УДК 004.056.53:004.415.53 DOI: 10.31673/2409-7292.2026.039402

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

МЕТОД ГРУПУВАННЯ ДЕФЕКТІВ ПОШКОДЖЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ У ТЕХНОЛОГІЧНІ ПОДІБНІ ГРУПИ

У статті розглянуто актуальну науково-практичну проблему групування дефектів пошкодженого програмного забезпечення, що виникають унаслідок дії кібератак, у технологічно подібні групи. Обґрунтовано, що сучасні кібератаки спричиняють складні порушення працездатності програмного забезпечення, які характеризуються високим рівнем невизначеності, динамічністю прояву та різною критичністю. Визначено, що традиційні методи класифікації дефектів не забезпечують достатньої ефективності в умовах кіберінцидентів, оскільки не враховують змінність характеристик атак і можливість часткової належності дефектів до кількох груп одночасно. Проведено аналіз сучасних наукових досліджень у сфері виявлення, прогнозування та класифікації дефектів програмного забезпечення, а також методів кластеризації та нечіткої логіки, що застосовуються у задачах кібербезпеки. Запропоновано метод групування дефектів пошкодженого програмного забезпечення на основі використання методу «динамічних ядер», який забезпечує адаптивне формування груп дефектів у багатовимірному просторі ознак. У роботі описано математичну модель представлення дефектів, визначено систему вагових коефіцієнтів ознак та запропоновано механізм оцінювання ступеня належності дефектів до відповідних груп із використанням елементів теорії нечітких множин. Практичну апробацію методу виконано на прикладі групування дефектів програмного забезпечення після кібератаки за такими ознаками, як рівень пошкодження цілісності даних, системних збоїв та порушення системи безпеки. Результати дослідження підтвердили ефективність запропонованого підходу щодо автоматизованого формування груп критичних та незначних дефектів, а також можливість виявлення дефектів із проміжними характеристиками. Встановлено, що застосування методу «динамічних ядер» і нечіткого групування дозволяє підвищити ефективність процесів ідентифікації, аналізу, прогнозування та відновлення пошкодженого програмного забезпечення в умовах невизначеності інформації та змін характеристик кібератак.

Ключові слова: групування, метод, програмне забезпечення, дефект, математична модель, відновлення програмного забезпечення.

Вступ і формулювання проблеми

Аналізуючи поточний стан розвитку інформаційних технологій, необхідно зазначити, що на теперішній час спостерігається інтенсивне зростання складності програмного забезпечення та масштабів його використання у критично важливих інформаційних системах. Одночасно із цим суттєво зростає кількість кібератак, спрямованих на порушення працездатності програмних компонентів, модифікацію даних, несанкціонований доступ до інформаційних ресурсів та дестабілізацію функціонування інформаційних систем. Наслідком дії кібератак є виникнення значної кількості дефектів програмного забезпечення, які можуть мати різну природу, ступінь критичності та вплив на працездатність системи.

Особливістю дефектів пошкодженого програмного забезпечення після впливу кібератак є їх складна структура, невизначеність ознак та динамічний характер прояву. Дія сучасних дефектів характеризується порушенням цілісності, конфіденційності та доступності даних, пошкодженням компонентів програмного забезпечення та механізмів інформаційної безпеки. За таких умов традиційні методи класифікації та аналізу дефектів, які базуються на фіксованих правилах порівняння або жорстких класифікаційних ознаках, не забезпечують достатньої ефективності під час проведення діагностування та відновлення програмного забезпечення.

Крім того, процес аналізу дефектів ускладнюється значним обсягом даних, високим рівнем невизначеності та необхідністю оперативного прийняття рішень щодо пріоритетності відновлення програмного забезпечення після кібератак. Існуючі підходи щодо класифікації дефектів орієнтуються на технології розробки та тестування програмного забезпечення, тому не враховують особливості дії кібератак та порушень, що виникають у результаті кіберінцидентів. На теперішній час актуальним науково-практичним завданням є розробка

методів інтелектуального групування дефектів пошкодженого програмного забезпечення у технологічно подібні групи з урахуванням їх ознак, ступеня критичності та можливості часткової належності до декількох груп одночасно.

Одним із перспективних напрямів вирішення зазначеної проблеми є застосування методів кластеризації та нечіткої логіки, які дозволяють здійснювати адаптивне групування дефектів у багатовимірному просторі ознак. Особливий інтерес для задач кібербезпеки становить метод «динамічних ядер», який забезпечує ітераційне формування груп дефектів із можливістю адаптації правил порівняння залежно від поточного стану системи та характеристик кібератак. Використання такого підходу створює передумови для підвищення ефективності процесів ідентифікації, класифікації, аналізу та прогнозування дефектів пошкодженого програмного забезпечення.

Таким чином, наукова проблема полягає у відсутності ефективної методики групування дефектів пошкодженого програмного забезпечення після впливу кібератак, яка б забезпечувала адаптивну класифікацію дефектів в умовах невизначеності інформації, враховувала різну важливість ознак дефектів та дозволяла виконувати їх нечітке групування у технологічно подібні групи. Розв’язання зазначеної проблеми є важливим для підвищення ефективності функціонування систем аналізу кіберінцидентів, автоматизованого тестування та відновлення програмного забезпечення

Аналіз літературних джерел

Питанням виявлення та аналізу дефектів програмного забезпечення присвячено значну кількість сучасних наукових праць у галузі розробки, впровадження та експлуатації програмного забезпечення. Аналізуючи напрямки досліджень цих наукових робіт, можна зазначити, що більшість провідних фахівців орієнтується на розробку та впровадження різних математичних моделей, методів та технологій ідентифікації та класифікації дефектів програмного забезпечення, пропонують застосування спеціалізованих автоматизованих систем щодо розпізнавання, аналізу та відновлення дефектів, які були виявлені під розробки час експлуатації програмного забезпечення.

Так у наукової роботи [1] аналіз наукових досліджень, присвячених методам і технологіям виявлення відмов та вразливостей програмного забезпечення, які показали, що на теперішній час існуючі дослідження не забезпечують належну ідентифікацію та класифікацію дефектів програмного забезпечення. Тому автори наукової статті стверджують про необхідність розробки певних класифікаційних правил щодо технології ідентифікації та класифікації дефектів програмного забезпечення. Для вирішення цього питання автори статті пропонують правила, які дозволяють за певної методикою здійснювати класифікацію дефектів програмного забезпечення.

В іншій науковій роботі [2] автори стверджують, що ранні виявлення дефектів під час розробки програмного забезпечення, сприяють належній розробці працездатного програмного забезпечення, що задовольняє клієнта. Але аналіз відомостей щодо дефектів програмного забезпечення ускладнюється низкою проблем, які включають ознаки дублювання та кореляції. У дослідженні представляється та впроваджується на практиці певне програмне забезпечення, в якому дані щодо програмного забезпечення попередньо обробляються та оцінюються перед класифікацією дефектів. У науковій роботі також представлено окремі програмні елементи групування дефектів програмного забезпечення.

Ефективне прогнозування дефектів є вирішальним аспектом забезпечення якості програмного забезпечення, що дозволяє ідентифікувати дефектні модулі до фази тестування, стверджують автори наукової статті [3]. Автори статті пропонують комплексну структуру для прогнозування дефектів програмного забезпечення, яка включає п’ять етапів для вирішення проблем під час працездатності програмного забезпечення. Перший етап включає вибір версії наборів даних про дефекти. На другому етапі застосовується метод вибору ознак на основі алгоритму для визначення оптимальної підмножини ознак. На третьому етапі використовуються класифікатори, які за допомогою налаштування оптимізуються для

© Добришин Ю.Є. Метод групування дефектів пошкодженого програмного забезпечення у технологічні подібні групи. Сучасний захист інформації, 3(67), 76–84.
https://doi.org/10.31673/2409-7292.2026.039402

досягнення найвищого рівня точності. На четвертому етапі застосовується метод машинного навчання, як головний класифікатор, призначений щодо прийняття рішень. На заключному етапі оцінюються показники щодо точності прогнозування дефектів програмного забезпечення. Така методика вже передбачає виконання групування дефектів програмного забезпечення у технологічні подібні групи.

У рамках наукової роботи [4] запропоновано новий підхід для прогнозування дефектних модулів програмного забезпечення з метою досягнення високої точності. Для виявлення несправних модулів використовується методика групування дефектів за допомогою методів машинного навчання, включаючи дерево рішень, метод опорних векторів та наївний байєсівський метод. Науковими фахівцями здійснювалися експерименти, що використовували об'єднаний набір даних про дефекти програмного забезпечення, з метою гарантувати, що наведена авторами методика, працює ефективно на різноманітних наборах даних. Таким чином, за результатами дослідження запропонована методика групування дефектів продемонструвала достатню продуктивність щодо прогнозування дефектів програмного забезпечення.

Питання виявлення та прогнозування дефектів програмного забезпечення розглядається також у науковій роботі [5, 6]. Автори пропонують застосовувати певні методи машинного навчання для відбору найбільш важливих ознак дефектів із великого обсягу їх даних. Такий метод допомагає зосередитися лише на тих характеристиках, що найбільше впливають на результат. Завдяки цьому моделі, які запропоновані авторами, працюють ефективніше, оскільки зменшується кількість зайвої інформації та позитивно впливають на загальну точність під час відбору важливих ознак дефектів програмного забезпечення. Розроблені методики та практичні дослідження можуть бути застосовані для виконання робіт під час проведення дефектації пошкодженого програмного забезпечення.

Інтерес щодо технології класифікації та групування дефектів програмного забезпечення становить стаття наукового фахівця [7]. Автор стверджує, що для належного процесу запобігання дефектам, потрібно виділити поширені дефекти, а потім забезпечити їх усунення раніше за всіма іншими дефектами. У статті передбачається застосовувати елементи групування дефектів програмного забезпечення шляхом аналізу різних типів періодичних відмов програмних модулів, які мають однакову першопричину, яке буде відповідати загальному дефекту. Такий загальний дефект формується на підставі однакових груп дефектів, які мають тільки суттєві ознаки, що впливають на працездатність програмного забезпечення. Автори іншої наукової роботи [8] пропонують здійснювати групування різноманітних дефектів з метою перевірки коду та статичного аналізу програмного забезпечення. За рахунок цього їх дослідження дозволяють суттєво зменшити витрати під час проведення тестування програмного забезпечення. У статті зазначено, що така методика дозволила визначити 116 типів дефектів, які згрупували в 15 груп для створення класифікації дефектів. Крім того, 38% цих дефектів можна було точно виявити автоматично.

Питання класифікації та можливості групування дефектів програмного забезпечення останнім часом активно досліджуються з використанням компонентів штучного інтелекту. У роботі [9] системи штучного інтелекту використовуються як критично важливі програмні рішення, що робить виявлення та усунення дефектів програмного забезпечення важливим для забезпечення якості системи. У цьому дослідженні представлено спеціалізовану систему класифікації та групування дефектів програмного забезпечення, спеціально розроблену для вирішення складних проблем систем штучного інтелекту. Наведена система на базі компонентів штучного інтелекту забезпечує більш детальний та якісний підхід до класифікації та групування дефектів програмного забезпечення у середовищах штучного інтелекту. Автори багатьох сучасних наукових робіт [10], [11], [12] для класифікації та групування дефектів пошкодженого програмного забезпечення внаслідок дії кібератак, застосовують методи кластеризації, які дозволяють автоматично об’єднувати схожі дефекти у групи на основі їх характеристик. Дефекти пошкодженого програмного забезпечення розглядаються як об’єкт у

© Добришин Ю.Є. Метод групування дефектів пошкодженого програмного забезпечення у технологічні подібні групи. Сучасний захист інформації, 3(67), 76–84.
https://doi.org/10.31673/2409-7292.2026.039402

багатовимірному просторі ознак, а алгоритми кластеризації визначають ступінь подібності між ними, що дозволяє сформувати відповідні кластери.

Аналізуючи вказані роботи необхідно зазначити, що для задач кібербезпеки особливо ефективними є методи нечіткої кластеризації та метод «динамічних ядер», які забезпечують побудову структури груп виявлених дефектів та підтримують роботу в умовах невизначеності вихідних даних. Особливої уваги потребує дослідження щодо можливості застосування методу «динамічних ядер» для групування дефектів пошкодженого програмного забезпечення після впливу кібератак. Перевагою даного методу є можливість ітераційного формування кластерів у багатовимірному просторі ознак із урахуванням належності дефектів до декількох груп одночасно. Сучасні дослідження підтверджують ефективність динамічної кластеризації для задач аналізу кіберзагроз та прогнозування дефектів пошкодженого програмного забезпечення [13], [14].

Мета та завдання дослідження

Метою даної статті є розробка методики групування дефектів пошкодженого програмного забезпечення внаслідок дії кібератак у технологічно подібні групи на основі застосування методу «динамічних ядер», що забезпечить підвищення ефективності ідентифікації, класифікації, аналізу та прогнозування дефектів в умовах невизначеності інформації. Завданнями дослідження є перевірка ефективності запропонованої методики та оцінка можливості практичного її застосування у системах аналізу кіберінцидентів, тестування та відновлення програмного забезпечення.

Основна частина

Для математичного опису задачі групування дефектів пошкодженого програмного забезпечення внаслідок дії кібератак опишемо дефект програмного забезпечення сукупністю певної кількості ознак:

$$G_i = \{f_1, f_2, f_3, \dots \dots f_n^i\}, \quad (1)$$

де: f_n^i певне значення ознаки G_i дефекту.

Під час проведення діагностування, як правило, ознаки дефектів визначають суб'єктивно або статистичним шляхом, що є неприйнятним для подальшого визначення способу відновлення пошкодженого програмного забезпечення. Тому для усунення зазначеного недоліку, вирішення задачі вибору та групування номенклатури дефектів необхідно розглядати як одну з задач теорії розпізнавання образів, шляхом здійснення класифікації дефектів у - вимірному просторі та виконання низку підзадач:

- визначення кількості груп дефектів GR_n ;
- визначення простору ознак дефектів F_n;
- визначення вагових параметрів ознак дефектів ;
- визначення якості групування дефектів T_g;
- визначення складу груп дефектів GD_i

Вказані задачі можуть бути вирішені за допомогою відомого методу «динамічних ядер», який активно застосовується у машинному навчанні, та застосовує алгоритм порівняння об'єктів у залежності поточної ситуації, де міра схожості між об'єктами не є фіксованою, а змінюється залежно від даних або стану системи. Це важливо для задач класифікації та групування, тобто потрібно зрозуміти наскільки два об'єкти схожі між собою, наприклад: два модулі програмного забезпечення; два файли після кібератаки; два набори логів.

Звичайні методи, які використовують фіксоване ядро та визначають, що якщо об'єкти класифікації близькі за ознаками, то вони схожі, якщо їх ознаки відрізняються між собою, то вони різні. Але у питаннях класифікації та групування дефектів програмного забезпечення внаслідок дії кібератак фіксована формула щодо групування перестає добре працювати тому що дані про дефекти змінюються з часом, атаки бувають різні, поведінка інформаційних систем та програмних модулів змінюється. Тому головна ідея методу «динамічних ядер» передбачає змінювати "правила порівняння" залежно від ситуації. Найбільш зручним із

© Добришин Ю.Є. Метод групування дефектів пошкодженого програмного забезпечення у технологічні подібні групи. Сучасний захист інформації, 3(67), 76–84.
https://doi.org/10.31673/2409-7292.2026.039402

технологічної точки зору є представлення ядра у вигляді поняття комплексного дефекту, де для його визначення, необхідно зазначити центр ваги кожного класу дефектів. Для виконання групування вважається, що всі дефекти будуть мати однакову значущість, тобто масу. У разі якщо до складу ядра входить більше одного дефекту, то характеристики комплексного дефекту будуть визначатися за таким правилом:

Припустимо, що $\{S_g\}$ м – множина дефектів характеризується певними основними класифікаційними ознаками, а саме: (d_i– пошкодження цілісності даних, Pr_f – системні збої у роботі програмного забезпечення, b_f – збій системи безпеки), то ознаки комплексного дефекту $\{G_k\}$ визначаються як середнє значення параметрів дефектів, що входять до ядра.

$$\{G_k\} \to \{d_i, Pr_f, b_f\}. \tag{2}$$

У загальному вигляді алгоритм групування за допомогою «динамічних ядер» зводиться до вибору початкового набору ознак класифікації, тобто $\{k\}$ вихідних ядер. Потім здійснюється групування шляхом приєднання решти об'єктів до найближчого ядра. У отриманих групах обчислюють ядра та порівнюють новий поділ ядер з попереднім поділом. Якщо поділ не змінився, процес групування припиняється.

Для роботи алгоритму «динамічних ядер» необхідно визначати ступінь схожості між дефектами програмного забезпечення на підставі раніше визначеного простору ознак (1). Оскільки ознаки дефектів можуть мати різну важливість під час процедури групування дефектів, слід врахувати вагові коефіцієнти, що визначають значимість ознак дефектів пошкодженого програмного забезпечення:

$$V = \{V_1, V_2, \dots \dots V_n,\}, \tag{3}$$

де V_n визначає значущість відповідної ознаки.

Тоді відстань між двома дефектами G_iта G_jвизначається наступним виразом:

$$D\left(G_i, G_j\right) = \sqrt{\sum_{i=1}^{n} V_i \left(C_{il} - C_{jl}\right)^2}\ , \tag{4}$$

де C_{il}, C_{jl} – значення l-ї ознаки для дефектів G_iта G_j, V_l – ваговий коефіцієнт ознаки, n – кількість ознак.

Вагові коефіцієнти ознак можуть визначатися експертним шляхом або далі уточнятися під час роботи системи на основі накопичених статистичних даних. Це дозволяє адаптувати процес групування до змін характеристик кібератак та особливостей пошкодженого програмного забезпечення. Після обчислення відстаней між дефектами виконується їх групування навколо найближчих ядер кластерів. На кожній ітерації центри груп уточняються до моменту стабілізації результатів класифікації.

Складність процесу групування дефектів пошкодженого програмного забезпечення після впливу кібератак обумовлена невизначеністю ознак та різними факторами класифікації. У зв'язку з цим визначення оптимальних вагових коефіцієнтів ознак доцільно виконувати ітераційним шляхом, тобто шляхом послідовного уточнення параметрів групування.

Для зменшення кількості загальносистемних перерахунків можна використовувати принцип нечіткого групування, заснований на елементах теорії нечітких множин. Такий підхід дозволяє одному дефекту одночасно належати до кількох груп із різним ступенем приналежності.

Припустимо, що $\{PL\}$ - множина всіх дефектів, а $A \subset PL$ - окрема група дефектів. Тоді ступінь належності дефекту G_i до групи A визначається функцією приналежності:

$$f_A(G_i) \in [0,1], \tag{5}$$

де $f_A(G_i) = 0$ – дефект не належить групі; $f_A(G_i) = 1$ – дефект повністю належить групі; IM_g – проміжні значення характеризують часткову належність дефекту до групи.

© Добришин Ю.Є. Метод групування дефектів пошкодженого програмного забезпечення у технологічні подібні групи. Сучасний захист інформації, 3(67), 76–84.
https://doi.org/10.31673/2409-7292.2026.039402

Тоді нечітка група дефектів може бути представлена у вигляді:

$$A = \{(G_1, f_1), (G_2, f_2), \dots, (G_n, f_n)\}\,, \tag{6}$$

де G_n – дефекти програмного забезпечення; f_n – ступені їх приналежності до відповідної групи.

Ступінь приналежності дефекту визначається на основі відстані між дефектом та центром групи:

$$f_i(G_i) = \frac{G_i}{G_{max}}\,, f_i \in [0,1]\,. \tag{7}$$

Припустимо, що після здійснення кібератаки на інформаційну систему підприємства було виявлено п’ять дефектів програмного забезпечення, які розміщені у окремій групі:

$$PL = \{G_1, G_2, G_3, G_4, G_5\}\,. \tag{8}$$

Як було визначено раніше кожен дефект характеризується трьома основними ознаками:

d_i - рівень пошкодження цілісності даних;

Pr_f - рівень системних збоїв програмного забезпечення;

b_f - рівень порушення системи безпеки.

Для спрощення проведення розрахунків, усі ознаки будемо оцінювати за шкалою від 0 до 10. Тоді множина ознак дефектів відповідно до виразу (2) набуде наступного вигляду:

$$G_i = \{d_i, Pr_f, b_f\}\,. \tag{9}$$

За результатами діагностування пошкодженого програмного забезпечення були отримані такі значення ознак дефектів, враховуючи шкалу їх оцінки від 0 до 10 (табл. 1).

Таблиця 1

Показники оцінки ознак дефектів пошкодженого програмного забезпечення

Позначення дефекту	d_i – рівень пошкодження цілісності даних	Pr_f – рівень системних збоїв програмного забезпечення	b_f – рівень порушення системи безпеки
G_1	9	8	9
G_2	8	7	8
G_3	2	3	2
G_4	3	2	3
G_5	6	7	5

Аналіз показників таблиці показує, що:

- дефекти G_1та G_2мають ознаки серйозного пошкодження;
- G_3та G_4характеризуються незначними порушеннями;
- G_5займає проміжне положення.

Для подальшого розрахунку експертним шляхом відповідно до виразу (3) встановимо такі вагові коефіцієнти:

$$V = \{0.5; 0.3; 0.2\}\,, \tag{10}$$

де 0.5 – важливість пошкодження даних; 0.3 – важливість системних збоїв; 0.2 – важливість порушень безпеки. Це означає, що найбільш критичною ознакою є пошкодження цілісності даних.

Здійснимо формування початкових ядер кластерів, для чого на початковому етапі обираємо два ядра груп:

$$K_1 = G_1 = (9,8,9)\,;\; K_2 = G_3 = (2,3,2)\,, \tag{11}$$

де перше ядро відповідає критичним дефектам; друге ядро – незначним дефектам.

© Добришин Ю.Є. Метод групування дефектів пошкодженого програмного забезпечення у технологічні подібні групи. Сучасний захист інформації, 3(67), 76–84.
https://doi.org/10.31673/2409-7292.2026.039402

Використовуємо формулу (4), виконаємо обчислення відстаней між дефектами. За результатами отримуємо:

-відстань між та ядром :

$$D(G_2, K_1) = \sqrt{0.5(8-9)^2 + 0.3(7-8)^2 + 0.2(8-9)^2} = \sqrt{0.5 + 0.3 + 0.2} = \sqrt{1} = 1 \ ;$$

-відстань між та ядром

$$D(G_2, K_2) = \sqrt{0.5(8-2)^2 + 0.3(7-3)^2 + 0.2(8-2)^2} = \sqrt{18 + 4.8 + 7.2} = \sqrt{30} \approx 5.48 \ .$$

Оскільки: $1 < 5.48$ дефект G_2належить до груп ядра K_1.

Аналогічно здійснимо вказані розрахунки для інших ознак дефектів пошкодженого програмного забезпечення. Після розрахунків отримуємо наступні дані (табл. 2).

Таблиця 2

Результати обчислення відстаней між дефектами пошкодженого програмного забезпечення

Позначення дефекту	Відстань до K_1	Відстань до K_2	Група ядер
G_1	0	6.04	K_1
G_2	1.00	5.48	K_1
G_3	6.04	0	K_2
G_4	5.48	1.00	K_2
G_5	2.86	4.06	K_1

Після першої ітерації отримано:

- групу критичних дефектів:

$$GD_1 = \{G_1, G_2, G_5\} \ ; \tag{12}$$

- групу незначних дефектів:

$$GD_2 = \{G_3, G_4\} \ . \tag{13}$$

Для більш детального групування здійснимо перерахунок центрів груп, для чого обчислюємо середні значення ознак для нового ядра групи GD_1:

$$K_1' = \left(\frac{9+8+6}{3}, \frac{8+7+7}{3}, \frac{9+8+5}{3}\right), \quad K_1' = (7.67, 7.33, 7.33) \ . \tag{14}$$

Для нового ядра групи середні значення будуть дорівнювати:

$$K_2' = \left(\frac{2+3}{2}, \frac{3+2}{2}, \frac{2+3}{2}\right), \quad K_2' = (2.5, 2.5, 2.5) \ . \tag{15}$$

Розглянемо дефект G_5, який має проміжні характеристики. Визначимо ступінь його приналежності до групи критичних дефектів. Відповідно до виразу (7):

$$f(G_5) = \frac{2.86}{6} = 0.48.$$

Отже:

$$f(G_5) = 0.48 \ . \tag{16}$$

Це означає, що дефект має часткову належність до критичної групи та може бути додатково проаналізований експертами.

Висновки і рекомендації

Таким чином, практичне використання методу «динамічних ядер» та нечіткого ітераційного групування дозволяє реалізувати та автоматизувати адаптивну систему класифікації дефектів програмного забезпечення та застосовувати вказаний метод на будь-якій стадії як проєктування, так і відновлення програмного забезпечення. Використання неповних та нечітких даних для ітераційного уточнення результатів діагностування пошкодженого програмного забезпечення створює умови для проведення подальших

© Добришин Ю.Є. Метод групування дефектів пошкодженого програмного забезпечення у технологічні подібні групи. Сучасний захист інформації, 3(67), 76–84.
https://doi.org/10.31673/2409-7292.2026.039402

досліджень з визначенням критичності дефектів та пріоритету відновлення пошкодженого програмного забезпечення, а також адаптувати процес групування до змін характеристик кібератак.

Перелік посилань

1. Медзатий, Д., Войчур, Ю., & Войчур, О. (2023). Технологія ідентифікації та класифікації відмов і вразливостей програмного забезпечення. Вимірювальна та обчислювальна техніка в технологічних процесах, 1, 53–57. https://doi.org/10.31891/2219-9365-2023-73-1-8.

2. Bhaskaran, N. A., & Durairaj, M. (2023). Highlighting bugs in software development codes using SDPET for enhancing security. Measurement: Sensors, 30, 100930. https://doi.org/10.1016/j.measen.2023.100930.

3. Ali, M., Mazhar, T., Al-Rasheed, A., Shahzad, T., Yasin Ghadi, Y., & Amir Khan, M. (2024). Enhancing software defect prediction: A framework with improved feature selection and ensemble machine learning. PeerJ Computer Science, 10, e1860. https://doi.org/10.7717/peerj-cs.1860.

4. Abbas, S., Aftab, S., Khan, M. A., Ghazal, T. M., Hamadi, H. A., et al. (2023). Data and ensemble machine learning fusion based intelligent software defect prediction system. Computers, Materials & Continua, 75(3), 6083–6100. https://doi.org/10.32604/cmc.2023.037933.

5. Mumtaz, B., Kanwal, S., Alamri, S., & Khan, F. (2021). Feature selection using artificial immune network: An approach for software defect prediction. Intelligent Automation & Soft Computing. https://doi.org/10.32604/iasc.2021.018405.

6. Alazba, A., & Aljamaan, H. (2022). Software defect prediction using stacking generalization of optimized tree-based ensembles. Applied Sciences, 12(9), 4577. https://doi.org/10.3390/app12094577.

7. Elentukh, A. (2023). People make mistakes – A survey of common causes of software defects. In Computer Science and Education in Computer Science (pp. 117–133). https://doi.org/10.1007/978-3-031-44668-9_9.

8. Gunawardena, S., Tempero, E., & Blincoe, K. (2023). Concerns identified in code review: A fine-grained, faceted classification. Information and Software Technology, 153, 107054. https://doi.org/10.1016/j.infsof.2022.107054.

9. Alannsary, M. O. (2025). A defect classification framework for AI-based software systems (AI-ODC). arXiv. https://doi.org/10.48550/arXiv.2508.17900.

10. Zhao, F., Yang, Y., Liu, H., & Wang, C. (2024). A robust multi-view knowledge transfer-based rough fuzzy C-means clustering algorithm. Complex & Intelligent Systems, 10, 5331–5358. https://doi.org/10.1007/s40747-024-01431.

11. Oskoueie, A. G., Samadi, N., Khezri, S., Najafi Moghaddam, A., Babaei, H., Hamini, K., Nojavan, S. F., Bouyer, A., & Arasteh, B. (2025). Feature-weighted fuzzy clustering methods: An experimental review. Neurocomputing, 619, 129176. https://doi.org/10.1016/j.neucom.2024.129176.

12. Yin, H., Aryani, A., Petrie, S., Nambissan, A., Astudillo, A., & Cao, S. (2024). A rapid review of clustering algorithm. arXiv. https://doi.org/10.48550/arXiv.2401.07389.

13. Black, P., Gondal, I., Bagirov, A., & Moniruzzaman, M. (2021). Malware variant identification using incremental clustering. Electronics, 10(14), 1628. https://doi.org/10.3390/electronics10141628.

14. Jurečková, O., Jureček, M., Stamp, M., Di Troia, F., & Lórenc, R. (2024). Classification and online clustering of zero-day malware. Journal of Computer Virology and Hacking Techniques, 20, 579–592. https://doi.org/10.1007/s11416-024-00513-5.

Yurii Dobryshyn
Candidate of Technical Sciences, Associate Professor, Associate Professor of the Department of Cybersecurity
National Academy of the Security Service of Ukraine, Kyiv, Ukraine
ORCID: 0000-0003-2473-9507
E-mail: ydobryshyn@gmail.com

METHOD OF GROUPING DEFECTS OF DAMAGED SOFTWARE INTO TECHNOLOGICALLY SIMILAR GROUPS

The article considers the current scientific and practical problem of grouping defects of damaged software, which arise as a result of cyberattacks, into technologically similar groups. It is substantiated that modern cyberattacks cause complex software malfunctions, which are characterized by a high level of uncertainty, dynamic manifestation and different criticality. It is determined that traditional methods of classifying defects do not provide sufficient efficiency in the conditions of cyberincidents, since they do not take into account the variability of attack characteristics and the possibility of partial belonging of defects to several groups at the same time. An analysis of modern scientific research in the field of detecting, predicting and classifying software defects, as well as clustering and fuzzy logic methods used in cybersecurity tasks, is carried out. A method of grouping defects of damaged software is proposed based on the use of the "dynamic kernels" method, which provides adaptive formation of groups of defects in a multidimensional feature space. The paper describes a mathematical model of defect representation, defines a system of feature weighting coefficients,

© Добришин Ю.Є. Метод групування дефектів пошкодженого програмного забезпечення у технологічні подібні групи. Сучасний захист інформації, 3(67), 76–84.
https://doi.org/10.31673/2409-7292.2026.039402

and proposes a mechanism for assessing the degree of defect membership in the corresponding groups using elements of fuzzy set theory. The practical testing of the method was performed on the example of grouping software defects after a cyberattack by such features as the level of data integrity damage, system failures, and security system violations. The results of the study confirmed the effectiveness of the proposed approach for the automated formation of groups of critical and minor defects, as well as the possibility of detecting defects with intermediate characteristics. It was established that the use of the "dynamic kernels" method and fuzzy grouping allows to increase the efficiency of the processes of identification, analysis, prediction, and recovery of damaged software in conditions of information uncertainty and changes in the characteristics of cyberattacks.

Keywords: grouping, method, software, defect, mathematical model, software recovery.

Надійшла до редакції (Received): 25.07.2026
Прийнята до друку (Accepted): 08.09.2026
Опубліковано онлайн (Available online): 15.09.2026

УДК 004.056:004.415.53:004.89 DOI: 10.31673/2409-7292.2026.035907

Максимович Максим Віталійович
аспірант кафедри безпеки інформаційних технологій
Національний університет «Львівська політехніка», Львів, Україна
ORCID: 0009-0008-3604-8424
E-mail: maksym.v.maksymovych@lpnu.ua

МЕТОД КОНТЕКСТНО-КЕРОВАНОГО ДИНАМІЧНОГО ТЕСТУВАННЯ БЕЗПЕКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ АГЕНТІВ НА ОСНОВІ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Розвиток методів штучного інтелекту, зокрема великих мовних моделей та побудованих на їх основі агентних систем, відкриває нові можливості для вдосконалення методів автоматизованого тестування безпеки програмного забезпечення (ПЗ). Серед методів динамічне тестування безпеки (DAST) є найперспективнішим напрямом для вдосконалення, оскільки воно здійснює наскрізну перевірку реального працюючого застосунку та відображає його фактичну експлуатовану поведінку. Водночас класичні DAST-інструменти формують сценарії атаки шляхом випадкового перебору, що зумовлює неповне покриття та є доволі ресурсозатратним. Метою роботи є розроблення методу контекстно-керованого динамічного тестування безпеки ПЗ на основі агентів, побудованих на великих мовних моделях, що дозволяє підвищити повноту виявлення вразливостей та знизити ресурсозатратність сканування. Запропонований метод використовує специфікацію OpenAPI як джерело семантичного контексту, на основі якого агенти генерують набір наскрізних (E2E) тестів безпеки, що виступають детермінованим джерелом трафіку та скеровують динамічне сканування реальними бізнес-процесами замість випадкового обходу. Метод забезпечує два покращення: розширення покриття на вразливості, недосяжні для сигнатурних сканерів, та скорочення обсягу надлишкових запитів завдяки перевірці лише згенерованого трафіку. Описано модель покриття вразливостей, робочий процес та стратегію формування тестових сценаріїв, а також визначено метрики оцінювання ефективності методу. Отримані результати свідчать про доцільність агентного підходу для розширення покриття та підвищення ресурсоефективності динамічного тестування і окреслюють напрям подальшого експериментального дослідження.

Ключові слова: динамічне тестування безпеки, великі мовні моделі, агентні системи, OpenAPI, BOLA, IDOR, автоматизація тестування.

Вступ

У сучасних умовах стрімкої цифровізації та зростання кількості кібератак питання захисту програмного забезпечення (ПЗ) набуває особливої актуальності [1]. Збільшення складності програмних систем, поширення веб застосунків і програмних інтерфейсів (API), а також скорочення циклів розробки зумовлюють потребу в автоматизованих засобах виявлення вразливостей, здатних інтегруватися безпосередньо у процес неперервної інтеграції та розгортання (CI/CD). Паралельно стрімкий розвиток методів штучного інтелекту, зокрема великих мовних моделей (LLM) та побудованих на їх основі агентних систем, відкриває нові

© Добришин Ю.Є. Метод групування дефектів пошкодженого програмного забезпечення у технологічні подібні групи. Сучасний захист інформації, 3(67), 76–84.
https://doi.org/10.31673/2409-7292.2026.039402