DOI: 10.58254/viti.5.2024.16.181

УДК 004.056.57

д-р філософії Фесьоха В. В. ORCID: 0000-0001-6612-1970 (ВІТІ ім. Героїв Крут)
Кисиленко Д. Ю. ORCID: 0000-0001-5491-6231 (ВІТІ ім. Героїв Крут)
д-р філософії Фесьоха Н. О. ORCID: 0000-0002-9797-5589 (ВІТІ ім. Героїв Крут)

ОБҐРУНТУВАННЯ ВИБОРУ ПІДХОДУ ДО ВИЗНАЧЕННЯ ІНВАРІАНТНОЇ КОМПОНЕНТИ У ПОВЕДІНЦІ ПОЛІМОРФНОГО (МЕТАМОРФНОГО) ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ЗНИЖЕННЯ РОЗМІРНОСТІ ПРОСТОРУ ОЗНАК

Еволюція сценаріїв застосування шкідливого програмного забезпечення зумовлює потребу в розробці дієвих стратегій нейтралізації їхнього деструктивного впливу. Одним із найбільш загрозливих типів шкідливого програмного забезпечення є поліморфні (метаморфні) віруси, оскільки значною мірою спроможні уникати виявлення системами виявлення вторгнень, управління інформаційною безпекою (подіями безпеки), антивірусними програмами та системами проактивного виявлення нетипових загроз і цільових атак на кінцевих точках завдяки властивості змінювати власну сигнатуру. До того ж, протягом останнього часу зафіксовано стрімке збільшенням кіберінцидентів, пов'язаних із застосуванням поліморфного (метаморфного) шкідливого програмного забезпечення. Основна причина цього зросту – доступність технологій штучного інтелекту, які дозволяють зловмисникам досить швидко та ефективно модифікувати код вже класифікованих шкідливих програм, не потребуючи значних спеціалізованих технічних компетенцій.

Проведено порівняльний аналіз існуючих підходів до виявлення поліморфного, олігоморфного і метаморфного шкідливого програмного забезпечення. Виявлено, що ні одна група методів не використовує на свою користь ключову особливість поліморфного (метаморфного) шкідливого програмного забезпечення – інваріантну поведінку за певною підмножиною ознак, яка характеризує один і той самий вектор деструктивного впливу шкідливого програмного забезпечення.

З метою нівелювання властивості модифікації власного коду поліморфним (метаморфним) шкідливим програмним забезпеченням запропоновано підхід до визначення його інваріантної компоненти під час поведінкового аналізу на основі поєднання переваг поведінкового аналізу та техніки машинного навчання – зменшення розмірності досліджуваного простору ознак. Такий підхід потенційно дозволить визначати інваріантну поведінку шкідливого програмного забезпечення у вигляді підмножини досліджуваних ознак для кожного відомого його типу, що у свою чергу формує підґрунтя для реалізації нового підходу до ефективного виявлення модифікованого (удосконаленого) шкідливого програмного забезпечення.

Ключові слова: *кібербезпека, шкідливе програмне забезпечення, поліморфні (метаморфні) віруси, поведінковий аналіз, простір ознак, штучний інтелект, машинне навчання, зменшення розмірності даних, генетичні алгоритми.*

V. Fesokha, D. Kysylenko, N. Fesokha. Justification of the choice of the approach to the determination of the invariant component in the behavior of polymorphic (metamorphic) malware on the basis of reducing the dimensionality of the sign space

The evolution of malware use scenarios necessitates the development of effective strategies to neutralise their destructive impact. One of the most threatening types of malware is polymorphic (metamorphic) viruses, as they are largely able to evade detection by intrusion detection systems, information security management (security events), antivirus software and systems for proactive detection of atypical threats and targeted attacks on endpoints due to their ability to change their own signature. In addition, there has been a rapid increase in recent cyber incidents involving the use of polymorphic (metamorphic) malware. The main reason for this growth is the availability of artificial intelligence technologies that allow attackers to modify the code of already classified malware quickly and efficiently, without requiring significant specialised technical competence.

A comparative analysis of existing approaches to detecting polymorphic, oligomorphic and metamorphic malware is carried out. It is found that no group of methods uses to its advantage the key feature of polymorphic (metamorphic) malware – invariant behaviour by a certain subset of features that characterise the same vector of destructive impact of malware.

With a view to neutralising the property of modification of its own code by polymorphic (metamorphic) malware, the article proposes an approach to determining its invariant component during behavioural analysis based on a combination of the advantages of behavioural analysis and machine learning techniques – reducing the dimensionality of the studied feature space. Such an approach will potentially allow determining the invariant behaviour of malware as

a subset of the studied features for each known type of malware, which in turn forms the basis for implementing a new approach to the effective detection of modified (advanced) malware.

Keywords: *cybersecurity, malware, polymorphic (metamorphic) viruses, dynamic analysis, feature space, artificial intelligence, machine learning, data dimensionality reduction, genetic algorithms.*

Актуальність та постановка завдання в загальному вигляді. Серед численних завдань забезпечення безпеки в кібернетичному просторі, розробка ефективних методів виявлення і нейтралізації шкідливого програмного забезпечення (ШПЗ) відіграє критично важливу роль. Згідно зі статистичними даними, наведеними у [1–3], зафіксовано збільшення кількості кіберінцидентів, пов'язаних із застосуванням ШПЗ на 123 % у 2023 році порівняно з попередніми роками. Такий стрімкий зріст застосування ШПЗ пояснюється відносно простим і досить ефективним підходом до здійснення прихованого інформаційно-руйнівного впливу, обумовленого його поліморфізмом (мутацією).

Ключову роль у процесі створення поліморфного, олігоморфного і метаморфного ШПЗ відіграють все більш доступні та широко впроваджені системи і технології штучного інтелекту (ШІ), що дозволяє зловмисникам досить швидко та ефективно модифікувати код та/або структуру вже відомих шкідливих програм, не потребуючи при цьому значних спеціалізованих технічних компетенцій [4].

Дослідження канадської телекомунікаційної компанії BlackBerry показують, що 48 % зловмисників використовують ChatGPT для створення нових зразків ШПЗ, тоді як 46 % – модифікують вже існуюче ШПЗ [5]. Так, на початку 2023 року зафіксовано факти використання зловмисниками системи генеративного ШІ, зокрема мовної моделі ChatGPT для створення адаптивного ШПЗ та інструментів шифрування.

Як зазначає компанія Check Point – розробник програмного і апаратного забезпечення для кіберзахисту, – російські кіберзлочинці активно тестують різноманітні способи обходу обмежень OpenAI – лабораторії досліджень технологій ШІ, щоб використовувати ChatGPT у незаконних цілях [6]. Одним із підрозділів компанії Check Point зафіксовано численні випадки маніпулювання прикладним програмним інтерфейсом ChatGPT з метою генерації шкідливих кодів і створення фішингових електронних листів [7].

Наступним зафіксованим випадком використання ChatGPT у зловмисних цілях є модифікація ШПЗ, відомого як Infostealer [8], суть застосування якого зводиться до пошуку поширених типів файлів в операційній системі об'єкта кібератаки з подальшим їх архівуванням та відправкою на віддалений FTP-сервер.

У звітах компанії CyberArk [9], діяльність якої тісно пов'язана з інформаційною безпекою, представлено результати експерименту щодо генерації поліморфного (метаморфного) ШПЗ засобами ChatGPT на основі коду відомих шкідливих програм, що дозволило отримати кілька варіантів одного інформаційно-руйнівного впливу без демонстрації підозрілої активності.

На початку 2023 року ІТ-спеціалісти компанії HYAS, яка спеціалізується на протидії кіберзагрозам, представили докази на основі концепції PoC (Proof of Concept – демонстрація практичної реалізації) щодо використання технологій-ШІ у сфері кібербезпеки. Перший PoC-експлойт, BlackMamba [10], використовує технології ШІ для синтезу функцій поліморфного кейлоггера – програмного забезпечення (ПЗ) реєстрації дій користувача, який здатний динамічно модифікувати безпечний код під час виконання, без залучення командно-контрольованої інфраструктури.

Розроблений компанією HYAS PoC-експлойт – EyeSpy [11] являє собою новий тип поліморфного (метаморфного) ШПЗ, який використовує технології ШІ для прийняття обґрунтованих рішень та розширення власних можливостей. Ця шкідлива програма постійно модифікує власний код з метою уникнення виявлення системами захисту, а також використовує стратегії, які постійно розвиваються для ускладнення процесу виявлення.

Очевидно, головна мета неперервного вдосконалення шкідливих програм – заподіяння інформаційно-руйнівного впливу об'єкту атаки за умови запобігання ідентифікації системами виявлення вторгнень (кібератак) – IDS/IPS (Intrusion Detection and Prevention System), системами управління інформаційною безпекою (подіями безпеки) – SIEM (Security Information and Event Management), антивірусними програмами та системами проактивного виявлення нетипових загроз і цільових кібератак на кінцевих точках – EDR (Endpoint Detection and Response). Так, хоч зазначені системи постійно вдосконалюються, проте ефективність застосування поліморфного і метаморфного ШПЗ залишається високою протягом багатьох років [12]. Успіх їхнього застосування обумовлено, насамперед, тим, що існуючі методики і стратегії удосконалення систем кіберзахисту ґрунтуються на дослідженні інцидентів безпеки постфактум (після їх виникнення). По-друге, використання інструментів статичного аналізу ШПЗ часто не є ефективним, внаслідок використання прийомів обфускації та шифрування коду ШПЗ. По-третє, існуючі підходи до динамічного аналізу ШПЗ часто зосереджені на досліджені обмеженого кола дій (спроб доступу до певних файлів або системних ресурсів) [13].

Враховуючи темпи розвитку сучасних інформаційних технологій, слід зазначити, що використання систем і технологій ШІ значно прискорює еволюцію поліморфного (метаморфного) ШПЗ, роблячи процес його створення автоматизованим, значно швидшим та доступнішим, що у свою чергу дозволяє ефективно здійснювати приховану деструктивну діяльність протягом тривалого часу.

Таким чином, ефективне виявлення і нейтралізація поліморфного (олігоморфного) і метаморфного ШПЗ вимагають постійного аналізу та адаптації до нових сценаріїв їх застосування, що у свою чергу зумовлює необхідність розробки інноваційних підходів за цією тематикою.

Аналіз попередніх досліджень. Науковим дослідженням щодо виявлення ШПЗ, яке постійно самомодифікується (еволюціонує) та/або використовує різноманітні методи обходу існуючих систем захисту, присвячено значну кількість робіт [14–25], аналіз найперспективніших із яких викладено нижче.

У дослідженнях [14–15] розглядаються перспективи використання ***статичного аналізу*** для вивчення програмного коду, усіх можливих його станів та дефектів без фактичного виконання. Основна перевага зазначеного підходу полягає в здатності досліджувати всі можливі шляхи виконання коду та значень змінних, що в подальшому сприяє виявленню аномальної поведінки. Враховуючи те, що статичний аналіз базується на раніше набутому досвіді, він є ефективним інструментом для виявлення відомого ШПЗ, однак має суттєві обмеження у виявленні нових, поліморфних (метаморфних) зразків ШПЗ.

У роботах [14–15] використовується ***динамічний аналіз*** для вивчення ШПЗ безпосереднього під час його фактичного виконання в ізольованому середовищі (пісочниці). Цей підхід формує поведінкові шаблони ПЗ у реальному часі, які в подальшому можуть бути використані для виявлення нових зразків ШПЗ, але з певними обмеженнями у часі. До того ж, такий підхід неспроможний надавати латентні (приховані/неочевидні) шаблони поведінки ШПЗ, що у свою чергу не дозволяє глибоко зрозуміти моніторингові дані та обґрунтовувати відповідні прийняті рішення з належним рівнем впевненості.

У дослідженнях [16–17] описується використання ***евристичного аналізу*** для виявлення ШПЗ. Такий підхід ґрунтується на застосуванні набору правил (евристик) для ідентифікації підозрілої поведінки програм, що може вказувати на потенційну присутність ШПЗ у системі. Це дозволяє виявляти як відоме, так і певною мірою поліморфне, олігоморфне і метаморфне ШПЗ. Проте цей підхід має досить велику кількість хибних спрацювань.

У роботах [15; 18] використовується метод ***аналізу поведінки*** для виявлення ШПЗ, що передбачає постійний моніторинг активності програм та системи з подальшим порівнянням з

еталонними зразками поведінки. Будь-яке відхилення від типової поведінки свідчить про можливе функціонування ШПЗ. Хоча аналіз поведінки ефективний у виявленні відомих зразків ШПЗ, проте він менш ефективний у виявленні поліморфного (метаморфного) ШПЗ. Однак характеризується відсутністю обмежень, пов’язаних із необхідністю попереднього знання про конкретні віруси або їхні сигнатури.

У роботах [19; 20] описується застосування методів ***машинного навчання*** для виявлення ШПЗ. Ці методи здатні розпізнавати закономірності (шаблони), що властиві ШПЗ у великих наборах даних з великою ефективністю. Поряд з цим, їхня ефективність значно залежить від тренувального набору даних (навчальної вибірки), для якого часто не піднімається питання актуальності і дисбалансу класів.

У дослідженнях [21–24] запропоновано використання ***декларативного підходу*** для виявлення ШПЗ. Так, передбачається порівняння поведінки відомого ШПЗ та нового ПЗ на предмет виявлення збіжностей і повторюваних шаблонів у трасах системних викликів, які властиві шкідливим програмам. Реалізація такого підходу забезпечується засобами Declare – мовою моделювання процесів як множини обмежень у системі і послідовності дій. Основними недоліками цього підходу є складність інженерії коректних трас системних викликів, а також необхідність реєстрації трас системних викликів для кожної операційної платформи окремо.

У роботі [25] зазначаються дві стратегії виявлення поліморфного (метаморфного) ШПЗ: виявлення аномальної діяльності на основі наявних шаблонів функціонування системи, яка підлягає захисту, та виявлення за ознаками відомого ШПЗ. Однак з огляду на те, що більшу кількість поліморфного (метаморфного) ШПЗ складають модифіковані зразки вже існуючого шкідливого коду, доцільним є використання останньої стратегії.

Підсумовуючи вищезазначене, можна зробити висновок, що кожен з аналізованих підходів до виявлення поліморфного, олігоморфного і метаморфного ШПЗ характеризується тією чи іншою мірою певною множиною обмежень. До того ж, ні одна група методів не використовує на власну користь ключову особливість поліморфного (метаморфного) ШПЗ – інваріантну поведінку за певною підмножиною ознак, яка характеризує один і той самий вектор впливу ШПЗ.

У зв’язку з цим виникає актуальне завдання пошуку ефективного рішення щодо виявлення поліморфного і метаморфного ШПЗ, яке б забезпечило нівелювання властивості модифікації коду ШПЗ, а також дозволило аналізувати такі властивості ШПЗ, які характеризують його вектор інформаційно-руйнівного впливу (є інваріантними для конкретного типу ШПЗ). Відтак вирішення цього завдання доцільно здійснювати шляхом визначення інваріантної компоненти у поведінці ШПЗ на основі поєднання переваг поведінкового аналізу та техніки машинного навчання – зменшення розмірності простору ознак (Dimensionality reduction [26] – перетворення досліджуваних даних, що полягає у зменшенні числа ознак (проєкції на нову розмірність) шляхом отримання множини найбільш значущих із них).

Метою статті є обґрунтування вибору підходу до визначення інваріантної компоненти у поведінці поліморфного (метаморфного) ШПЗ на основі зниження розмірності простору ознак.

Обґрунтування вибору підходу до визначення інваріантної компоненти у поведінці ШПЗ. Вирішення завдання нівелювання результатів метапрограмування коду поліморфним (метаморфним) ШПЗ шляхом застосування методів зниження розмірності досліджуваного простору ознак обумовлено потенційним паритетом існуючих систем кіберзахисту та ефективністю уникати виявлення новими зразками ШПЗ, які побудовано на основі раніше відомого (класифікованого) ШПЗ. Важливість цього підходу полягає у можливості адаптивної реакції на постійно змінювані тактики ШПЗ, що вимагає постійного вдосконалення захисних систем без втручання експертів. Методи зниження розмірності дозволяють виділити

статистично значущі, але неочевидні патерни поведінки, що є ключовими для розпізнавання зловмисного ПЗ навіть у випадку його значних модифікацій. До того ж, застосування таких методів дозволяє покращити розуміння структури даних, що представляються навчальною вибіркою про поведінку ШПЗ. Ключовою складовою такого підходу є використання розширеного аналізу простору ознак з метою ідентифікації таких, які залишаються стабільними незалежно від зовнішніх модифікацій коду. Оскільки переважну більшість екземплярів поліморфного (метаморфного) ШПЗ складають модифіковані зразки вже існуючого шкідливого коду [13], то за умови збереження вектора здійснення інформаційно-руйнівного впливу залишається незмінною певна підмножина ознак для кожного типу (класу) ШПЗ. Звідси шукана підмножина ознак описує інваріантну поведінку для кожного типу (класу) ШПЗ, тоді як решта ознак описують його поліморфну (метаморфну) компоненту.

На рисунку 1 зображено узагальнену схему визначення інваріантної компоненти у поведінці ШПЗ на основі зниження розмірності простору досліджуваних ознак, де x_i – ознака.

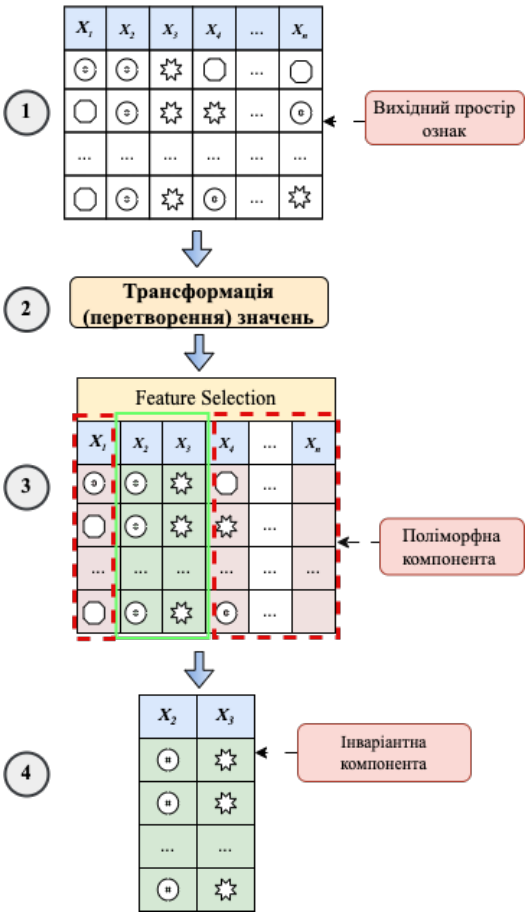

Рис. 1. Узагальнена схема визначення інваріантної компоненти у поведінці ШПЗ на основі зниження розмірності простору досліджуваних ознак

На *першому* кроці ініціалізується вихідний простір ознак $x_1 - x_n$, який описує поведінку ШПЗ, представлену навчальною вибіркою.

На *другому* кроці здійснюється трансформація (перетворення) значень ознак шляхом кодування, наприклад, нечіткими лінгвістичними термами з метою подальшого ефективного формування як множин нечітких правил для кожного відомого класу ШПЗ, так і підмножин нечітких інваріантних компонент для поліморфного (метаморфного) ШПЗ, асоційованого з цими класами ШПЗ.

Таблиця 1

Порівняльна характеристика методів зниження розмірності простору ознак

№	Метод	Основна ідея	Стійкість до шуму	Часова складність	Типи вихідних даних	Результат зменшення розмірності	Можливість інтерпретації отриманого результату правилами	Лінійність
1.	Метод фільтрації	Виділення найбільш інформативних ознак з великої кількості ознак	Чутливий	Середня	- числові; - категоріальні; - текстові; - зображення	Оцінка та вибір найбільш інформативних ознак з урахуванням статистичних метрик, без створення нових ознак	+	Лінійний
2.	Метод опорних ознак	Вибір найкращої підмножини ознак для використання в моделі машинного навчання, виходячи з якості продуктивності моделі на конкретних комбінаціях ознак	Чутливий	Висока	- числові; - категоріальні	Оцінка та вибір найкращої підмножини ознак, без створення нових ознак	+	Гібридний
3.	Топологічний аналіз даних (TDA)	Виявлення складних зав'язків (структур) між даними	Стійкий	Висока	- числові; - зображення; - категоріальні; - геометричні; - текстові; - мережеві; - сигнали; - часові ряди; - множини даних	Топологічні структури (нові дані)	+/-	Нелінійний
4.	Факторний аналіз (FA)	Виявлення латентних зав'язків між змінними	Чутливий	Середня	- числові; - категоріальні	Проєкція існуючих змінних на нижчу розмірність. В результаті отримуються нові змінні, які являють собою лінійні комбінації вихідних ознак	+	Лінійний
5.	Дискримінантний аналіз	Знаходження дискримінантних функцій	Чутливий	Низька	- числові	Перетворення даних на новий простір ознак, використовуючи лінійні комбінації вихідних ознак	+/-	Лінійний
6.	T-розподілене вкладення стохастичної близькості (t-SNE)	Знаходження нелінійних залежностей між вихідними даними	Чутливий	Висока	- числові; - категоріальні	Подання даних для візуалізації на зменшеному просторі, створення нових ознак	+	Нелінійний
7.	Метод головний компонент (PCA)	Знаходження головних компонент	Чутливий	Низька	- числові	Нові ознаки	+/-	Лінійний
8.	Генетичний алгоритм	Моделювання процесів природного відбору та еволюції для пошуку оптимальних рішень	Стійкий	Висока	- числові	Відбір оптимальної комбінації ознак, яка зберігає найбільшу кількість інформації при зниженні розмірност	+	Гібридний
9.	Розклад невід'ємних матриць (NMF)	Розклад матриці на невід'ємні складові	Чутливий	Низька	- текст; - числові; - зображення; - сигнали; - спектральні	Інтерпретація та відображення латентних структур даних, які є новими ознаками	+/-	Нелінійний
10.	Сингулярний розклад матриці (SVD)	Розклад матриці на такий спосіб, щоб виділити основні характеристики даних	Чутливий	Висока	- числові; - текстові; - сигнали; - спектральні	Нові ознаки, які є лінійними комбінаціями вихідних ознак	+	Лінійний
11.	Канонічно-кореляційний аналіз (CCA)	Виявлення кореляційних зав'язків	Чутливий	Висока	- числові	Нові змінні	+/-	Лінійний

На *третьому, четвертому* кроках застосовуються методи зниження розмірності досліджуваного простору ознак з метою визначення інваріантної компоненти у поведінці поліморфного (метаморфного) ШПЗ для кожного відомого класу ШПЗ. Досягнення зазначеної цілі ґрунтується на тому, що кожен окремий екземпляр конкретного класу ШПЗ за своєю суттю є поліморфним (олігоморфним/метаморфним) іншим екземплярам цього класу ШПЗ.

З метою реалізації зазначеного необхідно здійснити порівняльний аналіз існуючих методів зниження розмірності простору ознак для подальшого використання найефективнішого з них.

В математичній статистиці та деяких розділах штучного інтелекту, зокрема у машинному навчанні, методи зниження розмірності простору ознак дозволяють відібрати (Feature selection [26]) або згенерувати (Feature engineering [27]) найбільш інформативні ознаки з великих масивів досліджуваних даних шляхом виключення неінформативних, дубльованих і шумових ознак, зберігаючи при цьому важливу інформацію (приховану структуру даних у процесі проєкції даних на меншу розмірність ознак). У таблиці 1 наведено порівняльну характеристику найбільш поширених методів зниження розмірності простору ознак за характеристиками: стійкості до шуму у даних, часової складності, типів даних, можливості інтерпретації отриманого результату і лінійності.

Метод фільтрації [28] в машинному навчанні здійснює відбір підмножини найбільш інформативних ознак з вихідної множини. Цей метод оцінює і ранжує ознаки за мірою їхньої інформативності для подальшого навчання моделі, використовуючи статистичні метрики або евристики, такі як дисперсія, інформаційний приріст та взаємна інформація. Після відбору підмножини ознак здійснюється оцінка ефективності моделі на предмет наявності випадкових та/або шумових ознак.

Переваги: швидкість та простота реалізації, зменшення ризику перенавчання моделі.

Недоліки: не враховуються взаємозв’язки між ознаками, неефективний у роботі з великими наборами даних, чутливий до шуму даних.

Метод опорних ознак (Support Vector Machines, SVM) [29] – ефективний алгоритм машинного навчання для класифікації та регресії ознак на основі навчальної вибірки даних. Метод спрямований на пошук оптимального розміщення гіперплощини для ефективної сепарабельності вихідної множини ознак на класи, максимізуючи відстань між опорними векторами кожного класу. Для досягнення високої точності вводиться параметр, який контролює допустиму кількість невірно обраних ознак, що дозволяє ефективніше регулювати розподільну гіперплощину.

Переваги: здатність працювати у лінійному та нелінійному просторах, обробка великих наборів даних, висока точність та швидкість класифікації.

Недоліки: значна часова складність, значна залежність від шуму у даних.

Топологічний аналіз даних (Topological Data Analysis, TDA) [30] є інноваційним підходом до аналізу багатовимірних даних, що ґрунтується на принципах топології, вивчаючи властивості об’єктів (петель, порожнеч, групувань) під час неперервних змін. Суть використання TDA в задачах зниження розмірності простору ознак полягає у виявленні складних неочевидних структур (форм) та взаємозв’язків, які можуть бути приховані за шумом та нелінійністю даних. Особливості виявлених топологічних структур представляються меншою кількістю ознак.

Переваги: виявлення складних структур, стійкість до шуму, робота з різними типами даних.

Недоліки: значна часова складність, складна інтерпретація результатів.

Факторний аналіз [31] – це статистичний метод, який використовується для обробки великих наборів змінних (ознак). Цей метод дозволяє описати загальну варіацію даних вихідного набору ознак на основі підходу Feature engineering. Так, завдання зниження

простору ознак досягається шляхом опису вихідного простору ознак меншою кількістю знайдених прихованих взаємозв'язків – факторів (лінійних комбінацій ознак) із кореляційної матриці. В подальшому отримані фактори представляються новими ознаками.

Переваги: виявлення прихованих взаємозв'язків між ознаками, збереження прихованої структури у даних.

Недоліки: складна інтерпретація результатів, складність вибору оптимальної кількості факторів для аналізу, у випадку побудови класифікатора з'являється необхідність постійного перетворення вхідних даних.

Дискримінантний аналіз [32] – це статистичний метод, що призначений для класифікації об'єктів на групи на основі їхніх ознак. Суть застосування такого підходу в задачах зниження розмірності простору ознак зводиться до пошуку таких лінійних комбінацій вихідних ознак (дискримінантних функцій), які найефективніше класифікують об'єкти. Пошук дискримінантних функцій ґрунтується на визначенні напряму, вздовж яких максимізується відношення міжкласової дисперсії до внутрішньокласової дисперсії, що дозволяє ефективніше розділяти класи.

Переваги: збереження прихованої структури у даних.

Недоліки: чутливий до аномалій, у випадку побудови класифікатора з'являється необхідність постійного перетворення вхідних даних.

T-розподілене вкладення стохастичної близькості (T-Distributed Stochastic Neighbor Embedding, T-SNE) [33] – метод нелінійного зменшення розмірності даних та їх візуалізації у меншому просторі. В багатовимірному просторі на основі нормального закону розподілу обчислюються подібності між точками в просторі координат, які представляють досліджувані дані з метою подальшої мінімізації різниці між попарними подібностями. Після оптимізації відстаней координати точок представляють зменшену розмірність даних, зберігаючи при цьому початкову структуру і відношення.

Переваги: ефективна візуалізація, збереження локальних структур, чутливість до збіжностей.

Недоліки: низька масштабованість, значна часова складність.

Метод головних компонент (Principal Component Analysis, PCA) [32] – один із найбільш поширеніших лінійних методів зниження розмірності даних, що дозволяє зберегти важливу інформацію (структуру) з вихідного набору даних. Основна ідея полягає у знаходженні ортогональних компонент, які максимально відображають варіацію у вихідних даних, шляхом їх проєкції, щоб всі вісі захоплювали максимальну дисперсію даних.

Переваги: збереження основної структури при проєкції на зменшену розмірність, проєкція для максимальної дисперсії.

Недоліки: чутливість до викидів та шуму, у випадку побудови класифікатора з'являється необхідність постійного перетворення вхідних даних.

Генетичні алгоритми (Genetic Algorithms) [34] є еволюційним підходом до вирішення задачі зниження розмірності простору ознак, який базується на імітації природного процесу еволюції шляхом відбору підмножини найбільш пристосованих ознак з усієї множини простору ознак. Суть застосування цього підходу полягає у виборі таких рішень – хромосом (підмножин ознак), побудованих з генів, – ознак, які найбільше відповідають функції пристосованості

Основні кроки включають створення початкової популяції хромосом, де кожна хромосома являє собою комбінацію ознак, оцінку пристосованості кожної хромосоми та відбір найбільш пристосованих для наступного покоління, генерацію нового покоління шляхом мутації або кросоверу (схрещування) та оцінку нового покоління для подальшої заміни в популяції. Такий процес повторюється до досягнення критерію зупинки, такого як кількість ітерацій або досягнення заданого рівня пристосованості.

Переваги: пошук глобального оптимуму, гнучкість, стійкість до шуму.

Недоліки: значна часова складність.

Розклад невід'ємних матриць (Non-negative Matrix Factorization, NMF) [35] – алгоритми багатовимірного аналізу даних, які використовуються для зменшення розмірності простору ознак шляхом розкладання вихідної матриці даних на дві невід'ємні матриці меншої розмірності. Вихідна матриця містить об'єкти (рядки) та їхні ознаки (стовпці). Під час розкладання матриці отримуємо: матрицю базисних векторів, яка визначає базові ознаки досліджуваних об'єктів, та матрицю зі зваженими ознаками.

Переваги: легка інтерпретація результатів, збереження важливої інформації.

Недоліки: вибір початкових значень матриці, локальні мінімуми – менш точні результати, чутливий до шуму.

Сингулярний розклад матриці (Singular Value Decomposition, SVD) [32] – метод розкладання матриці для аналізу та обробки даних, зокрема для зниження розмірності даних. Перший етап SVD полягає в побудові вихідної матриці даних, де рядки відповідають об'єктам, а стовпці – їхнім ознакам. Для коректності розкладу застосовується нормалізація даних, оскільки SVD є дуже чутливим до масштабування. Наступним етапом є розкладання вихідної матриці на добуток трьох ранг-матриць:, кожна з яких представляє окрему частину інформації вихідної матриці. Головні компоненти, які необхідно зберегти, обираються на основі важливості інформації та використовуються у формуванні нової матриці зі зменшеною розмірністю.

Переваги: стійкий до шуму, робота з різними типами даних.

Недоліки: складний для розуміння та реалізації, значна часова складність, складна інтерпретація результатів, часткова втрата інформації.

Канонічно-кореляційний аналіз (Canonical-Correlation Analysis, CCA) [36] – статистичний метод дослідження взаємозв'язків між наборами даних. Основною метою CCA є знаходження лінійних комбінацій (нових змінних) кожного набору даних, які максимально корелюють між собою. Це дозволяє виокремити найважливіші кореляційні структури для подальшої інтерпретації результатів. Під час зниження розмірності за допомогою CCA обирається кількість канонічних змінних, які найефективніше пояснюють взаємозв'язки між наборами даних, що дозволяє забезпечити оптимальне співвідношення між розмірністю даних та їхнім інформаційним вмістом.

Переваги: максимальна кореляція, виокремлення важливої інформації.

Недоліки: складність інтерпретації результатів, залежність від лінійності, вразливість до викидів.

На основі проведеного аналізу зазначених методів зниження розмірності простору досліджуваних ознак можна зробити наступні висновки:

1. Для вирішення поставленого завдання найбільш доцільними є застосування методів, суть застосування яких зводиться до використання підходу ***Feature selection***: *метод фільтрації, метод опорних ознак, генетичні алгоритми*, оскільки на відміну від підходу ***Feature engineering***: *топологічний аналіз даних, факторний аналіз, дискримінантний аналіз, Т-розподілене вкладення стохастичної близькості, метод головних компонент, розкладання невід'ємних матриць, сингулярне розкладання матриці, канонічно-кореляційний аналіз*, не передбачається перетворення вихідного простору ознак на нові ознаки.

2. З-поміж методів ***Feature selection*** найбільшої уваги заслуговують методи, функціональне ядро яких передбачає використання генетичних алгоритмів, оскільки дозволяють потенційно досягти глобального оптимуму в задачах пошуку оптимальної підмножини найбільш значущих ознак.

3. Використання генетичних алгоритмів для вирішення завдання визначення інваріантної і поліморфної компонент у поведінці ШПЗ на основі зниження розмірності

простору досліджуваних ознак не є чутливим до шуму (аномалій) в наборах даних, які обираються у якості навчальної вибірки, що значно підвищує показник точності моделі.

Таким чином, для підвищення ефективності виявлення поліморфного, олігоморфного і метаморфного ШПЗ запропоновано визначати інваріантну компоненту в поведінці відомих його типів на основі зниження розмірності простору ознак засобами генетичних алгоритмів. Особливістю застосування цього підходу є збереження основної структури (топологічної форми) даних, прихованої у навчальній вибірці значень про поведінку ШПЗ у системі в умовах наявності шуму у даних, а також використання ключової характеристики поліморфного (метаморфного) ШПЗ – інваріантної поведінки, властивої конкретному типу ШПЗ, описану певною підмножиною ознак. Такий підхід дозволяє сформувати фундамент для пошуку неочевидних збіжностей у поведінці різних екземплярів деструктивного ПЗ (вірусів).

Висновки. У статті вирішується завдання обґрунтування вибору підходу до визначення інваріантної компоненти у поведінці поліморфного (метаморфного) ШПЗ на основі зниження розмірності простору ознак.

Актуальність зазначеного завдання обумовлено стрімким збільшенням кіберінцидентів протягом останнього часу, пов’язаних із застосуванням поліморфного (метаморфного) ШПЗ. Основна причина цього зросту – доступність технологій ШІ, які дозволяють зловмисникам досить швидко та ефективно модифікувати код вже класифікованих шкідливих програм, не потребуючи значних спеціалізованих компетенцій.

Проведені дослідження існуючих підходів до виявлення поліморфного, олігоморфного і метаморфного ШПЗ показали наявність певних обмежень, що значно знижують їхню ефективність. До того ж, ні одна група методів не використовує на свою користь ключову особливість поліморфного (метаморфного) ШПЗ – інваріантну поведінку за певною підмножиною ознак, яка характеризує один і той самий вектор деструктивного впливу ШПЗ. З урахуванням зазначеного, запропоновано підхід до вирішення завдання виявлення поліморфного (метаморфного) ШПЗ, який ґрунтується на ідеї визначення інваріантної компоненти у поведінці ШПЗ на основі поєднання переваг поведінкового аналізу та зменшення розмірності досліджуваного простору ознак ШПЗ.

Проведено аналіз існуючих методів зниження розмірності простору ознак, результати аналізу якого виділяють генетичні алгоритми як групу найефективніших еволюційних методів пошуку оптимальної підмножини значущих ознак.

Такий підхід потенційно дозволить значно ефективніше виявляти функціонування поліморфного (метаморфного) ШПЗ на основі інформації про типову інваріантну поведінку відомих класів ШПЗ у вигляді підмножини досліджуваних ознак, що у свою чергу формує підґрунтя для реалізації нового підходу до ефективного виявлення модифікованого (удосконаленого) ШПЗ, в тому числі і засобами технологій ШІ.

Таким чином, отримані результати є підґрунтям для подальших наукових досліджень, які полягають у розробці математичної моделі визначення інваріантної компоненти у поведінці ШПЗ під час динамічного аналізу на основі зменшення розмірності простору ознак засобами генетичних алгоритмів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Системи виявлення вразливостей і реагування на кіберінциденти та кібератаки. Оперативний центр реагування на кіберінциденти державного центру кіберзахисту Державної служби спеціального зв’язку та захисту інформації України. 2023. 14 с.

2. Російські кібероперації. Аналітика за перше півріччя 2023 року. Державна служба спеціального зв’язку та захисту інформації України. 2023. 23 с.

3. Кібербезпека в інформаційному суспільстві: Інформаційно-аналітичний дайджест / відп. ред. О. Довгань; упоряд. О. Довгань, Л. Литвинова, С. Дорогих; Державна наукова установа «Інститут інформації, безпеки і права НАПрН України»; Національна бібліотека України ім. В. І. Вернадського. К., 2024. № 1 (січень). 327 с.

4. Islam M. The next frontier: AI and the evolution of polymorphic malware. *LinkedIn: Log In or Sign Up*. URL: https://www.linkedin.com/pulse/next-frontier-ai-evolution-polymorphic-malware-moinul-islam-ov4nc?trk=public_post_main-feed-card_feed-article-content.

5. ChatGPT AI technology of the century or potential weapon in the hands of cybercriminals? URL: https://blackberry.bakotech.com/chatgpt-en.

6. Generative AI is the pride of cybercrime services. *Check Point*. URL: https://blog.checkpoint.com/research/generative-ai-is-the-pride-of-cybercrime-services/.

7. Ben-Moshe S., Gekker G., Cohen G. OpwnAI: AI that can save the day or HACK it away - check point research. *Check Point Research*. URL: https://research.checkpoint.com/2022/opwnai-ai-that-can-save-the-day-or-hack-it-away/.

8. Kaspersky: more than 36 million AI & gaming credentials compromised by infostealers in 3 years. *www.kaspersky.com*. URL: https://www.kaspersky.com/about/press-releases/2024_kaspersky-more-than-36-million-ai-gaming-credentials-compromised-by-infostealers-in-3-years.

9. Shimony E., Tsarfati O. Chatting our way into creating a polymorphic malware. *Identity Security and Access Management Leader. CyberArk*. URL: https://www.cyberark.com/resources/threat-research/chatting- our-way-into-creating-a-polymorphic-malware.

10. Sims J. BlackMamba: using AI to generate polymorphic malware. *HYAS The Authority on Cyber Threat Adversary Infrastructure*. URL: https://www.hyas.com/blog/blackmamba-using-ai-to-generate-polymorphic- malware.

11. Sims J. EyeSpy Proof-of-Concept. *HYAS The Authority on Cyber Threat Adversary Infrastructure*. URL: https://www.hyas.com/blog/eyespy-proof-of-concept.

12. Sharma S. ChatGPT creates mutating malware that evades detection by EDR. *CSO Online*. URL: https://www.csoonline.com/article/575487/chatgpt-creates-mutating-malware-that-evades-detection-by-edr.html.

13. Фесьоха В. В., Кисиленко Д. Ю., Нестеров О. М. Аналіз спроможності існуючих систем антивірусного захисту та покладених у їхню основу методів до виявлення нового шкідливого програмного забезпечення у військових інформаційних системах. *Системи і технології зв'язку, інформатизації та кібербезпеки*. 2023. Т. 3. С. 143–151.

14. The differences between static and dynamic malware analysis. *Bitdefender Blog*. URL: https://www.bitdefender.com/blog/businessinsights/the-differences-between-static-malware-analysis-and-dynamic-malware-analysis/.

15. A state of the art survey on polymorphic malware analysis and detection techniques / E. Masabo et al. *RUFORUM Institutional Repository | RUFORUM Institutional Repository*. URL: https://repository.ruforum.org/sites/default/files/IJSC_Vol_8_Iss_4_Paper_9_1762_1774.pdf.

16. What is heuristics evasion? Outsmarting heuristic antivirus systems. *ReasonLabs Cyberpedia*. URL: https://cyberpedia.reasonlabs.com/EN/heuristics%20evasion.html.

17. Shetty S. What is heuristic analysis and why is it important for cybersecurity? *TechGenix*. URL: https://techgenix.com/heuristic-analysis-cybersecurity/.

18. Deng X., Mirkovic J. Malware behavior through network trace analysis. *Lecture Notes in Networks and Systems. Selected Papers from the 12th International Networking Conference*. 2020. Vol. 180. P. 3–18. URL: https://doi.org/10.1007/978-3-030-64758-2.

19. Selamat N. S., Al F. H. M. Polymorphic malware detection based on supervised machine learning. *Journal of positive school psychology*. 2022. Vol. 6, no. 3. P. 8538–8547. URL: https://journalppw.com/index.php/jpsp/issue/view/30.

20. Akhtar M. S., Feng T. Malware analysis and detection using machine learning algorithms. *Symmetry*. 2022. Vol. 14, no. 11. P. 2304. URL: https://doi.org/10.3390/sym14112304.

21. Using discriminative rule mining to discover declarative process models with non-atomic activities/ M. Bernardi et al. *International web rule symposium*. 2014. URL: https://www.semanticscholar.org/paper/Using-Discriminative-Rule-Mining-to-Discover-Models-Bernardi-Cimitile/ef426bfa04caac0c91e9e3fc476d938f27321db8.

22. Bernardi M. L., Cimitile M., Mercaldo F. Process mining meets malware evolution: a study of the behavior of malicious code. *International symposium on computing and networking – across practical development and theoretical research*. 2016. URL: https://www.semanticscholar.org/paper/Process-Mining-Meets-Malware-Evolution:-A-Study-of-Bernardi-Cimitile/7838664913ba2ab34d78f6120188293bd77a7fb3.

23. Ardimento P., Bernardi M. L., Cimitile M. Malware phylogeny analysis using data-aware declarative process mining. *IEEE conference on evolving and adaptive intelligent systems (EAIS)*. 2020. URL: https://www.semanticscholar.org/paper/Malware-Phylogeny-Analysis-using-Data-Aware-Process-Ardimento-Bernardi/859dd8a091b4af71426a189225ee09a3a2e78a69.

24. Data-Aware declarative process mining for malware detection / P. Ardimento et al. *EEE international joint conference on neural network*. 2020. URL: http://vigir.missouri.edu/~gdesouza/Research/Conference_CDs/IEEE_WCCI_2020/IJCNN/Papers/N-21418.pdf.

25. Субач І., Фесьоха В., Фесьоха Н. Фесьоха Н. О. Аналіз існуючих рішень запобігання вторгненням в інформаційно-телекомунікаційні мережі. *Information Technology and Security*. 2017. Т. 5, № 1. С. 29–41. URL: http://nbuv.gov.ua/UJRN/inftech_2017_5_1_6.

26. Sanjyal A. Dimensionality reduction VS feature selection. *Medium*. URL: https://medium.com/@asanjyal81/dimensionality-reduction-vs-feature-selection-e68f91aa8724.

27. Kumar B. What is feature engineering in dimensionality reduction - 360digitmg. *360digitmg.com*. URL: https://360digitmg.com/blog/feature-engineering-in-dimensionality-reduction.

28. Calledda C. Focus on filter methods for feature selection. *LinkedIn: Log In or Sign Up*. URL: https://www.linkedin.com/pulse/focus-filter-methods-feature-selection-carlo-calledda-7kene.

29. Banerjee S. From high dimensions to clarity: unraveling complex data with support vector machines and principal. *Medium*. URL: https://shekhar-banerjee96.medium.com/from-high-dimensions-to-clarity-unraveling-complex-data-with-support-vector-machines-and-principal-78d3871af248.

30. Munch E. A user's guide to topological data analysis. *Journal of learning analytics*. 2017. Vol. 4, no. 2. P. 47–61. URL: https://doi.org/10.18608/jla.2017.42.6.

31. Factor analysis: how to reduce the complexity and dimensionality of your data - fastercapital. *FasterCapital*. URL:https://fastercapital.com/content/Factor-Analysis--How-to-Reduce-the-Complexity-and-Dimensionality-of-Your-Data.html.

32. Baruah I. D. Dimensionality reduction techniques – PCA, LCA and SVD. *Medium*. URL: https://medium.com/nerd-for-tech/dimensionality-reduction-techniques-pca-lca-and-svd-f2a56b097f7c#:~:text=SVD%20allows%20for%20dimensionality%20reduction,significant%20singular%20values%20and%20vectors.&text=SVD%20is%20used%20in%20data,storage%20requirements%20of%20a%20matrix.&text=By%20using%20only%20the%20most,of%20noise%20in%20the%20data.

33. Pajak A. T-SNE: t-distributed stochastic neighbor embedding. *Medium*. URL: https://medium.com/@pajakamy/dimensionality-reduction-t-sne-7865808b4e6a.

34. Метод виявлення кіберзагроз на основі еволюційних алгоритмів / С. М. Лисенко, Д. І. Стопчак, В. В. Самотес. *Вісник Хмельницького національного університету. Технічні науки*. 2017. № 6. С. 81–88. URL: http://nbuv.gov.ua/UJRN/Vchnu_tekh_2017_6_15.

35. Olaya J., Otman C. Non-negative matrix factorization for dimensionality reduction. *ITM web of conferences*. 2022. Vol. 48. P. 03006. URL: https://doi.org/10.1051/itmconf/20224803006.

36. Karwowska Z. Canonical Correlation analysis−simple explanation and python example. *Medium*. URL: https://medium.com/@pozdrawiamzuzanna/canonical-correlation-analysis-simple-explanation-and-python-example-a5b8e97648d2.