Шифр «MALWARE»

Секція «Кібербезпека»

МОДЕЛЬ І МЕТОД МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ПРИСТРОЯХ ІНТЕРНЕТУ РЕЧЕЙ

2021

ЗМІСТ

ВСТУП 3
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ 5
1.1 Тенденції розвитку технологій аналізу шкідливих програм в мережах інтернету речей 5
1.2 Моделі і методи класифікаційного аналізу даних 11
1.3 Формалізована постановка задачі 15
РОЗДІЛ 2 ТЕХНОЛОГІЯ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ПРИСТРОЯХ ІНТЕРНЕТУ РЕЧЕЙ 17
2.1 Модель і метод навчання класифікатора шкідливого програмного забезпечення 17
2.2 Критерій оптимізації вирішувальних правил 22
2.3 Результати фізичного моделювання 27
ВИСНОВКИ 31
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 32

ВСТУП

В умовах гібридної війни кіберпростір України є ціллю для кібератак, наслідком яких можуть бути значні економічні втрати, а в деяких ситуаціях і екологічні катастрофи. Існує прецедент 2017 року, коли комп'ютерний вірус Petya вчинив дестабілізуючий вплив на економіку України. Тому дослідження і розробка технологій кіберзахисту є актуальними для вирішення безпекових питань держави. При цьому будь-якій суверенній державі корисно мати власні ефективні розробки в сфері кібербезпеки, оскільки це дозволяє підвищити контроль за їх створенням для уникнення загроз, пов'язаних з експлойтами, і адаптувати дані розробки під особливості безпекової стратегії України.

Кіберфізичні системи та інтернет речей, як їх складова, збільшують свій економічний, комерційний та соціальний вплив на наше життя. Крім того, кіберфізичні системи стають звичайним елементом критичної інфраструктури. При цьому розподілені пристрої кіберфізичних систем, як правило, обмежені в ресурсах, що робить їх привабливими цілями для кібератак. Згідно з даними лабораторії AV-TEST, очікується, що у 2021 році з'явиться понад 1 мільярд шкідливих програм. Попри те, що традиційно ціллю шкідливого програмного забезпечення були персональні комп'ютери, останнім часов зростає кількість атак на пристрої інтернету речей, такі як смартфони, холодильники, пральні машини, розподілені сенсори, медичне обладнання та інше.

У загальному випадку пристрої інтернету речей є незахищеними внаслідок складності створення уніфікованих стандартів для гетерогенного кіберфізичного середовища. Пристрої інтернету речей мають досить різноманітну архітектуру апаратного і програмного забезпечення. На відміну від персональних комп'ютерів в пристроях інтернету речей домінують різноманітні операційні системи сімейства Linux. На ринку відсутні ефективні розробки для розпізнавання шкідливого програмного забезпечення на даних платформах. При цьому традиційні методи синтезу систем розпізнавання шкідливого програмного забезпечення є обчислювально трудомісткими і чутливими до атак "нульового дня". Тому задача

щодо розробки обчислювально ефективних засобів розпізнавання шкідливого програмного забезпечення, що забезпечують робастність до цільової платформи і високу узагальнюючу здатність, є актуальною задачею.

Мета роботи – розробка ефективних моделі та алгоритму навчання для розпізнавання шкідливого програмного забезпечення у пристроях інтернету речей за умов обмежених обсягів обчислювальних ресурсів та актуальної розміченої навчальної вибірки.

Об'єкт дослідження — слабоформалізований процес розпізнавання шкідливого програмного забезпечення в мережі інтернету речей.

Предмет дослідження — моделі і методи інформаційної технології розпізнавання шкідливого програмного забезпечення в пристроях кіберфізичних систем.

Методи дослідження — методи технології нейронних мереж, інформаційно-екстремальної інтелектуальної технології.

Для досягнення мети дослідної роботи здійснюється формування вхідних навчаних та тестових вибірок з наборів даних KISA Data Challenge 2019 та Virus-Share Dataset, розроблено модель та метод машинного навчання. Здійснюється оцінка ефективності запропонованих алгоритмів за тестовими даними.

Робота має теоретичний та прикладний характер, запропоновано нову нейромережеву модель та новим багатофазний метод її навчання і порівняно запропонований підхід з традиційним. Результати, отримані на вибірках з відкритих наборів даних KISA Data Challenge 2019 та Virus-Share Dataset підтверджують придатність моделей і алгоритмів навчання для практичного використання. Запропонований алгоритм навчання використовує 36000 розмічених навчальних зразків, забезпечуючи 95% правильного розпізнавання шкідливого програмного забезпечення для операційної системи Windows та Linux на 14000 тестових зразків.

РОЗДІЛ 1
АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Тенденції розвитку технологій аналізу шкідливих програм в мережах інтернету речей

Кібер-загрози від шкідливих програм (malware) зростають з кожним роком. Традиційні сигнатуні методи розпізнавання шкідливого програмного забезпечення є малоефективними внаслідок швидкого зростання кількості і різноманітності шкідливого програмного забезпечення. Для протидії таким атакам потрібно регулярно оновлювати антивірусне програмне забезпечення, однак навіть це не убезпечить від атак “нульового дня”. Методи машинного навчання завдяки узагальненню даних дозволяють прогнозувати і розпізнавати нові шкідливі програми, однак застосування традиційних підходів призводить до частих хибних спрацювань (false positive). Багато досліджень проводяться з метою зменшення кількості хибних спрацювань.

Поширення технологій побудови розумних будинків привело до появи мереж вбудованих пристроїв, до яких належать телевізори, радіо, очищувачі повіртря і різноманітні побутові електронні прилади. Так само розвиваються технології розумних міст, зростає кількість сенсорних систем і виконуючих пристроїв, об’єднаних мережевою взаємодією. Ці пристрої називають інтернетом речей. До них також відносять такі смарт-пристрої як смартфони, планшети, трекери здоров’я та різноманітне медичне обладнання. При цьому більшість кібер-атак спрямовані на зараження пристроїв для його контролю з метою несанкціонованого доступу до конфіденційної інформації, реаліції ботнет мереж чи вчинення інших неправомірних дій.

Гетерогенність інфокомунікаційного середовища інтернету речей на вбудованих пристроїв ускладнює розробку універсальних засобів захисту. Крім того більшість існуючих засобів розпізнавання шкідливої кібер-активності на основі сигнатур та штучного інтелекту розроблені для операційної системи Windows та процесорних архітектур настільного ком’тера. Крос-архітектурні ж рішення як

правило основані на аналізі мережевих журналів, що зменшує обсяг доступних інформативних даних для виявлення шкідливих кодів і процесів. Однак 71,8% операційних систем, що використовуються для IoT-пристроїв, шлюзів та хмарних внутрішніх пристроїв і промислових контролерів, мають Linux або Unix подібну архітектуру. В існуючих дослідженнях щодо використання штучного інтелекту для захисту смарт-пристроїв розглядаються в основному операційні системи Windows та Android. Але існує велика палітра комбінацій апартної архітектури та операційних Linux-систем, що не були вивчені з точки зору захисту від шкідливих кодів. Існує необхідність розробки універсальних та уніфікованих методів захисту кіберфізичного інфокомунікаційного середовища від шкідливого програмного забезпечення замість розробки окремих реалізацій під кожний частковий випадок.

Існуючі системи кіберзахисту пристроїв кіберфізичних систем не забезпечують високої достовірності рішень, оскільки кількість та різноманітність нових джерел кіберзагроз постійно зростає, а кількість актуальних розмічених даних є досить обмеженою [1]. Гетерогенність кіберфізичних систем обумовлює високу варіативність спостережень і знижує репрезентативність будь-яких вибірок даних.

Вручну спроектовані ознаки, що використовуються у класичних методах машинного навчання, не є гнучкими і адаптованими до появи нових реалізацій шкідливого програмного забезпечення. Тому варто розглядати застосування здатного до навчання екстрактора ознак, який здатний автоматино сформувати ефективні ознаки з вхідних навчальних даних. Екстрактори ознак, як правило, основані на методах глибокого машинного навчання. Найбільш потужною частиною глибокого навчання є навчання ієрархічного ознакового подання (рис. 1.2).

Високорівневі ознаки формуються шляхом композиції низькорівневих. Вище по ієрархії знаходиться менше ознак, проте вони є складнішими, інформативнішими і менш чутливими до варіації спостережень одного і того ж класу розпізнавання. Як неієрархічні, так і ієрархічні моделі здатні апроксимувати будь-яку функцію з однаковою точністю, проте ієрархічні моделі потребують меншу кількість параметрів для цього [2].

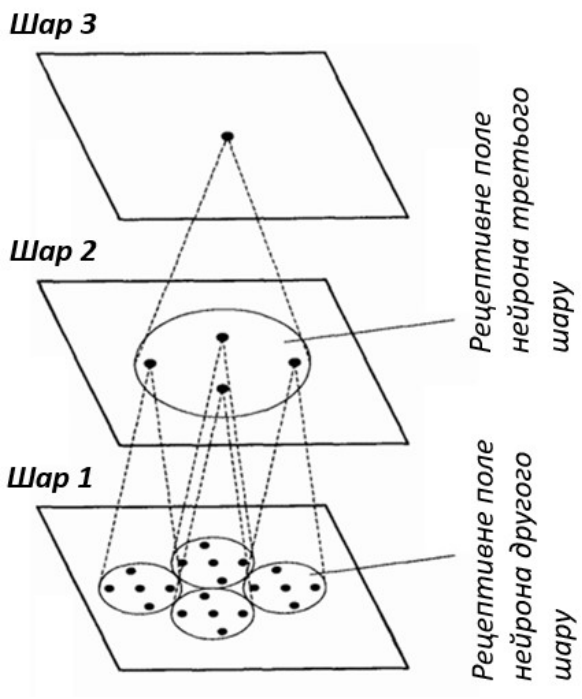

Рисунок 1.2 – Схема ієрархічного ознакового подання даних

Узагальнену структуру моделі класифікаційного аналізу шкідливого програмного забезпечення в рамках кожного окремого підходу показано на рис. 1.3.

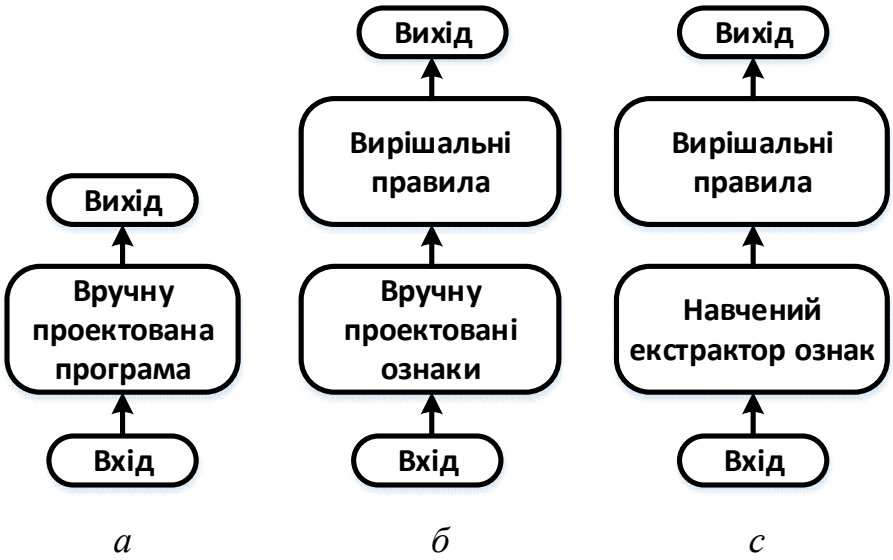

Рисунок 1.3 – Узагальнена структура моделі аналізу програмного забезпечення : а – моделі, основані на правилах (наприклад, сигнатурний аналіз); б – моделі, основані на використанні класичного машинного навчання; с – моделі, основані на навчанні ієрархічного екстрактора ознак

Дослідники з Intel та Microsoft показали ефективність аналізу шкідливого програмного забезпечення комп’ютерів з використанням згорткових глибоких нейронних мереж [3]. При цьому бінарні виконувані файли напряму байт за байтом подавалися у вигляді однокального 2D зображення у сірих відтінках. Такий підхід дозволяє застососувавати добре напрацьовані моделі і методи навчання згорткових мереж. Проте дана робота розглядала як цільову платформу персональні комп’ютери під управлінням операційної системи Windiws. Однак пристрої інтернету речей мають більшу варіативність операційних систем, бібліотек та апартного забезпечення, що потребує підвищення інформативності ознакового подання і удосконалення самої моделі аналізу даних з метою запобігання ефектів упередженості чи насиченості.

Згорткові багатошарові нейронні мережі дозволяють сформувати високорівневе інформативне ознакове подання спостережень [4]. При цьому вони вже показали високу ефективність у задачах машинного зору та аналізу часових рядів [8]. З 1989 року по сьогодні було виконано ряд архітектурних вдосконалень згорткової нейронної мережі. Вдосконалення стосувалися таких аспектів як оптимізація параметрів, регуляризація, структурне переформулювання тощо. Проте останні тенденції в модернізації згорткової мережі пов'язані з реструктуризацією існуючих компонентів обробки даних та проектуванням нових блоків і модулів. Залежно від типу архітектурних модифікацій згорткові мережі можна класифікувати на сім різних груп (рис. 1.4) :

– експлуатація простору (spatial exploitation), тобто використання різномасштабних фільтрів;

– експлуатація глибини (depth exploitation), тобто збільшення кількості шарів і міри щодо забезпечення незатухання градієнту на нижніх шарах;

– експлуатація мульти-шляхів (multi-path exploitation), тобто використання міжшарових з’єднань та псевдо-ансамблювання на основі dropout і drop-connection;

– експлуатація ширини (width exploitation), тобто міри щодо збільшення ширини нейронної мережі за рахунок мульти-з’єднань;

– експлуатація карти ознак (feature map exploitation), тобто підвищення

інформативності ознак за рахунок їх селекції/зважування на основі Squeeze-Excitation-модуля;

– експлуатація підсилення каналів (channel boosting exploitation), тобто додавання штучних каналів до оригінальних каналів, що сформовані за рахунок переносу знань (transfer learning), або використання генеративних моделей навчання;

– використання модулів уваги (attention based CNNs), тобто використання м'яких масок для карти ознак з метою фокусування моделі на важливих регіонах зображення/карти ознак.

Окрім архітектурних вдосконалень активно розробляються нові функції втрат та протоколи машинного навчання. При цьому для вирішення задач ефективного навчання класифікаційних моделей за умов незбалансованого та обмеженого обсягу розмічених навчальних зразків набули широкого використання два основні підходи до екстракції компактного ознакового подання, які ще називають ембедінгами (embedding). В рамках першоного підходу використовується автоекодери, що навчаються без вчителя на великому обсязі нерозмічених навчальних даних, що дозволяє виявити структуру вхідних даних. В рамках другого підходу реалізують навчання метрики подібності (similarity metric learning) на обмеженому обсязі розмічених зразків, що дозволяє виявити відношень схожості між зразками різних класів. З точки зору обчислювальної ефективності останній підхід є найбільш оптимальний. При цьому модель, що навчається за даною схемою навчання, ще називають сіамською нейромережею.

Сіамська мережа складається з двох або трьох гілок екстракторів ознак зі спільними параметрами (рис. 1.5). Параметри W багатошарової моделі екстракції ознак можна вважати оптимальними, якщо для пари семантично схожих зразків вихід мережі має близьке до нуля значення, а для пари семантично відмінних зразків – значення виходу, що близьке до одиниці [5, 6].

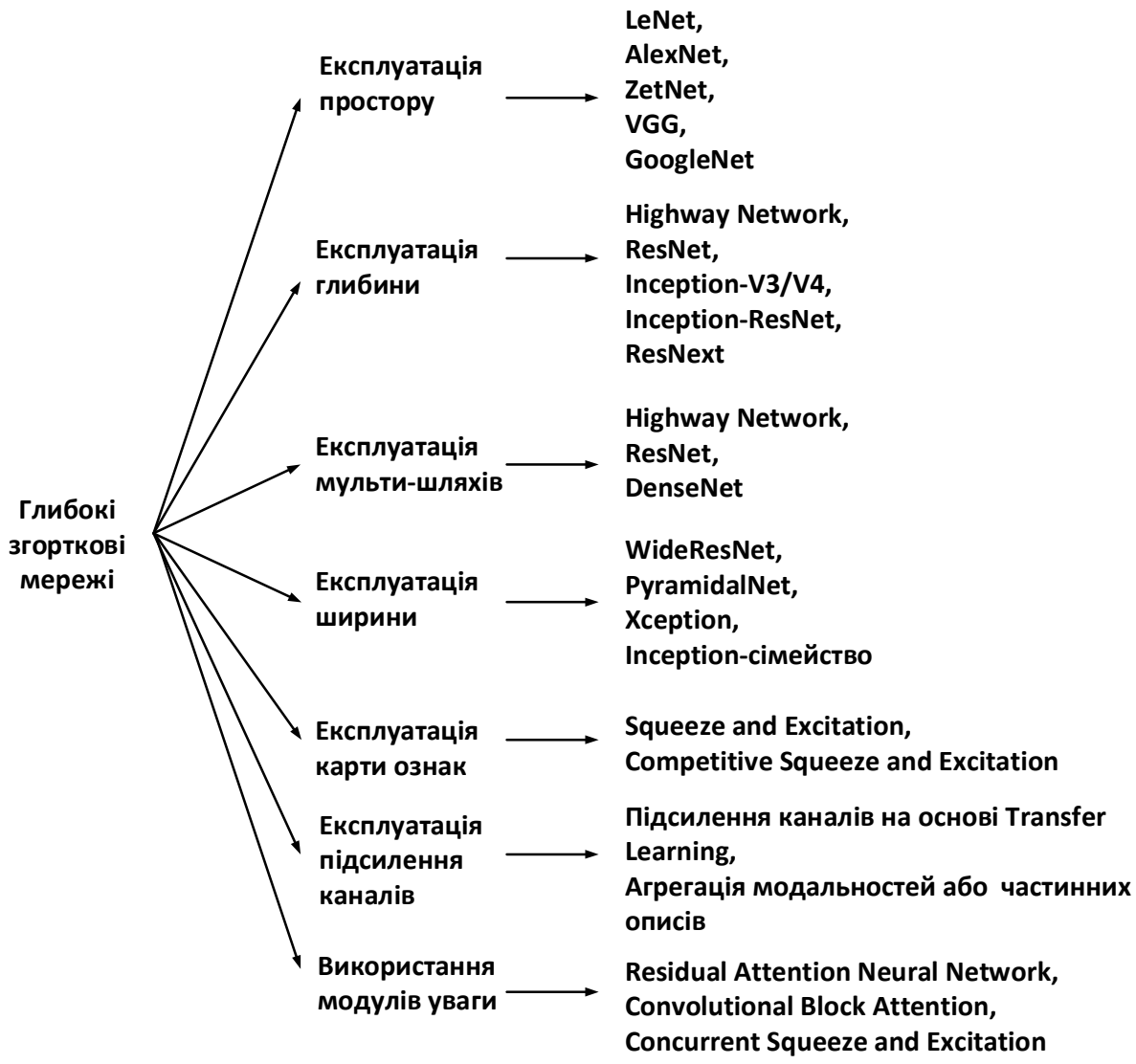

Рисунок 1.4 – Таксономія багатошарових (глибоких) згорткових нейронних мереж

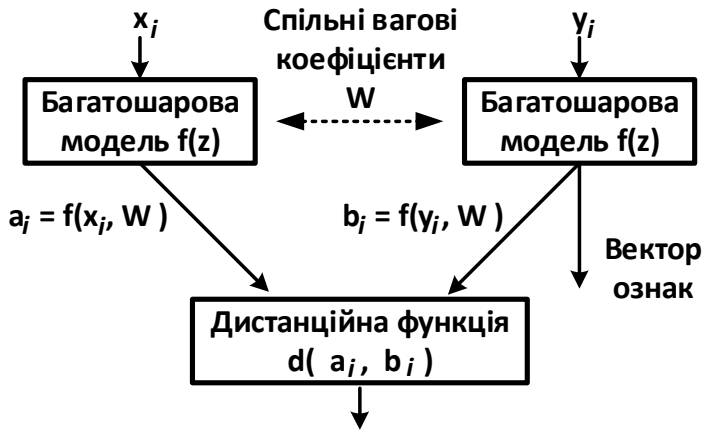

Рисунок 1.5 – Структура сіамської моделі аналізу даних

Алгоритми навчання, основані на сіамських нейронних мережах, ще називають “навчанням з декількох поглядів” (few-shot learning).

Таким чином, глибоке навчання є найбільш перспективним для застосування під час вирішення задачі детектування шкідливого програмного забезпечення в мережі інтернету речей. При цьому сіамські згорткові мережі розглядаються як обчислювально ефективний підхід в глибокому машинному навчанні, що здатний здійснювати аналіз образів.

1.2 Моделі і методи класифікаційного аналізу даних

Крім екстрактора інформативного високорівневого ознакового опису спостережень в системі детектування шкідливого трафіку важливим компонентом є вирішальні правила, що, як правило, представляють із себе класифікатор. При цьому ефективність навчання класифікатора часто розглядається як міра ефективності навчання екстрактора ознак.

В задачах класифікації схожі об’єкти частіше знаходяться в одному класі, ніж в інших. Тому часто користуються гіпотезою компактності, яка припускає, що класи утворюють компактно локалізовані підмножини в просторі об’єктів. При цьому для формалізації поняття «схожості» вводиться функція відстані або метрика d(x, y) в N-вимірному просторі об’єктів. Алгоритми, основані на аналізі схожості об’єктів, часто називають метричними, навіть у тих випадках, коли функція d не задовольняє всім аксіомам метрики (наприклад, аксіомі трикутника).

Алгоритм найближчого сусіда (nearest neighbor, NN) є найбільш простим метричним алгоритмом, який класифікує об’єкт як реалізацію того класу, якому належить найближчий об’єкт навчальної вибірки. Навчання такого класифікатора зводиться до елементарного запам’ятовування навчальної вибірки $\{y_{m,i}^{(j)} \mid m=\overline{1,M}; j=\overline{1,n}; i=\overline{1,N}\}$, де M – потужність алфавіту класів розпізнавання, n – обсяг спостережень класу X_m^o, N – кількість ознак розпізнавання. Єдиною перевагою цього алгоритму є простота реалізація. Однак недоліків набагато більше. По-перше, наявності викидів у навчальній вибірці призводить до нестійкості і

похибок. По-друге відсутні параметри, які можна б було настроювати за навчальною вибіркою. Алгоритм повністю залежить від успішності вибору дистанційної міри $d(x, y)$. Міри близькості обирають виходячи з властивостей об'єктів. Найбільшого поширення набули такі метрики, як Евклідова, Манхеттенська, Степенева та Журавльова [6].

Метричні алгоритми здійснюють локальну апроксимацію вибірки, при якій обчислення відкладаються до моменту, доки не стане відомий вхідний об'єкт. Тому метричні алгоритми відносяться до методів лінивого навчання (lazy learning).

Розвиток теорії машинного навчання відбувався в рамках статистичного підходу до розпізнавання образів, що є його головною заслугою. Зміст усіх статистичних методів навчання полягає у відновлені роздільної функції шляхом мінімізації середнього ризику помилкового розпізнавання [7]. Статистичні методи дозволяють побудувати вирішальні правила у випадках перетину класів розпізнавання, що має місце в практичних задачах контролю та керування слабо формалізованими процесами. Основними недоліками статистичних методів, які обмежують їх використання на практиці, є необхідність великих обсягів статистики для апроксимації функції щільності розподілу імовірностей, виконання жорстких умов для забезпечення статистичних стійкості та однорідності та висока чутливість до репрезентативності навчальних вибірок.

У рамках найбільш універсального підходу до аналізу і синтезу здатних навчатися систем прийняття рішень, яким є геометричний підхід, набув популярності метод опорних векторів SVM (support vector machine) [6]. В основі SVM лежить ідея розділення згущення векторів гіперплощинами, що знаходяться на максимальній відстані від згущень при мінімізації зміщення реалізацій класу від опорного вектора (рис. 1.6).

Задачу синтезу вирішальних правил SVM зручно формулювати для випадку двохкласової класифікації. При цьому багатокласова класифікація, як правило, зводиться до серії двокласових класифікацій. Прямі багатокласові узагальнення SVM існують, проте навчити один складний багатокласовий класифікатор набагато довше, ніж багато простих однокласових.

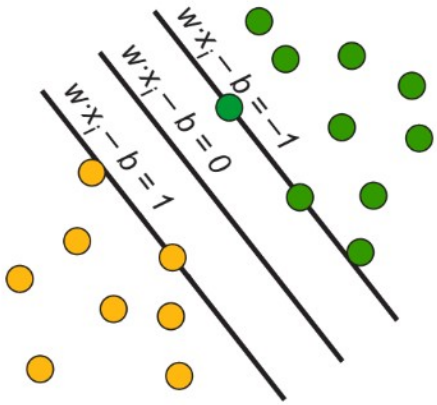

Рисунок 1.6 – Лінійне розділення образів в SVM

Проблема лінійної нероздільності в методі SVM вирішується шляхом переходу від початкового простору ознак до нового розширеного простору за допомогою деякого перетворення. На рис. 1.7 показано приклад переходу від двовимірного простору до розширеного тривімірного шляхом переносу точок на поверхню сфери.

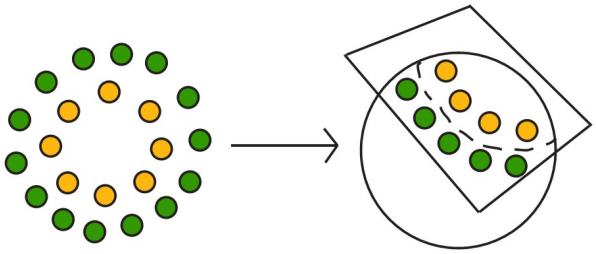

Рисунок 1.7 – Трансформація простору ознак в SVM

Аналіз рис. 1.7 показує в новому просторі більшої розмірності вдалося розділити точки вибірок двох класів площиною. Однак вибір спрямляючого ядра, що здійснює нелінійне перетворення простору ознак, для конкретних вхідних даних досі залишається нетривіальною задачею. Цей алгоритм потребує багато ручних настройок для регуляризації моделі і його продуктивність залежить від складності ядерних функцій перетворення простору ознак і кількості опорних векторів [7].

В праці [18] було показано, що використання дискретного двійкового подання даних забезпечує регуляризуючі та метарегуляризуючі ефекти. Даний підхід є ефективний з точки зору боротьби з перенавчанням і трудомісткості побудови

вирішувальних правил. Схожий підхід розроблено під час розв'язання задач багатокласової класифікації шляхом її заміни еквівалентною множиною двох-класових задач. Одним з найбільш ефективних методів зведення багатокласової класифікації до серії двохкласових є двійкове кодування міток класів кодами, що коригують помилки (Error Correcting Output Codes, ECOC) [6, 7].

Під час використання кодів, що коригують помилки, номер класу записують у вигляді *k*-значного двійкового числа. Для цього здіснюють навчання *k*-класифікаторів, кожен із яких розпізнає один із *k* розрядів номера класу. За результатами розпізнавання вхідного вектора кожним із класифікаторів однозначно відновлюють номер класу, до якого він належить. Якщо окремі класифікатори помиляються, то номер класу відновлюють методом заміни одержаного номера номером класу, що найближчий до одержаного за метрикою Хеммінга (рис. 1.12).

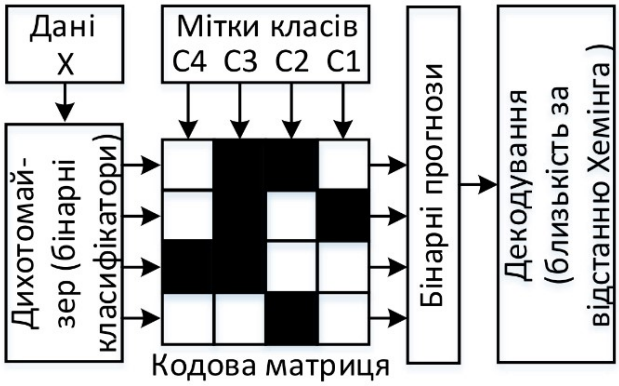

Рисунок 1.8 – Ілюстрація до схеми багатокласової класифікація з використанням двійкового кодування номера класу кодами, що виправляють помилки

На рис. 1.8 показано кодову матрицю, де чорні й білі комірки позначають мітки 0 або 1 для зразків відповідних класів на етапі навчання бінарних класифікаторів. У режимі екзамену цю матрицяю використовують для порівняння прогнозованого коду з кодами класів. Ефективна кодова матриця повинна забезпечувати найбільшу відстань Хеммінга як між рядками кодової матриці, так і між стовбцями. Відстань між рядками забезпечує можливість самовиправлення помилок, а відстань між стовбцями забезпечує некорельованість результатів кожного з бінарних класифікаторів, що навчаються. При цьому існує багато

способів формування номера класу для кодової матриці, однак саме коди, що виправляють помилки, (самокорегувальні коди) забезпечують кодування з необхідними властивостями. Поширеним способом формування самокорегувальних кодів є використання матриць і кодів Адамара [6].

Основним недоліком методу ECOC є ігнорування структури класів під час побудови кодової матриці без можливості оптимізації коду кожного класу в процесі навчання. Крім того, відсутність кодового радіуса для кожного номера класу, що вказує на кратність помилок, які можуть бути виправлені, ускладнює виявлення викидів або новизни в даних.

Таким чином, питання вибору оптимальних в інформаційному сенсі параметрів моделей розпізнавання вторгнень є актуальним. Його вирішення ускладнене неповною визначеністю даних, що обумовлена нестаціонарністю процесів формування шкідливого трафіку і обмеженим обсягом актуальних розмічених навчальних даних. При цьому одним з найбільш перспективних підходів до класифікаційного аналізу шкідливого трафіку, описаного високорівневими ознаками, є застосування ідей і методів інформаційно-екстремальної інтелектуальної технології. Інформаційно-екстремальний класифікатор є обчислювально ефективним та характеризується ефективністю машинного навчання за умов обмеженого обсягу розмічених навчальних вибірок.

1.3 Формалізована постановка задачі

Нехай дано два набори даних KISA Data Challenge 2019 та Virus-Share Dataset, які містять зразки шкідливого та нешкідливого програмного забезпечення для платформи IoT з операційними системами Windows та Linux. Алфавіт класів містить чотири класи : шкідливий код для Linux; нешкідливий код для Linux; шкідливий код для Windows; нешкідливий код для Windows.

Дано структурований вектор параметрів функціонування моделі розпізнавання шкідливого програмного забезпечення, який у загальному випадку має структуру

$$g = < e_1, ..., e_{\xi_1}, ..., e_{\Xi_1}, f_1, ..., f_{\xi_2}, ..., f_{\Xi_2} >, \quad (1.1)$$

$$\Xi_1 + \Xi_2 = \Xi \ ,$$

де $< e_1, ..., e_{\xi_1}, ..., e_{\Xi_1} >$ – генотипні параметри функціонування системи інтелектуального аналізу спостережень, які впливають на параметри формування ознакового опису спостережень;

$< f_1, ..., f_{\xi_2}, ..., f_{\Xi_2} >$ – фенотипні параметри функціонування системи, які прямо впливають на точність детектування шкідливого програмного забезпечення.

При цьому відомі обмеження на відповідні параметри функціонування :

$$R_{\xi_1}(e_1, ..., e_{\xi_1}, ..., e_{\Xi_1}) \le 0 \, ; \; R_{\xi_2}(f_1, ..., f_{\xi_2}, ..., f_{\Xi_2}) \le 0 \, .$$

Потрібно на етапі навчання моделі розпізнавання шкідливого програмного забезпечення оптимальний вектор $\{g_\xi^* \mid \xi = \overline{1, \Xi_1 + \Xi_2}\}$ параметрів функціонування, який забезпечує максимум усередненого за алфавітом класів інформаційного критерію ефективності навчання

$$\overline{E}^* = \frac{1}{K} \sum_{k=1}^{K} \max_{\{s\}} E_k, \quad (1.2)$$

де E_k – інформаційний критерій навчання вирішальних правил розпізнавати реалізації класу X_k^o; K – кількість класів розпізнавання; $\{s\}$ – множина кроків машинного навчання.

При функціонуванні системи розпізнавання безпосередньо в робочому режимі необхідно забезпечити максимальну точність класифікації двійкових виконуваних файлів програмного забезпечення.

РОЗДІЛ 2
ТЕХНОЛОГІЯ МАШИННОГО НАВЧАННЯ ДЛЯ РОЗПІЗНАВАННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ПРИСТРОЯХ ІНТЕРНЕТУ РЕЧЕЙ

2.1 Модель і метод навчання класифікатора шкідливого програмного забезпечення

Аналіз двійкових файлів програмного забезпечення пропонується шляхом його перетворення у кольорові зображення (рис. 2.1). Для цього двійковий файл перетворюється в послідовність байтів, що групуються по 3 для формування пікселів кольорового RGB-зображення. При цьому для побудови 2D кольорового зображення необхідно обрати його ширину, яка обирається з табл. 2.1 [3]. Для використання ефективної ініціалізації вагових коефіцієнтів великих згорткових нейронних мереж згідно принципу переносу знань (transfer learning), потрібно виконати зміну розміру зображення. Розмір зображення повинен бути рівним одному з тих розмірів, які використовувалися під час попереднього навчання нейромережі на наборі ImageNet.

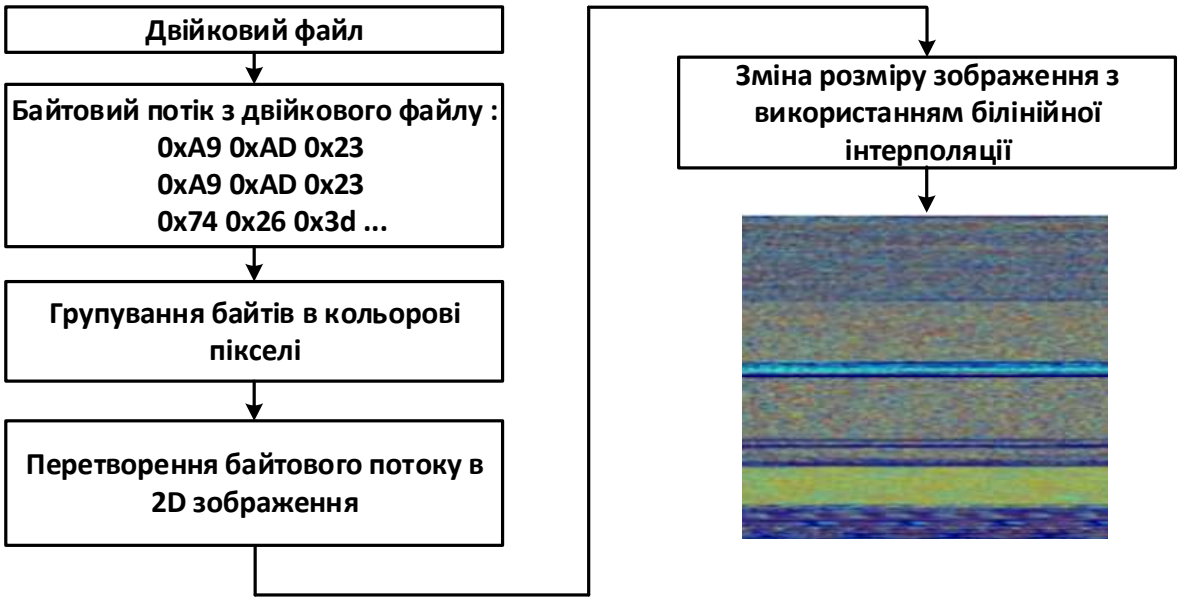

Рисунок 2.1 – Етапи запропонованого методу навчання

Таблиця 2.1 – Залежність рекомендованої ширини зображення від кількості сформованих пікселів

Кількість пікселів	Ширина зображення
Від 1 до 10	32
Від 10 до 30	64
Від 30 до 60	128
Від 60 до 100	256
Від 100 до 200	384
Від 200 до 1000	512
Від 1000 до 1500	1024
Більше ніж 1500	2048

Для експериментів і перевірки концепції як основу для екстрактора ознак пропонується використати згорткову мережу загального призначення MobileNet без повнозвязних вихідних шарів з коефіцієнтом ємності, що дорівнює 0,25 [4]. При цьому роздільну здатність входу мережі можна вибрати з ряду : 128x128, 160x160, 192x192 та 224x224. Структуру запропонованої моделі класифікаційного аналізу даних показано на рис. 2.2*а*. Для визначення ефективності запропонованого підходу варто дослідити ефективність даного екстрактор ознак з класичними вихідними шарами. На рис. 2.2*б* показано базову (baseline) модель для порівння ефективності з запропонованим підходом.

MobileNet екстрактор ознак (backbone)
Шар Global Average Pooling 2D
Шар Dropout (rate=0.5)
Повнозв’язний шар, Dense (128 вузлів)
Сігмоїдний шар, Sigmoid
Округлючий шар
Радіально-базисний шар

а

MobileNet екстрактор ознак (backbone)
Шар Global Average Pooling 2D
Шар Dropout (rate=0.5)
Повнозв’язний шар, Dense (128 вузлів)
Сігмоїдний шар, Sigmoid
Повнозв’язний шар
Нормуючий Softmax шар

б

Рисунок 2.2 – Структура моделі класифікатора : *а* – запропонована модель; *б* – модель з класичними вихідними шарами для порівняння ефективності

Запропонований метод складається з 5 основних етапів (фаз) (рис. 2.3). Перший етап включає в себе різноманітні операції з даними, такі як повороти, шкалювання, зміну яскравості і додавання шуму. Наступні 3 етапи – це безпосередньо навчання класифікатора. На останньому етапі вирішальні правила коригуються враховуючи дисперсію спостережень всередині класів в двійковому просторі Хеммінга [8].

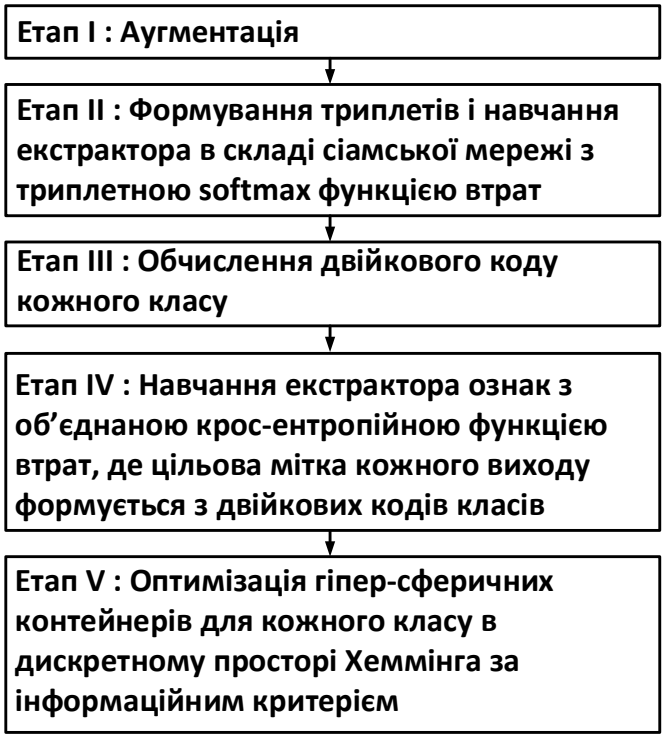

Рисунок 2.3 – Етапи запропонованого методу навчання

Другий етап навчання моделі аналізу зображення проводиться із застосуванням адаптивних алгоритмів зворотного поширення помилки, з яких найбільш популярним є Adam [8]. На вхід моделі подається міні-пакет, в якому *M* зображень кожного класу. Триплетна функція втрат розраховується за формулою

$$L = -\log \frac{\exp(\|f(x_a) - f(x_{ep})\|)}{\exp(\|f(x_a) - f(x_{ep})\|) + \exp(\|f(x_a) - f(x_{shn})\|)} \qquad (2.1)$$

де f(x) – функція, що описує екстрактор та встановлює залежність між вхідним зображенням та вектором виходу сигмоїдного шару; x_a – зображення, що

обрано випадковим чином з міні-пакету; x_{ep} – найближчий сусід з міні-батчу, що належить до того ж самого класу, тобто

$$x_{ep} = \arg \min_{x:C(x)=C(x_a)} \left\| f(x_a) - f(x) \right\|, \tag{2.2}$$

де $C(x)$ – функція що повертає клас зображення; x_{shn} – зразок зображення з міні-пакету, що є найближчим серед зразків протилежних класів, проте знаходиться далі ніж зразок, тобто

$$x_{shn} = \arg \min_{\substack{C(x)\neq C(x_a) \\ x:\left\| f(x_a)-f(x) \right\| > \left\| f(x_a)-f(x_p) \right\|}} \left\| f(x_a) - f(x) \right\|. \tag{2.3}$$

Для підвищення ефективності апроксимації дискретного подання даних пропонується додати регуляризуючу складову, що враховуватиме помилку округлення вихідного сигналу мережі до двійкового коду. В результаті функція втрат набуде наступного вигляду [8]

$$\begin{gathered} L = -\log \frac{\exp(\left\| f(x_a) - f(x_{ep}) \right\|)}{\exp(\left\| f(x_a) - f(x_{ep}) \right\|) + \exp(\left\| f(x_a) - f(x_{shn}) \right\|)} + \\ +\lambda\left(f(x_a)^T (e - f(x_a)) + f(x_{ep})^T (e - f(x_{ep})) + f(x_{shn})^T (e - f(x_{shn})) \right), \end{gathered} \tag{2.4}$$

де $e = [1, 1, ..., 1]^T$ – одинична матриця-стовбець; λ – коефіцієнт регуляризації.

Наступна фаза потрібна для перетворення в бінарну форму вихідного вектора відповідно до принципів самокоректуючих кодів (error-correcting output codes), однак з урахуванням внутрішньої структури класів та відношень між зразками різних класів. Для цього навчальна вибірка $\{x_{z,s} \mid z = \overline{1,Z}, s = \overline{1,n_z}\}$, що містить n_z зразків z-го класу кодується дискретним поданням $\{b_{z,s,i} \mid z = \overline{1,Z}, s = \overline{1,n_z}, i = \overline{1,N}\}$ з

розмірністю N та обчислюється еталонний двійковий вектор кожного класу. Двійкове кодування здійснюється шляхом подавання зображень з множини $\{x_{z,s}\}$ на вхід моделі і округлення виходу сігмоїдного шару до цілого числа

$$b_{z,s,i} = \begin{cases} 1, & \text{if } f(x_{z,s,i}) > 0.5; \\ 0, & \text{otherwise.} \end{cases}, \tag{2.5}$$

Двійковий еталонний вектор b_z для z-го класу можна визначити шляхом порозрядного порівняння частоти двійкових одиниць в z-му класі з фоновою частотою одиниць в навчальній вибірці [9]

$$b_{z,i} = \begin{cases} 1, & \text{if } \frac{1}{n_z}\sum_{s=1}^{n_z} b_{z,s,i} > \frac{1}{Z}\sum_{c=1}^{Z}\frac{1}{n_c}\sum_{s=1}^{n_c} b_{c,s,i}; \\ 0, & \text{otherwise.} \end{cases}, \tag{2.6}$$

Обчислений еталонний вектор z-го класу b_z можна використовувати як розмітку (label) зразка для подальшого навчання з використанням Joint Binary Cross Entropy Loss, яка для кожного вхідного зразка *x* може бути обчислена за формулою

$$L = -\sum_{i=1}^{N}\left(b_i \log f_i(x) + (1 - b_i)\log(1 - f_i(x))\right), \tag{2.7}$$

де $f_i(x)$ – значення i-го виходу сігмоїдного шару для вхідного зображення x;

b_i – значення i-го розряду еталонного вектора класу, до якого належить зображення *x*.

Остання фаза машинного навчання пов’язана з оптимізацію радіусу контейнерів за інформаційним критерієм для врахування меж відхилення двійкового подання спостережень кожного класу від відповідних еталонних векторів

$$E_z^* = \max_{\{d\}} E_z(d), \qquad (2.8)$$

де $\{d\} = \{0,1,...,\left(\sum_i b_{z,i} \oplus b_{c,i} - 1\right)\}$ – набір концентричних радіусів з центром b_z розподілу даних в z-му класі, які обчислюються за правилом (2.6);

E_z – інформаційний критерій для z-го класу, який є функцією від характеристик точності [10].

Таким чином, для побудови інформативного ознакового опису і завадозахищених вирішувальних правил варто будувати модель шляхом поєднання ідей та методів сіамських мереж та інформаційно-екстремального навчання.

2.2 Критерій оптимізації вирішувальних правил

Оцінка Важливим питанням інформаційного синтезу моделей аналізу даних є оцінка функціональної ефективності процесу навчання, яка визначає максимальну асимптотичну достовірність рішень, що приймаються на екзамені. Як критерій функціональної ефективності можуть використовуватися різні інформаційні критерії, які задовольняють таким властивостям інформаційних мір [9, 11]:

- інформаційна міра є величина дійсна і знакододатна як функція від імовірності;
- кількість інформації для детермінованих змінних ($p_i = 1$ або $p_i = 0$) дорівнює нулю;
- інформаційна міра має екстремум при значенні ймовірності $p_i = \frac{1}{m}$, де m – кількість якісних ознак розпізнавання.

Серед інформаційних мір для оцінки функціональної ефективності моделі аналізу даних перевагу слід віддавати статистичним логарифмічним критеріям, які дозволяють працювати з навчальними вибірками відносно малих обсягів [6]. Серед таких критеріїв найбільшого використання знайшла ентропійна міра. Подамо нормований ентропійний критерій ефективності навчання моделі розпізнавати спостереження класу X_m^o у вигляді :

$$E_m^{(k)} = \frac{I_m^{(k)}}{I_{\max}^{(k)}} = \frac{H_m^{(k)} - H_m^{(k)}(\gamma)}{H_m^{(k)}}, \qquad (2.9)$$

де $I_m^{(k)}$ – кількість умовної інформації, що обробляється на k-му кроці навчанні моделі розпізнавати спостереження класу X_m^o;

$I_{\max}^{(k)}$ – максимально можлива кількість умовної інформації, одержаної на k-му кроці навчанні моделі розпізнавати спостереження одного із класів із заданого алфавіту $\{X_m^o\}$, $m = \overline{1, M}$;

$$H_m^{(k)} = -\sum_{l=1}^{M} p(\gamma_{l,k}) \log_2 p(\gamma_{l,k}) \qquad (2.10)$$

апріорна (безумовна) ентропія, що існує на k-му кроці навчання моделі розпізнавати реалізації класу X_m^o;

$$H_m^{(k)}(\gamma) = -\sum_{l=1}^{M} \sum_{m=1}^{M} p(\gamma_{l,k}) p(\mu_{m,k} / \gamma_{l,k}) \log_2 p(\mu_{m,k} / \gamma_{l,k}) \qquad (2.11)$$

– апостеріорна (умовна) ентропія, що характеризує залишкову невизначеність після k-го кроку навчання системи розпізнавати реалізації класу X_m^o;

$p(\gamma_{l,k})$ – безумовна ймовірність прийняття на k-му кроці навчання гіпотези $\gamma_{l,k}$;

$p(\mu_{m,k} / \gamma_{l,k})$ – апостеріорна ймовірність прийняття на k-му кроці навчанні рішення $\mu_{m,k}$ за умови, що прийнята гіпотеза $\gamma_{l,k}$.

Для двохальтернативної системи оцінок ($M = 2$) і рівноймовірних гіпотез, що характеризує найбільш важкий у статистичному сенсі випадок прийняття рішень, після відповідної підстановки ентропій (2.2) і (2.3) у вираз (2.1) та заміни

відповідних апостеріорних ймовірностей на апріорні за формулою Байєса ентропійний критерій набуває вигляду [11]

$$E_m^{(k)} = 1 + \frac{1}{2}\left(\frac{\alpha_m^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \log_2 \frac{\alpha_m^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} + \right.$$

$$+ \frac{\beta_m^{(k)}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} \log_2 \frac{\beta_m^{(k)}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} + \frac{D_{1,m}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} \log_2 \frac{D_{1,m}(d)}{D_{1,m}^{(k)}(d) + \beta_m^{(k)}(d)} +$$

$$\left. + \frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \log_2 \frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d) + D_{2,m}^{(k)}(d)} \right), \qquad (2.12)$$

де $\alpha_m^{(k)}(d)$ – частота помилок першого роду (false postive rate) – точністна характеристика рішення на k-му кроці навчання;

$\beta_m^{(k)}(\mathrm{d})$ – частота помилок другого роду (false negative rate);

$D_{1,m}^{(k)}(\mathrm{d})$ – чутливість (true positive rate, sensitivity) або перша достовірність;

$D_{2,m}^{(k)}(\mathrm{d})$ – специфічність (true negative rate, specificity) або друга достовірність;

d – дистанційна міра, яка визначає радіуси гіперсферичних контейнерів, побудованих в радіальному базисі простору Хеммінга.

Оскільки точністні характеристики є функціями відстані від еталонних від геометричних центрів контейнерів (кодів) відповідних класів розпізнавання, то критерій (2.4) слід розглядати як нелінійний і взаємно-неоднозначний функціонал від точністних характеристик, що потребує знаходження в процесі навчання робочої (допустимої) області для його визначення. Робочу область визначення функції інформаційного критерію можна задати у вигляді системи нерівностей

$$\begin{cases} \beta_m^{(k)}(\mathrm{d}) < 0{,}5; \\ \alpha_m^{(k)}(\mathrm{d}) < 0{,}5; \\ \mathrm{d} < \mathrm{d}(\mathrm{x_c} \oplus \mathrm{x_m}), \end{cases}$$

де $\mathrm{d}(\mathrm{x_c} \oplus \mathrm{x_m})$ – відстань Хемінга між центрами сусідніх класів.

Таким чином, двійкові еталонні вектори класів використовуються по аналогії з самокоректуючими кодами Хемінга, а радіус контейнера характеризує максимальну кількість помилок, які можуть бути виправлені після отримання повідомлення по каналу зв'язку з завадами. Інформаційний критерій характеризує міру зняття невизначеності після отримання повідомлення з заданого алфавіту. Оптимізація радіусу гіперсферичних контейнерів дозволяє підвищити ефективність прийняття рішень за умов перетину класів з різними габаритами розподілу спостережень та виявляти новизну в даних у випадку появи спостережень, що не потрапляють в контейнер жодного класу. При цьому критерієм оптимізації є логарифмічний інформаційний критерій, що підвищує узагальнюючу здатність моделі та забезпечує її робастність за найскладніших у статистичному розумінні випадках.

Оцінку ефективності навченої моделі здійснюють на зовнішній незалежній розміченій вибірці, яку називають тестовою. Прогнозовані порівнюють з правильними (ground truth) мітками, що наносилися експертом, для обчислення метрик продуктивності моделі.

Метрики продуктивності моделі обчислюються на основі підрахунку результатів статистичних тестів. Для одного тестового зразка можливі 4-ри можливі результати статистичного тесту [7, 12] :

1) істинно позитивний (True Positive,TP) – якщо розпізнано об’єкт інтересу, який дійсно присутній в даному тестовому спостереженні;

2) хибно негативний (False Negative, FN) – якщо не було розпізнано об’єкт інтересу, який дійсно присутній в даному тестовому спостереженні;

3) істинно негативний (True Negative, TN) – якщо не було розпізнано об’єкт інтересу, який дійсно не присутній в даному тестовому спостереженні;

4) хибно позитивні (False Positive, FN) – якщо розпізнано об’єкт інтересу, який дійсно не присутній в даному тестовому спостереженні.

Важливими характеристиками моделі аналізу даних вважаються чутливість, специфічність, точність, частота помилок першого та другого роду, правильність, збалансована правильність та F1-міра. На основі цих характеристик приймають рішення про придатність моделі до практичного використання.

Чутливість (Sensitivity, або True Positive Rate або Recall) – кількість істинно позитивних результатів, поділена на реальну кількість позитивних зразків, тобто

$$\text{Sensitivity} = \text{Recall} = TP / (TP+FN).$$

Специфічність (або True Negative Rate) – кількість істинно негативних результатів, поділена на реальну кількість негативних зразків, тобто

$$\text{Specificity} = TNR = TN / (FP+TN).$$

Точність (або Precision) – кількість істинно позитивних результатів поділена на загальну кількість позитивних прогнозів, тобто

$$\text{Precision} = TP/(TP+FP).$$

Частота помилок першого роду (або False Positive Rate) – кількість хибно позитивних поділена на загальну кількість істинно негативних, тобто

$$FPR = FP/(FP+TN).$$

Частота помилок другого роду (або False Negative Rate) – кількість хибно негативних, поділена на загальну кількість істинно позитивних, тобто

$$FNR = FN/(FN+TP).$$

Правильність (або Accuracy) – частка правильних прогнозів від загальної кількості прогнозів, тобто

$$\text{Accuracy} = (TP+TN) / (TP+TN+FP+FN)$$

Однак правильність може бути адекватною метрикою тільки якщо кількість зразків кожного класу однакова, тобто класи є збалансованими.

Збалансована правильність (або Balanced Accuracy) – середнє арифметичне між чутливістю та специфічністю, тобто

$$\text{Balanced Accuracy} = (\text{Sensitivity} + \text{Specificity})/2$$

Таким чином, існує багато метрик оцінки продуктивності моделей класифікаційного аналізу. Серед цих метрик продуктивності найбільш інформативними є Precision та Balanced Accuracy.

2.3 Результати фізичного моделювання

Запропоновані модель і метод навчання перевірялися як на загальнодоступних, так і на самостійно зібраних наборах даних. Оскільки розглядається алгоритм робастний до платформи пристроїв інтернету речей, то вибіркові дані містять двійкові файли для популярних архітектур з операційною системою Windows та Linux. Вибірка даних для платформи Windows (тобто PE-файли) сформована з набору даних KISA Data Challenge 2019 [3] і містить 18 000 тисяч зразків шкідливого та 12 000 нешкідливого програмного забезпечення, з яких по 5000 зразків відібрано для тестування моделі. Вибірка даних для платформи Linux (тобто ELF-файли) складається з 10 000 зразків щкідливого програмного забезпечення, отриманих з набору даних Virus-Share Dataset [3], та 10 000 нешкідливих зразків зібраних з директорій /bin та /usr/bin на одноплатному комп’ютері Raspberry pi та Odroid XU з різними Linux-операційними системами. При цьому в тестову вибірку шкідливих та нешкідливих програм було відібрано по 2000 зразків. Роздільну здатність зображень під час кодування виконуваних файлів в зображення дорівнює 128x128 пікселів.

Розглянемо результати машинного навчання класифікатора шкідливого програмного забезпечення на сформованих навчальних даних. На рис. 2.4 показано зміни точності моделі на тестовій (test_acc) і навчальній (train_acc) вибірках. Кожен

міні-батч містить 128 зображення, а коефіцієнт швидкості навчання дорівнює 0,0001.

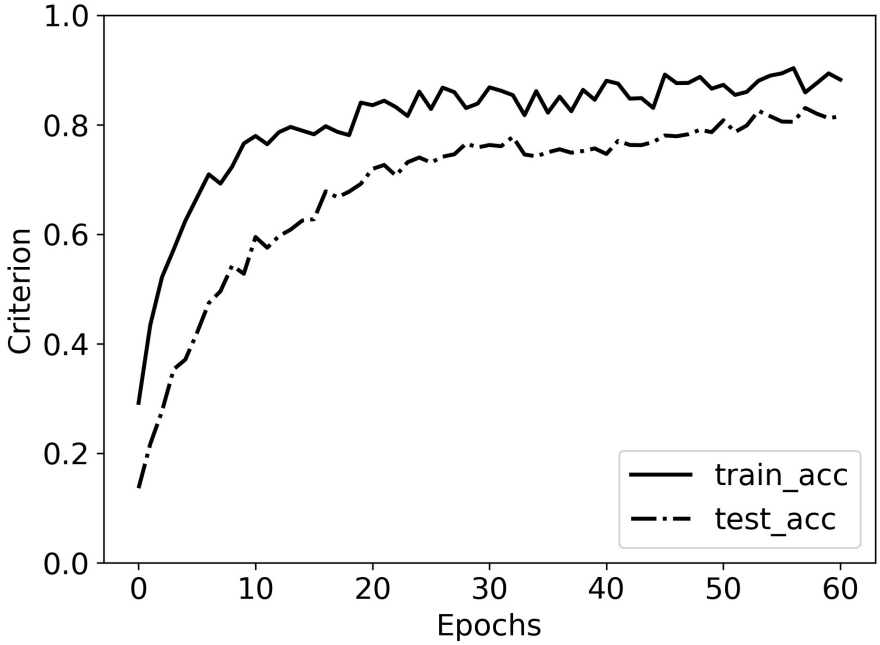

Рисунок 2.4 – Графік залежності точності під час навчання класифікатора дефектів від кількості епох навчання в рамках традиційного підходу

Аналіз рис. 2.4 показує, що підвищення точності зупиняється після 43 епох навчання з максимальним значенням точності результуючої моделі, що дорівнює 81,5% на стестовій вибірці. При цьому відстань між кривою на тестовій і на навчальні вибірці свідчить про наявність невеликого ефекту перенавчання.

На рис. 2.5 показано результати навчання запропонованого екстрактора ознак з сігмоїдним шаром і функцією втрат (2.1). При цьому кожні 5 епох відбувається побудова інформаційно-екстремальних вирішувальних правил і обчислення усередненого за алфавітом класів інформаційного критерію (2.12) на тестовій (test_info_cri) та навчальній (train_info_cri) вибірках. Після 30 епох навчання модель було збережено для наступного експерименту.

Аналіз рис. 2.5 показує, що з 10 епохи зростання точності значно уповільнилося, а після 40 епох було досягнуто максимальну точність, що дорівнює 87,3%, що на 7,11% більше ніж в базової (baseline) моделі. Однак після 40 епох навчання, подальше покращення практично припинилося. При цьому тестові і

навчальні криві майже співпадають, що свідчить про високу узагальнюючу здатність отриманих вирішувальних правил.

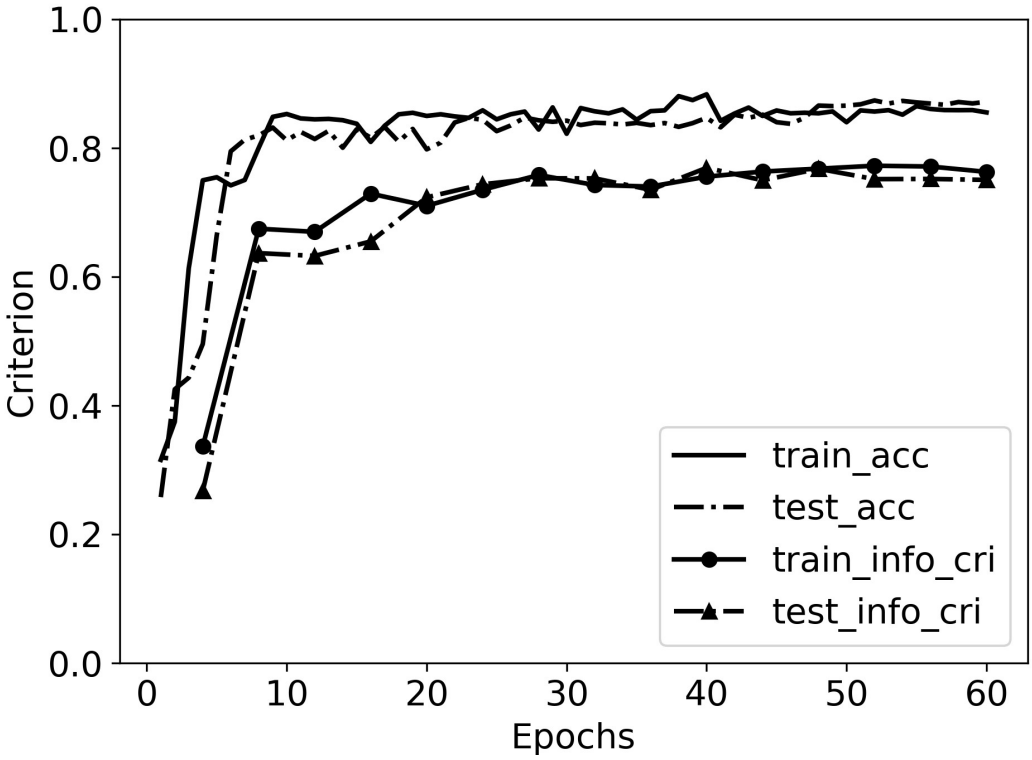

Рисунок 2.5 – Графік залежності точності та інформаційного критерію від кількості епох навчання в рамках запропонованого підходу з функцією втрат (2.1)

Усереднє за алфавітом класів значення інформаційного критерію дорівнює 0,751 а середня відстань між центрами гіперсферичних контейнерів дорівнює 15 кодових одиниць, а середній радіус контейнерів класів рівний 9 кодових одиниць.

На рис. 2.6 показано подальші результати навчання моделі, збереженої на попередньому етапі після 30 епох навчання. Наступне навчання здійснюється також 30 епох, але з використанням функції втрат (2.7) з метою зменшення внутрішньокласової дисперсії в просторі Хемінга [8].

Аналіз рис. 2.6 показує, що використання функції joint binary cross-entropy loss (2.7) забезпечує зростання точності на 10% (точність 96,1%) порівняно з попереднім етапом. При цьому усереднене за алфавітом класів значення інформаційного критерію (2.12) дорівнює 0,91.

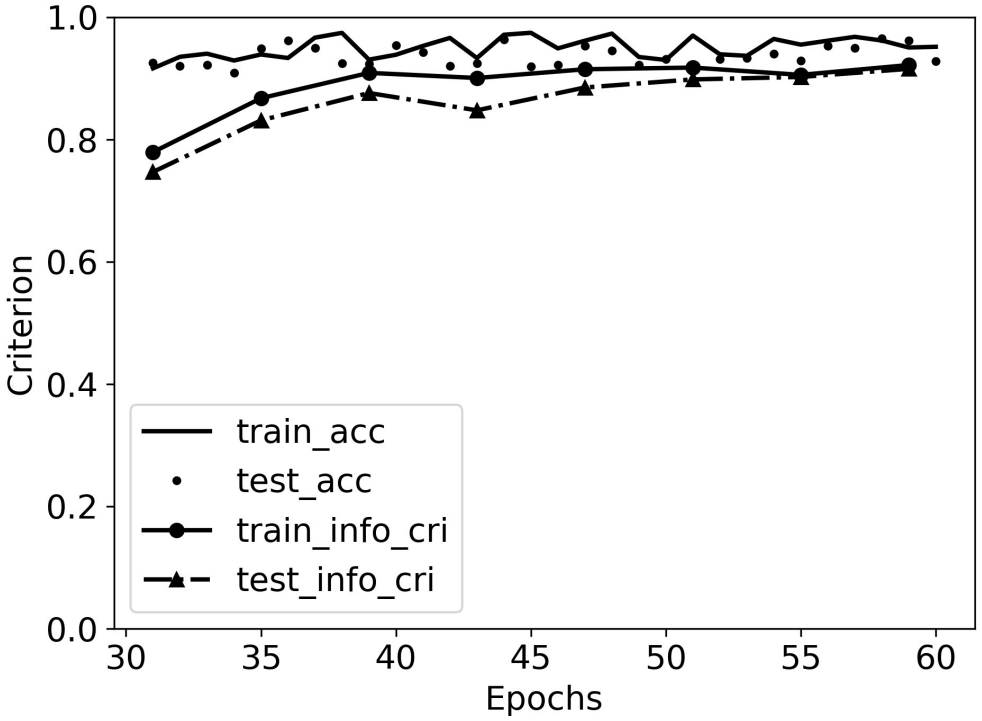

Рисунок 2.6 – Графік залежності точності та інформаційного критерію від кількості епох навчання в рамках запропонованого підходу з функцією втрат (2.7)

Таким чином мультифазне навчання з інформаційно-екстремальними вирішувальними правилами забезпечує підвищення точності розпізнавання шкідливого програмного забезпечення порівняно з традиційним підходом на 18% за туж кількість епох навчання. При цьому усереднє за алфавітом класів значення інформаційного критерію дорівнює 0,751 а середня відстань між центрами гіперсферичних контейнерів дорівнює 16 кодових одиниць, а середній радіус контейнерів класів рівний 8 кодових одиниць. Це свідчить про високу завадозахищеність вирішувальних правил. Причиною може бути регуляризаційний та мета-регуляризаційний ефект внаслідок поєднання принципів геометричного та інформаційного підходів.

Таким чином, запропоновані алгоритми дозволяють отримати вирішальні правила, достовірність яких прийнятна для практичного використання.

ВИСНОВКИ

1. Наукова новизна одержаних результатів :

– вперше запропоновано метод візуального подання виконуваних кодів програмного забезпечення та модель і метод навчання багатошарової згорткової мережі з інформаційно-екстремальними радіально-базисними вирішувальними правилами для розпізнавання шкідливого програмного забезпечення в пристроях інтернету речей незалежно від їх операційної системи і платформи;

– здійснена перевірку ефективності запропонованого підходу на відкритих даних KISA Data Challenge 2019 та Virus-Share Dataset шляхом порівняння результатів навчання за запропонованою та традиційною моделями та схемами навчання. Отримана модель класифікатора виконуваних програмних кодів забезпечує прийнятну для практичного використання точність класифікації на тестовій вибірці, що становить 96,1% і перевищує результат традиційної схеми навчання з softmax вихідним шаром на 18 %.

2. Практична цінність отриманих результатів для систем розпізнавання шкідливого програмного забезпечення полягає у розробці нового універсального та уніфікованого методу синтезу алгоритму розпізнавання шкідливого програмного забезпечення, що забезпечує робастність до апаратної та програмної платформи пристрою інтернету речей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Jiawei Su. Lightweight Classification of IoT Malware based on Image Recognition [Text] / J. Su, D. V. Vasconcellos; S. Prasad; D. Sgandurra; Y. Feng; K. Sakurai // 42nd IEEE Annual Computer Software and Applications Conference (COMPSAC). – Tokyo, Japan, 22 June 2018. – 7 pp. – DOI : 10.1109/COMPSAC.2018.10315

2. Ke He. Malware Detection with Malware Images using Deep Learning Techniques [Text] / Ke He, Dong Seong Kim // 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications. – Rotorua, New Zealand, New Zealand, 31 October 2019. – P. 95-102. – DOI: 10.1109/TrustCom/BigDataSE.2019.00022

3. Chanwoong Hwang. Platform-Independent Malware Analysis Applicable to Windows and Linux Environments / Chanwoong Hwang, Junho Hwang, Jin Kwak and Taejin Lee // https://www.mdpi.com/journal/electronics. – V. *9*, 2020. – 18 pp. – DOI : 10.3390/electronics9050793

4. Yousefi-Azar, M. Autoencoder-based feature learning for cyber security applications [Text] / M. Yousefi-Azar, V. Varadharajan, L. Hamey, U. Tupakula // Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN). – Anchorage, Alaska, USA, 14–19 May 2017. – 3854–3861 p.

5. Szegedy, C. Going deeper with convolutions [Text] / C. Szegedy, W.Liu, Y. Jia, P. Sermanet et al.// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – Boston, MA, 7–12 June, 2015. – 1–9 p.

6. Москаленко В. В. “Моделі і методи інтелектуального аналізу багатовимірних даних да умов апріорної невизначеності”: монографія / В. В. Москаленко. – Суми: Видавництво СумДУ, 2019. – 184 с.

7. Kim, S. Deep learning of support vector machines with class probability output networks [Text] / S. Kim, Z. Yu, R. Man Kil, M. Lee // Neural Networks. – 2015. – Vol. 64. – P. 19–28.

8. Moskalenko, V. Sewer Pipe Defects Classification Based on Deep Convolutional Network with Information-extreme Error-correction Decision Rules / A. S. Moskalenko, V.V. Moskalenko, M. O. Zaretskyi, V. Lysyuk //Communications in Computer and Information Science (CCIS-2020) – Springer : Cham, Switzerland. – 2020. – 20 p. – DOI : 10.1007/978-3-030-61656-4_16

9. Dovbysh, A. S. Information-extreme learning algorithm for a system of recognition of morphological images in diagnosing oncological pathologies [Text] / A. S. Dovbysh, M. S. Rudenko // Cybernetics and Systems Analysis. – 2014. – Vol. 50, I. 1. – P. 157–162.

10. Moskalenko, V. Optimizing the parameters of functioning of the system of management of data center IT infrastructure [Text] / V. Moskalenko, S. Pimonenko // Eastern-European Journal of Enterprise Technologies. – 2016. – Vol. 5, I. 2 (83). – P. 21–29.

11. Kudryavtsev A. M. Multi-Layer Model And Training Method For Malware Traffic Detection Based On Decision Tree Ensemble / V. V. Moskalenko, A. S. Moskalenko, M. O. Zaretsky, A. M. Kudryavtsev, V. A. Semashko // Radioelectronic and Computer Systems – No 2. – 2020. – P. 92-101.

12. Кудрявцев А. М. Метод самонавчання згорткового екстрактора ознак для розпізнавання багатоканальних діагностичних зображень / В. В. Москаленко, А. С. Москаленко, А. М. Кудрявцев // XIV International scientific conference «Intellectual systems of decision-making and problems of computational intelligence (ISDMCI’2018)», Україна. – 2018. – С. 258-260.