що кластери не дуже відокремлені один від одного. Adjusted Mutual Information Score для DBSCAN також низький, що свідчить про низький рівень взаємозалежності між вхідними даними та отриманими кластерами.

Отже, можна зробити висновок, що алгоритм K-means більш ефективний для даного набору даних, оскільки досягнув більш високих значень метрик, але також варто зазначити, що результати метрик все рівно є високими, що свідчить про правильно обрані алгоритми та методи для кластеризації записів людського серцебиття.

7. Аналіз отриманих результатів класифікації часових рядів

Для оцінки алгоритму класифікації було обрано стандартні метрики для класифікації: accuracy, f1, precision, recall та матриця помилок (confusion matrix).

Метрики accuracy, f1, precision, recall та матриця помилок (confusion matrix) дуже добре підходять для задач класифікації часових рядів через те, що вони дають змогу оцінити якість класифікації на різних рівнях: загальну точність (accuracy), точність визначення позитивних класів (precision), точність визначення негативних класів (recall) та збіги та розбіжності в класифікації кожного з класів (confusion matrix). Враховуючи, що класифікація часових рядів є завданням з високою вимогою до точності та чутливості, використання цих метрик є дуже важливим для оцінки результатів роботи алгоритмів.

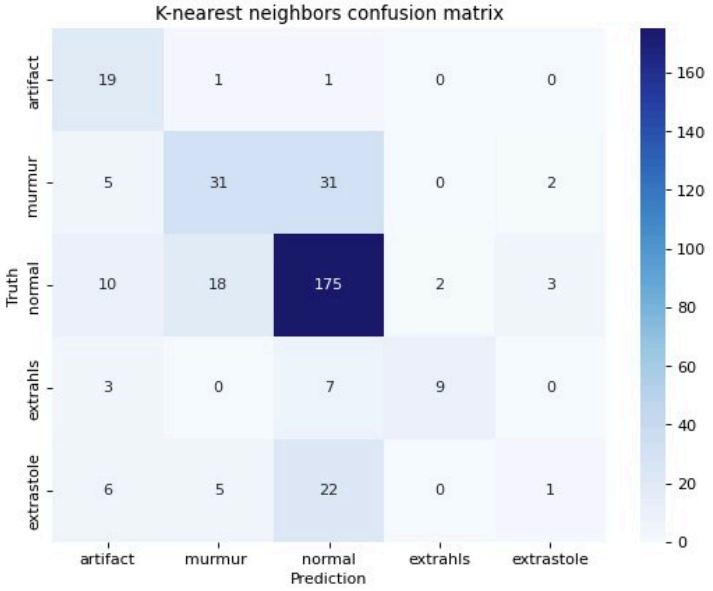


Рис. 6 – Матриця помилок для алгоритму KNN
Fig. 6 – Confusion Matrix for the KNN Algorithm

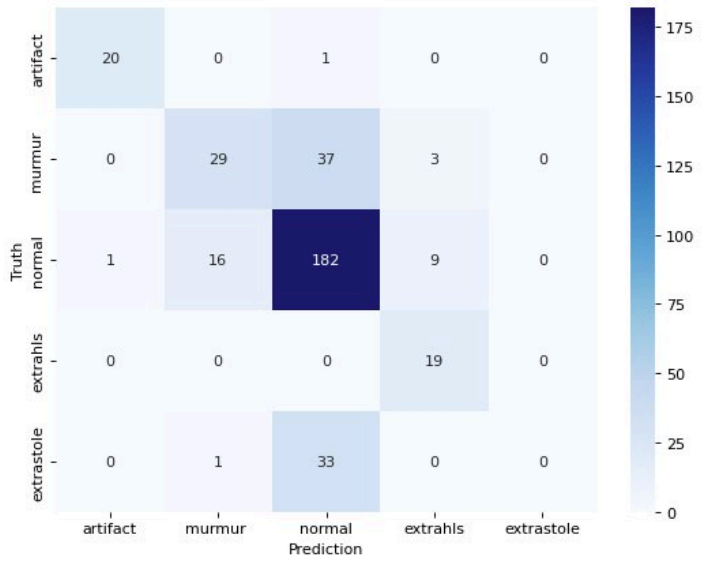


Рис. 7 – Матриця помилок для моделі LSTM
Fig. 7 – Confusion Matrix for the LSTM Model

Для KNN:

- Accuracy: 0.658,
- F1 score: 0.651,

- Recall: 0.658,
- Precision: 0.667

Для LSTM:

- Accuracy: 0.681,
- F1 score: 0.693,
- Recall: 0.681,
- Precision: 0.713

Дані результати метрик Accuracy, F1 score, Recall та Precision є високими для алгоритму K-Nearest Neighbors (KNN) у багатокласовій класифікації часових рядів.

Значення Accuracy вказує на те, що наш алгоритм правильно класифікував 65,8% зразків датасету, що є досить високим результатом. F1 score є гармонійним середнім між точністю (Precision) та повнотою (Recall), і також демонструє високу точність та повноту класифікації нашого алгоритму. Значення Precision та Recall становлять відповідно 0,667 та 0,658, що також є досить високими показниками.

Отже, високі результати метрик свідчать про ефективність використання алгоритму KNN для багатокласової класифікації часових рядів.

Як ми бачимо, результати метрик для LSTM не набагато краще, і може здатись, що алгоритм KNN, який простіший і швидший, краще підходить для даного набору даних, але це не так. Основна проблема полягає в тому, що комп’ютер, на якому проводились обчислення, не міг обробити велику кількість епох. Таким чином можна стверджувати, що отримані результати для LSTM можна значно покращити.

8. Висновки

1. Для кластеризації були використані алгоритми k-means та DBSCAN, що дозволило розділити записи на кілька категорій залежно від характеристик звуків. Для класифікації були використані алгоритми KNN та LSTM, що дозволило відрізнити звукові записи різних категорій та визначити, до якої категорії відноситься конкретний запис.

2. Отримані результати свідчать про ефективність використаних методів для аналізу звукових записів серцебиття людей та можуть бути використані для діагностики різних захворювань серця. Дослідження можуть бути продовжені з використанням інших алгоритмів та наборів даних з метою поліпшення точності класифікації та кластеризації.

3. Дослідження продемонструвало, що кластеризація та класифікація часових рядів з використанням алгоритмів k-means, DBSCAN, KNN і LSTM є ефективним методом для аналізу даних серцевих звуків.

4. Алгоритм k-means дозволяє кластеризувати дані серцевих звуків за їх характеристиками та дозволяє виявляти спільні риси між різними звуками. DBSCAN може бути корисним у виявленні аномальних звуків та відокремленні їх від нормальних. Алгоритм KNN забезпечує ефективну класифікацію звуків за їх характеристиками, тоді як LSTM може використовуватися для класифікації звуків на основі їх часових характеристик.

5. Отже, в даній роботі було успішно використано кластеризацію та класифікацію часових рядів для аналізу даних зі звуковими записами серцебиття людей, ділячи їх на категорії. Результати цієї роботи можуть бути корисними для медичних досліджень та можуть допомогти у розробці нових методів діагностики та лікування серцевих захворювань, а якщо їх продовжувати і розвивати, то і стати справді революційним методом у діагностуванні та виявленні захворювань серця і судинної системи.

Конфлікт інтересів

Автори повідомляють про відсутність конфлікту інтересів.

References

1. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735
2. Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10), 2222-2232. https://doi.org/10.1109/TNNLS.2016.2582924
3. Zhang, Z. (2004). Nearest neighbor search algorithms and applications. Springer. https://doi.org/10.1007/978-3-319-14717-8_39
4. Dasarathy, B. V. (1991). Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Press.
5. Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media. https://doi.org/10.1007/978-0-387-84858-7
6. Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645–678. https://doi.org/10.1109/TNN.2005.845141
7. Martin Ester, Jörg Sander (1996). "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications". *Data Mining and Knowledge Discovery*. 2 (2): 169–194. https://doi.org/10.1007/BF00457189
8. Hoang A.D., Bagnall A., Kaveh K., Chin-Chia M.Y., Zhu Y., Shaghayegh G., Chotirat A.R., Eamonn K. The UCR Time Series Archive URL: arxiv.org/abs/1810.07758
9. Schubert, E., Sander, J., Ester, M., Kriegel, H.-P., & Xu, X. (2017). "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN". *ACM Transactions on Database Systems (TODS)*, 42(3), 19. https://doi.org/10.1145/3068335
10. Kachanov Stanislav (2024) *Clustering and Classification of Time Series Data* (master diploma) V. N. Karazin Kharkiv National University

CLUSTERING AND CLASSIFICATION OF TIME SERIES SOUND DATA

Stanislav Kachanov[1], PhD Student; e-mail: staskachanov2000@gmail.com;
ORCID: https://orcid.org/0009-0002-6938-6717
Dmytro Vlasenko[1], senior lecturer of the Department of Theoretical and Applied Computer Sciences, PhD in mathematics; e-mail: vlasenkod@karazin.ua; ORCID: https://orcid.org/0009-0006-8780-2066
[1] *V. N. Karazin Kharkiv National University, Ukraine*

Manuscript was received March 17, 2024; Received after review April 19, 2024; Accepted May 20, 2024

Abstract. This scientific article addresses two critical tasks in data analysis—time series classification and clustering, particularly focusing on heart sound recordings. One of the main challenges in analyzing time series lies in the difficulty of comparing different series due to their variability in length, shape, and amplitude. Various algorithms were employed to tackle these tasks, including the Long Short-Term Memory (LSTM), KNN, recurrent neural network for classification and the K-means and DBSCAN methods for clustering. The study emphasizes the effectiveness of these methods in solving classification and clustering problems involving time series data containing heart sound recordings. The results indicate that LSTM is a powerful tool for time series classification due to its ability to retain contextual information over time. In contrast, KNN demonstrated high

accuracy and speed in classification, though its limitations became apparent with larger datasets. For clustering tasks, the K-means method proved to be more effective than DBSCAN, showing higher clustering quality based on metrics such as silhouette score, Rand score, and others. The data used in this research were obtained from the UCR Time Series Archive, which includes heart sound recordings from various categories: normal sounds, murmurs, additional heart sounds, artifacts, and extra systolic rhythms. The analysis of results demonstrated that the chosen classification and clustering methods could be effectively used for diagnosing heart diseases. Furthermore, this research opens up new opportunities for further improvement in data processing and analysis methods, particularly in developing new medical diagnostic tools. Thus, this work illustrates the effectiveness of machine learning algorithms for time series analysis and their significance in improving cardiovascular disease diagnosis.

Keywords: *time series classification, time series clustering, recurrent neural network, LSTM, KNN, K-means, DBSCAN, sound data analysis, heart sounds, machine learning, heart disease diagnosis*

Conflicts of Interest: the authors declare no conflict of interest.

ISSN 2519-2310 (Online)

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА, 2024, № 1 (25)

COMPUTER SCIENCE AND CYBERSECURITY, 2024, Iss. 1 (25)

DOI: https://doi.org/10.26565/2519-2310-2024-1-05
УДК 004.056.5

ФУНКЦІОНАЛЬНІ ОСОБЛИВОСТІ ВІДОМИХ ЗАСОБІВ МІЖМЕРЕЖЕВОГО ЕКРАНУВАННЯ

Михайло Січкар[1], студент бакалавр спеціальності «Комп'ютерні системи та мережі», кафедра захисту інформаційних систем та технологій, e-mail: sichkar2020kb13@student.karazin.ua
Миколай Карпінський[2], професор, e-mail: mikolaj.karpinski@up.krakow.pl,
ORCID: https://orcid.org/0000-0002-8846-332X
Сергій Малахов[1], доктор філософії, старший науковий співробітник, кафедра комп'ютерних наук, e-mail: malakhov@karazin.ua, ORCID: https://orcid.org/0000-0001-8826-1616
[1]*Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, 61022, Україна*
[2]*Інститут безпеки та комп'ютерних наук, Університет комісії національної освіти, 30-084 Краків, Польща*
Рукопис надійшов 2 квітня 2024 р. Отримано після рецензування 2 травня 2024 р.
Прийнято 3 червня 2024 р.

Анотація: В роботі коротко розглядається історія, типи та можливості основних типів фаєрволів (FW). Міжмережеві екрани є важливим засобом захисту мережевих ресурсів від різноманітних загроз інформаційній безпеці. З розвитком технологій і зміною характеру атак, особливо тих, що включають штучний інтелект (AI), брандмауери також еволюціонували, набуваючи нових функцій і можливостей. У цій роботі наведено короткий огляд основних типів та можливостей міжмережевих екранів, що забезпечують вирішення питань комплексного захисту мережевого обладнання та їх інформаційних ресурсів від сучасних загроз безпеки. Різні типи фаєрволів знаходять своє застосування в залежності від умов функціонування і призначення базової інформаційно-комунікаційної системи (ІКС), а також від місця їх (фаєрволів) інтеграції в мережеву чи віртуальну інфраструктуру сучасних інформаційних систем. Для інтегрованих мереж, що вимагають високого рівня їх безпеки, продуктивності і гнучкості, брандмауери бізнес-сегменту покоління Next-generation та Threat-focused NGFW, безумовно є кращим вибором. Звернено увагу на те, що мобільні фаєрволи повинні всіляко сприяти підтримці ресурсного консенсусу та усувати можливий диспаритет в продуктивності мережевих мобільних застосунків. Адаптивність до мобільності сучасних систем зв'язку (Wi-Fi, GSM та інші) визначає специфічність загроз безпеки для мобільних пристроїв та зумовлює їх ключову особливість. Ця особливість базується на перманентній готовності до безшовних переходів (перепідключень) між різними мережами в умовах постійного енергодефіциту та обмеженості наявних обчислювальних ресурсів (мається на увазі гаджетів). Висвітлено основні тенденції, перспективи розвитку та впровадження різних типів міжмережевих екранів, включаючи вплив штучного інтелекту, машинного навчання, хмарних технологій та Інтернету речей (IoT), а також важливі аспекти сфери їх (фаєрволів) застосування. Підкреслено, що впровадження FW не підміняє собою інших технологій і інструментів безпеки, а лише ефективно розширює наявний арсенал протидії новим загрозам безпеки (як інструмент проактивної протидії та швидкого реагування на складні мережеві інциденти). Стаття може

бути корисною для студентів, науковців та фахівців з інформаційної безпеки, які прагнуть розширити рівень своїх компетенцій, пов’язаних з розробкою і експлуатацією сучасних технологій міжмережевого захисту.

Ключові слова: *FW, фаєрвол, інформаційна безпека, загрози безпеки, зловмисне програмне забезпечення*

Як цитувати: Січкар М., Карпінський М., Малахов С.. Функціональні особливості відомих засобів міжмережевого екранування. *Комп’ютерні науки та кібербезпека.* 2024; № 1(25): С. 53–65. https://doi.org/10.26565/2519-2310-2024-1-05

In cites: Sichkar M., Karpinski M., Malakhov S. (2024). Functional features of well-known means of network shielding. *Computer Science and Cybersecurity.* 1(25): 53–65. https://doi.org/10.26565/2519-2310-2024-1-05 (in Ukrainian)

1. Вступ

У сучасному світі інформація є найціннішим ресурсом, тому питання безпеки функціонування сучасних інформаційно-комунікаційних систем (ІКС) стають критично важливим напрямом діяльності, оскільки масштаби та складність нових кіберзагроз дедалі тільки зростають. В рамках заходів з протидії сучасним мережевим атакам, важливу роль незмінно продовжують відіграють міжмережеві екрани (*фаєрволи, від англ. Firewall*). Ці засоби, безперечно, є одними з найефективніших та найпоширеніших інструментів захисту інформаційних ресурсів сучасних інформаційних систем та/чи окремих мобільних гаджетів пересічних користувачів, від різноманітних типів мережевих загроз [1-7]. Вони надають можливість управління доступом до різних сегментів мережі (*безвідносно цілей їх утворення*), аналізуючи циркуляцію мережевого трафіку, мережеву поведінку користувачів та мережеву активність відповідних додатків, завчасно виявляючи й блокуючи потенційно небезпечну (недекларовану) мережеву активність [3-8]. Враховуючи неперервний розвиток загроз безпеки і еволюцію технологій та засобів їх парирування, вочевидь, що розгляд питань, щодо узагальнення основних функціональних особливостей, та практики використання відомих рішень міжмережевого екранування, є безумовно актуальним.

2. Основна частина

В загальному випадку, фаєрвол - це фізичний пристрій або спеціалізоване програмне забезпечення, що контролює мережевий трафік між двома чи більше мережами та/або різними сегментами однієї і тієї ж мережі, згідно з встановлених для неї (*мережі чи сегменту мережі*) набором правил безпеки. Іншими словами, *Firewalls* використовують відповідні правила, щоб дозволити чи заборонити певним пакетам даних циркулювати між різними мережами та/чи різними сегментами однієї мережі. Як правило, файрволи встановлюють між локальною/внутрішньою мережею підприємства та Інтернет. Таке розміщення *Firewalls* покликане захистити внутрішні інформаційні ресурси корпоративної ІКС від шкідливого програмного забезпечення (ПЗ) та інших − «зовнішніх» загроз з Інтернету. Крім того, фаєрволи активно використовуються для контролю циркуляції трафіку всередині периметру безпеки корпоративної ІКС (*тобто за 1-м, вхідним Firewall*), наприклад: - для контролю мережевої активності на всіх внутрішніх шлюзах–мостах і блокування доступу внутрішніх користувачів до певних веб-сайтів та/або додатків.

В межах своїх функціональних задач, фаєрвол аналізує весь мережевий трафік, що циркулює в місті його інтеграції, де для кожного пакету даних здійснюється їх верифікація на відповідність наперед заданим правилам. Так, якщо пакет відповідає дозволеним − легітимним правилам мережевої активності, то фаєрвол транслює його далі. В іншому випадку, мережева активність припиняється у відповідності із заданими сценарієм поточних налаштувань кожного окремого *Firewalls* (*наприклад, блокування назавжди чи тимчасово (на певний час) або перенаправлення пакетів к іншому узлу/маршруту тощо*).

В цілому, правила функціонування *Firewalls* налаштовуються відповідно до потреб організації. Наприклад, корпоративною політикою інформаційної безпеки (ПІБ), може передбачатися доступ до Інтернету лише певним користувачам, комп'ютерам та/або програмам. Крім того, може бути заборонений доступ до певних зовнішніх інформаційних ресурсів (*веб-сайтів і онлайн сервісів*), так й деяких внутрішніх функцій (*наприклад, доступ до корпоративних принтерів для певної групи користувачів та ін.*).

Таким чином міжмережеві екрани є вкрай важливим елементом в загальній системі безпеки інформаційних ресурсів сучасних ІКС та забезпечують адаптивний захист мережевого устаткування від деструктивного впливу шкідливого ПЗ в дуже широкому спектрі загроз, причому, як зовнішнього, так і внутрішнього походження [4-5, 9]. Для цілісного розуміння принципів функціонування *різних* типів фаєрволів, важливо усвідомлювати їх «місце» і роль на відповідних рівнях моделі *OSI* та виокремлювати *основні етапи* розвитку технологій міжмережевого екранування, як окремої складової загального процесу еволюції інформаційних технологій.

Етапи еволюції засобів міжмережевого екранування та їх особливості.

У 1988 році компанія *«Digital Equipment Corporation»* (DEC) запропонувала перше покоління засобів міжмережевої фільтрації трафіку, відомих як «*Packet-Filter Firewall*» чи фаєрвол з фільтрацією пакетів [11]. У 1989 році з'явилось друге покоління, відоме як «*Stateful Firewall*». Третє покоління фаєрволів прикладного рівня, було визначено в 1991 році. У 2004 році «Міжнародна корпорація з обробки даних» (*IDC*) вперше використала термін «Об'єднаний брандмауер загроз» (*UTM - Unified Threat Management*). У 2009 році компанія *Gartner* запропонувала концепцію фаєрволу нового покоління, визначив її як «*Next-Generation Firewall*» (*NGFW*) [5, 10].

Фаєрволи 1-го покоління проводили аналіз пакетів інформації, які циркулювали між комп'ютерами в мережі. Правила фільтрації базувалися на різних параметрах, таких як адреси джерела та вузлу призначення, використовувані протоколи та номери портів на обох сторонах взаємодіючих комп'ютерів. При цьому, цей тип фаєрволів не враховував стан з'єднання пакету та не зберігав його стан. Тому його часто називали «брандмауерами без стану» (*Stateless Firewalls*). Вони операційно працювали на мережевому рівні моделі *OSI* [7] та були відомі, як фаєрволи «рівня мереж» (*Network Layer Firewalls*) [12].

У 1991 році, DEC представила рішення 3-го покоління (*SEAL - Secure External Access Link*), яке отримало назву «Брандмауер на рівні застосунків». Брандмауери на рівні застосунків (наприклад, *Gauntlet* від *Trusted Information Systems* та *FireWall-1* від *компанії Check Point*, у 1994 р.) керували трафіком застосунків, які підключалися до Інтернет та/чи інших «зовнішніх мереж» і адміністрували трафік на протоколах *FTP*, *Telnet* та *HTTP* [4, 10].

Як вже було зазначено вище, у 2004 році *IDC* запроваджує новий концепт мережевої безпеки - *UTM*, в межах якої еволюція «традиційних» брандмауерів (*контроль портів та протоколів*) трансформується у спробу створення комплексного рішення мережевої безпеки. *UTM* передбачає інтегроване використання різноманітних інструментів/засобів, таких як: − мережевий *FW*, фільтрація web-сторінок, шлюзовий антивірус, антиспам, *VPN* та ін. [7, 10].

У 2009 році Gartner® представляє нову концепцію - *Next-Generation FW*, яка в рамках одного рішення поєднує ідеї «традиційного» фаєрволу та нові технології, такі як: − системи виявлення і запобігання вторгненням (*IDS/IPS*); − глибока інспекція пакетів (*DPI*); − «пісочниця»; − управління застосунками; − фільтрація *URL*-адрес; − захист від поліморфного шкідливого ПЗ; − профілювання мережі; − політика ідентифікації; − *VPN* та ін. Головною особливістю *NGFW* є використання *DPI* на рівні застосунків, що відрізняє його від усіх попередніх рішень *FW*, які обмежувались моніторингом портів і протоколів [4-5, 7, 10, 12-13]. Комплексування в межах *NGFW* зазначених властивостей і функціоналу в значній мірі сприяє підвищенню загального рівня мережевої безпеки сучасних ІКС.

Цілком очевидно, що різні типи фаєрволів знаходять свою нішу застосування в залежності від умов функціонування й призначення базової ІКС [3, 7-8, 13-18] та місця їх (*FW*) інтеграції в мережевої чи віртуальної [1, 5-6] інфраструктурі сучасних інформаційних систем. Стисло розглянемо основні властивості зазначених різновидів *FW* [3-4, 7-18].

1. Proxy FW.

Проксі (*proxy*) фаєрвол вважається найбезпечнішим і найнадійнішим типом фаєрволів, який аналізує повідомлення на прикладному рівні, намагаючись захистити ресурси мережі. *FW* цього типу обмежують кількість програм, які здійснюють мережеву активність, що сприяє підвищенню безпеки, однак процес фільтрації потенційно може вплинути на швидкість і в деякій мірі на функціональність інформаційних систем, що захищаються.

2. UTM FW.

UTM фаєрволи є реалізацією комплексного рішення, котре поєднує функції антивірусного ПЗ і розширеної фільтрації контенту, що, вкупі, забезпечує протидію несанкціонованому витоку даних (тобто, функції *DLP*). В даному разі важливим є те, що з’являється можливість заощадити на витратах і технічному обслуговуванні такої системи, оскільки в цьому разі потрібно подбати лише про єдине рішення для управління загрозами.

3. Stateful Inspection FW.

Фаєрволи з перевіркою стану - це варіант захисту, який контролює стан активних мережевих з'єднань і одночасно аналізує вхідний трафік на предмет потенційних ризиків та загроз. Брандмауери із перевіркою стану функціонують на 3-му та 4-му рівнях моделі *OSI*, «переглядаючи» вміст пакетів даних і порівнюючи його з пакетами даних, які вже успішно пройшли через процес аналізу й фільтрації.

4. Next-generation FW.

Фаєрволи цього типу створені шляхом об'єднання функцій традиційних брандмауерів з різними мережевими засобами безпеки, перш за все системи запобігання вторгненням (*IPS*) та глибинний інспектування пакетів (*DPI*). *NGFW*, порівняно з іншими типами файрволів, зазвичай, використовують більш ретельний механізм перевірок, оцінюючи вміст пакетів і збігаючи їх сигнатури з відомими шкідливими зразками (*в т.ч. зловмисного ПЗ*). *FW* покоління *Next-generation* надають адміністраторам безпеки, кращу обізнаність і контроль над використовуваним ПЗ, а також більш глибші можливості, щодо спостереження поточної мережевої активності в тому числі за рахунок широкого залучення можливостей штучного інтелекту і машинного навчання (*AI/LM*) [19].

5. Threat-focused NG FW.

Ці фаєрволи є специфічною категорією *NGFWs*, які мають своїм основним завданням,

протидію впливу зловмисного ПЗ, атак на прикладному рівні та таргетованих атак. Крім того до сфери впливу *Threat-focused FW*, слід віднести протидію всім видам загроз, в тому числі й раніше невідомим.

6. Virtual FW.

Віртуальний або «хмарний» фаєрвол, це тип міжмережевого екрану, що призначений лише для сценаріїв, де розгортання апаратних брандмауерів є складним чи навіть неможливим завданням. Наприклад, у публічних/приватних хмарних середовищах чи *SDN* (*програмно-визначених мережах*). Також, вони можуть бути впроваджені, як віртуалізовані демони [1] для *NGFW* релізів.

Результати узагальнення основної функціональності, що притаманна для різних типів міжмережевих екранів, представлені в Табл.1 [4]. Вочевидь, така компіляція даних має вербальний характер, так як можливості різних релізів одного і того ж *FW*, можуть помітно відрізнятися між собою, в залежності від ступеню поточної актуалізації відомих загроз (*тобто, врахування їх механізмів й принципів дії*) [2, 15] та специфіки роботи ІКС (*топології, інтерфейсів, ступеню критичності основних процесів, швидкодії і типу використовуваних каналів передачі даних та ін.*) [4 ,7-8, 19]. Проте, вона надає кумульоване уявлення про загальний розподіл функціональних можливостей для основних різновидів засобів міжмережевого екранування та висвітлює найбільш показові відмінності у їх властивостях та комплектації відповідних продуктів.

Підсумовуючи відомості табл.1, можна зробити висновок, що засоби міжмережевого екранування потрібно обирати виключно під конкретну задачу та властивості базової ІКС. В рамках такого цілепокладання слід враховувати, що *UTM* рішення наближаються за своїми можливостями до *NGFW*, а для створення надійного безпекового базису для критично важливих ІКС, безумовно потрібно звернути увагу на технології *NGFW* та *Threat-focused NGFW*, які продовжують стрімко еволюціонувати.

Вочевидь, що торкаючись проблематики розвитку технологій і засобів міжмережевого екранування, ми неодмінно торкаємося питань, що стосуються функціональних особливостей *FW*, котрі мають різну цільову аудиторію, а саме: − рішення для корпоративного сегменту (*бізнес-клас*); − для приватних користувачів (*споживчій клас*).

Зрозуміло, що брандмауер споживчого класу реалізує більш простий користувальницький інтерфейс та має декілька звужений набір можливостей й налаштувань, що зумовлено необхідністю захисту лише декількох користувачів та/чи пристроїв з відносно простою топологією локальної мережі, якщо така взагалі є (*безвідносно інтерфейсів утворення цих мереж*). Інакше кажучи, для фаєрволів споживчого класу безумовними пріоритетами є їх швидкість і зручність використання (*Usability*), особливо з огляду на некомпетентність більшості кінцевих користувачів з питань забезпечення ІБ.

В загальному випадку *FW споживчого класу* призначені для «простої» домашньої мережі, з набагато меншим обсягом циркулюючих даних і меншим різновидом мережевих взаємодій та використовуваних протоколів. Ці фаєрволи в своєї переважній більшості є втіленням «реактивної» концепції мережевого захисту, що декілька знижує їх потенціал проти раніш невідомих загроз безпеки [2, 5, 15, 19]. Внаслідок власної функціональної обмеженості засоби міжмережевого екранування споживчого класу не можуть забезпечити виконання вимог з безпеки галузевих стандартів, що декларуються для відповідних бізнес-рішень (*наприклад, вимог стосовно обробки персоніфікованих даних*). При цьому, для фаєрволів бізнес-класу безумовними пріоритетами є безпека, котра включає в себе, в т.ч. таки можливості й якості, як віддалений доступ і масштабованість.

Таблиця 1 - Узагальнення функціоналу для основних типів фаєрволів
Table 1 - Generalization of functionality for the main types of firewalls

Підтримувані функції	**Тип реалізації FW**					
	Proxy FW	***UTM***	***Stateful Inspection FW***	***NGFW***	***Threat-focused NGFW***	***Virtual FW***
Антивірус та антиспам	-	+	-	+	+	+
Безпека *E-mail*	-	+	-	+	+	+
Управління додатками	-	+	-	+	+	+
Звітність	+	+	+	+	+	+
Управління репутацією та ідентифікацією	-	+	-	+	+	+
Рівень в моделі OSI	*7*	*7*	*3-4*	*2-7*	*2-7*	*3-4*
Управління пропускною здатністю	-	+	-	+	+	+
Фільтрація контенту *(web- сторінок)*	-	+	-	+	+	+
Фільтрація трафіку *(порти, IP/MAC - адреси, протоколи)*	+	+	+	+	+	+
DLP *(захист від витоку даних)*	-	+	-	+	+	+
IDS *(система виявлення вторгнень)*	-	+	-	+	+	+
IPS *(система запобігання вторгненням)*	-	+	-	+	+	+
NAT *(Network Address Translation)*	-	+	+	+	+	+
VPN *(Virtual Private Network)*	-	+	-	+	+	+

До основних складових захисту корпоративних FW слід віднести наступні [3-4, 7-14]:

- парирування загроз, в т.ч. за окремими векторами атак (*тобто, за конкретними вразливостями та/чи додатками*);
- поглиблений контроль поточних процесів для визначеного переліку ПЗ і використовуваного мережевого устаткування;
- перевірка *SSL* (*Secure Sockets Layer*);
- використання можливостей *AI* та *ML* для покращення процесу фільтрації та детектування ознак аномальної мережевої активності [1, 6, 19];
- аналіз з використанням репутаційних механізмів (*хмарних сервісів*);
- фільтрація трафіку на основі критеріїв геолокації і часових ознак;
- інтеграція з активним каталогом;
- фільтрація вмісту пакетів;
- антивірусні і антишпигунські функції;

- віддалена консоль безпеки/адміністрування;

- віртуальна кластеризація виконуваних модулів та динамічне балансування пропускної здатності (*тобто продуктивності фільтрації*);

- управління «активною» конфігурацією (*технологія програмних блейдів*).

Як слід з наведеного переліку складових, фаєрволи *бізнес-сегменту* розроблені з урахуванням набагато більш складних та декларованих умов їх подальшого застосування. Тому цілком зрозуміло, що корпоративні *FW* нового покоління в свої функціональній парадигмі орієнтуються на умови активного та адаптивного захисту критично важливих даних і мережевого устаткування від широкого спектру нових загроз [2, 15]. Для цього вони розробляються з набагато більш досконалим та різноманітним набором інструментів і функцій, зумовлених широким спектром нішевих інтересів, особливо в галузі високих технологій. До того ж фаєрволи *бізнес-класу* часто постачаються з постійною підтримкою, оновленнями та управлінням з боку фахівців розробника [4].

Таким чином, можна стверджувати, що фаєрволи споживчого класу, в своєї переважній більшості, надають їх користувачам базові функції (*такі як контроль портів/протоколів, захист від несанкціонованого доступу тощо*), а рішення бізнес-класу пропонують «агресивно» проактивний набір функцій, наприклад: - виявляють і блокують складні атаки та спроби ведення мережевої розвідки, забезпечують детальний контроль пакетів (*DPI*), підтримують функції *DLP* та *IPS* тощо. Крім того для рішень корпоративного сегменту більш виражена можливість оперативних оновлень діючих алгоритмів та процедурних блоків для ефективної протидії до змінних (поліморфних) загроз. На їх фоні фаєрволи споживчого рівня в більшої мірі сфокусовані на питаннях простоти використання, швидкодії та низької вартості підписки, а склад їх функціоналу орієнтований на забезпечення безпеки невеликих мереж з низьким рівнем ризиків. При цьому для складних, інтегрованих ІКС, котрі вимагають високого рівня їх безпеки, продуктивності і гнучкості (масштабованості), брандмауери бізнес-рівня покоління *Next-generation* та *Threat-focused NGFW*, безумовно є кращим вибором. В разі необхідності захисту і моніторингу програмно емульованих [1, 19] середовищ з глибоким рівнем вкладеності й кластеризації відповідних віртуалізованих платформ, слід звернути увагу на рішення рівня *Virtual FW*.

Специфіка реалізацій фаєрволів для мобільних платформ.

Мобільний фаєрвол - це програмний засіб безпеки, що протидіє мережевим загрозам, які притаманні для мобільних пристроїв, працюючи згідно принципів «традиційних» *FW*, однак на відміну від них, розроблений спеціально для умов користування саме мобільних гаджетів. До таких умов насамперед слід віднести наступні:

- високий ступінь мобільності гаджетів по відношенню до комунікаційних шлюзів верхнього рівня ієрархії (*провайдерів послуг*);

- обмеженість бортових обчислювальних можливостей (*перш за все, процесор та оперативна пам'ять*);

- обмеженість ємності *вбудованого* джерела електроживлення;

- необхідність адміністрування та врахування поточних параметрів енергоспоживання для «активних» додатків та резидентних процесів;

- переривчастість сеансів зв’язку;

- обмеженість і нестабільність у часі доступної смуги пропускання та необхідність буферизації даних для «вирівнювання» трафіку;

- яскраво виражена асиметричність трафіку (вгору/вниз) для більшості програмних додатків;

- можливість існування великої кількості проміжних репітерів та зростання пінг-таймінгу;

- необхідність підтримки режиму реального часу, для певного числа додатків (наприклад, *Skype*);

- можливість роботи пристрою, як вхідний proxy або шлюз для сукупності інших пристроїв (в т.ч. обслуговування скаттернет (*Scatternet*));

- висока динамічність мережевого оточення в рамках підтримуваної *Scatternet* (*в т.ч. в межах реалізації технології Інтернету речей - IoT*);

- робота через велику кількість випадкових точок доступу (шлюзів) з неконтрольованою на них ПІБ та фільтрації трафіку;

- різноманіття стандартів передачі та порядку їх використання в рамках однієї сесії (*в тому числі супутниковий канал зв’язку*) та ін..

В загальному випадку мобільний фаєрвол контролює обмін даними та забезпечує безпеку підключень до мережі (*в т.ч. через віртуальні приватні мережі - VPN*), цілеспрямовано оптимізуючі параметри використання наявних ресурсів і циркуляцію трафіку за всіма інтерфейсами (протоколами) взаємодії. Адміністрування ресурсами гаджету підтримується, завдяки здатності контролювати використання апаратних ресурсів з боку додатків, що здійснюють мережеву взаємодію та параметри обробки самих даних (*наприклад, за рахунок зміни бітрейту мультимедійного контенту для відповідного мережевого застосунку, в залежності від параметрів наявної смуги частот (вільного канального ресурсу)*). Таким чином мобільні фаєрволи повинні всіляко сприяти підтримці ресурсного консенсусу та усувати можливий диспаритет в продуктивності мобільних застосунків. В межах виконання зазначених вище завдань, мобільний *FW* на базовому рівні захисту повинен запобігати мережеву взаємодію з небажаними ресурсами, протидіяти підміні IP-адрес й спуфінгу, контролювати перелік додатків, які здійснюють мережеву взаємодію та не дозволяти пакетам даних, що містять шкідливе навантаження, перетинати умовний «периметр безпеки» гаджету. Це стає особливо актуально, як превентивний захід для забезпечення безпеки корпоративного трафіку даних в разі використання гаджетів, як віддаленого терміналу/консолі співробітників компанії (установи).

Таким чином, мобільні *FW* відрізняються від «традиційних» міжмережевих екранів завдяки всебічного врахування проблематики і умов функціонування мобільних пристроїв та мобільних мереж. Розробники відповідних рішень приділяють особливу увагу питанням економії заряду вбудованого акумулятора, використовуючи ресурсозберігаючі та обчислювально оптимізовані технології для мінімізації проявів й наслідків енергодефіциту при експлуатації мобільних платформ. При цьому, захист конфіденційності має першорядне значення, де основна увага приділяється контролю мережевої активності мобільних додатків для запобігання спробам несанкціонованого витоку даних і доступу до чутливої інформації чи окремим функціям самого гаджету. Усуваючи специфічні для мобільних гаджетів загрози безпеки, вони протидіють впливу шкідливого ПЗ та спробам реалізації найбільш поширених атак: – фішингу, спуфінгу (наприклад, *DNS Spoofing*) [15], таргетований спам та ін.

Адаптивність до мобільності сучасних мереж зв’язку, водночас, визначає специфічність загроз безпеки для мобільних пристроїв та зумовлює їх ключову особливість, яка базується на перманентній готовності до переходів (пере підключень) між різними мережами (*і цей процес значно складніший, ніж звичайний хендовер (handover) у рамках однієї мережі*). Зрозуміло, що зручні інтерфейси покращують *usability* мобільних *FW* завдяки сповіщенням у реальному часі та легким і «прозорим» налаштуванням. По суті, мобільні фаєрволи надають пріоритет безпеці, конфіденційності та забезпечення параметричного консенсусу для виконуваних процедур, з урахуванням унікальних характеристик і ризиків, що характерні для мобільних платформ. В рамках зазначеної цільової парадигми, мобільні *FW* в найбільшої мірі сфокусовані на виконанні наступних завдань [20-21]:

- Веб-фільтрація (*для блокування шкідливого вмісту*);
- Блокування небажаних web-джерел (*репутаційні хмарні сервіси, стоп-лист, батьківський контроль та ін.*);
- Впровадження політик/сценаріїв перегляду окремих *web*- сторінок;
- Контроль *Cookie* та скриптів;
- Усування передумов експлуатації вразливостей/експлойтів;
- VPN для безпечного підключення на «невідомих» точках доступу;
- Захист від відомих типів атак (*згідно відомих поведінкових сигнатур*);
- Блокування використання даних гаджету в межах задіяних обмежень;
- Підтримка правил контролю для різних типів трафіку і додатків;
- Управління напрямом трафіку додатків для блокування небажаних сеансів (*як функціонал DLP*);
- Блокування спаму і фішингових посилань в *E-mail* та *SMS* (*Smishing*);
- Нормування трафіку за типами інтерфейсів та звітування (*інколи логування*) про стан комунікаційної активності платформи.

3. Висновки

1. Міжмережеві екрани є вкрай важливою складовою в загальній системі безпеки інформаційних ресурсів сучасних ІКС та забезпечують адаптивний захист мережевого обладнання від деструктивного впливу широкого спектру загроз, як зовнішнього, так і внутрішнього походження.

2. Різні типи фаєрволів знаходять своє застосування в залежності від умов функціонування і призначення базової ІКС, а також від місця їх інтеграції в мережеву та/чи віртуальну інфраструктуру сучасних інформаційних систем.

3. Засоби міжмережевого екранування слід обирати виключно під конкретну задачу та властивості базової ІКС. В рамках такого цілепокладання слід враховувати, що *UTM* рішення наближаються за своїми можливостями до *NGFW*, а для створення надійного безпекового базису критично важливих ІКС, потрібно звернути увагу на технології *NGFW* та *Threat-focused NGFW*, які продовжують еволюціонувати.

4. Для інтегрованих ІКС, що вимагають високого рівня їх безпеки, продуктивності і гнучкості, брандмауери бізнес-сегменту покоління *Next-generation* та *Threat-focused NGFW*, безумовно є кращим вибором. При необхідності захисту та моніторингу програмно емульованих мережевих середовищ з глибоким рівнем кластеризації і відтворення відповідних віртуалізованих співтовариств, слід звернути увагу на рішення рівня *Virtual FW*.

5. Мобільні фаєрволи повинні всіляко сприяти підтримці ресурсного консенсусу та усувати можливий диспаритет в продуктивності мережевих мобільних застосунків. В рамках виконання цих завдань, мобільні *FW* повинні мати дружній *Usability*, запобігати мережеву взаємодію з небажаними ресурсами, протидіяти спуфінгу у всіх його проявах, контролювати перелік додатків, які здійснюють мережеву взаємодію та не дозволяти пакетам даних, що містять шкідливе навантаження, перетинати умовний «периметр безпеки» гаджету.

6. Адаптивність до мобільності сучасних мереж зв’язку (*Wi-Fi, GSM, CDMA, Bluetooth, супутникові канали зв’язку та ін.*), водночас, визначає й специфічність загроз безпеки для мобільних пристроїв та зумовлює їх ключову особливість, котра базується на перманентній готовності до «безшовних» переходів/перепідключень між різними мережами в умовах постійного енергодефіциту та обмеженості наявних обчислювальних ресурсів.

Конфлікт інтересів

Автори повідомляють про відсутність конфлікту інтересів.

Список літератури:

1. Азаров, С., Нємцев, М., & Малахов, С. Огляд аналогій та обґрунтування принципів створення демон юнітів відстеження мережевої активності користувачів. *Proceedings of the XX International Scientific and Practical Conference.* Graz, Austria. 2023. Pp. 447-453. https://doi.org/10.46299/ISG.2023.1.20
2. Богданова, Є., Чорна, Т., & Малахов, С. (2022). Огляд поточного стану загроз, що обумовлені впливом експлойтів. *Комп'ютерні науки та кібербезпека*, (2), 35-40. https://periodicals.karazin.ua/cscs/article/view/21039/19745
3. Sichkar, M., & Pavlova, L. (2024). A short survey of the capabilities of Next Generation firewalls. *Computer Science and Cybersecurity*, (1), 28-33. https://periodicals.karazin.ua/cscs/article/view/23090
4. Січкар, М., & Малахов, С. Узагальнення особливостей відомих засобів міжмережевого екранування. *Proceedings of the XXI International Scientific and Practical Conference.* Sofia, Bulgaria. 2024. URL: https://isg-konf.com/category/archiv-conference-rub/
5. Кохановська, Т., Нарежний, О., & Дьяченко, О. (2020). Дослідження можливостей технології Honeypot. *Комп'ютерні науки та кібербезпека*, 1(1), 33-42. https://doi.org/10.26565/2519-2310-2020-1-03
6. Михайленко Д., Нємцев М. Особливості технології мережевих пасток як інструменту активного захисту та аналізу дій атакуючої сторони. *Proceedings of the XXI International Scientific and Practical Conference.* Melbourne, Australia. 2023. Pp. 483-487. https://doi.org/10.46299/ISG.2023.1.21
7. Джон Маллери, & Джейсон Занн (2007). Безопасная сеть вашей компании. (Е. Линдеманн, пер. с англ.). – М.: НТ Пресс
8. Рондалєв, Д., Мелкозьорова, О., & Нарєжній, О. (2019). Особливості функціонування корпоративного міжмережевого екрану та питання взаємодії з системою IDS. *Комп'ютерні науки та кібербезпека*, (3), 11-21. URL: https://periodicals.karazin.ua/cscs/article/view/15614/14707
9. Брандмауер (FireWall). URL: http://surl.li/tuggo
10. Who Invented the Firewall? History, Types and Generations of Firewall. 28th September 2023 by Manish Sahay https://www.thepcinsider.com/who-invented-firewall-history-evolution-types-generations/
11. What Is a Firewall? URL: http://surl.li/fdtbp
12. Next-Generation Firewalls. URL: https://www.paloaltonetworks.com/network-security/next-generation-firewall
13. 8 Types of Firewalls: Know Which One Is Best for Your Network By John Villanueva / May 19, 2022. URL: https://techgenix.com/types-of-firewalls
14. Information Technology Gartner Glossary. URL: https://www.gartner.com/en/information-technology/glossary/next-generation-firewalls-ngfws
15. Яремчук, К., Воскобойников, Д., & Мелкозьорова, О. (2022). Сучасні загрози та способи забезпечення безпеки веб-застосунків. *Комп'ютерні науки та кібербезпека*, (2), 28-34. https://periodicals.karazin.ua/cscs/article/view/21038/19744
16. What is a Next-Generation Firewall (NGFW)? URL: https://www.zenarmor.com/docs/network-security-tutorials/next-generation-firewall
17. Top Next-Generation Firewall (NGFW) Software By Jenna Phipps July 19, 2022. URL: https://www.cioinsight.com/security/ngfw-software/#What-is-a-next-generation-firewall
18. What are the Types of Firewalls? URL: https://www.zenarmor.com/docs/network-security-tutorials/what-is-firewall
19. Михайленко, Д., Чорна, Т. & Малахов, С. Використання можливостей AI при реалізації Static та Dynamic Honeypot для покращення параметрів захисту інформаційних ресурсів. *Технології, інструменти та стратегії реалізації наукових досліджень: матеріали IV Міжнародної наукової конференції*, (с. 54-57). 7.10.2022 р. Суми, Україна: МЦНД. https://doi.org/10.36074/mcnd-07.10.2022

20. How do firewalls for mobile devices differ from traditional firewalls? URL: https://www.xcel.com/how-do-firewalls-for-mobile-devices-differ-from-traditional-firewalls/
21. How mobile firewalls protect against unique threat vectors. URL: http://surl.li/tugix

FUNCTIONAL FEATURES OF WELL-KNOWN MEANS OF NETWORK SHIELDING

Mykhailo Sichkar[1], CSD Student (bachelor), Department of Security of Information Systems and Technologies; e-mail: sichkar2020kb13@student.karazin.ua;
Mikolaj Karpinski[2], Prof., DSc, Professor (Full); e-mail: mikolaj.karpinski@up.krakow.pl;
ORCID: https://orcid.org/0000-0002-9790-7260
Serhii Malakhov[1], Ph.D., Senior Researcher, Computer Science Department;
e-mail: malakhov@karazin.ua; ORCID: https://orcid.org/0000-0001-8826-1616
[1] *V. N. Karazin Kharkiv National University, Ukraine*
[2] *VInstitute of Security and Computer Science, University of the National Education Commission, Krakow, Poland*
Manuscript was received April 2, 2024; Received after review May 2, 2024; Accepted June 3, 2024

Abstract. The work briefly reviews the history, types, and capabilities of the main types of firewalls (*FW*). Firewalls are an important tool for protecting network resources from various information security threats. With the development of technology and the changing nature of attacks, especially those involving artificial intelligence (*IoT*), firewalls have also evolved, acquiring new functions and capabilities. This work provides a short survey of the main types, and capabilities of firewall technology, providing solutions to issues of comprehensive protection of network equipment and information resources from modern security threats. Different types of firewalls are used depending on the conditions of operation and purpose of the basic information and communication system (*ICS*), as well as on the place of their (*FW*) integration into the network or virtual infrastructure of modern information systems. For integrated networks that require a high level of their security, productivity and flexibility, firewalls of the business segment of generation *«Next-generation»* and *«Threat-focused NGFW»* are definitely the best choice. Attention was drawn to the fact that mobile firewalls should in every way contribute to the maintenance of resource consensus and eliminate a possible disparity in the performance of networked mobile applications. Adaptability to mobility of current communication systems (*Wi-Fi, GSM and others*) determines the specificity of security threats for mobile devices and It defines their key feature. This feature is based on permanent readiness for seamless transitions (reconnections) between different networks in conditions of constant energy shortage and limited available computing resources (meaning gadgets). Highlights the main trends, prospects for the development and implementation of different types of firewalls, including the impact of artificial intelligence, machine learning, cloud technologies and the Internet of Things as well as important aspects of their (*FW*) scope. It is emphasized that the introduction of *FW* does not replace other security technologies and tools, but effectively expands the existing arsenal of countering new security threats (primarily as an instrument of proactive countermeasures and rapid response to complex network incidents). The article may be useful for students, researchers, and information security professionals who seek to expand their competencies related to the development and operation of modern means of network protection.

Keywords: *FW, Firewall, Information Security, Security Threats, Malicious Software*

Conflicts of Interest: the authors declare no conflict of interest.

References

1. Azarov, S., Nemtsev, M., & Malakhov, S. Review of analogies and justification of the principles of creation of daemon units for tracking users' network activity. *Proceedings of the XX International Scientific and Practical Conference*. Graz, Austria. 2023. Pp. 447-453. Available at: https://doi.org/10.46299/ISG.2023.1.20 [In Ukrainian]
2. Bohdanova, E., Chorna, T., & Malakhov, S. (2022). Overview of the current state of threats caused by the influence of exploits. *Computer Science and Cybersecurity*, (2), 35-40. URL: https://periodicals.karazin.ua/cscs/article/view/21039/19745 [In Ukrainian]
3. Sichkar, M., & Pavlova, L. (2024). A short survey of the capabilities of Next Generation firewalls. *Computer Science and Cybersecurity*, (1), 28-33. Retrieved from https://periodicals.karazin.ua/cscs/article/view/23090
4. Sichkar, M., & Malakhov, S. Generalization of features of known means of network shielding. *Proceedings of the XXI International Scientific and Practical Conference*. Sofia, Bulgaria. 2024. https://isg-konf.com/category/archiv-conference-rub/ [In Ukrainian]
5. Kokhanovska, T., Narezhny, O., & Dyachenko, O. (2020). Exploring the capabilities of Honeypot technology. *Computer Science and Cybersecurity*, 1(1), 33-42 https://doi.org/10.26565/2519-2310-2020-1-03 [In Ukrainian]
6. Mykhaylenko D., Nemtsev M. Peculiarities of the technology of network traps as a tool of active protection and analysis of the actions of the attacking party. *Proceedings of the XXI International Scientific and Practical Conference*. Melbourne, Australia. 2023. Pp. 483-487. Available at: https://doi.org/10.46299/ISG.2023.1.21 [In Ukrainian]
7. John Mallery, & Jason Zann (2007). Your company's secure network. (E. Lindemann, translated from English). - M.: NT Press [In Ukrainian]
8. Rondalev, D., Melkozyorova, O., & Narezhniy, O. (2019). Peculiarities of the operation of the corporate inter-network screen and the issue of interaction with the IDS system. *Computer Science and Cyber Security*, (3), 11-21. https://periodicals.karazin.ua/cscs/article/view/15614/14707 [In Ukrainian]
9. FireWall. URL: http://surl.li/tuggo
10. Who Invented the Firewall? History, Types and Generations of Firewall. 28th September 2023 by Manish Sahay URL: https://www.thepcinsider.com/who-invented-firewall-history-evolution-types-generations/
11. What Is a Firewall? URL: http://surl.li/fdtbp
12. Next-Generation Firewalls. URL: https://www.paloaltonetworks.com/network-security/next-generation-firewall
13. Types of Firewalls: Know Which One Is Best for Your Network By John Villanueva / May 19, 2022. https://techgenix.com/types-of-firewalls
14. Information Technology Gartner Glossary. https://www.gartner.com/en/information-technology/glossary/next-generation-firewalls-ngfws
15. Yaremchuk, K., Voskoboynikov, D., & Melkozyorova, O. (2022). Modern threats and ways to secure web applications. *Computer Science and Cybersecurity*, (2), 28-34. https://periodicals.karazin.ua/cscs/article/view/21038/19744 [In Ukrainian]
16. What is a Next-Generation Firewall (NGFW)? Вилучено з URL: https://www.zenarmor.com/docs/network-security-tutorials/next-generation-firewall
17. Top Next-Generation Firewall (NGFW) Software By Jenna Phipps July 19, 2022. URL: https://www.cioinsight.com/security/ngfw-software/#What-is-a-next-generation-firewall
18. What are the Types of Firewalls? URL: https://www.zenarmor.com/docs/network-security-tutorials/what-is-firewall
19. Mykhaylenko, D., Chorna, T. & Malakhov, S. The use of AI capabilities in the implementation of Static and Dynamic Honeypot to improve the parameters of protection of information resources. *Technologies, tools and strategies for the implementation of scientific research: materials of the IV International Scientific Conference*, (p. 54-57). October 7, 2022. Sumy, Ukraine: MCND. https://doi.org/10.36074/mcnd-07.10.2022 [In Ukrainian]

20. How do firewalls for mobile devices differ from traditional firewalls? https://www.xcel.com/how-do-firewalls-for-mobile-devices-differ-from-traditional-firewalls/
21. How mobile firewalls protect against unique threat vectors. URL: http://surl.li/tugix

ISSN 2519-2310 (Online)

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА, 2024, № 1 (25)

COMPUTER SCIENCE AND CYBERSECURITY, 2024, Iss. 1 (25)

DOI: https://doi.org/10.26565/2519-2310-2024-1-06
УДК 004.056.5

ВІДНОВЛЕННЯ ТРИВИМІРНИХ СЦЕН НА ОСНОВІ ДАНИХ ВІДЕО ПОТОКІВ

Денис Грульов[1], магістрант, e-mail: xa11800855@student.karazin.ua,
ORCID: https://orcid.org/0009-0005-8506-770X
Анастасія Морозова[1], доцент, доктор філософії, e-mail: a.morozova@karazin.ua,
ORCID: https://orcid.org/0000-0003-2143-7992
Петро Доля[1], доцент, доктор філософії, e-mail: pdolya@karazin.ua,
ORCID: https://orcid.org/0009-0002-4062-4443
Лілія Бєлова[1], старший викладач, e-mail: l.belova@karazin.ua,
ORCID: https://orcid.org/0009-0007-0805-4547
[1]*Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, 61022, Україна*

Рукопис надійшов 23 березня 2024 р. Отримано після рецензування 29 квітня 2024 р.
Прийнято 30 травня 2024 р.

Анотація: Дана робота присвячена застосуванню сучасних алгоритмів відновлення тривимірних сцен з зображень для отримання просторової інформації із відео. У роботі розглядається розмаїття сучасних методів, підходів та алгоритмів в області аналізу відео потоку. Приділено увагу послідовності розвитку підходів до вирішення задачі. У процесі дослідження області та результатів, пов'язаних з тривимірною реконструкцією на основі зображень та відео потоків, був винайдений алгоритм, що дозволяє будувати щільні мапи глибини, використовуючи інформацію з усіх кадрів відео. Ідея полягає у тому, щоб використовувати готові, загальноприйняті та перевірені рішення для вирішення двох задач: COLMAP - для візуальної одометрії, та RAFT - для обчислення оптичного потоку. Запропонований алгоритм показує досить точні результати, та відновлює мапу глибини в деталях на довільних статичних сценах.

Ключові слова: *відео потік, 3D-реконструкція, машинного навчання, одометрія, нейронна мережа, комп'ютерний зір, мапа глибини, оптичний потік*

Як цитувати: Грульов Д., Морозова А., Доля П., Бєлова Л.. Відновлення тривимірних сцен на основі даних відео потоків. *Комп'ютерні науки та кібербезпека.* 2024; № 1(25): С. 66–75. https://doi.org/ 10.26565/2519-2310-2024-1-06

In cites: Hrulov D., Morozova A., Dolia P., Bielova L. (2024). Reconstruction of three-dimensional scenes based on video flow data. *Computer Science and Cybersecurity.* 1(25): 66–75. https://doi.org/10.26565/2519-2310-2024-1-06 (in Ukrainian)

1. Вступ

Задача відновлення просторових даних з відео відноситься до загальної задачі відновлення структури із руху (Structure-from-Motion, SfM) [1]. На даний момент існують

алгоритми, що якісно відновлюють просторову інформацію з відео потоку. Відео потік заздалегідь містить більше інформації, ніж окремі фото, зокрема послідовність, в яких ці фото розташовані, і самі послідовні фото є зображеннями, що різняться дуже мало, що дозволяє, наприклад, точно знаходити щільні відповідності між точками та відстежувати траєкторії об'єктів. Проте, більшість алгоритмів, що аналізують просторову структуру на основі відео, спираються на попарний аналіз зображень. За допомогою попарної обробки кадрів, алгоритми можуть визначати відповідності між об'єктами на кадрах. Ці параметри можуть бути використані для подальшої реконструкції тривимірної моделі сцени. Недоліком цього підходу є те, що об'єм обчислень може бути неприйнятно великим для ряду застосувань.

Методи тривимірної реконструкції загалом можна розділити на класичні (традиційні) та підходи машинного навчання (ML). Класичні методи 3D-реконструкції зазвичай спираються на геометричні принципи, такі як тріангуляція, епіполярна геометрія та калібрування камери. Ці методи часто передбачають явне моделювання геометрії та фізики процесу створення зображень і можуть вимагати ручних або напівручних кроків для виділення ознак, зіставлення та реконструкції. З іншого боку, підходи машинного навчання (ML) для 3D-реконструкції використовують алгоритми, навчені на великих наборах даних, щоб вивчати шаблони та зв'язки між даними.

У багатьох випадках, в останніх дослідженнях класичні та засновані на машинному навчанні методи тривимірної реконструкції використовуються разом, щоб доповнити сильні та слабкі сторони першого та другого підходів. Поєднання класичного підходу та підходу на основі машинного навчання може використовувати переваги обох підходів, що призводить до підвищення точності та надійності завдань 3D-реконструкції.

Задача відновлення просторової інформації з відео потоку є актуальною задачею комп'ютерного зору, а саме його підрозділу – 3D реконструкції. Вилучення просторової інформації з відео потоку є дуже важливим для таких застосувань, як робототехніка, самокерування автомобілів, доповнена та віртуальна реальність, тощо.

Мета цієї статті є дослідити можливості застосування сучасних алгоритмів відновлення тривимірних сцен з двох зображень (Two-View Structure from Motion) для відновлення просторової інформації із відео потоку.

У статті розглядаються алгоритми та підходи, що допомагають відновити просторову інформацію з кадрів відео для відновлення тривимірних сцен лише із двох зображень. А також розробка алгоритму відновлення тривимірної інформації про сцену з усіх кадрів відео.

2. Огляд існуючих рішень

За останні два десятиліття сфера 3D-реконструкції з відеокадрів значно розвинулась завдяки прогресу як у традиційних методах вилучення ознак, так і в сучасних методах глибокого навчання.

Останні розробки демонструють потенціал глибокого навчання для вирішення складних сценаріїв реконструкції та продовжують розширювати межі можливого в цій галузі. У 2020 році розроблений алгоритм, що реконструює мапу глибини для кожного кадру відео [2]. Алгоритм показує якісні та стабільні результати застосовано до відео, що знімають довільні сцени, проте, у сенсі реконструювання динамічних сцен, спеціалізований до оцінки руху людей. Не зважаючи на те, що перевагою алгоритму висувається виключна ступінь узгодженості реконструкції між кадрами, алгоритм використовує попарну обробку кадрів, що вибираються за описаним принципом, як базовий етап навчання моделі.

В останні роки розробка застосувань візуальної одометрії на основі засобів, таких як ORB-SLAM [3] і DSO (пряма розріджена одометрія), продемонструвала здатність забезпечити оцінку руху камери в реальному часі з високою точністю. Ці методи використовують ефективні

методи вилучення функцій, зіставлення та оптимізації для досягнення продуктивності в реальному часі на сучасному обладнанні. ORB-SLAM – це алгоритм та програмне забезпечення, що активно доповнюється, розробляється та покращується. На даний момент розроблена вже третя 15 версія цього – алгоритму ORB-SLAM3 [4].

Сучасні методи глибокого навчання також дозволили побудувати алгоритми візуальної одометрії, що дають точні та надійні результати та на даний момент наближаються до готовності використання у реальних умовах. Одним із новітніх алгоритмів, що заслуговує уваги, є Deep Patch Odometry [5]. В основі даного алгоритму лежить розбиття кадрів відео на клаптики (patches) за допомогою однієї нейронної мережі та відстежування їх руху за допомогою іншої, рекурентної нейронної мережі.

Повертаючись до задачі відновлення тривимірних сцен з двох зображень, в 2021 була оприлюднена нейромережева модель RAFT-Stereo, що вирішує задачу, та на момент 2022 року посідає друге місце у рейтингу RVC Stereo [6]. Алгоритм є модифікацією нейромережі для знаходження оптичного потоку RAFT (Recurrent All-Pairs Field Transforms). У 2022 році був також винайдений алгоритм CREStereo, що показує ще кращий результат та може давати ще точніші реконструкції [7].

Проте, треба зазначити, що алгоритми, які базуються на машинному навчанні, сильно залежать від вибірки, на якій були навчені. Більш того, такі підходи часто мають погану узагальнювальну здібність між вибірками, тому на практиці потребують навчання під конкретні умови, для повного охоплення яких, потрібні надто великі вибірки. Тому адаптація алгоритму під конкретні умови, буде потребувати дуже серйозного об’єму обчислень без жодної гарантії на прийнятний результат в умовах, які можуть лише незначно відрізнятися від тих, до яких він був адаптований.

Хоча монокулярна реконструкція глибини відео досягла значного прогресу та застосовувалася до різних програм, таких як робототехніка, доповнена реальність та автономне водіння, вона залишається активною областю досліджень із постійними викликами та досягненнями. Тому задача відновлення мап глибини з двох зображень є актуальною задачею.

3. Класичний підхід до задачі відновлення просторової інформації з 2-вимірних зображень

У даній роботі розглядається задача аналізу просторової інформації у контексті знаходження мап глибини зображень. Мапа глибини – це зображення, в якому кожному пікселю відповідає значення, що відображає його відстань від камери або іншого датчика.

Структура з руху (Structure from Motion, SfM) — це техніка, яка використовується в комп’ютерному зорі для реконструкції 3Dструктури сцени або об’єкта з набору 2Dзображень або відеокадрів, знятих з різних точок зору. Основна ідея SfM полягає у використанні візуальної відповідності між об’єктами на двох зображеннях для оцінки 3D-положень цих об’єктів, а також позиції камери.

Рис. 3.1. Мапи глибини
Fig. 3.1. Depth maps

Існує декілька підходів до реалізації алгоритму Structure-from-Motion.

Класичний підхід до SfM передбачає ітеративне поєднання результати обробки двох кадрів. Для уточнення відновлених поз камер та розрідженої хмари точок використовується налаштування пучка (bundle adjustment), що мінімізує загальну помилку повторної проекції поміж усіма ракурсами (Рис. 3.2.)

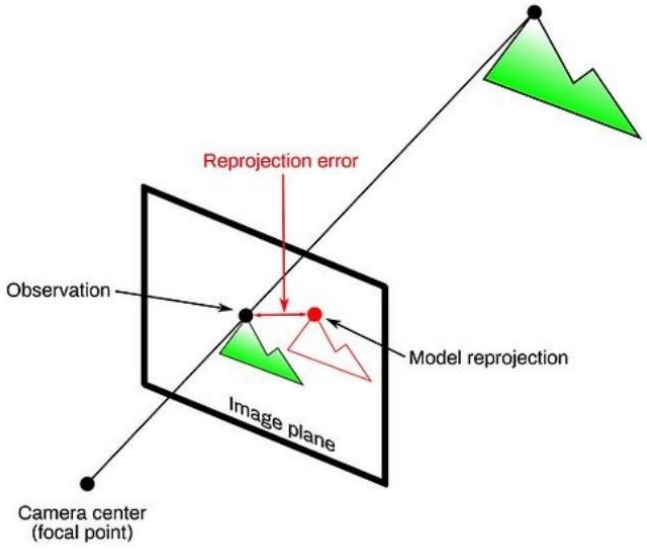


Рис. 3.2. Класичний підхід до SfM
Fig. 3.2. A classical approach to SfM

Розповсюдженою технікою аналізу відео потоку, є оптичний потік, що в тому числі дозволяє слідкувати за рухом об’єктів. Це поле векторів руху, яке описує, як об’єкти на зображенні рухаються відносно спостерігача.

Також, оптичний потік також можна використовувати безпосередньо для встановлення візуальних відповідностей між точками на двох зображеннях.

4. Комбінація сучасних засобів комп’ютерного зору для відновлення мап глибини з кадрів відео

Не дивлячись на те, що задача відновлення мап глибини з двох зображень, є складною, та дослідження якої ведеться багатьма науковцями у даний момент, реконструкція мап глибини з відео, як уже зазначалося, має досить задовільні рішення, якщо не брати до уваги швидкість роботи алгоритму. У даній роботі пропонується алгоритм відновлення мап глибини, що використовує традиційні геометричні методи для відновлення орієнтації та позиції камери у просторі для кожного кадру, та обчислення оптичного потоку за допомогою глибокого навчання.

Для реалізації класичного алгоритму Structure-from-Motion, існує достатньо готового програмного забезпечення. Алгоритм, що вирішує задачу Structure-From-Motion, за визначенням також вирішує задачу одометрії – задачі знаходження орієнтації та позиції камери. Вибір засобу для вирішення задачі одометрії не є принциповим, якщо швидкість виконання не є важливою характеристикою.

Основна проблема, що є у існуючих класичних засобів вирішення задачі Structure-from-Motion – знаходження щільних мап глибини. Мапи глибини, хоча і можуть бути отримані за допомогою таких програмних засобів як COLMAP, мають значні області невизначеності,

оскільки реконструкція базуються на знаходженні деякої кількості обраних пікселів на зображеннях та їх зіставленні як проекцій на площину зображення для знаходження відповідностей між ними. В результаті отримується хмара точок у просторі, кожна точка якої проектується на деяку кількість зображень. Таким чином, мапи глибини фактично будуються на базі проекцій хмари відтворених точок на різні кадри відео, що мають різні ракурси. Приклад мапи глибини, отриманої за допомогою програмних засобів як COLMAP, наведено на рис. 4.1.

Рис. 4.1. Мапа глибини за допомогою COLMAP
Fig. 4.1. Depth map using COLMAP

При застосуванні вищеописаного підходу, попіксельна мапа глибини не отримується. Більш щільні мапи глибини будуються вже на базі первинної реконструкції, проте, і ці мапи глибини на практиці мають області невизначеності. Для того, щоб відновити щільні мапи глибини, пропонується спочатку отримати позиції камер. Для того, щоб відновити мапу глибини для деякого кадру, пропонується використовувати інформацію про позиції камери інших кадрів, та поле оптичного потоку між кадром, для якого будується мапа глибини, та відповідними кадрами відео.

Першим етапом є відновлення положення камер. Для відновлення положення камер, обрано COLMAP, так як це добре перевірене ПЗ, що дуже тонко конфігурується та має зручний консольний інтерфейс та дозволяє зчитувати результати реконструкції, зокрема позиції камер для кожного кадру відео, та параметри калібровки камери.

Наступним етапом є переведення отриманих позицій камер у спільну систему координат. Дані про положення камери, згідно з традиційною схемою, отримуються у вигляді матриць повороту R та векторів переміщення t. Для того щоб перетворити вектор $(X, Y, Z)^T$ зі світової системи координат до локальної системи координат камери, треба застосувати вектор переміщення та матрицю повороту камери:

$$\begin{pmatrix} X' \\ Y' \\ Z' \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t \tag{1}$$

Відповідно, для того, щоб знайти глобальні координати вектора, треба виразити його з рівняння:

$$\begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{pmatrix} = \mathrm{R}^{-1} \left(\begin{pmatrix} \mathrm{X}' \\ \mathrm{Y}' \\ \mathrm{Z}' \end{pmatrix} - \mathrm{t} \right) \tag{2}$$

Враховуючи, що $R^{-1} = R^T$, бо матриця повороту завжди ортогональна, отримуємо зручний для обчислення перехід до єдиної системи координат.

Тепер, розглянемо вектор між фокусним центром камери та пікселем (u, v), що є точкою, що лежить на площині зображення. Z-координата будь якої точки площини зображення дорівнює фокальній відстані f, а центровані координати пікселя – $(u - c_x, v - c_y)$. Таким чином, координати точки на площині зображення у системі координат камери дорівнюють $(f, u - c_x, v - c_y)$. Оскільки оптичний центр камери знаходиться у центрі системи координат камери, то і шуканий вектор матиме вигляд

$$a = (f, u - c_x, v - c_y) \tag{3}$$

Тепер, якщо позиція камери відома, перетворенням вектору a до світової системи координат безпосередньо отримується напрям променя, на якому лежить точка, що відображається у піксель (u, v). Сам промінь отримується з обмеження, що у рівнянні

$$\begin{gathered} l = as + b \\ b = t, \text{при } s = 0 \end{gathered} \tag{4}$$

За допомогою оптичного потоку, для кожної точки першого зображення, можна знайти відповідну точку на іншому. Оптичний потік відображає відповідності краще всього, коли зображення, що порівнюються, відрізняються не сильно. Тому, для кожного кадру, є сенс підбирати деяку кількість найближчих кадрів. Для кожного кадру, що зіставляються, обчислимо оптичний потік, для кожного пікселя вихідного кадру (u, v), підберемо відповідний піксель (u', v') за допомогою обчислених полів оптичного потоку.

Для того, щоб отримати відповідну точку на іншому зображенні за допомогою оптичного потоку, треба просто змістити її на значення оптичного потоку між першим та другим зображенням:

$$(u', v') = (u, v) + flow_{u,v} \tag{5}$$

Тоді вектор між центром координат камери іншого кадру та відповідною точкою на іншому кадрі дорівнює

$$a' = \left(f, u' - c_x, v' - c_y\right) \tag{6}$$

Рис. 4.2. Відповідності між точками зображень
Fig. 4.2. Matches between image points

В такий спосіб, з наявної інформації про відповідності між точкою на кадрі, для якого будується мапа глибини, та точками на кожному іншому кадрі, отримуються промені у просторі, на перетині яких має безпосередньо лежати точка, що проектується у відповідні пікселі на двох кадрах.

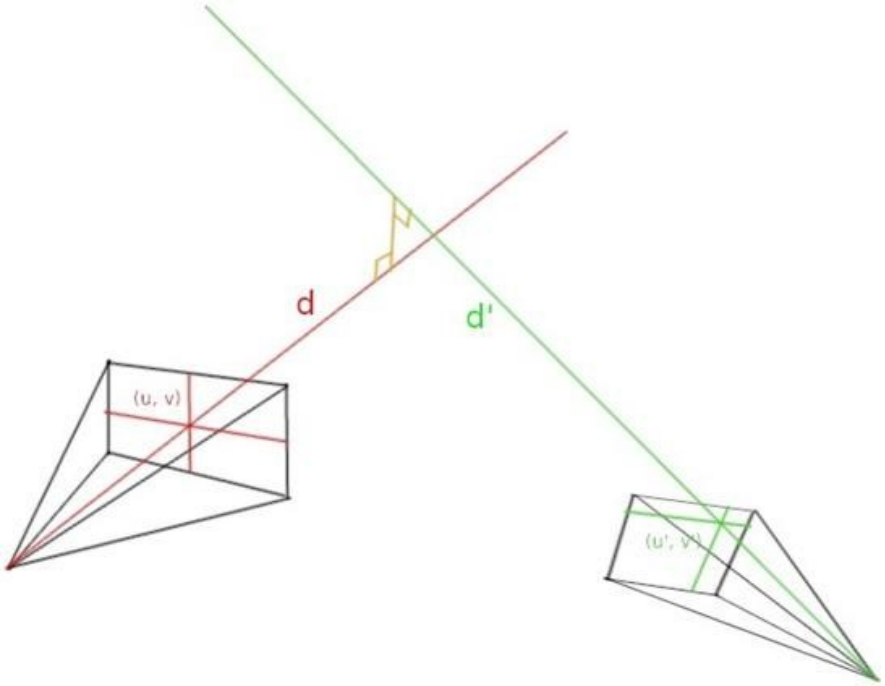


Рис. 4.3. Відстань до точки, що проектується на два кадри
Fig. 4.3. The distance to the point projected on two frames

На практиці, промені, що розглядаються, не будуть точно перетинатися, бо для знаходження відповідностей для одометрії використовується розріджене співставлення точок, за допомогою алгоритмів SIFT, а для знаходження щільних відповідностей використовується оптичний потік, отриманий за допомогою алгоритмів глибокого навчання. Тому, за допомогою такого методу точку перетину можна знайти лише з деякою точністю. Треба зауважити, що точку перетину при побудові мапи глибини знаходити не обов'язково, а лише треба знайти відстань d від початку променя у центрі координат камери до найближчої точки до прямої l' на прямій l.

Позначимо відповідний промінь як функцію від s:

$$l(s): l = as + b, s \geq 0 \tag{7}$$

Якщо вектор a нормалізований, то s дорівнює відстані від початку променя до $l(s)$. Таким чином, $d = s$.

Розглянемо прямі, що відповідають двом відповідним точках на двох кадрах:

$$\begin{aligned} l(d) &= ad + b \\ l'(d') &= a'd' + b' \end{aligned} \tag{8}$$

Тепер, задачу можна переформулювати як знаходження таких s, s', за яких відстань від $l(s)$ до $l'(s')$ мінімальна (рис 4.3).

Щоб вирішити цю задачу, треба прийняти до уваги той факт, що вектор між точками двох прямих у просторі, що має найменшу можливу довжину, перпендикулярний до обох прямих:

$$\begin{aligned} &< l(d) - l'(d'), a >= 0 \\ &< l(d) - l'(d'), a' >= 0 \end{aligned} \tag{9}$$

Дана система лінійна відносно d та d'. Безпосередньо після вирішення системи з двома рівняннями та невідомими, отримуємо значення d, що дорівнює відстані від фокусного центру камери до об'єкта у просторі, що відображається на обраний піксель.

Проводячи дані обчислення для кожного пікселя кадру, для якого будується мапа глибини стільки разів, скільки є кадрів, на яких було знайдено відповідний піксель до даного, отримуємо "гіпотези", які можна певним чином фільтрувати та усереднювати для отримання остаточного передбачення відстані до об'єкта. В імплементації цього алгоритма було вирішено брати до уваги тільки ті відповідності, для яких

$$\frac{\|l(d) - l'(d')\|}{d} \leq 0.01 \tag{10}$$

Також, для базової фільтрації хибних відповідностей, що генерує оптичний потік, алгоритм обчислює також оптичний потік між кадром, що є середнім за номером між вихідним та одним із кадрів на якому шукаються відповідності. Це означає, що крім оптичного потоку

між вихідним кадром I_j та кадром I_{j+k}, також обчислюється оптичний потік між $I_{j+\frac{k}{2}}$ та I_{j+k}.

Позначимо положення пікселів з кадру I_j на кадрі I_{j+k} як $I_j \to I_{j+k}$. Так, як оптичний потік між I_j та $I_{j+\frac{k}{2}}$ вже обчислений за побудовою алгоритму, то для кожного пікселя також можна знайти відповідність $(I_j \to I_{j+\frac{k}{2}}) \to I_{j+k}$.

Якщо оптичний потік знаходиться алгоритмом ідеально, то $I_j \to I_{j+k}$ повинне представляти ту ж саму відповідність що і $(I_j \to I_{j+\frac{k}{2}}) \to I_{j+k}$. Але на практиці, навіть такий точний алгоритм як RAFT, може давати хибні результати у ряді випадків. Для фільтрації хибних відповідностей у даній роботі пропонується перевіряти вищезазначену умову (10).

Проте, як і у випадку знаходження "приблизного" перетину між променями, треба задати деякий поріг допустимого відхилення. Загалом, це дає просте правило, що відсіює значну кількість хибних відповідностей.

Для усереднення результатів використовувалися емпіричні ваги, які виникли з ідеї про те, що при малих кутах між прямими, важливу роль мають помилки обчислення та неточності роботи алгоритмів одометрії та знаходження оптичного потоку, тому достовірність таких результатів знижується, і при нульовому куті – це взагалі вироджений випадок.

Простими для обчислення ваговими коефіцієнтами, що дозволяють "заглушити" шум від недостовірних результатів, є $sin^2(\alpha)$, де α – кут між α та α'. Обчислити, як відомо, ці коефіцієнти можна як

$$sin^2(\alpha) = 1 - cos^2(\alpha) = 1 - < a, a' >^2 \tag{11}$$

5. Висновки

У роботі були проаналізовані можливості застосування сучасних алгоритмів відновлення тривимірних сцен з зображень для відновлення просторової інформації із відео потоку. Були розглянуті методи, підходи та алгоритми, а також сучасні тренди в області реконструкції тривимірної інформації з відео. Приділено увагу послідовності розвитку підходів до вирішення задачі – можна підсумувати, що алгоритми відновлення тривимірної інформації починали розвиватися з суто геометричних підходів і в останні роки все більше використовують глибоке навчання. У процесі дослідження області та результатів, пов'язаних з тривимірною реконструкцією на основі зображень та відео отоків, був винайдений алгоритм, що дозволяє будувати щільні мапи глибини, використовуючи інформацію з усіх кадрів відео.

Конфлікт інтересів

Автори повідомляють про відсутність конфлікту інтересів.

References

1. Deep Two-View Structure-from-Motion Revisited. URL: https://openaccess.thecvf.com/content/CVPR2021/papers/Wang_Deep_Two-View_Structure-From-Motion_Revisited_CVPR_2021_paper.pdf
2. Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. 2020. Consistent video depth estimation. *ACM Trans. Graph*. 39, 4, Article 71 (August 2020), https://doi.org/10.1145/3386569.3392377

3. T. Caselitz, B. Steder, M. Ruhnke and W. Burgard, "Monocular camera localization in 3D LiDAR maps," 2016 *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1926-1931, DOI: https://doi.org/10.1109/IROS.2016.7759304
4. C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," in *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874-1890, Dec. 2021, https://doi.org/10.1109/TRO.2021.3075644
5. Teed Zachary, Lipson Lahav, Deng Jia, "Deep Patch Visual Odometry". arXiv e-print, 2022, DOI: https://doi.org/10.48550/arXiv.2208.04726
6. Lahav Lipson, Zachary Teed, Jia Deng, "RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching", arXiv e-print, 2021, https://doi.org/10.48550/arXiv.2109.07547
7. Jiankun Li, Peisen Wang, Pengfei Xiong, Tao Cai, Ziwei Yan, Lei Yang, Jiangyu Liu, Haoqiang Fan, Shuaicheng Liu, "Practical Stereo Matching via Cascaded Recurrent Network with Adaptive Correlation", arXiv e-print, 2022, https://doi.org/10.48550/arXiv.2203.11483
8. Richard Hartley, Andrew Zisserman, "Multiple View Geometry in Computer Vision", 2nd Edition, Cambridge University Press, 2003.
9. Weirong Chen, Suryansh Kumar, Fisher Yu, "Uncertainty-Driven Dense Two-View Structure from Motion", arXiv e-print , 2023, https://doi.org/10.48550/arXiv.2302.00523
10. Denys Hrulov (2024) *Analysis of Three-dimensional Scenes based on Video flow data* (master diploma) V. N. Karazin Kharkiv National University

RECONSTRUCTION OF THREE-DIMENSIONAL SCENES BASED ON VIDEO FLOW DATA

Denys Hrulov[1], Master's student; e-mail: xa11800855@student.karazin.ua;
ORCID: https://orcid.org/0009-0005-8506-770X
Anastasiia Morozova[1], Assistant Professor, PhD; e-mail: a.morozova@karazin.ua;
ORCID: https://orcid.org/0000-0003-2143-7992
Petro Dolia[1], Assistant Professor, PhD; e-mail: pdolya@karazin.ua;
ORCID: https://orcid.org/0009-0002-4062-4443
Liliia Bielova[1], Senior Lecturer; e-mail: l.belova@karazin.ua;
ORCID: https://orcid.org/0009-0007-0805-4547
[1] *V. N. Karazin Kharkiv National University, Ukraine*
Manuscript was received March 23, 2024; Received after review April 29, 2024; Accepted May 30, 2024

Abstract. This work is dedicated to the application of modern algorithms for reconstructing spatial scenes from images to restore spatial information from video. The work is looking at a variety of modern methods, approaches, algorithms and trends in the field. The attention was paid to the sequence of development of approaches to the completion of the task. While researching the field and results related to three-dimensional reconstruction based on images and video streams, an algorithm was invented that allows constructing dense depth maps using information from all video frames. The idea is to use ready-made, commonly accepted, and tested solutions to solve two problems: COLMAP for visual odometry, and RAFT for computing optical flow. The algorithm shows quite accurate results and reconstructs the depth map in detail on arbitrary static scenes.

Keywords: *video flow, 3D reconstruction, machine learning, odometry, neural network, computer vision, depth map, optical flow*

Conflicts of Interest: the authors declare no conflict of interest.

ISSN 2519-2310 (Online)

КОМП'ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА, 2024, № 1 (25)

COMPUTER SCIENCE AND CYBERSECURITY, 2024, Iss. 1 (25)

DOI: https://doi.org/10.26565/2519-2310-2024-1-07
УДК 004.056.5

РОЗРОБКА ТА РЕАЛІЗАЦІЯ МЕТОДА ПЕРЕВІРКИ ЦІЛІСНОСТІ ДИЗАЙНУ ОБ'ЄКТНО-ОРІЄНТОВАНОЇ СИСТЕМИ

Микита Пугач[1], аспірант кафедри теоретичної та прикладної інформатики, факультет математики та інформатики, e-mail: nikita.pugach.2000@gmail.com,
ORCID: https://orcid.org/0009-0004-8923-6489
Ірина Зарецька[1], доцент кафедри теоретичної та прикладної інформатики, факультет математики та інформатики, e-mail: zaretskaya@karazin.ua,
ORCID: https://orcid.org/0000-0001-8747-2737
[1]Харківський національний університет імені В.Н. Каразіна, майдан Свободи, 4, Харків, 61022, Україна
Рукопис надійшов 3 квітня 2024 р. Отримано після рецензування 5 травня 2024 р.
Прийнято 7 червня 2024 р.

Анотація: Для досягнення якості створення програмних продуктів, необхідно проводити різні заходи із тестування та верифікації на всіх етапах розробки, що є невід'ємним та одним з найважливіших етапів проектування ПЗ. У більшості моделей життєвого циклу програмного забезпечення (SDLC) даний етап є одним із перших, тож помилки, допущені при розробці дизайну приведуть до проблем у всіх наступних стадіях. Таким чином, через велику ціну помилки, важливою є перевірка цілісності розробленого дизайну на етапі проектування. У статті досліджується проблема пошуку протиріч у об'єктно-орієнтованому дизайні. Автори презентують набір протиріч, що можуть виникати у такому дизайні і ставлять за мету розробку методів виявлення та пошуку цих протиріч з метою покращення якості проектування, а також написання програмного забезпечення, що буде реалізовувати дані методи. Інструментом створення об'єктно-орієнтованого дизайну було обрано програму «diagrams.net», головною корисною рисою якої є можливість представлення створених діаграм у виді XML файлу у популярному форматі drawio. Автори пропонують метод за яким проводиться парсинг XML файлу діаграми і представлення її у виді набору об'єктів, таких як стрілки залежностей, класи, методи і т.д. Ці об'єкти повинні взаємодіяти за встановленими правилами. Порушення даних правил і є протиріччям об'єктно-орієнтованого дизайну. У результаті дослідження було представлено метод пошуку протиріч і реалізовано його на мові програмування Java.

Ключові слова: *розробка програмного забезпечення, UML, проектування програмного забезпечення, діаграми об'єктно-орієнтованого дизайну*

Як цитувати: Пугач М., Зарецька І.. Розробка та реалізація метода перевірки цілісності дизайну об'єктно-орієнтованої системи. *Комп'ютерні науки та кібербезпека.* 2024; № 1(25): С. 76–87. https://doi.org/10.26565/2519-2310-2024-1-07

In cites: Pugach M., Zaretska I. (2024). Development and implementation of a method for checking the integrity of the design of an object-oriented system. *Computer Science and Cybersecurity.* 1(25): 76–87. https://doi.org/10.26565/2519-2310-2024-1-07 (in Ukrainian)

1. Вступ

Проектування будь-якого програмного продукту є одним з найважливіших етапів розробки. Основна його мета полягає у створенні моделі майбутньої системи, яка буде визначати основні складові та взаємозв'язки. Результати проектування слугують фундаментом для подальшого написання програмного коду. Оскільки цей етап є дуже важким та витратним процесом розробки, дуже важливо максимально його оптимізувати та знайти найефективніші рішення для задоволення всіх вимог системи. Виявлення помилок, допущених на етапі проектування можуть у подальшому призвести до таких серйозних проблем, як збільшення часу виконання проекту, вслід чого збільшення його вартості, суттєве зниження якості готового програмного продукту, погіршення продуктивності системи або поганої масштабованості. Вкрай важливо приділяти багато уваги тому, щоб етап проектування не мав помилок, намагатися їх мінімізувати.

Стандартом у проектуванні програмного забезпечення є використання UML (Unified Modeling Language або уніфікованої мови моделювання). Він використовується для візуального представлення та документування програмних систем. UML надає нотацію та набір графічних символів для моделювання різних аспектів системи, таких як структура, функціональність, поведінка та взаємодія між компонентами.

В об'єктно орієнтованому дизайні може бути два типи протиріч. Перший, це коли протиріччя існує у рамках однієї діаграми. Другий, це протиріччя, що виникає на ґрунті несумісності двох UML діаграм. Такого виду протиріччя, що виникають на перетині двох або більше діаграм, не відслідковуються специфікацією UML, а отже не можуть бути перевірені Case-засобами. Протиріччя всередині однієї діаграми можуть виникати через те, що специфікація UML велика і Case-засоби не перевіряють повністю весь список вказаних там правил. А також існують ситуації, які виникають в рамках однієї діаграми і не описані в специфікації UML, але є несумісними з об'єктно орієнтованим дизайном. Наприклад, якщо в діаграмі класів є цикл залежностей, то це свідчить про те що вона була погано спроектована. Такі протиріччя також не можуть бути перевірені Case-засобами.

Мета даної статті це розробка методів виявлення та пошуку протиріч в об'єктно орієнтованому дизайні з метою покращення якості проектування, а також написання програмного забезпечення, що буде реалізовувати дані методи.

У статті розглядається проектування за допомогою мови моделювання UML. Чотири типи діаграм, що будуються з її допомогою складають основу на етапі проектування, це діаграми: класів, послідовностей, об'єктів та станів. Дослідження стосуються пошуку протиріч та несумісності саме в цих чотирьох діаграмах та між ними.

2. Огляд існуючих рішень

Для створення дизайну об'єктно-орієнтованих систем використовуються Case-засоби. Деякі сучасні Case-засоби, наприклад Rational Rose [11], мають можливість перевірки низки порушень цілісності дизайну на основі специфікації UML. Але вони є обмеженими і не надають можливість перевіряти цілісність між двома діаграмами різних видів.

Дослідження можливості верифікації цілісності між декількома видами діаграм проводилось у статтях «Cross-Diagram UML Design Verification» [9] та «Consistency of UML Design» [10]. Ці дослідження представляли дизайн у виді графової моделі та використовували логіку предикатів 1 порядку для пошуку протиріч, а також реалізовували пошук цих протиріч на мовах програмування Prolog та Java. Дане дослідження використовує більш простий підхід до пошуку порушень цілісності ніж вищеперераховані.

3. Опис протиріч

3.1. Протиріччя всередині діаграми класів, або між діаграмами класів

Діаграма класів – це UML-діаграма, яка описує систему, візуалізуючи різні типи об'єктів усередині системи та види статичних зв'язків, що існують між ними. Він також ілюструє операції та атрибути класів. Тому вкрай важливо звести до мінімуму протиріччя в цій діаграмі. В результаті аналізу специфікації UML 2.0 було знайдено такі види можливих протиріч:

- Клас має два або більше стереотипи, що не є сумісними між собою. Наприклад, клас одночасно позначений стереотипами «exception» та «enumeration».
- У випадку композиції кожен об’єкт-частина може належати тільки одному об’єкту-цілому. Для того, щоб ця вимога дотримувалася у діаграмі повинні бути враховані такі вимоги:
    - Кратність кінця-цілого при композиції не повинна перевищувати одиниці;
    - Один клас не може бути частиною більше ніж однієї композиції.
- Клас, що має стереотип «utility», не може містити у собі методи або атрибути з областю видимості екземпляру [4, с.682].
- Клас успадковує клас з іншого пакету. Така ситуація сигналізує про погано спроектований дизайн.
- Діаграма класів має циклічну залежність між класами. Така ситуація сигналізує про погано спроектований дизайн.
- Діаграма класів має циклічну залежність між пакетами. Така ситуація сигналізує про погано спроектований дизайн [2, с. 480].

3.2. Протиріччя між діаграмою класів та діаграмою послідовностей

У діаграмі послідовностей відображається послідовність взаємодії між сутностями системи, що проявляється за допомогою повідомлень. Описані в діаграмі послідовностей відносини не повинні суперечити тому, як вони описанні в діаграмі класів, це важливо врегулювати тому можна винести такий список протиріч.

- Діаграма послідовностей використовує клас, що не описаний у діаграмі класів.
- При відправленні повідомлення в діаграмі послідовностей відповідний метод у класі-отримувачі повинен мати відповідний ідентифікатор доступу. У випадку якщо взаємодіють нащадок та батьківський клас, тоді метод може мати ідентифікатор доступу public або protected. Якщо класи не пов’язані такими залежностями між собою, то метод може бути тільки public.
- При відправленні повідомлення у діаграмі послідовностей використовується метод, якого не існує у відповідному класі-отримувачі, що описаний у діаграмі класів.
- Класу можуть бути відправлені повідомлення з використанням виключно статичних методів.
- Об’єкт може відправляти повідомлення з використанням тільки статичних методів, якщо між класами задана асоціація і також явно вказана відсутність доступу класу-відправника до класу-отримувача.
- Якщо в діаграмі класів задана асоціація між двома класами з кратністю один, то на діаграмі послідовностей не може об’єкт першого класу посилати повідомлення до двох різних об’єктів другого класу.
- Об’єкти класу зі стереотипом «utility» не можуть існувати, бо такий клас не може існувати [4, с.682].

3.3. Протиріччя між діаграмою класів та діаграмою об’єктів

Діаграма об’єктів призначена для демонстрації сукупності об'єктів, що моделюються, і зв'язків між ними у фіксований момент часу. По суті є екземпляром діаграми класів. Тому усі зв’язки між об’єктами не повинні суперечити тим, що вказані в діаграмі класів. Спираючись на це можна виділити такі протиріччя:

- Значення поля класу не сумісне з його типом, що вказаний у діаграмі класів.
- При композиції об’єкт-частина може належати тільки одному об’єкту-цілому одночасно.
- При асоціації двох класів то кількість екземплярів першого класу та кількість екземплярів другого не повинні суперечити тому, яка кількість регламентована у діаграмі класів.
- Об’єкти класу зі стереотипом «utility» не можуть існувати, бо такі класи не можна ініціалізувати [4, с.682].
- На діаграмі послідовностей об’єкти зв’язані, але цього зв’язку не має у діаграмі класів. Для того, щоб об’єкти могли бути зв’язані необхідно, щоб виконувалася хоча б одна з двох таких умов:
    - Між відповідними класами або їх батьківськими класами існує зв’язок.
    - У одного з відповідних класів або його батьківського класу є атрибут, тип якого співпадає з іншим класом або одним з його батьківським класом.
- Об’єкт класу, що має стереотип «implementationClass», не може бути екземпляром більш ніж одного класу. Те саме стосується і його нащадків [4, с.681].
- Для кожного поля об’єкту має виконуватись одне з таких правил:
    - існує асоціація між класом об’єкта або одним з його батьківських класів та деяким іншим класом. При цьому роль, що відображена при асоціації не суперечить імені атрибута.
    - існує атрибут з таким самим ім’ям у класу об’єкта або у одного з його батьківських класів.
- Якщо жодне з даних правил не виконується, це свідчить про наявність протиріч.

3.4. Протиріччя між діаграмою станів та іншими діаграмами

Діаграма станів описує ті стани об’єктів, які вони можуть досягати в період свого життєвого циклу. Найчастіше перехід з одного стану до іншого відбувається за допомогою виклику методів. Так як всі можливі методи, що мають класи описані в діаграмі класів, важливо щоб методи присутні у діаграмі станів не суперечили їм. Також послідовність виклику методів описана в діаграмі послідовностей і описане в діаграмі станів не повинно з цим суперечити. Далі будуть описані протиріччя, що випливають з даних тверджень.

- Послідовність відправлень повідомлень у діаграмі послідовностей не повинна суперечити множині послідовностей переходів у діаграмі станів даного класу.
- Кожен метод, що використовується для зміни стану об’єкта у діаграмі станів, повинен бути описаним у діаграмі класів.

3.5. Протиріччя всередині діаграми станів

- Кожен стан об’єкту повинен бути досяжним з початкового стану.
- Кінцевий стан завжди має бути досяжним.

4. Опис структури XML файлу

XML файл формату drawio зберігає інформацію про елементи діаграми та їхню форму. Але він не дає чіткого визначення елементам, наприклад що це клас або залежність. Але кожен елемент має ряд характеристик, що дає змогу зрозуміти що саме знаходиться в елементі. Надалі буде представлений детальний опис наявних у файлі тегів:

- *mxfile* – головний тег, у якості атрибутів має дату останнього редагування діаграми, версію застосунка та іншу технічну інформацію. Включає в себе теги diagram.
- *diagram* – це тег, який саме включає в себе діаграму. У якості атрибутів має id та назву діаграми. Включає у себе тег mxGraphModel.
- *mxGraphModel* – тег, що в атрибутах має інформацію про площину, на якій малюється діаграма, її розміри, тощо. Включає в себе тег root.
- *root* – це тег, що не має атрибутів, він включає в себе набір тегів mxCell.
- *mxCell* – це тег, що є основним в побудові діаграми. Усі її елементи, такі як: клас, метод, поле, залежність (якщо ми говоримо про діаграму класів), а також стрілки виклику функцій та значень, що повертаються (в контексті діаграм послідовностей). Тег mxCell може включати тег mxGeometry. Тег має такий список атрибутів:
  - *id* – унікальний ідентифікатор кожного елементу
  - *value* – текст елементу
  - *style* – рядок, що включає CSS стиль елементу
  - *parent* – це id елементу який на діаграмі включає у себе поточний елемент. Наприклад, parent елементу, що відображає метод буде id елементу, що відображає клас, який має цей метод.
  - *source* – наявний у стрілок, являє собою id елементу від якого прямує стрілка.
  - *target* – також специфічний для стрілок атрибут. Це id елемента до якого прямує стрілка.
  - *connectable* – атрибут наявний у описах до стрілок. Наприклад, при відображенні агрегації подібний елемент може зберігати інформацію про те скільки об’єктів буде агреговано.
- *mxGeometry* – тег, що відповідає за інформацію про те, скільки простору буде займати елемент або іншу геометричну інформацію. Має атрибути: X, Y, width, height. Може зберігати в собі теги mxRectangle або mxPoint.
- *mxRectangle* – являє собою опис прямокутника, як геометричної фігури на площині, має такі самі характеристики, як і тег mxGeometry. Зазвичай цей тег з’являється у класів.
- *mxPoint* – це опис точки, що може бути початком, кінцем, або точкою зламу для стрілки на діаграмі.

Опис відповідностей між тегами XML та елементами діаграми:

Усі згадані теги описують геометричні фігури, їх стиль та написи на них, а також чітко прив’язують одну фігуру до іншої за допомогою атрибутів parent, source та target. Це дає змогу виділити унікальність кожного елемента діаграми, а також класифікувати його. Далі розглянемо, які саме особливості дають змогу точно класифікувати елементи.

Почнемо з діаграми класів:

- *Клас* – як і усі елементи діаграми описується тегом mxCell. Особливість тегів mxCell, що описують саме класи це те, що всередині вкладеного тегу mxGeometry має ще тег mxRectangle. При цьому mxGeometry буде зберігати інформацію про розмір прямокутника, у якому записана лише назва класу, а mxRectangle - розміри усього класу. Це обумовлено тим, що від класу до класу прямують стрілки залежностей, і чіткий опис границь необхідний для встановлення до яких саме класів вони належать.

- *Поля класу* – також описані тегами mxCell. Тег root містить у собі список тегів mxCell, при чому ці теги впорядковані таким чином, що всі поля та методи класу записані безпосередньо після тегу самого класу. Також вони обов'язково мають в атрибуті 'parent' id класу, якому належить. Також усі поля класу – це наступні теги mxCell після тегу самого класу.
- *Методи класу* – усі методи класу – це наступні теги mxCell після полів класу та одного тегу mxCell з порожнім атрибутом value. На діаграмі – це роздільна лінія між полями та методами класу. mxCell методу має в атрибуті 'parent' id класу, якому належить. У разі, якщо клас не має полів, усе одно буде присутній порожній простір і лінія, що відмежовує назву класу та його методи на діаграмі. А в XML файлі це означає, що mxCell з порожнім атрибутом value збережеться.
- *Стрілка залежності* – це ті теги mxCell, має атрибути source та target. Для різних типів залежностей стрілка може описуватися одним тегом mxCell або кількома. Наприклад, якщо це агрегація або композиція, то буде використаний один тег та у атрибуті value буде зазначено кількість залежних класів. Наслідування та реалізація мають порожній рядок в атрибуті value, з тою відмінністю, що реалізація у атрибуті style має рядок 'dashed=1', тобто пунктирна лінія.

Діаграма послідовностей також відображає класи, методи та залежності, але в інший спосіб. Опис класифікації елементів діаграми послідовностей:

- *Клас* – особливість класів у діаграмі послідовностей – це наявність лінії, від якої йдуть виклики методів у хронологічному порядку. Даний сервіс малює також прямокутник на лінії, саме від нього йдуть виклики, тобто його можна вважати частиною класу. В цьому випадку найпростіший спосіб розпізнати клас – це знайти mxCell тег, що в атрибуті style буде мати рядок 'perimeter=lifelinePerimeter', це буде сам клас. А також mxCell, що має в цьому атрибуті рядок 'perimeter=orthogonalPerimeter', а в атрибуті parent id класу – це прямокутник, що належить класу. Його важливо запам'ятовувати, бо стрілка виклику функції у початку або в кінці може прямувати саме до прямокутника і мати у source або target id саме прямокутника.
- *Метод* – відображається стрілкою виклику методу. Може бути заданий двома чи однією стрілкою. Перша містить інформацію про назву методу та набір аргументів. Друга – про значення, що повертається. Другої стрілки може не бути у випадках, коли метод нічого не повертає. Головною їх відмінністю є те, що стрілка значення, що повертається, завжди є пунктирною. За цією особливістю можна її розпізнати, воно у атрибуті style містить рядок 'dashed=1'. А на основі того, до якого класу прямує стрілка виклику, можна зробити висновок якому класу належить метод. На відміну від методів, які класифікуються з діаграми класів, методи з діаграми послідовностей будуть зберігати інформацію про класи, що їх викликають. Ця інформація буде корисна у подальшому пошуку протиріч.
- *Залежності* – вони не задаються явно і не можуть бути чітко встановленими. Головне – це те, що якщо об'єкт класу може викликати метод іншого класу тільки якщо він має якусь залежність із ним.

3. Методи верифікації цілісності

Головною метою дослідження є написання програми, що буде розпізнавати протиріччя в межах однієї діаграми та між двома діаграмами.

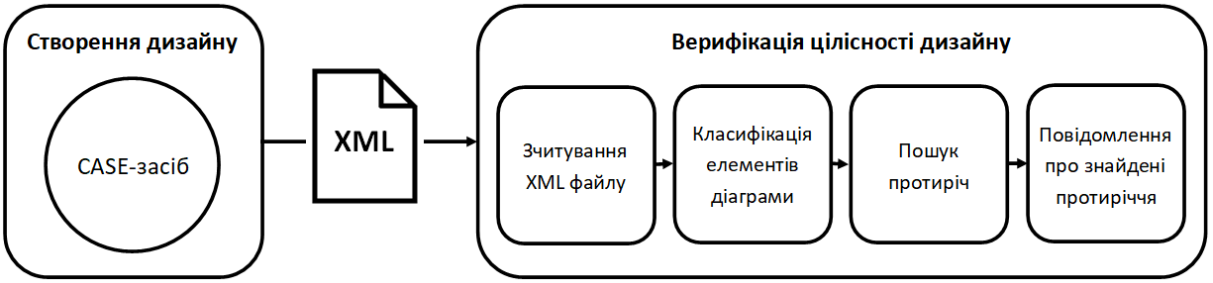


Рис.1 - Загальна схема пошуку протиріч.
Fig.1 - General schema of finding conflicts.

Результатом парсингу XML файлів діаграм класів та послідовностей є списки об’єктів, що описують класи, представленні у діаграмах.Далі будуть описані методи пошуку конкретних протиріч:

1. *У діаграмі послідовностей є клас, що не описаний у діаграмі класів.* Маючи список класів з діаграми класів, порівнюємо його зі списком класів з діаграми послідовностей. Якщо другий список містить класи, яких немає у першому, то це порушення цілісності.

2. *Нелегітимне використання методів класу.* Виклики методів у діаграмі послідовностей повинні узгоджуватись з діаграмою класів. Можливість використання методів інших класів залежить від багатьох факторів, кожен із яких має бути перевіреним. Надалі буде представлений опис перевірки кожного фактору:

2.1. *Перевірка чи є відповідний метод у класі, що викликається.* У метода є три характерні особливості, що для програми роблять його унікальним – це значення, що він повертає, назва та список аргументів. Перевіряти потрібно всі, отже:

2.1.1. *Назва.* Перевірка назви буде виконуватись першою і перевіряти чи є хоч один метод з такою назвою. Перевірка робиться рекурсивно для всіх батьківських класів, з модифікаторами доступу public, default та protected.

2.1.2. *Список аргументів.* З попереднього кроку формується список методів, що мають потрібну назву. Далі у цьому списку шукаються методи з необхідними списками аргументів.

2.1.3. *Значення, що повертається.* У списку класів, що сформований на попередніх кроках, тобто з потрібною назвою та списком аргументів, шукаємо метод з відповідним значенням, що повертається.

2.2. *Перевірка на легітимність виклику методу.* Після того, як метод, що викликається у діаграмі послідовностей, знайдений у діаграмі класів, перевіряємо його рівень доступу. Якщо private, то легітимним буде тільки його виклик цим же класом. Якщо protected, то рекурсивно перевіряються ієрархія класу, що містить цей метод. Якщо клас, що його викликає, не є його батьківським – це порушення цілісності. За назвою пакету перевіряється default. Для public перевірка не проводиться.

3. *Приналежність об’єкта-частини до кількох об’єктів-цілих при композиції.* Це протиріччя можна виявити в двох ситуаціях:

3.1. *Кратність композиції більше 1.* Композиція в діаграмі класів має позначку кратності. Після парсингу XML файлу ця інформація зберігається. Для перевірки використовується регулярний вираз, що знайде композицію, у якій вказана кратність більша за одиницю.

3.2. *Об’єкт є частиною більш ніж однієї композиції.* Для пошуку береться список усіх композицій в діаграмі класів. З цього списку утворюємо список класів, що є залежними в цих композиціях. Далі перевіряється чи містить знайдений список повторення.

4. *Циклічна залежність класів.* Діаграма класів представляється у вигляді орієнтованого графа, у якому вершини – це класи, а ребра – залежності. Далі проводиться пошук циклів у орієнтованому графі.

5. *Несумісні стереотипи класу.* Після парсингу XML файлу інформація про стереотипи зберігається. Програма, що перевіряє містить інформацію, про стереотипи, що не є сумісними. Тож для всіх класів перевіряється список стереотипів.

6. Застосунок на мові програмування Java, що реалізує описані методи пошуку протиріч

У ході роботи була створена програма, що перевіряє наявність протиріч у двох типах UML діаграм об'єктно-орієнтованого програмування: діаграмі класів та діаграмі послідовностей, та між ними. Програма написана на мові програмування Java. Програмний код розділено на 5 пакетів на основі зон відповідальності класів, які вони містять. Послідовність використання цих пакетів зображено на діаграмі.

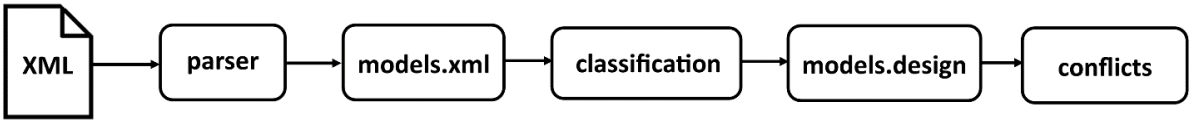


Рис.2 - Послідовність використання програмних пакетів.
Fig.2 - The sequence of program packages usage.

Далі розберемо як саме програма реалізує перевірку цілісності об'єктно-орієнтованого дизайну:

1. *Зчитування XML файлу, що описує діаграму, та представлення її у виді сукупності об'єктів різних типів у середині програми.*

Опис сутностей, створених у програмі для відображення сутностей XML файлу. Те, які теги та атрибути має XML файл діаграм було описано вище. Для зчитування і зберігання даних з файлу були створені спеціальні класи у програмі. Ці класи знаходяться у пакеті **models.xml**.

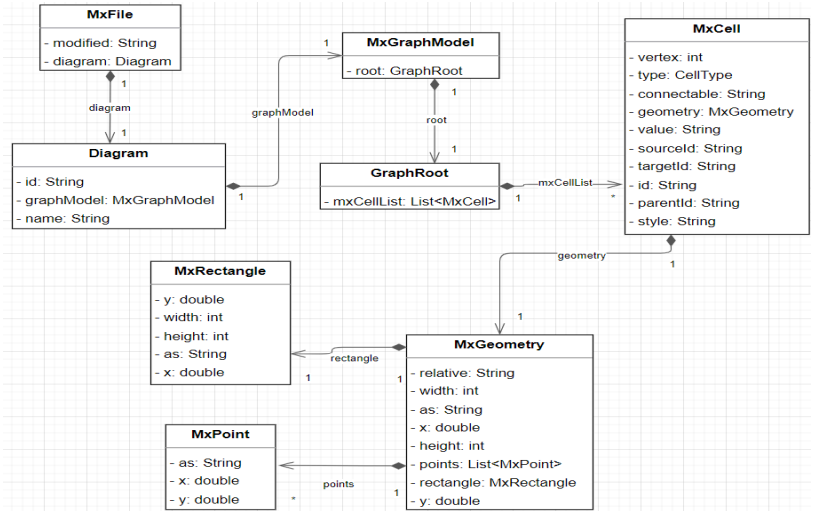


Рис.3 - Діаграма класів пакету **models.xml**.
Fig.3 - **models.xml** package class diagram.

Опис сутностей, що відображають елементи UML діаграм класів та послідовностей у програмі. Класи, що зберігаються в пакеті **models.xml**, не відображають UML діаграму у зрозумілому для об’єктно-орієнтованого програмування вигляді. Вони скоріш описують те, як малювати ту чи іншу діаграму. Сама ж UML діаграма містить такі сутності, як, наприклад, клас, метод, чи залежність. Тож для ефективного пошуку протиріч програма повинна зберігати діаграму саме в таких сутностях. Для цього були створені спеціальні класи, що зберігаються у пакеті **models.design**.

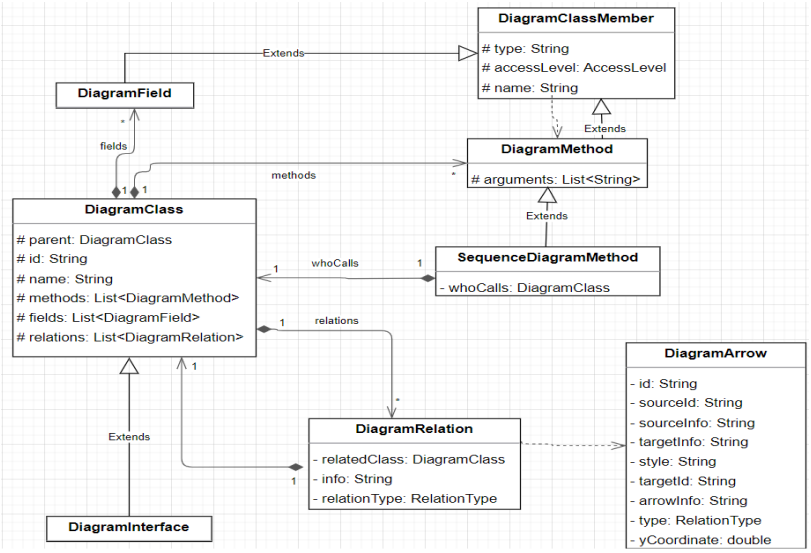


Рис.4 - Діаграма класів пакету **models.design**.
Fig.4 - **models.design** package class diagram.

Зчитування та парсинг XML файлу. Тепер, коли програмамає класи, у які буде записуватися інформація про діаграму, її можна зчитувати з XML файлу. Для зчитування та парсингу XML була використана бібліотека **org.xml.sax**. Результатом парсингу буде набір об’єктів класів з пакету **models.xml**. Класи, що займаються парсингом знаходяться у пакеті **parser**.

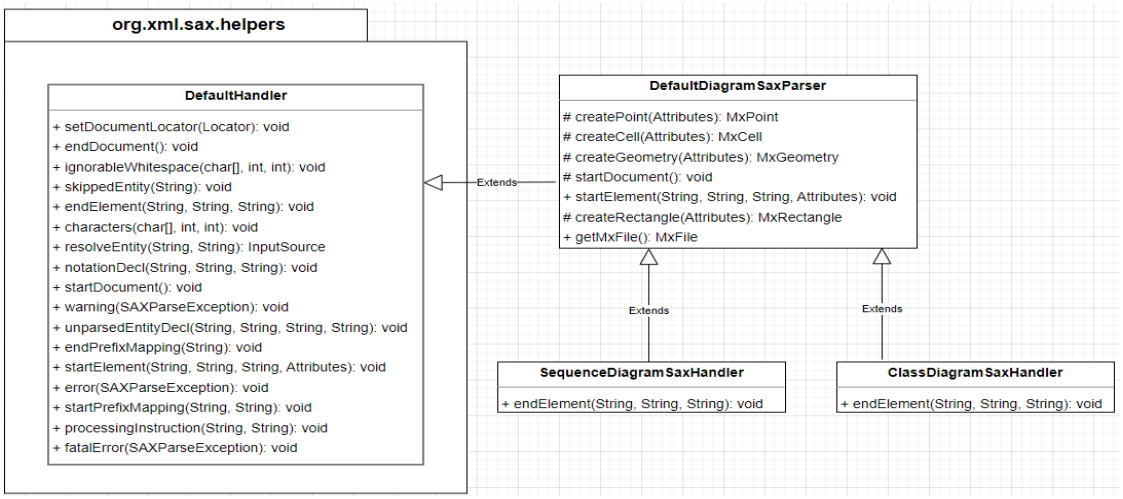


Рис.5 - Діаграма класів пакету **parser**.
Fig.5 - **parser** package class diagram.

Класифікація елементів UML діаграм. Наступна задача з набору об'єктів класів з пакету **models.xml**, що були отримані в минулому етапі класифікувати саме елементи діаграми, тобто класи, методи, поля, залежності, тощо. Для цього потрібно якимось чином створити об'єкти класів з пакету **models.design**. З цією метою було створено пакет класів **classification**.

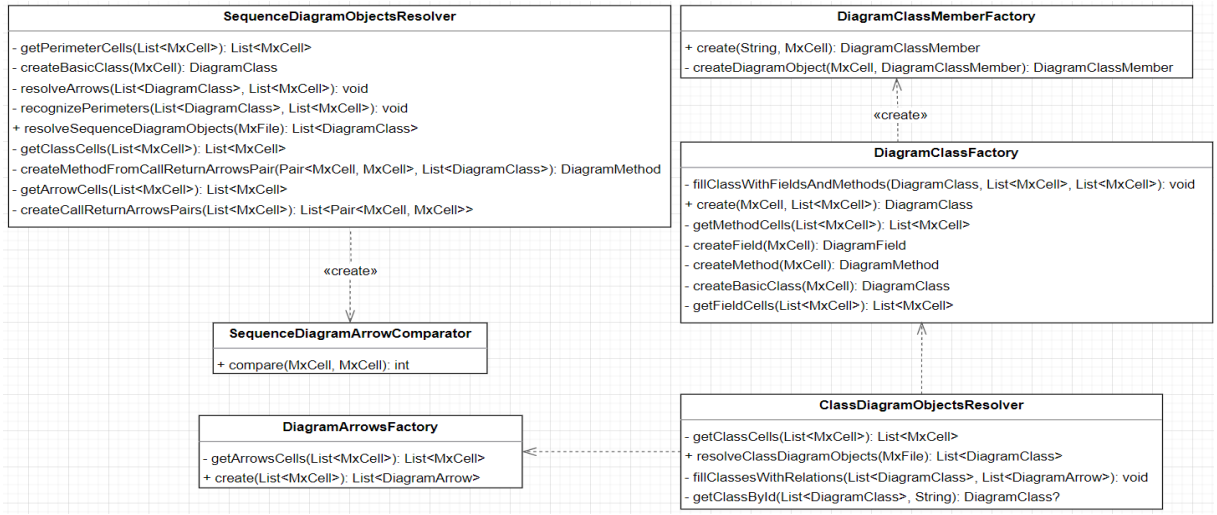


Рис.6 - Діаграма класів пакету **classification**.
Fig.6 - **classification** package class diagram.

2. *Пошук протиріч.*

Останнім і найголовнішим етапом буде пошук протиріч. Після класифікації, дані діаграми знаходяться у зручному для їх обробки вигляді. Усі сутності в програмі відповідають дійсним сутностям діаграм UML. Усі класи, що займаються пошуком протиріч знаходяться в пакеті **conflicts**.

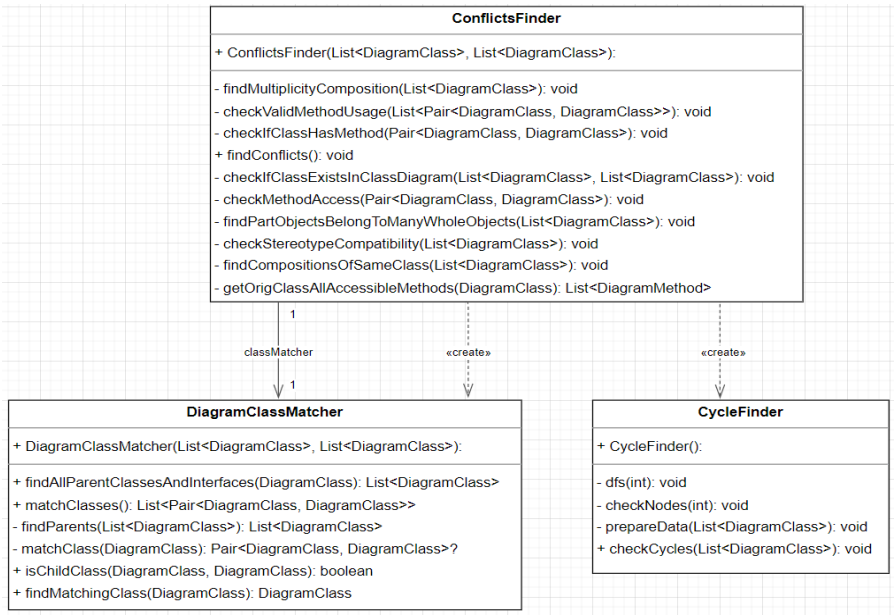


Рис.7 - Діаграма класів пакету **conflicts**.
Fig.7 - **conflicts** package class diagram.

7. Висновки

У результаті проведеного дослідження було виявлено близько двадцяти протиріч, що можуть виникати у об’єктно-орієнтованому дизайні. Були розроблені методи знаходження та виявлення протиріч. А також реалізовано додаток на мові програмування Java, який успішно реалізує виведені протиріччя.

Перспективою розвитку даної роботи можна вказати наступне. По-перше, виявлення ще більшої кількості потенційних протиріч у діаграмах об’єктно орієнтованого дизайну. По-друге, є дуже великі перспективи по масштабуванню та удосконаленню застосунку для пошуку протиріч. Є дуже багато варіантів його покращення, починаючи з удосконалення алгоритмів, закінчуючи універсальністю вхідних даних, тобто варіантів представлення діаграм, що подаються на вхід до програми, та вдосконаленням користувацького досвіду.

Завершуючи, можна сказати, що результати цього дослідження сприятимуть покращенню процесу проектування програмного забезпечення та розвитку даної галузі в цілому.

Конфлікт інтересів

Автори повідомляють про відсутність конфлікту інтересів.

References

1. Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, & Jim Conallen. (2007). *Object-Oriented Analysis and Design with Applications* (3rd ed.). https://zjnu2017.github.io/OOAD/reading/Object.Oriented.Analysis.and.Design.with.Applications.3rd.Edition.by.Booch.pdf
2. Craig Larman. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. https://bsituos.weebly.com/uploads/2/5/2/5/25253721/applying-uml-and-patterns-3rd.pdf
3. Vanessa Weber, Kleinner Farias, Lucian Gonçales & Vinícius Bischoff. (2016). Detecting Inconsistencies in Multi-view UML Models. *International Journal of Computer Science and Software Engineering (IJCSSE)*, Volume 5, Issue 12. https://www.researchgate.net/publication/313837603_Detecting_Inconsistencies_in_Multi-view_UML_Models
4. OMG. Unified Modeling Language 2.5.1 Specification. (2017). https://www.omg.org/spec/UML/2.5.1/
5. Gamma Erich, Helm Richard, Johnson Ralph & Vlissides John. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. https://www.javier8a.com/itc/bd1/articulo.pdf
6. Robert C. Martin. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. https://ptgmedia.pearsoncmg.com/images/9780132928472/samplepages/0132928477.pdf
7. Martin Fowler, Kent Beck, John Brant, William Opdyke & Don Roberts. (1999). Refactoring: Improving the Design of Existing Code. https://ptgmedia.pearsoncmg.com/images/9780201485677/samplepages/9780201485677.pdf
8. Joyce Farrell. (2017). Programming Logic and Design, Introductory. https://jamborebook.co/download/4867679-program-logic-and-design
9. Iryna Zaretska, Oleksandra Kulankhina & Hlib Mykhailenko. Cross-Diagram UML Design Verification. *ICT in Education, Research and Industrial Applications. CCIS*, Vol. 347, Springer-Verlag, Berlin Heidelberg (2013). – pp. 165-176. http://dx.doi.org/10.1007/978-3-642-35737-4_10
10. Iryna Zaretska, Oleksandra Kulankhina, Hlib Mykhailenko & Tamara Butenko. Consistency of UML Design. *International Journal of Information Technology and Computer Science (IJITCS)*, Vol.10, No.9, 2018. – pp. 47-56. https://doi.org/10.5815/ijitcs.2018.09.06
11. Rational Rose. https://www.ibm.com/docs/en/rational-clearquest/7.1.0?topic=developing-schemas-clearquest-designer
12. Diagrams.Net. https://app.diagrams.net/

DEVELOPMENT AND IMPLEMENTATION OF A METHOD FOR CHECKING THE INTEGRITY OF THE DESIGN OF AN OBJECT-ORIENTED SYSTEM

Mykyta Pugach[1], PhD student, Department of Theoretical and Applied Informatics, Faculty of Mathematics and Informatics; e-mail: nikita.pugach.2000@gmail.com;
ORCID: https://orcid.org/0009-0004-8923-6489
Iryna Zaretska[1], associate professor of the institution of higher education, Department of Theoretical and Applied Informatics, Faculty of Mathematics and Informatics; e-mail: zaretskaya@karazin.ua;
ORCID: https://orcid.org/0000-0001-8747-2737
[1] *V. N. Karazin Kharkiv National University, Ukraine*
Manuscript was received April 3, 2024; Received after review May 5, 2024; Accepted June 7, 2024

Abstract. Creating modern software products is a complex and long process consisting of many parts. To achieve quality, it is necessary to carry out various measures for testing and verifying software at all stages of development. This article discusses the software design stage, which is integral and one of the most important. In most software development life cycle (SDLC) models, this stage is one of the first, so design mistakes will lead to problems in all subsequent stages. Thus, due to the high cost of error, it is very important to check the integrity of the developed design at the design stage. The article examines the problem of finding contradictions in object-oriented design. The authors present a set of contradictions that can arise in such a design and aim to develop methods and algorithms for detecting and searching for these contradictions in order to improve the quality of the design, as well as writing software that will implement these algorithms and methods. The program "diagrams.net" was chosen as a tool for creating object-oriented design, the main useful feature of which is the ability to present the created diagrams in the form of an XML file in the popular drawio format. The authors of the study propose a method for parsing the XML file of the diagram and presenting it as a set of objects, such as dependency arrows, classes, methods, etc. These objects must interact according to the established rules.The violation of these rules is a contradiction of the object-oriented design. As a result of the study, a method of finding contradictions was presented and implemented in the Java programming language.

Keywords: *software development, UML, software design, object-oriented design diagrams*

Conflicts of Interest: the authors declare no conflict of interest.

Електронне наукове видання

КОМП’ЮТЕРНІ НАУКИ ТА КІБЕРБЕЗПЕКА
№ 1(25) 2024

Міжнародний електронний науково-теоретичний журнал

Українською та англійською мовами

Комп’ютерне верстання – Єсіна М.В., Власова В.В.

Підписано до розміщення 28.08.2024. Гарнітура Times New Roman.
Ум. друк. арк. 4,92. Обсяг 4,9 Мб. Зам. № 24/24.

Харківський національний університет імені В. Н. Каразіна,
61022, м. Харків, майдан Свободи, 4.
Свідоцтво суб’єкта видавничої справи ДК № 3367 від 13.01.2009