DOI 10.31891/2307-5732-2025-347-80
УДК 004.056:004.852:004.75

ДЕНИСЮК ДМИТРО
Хмельницький національний університет
https://orcid.org/0000-0002-7345-8341
e-mail: denysiuk@khmnu.edu.ua
СОРОЧИНСЬКИЙ ОЛЕКСАНДР
Хмельницький національний університет
https://orcid.org/0009-0003-7966-4861
e-mail: sorochinskiy159@gmail.com
ГНАТЧУК ЄЛИЗАВЕТА
Хмельницький національний університет
https://orcid.org/0000-0003-2989-3183
e-mail: liza_veta@ukr.net
ДРОЗД АНДРІЙ
Хмельницький національний університет
https://orcid.org/0009-0008-1049-1911
e-mail: andriydrozdit@gmail.com

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ВИЯВЛЕННЯ ЗЛОВМИСНИХ КОДІВ В ІНФОРМАЦІЙНИХ СИСТЕМАХ НА ОСНОВІ АНАЛІЗУ ПАРАЛЕЛЬНИХ ПРОЦЕСІВ

У статті запропоновано нову методику виявлення зловмисних команд, прихованих у графічних файлах за допомогою стеганографічних методів. Запропонований підхід поєднує статичний та динамічний аналіз, що дозволяє ідентифікувати як структурні аномалії у графічних файлах, так і специфічні поведінкові патерни підозрілих процесів. На етапі статичного аналізу графічні файли перевіряються на наявність ознак відомих стеганографічних технік, зокрема артефактів у структурі пікселів та аномальних шумових патернів. У разі виявлення підозрілих ознак процес, що працює з таким файлом, переміщується до ізольованого віртуального середовища для проведення безпечного поведінкового аналізу.

Для динамічного аналізу використано нейронну мережу типу Long Short-Term Memory (LSTM), яка аналізує часові послідовності параметрів, включаючи мережевий трафік, операції з файлами, споживання ресурсів та активність в оперативній пам'яті.

Експериментальні дослідження підтвердили високу ефективність запропонованої методики. Комбінований підхід забезпечив точність на рівні 98%, повноту — 96% та частоту хибних спрацьовувань (FPR) — 3%, що значно перевищує показники традиційних інструментів, таких як StegExpose та LSB-Steganalysis Toolkit. Зокрема, статичний аналіз окремо продемонстрував точність 89%, а динамічний аналіз на основі LSTM досяг точності 94%.

Запропонована методика забезпечує ефективне виявлення складних стеганографічних атак, однак її продуктивність значною мірою залежить від якості навчальної вибірки та потребує значних обчислювальних ресурсів. Подальші дослідження будуть спрямовані на розширення можливостей методики для аналізу інших мультимедійних форматів, а також на вдосконалення механізмів навчання нейронної мережі для підвищення її адаптивності до нових типів стеганографічних атак.

Ключові слова: стеганографія; зловмисні команди; графічні файли; статичний аналіз; динамічний аналіз; LSTM-мережа; виявлення загроз; інформаційна безпека; приховані канали зв'язку; аналіз поведінки.

DENYSIUK DMYTRO
SOROCHYNSKYI OLEKSANDR
HNATCHUK YELYZAVETA
DROZD ANDRIY
Khmelnytskyi National University

INFORMATION TECHNOLOGY FOR DETECTING MALICIOUS CODES IN INFORMATION SYSTEMS BASED ON PARALLEL PROCESS ANALYSIS

The paper proposes a novel methodology for detecting malicious commands concealed in graphic files using steganographic techniques. The proposed approach combines static and dynamic analysis, enabling the identification of both structural anomalies in graphic files and specific behavioral patterns of suspicious processes. During the static analysis phase, graphic files are examined for signs of known steganographic techniques, including pixel structure artifacts and abnormal noise patterns. If suspicious characteristics are detected, the process interacting with such a file is transferred to an isolated virtual environment for safe behavioral analysis.

For dynamic analysis, a Long Short-Term Memory (LSTM) neural network is employed, which analyzes time sequences of parameters, including network traffic, file operations, resource consumption, and memory activity.

Experimental studies confirmed the high efficiency of the proposed methodology. The combined approach achieved an accuracy of 98%, recall of 96%, and a false positive rate (FPR) of 3%, significantly outperforming traditional tools such as StegExpose and LSB-Steganalysis Toolkit. Specifically, static analysis alone demonstrated an accuracy of 89%, while LSTM-based dynamic analysis achieved an accuracy of 94%.

The proposed methodology effectively detects complex steganographic attacks; however, its performance largely depends on the quality of the training dataset and requires significant computational resources. Future research will focus on expanding the methodology's capabilities to analyze other multimedia formats and improving the neural network training mechanisms to enhance its adaptability to new types of steganographic attacks.

Keywords: steganography; malicious commands; graphic files; static analysis; dynamic analysis; LSTM network; threat detectio; information securit; covert communication channel; behavior analysis.

Постановка проблеми у загальному вигляді

Проблема виявлення зловмисного програмного забезпечення (ЗПЗ) набула особливої гостроти у зв'язку зі зростанням кількості кіберзагроз упродовж останніх років. За даними на 2021 рік, світові фінансові втрати від кібератак сягнули трильйонів доларів, а щорічне зростання кількості шкідливих програм перевищує 20%. Сучасні ЗПЗ активно використовують методи обфускації та ухилення від виявлення, що значно ускладнює їхню ідентифікацію на ранніх стадіях. У відповідь на ці загрози дослідники розробляють різноманітні підходи до аналізу та детекції шкідливого коду [1]. Зокрема, активно застосовуються статичний аналіз[2], що виключає виконання коду, динамічний аналіз[3], який базується на моніторингу поведінки під час виконання, а також гібридні методи, що поєднують обидва підходи. Методи виявлення також розрізняються за принципом роботи: сигнатурні (пошук відомих шаблонів), поведінкові та евристичні підходи[4].

Однак зростання складності та витонченості методів приховування ЗПЗ, зокрема із використанням стеганографії[5] для прихованої передачі зловмисних команд через графічні файли, створює серйозні виклики для наявних систем детекції. Традиційні методи[6] виявлення демонструють обмежену ефективність у виявленні таких загроз, оскільки стеганографічні техніки дозволяють приховувати шкідливий вміст у графічних даних із мінімальними візуальними чи структурними змінами. У зв'язку з цим постає необхідність розробки нових методів виявлення, що враховують специфіку стеганографічних атак, а також здатні поєднувати аналіз структури файлів, поведінкові характеристики системи та мережеву активність. Розробка таких методів, особливо із використанням алгоритмів машинного навчання, має потенціал значно підвищити ефективність виявлення сучасних загроз і забезпечити надійний захист інформаційних систем.

Аналіз досліджень та публікацій

Аналіз сучасних досліджень та наукових публікацій є важливим етапом у розробці нових методів виявлення зловмисного програмного забезпечення[7]. Особлива увага приділяється поведінковим підходам, що демонструють високу ефективність у виявленні складних та раніше невідомих загроз. У рамках даного дослідження було проведено аналіз основних підходів до детекції шкідливого ПЗ, зокрема поведінкових методів виявлення, гібридних підходів, що поєднують елементи статичного та динамічного аналізу, а також методів машинного навчання, які забезпечують високу точність і швидкість ідентифікації загроз на основі великих обсягів даних.

Поведінкові методи виявлення

Поведінкові методи виявлення зловмисного програмного забезпечення базуються на динамічному аналізі, що передбачає запуск підозрілого файлу в ізольованому середовищі, такому як пісочниця, з метою спостереження за його діями. Під час цього процесу фіксуються ключові системні події, зокрема виклики API та системних функцій, операції з файлами та реєстром, мережева активність, а також використання процесів і пам'яті. Подібні ознаки відображають реальну поведінку програми під час її виконання. Наприклад, у середовищі Windows більшість операцій програмного забезпечення реалізується через виклики API[8], і саме їхній аналіз дозволяє виявити характерні послідовності, що є типовими для шкідливих програм і рідко зустрічаються в легітимному програмному забезпеченні. Виявлення таких патернівдозволяє ідентифікувати потенційно небезпечну активність.

Перевагою поведінкових методів є їхня стійкість до технік статичного маскування коду. Оскільки аналіз фокусується на спостереженні за реальними діями програмного забезпечення, використання зловмисниками методів шифрування чи упаковування виконуваного файлу не перешкоджає виявленню загрози. Навіть у разі застосування поліморфних чи обфускованих[9] механізмів, шкідлива активність таких програм стає помітною під час виконання. На відміну від статичних сигнатурних методів, поведінковий підхід дозволяє виявляти нові, раніше невідомі зразки шкідливого ПЗ, включно з атаками типу «нульового дня», базуючись на характерній активності програмного забезпечення. Такий підхід вважається більш надійним критерієм виявлення загроз порівняно зі статичними методами, що обмежуються пошуком збігів із відомими сигнатурними шаблонами. Дослідження підтверджують, що поведінковий аналіз забезпечує вищу точність і ефективність у виявленні нових загроз порівняно з виключно статичними підходами.

Сучасні системи поведінкового аналізу[10] активно інтегрують алгоритми машинного навчання для класифікації на основі динамічних ознак. Поширеними підходами є моделі, що інтерпретують послідовності системних викликів як часові ряди або текстові дані. Зокрема, застосування рекурентних нейронних мереж (RNN) та архітектури Long Short-Term Memory дозволяє ефективно обробляти послідовності API-викликів. Крім того, перспективними є одноосні згорткові нейронні мережі (1D-CNN), здатні виявляти характерні підпослідовності у потоці викликів. Один із підходів, запропонований у 2024 році, поєднує частотний аналіз послідовностей API з глибокою нейронною мережею TextCNN. На початковому етапі алгоритм PrefixSpan[11] ідентифікує часті шкідливі патерни викликів, після чого решта послідовностей, що не відповідають відомим шаблонам, класифікуються за допомогою моделі CNN. Така дворівнева архітектура продемонструвала високу ефективність, досягнувши близько 92,8% точності класифікації, що перевершило класичні алгоритми машинного навчання, такі як логістична регресія та k-NN, за всіма ключовими метриками.

Інший перспективний підхід, розроблений у 2023 році, доповнює традиційний аналіз послідовності викликів API інформацією про їхні параметри. Такий підхід формує так звані «семантичні ланцюжки» поведінки, які містять додаткові відомості про параметри викликів, що підвищує стійкість системи до модифікацій середовища та змін у викликах API. Експериментальні дослідження підтвердили, що ця методика суттєво покращує показники виявлення на тестових наборах, зокрема підвищує рівень стійкості до невідомих параметрів API та забезпечує вищий загальний рівень детекції. У тестуванні на наборі даних Datacon2019 запропонований підхід продемонстрував перевагу над базовими методами, демонструючи особливо високу ефективність у виявленні нових і модифікованих зразків шкідливого ПЗ[12].

Попри значні переваги, поведінкові методи мають певні недоліки, що обмежують їхнє застосування. Зокрема, вони є ресурсомісткими та повільнішими порівняно зі статичними методами, оскільки потребують запуску підозрілого коду та спостереження за його діями. Затримка в аналізі може стати критичною у випадках, де необхідне виявлення загроз у режимі реального часу. Крім того, сучасні зловмисні програми часто включають механізми ухилення від поведінкової детекції. Зокрема, близько 98% сучасного шкідливого ПЗ використовують техніки для виявлення ознак віртуальних машин або пісочниць, що дозволяє їм змінювати поведінку або переходити у «сплячий режим», уникаючи виконання зловмисних дій під час спостереження. Така тактика дозволяє обійти поведінкові системи виявлення. Ще одним обмеженням є неповне охоплення можливих сценаріїв виконання. Якщо активність шкідливого коду виявляється лише за специфічних умов, що не виникають у процесі тестового прогону, його небезпечна поведінка може залишитися непоміченою.

Гібридні методи виявлення

Гібридні методи виявлення поєднують у собі елементи статичного та динамічного аналізу, що дозволяє суттєво підвищити ефективність і точність ідентифікації шкідливого програмного забезпечення. Основна концепція такого підходу базується на взаємному доповненні цих двох методів: недоліки одного компенсуються перевагами іншого. Зокрема, статичний аналіз забезпечує оперативну перевірку та ефективно ідентифікує відомі загрози без необхідності виконання коду, тоді як динамічний аналіз дозволяє виявляти нові зразки, аналізуючи їхню поведінку під час виконання. Комбінація цих підходів забезпечує вищу точність детекції порівняно з використанням кожного з методів окремо.

Існує декілька підходів до реалізації гібридного аналізу[13]. Один із варіантів передбачає об’єднання ознак, де з підозрілого виконуваного файлу витягуються як статичні характеристики (зокрема хеші, опкоди, імпортовані бібліотеки), так і динамічні журнали виконання (наприклад, послідовності API-викликів або зміни у системі). Ці дані поєднуються в єдиний вектор ознак, що подається на вхід класифікатору. Інший підхід базується на багатокроковій детекції, де на першому етапі застосовується швидкий статичний сканер для виявлення відомих загроз або підозрілих патернів, а на другому етапі файли, що залишаються непідтвердженими, підлягають глибшому динамічному аналізу у пісочниці. Такий підхід забезпечує оптимальний баланс між швидкістю аналізу та повнотою перевірки, що позитивно впливає на загальну ефективність детекції загроз.

Ефективність гібридних методів підтверджується численними науковими дослідженнями. Зокрема, у порівняльному аналізі, проведеному у 2023 році для виявлення загроз на платформі Android, встановлено, що хоча як статичні, так і динамічні методи здатні успішно ідентифікувати шкідливе ПЗ (зокрема програми-вимагачі), саме їхня комбінація в межах гібридного підходу продемонструвала найвищу ефективність. За результатами експерименту, така модель досягла майже 100% точності (precision) у виявленні ransomware із рівнем хибних спрацювань менше ніж 4%. Інші дослідження також засвідчують, що поєднання статичних і динамічних ознак суттєво підвищує повноту виявлення загроз, збільшуючи показник True Positive Rate (TPR) і зменшуючи ймовірність пропущення складних зразків[14], що могли б уникнути виявлення одним із методів. Наприклад, якщо шкідливе ПЗ використовує техніки упаковування або шифрування коду, статичний аналіз може не зафіксувати загрозу, однак її активність стане помітною під час виконання програми. У протилежному випадку, якщо зловмисне ПЗ навмисно уникає небезпечної поведінки у пісочниці, статичний аналіз може виявити підозрілі фрагменти коду або аномальні характеристики у структурі файлу. Таким чином, поєднання статичних і динамічних ознак у межах гібридного підходу значно підвищує стійкість системи до сучасних технік обходу захисту, забезпечуючи всебічну перевірку на різних рівнях аналізу.

Однак, попри високу ефективність, гібридні системи мають і певні обмеження. Насамперед, їхня складність і повільніша швидкодія є суттєвими недоліками. Виконання обох типів аналізу — статичного та динамічного — потребує більших обчислювальних ресурсів і може значно збільшити час обробки. Дослідження показують, що моделі, які поєднують послідовний пошук шаблонів і глибоке навчання, демонструють найбільший час виконання серед протестованих методів. Через це гібридний аналіз менш придатний для застосування в умовах жорстких часових обмежень, зокрема для сканування мережевого трафіку в режимі реального часу. Крім того, інтеграція ознак різної природи створює додаткові виклики у побудові моделі. Під час поєднання статичних і динамічних характеристик виникає ризик надлишкових або дублюючих ознак, що може призвести до перенасичення моделі непотрібною інформацією. Це, своєю чергою, створює загрозу зниження ефективності класифікатора або дублювання корельованих даних. Для вирішення цієї проблеми деякі дослідники застосовують методи відбору ознак або створюють

дворівневі системи, де на першому етапі працюють окремі детектори, а на другому — мета-класифікатор, що об'єднує їхні висновки.

Методи машинного навчання

Методи машинного навчання стали ключовим компонентом сучасних систем виявлення шкідливого програмного забезпечення (ЗПЗ), охоплюючи широкий спектр завдань — від фільтрації спаму[15] до антивірусного сканування. Упродовж останнього десятиліття класичні алгоритми, такі як логістична регресія, метод опорних векторів (SVM), дерева рішень, випадковий ліс та Naive Bayes, активно застосовувалися для класифікації програмного забезпечення на основі різноманітних статичних і динамічних ознак. Водночас сучасна тенденція, що простежується у період 2021–2025 років, свідчить про стрімке зростання популярності методів глибинного навчання, що демонструють високу здатність автоматично виявляти приховані патерни у даних та узагальнювати знання на нові приклади. Завдяки цим властивостям нейронні мережі значно підвищують ефективність детекції загроз.

Серед сучасних підходів особливо виділяються згорткові нейронні мережі (CNN), які застосовуються для розпізнавання шкідливих програм, представлених у вигляді зображень або послідовностей байтів, а також рекурентні нейромережі (LSTM/GRU), що ефективно аналізують послідовності API-викликів та системних подій. Графові нейронні мережі, своєю чергою, демонструють високі результати у завданнях виявлення шкідливого ПЗ шляхом дослідження графів викликів функцій[16] або потоків управління. Зокрема, кілька моделей, що класифікують шкідливе ПЗ у форматі зображень, досягли точності на рівні 98–99% на наборі даних MalImg, тоді як алгоритми, розроблені для виявлення обфусцьованого шкідливого ПЗ, продемонстрували точність понад 99%. Особливо високу ефективність показали моделі, що поєднують методи глибинного навчання з інженерією ознак. Наприклад, модель, яка інтегрувала підходи обробки природної мови (NLP) для аналізу текстових рядків із глибокими нейронними мережами, досягла 99,91% точності та виявляла навіть упаковане шкідливе ПЗ.

Окрім глибинних нейромереж, широкого поширення набули ансамблеві методи, які поєднують декілька класифікаторів для підвищення надійності та точності виявлення загроз. Серед них найбільшу популярність отримали алгоритми Random Forest та Gradient Boosting (наприклад, XGBoost і LightGBM), які активно використовуються у статичному аналізі виконуваних файлів. Зокрема, на еталонному наборі даних EMBER 2018 року метод градієнтного бустингу продемонстрував високу ефективність, досягнувши значення AUC понад 0.99 при статичному виявленні шкідливого ПЗ. У дослідженні 2024 року порівняльний аналіз класичних ML-алгоритмів для класифікації послідовностей API показав, що випадковий ліс продемонстрував найкращі показники точності, прецизії та повноти порівняно з методами k-NN та логістичною регресією. Проте, навіть попри високу ефективність ансамблевих методів, у тестах на тих самих вибірках глибокі нейромережі демонстрували вищі результати, підтверджуючи загальну тенденцію до домінування методів глибинного навчання у цій сфері.

Окрім наглядових методів класифікації, у сфері кібербезпеки активно застосовуються ненаглядові підходи, зокрема кластеризація. Цей метод дозволяє групувати схожі зразки шкідливого ПЗ у відповідні сімейства або виявляти аномальні об'єкти, які не належать до жодної відомої категорії. Такий підхід є особливо корисним для виявлення атак типу zero-day, коли навчальні дані без відповідних міток ще недоступні. Одним із перспективних підходів у цій сфері є метод UMD (Unsupervised Malware Detection), який поєднує адверсарний автоенкодер із глибоким кластеризаційним алгоритмом. Цей підхід продемонстрував високу ефективність у виявленні нових загроз, що підтверджено експериментальними дослідженнями.

Попри значні досягнення методів машинного навчання у сфері кібербезпеки, їхнє впровадження супроводжується низкою викликів. Однією з основних проблем є забезпечення стабільно високої точності виявлення із мінімальною кількістю хибних спрацювань[17] у реальних умовах. Хоча багато ML-алгоритмів демонструють майже ідеальні результати на фіксованих тестових наборах, їхня ефективність може знижуватися при зіткненні з новими, невідомими зразками шкідливого ПЗ, особливо якщо зловмисники свідомо адаптують свої атаки для обходу відомих детекторів. Одним із найбільш серйозних викликів є високий рівень хибнопозитивних спрацювань (False Positive Rate, FPR), що створює додаткове навантаження на аналітиків та системи моніторингу. Додатково ускладнює ситуацію низька інтерпретованість глибоких нейромережевих моделей, що працюють за принципом «чорної скриньки», що ускладнює пояснення прийнятих рішень.

Для подолання зазначених викликів активно розробляються підходи на основі Explainable AI (XAI), що спрямовані на підвищення прозорості роботи ML-систем у сфері безпеки. Також перспективними напрямами залишаються методи захисту ML-моделей від адверсаріальних атак, які передбачають навмисне внесення змін до структури або поведінкових ознак шкідливого коду для обходу детектора. Наприклад, додавання випадкового коду, що не виконує шкідливих дій, або зміна порядку інструкцій можуть допомогти обійти систему детекції. Таким чином, хоча методи машинного навчання демонструють значний потенціал у виявленні шкідливого ПЗ, їхня ефективність залежить від здатності протидіяти сучасним методам обходу та забезпечення стабільної роботи в умовах змінного кіберпростору.

Виклад основного матеріалу

Модель атаки ґрунтується на використанні стеганографічних технік для прихованої передачі команд управління шляхом інтеграції цих команд у графічні файли. Зловмисник кодує необхідні керуючі інструкції безпосередньо всередині зображення, використовуючи, наприклад, найменш значущі біти (LSB), дискретне косинусне перетворення (DCT) або інші стеганографічні методи[18]. Це дозволяє організувати прихований канал комунікації, який важко виявити традиційними методами, орієнтованими на відкритий аналіз мережевого трафіку чи поведінкових аномалій процесів. Для протидії таким прихованим атакам запропоновано методику, що складається з двох основних етапів: статичного та динамічного аналізу. На першому етапі (статичний аналіз) графічні файли, що надходять у систему, перевіряються на наявність характерних ознак застосування відомих стеганографічних технік. Формально цей етап описується так:

$$A_S(F) = \begin{cases} 1, \text{якщо } \exists s_i \in S : s_i \subset F \\ 0, \quad \text{в решті випадків} \end{cases} \quad (1)$$

де $A_S(F)$ — результат статичного аналізу для файлу F, S — $\{s_1, s_2, s_3 \dots s_n\}$множина ознак, що свідчать про застосування стеганографії, таких як артефакти у структурі пікселів чи незвичні патерни шумів. У випадку позитивного результату первинного аналізу $A_S(F) = 1$, підозрілий процес P, який працює з таким файлом, переміщується для додаткового аналізу у спеціально створену ізольовану віртуальну машину, що повністю дублює початкове робоче середовище. Таке ізольоване середовище гарантує безпечне виконання аналізу без ризику компрометації основної системи.

На другому етапі (динамічний аналіз) здійснюється глибокий моніторинг та аналіз поведінки процесу з використанням нейронної мережі типу Long Short-Term Memory (LSTM). Вхідними даними для LSTM є часові послідовності таких параметрів[19]: мережевий трафік $T(t)$ (інтенсивність, напрямок, частота запитів), операції з файлами $Q_f(t)$ (створення, модифікація, видалення файлів), споживання ресурсів системи $R(t)$ (навантаження процесора, оперативної пам'яті), а також активність в оперативній пам'яті $M(t)$ (обсяг пам'яті, типи звернень, адресні запити). Вектор стану поведінки процесу у будь-який момент часу визначається наступним чином:

$$X(t) = [T(t), Q_f(t), R(t), M(t)] \quad (2)$$

LSTM-мережа аналізує отриманий набір даних, виявляючи характерні патерни, що дозволяє оцінити ймовірність належності процесу до класу зловмисних:

$$P(y = 1|X(t), W) = \sigma\left(\sum_i W_i h_i(t) + b\right) \quad (3)$$

де $P(y = 1|X(t), W)$ — умовна ймовірність того, що процес є шкідливим; $h_i(t)$ —стан прихованих шарів нейромережі; W_i —вагові коефіцієнти; b —зміщення; σ —сигмоїдна функція активації.

Ефективність запропонованої методики оцінюється з використанням таких метрик: точність класифікації (Accuracy), повнота виявлення атак (Recall) та частота хибних спрацьовувань (FPR), які визначаються як:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$FPR = \frac{FP}{FP + TN} \quad (5)$$

де TP — правильно виявлені атаки, TN —правильно визначені безпечні процеси, FP —хибні спрацьовування на легітимних процесах, FN —нерозпізнані атаки. Таким чином, запропонована модель і методика дозволяють ефективно ідентифікувати та блокувати складні стеганографічні атаки, що використовують графічні файли для прихованої комунікації.

Експеримент

Метою експериментів є оцінка ефективності запропонованої методики виявлення зловмисних команд[20], що передаються через графічні файли із використанням стеганографічних методів. Дослідження спрямоване на демонстрацію точності, повноти та частоти хибних спрацьовувань запропонованої системи у порівнянні з традиційними методами.

Для проведення експериментів було розгорнуто середовище, що включало серверну систему для аналізу трафіку та процесів[21]. У рамках експериментів застосовувалася віртуальна машина для імітації атак та виконання підозрілих процесів. Для створення та детекції стеганографічних даних використовувались інструменти OpenStego, StegExpose та LSB-Steganalysis Toolkit. Реалізацію LSTM-мережі здійснено на основі бібліотек TensorFlow та Keras.

Для навчання та тестування запропонованої моделі було сформовано набір даних, що включав 6000 графічних файлів без прихованих даних (чисті зразки) та 6000 графічних файлів зі зловмисними командами, прихованими стеганографічними методами, зокрема LSB та DCT. Загалом у дослідженні використано 12 000 зображень. Для кожного зразка здійснювалися симуляції підозрілих процесів із

різними сценаріями активності, включаючи мережевий трафік, звернення до файлової системи та споживання ресурсів.

Запропонована методика включала два етапи: статичний аналіз для виявлення аномалій у структурі файлів та динамічний аналіз на основі LSTM-мережі для оцінювання поведінкових характеристик[22] підозрілих процесів. Комбінований підхід дозволяв поєднати переваги обох методів, що значно підвищувало ефективність виявлення загроз. Слід зазначити, що запропонована методика вимагає формування коректної та збалансованої навчальної вибірки для досягнення високих показників ефективності. Крім того, метод характеризується підвищеними вимогами до обчислювальних ресурсів у порівнянні з традиційними підходами, що пов'язано з глибоким аналізом поведінкових характеристик процесів у реальному часі.

Експерименти здійснювалися у три етапи. На першому етапі перевірялася здатність алгоритму виявляти ознаки стеганографічних маніпуляцій у файлах на основі статичного аналізу[23]. Другий етап передбачав оцінку ефективності динамічного аналізу на основі LSTM, де аналізувалися поведінкові патерни підозрілих процесів. На третьому етапі виконувалося порівняння запропонованої методики з існуючими інструментами, такими як StegExpose та LSB-Steganalysis Toolkit, за метриками Accuracy, Recall та FPR.

Результати тестування продемонстрували, що статичний аналіз показав точність на рівні 89%, повноту — 84% та частоту хибних спрацьовувань — 12%. Динамічний аналіз на основі LSTM досяг показників точності 94%, повноти — 92% та FPR — 7%. Найкращі результати продемонстрував запропонований метод, який забезпечив точність 98%, повноту — 96% та найнижчий показник FPR — 3%. Для порівняння, інструмент StegExpose показав точність 72%, повноту — 70% та FPR — 28%, тоді як LSB-Steganalysis Toolkit досяг показників 76%, 73% та 25% відповідно.

Висновки підтверджують ефективність запропонованої методики у виявленні зловмисних команд, прихованих у графічних файлах[24]. Запропонований метод, що поєднує статичний та динамічний аналіз, продемонстрував найвищі показники точності (98%) та мінімальну кількість хибних спрацьовувань (3%) серед усіх досліджених методів. Порівняльний аналіз ефективності методик представлено в таблиці 1

Таблиця 1

Порівняння ефективності методик

Метод	Accuracy	Recall	FPR
Запропонований метод	98%	96%	3%
Статичний аналіз	89%	84%	12%
Динамічний аналіз (LSTM)	94%	92%	7%
Комбінований підхід	97%	95%	4%
StegExpose	72%	70%	28%
LSB-Steganalysis Toolkit	76%	73%	25%

Запропонована методика має також певні недоліки. Основним обмеженням є висока залежність від коректно сформованої навчальної вибірки, що може впливати на результати при недостатньо репрезентативних даних. Крім того, метод потребує значних обчислювальних ресурсів, особливо на етапі динамічного аналізу, що може ускладнити його застосування в умовах обмеженої апаратної інфраструктури. Подальші дослідження планується зосередити на розширенні можливостей системи для роботи з іншими форматами мультимедійних файлів та вдосконаленні механізмів тренування нейронної мережі

Висновки

У ході дослідження було запропоновано методику виявлення зловмисних команд, прихованих у графічних файлах за допомогою стеганографічних методів. Запропонований підхід поєднує статичний та динамічний аналіз, що дозволяє ідентифікувати як структурні аномалії у графічних файлах, так і специфічні поведінкові патерни підозрілих процесів.

Проведені експериментальні дослідження підтвердили високу ефективність розробленої методики. Комбіноване застосування статичного та динамічного аналізу дозволило досягти показників точності на рівні 98%, повноти — 96% та частоти хибних спрацьовувань — 3%. Ці результати значно перевищують ефективність традиційних інструментів, таких як StegExpose та LSB-Steganalysis Toolkit, що демонструють нижчі показники якості класифікації.

Запропонована методика виявилась ефективною не лише у виявленні прихованих команд, а й у забезпеченні безпечного середовища для аналізу підозрілих процесів шляхом їх ізоляції у віртуальній машині. Це забезпечує додатковий рівень захисту інформаційної системи під час дослідження потенційних загроз.

Водночас методика має певні обмеження. Зокрема, її ефективність значною мірою залежить від якості та збалансованості навчальної вибірки. Недостатньо репрезентативні дані можуть негативно вплинути на точність моделі. Крім того, застосування динамічного аналізу на основі LSTM-мережі потребує значних обчислювальних ресурсів, що може ускладнити впровадження методу у системах з обмеженою апаратною потужністю.

Перспективи подальших досліджень включають розширення можливостей запропонованої методики для аналізу інших мультимедійних форматів, а також вдосконалення механізмів навчання нейронної мережі для підвищення її адаптивності до нових видів стеганографічних атак.

Література

1. Owoh, N., Adejoh, J., Hosseinzadeh, S., Ashawa, M., Osamor, J., & Qureshi, A. (2024). Malware detection based on API call sequence analysis: A gated recurrent unit–generative adversarial network model approach. *Future Internet, 16*(10), 369. https://doi.org/10.3390/fi16100369
2. García, D. E., & DeCastro-García, N. (2021). Optimal feature configuration for dynamic malware detection. *Computers & Security, 105*, 102250. https://doi.org/10.1016/j.cose.2021.102250
3. Li, C., Cheng, Z., Zhu, H., Wang, L., Lv, Q., Wang, Y., et al. (2022). DMalNet: Dynamic malware analysis based on API feature engineering and graph learning. *Computers & Security, 122*, 102872. https://doi.org/10.1016/j.cose.2022.102872
4. Ilić, S., Gnjatović, M., Tot, I., Jovanović, B., Maček, N., & Božović, M. G. (2024). Going beyond API calls in dynamic malware analysis: A novel dataset. *Electronics, 13*(17), 3553. https://doi.org/10.3390/electronics13173553
5. Subrahmanyam, S. S. B., Goutham, P., Ambati, V. K. R., Bijitha, C. V., & Nath, H. V. (2023). A hybrid method for analysis and detection of malicious executables in IoT network. *Computers & Security, 132*, 103339. https://doi.org/10.1016/j.cose.2023.103339
6. Savenko, B., Kashtalian, A., Lysenko, S., & Savenko, O. (2023). Malware detection by distributed systems with partial centralization. In *Proceedings of the 2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)* (pp. 265–270). IEEE. https://doi.org/10.1109/IDAACS58523.2023.10348773
7. Kılıç, K., Atacak, İ., & Doğru, İ. A. (2025). FABLDroid: Malware detection based on hybrid analysis with factor analysis and broad learning methods for Android applications. *Engineering Science and Technology, an International Journal, 62*, 101945. https://doi.org/10.1016/j.jestch.2024.101945
8. Denysiuk, D., Savenko, O., Lysenko, S., Savenko, B., & Kashtalian, A. (2023). Method for detecting steganographic changes in images using machine learning. In *Proceedings of the 2023 13th International Conference on Dependable Systems, Services and Technologies (DESSERT)* (pp. 1–6). IEEE. https://doi.org/10.1109/DESSERT61349.2023.10416453
9. Bensaoud, A., Kalita, J., & Bensaoud, M. (2024). A survey of malware detection using deep learning. *Machine Learning with Applications, 16*, 100546. https://doi.org/10.1016/j.mlwa.2024.100546
10. Kim, H., & Kim, M. (2024). Malware detection and classification system based on CNN-BiLSTM. *Electronics, 13*(13), 2539. https://doi.org/10.3390/electronics13132539
11. Guo, W., Du, W., Yang, X., Xue, J., Wang, Y., Han, W., & Hu, J. (2025). MalHAPGNN: An enhanced call graph-based malware detection framework using hierarchical attention pooling graph neural network. *Sensors, 25*(2), 374. https://doi.org/10.3390/s25020374
12. Kashtalian, A., Lysenko, S., Savenko, O., Nicheporuk, A., Sochor, T., & Avsiyevych, V. (2024). Multi-computer malware detection systems with metamorphic functionality. *Radioelectronic and Computer Systems, 2024*(1), 152–175. https://doi.org/10.32620/reks.2024.1.13
13. Deng, L., Wen, H., Xin, M., Li, H., Pan, Z., & Sun, L. (2023). Enimanal: Augmented cross-architecture IoT malware analysis using graph neural networks. *Computers & Security, 132*, 103323. https://doi.org/10.1016/j.cose.2023.103323
14. Badar, L. T., Carminati, B., & Ferrari, E. (2025). A comprehensive survey on stegomalware detection in digital media, research challenges and future directions. *Signal Processing, 231*, 109888. https://doi.org/10.1016/j.sigpro.2025.109888
15. Liu, Y., Tantithamthavorn, C., Li, L., & Liu, Y. (2022). Explainable AI for Android malware detection: Towards understanding why the models perform so well? In *Proceedings of the 33rd IEEE International Symposium on Software Reliability Engineering (ISSRE)* (pp. 169–180). IEEE. https://doi.org/10.1109/ISSRE55969.2022.00026
16. Vouvoutsis, V., Casino, F., & Patsakis, C. (2025). Beyond the sandbox: Leveraging symbolic execution for evasive malware classification. *Computers & Security, 149*, 104193. https://doi.org/10.1016/j.cose.2024.104193
17. Taheri, R., Shojafar, M., Arabikhan, F., & Gegov, A. (2024). Unveiling vulnerabilities in deep learning-based malware detection: Differential privacy driven adversarial attacks. *Computers & Security, 146*, 104035. https://doi.org/10.1016/j.cose.2024.104035

18. Wangwang, W., et al. (2021). Network traffic oriented malware detection in IoT (Internet-of-Things). In *Proceedings of the 2021 International Conference on Networking and Network Applications (NaNA)* (pp. 301–307). IEEE. https://doi.org/10.1109/NaNA53461.2021.00060

19. Aboaoja, F. A., et al. (2022). Malware detection issues, challenges, and future directions: A survey. *Applied Sciences, 12*(17), 8482. https://doi.org/10.3390/app12178482

20. Redhu, A., et al. (2024). Deep learning-powered malware detection in cyberspace: A contemporary review. *Frontiers in Physics, 12*, 1349463. https://doi.org/10.3389/fphy.2024.1349463

21. Lysenko, S., et al. (2022). IoT multi-vector cyberattack detection based on machine learning algorithms: Traffic features analysis, experiments, and efficiency. *Algorithms, 15*(7), 239. https://doi.org/10.3390/a15070239

22. Zhang, S., et al. (2025). A malware-detection method using deep learning to fully extract API sequence features. *Electronics, 14*(1), 167. https://doi.org/10.3390/electronics14010167

23. Jian, Y., et al. (2021). A novel framework for image-based malware detection with a deep neural network. *Computers & Security, 109*, 102400. https://doi.org/10.1016/j.cose.2021.102400

24. Alshomrani, M., et al. (2024). Survey of transformer-based malicious software detection systems. *Electronics, 13*(23), 4677. https://doi.org/10.3390/electronics13234677