КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.34.1264
УДК 004.056.53

Добришин Юрій Євгенович
кандидат технічних наук, доцент
Національна академія Служби безпеки України, Київ, Україна
ORCID:0000-0003-2473-9507
ydobryshyn@gmail.com

АВТОМАТИЗАЦІЯ ПРОЄКТНОГО РІШЕННЯ ЩОДО ВІДНОВЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВНАСЛІДОК ДІЇ КІБЕРАТАК

Анотація. У статті розглянуто актуальну науково-прикладну проблему автоматизації процесів відновлення пошкодженого програмного забезпечення внаслідок дії кібератак, що набуває особливої важливості в умовах зростання кількості та складності кіберзагроз. Проведено аналіз сучасних наукових досліджень, який показав, що основними напрямами розвитку є формалізація процесів відновлення на основі математичних моделей, а також застосування методів штучного інтелекту та машинного навчання. Запропоновано представлення процесу відновлення пошкодженого програмного забезпечення у вигляді сукупності взаємопов'язаних задач, організованих за принципом багаторівневої декомпозиції. Кожна задача описується як система множин, що включає вхідні дані, обмеження, варіанти рішень, критерії оцінювання, моделі та процедури їх реалізації. Це дозволяє формалізувати процес прийняття рішень, визначити залежності між задачами та забезпечити їх алгоритмізацію. У роботі також наведено приклад математичної моделі вибору оптимального способу відновлення програмного забезпечення після атак програм-вимагачів. Модель враховує структуру програмної системи, множину дефектів, їх властивості та критичність, а також доступні способи відновлення з урахуванням витрат часу, ресурсів і ризиків. Запропоновано цільову функцію оптимізації з ваговими коефіцієнтами, що дозволяє визначити найефективнішу стратегію відновлення з урахуванням заданих критеріїв. Отримані результати свідчать про доцільність використання формалізованих підходів у поєднанні з інтелектуальними методами для підвищення ефективності процесів відновлення програмного забезпечення. Запропонований підхід створює теоретичну основу для розроблення інтелектуальних систем підтримки прийняття рішень у сфері кібербезпеки. Перспективи подальших досліджень пов'язані з розробкою адаптивних і самоорганізованих моделей, інтеграцією із системами моніторингу та управління інцидентами, а також створенням галузевих баз знань для підвищення точності діагностики та ефективності відновлення.

Ключові слова: кібербезпека, кібератаки, відновлення програмного забезпечення, автоматизація, формалізація, математична модель, прийняття рішень, штучний інтелект.

ВСТУП

Постановка проблеми. Проблема автоматизації процесів відновлення пошкодженого програмного забезпечення внаслідок дії кібератак є однією з ключових проблем у сучасній кібербезпеці. Зростання кількості, складності та різноманітності кібератак призводить до суттєвих порушень функціонування інформаційних систем, що особливо критично для об'єктів критичної інфраструктури, державного управління та бізнес-середовища. У таких умовах традиційні підходи до відновлення, які базуються переважно на ручному аналізі та втручанні фахівців, стають недостатньо ефективними через значні витрати часу та ресурсів.

Основною проблемою є складність формалізації процесів відновлення, оскільки кожна кібератака має унікальні характеристики, сценарії розвитку та наслідки для програмного забезпечення. Це ускладнює створення універсальних алгоритмів, які могли б автоматично визначати тип пошкодження, оцінювати його критичність і обирати оптимальний спосіб відновлення. Крім того, відновлення програмного забезпечення потребує врахування багатьох факторів, зокрема архітектури системи, наявних ресурсів, рівня пошкодження даних та вимог до безперервності функціонування.

Ще однією суттєвою проблемою є необхідність інтеграції процесів відновлення з іншими компонентами систем кіберзахисту, такими як системи виявлення вторгнень, моніторингу подій безпеки

та управління інцидентами. Відсутність узгодженості між цими компонентами ускладнює автоматизацію, оскільки потребує обробки великої кількості різнорідних даних у реальному часі.

Також важливим аспектом є забезпечення достовірності діагностики пошкоджень, адже помилки на цьому етапі можуть призвести до неправильного вибору стратегії відновлення та поглиблення наслідків атаки.

Незважаючи на розвиток методів штучного інтелекту та машинного навчання, їх застосування у цій сфері кіберзахисту інформаційних ресурсів, також супроводжується низкою труднощів. Зокрема, існує проблема недостатньої кількості якісних навчальних даних, складності інтерпретації результатів моделей та забезпечення їх надійності в умовах нових, раніше невідомих типів атак. Крім того, автоматизовані рішення повинні бути адаптивними та здатними до самонавчання, що потребує значних обчислювальних ресурсів і складної реалізації.

Таким чином, проблема автоматизації процесів відновлення пошкодженого програмного забезпечення полягає у необхідності поєднання формалізованих підходів, інтелектуальних методів та системної інтеграції різних компонентів кіберзахисту. Вирішення цієї проблеми потребує розробки універсальних моделей і методів, здатних забезпечити швидке, точне та ефективне відновлення програмних систем у умовах постійно зростаючих кіберзагроз.

Аналіз останніх досліджень і публікацій. Проблемам відновлення пошкодженого програмного забезпечення внаслідок дії кібератак присвячена значна кількість наукових праць, як вітчизняних та і закордонних фахівців [1-21]. Аналіз досліджень авторів свідчать про те, що відновлення пошкодженого програмного забезпечення, стає все більш актуальним в умовах зростання кількості кібератак на державні, комерційні інформаційні ресурси та об’єкти критичної інфраструктури [1, 2].

Фахівцями наукових праць стверджується, що застосування у сучасних умовах різного роду кібератак має комплексний характер, їх вплив суттєво дії на ключові властивості інформації, а саме: доступність, цілісність та конфіденційність [1]. У зв’язку з цим основна увага приділяється аналізу технологічних процесів відновлення та розробці методів формалізації процесів реагування і відновлення пошкодженого програмного забезпечення [2].

У більшості досліджень використовується як правило системний підхід до аналізу процесів відновлення пошкодженого програмного забезпечення [3, 4], згідно з яким відновлення пошкодженого програмного забезпечення розглядається як окрема складова частина життєвого циклу програмного забезпечення, опис та рішення якої потребує врахування технічних, організаційних та алгоритмічних аспектів [4].

Враховуючі складність та непередбаченість сучасних кібератак, ряд авторів [5] зазначають, що ключовим завданням на теперішній час є мінімізація наслідків дії від атак. Це потребує забезпечення швидкого відновлення автоматизованих інформаційних систем та формування підходів до проєктування автоматизованих систем із вбудованими механізмами автоматичного відновлення пошкодженого програмного забезпечення.

Сучасні дослідження демонструють активне застосування методів штучного інтелекту для автоматизації процесів усунення дефектів [6, 7, 8, 9], та використання методів автоматизації відновлення програмного забезпечення, які базуються на формальних моделях, що описують складові автоматизованих систем [10].

Значна увага приділяється використання методів машинного навчання, які дозволяють, використовуючи властивості кібератак, прогнозувати та класифікувати інциденти, надавати пропозиції щодо вибору способів відновлювання дефектів пошкодженого програмного забезпечення [11], запроваджувати адаптивний підхід щодо автоматичного формування стратегій відновлення. Методи машинного навчання дозволяють представити кібератаку як формалізований набір параметрів, що може бути використаний для автоматичного формування плану відновлення дефектів та суттєво зменшити час їх діагностування [11].

У сучасних дослідженнях значна увага приділяється створенню інтелектуальних систем підтримки прийняття рішень у сфері кібербезпеки , де на підставі формалізованого аналізу кібератак, формується каталог подій і обирається оптимальний сценарій їх автоматичного відновлення [12-13]. Це забезпечує перехід до автоматизованого формування стратегій відновлення пошкодженого програмного забезпечення.

Важливим напрямом досліджень у галузі автоматизованого проєктування технологічних процесів усунення дефектів програмного забезпечення, є формалізація процесів відновлення пошкодженого програмного забезпечення після кібератак [14, 15, 16]. У зазначених наукових роботах запропоновано певний підхід, який передбачає представлення процесу відновлення пошкодженого програмного забезпечення як задачу прийняття рішень, із урахування ресурсних обмежень. Автори описують методи формалізації параметрів атаки, моделювання пошкодженого стану програмного забезпечення,

визначають множину відновлювальних дій, та у межах цього підходу пропонують вибір з множини альтернатив оптимальну стратегію відновлення програмного забезпечення.

Інтерес представляють окремі наукові роботи [17], у яких автори, аналізуючи пошкоджене програмне забезпечення стверджують, що важливим елементом відмовостійкості програмного забезпечення є відновлення після збоїв.

Так у науковій праці [17], група авторів, аналізуючи дефекти програмного забезпечення, орієнтуються на локальне відновлення його компонентів, яке є ефективним підходом, за якого відновлюються лише помилкові частини системи, тоді як інші частини залишаються доступними. У роботі зазначається, що для досягнення локального відновлення пошкодженого програмного забезпечення, його архітектуру необхідно розбити на окремі блоки, які можна відновити ізольовано. Для чого авторами у роботі пропонується системний підхід, спрямований на оптимізацію розбиття архітектури програмного забезпечення для локального відновлення. Цей підхід нагадує процес декомпозиції технології відновлення програмного забезпечення, дозволяє сформувати пропозиції та рекомендації щодо автоматизації проєктного рішення щодо відновлення компонентів програмного забезпечення.

Наукові фахівці у своєї роботі [18] вважають, що аналіз першопричин, а саме, процесів діагностування, є однією з найважливіших операцій під час проєктування технології відновлення програмного забезпечення. Автори стверджують, що існуючі поширені підходи, що базуються на кореляції даних або повних знаннях експертів у предметній області відновлення програмного забезпечення, є неточними або непрактичними у більшості промислових випадків, оскільки кореляція не означає причинно-наслідкового зв'язку, а експерти в предметній області можуть не мати повних знань про складні системи в реальному часі. Тому у роботі пропонується нова модель знань про причинно-наслідкові зв'язки між компонентами технологічного процесу відновлення програмного забезпечення, що дозволяє експертам у предметній області вносити знання про предметну область для аналізу першопричин виходу з ладу програмного забезпечення.

Автори наукової праці [18] представляють, алгоритм, який за допомогою процесів виявлення, покращення, уточнення та віднімання причинно-наслідкового графа здатний вивести підграф зв'язків між компонентами програмного забезпечення. Це дозволяє, з урахуванням розробленого причинно-наслідкового графа, побудувати інформаційну модель з подальшим застосуванням її для автоматизації проєктування відновлення пошкодженого програмного забезпечення.

Враховуючи, що відновлення програмного забезпечення розглядається, як критично важливий процес, спрямований на повернення системи до працездатного стану після збоїв, атак або помилок, ряд фахівців наукових праць [19-21] з метою ефективного аналізу та оптимізації процесу відновлення пропонують використовувати графові моделі, які дозволяють представити залежності між етапами щодо появи дефектів, їх виявлення та призначення оптимальної технології відновлення.

У науковій статті [19] та дослідженні [20] запропоновано модель виявлення програм, яка базується на використанні графових нейронних мереж для аналізу поведінки систем, ідентифікації аномалій та процесу навчання. У роботі [21] представлено моделювання кібератак із застосуванням засобів візуальної аналітики. Запропонований у цьому дослідженні підхід сприяє більш точному виявленню та розумінню структури атак, а також дозволяє обрати ефективний спосіб відновлення пошкодженого програмного забезпечення.

Таким чином, на підставі аналізу літературних джерел необхідно зазначити, що сучасний стан досліджень щодо автоматизації проєктування процесів відновлення пошкодженого програмного забезпечення передбачає розробку, використання та удосконалення декількох напрямків діяльності фахівців у галузі автоматизації.

Головним із них є формалізований підхід, який базується на розробці та застосування математичних моделей процесів відновлення пошкодженого програмного забезпечення. Іншим, інтелектуальний, який передбачає використання методів штучного інтелекту. Комплексне поєднання вказаних напрямків створює передумови для автоматизації проєктування процесів відновлення пошкодженого програмного забезпечення.

Але незважаючи на дослідження, розробку та застосування вказаних підходів, на теперішній час, залишається значна кількість проблем, пов’язаних з відсутністю універсальних моделей щодо автоматизації проєктного рішення щодо відновлення програмного забезпечення внаслідок дії кібератак.

Розробка та застосування таких моделей, надавало би можливість вирішувати певні ризики під час виявлення та діагностування дефектів програмного забезпечення, а також забезпечувати процес автоматизованого формування планів відновлення дефектів та у подальшому інтегрування до сучасних систем підтримки прийняття рішень, що підтверджує актуальність тематики дослідження.

Формалізація задач проєктування через використання математичного апарату множин, повинна включати визначення вхідних даних, обмежень, проєктних рішень, критеріїв оцінювання та відповідних моделей. Це дозволяє забезпечити уніфіковане представлення процесу прийняття рішень, визначити порядок їх виконання (послідовний або паралельний), а також встановити залежності між задачами різних рівнів декомпозиції. Така формалізація сприяє підвищенню обґрунтованості вибору способів відновлення та забезпечує можливість їх повторного використання в аналогічних ситуаціях.

Побудова комплексної моделі процесу автоматизації, яка включає морфологічну, інформаційну, математичну та алгоритмічну складові дозволяє описати як структуру задач, так і процеси обробки інформації між ними, включаючи як автоматизовані процедури, так і інтерактивні етапи прийняття рішень. У результаті формується теоретична основа для створення інтелектуальних систем підтримки прийняття рішень, спрямованих на підвищення ефективності, швидкості та надійності відновлення програмного забезпечення в умовах кіберзагроз.

Мета статті. Метою даної статті є розроблення формалізованого підходу до автоматизації проєктних рішень щодо відновлення пошкодженого програмного забезпечення внаслідок дії кібератак. Основна увага приділяється представленню процесу відновлення як сукупності взаємопов'язаних задач, організованих у вигляді багаторівневої декомпозиції, що дозволяє структурувати складні процеси, визначити логічні залежності між окремими етапами та забезпечити їх подальшу алгоритмізацію. Такий підхід створює передумови для ефективного застосування сучасних інформаційних технологій і мов програмування при побудові автоматизованих систем реагування на кіберінциденти.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Процес автоматизації проєктного рішення щодо відновлення пошкодженого програмного забезпечення внаслідок дії кібератак, у своїй основі представляє сукупність логічно впорядкованих проєктних процедур, формалізований опис яких полегшує їх алгоритмізацію, робить процес проєктування придатним для подальшої обробки за допомогою сучасних мов програмування та у подальшому є основою для створення та застосування автоматизованих систем виявлення та реагування на кібератаки, які загрожують безпеці, конфіденційності чи цілісності інформаційних систем та даних.

Для формального представлення проєктного рішення щодо відновлення пошкодженого програмного забезпечення виконаємо його декомпозицію (див. рис.1).

За результатами декомпозиції можна виділити безліч окремих задач $\{A\}_{ij}$, де i – номер рівня декомпозиції у структурі вихідної задачі, j - порядковий номер завдання на i -му рівні.

Зазначимо відношення, що мають місце між задачами різних рівнів та задачами одного рівня, позначивши їх відповідно $\{D\}_a$, $\{Q\}_a$. Якщо існує задача $\{A\}_{ij}$ яка належить до одного рівня $\{Q\}_a$, то вказана задача $\{A\}_{ij}$ є частиною задач рівня $\{Q\}_a$.

Якщо задача $\{A\}_{ij}$ одного рівня $\{Q\}_a$ пов'язана з задачами $\{A\}_{ik}$, різних рівнів $\{D\}_a$, то розв'язання задачі $\{A\}_{ij}$ неможливе без попереднього розв'язання задачі $\{A\}_{ik}$.

Розглядаючи склад завдань, що виникають при проєктуванні рішення щодо відновлення пошкодженого програмного забезпечення внаслідок дії кібератак, можна виділити на кожному рівні певну кількість блоків, для яких необхідно провести математичну інтерпретацію.

Так, на першому рівні, рівні грубої декомпозиції, це задачі від $\{A\}_{11}$ до $\{A\}_{14}$. Для другого рівня характерна середня декомпозиція з задачами $\{A\}_{21}$ до $\{A\}_{28}$. Третій рівень, рівень точної декомпозиції, вирішує конкретні задачі від $\{A\}_{31}$ до $\{A\}_{313}$, безпосередньо пов'язані з проєктуванням рішення щодо відновлення пошкодженого програмного забезпечення.

Як було зазначено раніше, кожна задача може бути представлена у вигляді системи множин:

$$A_{ij}=\{W_{ij}, O_{ij}, P_{ij}, K_{ij}, F_{ij}, M_{ij}\} \quad (1)$$

де: W_{ij} – вхідні дані, O_{ij} – обмеження, P_{ij} – проектні рішення, K_{ij} – оцінки проектних рішень, F_{ij} – вирішальна процедура, M_{ij} – модель задачі, яка належить множині моделей M_a.

Отримані, за результатами проєктування, рішення кожної конкретної задачі можна описати у вигляді наступного математичного виразу:

$$F_{ij}=\{W_{ij}, O_{ij}, K_{ij}\}\rightarrow P_{ij}\rightarrow M_{ij} \quad (2)$$

де → логічна операція слідування між компонентами множин.

Позначимо множину рішень для окремої задачі $\{A\}_{ijn}$ як $\{p\}_{ij}$ метою їх використання у якості вхідних даних для наступної задачі $\{A\}_{ijs}$. Враховуючі те, щодо відношення між задачами справедливо для одного рівня декомпозиції, можна записати наступній вираз:

$$\{Q\}_a = \{Q_{injs}\} = \{A_{ijn}, A_{ijs,}\, p_{ijns}\ \} \tag{3}$$

З урахуванням окремих рішень, які будуть здійснені на кожному рівні декомпозиції процесу відновлення пошкодженого програмного забезпечення, позначим схему впорядкованих проєктних процедур як $\{L\}$, тоді процес автоматизації проєктного рішення щодо відновлення пошкодженого програмного забезпечення внаслідок дії кібератак можна загально записати у вигляді множин, які приймають участь у проєктуванні:

$$L=\{A_{ij}\,, O_a\,, Q_{a,}\, M_a\} \tag{4}$$

де, O_a – обмеження, які існують під час рішення задач.

M_a – множина математичних моделей рішення задач.

Проєктування процесу автоматизації рішення щодо відновлення пошкодженого програмного забезпечення може бути виражене через морфологічну, інформаційну, математичну та алгоритмічну моделі. Морфологічна модель $\{L_1\}$ будується шляхом проведення ретельної декомпозиції задачі проєктування з урахуванням того, що вирішення певного кола завдань можливе шляхом використання математичних моделей, які адекватно вирішують кожну задачу на різних рівнях декомпозиції. Для інших завдань процедура вибору проєктних рішень може бути вирішена лише у режимі діалогу. Виходячи з цього задачу проєктування рішення щодо відновлення пошкодженого програмного забезпечення внаслідок дії кібератак можна записати як об’єднання двох множин:

$$A_n=\left\{A^{\,f}_{ij}\right\} \cup \left\{A^{\,n}_{ji}\right\} \tag{5}$$

Для грубої декомпозиції (див. рис.1) порядок виконання проєктних задач буде наступним:

$$\ni_{i=1} \ni [A_{11} \to A_{12} \to A_{13} \to A_{14}] \tag{6}$$

де: $\to$ знак слідування.

На рівні середньої і точної декомпозиції частина завдань виконується паралельно, тобто, вони пов'язані відношенням сумісності, яке позначається як $\leftrightarrow$. Інші завдання пов'язані відношенням слідування $\to$, тобто. виконуються у певному порядку. На рівні середньої декомпозиції цей порядок такий:

$$\ni_{i=2} \ni \{[A_{21} \leftrightarrow A_{22}]\} \to \{[A_{23} \leftrightarrow A_{24}]\} \to \{[A_{25} \to A_{26}]\} \to \{[A_{27} \to A_{28}]\} \tag{7}$$

На рівні точної декомпозиції деталізується порядок виконання приватних задач:

$$\ni_{i=3} \ni \{[A_{31}]\} \to \{[A_{32} \leftrightarrow A_{33}]\} \to \{[A_{34} \to A_{35} \leftrightarrow A_{36} \to A_{37}]\} \to \{[A_{38} \leftrightarrow A_{39}]\} \to \{[A_{310} \leftrightarrow A_{311}]\} \to \{[A_{312} \leftrightarrow A_{313}]\} \tag{8}$$

Інформаційна модель $\{L_2\}$ відображає процес циркуляції інформації між задачами, які визначаються на останньому рівні декомпозиції проєктного рішення щодо відновлення пошкодженого програмного забезпечення. У процесі проєктування інформація обробляється і перетворюється на блоках переробки інформації з урахуванням алгоритмічних процедур і точках діалогу, у яких проєктування ведеться в інтерактивному режимі.

Масиви інформації, що переробляється, складаються з постійної частини, що зберігається в базах даних, і змінної частини, що вводиться проєктувальником для відображення конкретної виробничої ситуації.

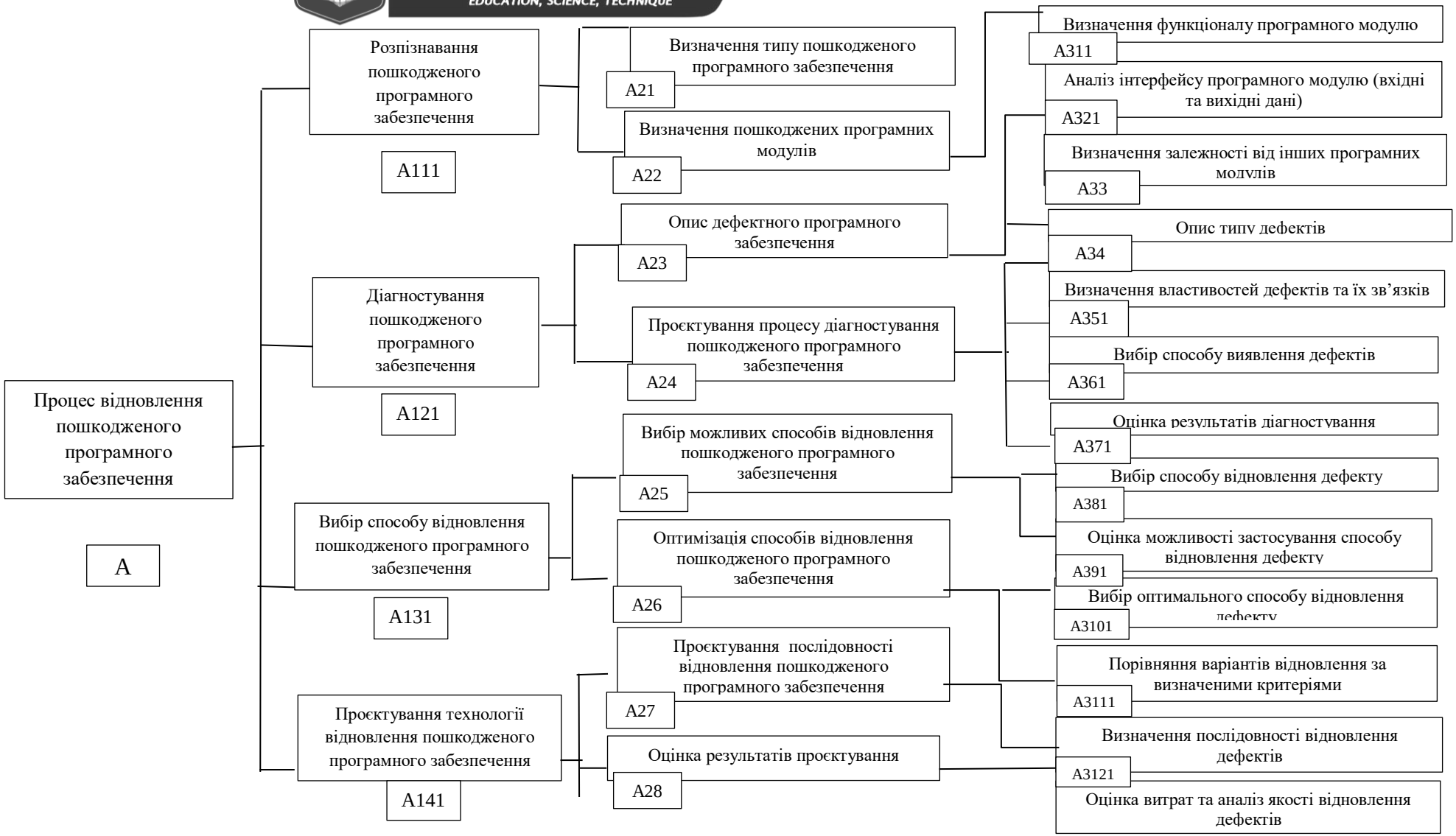

Рис 1. Декомпозиція проектного рішення щодо відновлення пошкодженого програмного забезпечення

Інформаційна модель $\{L_2\}$ може бути представлена графом (рис. 2)

$$G = \{A_n, B_n, V_n\} \quad (9)$$

де, для задач рівня точної декомпозиції $A_{ij} \in A_n$ існують зовнішні джерела інформації - бази даних $b_{ij} \in B_n$, а дуги графа $v_{ij} \in V_n$показують інформаційні зв'язки.

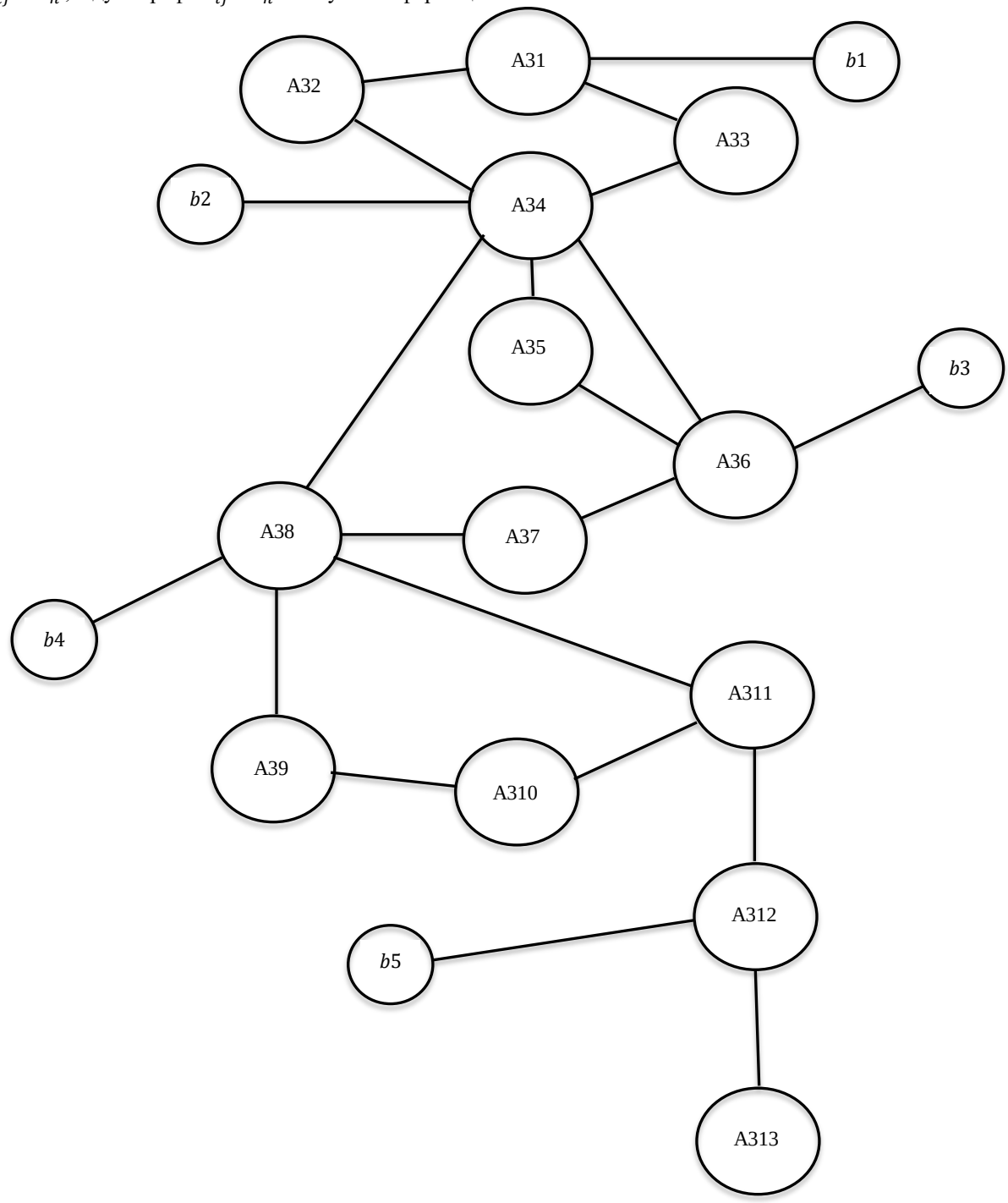

Рис.2 Інформаційна модель процесу відновлення пошкодженого програмного забезпечення

На рис. 2. зображені зовнішні джерела інформації, представлені у вигляді баз даних B_n, а саме: b_1-база даних щодо характеристик та складу пошкодженого програмного забезпечення, b_2 – база даних щодо властивостей дефектів пошкодженого програмного забезпечення, b_3 – база даних способів виявлення дефектів пошкодженого програмного забезпечення, b_4 – база даних способів відновлення пошкодженого програмного забезпечення, b_5 – база даних маршрутів послідовності відновлення пошкодженого програмного забезпечення.

Кожна задача проєктування вирішується на основі адекватної математичної моделі $\{L_3\}$або логіко-евристичної процедури, і тому розробка її є найважливішим етапом при створенні формалізованої методики проєктування процесу відновлення пошкодженого програмного забезпечення. При чому головним моментом є те, що для кожної задачі вибирається математичний апарат (метод), який дозволяє перетворити задачу проєктування у доступну алгоритмічну модель $\{L_4\}$, з подальшим застосуванням засобами автоматизованого програмування.

Таким чином процедуру автоматизації проєктного рішення щодо відновлення пошкодженого програмного забезпечення внаслідок дії кібератак можна представити наступним чином:

$$[L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow L_4] \tag{10}$$

ПРАКТИЧНЕ ЗАСТОСУВАННЯ ДЕКОМПОЗИЦІЇ ПРОЄКТНОГО РІШЕННЯ

На підставі розробленої декомпозиції проєктного рішення щодо відновлення пошкодженого програмного забезпечення внаслідок дії кібератак, наведемо окремий приклад застосування математичної моделі для прийняття рішень щодо відновлення програмного забезпечення після атак програм-вимагачів. Ця модель відображає проєктне рішення у вигляді формальної системи та інтегрує результати діагностики пошкодженого програмного забезпечення, залежності програмних модулів та виявлених дефектів, альтернативні способи відновлення та критерії оцінки.

Представимо пошкоджене програмне забезпечення після дії атак програм-вимагачів P_z, де $z \in \{PR\}$, у вигляді набору дефектних програмних модулів P_i де $i \in \{PM\}$:

$$P_i = \{P_1, P_2, \dots P_i \dots P_n\} \tag{11}$$

де кожен модуль виконує певну функцію в системі. Зв'язок між модулями пошкодженого програмного забезпечення можна представити у вигляді графа залежностей:

$$G_m = \{P_n, E_n\} \tag{12}$$

де, P_n множина програмних модулів;
E_n дуги графа, які показують інформаційні зв'язки між модулями.

Під час атаки програми-вимагача підмножина модулів пошкоджується або стає недоступною.

Представимо множину дефектів, пошкодженого програмного забезпечення, як G_g де $g_f \in \{GP\}$. У свою чергу характеристики кожного з дефектів пошкодженого програмного забезпечення внаслідок впливу кібератак g_f, де $f \in \{GP\}$ можна описати за допомогою їх властивостей, а саме:

$$g_f = \{S_g^1, S_g^2, \quad S_g^3 \dots \dots, S_g^f\} \tag{13}$$

Властивості дефектів пошкодженого програмного забезпечення та їх зав’язків представимо кортежом:

$$g_f = \{S_g^f, P_{ln}\} \tag{14}$$

де, S_g^f – властивість дефекту (шифрування, пошкодження, шкідлива модифікація тощо);
P_{ln} – розташування дефекту в системі пошкодженого програмного забезпечення;
C_g – рівень критичності дефекту.

Рівень критичності дозволяє розмістити всі дефекти в порядку убування, з урахуванням властивостей кожного дефекту. На першому місті будуть знаходитися дефекти, властивості яких найбільш суттєво впливають на працездатність програмного забезпечення.

Інші дефекти, які можна вважати неосновними, можуть розглядатися як умовно допустимі і пов'язані з основними дефектами певними закономірностями. Тобто оцінка результатів діагностування буде передбачати, що втрати від величини неосновного дефекту невеликі, тому в окремих випадках їх можна не враховувати. Вказана умова дозволяє зменшити число дефектів під час діагностування та суттєво зменшити вказану технологічну операцію.

На підставі чого оцінка результатів діагностування буде передбачати наступну залежність:

$$g_{fні} = F(g_{foi}) \tag{15}$$

де, $g_{fні}$ величина неосновного дефекту, g_{foi} величина основного дефекту.

Під час проведення діагностування дефектів пошкодженого програмного забезпечення однієї з важливих ознак класифікації є розподіл дефектів у залежності від способу їх виявлення $\{S_{vka}\}$. Таки ознаки забезпечують оптимальне використання спеціалізованого програмного та технічного забезпечення, сприяють оптимальному застосуванню операцій щодо налаштування системи моніторингу та виявлення кібератак.

У формальному вигляді умови сумісності між дефектом g_f, де $f \in \{GP\}$ та способом його виявлення S_{va_i} можна записати таким чином:

$$\forall g_f \forall S_{vai} \exists S_g \exists S_{va} \ni P\{(g_f\left[S_g^i\right] = S_{vai}[S_{va}^f])\} \to [g_f \leftrightarrow S_{vai}] \tag{16}$$

де, S_{va}^f – властивості способів виявлення дефектів від атак програм-вимагачів.

Представимо набір способів відновлення пошкодженого програмного забезпечення після дії атак програм-вимагачів R_m, де $m \ni \{RM\}$, у вигляді множини доступних способів відновлення:

$$R_m = \{R_1, R_2, \dots R_n\} \tag{17}$$

Доступні способи відновлення включають: відновлення з резервних копій, перевстановлення пошкоджених програмних модулів, використання засобів розшифрування файлів, відновлення конфігурації пошкодженого програмного забезпечення. Кожний спосіб має різні властивості.

Сформулюємо умови сумісності між дефектом g_f, де $f \in \{GP\}$ та способом його відновлення R_m де $m \in \{RM\}$. У формальному вигляді це можливо записати таким чином:

$$\forall g_f \ \forall R_m \ \exists S_g \ \exists S_r \ni P\left\{(g_f\left[S_g^{\text{n}}\right] = R_m[S_r^{\text{n}}])\right\} \to [g_f \leftrightarrow R_m] \tag{18}$$

Введемо бінарну змінну вибору способу відновлення для кожного дефекту:

x_{fm} =1, якщо для дефекту g_f, обрано спосіб відновлення R_m;

x_{fm} = 0, якщо з урахуванням умов сумісності (16), на дефект накладається обмеження по способу відновлення.

Кожному способу відновлення пошкодженого програмного забезпечення після впливу кібератак відповідають певні витрати ресурсів V_{fm}, часу та ризику:

$$V_{fm} = \{t_{fm}, c_{fm}, r_{fm}\} \tag{19}$$

де: t_{fm} – час відновлення дефекту g_f способом R_m; c_{fm} – вартість відновлення; r_{fm} – ризик (ймовірність невдалого відновлення або втрати даних).

На вибір оптимального способу відновлення дефекту пошкодженого програмного забезпечення також можуть впливати окремі критерії. Тоді цільову функцію оптимізації F_oможна представити у вигляді певних критеріїв:

$$\min F_o = \sum_{f \in GP} \left(\sum_{m \in RM} x_{fm} \ (\alpha t_{fm}, + \beta c_{fm}, + \gamma r_{fm}) \right) \tag{20}$$

де α, β, γ – вагові коефіцієнти важливості критеріїв.

З урахуванням введеного раніше рівня критичності дефекту C_g, можна модифікувати цільову функцію:

$$\min F_o = \sum_{f \in GP} \left(\sum_{m \in RM} x_{fm} \; C_g^f \left(\alpha t_{fm} , + \beta c_{fm} , + \gamma r_{fm} \right) \right) \quad (21)$$

Це дозволяє надати пріоритет усуненню критичних дефектів у подальшому визначити послідовність усунення дефектів пошкодженого програмного забезпечення.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У результаті проведеного дослідження встановлено, що проблема відновлення пошкодженого програмного забезпечення внаслідок дії кібератак є складною багатофакторною задачею, яка потребує системного та формалізованого підходу. Показано, що традиційні методи відновлення, які ґрунтуються на ручному аналізі, не відповідають сучасним вимогам оперативності, точності та ефективності реагування на кіберінциденти, особливо в умовах функціонування об’єктів критичної інфраструктури. Це обумовлює необхідність автоматизації процесів прийняття рішень щодо відновлення програмного забезпечення.

У роботі узагальнено сучасні наукові підходи до автоматизації відновлення програмного забезпечення та встановлено, що ключовими напрямами є формалізація процесів відновлення на основі математичних моделей і застосування методів штучного інтелекту. Обґрунтовано доцільність представлення процесу відновлення як задачі прийняття рішень, що дозволяє враховувати множину альтернатив, обмежень і критеріїв ефективності.

Запропоновано формалізований підхід до автоматизації проєктних рішень, який базується на багаторівневій декомпозиції процесу відновлення та представленні окремих задач у вигляді систем множин.

Такий підхід дозволяє структурувати процес відновлення, визначити взаємозв’язки між його етапами, забезпечити можливість алгоритмізації та подальшої реалізації у вигляді програмних засобів. Важливим результатом є формування комплексної моделі, що включає морфологічну, інформаційну, математичну та алгоритмічну складові, які у сукупності забезпечують цілісне представлення процесу автоматизації.

Розроблена математична модель вибору способів відновлення пошкодженого програмного забезпечення дозволяє враховувати характеристики дефектів, їх критичність, залежності між програмними модулями, а також ресурси, час і ризики відновлення. Використання цільової функції з ваговими коефіцієнтами забезпечує можливість оптимізації процесу відновлення з урахуванням заданих критеріїв, що підвищує обґрунтованість прийняття рішень та ефективність усунення наслідків кібератак.

Разом з тим встановлено, що незважаючи на наявні наукові напрацювання, проблема створення універсальних моделей автоматизації процесів відновлення залишається відкритою. Основними обмеженнями є складність формалізації різноманітних сценаріїв кібератак, недостатня інтеграція з існуючими системами кіберзахисту, а також обмеженість даних для навчання інтелектуальних систем.

Перспективи подальших досліджень полягають у розробці адаптивних та самоорганізованих моделей відновлення програмного забезпечення, здатних ефективно функціонувати в умовах невизначеності та появи нових типів кібератак. Особливо актуальним є поєднання формалізованих математичних підходів із методами машинного навчання та штучного інтелекту для створення інтелектуальних систем підтримки прийняття рішень.

Крім того, доцільним є дослідження питань інтеграції розроблених моделей із системами моніторингу, виявлення вторгнень та управління інцидентами, що дозволить забезпечити комплексну автоматизацію процесів реагування на кібератаки. Важливим напрямом є також створення галузевих баз знань щодо дефектів програмного забезпечення та способів їх усунення, що сприятиме підвищенню точності діагностики та ефективності відновлення.

Таким чином, подальший розвиток досліджень у цьому напрямі сприятиме створенню ефективних автоматизованих систем відновлення програмного забезпечення, здатних забезпечити високий рівень стійкості інформаційних систем до сучасних кіберзагроз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kozlovskyi, V. V., & Kopiika, O. O. (2020). Analysis of modern cyber threats and methods of information systems protection. *Information Protection, 22*(3), 45-52.
2. Lytvynenko, O. V., & Kovalenko, A. P. (2021). Cybersecurity of critical infrastructure: Current state and development prospects. *Information Processing Systems*, (2), 120-126.

3. Ilau, M.-C., Baldwin, A., Caulfield, T., & Pym, D. (2025). Modelling and simulating organizational ransomware recovery: Structure, methodology, and decisions. *Journal of Cybersecurity*, *11*(1), Article tyaf035. https://doi.org/10.1093/cybsec/tyaf035
4. Zhao, M., & Chen, R. (2025). Automated cybersecurity incident response: A reinforcement learning approach. *AI and Data Science Journal*, *2*(1), 23-28.
5. Leventopoulos, S., Pipyros, K., & Gritzalis, D. (2024). Retaliating against cyber-attacks: A decision-taking framework for policy-makers and enforcers of international and cybersecurity law. *International Cybersecurity Law Review*, *5*, 237-262.
6. Bondarenko, O. I. (2020). Intelligent decision support systems in cybersecurity. *Information Technologies and Security*, (1), 15-22.
7. Zhang, Q., & Chen, M. (2020). AI-driven self-healing cloud systems. *IEEE Cloud Computing*, *7*(3), 40-50.
8. Kenmogne, L. A., & Mocanu, S. (2024). Explainable AI for process-aware attack detection in industrial control systems. In *Proceedings of the 2024 IEEE 10th International Conference on Network Softwarization (NetSoft)* (pp. 363-368). IEEE. https://doi.org/10.1109/NETSOFT60951.2024.10588940
9. Kasali, K., Orekha, P., & Usoro, S. (2025). AI-driven strategies for mitigating cybersecurity threats in U.S. small and medium enterprises (SMEs). *International Journal of Computer Science and Information Technologies*, *16*(3), 110-116.
10. Petrenko, S. O. (2022). Formalization of software recovery method selection after cyberattacks. *Bulletin of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". Series: Informatics, Control and Computer Engineering*, (75), 78-85.
11. Dolhova, N. H., Shapovalova, O. O., & Solodovnyk, H. V. (2025). Decision support system for cybersecurity design of critical infrastructure. *Cybersecurity: Education, Science, Technique, 1*(29), 348-368.
12. Dobryshyn, Yu. Ye., Sydorenko, S. M., & Vorokhob, M. V. (2023). Automated decision support system for recovering software damaged by cyberattacks. *Cybersecurity: Education, Science, Technique, 4*(20), 174-180.
13. Van der Kleij, R., Cadet, B., & Young, H. (2022). Developing decision support for cybersecurity threat and incident managers. *Computers & Security, 113,* Article 102535. https://doi.org/10.1016/j.cose.2021.102535
14. Lu, P., Zhang, L., Liu, M., et al. (2024). Recovery from adversarial attacks in cyber-physical systems: Shallow, deep and exploratory works. *ACM Computing Surveys*. https://doi.org/10.1145/3653974
15. Kott, A., Weisman, M. J., & Vandekerckhove, J. (2022). Mathematical modeling of cyber resilience. In *Proceedings of the MILCOM 2022 IEEE Military Communications Conference*. IEEE. https://doi.org/10.1109/MILCOM55135.2022.10017731
16. Sözer, H., Tekinerdogan, B., & Aksit, M. (2014). Optimizing decomposition of software architecture for local recovery. *Software Quality Journal*, *22*(4), 703-736.
17. Tonon, A., Zhang, M., Caglayan, B., Shen, F., Gui, T., Wang, M., & Zhou, R. (2025). *RADICE: Causal graph based root cause analysis for system performance diagnostic* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2501.11545
18. Li, J., Yang, G., & Shao, Y. (2023). Ransomware detection method based on TextGCN. In *Proceedings of the 6th International Conference on Artificial Intelligence and Big Data (ICAIBD)*. Chengdu, China.
19. Wei, R., Cai, L., Yu, A., & Meng, D. (2021). DeepHunter: A graph neural network based approach for robust cyber threat hunting. In *Security and Privacy in Communication Networks* (pp. 3-24). https://arxiv.org/abs/2104.09806
20. Rabzelj, M., Bohak, C., Južnič, L., Kos, A., & Sedlar, U. (2023). Cyberattack graph modeling for visual analytics. *IEEE Access, 11*, 1-34. https://doi.org/10.1109/ACCESS.2023.3304640

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (34), 2026
ISSN 2663 – 4023

Yurii Dobryshyn
PhD in Technical Sciences, Associate Professor
National Academy of the Security Service of Ukraine, Kyiv, Ukraine
ORCID:0000-0003-2473-9507
ydobryshyn@gmail.com

AUTOMATION OF SOFTWARE RECOVERY DESIGN DECISIONS FOLLOWING CYBERATTACKS

Abstract. This paper addresses the current scientific and applied problem of automating software recovery processes after cyberattacks, which has become increasingly important due to the growing number and sophistication of cyber threats. An analysis of contemporary research indicates that the primary directions of development include the formalization of recovery processes using mathematical models, as well as the application of artificial intelligence and machine learning techniques. The software recovery process is represented as a set of interconnected tasks organized according to the principle of multilevel decomposition. Each task is described as a system of sets comprising input data, constraints, solution alternatives, evaluation criteria, models, and implementation procedures. Such a representation enables the formalization of the decision-making process, the identification of interdependencies among tasks, and their algorithmic implementation. The paper also presents an example of a mathematical model for selecting the optimal software recovery strategy following ransomware attacks. The proposed model takes into account the software system structure, the set of detected defects, their properties and criticality, as well as the available recovery methods considering time, resource, and risk constraints. An optimization objective function with weighting coefficients is proposed to determine the most effective recovery strategy according to predefined evaluation criteria. The obtained results demonstrate the feasibility of combining formalized approaches with intelligent methods to improve the efficiency of software recovery processes. The proposed approach provides a theoretical foundation for the development of intelligent decision support systems in the field of cybersecurity. Future research will focus on the development of adaptive and self-organizing models, their integration with monitoring and incident management systems, and the creation of domain-specific knowledge bases to improve diagnostic accuracy and software recovery effectiveness.

Keywords: cybersecurity; cyberattacks; software recovery; automation; formalization; mathematical model; decision-making; artificial intelligence.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Kozlovskyi, V. V., & Kopiika, O. O. (2020). Analysis of modern cyber threats and methods of information systems protection. *Information Protection, 22*(3), 45-52.
2. Lytvynenko, O. V., & Kovalenko, A. P. (2021). Cybersecurity of critical infrastructure: Current state and development prospects. *Information Processing Systems*, (2), 120-126.
3. Ilau, M.-C., Baldwin, A., Caulfield, T., & Pym, D. (2025). Modelling and simulating organizational ransomware recovery: Structure, methodology, and decisions. *Journal of Cybersecurity, 11*(1), Article tyaf035. https://doi.org/10.1093/cybsec/tyaf035
4. Zhao, M., & Chen, R. (2025). Automated cybersecurity incident response: A reinforcement learning approach. *AI and Data Science Journal, 2*(1), 23-28.
5. Leventopoulos, S., Pipyros, K., & Gritzalis, D. (2024). Retaliating against cyber-attacks: A decision-taking framework for policy-makers and enforcers of international and cybersecurity law. *International Cybersecurity Law Review, 5*, 237-262.
6. Bondarenko, O. I. (2020). Intelligent decision support systems in cybersecurity. *Information Technologies and Security*, (1), 15-22.
7. Zhang, Q., & Chen, M. (2020). AI-driven self-healing cloud systems. *IEEE Cloud Computing, 7*(3), 40-50.
8. Kenmogne, L. A., & Mocanu, S. (2024). Explainable AI for process-aware attack detection in industrial control systems. In *Proceedings of the 2024 IEEE 10th International Conference on Network Softwarization (NetSoft)* (pp. 363-368). IEEE. https://doi.org/10.1109/NETSOFT60951.2024.10588940

9. Kasali, K., Orekha, P., & Usoro, S. (2025). AI-driven strategies for mitigating cybersecurity threats in U.S. small and medium enterprises (SMEs). *International Journal of Computer Science and Information Technologies*, *16*(3), 110-116.
10. Petrenko, S. O. (2022). Formalization of software recovery method selection after cyberattacks. *Bulletin of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute". Series: Informatics, Control and Computer Engineering*, (75), 78-85.
11. Dolhova, N. H., Shapovalova, O. O., & Solodovnyk, H. V. (2025). Decision support system for cybersecurity design of critical infrastructure. *Cybersecurity: Education, Science, Technique, 1*(29), 348-368.
12. Dobryshyn, Yu. Ye., Sydorenko, S. M., & Vorokhob, M. V. (2023). Automated decision support system for recovering software damaged by cyberattacks. *Cybersecurity: Education, Science, Technique, 4*(20), 174-180.
13. Van der Kleij, R., Cadet, B., & Young, H. (2022). Developing decision support for cybersecurity threat and incident managers. *Computers & Security, 113*, Article 102535. https://doi.org/10.1016/j.cose.2021.102535
14. Lu, P., Zhang, L., Liu, M., et al. (2024). Recovery from adversarial attacks in cyber-physical systems: Shallow, deep and exploratory works. *ACM Computing Surveys*. https://doi.org/10.1145/3653974
15. Kott, A., Weisman, M. J., & Vandekerckhove, J. (2022). Mathematical modeling of cyber resilience. In *Proceedings of the MILCOM 2022 IEEE Military Communications Conference*. IEEE. https://doi.org/10.1109/MILCOM55135.2022.10017731
16. Sözer, H., Tekinerdogan, B., & Aksit, M. (2014). Optimizing decomposition of software architecture for local recovery. *Software Quality Journal*, *22*(4), 703-736.
17. Tonon, A., Zhang, M., Caglayan, B., Shen, F., Gui, T., Wang, M., & Zhou, R. (2025). *RADICE: Causal graph based root cause analysis for system performance diagnostic* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2501.11545
18. Li, J., Yang, G., & Shao, Y. (2023). Ransomware detection method based on TextGCN. In *Proceedings of the 6th International Conference on Artificial Intelligence and Big Data (ICAIBD)*. Chengdu, China.
19. Wei, R., Cai, L., Yu, A., & Meng, D. (2021). DeepHunter: A graph neural network based approach for robust cyber threat hunting. In *Security and Privacy in Communication Networks* (pp. 3-24). https://arxiv.org/abs/2104.09806
20. Rabzelj, M., Bohak, C., Južnič, L., Kos, A., & Sedlar, U. (2023). Cyberattack graph modeling for visual analytics. *IEEE Access, 11*, 1-34. https://doi.org/10.1109/ACCESS.2023.3304640

Отримано редакцією журналу / Received: 16.01.26
Прорецензовано / Revised: 02.02.26
Схвалено до друку / Accepted: 24.09.26