УДК 330.46:004.49 DOI 10.33111/mise.102.15

Фролов Д.І.,
кафедра математичного моделювання та статистики
Київський національний економічний університет
імені Вадима Гетьмана

Матвійчук А.В., д.е.н., професор
кафедри математичного моделювання та статистики
Київський національний економічний університет
імені Вадима Гетьмана

Frolov D.I.,
Department of Mathematical Modeling and Statistics,
Kyiv National Economic University named after Vadym Hetman

Matviychuk A.V., Doctor of Economic Sciences,
Professor of Department of Mathematical Modeling and Statistics,
Kyiv National Economic University named after Vadym Hetman

КОНЦЕПТУАЛЬНИЙ ПІДХІД ДО РОЗПІЗНАВАННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ ТЕХНОЛОГІЙ МАШИННОГО НАВЧАННЯ

CONCEPTUAL APPROACH TO MALWARE RECOGNITION BASED ON MACHINE LEARNING TECHNIQUES

Анотація. *Україна протягом останніх років знаходиться в стані неоголошеної кібервійни. За останнє десятиліття аналіз та методи виявлення шкідливих програм пройшли значні зміни, що віддзеркалює відповідний розвиток різноманітних технік з розробки шкідливого програмного забезпечення. Системи виявлення шкідливих програм (MDS) — це перша лінія захисту від зловмисних атак. Тому для таких систем критично важливим є максимально точне й ефективне виявлення загроз. Зазвичай MDS використовують традиційні алгоритми машинного навчання, які потребують вибору та видобування ознак, що займає багато часу та може викликати помилки. В даній статті представлений концептуальний підхід до розпізнавання шкідливого програмного забезпечення з використанням новітніх моделей машинного навчання, розроблених для опрацювання зображень. Техніка конвертації файлів шкідливого програмного забезпечення в зображення у відтінках сірого відкриває можливості використання нейромережевих архітектур, розроблених для їх розпізнавання та класифікації. Згорткові нейронні мережі, а також найновітніші трансформери Swin 1-ої та 2-ої версій, разом із гібридною нейронною мережею CoAtNet, виступають перспективними кандидатами для проведення дослідження з визначення найбільш ефективної моделі для виявлення та класифікації шкідливого програмного забезпечення. Дана стаття може стати важливим підґрунтям для майбутніх дослідників у міждисциплінарній області використання методів та технік машинного (та глибинного) навчання в сфері кібербезпеки.*

Ключові слова: *кібербезпека, MDS, розпізнавання шкідливого програмного забезпечення, глибинне навчання, згорткова нейронна мережа, CoAtNet, Swin трансформер*

Abstract. *Ukraine has been in a state of undeclared cyberwar for several years. Over the past decade, malware analysis and detection methods have undergone significant changes, reflecting the corresponding development of various techniques for developing malicious software. Malware Detection Systems (MDS) are the first line of defense against malicious attacks. Therefore, it is critically important for such systems to accurately and effectively detect threats. Typically, MDS use traditional machine learning algorithms, which require feature selection and extraction, a process that is time-consuming and error-prone. This article presents a conceptual approach to recognizing malicious software using state-of-the-art machine learning models developed for image processing. The use of the technique of converting malicious software files into grayscale images opens up opportunities for the use of neural network architectures developed for image recognition and classification. Convolutional neural networks, as well as the latest Swin transformers of the 1st and 2nd versions, along with the CoAtNet hybrid neural network, are promising candidates for further research to determine the most effective model for recognition and classification of malicious software. This article could be an important milestone for future researchers in the interdisciplinary field of using machine (and deep) learning methods and techniques in cybersecurity.*

Keywords: *cybersecurity, MDS, malware detection, deep learning, convolutional neural network, CoAtNet, Swin transformer*

Постановка проблеми. У 2022 році в Україні почалася повномасштабна війна. Для того, щоб завдати найбільшої шкоди нашій країні, з боку агресора використовується різноманітна зброя. Особливе місце в цьому займають кібератаки. При цьому, в Україні фактично вже декілька років триває повномасштабна кібервійна.

Зростання кількості випадків атак зловмисного програмного забезпечення (написаного як досвідченими злочинцями, так і початківцями за допомогою загальнодоступних моделей машинного навчання, таких як ChatGPT), зниження вартості процесорної потужності та прогрес, досягнутий у цій галузі, сприяють появі нових досліджень та пропозицій щодо покращення аналізу шкідливого програмного забезпечення (ШПЗ). Так, протягом останніх років машинне, зокрема глибинне навчання, активно використовуються в світі як підхід до виявлення та аналізу шкідливих програм.

У зв’язку з цим, тематика дослідження теоретичних та технологічних аспектів інтелектуальних систем [1], використання методів та технік штучного інтелекту в сфері кібербезпеки, а також, як результат, проблематика вибору найбільш ефективних моделей машинного навчання для розпізнавання та класифікації шкідливого програмного забезпечення є особливо актуальною в цей час.

Системи виявлення шкідливих програм (Malware Detection System або MDS) — це перша лінія захисту від зловмисних атак [2]. Тому для таких систем критично важливим є максимально точне й ефективне виявлення загроз. Зазвичай MDS використовують традиційні алгоритми машинного навчання, які потребують вибору та видобування ознак, що займає багато часу та може викликати помилки. В даній статті представлений концептуальний підхід до розпізнавання шкідливого програмного забезпечення з використанням новітніх моделей машинного навчання, розроблених для опрацювання зображень.

На початкових етапах застосування машинного навчання до проблеми розпізнавання зловмисного програмного забезпечення застосовувалися алгоритми кластеризації. Однак ці підходи, здебільшого, не вирішують проблему, коли в наборі даних, що аналізується, присутній широкий спектр класів шкідливих програм.

Аналіз останніх досліджень і публікацій. За останні два десятиріччя проводилось багато досліджень у сфері виявлення та класифікації шкідливого програмного забезпечення із застосуванням методів та технік машинного навчання, деякі з котрих заслуговують на особливу увагу [3-10].

Отримані за результатами зазначених досліджень показники точності варіюються та, за певних умов, досягають 99 %. Так, в дослідженні [3] представлені результати застосування декількох модифікованих алгоритмів перцептрона для виявлення шкідливого програмного забезпечення — було досягнуто точності від 69,90 % до 96,18 %, при цьому більш точні моделі мали також більшу кількість помилкових спрацювань.

Разом з цим, в науковій роботі [4] досліджувався метод виявлення шкідливого програмного забезпечення, що базується на модифікованому алгоритмі випадкового лісу (random forest) в поєднанні з коефіцієнтом інформаційного приросту для кращого представлення ознак. Такий підхід забезпечив точність 97 % за низького рівня помилкових спрацювань, що робить його придатним для захисту корпоративних мереж.

В рамках комплексної роботи з дослідження різних методів машинного навчання для класифікації статичних характеристик 32-бітних зловмисних виконуваних файлів (portable executable або PE32) для Windows [6], автори видобули n-граму байтів, n-граму коду операції, виклики API та PE32 із виконуваного файлу Windows і застосували алгоритми на основі статистики (наївний Байєс), методу опорних векторів (support vector machine або SVM) і методу найближчих сусідів (k-nearest neighbors або k-NN).

Також було виявлено, що у більшості випадків класифікаційний алгоритм C4.5 та k-NN виявляють кращу ефективність, ніж інші методи, в той час як SVM та штучні нейронні мережі (artificial neural network або ANN) на деяких наборах ознак показали гарну продуктивність. З іншого боку, мережа Байєса та наївний Байєс мають погану ефективність порівняно з іншими методами машинного навчання. Отримані результати підтверджують тезу, що машинне навчання допомагає в аналізі шкідливих програм та може використовуватися в рамках діяльності з кіберзагрозами для автоматизації виявлення індикаторів компрометації. Незважаючи на те, що в результаті цього дослідження були отримані всі статистичні характеристики з виконуваного файлу, жоден із алгоритмів, які використовувались, не зміг досягнути рівня точності 96 %.

Результати новітнього дослідження з аналізу та виявлення шкідливих програм за допомогою алгоритмів машинного навчання [10] показали, що згорткові нейронні мережі (Convolutional Neural Networks або CNN) мають точність розпізнавання 98,76 %, опорно-векторні машини (SVM) — 96,41 %, а дерева рішень (decision trees або DT) — 99 %, таким чином, перевершуючи інші класифікатори за цим показником. Було проведено порівняння ефективності алгоритмів DT, CNN та SVM щодо виявлення шкідливого програмного забезпечення на тестовому наборі даних за умови низького рівня помилкових спрацювань (false positive rate або FPR). Помилкові спрацювання склали: 2,01 % для DT, 3,97 % для CNN та 4,63 % для SVM, відповідно. Таким чином, метод машинного навчання DT показав найкращі показники на наборі даних від Канадського інституту кібербезпеки. Представлені результати мають особливе значення в поточних умовах, коли програмне забезпечення стає все більш поширеним і складним для виявлення зловмисного коду.

Шкідливі програми також постійно еволюціонують і стають все складнішими. Внаслідок цього, виявлення та розуміння їх внутрішніх механізмів також становить задачу з високим рівнем складності. Крім того, зростаюча динаміка та різноманітність сучасних технологій зв'язку та обчислення дозволяють одній шкідливій програмі проявлятися в семантично та структурно різних формах. Як результат, процес її ефективного виявлення та класифікації потребує, відповідно, застосування найбільш сучасного інструментарію. Враховуючи ризики, які створює ШПЗ, в тому числі для критичної інфраструктури, точність його розпізнавання має прямувати до 100 %.

Разом з цим, ще у 2011 році в статті «Зображення шкідливого програмного забезпечення: візуалізація та автоматична класифікація» колективом авторів [11] було запропоновано підхід до оцінювання та класифікації ШПЗ на основі його візуалізації у вигляді зображень у відтінках сірого. За результатами зазначеної роботи був створений набір даних Malimg, який буде більш детально розглянутий далі у цій статті.

Метод Натараджа та співавторів [11] зосереджений на виконуваних файлах, що містять код шкідливого програмного забезпечення. Вони використовували необроблені двійкові дані з таких файлів в процесі побудови байтових зображень, зрештою генеруючи зображення в градаціях сірого для класифікації (рис. 1).

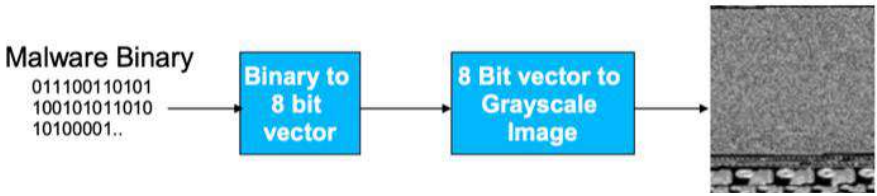

Рис. 1. Візуалізація зловмисного програмного забезпечення як зображення [11]

Процес отримання зображення з виконуваного файлу містить такі етапи:

- попередня обробка — очищення зразків ШПЗ для видалення непотрібної інформації, такої як заголовки та залишкові дані; як результат, зображення зосереджуються на основній функціональності такого програмного забезпечення;
- побудова байтових зображень — цей метод перетворює необроблені двійкові дані шкідливого програмного забезпечення (послідовність байтів) на двомірне зображення в градаціях сірого, коли кожному значенню байта у вихідних даних відповідає певна інтенсивність пікселя на зображенні (наприклад, нижчі значення байтів відображаються темнішими пікселями, ближче до чорного, а вищі значення — світлішими пікселями, ближче до білого);
- зміна розміру зображення — для налаштування зображення до розміру, придатного для подальшої обробки та аналізу, використовуються такі методи, як інтерполяція;
- стандартизація — додаткова стандартизація зображень для покращення ефективності алгоритмів класифікації можлива за

допомогою методів обробки зображень, таких як нормалізація або гістограмне вирівнювання.

Підхід [11] не обмежується певним типом формату виконуваного файлу. Конкретний тип файлу може змінюватися залежно від того, який зразок шкідливого програмного забезпечення аналізується.

Переформувавши масив 8-бітних елементів коду у матрицю та розглянувши його як зображення у градаціях сірого, авторам [11] вдалося виявити важливі візуальні кореляції в текстурі зображення шкідливих програм, що належать до того самого сімейства. Це може бути наслідком широкого розповсюдження методу створення нових варіантів шкідливого ПЗ через повторне використання коду в програмі зборки.

Виконуваний файл (наприклад, з розширенням .exe для Windows файлів) складається з трьох основних розділів:

– .data (initialized data) — розділ даних використовується для оголошення ініціалізованих даних або констант, які не змінюються під час виконання, таких як константні значення, імена файлів або розмір буфера;

– .bss (block started by symbol) використовується для оголошення змінних, наприклад, неініціалізованих даних;

– .text (code) — у текстовій частині розміщено фактичний машинний код програми.

Крім перелічених, можуть бути наявні такі додаткові розділи, як:

– .rsrc (resources) — містить усі ресурси для програми (наприклад, .ico, .rc, .dialog);

– .rdata (read-only data) — використовується для зберігання даних, які не належать до розділу .data або .bss (це також дані, які доступні лише для читання та містять літеральні рядки, константи та інформацію про каталог налагодження);

– .idata (import data) — містить інформацію про імпорт, наприклад, DLL програми, включно з каталогом імпорту та таблицею адрес імпорту;

– .edata (export data) — містить інформацію про імена та адреси експортованих функцій, а також каталог експорту, який надає адресу та зміщення функцій програмам, які імпортують DLL;

– .reloc (relocation) — містить таблицю базових переміщень (базове переміщення — це зміна інструкції або ініціалізованого значення змінної, яка потрібна, якщо завантажувач не може завантажити програму).

Графічне зображення формату виконуваного файлу представлене на рис. 2.

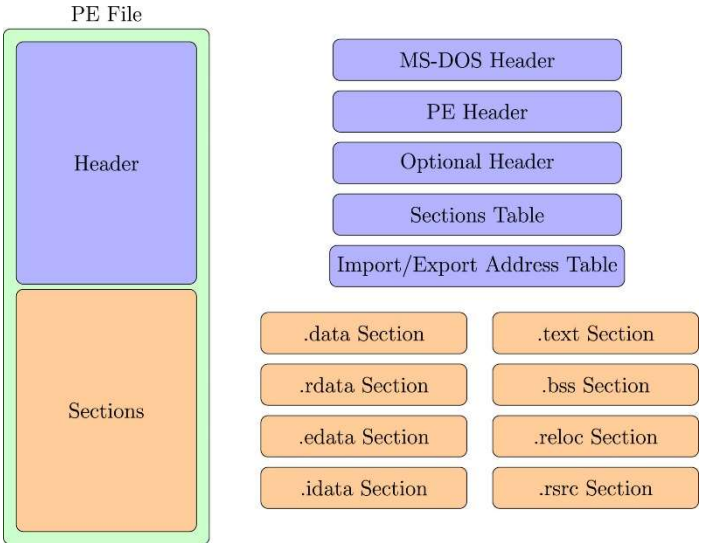

Рис. 2. Графічне представлення формату виконуваного файлу [12]

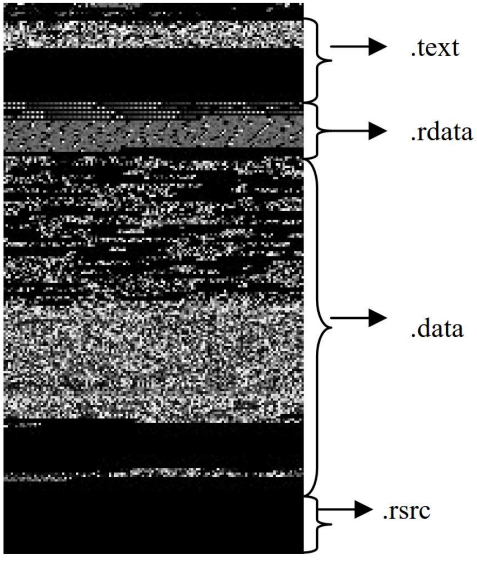

Рис. 3. Зображення бінарних фрагментів зразка ШПЗ (троян Dontovo.A) [11]

Інші розділи виконуваного файлу також можуть з'явитися в результаті використання поліморфних або метаморфічних методів, щоб приховати фактичний код ШПЗ. У деяких трансформаціях зловмисного програмного забезпечення можна побачити різні бінарні фрагменти, а секцію складання зловмисного програмного забезпечення можна ідентифікувати за різними текстурами на зображеннях.

Підхід, запропонований у [11], дозволяє зафіксувати незначні зміни, зберігаючи глобальну структуру, та допомагає виявити варіації ШПЗ.

На рис. 3 представлений зразок результату перетворення файлу на прикладі шкідливого програмного забезпечення троян Dontovo.A.

Набір даних Malimg. Оригінальний набір даних Malimg був створений в рамках проєкту з обробки сигналів для аналізу шкідливих програм на кафедрі електротехніки та комп'ютерної інженерії університету Каліфорнії (США). Метою цього проєкту було дослідження методів обробки сигналів та зображень для аналізу шкідливого програмного забезпечення [13].

Двійкові файли шкідливого програмного забезпечення були візуалізовані у вигляді зображень у сірій шкалі. При цьому спостерігалося, що для багатьох сімейств шкідливих програм зображення (які належать до одного сімейства), їх зображення виглядають дуже схожими за макетом і текстурою. Як зазначалось раніше за текстом, більшість нових шкідливих програм є модифікаціями вже існуючих. Таким чином, варіанти такого ШПЗ мають майже однаковий вміст.

Протягом реалізації вказаного проєкту, колективом дослідників було зроблено два основних спостереження [13]:

1. Існує візуальна схожість у варіантах шкідливого програмного забезпечення в межах сімейств.

2. Існує візуальна несхожість між варіантами шкідливого програмного забезпечення різних сімейств.

Для подальшої роботи були використані ці візуальні подібності та відмінності й запропоновано функції, засновані на схожості зображень, для вирішення проблем класифікації, виявлення, пошуку шкідливого програмного забезпечення та інших завдань.

При цьому, набір даних Malimg використовується вже більше десяти років в різних роботах із розпізнавання шкідливого програмного забезпечення і, де-факто, став бенчмаркінг стандартом для дослідження ефективності використання різних методів і моделей машинного навчання.

Відповідно, у нашому дослідженні з розпізнавання (класифікації) шкідливого програмного забезпечення також використаємо загальнодоступний набір даних Malimg, який розміщений на інтернет платформі Kaggle [14].

Розподіл зображень з даного набору даних, що містить 25 сімейств (класів) шкідливого програмного забезпечення, представлений на рис. 4.

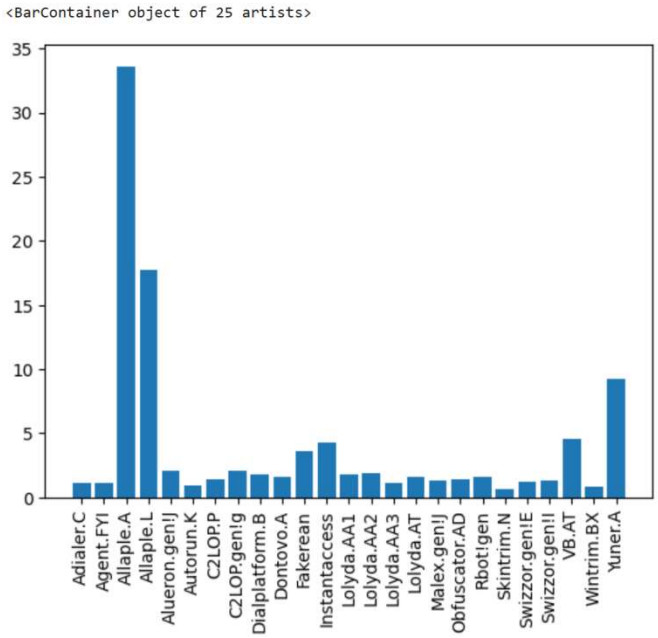

Рис. 4. 25 класів ШПЗ в наборі даних Malimg

Даний набір містить 9340 зображень байт-плотів шкідливих програм з 25 різних сімейств, а саме: Adialer.C, Agent.FYI, Allaple.A, Allaple.L, Alueron.gen!J, Autorun.K, C2LOP.P, C2LOP.gen!g, Dialplatform.B, Dontovo.A, Fakerean, Instantaccess, Lolyda.AA1, Lolyda.AA2, Lolyda.AA3, Lolyda.AT, Malex.gen!J, Obfuscator.AD, Rbot!gen, Skintrim.N, Swizzor.gen!E, Swizzor.gen!I, VB.AT, Wintrim.BX, Yuner.A.

На приведених нижче зображеннях шкідливого програмного забезпечення (рис. 5) візуально можна прослідкувати незначні модифікації. В той саме час, у зразків, що належать до одного сі-

мейства, загальна структура зображення зберігається. Однак вони візуально відрізняються від зразків зловмисних програм інших сімейств.

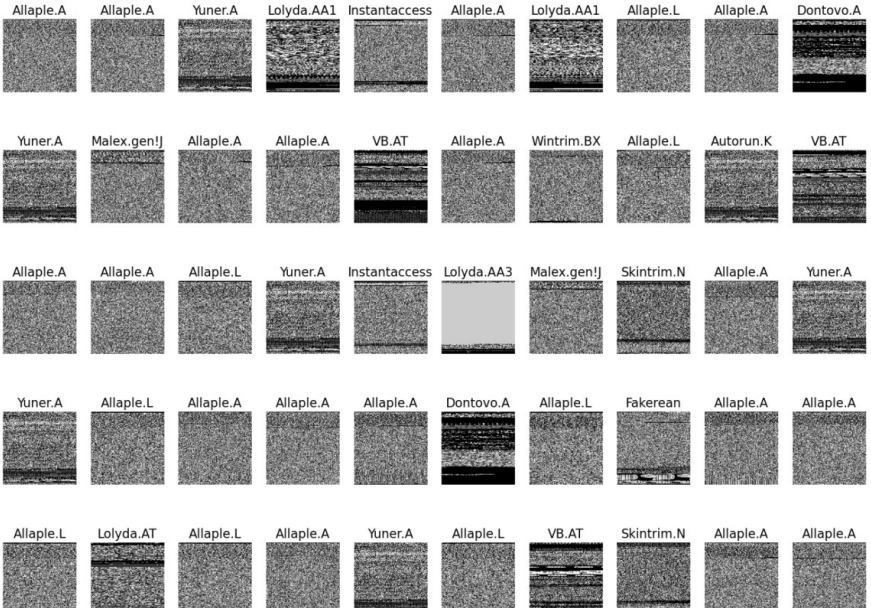

Рис. 5. Зразки зображень шкідливого програмного забезпечення різних класів у наборі даних Malimg

Важливо зазначити, що набір даних Malimg має певні обмеження, такі як нерівномірність розподілу, що потенційно впливає на навчання моделей машинного навчання. При цьому, набір даних Malimg має визнану цінність для виконання завдань з розпізнавання ШПЗ при порівнянні моделей машинного навчання, розроблених для опрацювання зображень.

Використання інтелектуальних методів аналізу даних. За останні десятиліття сфера машинного навчання пережила прорив у вирішенні багатьох завдань.

Для обчислення особливостей текстури в зображеннях зловмисного програмного забезпечення успішно застосовувався алгоритм GIST (Global Image Structure Tensor) [15], який використовує вейвлет-розкладання для видобування ознак із глобальної структури зображення. Отримані елементи використовуються для порівняння з раніше ідентифікованими шкідливими шабло-

нами. Хоча таке зображення, що відтворене на основі функцій глобальної структури, є вразливим до структурних змін, кіберзлочинці, які знають про таку техніку розпізнавання, можуть уникнути виявлення, перемістивши розділи коду або додавши фіктивні дані (наприклад, через обфускацію).

У 2015 році в рамках задачі класифікації зображень ImageNet було запропоновано використання функції активації PReLU (Parametric Rectified Linear Unit), яка за результатами дослідження перевершила людську продуктивність [16].

Винахід згорткових нейронних мереж (CNN) став важливою віхою у розвитку розпізнавання зображень. CNN є формою штучної нейронної мережі, яка імітує спосіб опрацювання зображень зоровою корою головного мозку. Так, було запропоновано підхід до протидії контрзаходам, які використовують кіберзлочинці, через використання згорткових нейронних мереж для вилучення локальних та інваріантних характеристик із зображення, а також знаходження шаблонів незалежно від їхнього розташування у файлі [17].

Так, використання CNN дозволяє виявляти шаблони відомого шкідливого програмного забезпечення, присутнього на зображенні. Далі за текстом представлена структура згорткової нейронної мережі Гіберта (Gibert's CNN), яка використовується для класифікації зловмисного програмного забезпечення, представленого у вигляді зображень у відтінках сірого (див. рис. 6).

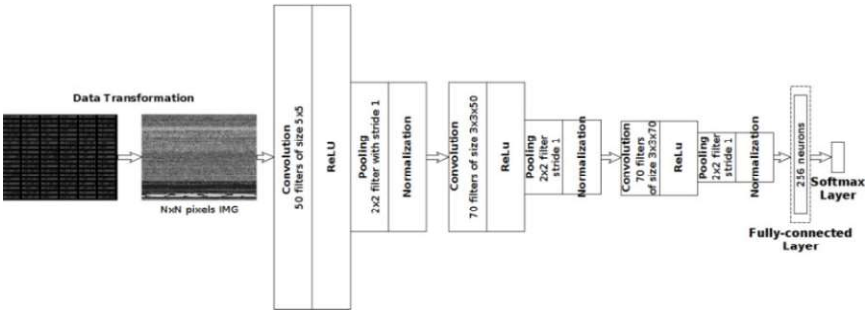

Рис. 6. Згорткова нейронна мережа Гіберта [17]

Як зазначалось раніше, за допомогою аналізу зображень у градаціях сірого, отриманих у результаті перетворення бінарного коду відомих зразків зловмисного програмного забезпечення, було зроблено висновок, що зображення з одного сімейства зловмисного програмного забезпечення схожі між собою [11]. З вве-

денням підходу щодо використання згорткових нейронних мереж можна було видобувати локальні та інваріантні особливості із зображення, знаходячи шаблони незалежно від їхнього положення. Таким чином, нейронна мережа давала змогу виявляти зразки відомого зловмисного програмного забезпечення на зображенні.

Хоча запропоноване рішення має ряд переваг, які дозволяють ефективно виявляти шкідливі програми, ця стратегія має проблеми з певними зразками, які були стиснуті або зашифровані, а також з тими, які можуть мати зовсім іншу загальну структуру.

Зазначені обставини обумовлюють подальший пошук та валідацію більш новітніх моделей машинного навчання з метою визначення найбільш ефективного рішення задачі розпізнавання та класифікації шкідливого програмного забезпечення, представленого як зображення.

Альтернативна до CNN архітектура штучної нейронної мережі, названа трансформерами зору (vision transformer або ViT), була представлена у 2020 році [18]. Це ознаменувало собою появу нового сімейства штучних нейронних мереж. Архітектура трансформеру створена для вивчення контексту і відстеження взаємозв'язків у послідовних даних. Вперше трансформери були використані з великим успіхом у програмах машинного навчання, пов'язаних з мовою та перекладом [19]. Однак, виникнення даної технології викликало значний інтерес до неї в задачах комп'ютерного зору (computer vision). У 2021 році було представлено, що трансформери, які застосовуються в задачах комп'ютерного зору, за наявності достатньої кількості навчальних даних демонструють ефективність, порівнянну з CNN, а в деяких випадках навіть вищу за них [20]. У зв'язку з цим, більшість програм, що використовують трансформери, навчаються на великих масивах даних.

Використання традиційних трансформерів зору для обробки зображень має квадратичну складність обчислень відносно розміру зображення через глобальні обчислення самоуваги [21]. Так, глобальний механізм самоуваги на зображеннях з використанням згорткових нейронних мереж детально демонструється в статті [22].

Одним із підходів до більш ефективного вилучення ознак із зображень став розроблений у 2021 р. Swin трансформер [21, 23].

Swin трансформер — це трансформерна модель глибинного навчання з найбільшою на даний момент продуктивністю в задачах зору. Swin трансформер має вищу точність порівняно з трансформером зору [18], який йому передує. Завдяки цим властивос-

тям Swin трансформери використовуються як основа в багатьох моделях розпізнавання зображень.

Прикладом такого використання є модель CoAtNet, яка була представлена у дослідженні «CoAtNet: Поєднання згортки та уваги для всіх розмірів даних» [24] у 2021 р. У наведеній роботі колектив авторів показав рішення проблеми гібридизації згортки та уваги з точки зору двох фундаментальних аспектів машинного навчання — узагальнення та потужності моделі. Назва CoAtNet походить від об'єднання слів Convolution та self-Attention. Ця гібридна модель, яка поєднала ознаки згорткової мережі та трансформеру, була виділена у нове сімейство моделей. Як заявлено авторами, CoAtNet має сильні сторони як згорткових мереж, так і трансформерів.

Разом з цим, у 2022 році була опублікована наукова робота, яка представила другу версію Swin трансформеру (Swin v.2) [25]. Першу версію було масштабовано до 3 мільярдів параметрів, що є найбільшою та найефективнішою моделлю щільного бачення станом на 2022 рік. Крім того, адаптована версія використовує в 40 разів менше мічених даних і потребує в 40 разів менше часу на навчання, ніж попередні моделі. Як результат, її використання в задачах розпізнавання зображень має більший потенціал та потенційно більшу ефективність, ніж у першої версії Swin трансформеру.

Для забезпечення кращої ефективності навчання пропонується до застосування технологію передавального навчання (Transfer Learning або TL). TL — це техніка машинного навчання, яка використовує знання, отримані з попередньо навчених моделей для покращення продуктивності моделей, що навчаються на інших, але пов'язаних завданнях. Разом з цим, передавальне навчання визначають як повторне використання моделей (навчених на попередньо існуючих наборах даних) для вирішення нових актуальних цільових завдань. TL є інструментом оптимізації, який підвищує продуктивність моделювання. До його ключових характеристик відносяться повторне використання знань, отриманих з попередніх завдань, що призводить до покращення продуктивності моделей машинного навчання та, в результаті, до економії часу та ресурсів, потрібних для навчання таких моделей [26]. TL може бути корисним у багатьох ситуаціях, коли доступні дані для попереднього навчання та нове завдання подібне до попереднього.

В підході до розпізнавання шкідливого програмного забезпечення, який пропонується в даній роботі, набір даних Malimg ви-

користовується для попереднього навчання обраних моделей з їх подальшим навчанням на іншому наборі зображень в градаціях сірого, отриманих з виконуваних файлів, які потенційно містять ШПЗ (з метою визначення та класифікації можливих загроз). Таким чином, за результатами проведеного мета-аналізу наукових публікацій з виявлення та класифікації ШПЗ, концептуальний підхід до розпізнавання шкідливого програмного забезпечення (представленого як зображення у відтінках сірого) з використанням найсучасніших моделей машинного навчання, побудованих для опрацювання зображень, можна формалізувати в двох складових: практичній та дослідницькій (див. рис. 7).

Рис. 7. Концептуальний підхід до розпізнавання шкідливого програмного забезпечення на основі технологій машинного навчання

Наведемо деталізацію етапів концептуального підходу, представленого на рис. 7, які відносяться до практичної та дослідницької складових:

1. **Отримання зображень з виконуваних файлів та підготовка набору даних (практична складова)** — формування набору зображень в градаціях сірого з виконуваних файлів, які потенційно містять ШПЗ (в тому числі: перетворення зразків шкідливого програмного забезпечення на зображення в градаціях сірого; зменшення розміру зображень для подальшого підвищення ефективності їх опрацювання; розділення отриманого первинного набору даних на тренувальну, валідаційну та тестову вибірки).

2. **Специфікація моделей (дослідницька складова)** — вибір найбільш адекватного поставленій задачі та наявному набору даних математичного інструментарію (моделей глибинного навчання) для розпізнавання/класифікації ШПЗ на основі візуальних патернів.

3. **Налаштування моделей на наборі даних Malimg (дослідницька складова)** — попереднє навчання обраних моделей та обрання найбільш ефективної моделі (моделей) для подальшого використання в реальному середовищі (в тому числі: оптимізація обраних моделей на тренувальному наборі даних; оцінювання ефективності та вдосконалення моделей на тестовому наборі даних; верифікація моделей на валідаційному наборі даних; порівняння адекватності побудованих моделей).

4. **Підготовка моделей (практична складова)** — додаткове навчання моделей на наборі даних з реального середовища (а саме: оптимізація обраних моделей на тренувальному наборі даних; оцінювання та вдосконалення моделей на тестовому наборі даних; оцінка точності та ефективності моделей на валідаційному наборі даних; вибір найкращої моделі нейромережевої архітектури).

5. **Вдосконалення моделей (практична складова)** — використання методів регуляризації, додаткові експерименти з наборами даних та моделями (за можливості розширення/зміна тренувального набору даних та проведення експериментів з різними архітектурами та параметрами моделей).

6. **Розгортання та удосконалення в реальному середовищі (практична складова)** — подальший розвиток рішення з виявлення та класифікації ШПЗ на основі візуальних патернів (в тому числі: використання обраної моделі (моделей) для виявлення та класифікації шкідливого програмного забезпечення; постійне удосконалення системи через використання нових даних та най-

сучасніших моделей нейромережевих архітектур, розроблених для опрацювання зображень).

Висновки та перспективи подальшого дослідження. В результаті проведеного мета-аналізу наукових публікацій за останні роки виявлено, що для вирішення завдань класифікації зображень високу ефективність демонструє згорткова нейронна мережа, а також найбільш сучасні моделі машинного навчання, такі як:

– трансформер Swin v.1 (2021 р.) [21, 23];

– гібридна згорткова нейронна мережа CoAtNet (2021 р.) [24];

– трансформер Swin v.2 (2022 р.) [25].

У зв'язку з цим, використання техніки конвертації файлів шкідливого програмного забезпечення в зображення у відтінках сірого відкриває можливості використання зазначених архітектур нейронних мереж для розпізнавання та класифікації такого ШПЗ. Найбільш відомим набором даних з таких зображень шкідливих програм є Malimg, що обумовлює його використання в подальшому дослідженні.

На підставі хронологічної послідовності зазначених наукових публікацій, а також заявлених дослідниками характеристик нейромережевої архітектури Swin трансформерів другої версії [25], логічно висунути гіпотезу про те, що модель машинного навчання, побудована на базі цього типу нейронної мережі, буде досягати вищої точності в розпізнаванні шкідливого програмного забезпечення порівняно з іншими архітектурами (згорткової нейронної мережі, трансформеру Swin першої версії, а також гібридної нейромережі CoAtNet).

Враховуючи зазначене, в даній роботі для підтвердження цієї гіпотези щодо ефективності класифікації шкідливого програмного забезпечення рекомендується проведення подальшого дослідження (як дослідницької складової представленого концептуального підходу), яке включатиме імплементацію зазначених чотирьох нейромережевих архітектур. Також для вирішення цього завдання доцільним є використання набору даних Malimg.

Проведене в даній статті дослідження дозволило сформулювати концептуальний підхід до розпізнавання шкідливого програмного забезпечення з використанням найсучасніших нейромережевих архітектур, розроблених для опрацювання зображень. Згідно даного підходу згорткові нейронні мережі, а також найбільш новітні архітектури Swin трансформерів 1-ої та 2-ої версій, разом із гібридною нейронною мережею CoAtNet, ви-

ступають перспективними кандидатами для проведення подальшого дослідження з визначення найбільш ефективної моделі машинного навчання для класифікації шкідливого програмного забезпечення.

Дана стаття може стати важливим підґрунтям для дослідників у міждисциплінарній області використання методів та технік машинного (зокрема глибинного) навчання в сфері кібербезпеки.

Бібліографічні посилання

1. Frolov, D., Radziewicz, W., Saienko, V., Kuchuk, N., Mozhaiev, M., Gnusov, Y., & Onishchenko, Y. (2021). Theoretical and Technological Aspects of Intelligent Systems: Problems of Artificial Intelligence. *International Journal of Computer Science and Network Security, 21*(5), 35-38. https://doi.org/10.22937/IJCSNS.2021.21.5.6

2. He, K., & Kim, D.-S. (2019). Malware detection with malware images using deep learning techniques. In *Proceedings of the 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE)* (pp. 95-102). IEEE. https://doi.org/10.1109/TrustCom/BigDataSE.2019.00022

3. Gavriluţ, D., Cimpoeşu, M., Anton, D., & Ciortuz, L. (2009). Malware detection using machine learning. In *Proceedings of the 2009 International Multiconference on Computer Science and Information Technology* (pp. 735-741). IEEE. https://doi.org/10.1109/IMCSIT.2009.5352759

4. Singhal, P., & Raul, N. (2012). Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks. *International Journal of Network Security & Its Applications, 4*(1), 61-71. https://doi.org/10.5121/ijnsa.2012.4106

5. Arp, D., Spreitzenbarth, M., Hübner, M., Gascon, H., & Rieck, K. (2014). DREBIN: Effective and explainable detection of Android malware in your pocket. In *Proceedings of the Network and Distributed System Security Symposium* (Article 23247). The Internet Society. https://doi.org/10.14722/ndss.2014.23247

6. Shalaginov, A., Banin, S., Dehghantanha, A., & Franke, K. (2018). Machine Learning Aided Static Malware Analysis: A Survey and Tutorial. In A. Dehghantanha, M. Conti, T. Dargahi (Eds.), *Advances in Information Security: Vol. 70. Cyber Threat Intelligence* (pp. 7–45). Springer. https://doi.org/10.1007/978-3-319-73951-9_2

7. Zhang, X., Wu, K., Chen, Z., & Zhang, C. (2021). MalCaps: A Capsule Network Based Model for the Malware Classification. *Processes, 9*(6), Article 929. https://doi.org/10.3390/pr9060929

8. Hemalatha, J., Roseline, S. A., Geetha, S., Kadry, S., & Damaševičius, R. (2021). An efficient DenseNet-based deep learning model for malware detection. *Entropy, 23*(3), Article 344. https://doi.org/10.3390/e23030344

9. Lin, W.-C., & Yeh, Y.-R. (2022). Efficient Malware Classification by Binary Sequences with One-Dimensional Convolutional Neural Networks. *Mathematics, 10*(4), Article 608. https://doi.org/10.3390/math10040608

10. Akhtar, M. S., & Feng, T. (2022). *Malware Analysis and Detection Using Machine Learning Algorithms*. *Symmetry, 14*(11), Article 2304. https://doi.org/10.3390/sym14112304

11. Nataraj, L., Karthikeyan, S., Jacob, G., & Manjunath, B.S. (2011). Malware images: Visualization and automatic classification. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security* (Article 4). ACM. https://doi.org/10.1145/2016904.2016908

12. Gibert, D., Mateu, C., & Planes, J. (2020). The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications, 153*, Article 102526. https://doi.org/10.1016/j.jnca.2019.102526

13. Department of Electrical and Computer Engineering, University of California. (n.d.). Signal Processing for Malware Analysis. Retrieved from https://vision.ece.ucsb.edu/research/signal-processing-malware-analysis

14. Sunkari, M. (2022). *Malimg_dataset9010* [Data set]. Kaggle. Retrieved from https://www.kaggle.com/datasets/manaswinisunkari/malimg-dataset9010

15. Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision, 42*(3), 145-175. https://doi.org/10.1023/A:1011139631724

16. He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification*. arXiv. https://doi.org/10.48550/arXiv.1502.01852

17. Gibert, D., Mateu, C., Planes, J., & Vicens, R. (2019). Using convolutional neural networks for classification of malware represented as images. *Journal of Computer Virology and Hacking Techniques, 15*(1), 15-28. https://doi.org/10.1007/s11416-018-0323-0

18. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). *An image is worth 16x16 words: Transformers for image recognition at scale*. arXiv. https://doi.org/10.48550/arXiv.2010.11929

19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention is all you need*. arXiv. https://doi.org/10.48550/arXiv.1706.03762

20. Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., Tian, J., Zhang, Y., Shi, Z., Fan, J., & He, Z. (2021). *A survey of visual transformers*. arXiv. https://doi.org/10.48550/arXiv.2111.06091

21. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). *Swin transformer: Hierarchical vision transformer using shifted windows*. arXiv. https://doi.org/10.48550/arXiv.2103.14030

22. Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2018). *Self-attention generative adversarial networks*. *arXiv.* https://doi.org/10.48550/arXiv.1805.08318

23. Loy, J. (2022, May 20). A Comprehensive Guide to Microsoft's Swin Transformer. In-depth Explanation and Animations. *Towards Data Science*. https://towardsdatascience.com/a-comprehensive-guide-to-swin-transformer-64965f89d14c

24. Dai, Z., Liu, H., Le, Q.V., & Tan, M. (2021). *CoAtNet: Marrying Convolution and Attention for All Data Sizes*. ArXiv. https://doi.org/10.48550/arXiv.2106.04803

25. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., Wei, F., & Guo, B. (2022). Swin Transformer V2: Scaling Up Capacity and Resolution. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11999-12009). IEEE. https://doi.org/10.1109/CVPR52688.2022.01170

26. Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M. A. (2022). Transfer learning: A friendly introduction. *Journal of Big Data*, *9*, Article 102. https://doi.org/10.1186/s40537-022-00652-w