DOI: https://doi.org/10.26565/2304-6201-2025-67-07

УДК (UDC) 004.056.53:004.032.26

Ланін Євген Сергійович *студент магістратури ННІ комп'ютерних наук та штучного інтелекту, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, м. Харків, 61022*
e-mail:lanin2020ki12@student.karazin.ua;
https://orcid.org/0009-0003-2639-6218

Бакуменко Ніна Станіславівна *к.т.н., доцент, доцент кафедри комп'ютерних систем та робототехніки, Харківський національний університет імені В. Н. Каразіна, майдан Свободи, 4, м. Харків, 61022*
e-mail: n.bakumenko@karazin.ua ;
https://orcid.org/0000-0003-3496-7167

Застосування методів машинного навчання для детекції зловмисного програмного забезпечення в дампах оперативної пам'яті

Актуальність. У сучасних умовах постійного зростання кіберзагроз особливу актуальність набуває проблема виявлення зловмисного програмного забезпечення, яке може функціонувати приховано в оперативній пам'яті, використовуючи техніки безфайлових атак. Традиційні антивірусні рішення, що базуються переважно на сигнатурному підході, виявляються неефективними проти сучасних advanced persistent threats (APT) та нових модифікованих загроз. Це робить актуальною розробку інноваційних підходів до детекції зловмисного програмного забезпечення на основі аналізу поведінкових патернів в оперативній пам'яті з використанням методів машинного навчання.
Мета роботи: розробка та апробація системи автоматизованого виявлення зловмисного програмного забезпечення шляхом аналізу дампів оперативної пам'яті з використанням методів машинного навчання, а також порівняльна оцінка ефективності різних алгоритмів класифікації для багатокласової детекції типів загроз.
Методи дослідження: порівняльний аналіз алгоритмів машинного навчання, статичний аналіз дампів пам'яті, багатокласова класифікація, експериментальна апробація.
Результати. Створено технологічний конвеєр (pipeline) для автоматизованої обробки та класифікації дампів оперативної пам'яті. Проведено порівняльний аналіз 13 алгоритмів машинного навчання, який продемонстрував, що найкращі результати для задачі багатокласової класифікації ЗПЗ показує Random Forest з точністю 85.49% та F1-score 85.52%. Розроблена система реалізована на мові Python з використанням бібліотек scikit-learn (для класичних ML моделей), TensorFlow/Keras (для нейронних мереж) та pandas (для обробки даних).
Висновки. Дослідження підтвердило високу ефективність класичних методів машинного навчання, зокрема ансамблевих алгоритмів, для виявлення зловмисного програмного забезпечення в дампах оперативної пам'яті. Створена модель на основі Random Forest забезпечує оптимальний баланс між точністю класифікації (85.52% F1-score), швидкістю навчання (1.3 с) та обчислювальною ефективністю, демонструючи значні переваги над нейронними мережами у даному контексті. Розроблена система має високу практичну значущість і може бути інтегрована у форензічні платформи, системи моніторингу інцидентів кібербезпеки та експертні системи для автоматизованого виявлення загроз і прискорення процесу аналізу інцидентів. Результати дослідження підтверджують доцільність використання методів машинного навчання для створення систем захисту від сучасних кіберзагроз, що функціонують виключно в оперативній пам'яті.

Як цитувати: Ланін Є. С., Бакуменко Н. С. Застосування методів машинного навчання для детекції зловмисного програмного забезпечення в дампах оперативної пам'яті. *Вісник Харківського національного університету імені В. Н. Каразіна, серія Математичне моделювання. Інформаційні технології. Автоматизовані системи управління*. 2025. вип. 67. С.76-82. https://doi.org/10.26565/2304-6201-2025-67-07

How to quote: Y. Lanin, and N. Bakumenko, "Machine Learning Approaches to Malware Detection in RAM" *Bulletin of V. N. Karazin Kharkiv National University, series "Mathematical modelling. Information technology. Automated control systems*, vol. 67, pp. 76-82, 2025. https://doi.org/10.26565/2304-6201-2025-67-07

Вступ

У сучасних умовах постійного зростання кіберзагроз особливу актуальність набуває проблема виявлення зловмисного програмного забезпечення (ЗПЗ), яке може функціонувати приховано, зокрема, тільки в оперативній пам'яті. Сучасні кібератаки стають все частішими, витонченішими та результативнішими, ставлячи під загрозу конфіденційну інформацію та об'єкти критичної інфраструктури. Додаткове занепокоєння викликають нові вектори атак, що експлуатують вразливості та ризики, пов'язані з новітніми технологіями, зокрема зі штучним інтелектом [1].

Зловмисники все частіше використовують техніки безфайлових атак, коли шкідливий код завантажується безпосередньо в оперативну пам'ять, не залишаючи слідів у файловій системі. Це, а також поширення програм-вимагачів (ransomware) [6], значно ускладнює виявлення традиційними засобами захисту.

Традиційні антивірусні рішення, що базуються переважно на сигнатурному підході, виявляються неефективними проти сучасних advanced persistent threats (APT) та нових модифікованих загроз [6]. Це робить актуальною розробку інноваційних підходів до детекції ЗПЗ. Особливу увагу слід приділити методам, що базуються на аналізі поведінкових патернів в оперативній пам'яті, таким як аналіз викликів API та бібліотек DLL за допомогою моделей машинного навчання.

1. Огляд літератури та існуючих методів

Метою дослідження є розробка комп'ютерної моделі, яка дозволяє виявляти ознаки присутності ЗПЗ шляхом аналізу дампів оперативної пам'яті. Основні завдання включають побудову системи збору дампів, їх попередню обробку, а також розробку алгоритмів ідентифікації підозрілих структур.

У процесі дослідження проаналізовано сучасні інструменти, зокрема Volatility та Rekall, а також підходи до використання машинного навчання для виявлення зразків ЗПЗ. Порівняльний аналіз основних інструментів аналізу пам'яті наведений в таблиці 1.

Табл.1 Порівняльний аналіз інструментів для аналізу дампів пам'яті
Table. 1 Comparative Analysis of Memory Dump Analysis Tools

Інструмент	Переваги	Недоліки	Продуктивність
Volatility	Широкий функціонал, активна спільнота	Повільна обробка великих дампів	2-5 ГБ/год
Rekall	Швидша обробка, модульна архітектура	Обмежена підтримка ОС	5-8 ГБ/год
MemProcFS	Автоматизований аналіз, низьке споживання ресурсів	Обмежений функціонал	10-15 ГБ/год

Сучасні методи виявлення шкідливого програмного забезпечення можна класифікувати на статичні, динамічні та методи машинного навчання. Статичні методи аналізують код програм без їх виконання та мають точність 85-90% для відомих загроз, але лише 30-40% для нових варіантів. Динамічні методи спостерігають за поведінкою програм під час виконання, досягаючи точності 70-80% з високою кількістю помилкових спрацювань. Методи машинного навчання використовують алгоритми класифікації для виявлення нових загроз, демонструючи найкращі результати з точністю 90-95% [7].

Аналіз останніх досліджень показує, що найперспективнішими є гібридні підходи, які поєднують різні методи детекції. Зокрема, дослідження Li et al. (2024) [8] продемонстрували ефективність використання графових нейронних мереж для аналізу структури процесів у пам'яті, досягнувши точності 96.2%. Aljabri et al. (2024) [2] запропонували спеціалізований підхід для детекції ransomware на основі аналізу memory features з точністю 94.8%.

Основними недоліками існуючих рішень є їх висока залежність від сигнатурних баз, що не дозволяє ефективно виявляти нові та модифіковані типи ЗПЗ. Більшість аналізаторів пам'яті працюють у пост-інцидентному режимі і потребують ручного аналізу експертами. Комерційні рішення, такі як FireEye та CrowdStrike, хоча й демонструють високу ефективність, мають значну вартість та потребують спеціалізованої інфраструктури.

2. Структура pipeline для детекції malware методами машинного навчання

Запропонована модель ґрунтується на принципах статичного аналізу дампів оперативної пам’яті із застосуванням методів машинного навчання, що забезпечує можливість виявлення ознак зловмисної активності без необхідності виконання коду в контрольованому середовищі. Архітектура системи побудована за модульним принципом і включає кілька послідовних етапів обробки даних, кожен з яких виконує окрему функцію у загальному процесі аналізу та класифікації.

На першому етапі реалізується модуль попередньої обробки дампів пам’яті, який відповідає за завантаження первинних даних, їх очищення, нормалізацію та стандартизацію. На цьому етапі також здійснюється екстракція ключових ознак, що характеризують поведінку процесів у пам’яті, таких як кількість потоків виконання, активні дескриптори, використані бібліотеки та інші структурні параметри. Метою цього етапу є підготовка однорідного та придатного до аналізу набору даних, який може бути безпосередньо використаний у подальших модулях системи.

Другий етап представлений компонентом feature engineering, що забезпечує поглиблене опрацювання та розширення простору ознак. У межах цього процесу здійснюється виділення статистичних характеристик процесів, аналіз поведінкових патернів, а також кодування категоріальних змінних для узгодження їх з вимогами алгоритмів машинного навчання. Застосування цього підходу дозволяє зменшити вплив надлишкових або корельованих параметрів, підвищуючи точність і стійкість побудованих моделей.

На третьому етапі функціонує модуль машинного навчання, у якому реалізовано порівняльний аналіз різних алгоритмів класифікації. Дослідження охоплює як класичні методи машинного навчання, так і архітектури штучних нейронних мереж. Для кожного алгоритму проводиться оцінювання продуктивності, точності, стабільності та часу навчання, що дозволяє визначити оптимальні підходи до вирішення задачі багатокласової класифікації зразків зловмисного програмного забезпечення. Особлива увага приділяється аналізу ефективності ансамблевих методів, які продемонстрували високу точність при помірних витратах обчислювальних ресурсів.

Завершальним компонентом є оцінювання та звітності, яка відповідає за інтерпретацію результатів класифікації. На цьому етапі здійснюється багатокласова класифікація типів загроз із використанням метрик якості, таких як accuracy, precision, recall та F1-score, що дає змогу об’єктивно оцінити ефективність кожної моделі. Крім того, система генерує детальні звіти з візуалізацією отриманих результатів, що спрощує аналіз та подальше вдосконалення алгоритмів.

Запропонована модульна архітектура системи забезпечує комплексний підхід до виявлення зловмисного програмного забезпечення через статичний аналіз дампів оперативної пам'яті. Послідовна реалізація чотирьох основних етапів – попередньої обробки даних, інженерії ознак, машинного навчання та оцінювання результатів створює ефективний технологічний конвеєр для автоматизованої детекції та класифікації загроз, що забезпечує повний цикл обробки даних – від збору та підготовки дампів пам’яті до формування підсумкових аналітичних звітів, що робить систему гнучким і масштабованим інструментом для дослідження та виявлення зловмисного програмного забезпечення.

3. Тестовий набір даних

У дослідженні використовується датасет Obfuscated-MalMem2022 [3], який містить понад 58 000 записів з 58 ознаками, що характеризують поведінку процесів Windows в оперативній пам'яті. Ключовими перевагами датасету є збалансованість класів, наявність розширеної категоризації типів шкідливого ПЗ та відсутність пропущених значень, що спрощує процес побудови та валідації моделей машинного навчання.

Набір даних охоплює різноманітні метрики: кількість потоків виконання (threads), список завантажених бібліотек (loaded modules), використані дескриптори ресурсів (handles), ознаки ін'єкції коду (code injection indicators), характеристики служб (services) та модулів ядра (kernel modules). Набір даних є збалансованим відносно класів з можливістю багатокласової класифікації типів ЗПЗ. Дослідження реалізовано як задачу мультикласової класифікації з чотирма основними категоріями[4]:

- Benign (легітимні процеси);
- Trojan (троянські програми);
- Spyware (шпигунське ПЗ);

- Ransomware (програми-вимагачі).

4. Використання методів машинного навчання для класифікації ЗПЗ

Формально задачу багатокласової класифікації можно сформулювати так:

$$f : X \to Y \tag{1}$$

де $X \in \mathbb{R}^{58}$ – вектор ознак процесу, $Y \in \{0,1,2,3\}$ - мітка класу (0 - Benign, 1 - Ransomware, 2 - Spyware, 3 - Trojan).

Для багатокласової класифікації використовуються метрики weighted averaging :

$$Precision_{weighted} = \frac{\sum_{i=1}^{K} n_i Precision_i}{\sum_{i=1}^{K} n_i} \tag{2}$$

де K – кількість класів, n_i – кількість зразків класу i.

$$Recall_{weighted} = \frac{\sum_{i=1}^{K} n_i * Recall_i}{\sum_{i=1}^{K} n_i} \tag{3}$$

де K – кількість класів, n_i – кількість зразків класу i.

Для класифікації випадків були застосовані методи випадкового лісу (Random Forest), градієнтного бустінгу (Gradient Boosting), метод дерев рішень (Decision Tree), k найближчих сусідів (k-Nearest Neighbors) та нейромережеві методи [5]. Для навчання та оцінки моделей використовувалася стратифікована розбивка даних з співвідношенням 80/20 для навчальної та тестової вибірок. Для забезпечення відтворюваності результатів та об'єктивності порівняння, у всіх експериментах використовувалися фіксовані параметри ініціалізації генератора псевдовипадкових чисел. Метрики оцінювання включали: accuracy, precision, recall, F1-score з weighted averaging для коректної обробки багатокласової задачі. Також враховувався час навчання кожної моделі для оцінки практичної застосовності. Результати порівняльного аналізу роботи алгоритмів наведені в таблиці 2.

Табл. 2 Результати порівняльного аналізу алгоритмів машинного навчання
Table. 2 Comparative Analysis Results of Machine Learning Algorithms

Алгоритм	Тип	Accuracy	Precision	Recall	F1-Score	Час навчання (с)
Random Forest	Classical ML	0.8549	0.8558	0.8549	0.8552	1.3
Gradient Boosting	Classical ML	0.8457	0.8463	0.8457	0.8458	220.62
Decision Tree	Classical ML	0.8445	0.8458	0.8445	0.8449	1.65
K-Nearest Neighbors	Classical ML	0.8119	0.8131	0.8119	0.8117	6.7
Extra Trees	Classical ML	0.8044	0.8117	0.8044	0.8046	0.59
Wide & Deep Network	Neural Network	0.7707	0.7780	0.7707	0.7688	29.99
Deep NN (with BatchNorm)	Neural Network	0.7654	0.7737	0.7654	0.7581	45.86
Feedforward NN	Neural Network	0.7591	0.7778	0.7591	0.7533	27.84

Серед класичних алгоритмів машинного навчання найкращу продуктивність показали Random Forest з оптимальним співвідношенням точності та швидкості, Decision Tree з високою інтерпретовністю при хорошій точності, та Extra Trees з найшвидшим часом навчання 0.59 секунди при прийнятній точності. Нейронні мережі продемонстрували стабільні, але менш вражаючі результати, де Wide & Deep Network показала найкращі результати серед нейронних мереж з F1-score 76.88%, проте всі нейронні мережі потребували значно більше часу на навчання від 27 до 46 секунд і мають можливості для покращення через оптимізацію архітектури та гіперпараметрів. Дослідження показало, що для задачі багатокласової класифікації зловмисного програмного забезпечення на основі аналізу дампів пам'яті найефективнішими є класичні алгоритми машинного навчання, зокрема Random Forest, що узгоджується з результатами інших досліджень у галузі кібербезпеки, де ансамблеві методи часто демонструють кращу продуктивність.

5 Аналіз результатів

Найкращі результати продемонстрував алгоритм Random Forest з точністю 85.49% та F1-score 85.52% при мінімальному часі навчання 1.3 секунди. Це підтверджує ефективність ансамблевих методів для задач багатокласової класифікації зловмисного ПЗ.

Gradient Boosting показав дуже близькі результати (F1-score 84.58%), але з значно більшим часом навчання (220.62 секунди), що робить його менш практичним для масштабного використання.

Нейронні мережі показали помірні результати з F1-score в діапазоні 76-77%, що може бути пов'язано з відносно невеликим розміром датасету або потребою в додатковому налаштуванні гіперпараметрів.

6. Висновки та напрямки подальших досліджень

Результати проведеного дослідження засвідчили високу ефективність використання методів машинного навчання для розв’язання задачі багатокласової класифікації зловмисного програмного забезпечення на основі аналізу дампів оперативної пам’яті. Запропонований підхід довів доцільність застосування інтелектуальних алгоритмів для автоматизованого виявлення та ідентифікації загроз, що функціонують у пам’яті системи, без необхідності прямого втручання експерта.

У межах виконаного дослідження реалізовано комплексний технологічний конвеєр (pipeline), який забезпечує проведення порівняльного аналізу тринадцяти алгоритмів машинного навчання для задачі класифікації зразків зловмисного програмного забезпечення. Проведене моделювання дало змогу визначити алгоритм Random Forest як найбільш ефективний серед протестованих моделей, що підтверджується досягнутими показниками точності (85,49%) та метрики F1-score (85,52%). Отримані результати демонструють переваги класичних методів машинного навчання над нейронними мережами у контексті даного типу задач, зокрема завдяки меншій обчислювальній складності, стабільності результатів і високій швидкості навчання. Крім того, створено практичну систему, орієнтовану на статичний аналіз дампів пам’яті, яка може бути використана як базовий компонент для побудови модулів автоматичного моніторингу загроз.

Подальший розвиток дослідження доцільно спрямувати на вдосконалення архітектур нейронних мереж і оптимізацію їх гіперпараметрів з метою підвищення точності та узагальнювальної здатності моделей. Перспективним напрямом є розширення навчального набору даних за рахунок нових типів зловмисного програмного забезпечення, що дозволить підвищити стійкість системи до нових і модифікованих варіантів загроз. Значний потенціал має також інтеграція розробленої моделі з іншими джерелами інформації — зокрема, з аналізом мережевого трафіку, системними журналами (лог-файлами) та телеметричними даними, що сприятиме формуванню комплексної системи кіберзахисту.

Особливої уваги потребує дослідження можливостей застосування методів пояснюваного штучного інтелекту (Explainable AI), які забезпечують прозорість процесу прийняття рішень і дозволяють інтерпретувати внутрішні механізми класифікації. Крім того, перспективним є використання підходів трансферного навчання для адаптації вже навчених моделей до нових типів загроз без потреби у повному перенавчанні.

СПИСОК ЛІТЕРАТУРИ

1. Гайдук О., Звєрєв В. Аналіз кіберзагроз в умовах стрімкого розвитку інформаційних технологій. *Кібербезпека: освіта, наука, техніка*. 2024. Т. 3, № 23. С. 225–236. URL: https://csecurity.kubg.edu.ua/index.php/journal/article/view/552.
2. Aljabri M., Al. E. Ransomware detection based on machine learning using memory features. *Egyptian informatics journal*. 2024. Vol. 25. P. 100445. URL: https://doi.org/10.1016/j.eij.2024.100445.
3. Canadian Institute for Cybersecurity. Malware memory analysis. URL: https://www.unb.ca/cic/datasets/malmem-2022.html.
4. Dhanya K. A., Al. E. Detection of network attacks using machine learning and deep learning models. *Procedia computer science*. 2023. Vol. 218. P. 57–66. URL: https://doi.org/10.1016/j.procs.2022.12.401.
5. Géron A. Hands-On machine learning with scikit-learn, keras, and tensorflow: concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Incorporated, 2022. 483 p.
6. Impact, vulnerabilities, and mitigation strategies for cyber-secure critical infrastructure / H. Riggs et al. *MDPI*. URL: https://www.mdpi.com/1424-8220/23/8/4060 (Last accessed: 29.10.2025).
7. Kumar S., Al. E. Malware classification using machine learning models. *Procedia computer science*. 2024. Vol. 235. P. 1419–1428. URL: https://doi.org/10.1016/j.procs.2024.04.133.
8. Li Q., Al E. MDGraph: a novel malware detection method based on memory dump and graph neural network. *Expert systems with applications*. 2024. P. 124776. URL: https://doi.org/10.1016/j.eswa.2024.124776.

REFERENCES

1. O. Haiduk, V. Zvieryev, "Analysis of cyber threats in the context of rapid development of information technologies", Cybersecurity: education, science, technology, vol. 3, no. 23, pp. 225–236, 2024. [in Ukrainian]. URL: https://csecurity.kubg.edu.ua/index.php/journal/article/view/552.
2. M. Aljabri, et al., "Ransomware detection based on machine learning using memory features", Egyptian Informatics Journal, vol. 25, p. 100445, 2024. DOI: 10.1016/j.eij.2024.100445.
3. Canadian Institute for Cybersecurity, "Malware memory analysis". URL: https://www.unb.ca/cic/datasets/malmem-2022.html.
4. K. A. Dhanya, et al., "Detection of network attacks using machine learning and deep learning models", Procedia Computer Science, vol. 218, pp. 57–66, 2023. DOI: 10.1016/j.procs.2022.12.401.
5. A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Incorporated, 2022, 483 p.
6. H. Riggs, et al., "Impact, vulnerabilities, and mitigation strategies for cyber-secure critical infrastructure", MDPI. URL: https://www.mdpi.com/1424-8220/23/8/4060 (Last accessed: 29.10.2025).
7. S. Kumar, et al., "Malware classification using machine learning models", Procedia Computer Science, vol. 235, pp. 1419–1428, 2024. DOI: 10.1016/j.procs.2024.04.133.
8. Q. Li, et al., "MDGraph: a novel malware detection method based on memory dump and graph neural network", Expert Systems with Applications, p. 124776, 2024. DOI: 10.1016/j.eswa.2024.124776.Гайдук О., Звєрєв В. Аналіз кіберзагроз в умовах стрімкого розвитку інформаційних технологій. *Кібербезпека: освіта, наука, техніка*. 2024. Vol. 3, no. 23. P. 225–236. URL: https://csecurity.kubg.edu.ua/index.php/journal/article/view/552.

Lanin Yevhen *Master student of the Education and Research Institute of Computer Sciences and Artificial Intelligence, V.N. Karazin Kharkiv National University, 6 Svobody sq., Kharkiv, Ukraine, 61022*

Bakumenko Nina *Ph.D, associate professor of the Department of Computer Systems and Robotics, Education and Research Institute of Computer Sciences and Artificial Intelligence, V.N. Kharkiv National University, 6 Svobody sq., Kharkiv, Ukraine, 61022;*

Machine Learning Approaches to Malware Detection in RAM

Relevance. In the current context of constantly growing cyber threats, the problem of detecting malicious software that can operate covertly in RAM using fileless attack techniques has become particularly relevant. Traditional antivirus solutions based primarily on signature-based approaches prove ineffective against modern advanced persistent threats (APT) and new modified threats. This makes it essential to develop innovative approaches to malware detection based on behavioral pattern analysis in RAM using machine learning methods.

Goal. Development and testing of an automated malware detection system through RAM dump analysis using machine learning methods, as well as comparative evaluation of the effectiveness of various classification algorithms for multi-class threat type detection.

Research methods: comparative analysis of machine learning algorithms, static analysis of memory dumps, multi-class classification, experimental validation on the Obfuscated-MalMem2022 dataset containing over 58,000 records with 58 Windows process features. Models were evaluated using accuracy, precision, recall, and F1-score metrics with weighted averaging.

Results. A fully functional technological pipeline was created for automated processing and classification of RAM dumps, including modules for data preprocessing, feature engineering, machine learning, and results evaluation. A comparative analysis of 13 machine learning algorithms was conducted, including classical methods (Random Forest, Gradient Boosting, Decision Tree, k-NN, SVM) and neural network architectures (Wide & Deep Network, CNN). It was established that the Random Forest algorithm demonstrates the best results for the multi-class malware classification task with an accuracy of 85.49% and F1-score of 85.52% at a training time of 1.3 seconds. The developed system is implemented in Python using scikit-learn libraries (for classical ML models), TensorFlow/Keras (for neural networks), and pandas (for data processing).

Conclusions. The study confirmed the high effectiveness of classical machine learning methods, particularly ensemble algorithms, for malware detection in RAM dumps. The developed Random Forest-based model provides an optimal balance between classification accuracy (85.52% F1-score), training speed (1.3 s), and computational efficiency, demonstrating significant advantages over neural networks in this context. The developed system has high practical significance and can be integrated into forensic platforms, cybersecurity incident monitoring systems, and expert systems for automated threat detection and accelerated incident analysis. The research results confirm the feasibility of using machine learning methods to create defense systems against modern cyber threats that operate exclusively in RAM.

Keywords: *machine learning, memory dump analysis, malware detection, Random Forest, multi-class classification, pipeline, digital forensics, cybersecurity, Python. Keywords: machine learning, memory dump analysis, malware detection, Random Forest, classification, pipeline, forensics, Python.*

Надійшла у першій редакції 08.09.2025, в останній - 12.10.2025.
The first version has been received on 08.09.2025, the final version – on 12.10.2025.