УДК 002.6, 002.001; 002:001.8
УКПП
№ держреєстрації 0122U000782
Інв. №

Міністерство освіти та науки України
Сумський державний університет (СумДУ)
40007, м. Суми, вул. Римського-Корсакова, 2
тел. (0542) 33-54-79 факс (0542) 33-54-79

ЗАТВЕРДЖУЮ
Проректор з наукової роботи
д-р фіз.-мат. наук, професор

_____ А.М. Чорноус

ЗВІТ
ПРО НАУКОВО-ДОСЛІДНУ РОБОТУ
Інформаційна технологія забезпечення резільєнтності систем штучного інтелекту для захисту кібер-фізичних систем

РОЗРОБЛЕННЯ МОДЕЛЕЙ І МЕТОДІВ ВИМІРЮВАННЯ ТА СЕРТИФІКАЦІЇ РЕЗІЛЬЄНТНОСТІ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ ДЛЯ ЗАХИСТУ КІБЕР-ФІЗИЧНИХ СИСТЕМ
(проміжний)

Керівник НДР
доцент, канд. техн. наук В.В. Москаленко

2022

Рукопис завершено 16 грудня 2022 р.

Результати роботи розглянуто науковою радою СумДУ, протокол від 22.12.2022 № 7

СПИСОК АВТОРІВ

Керівник НДР,

канд. техн. наук, провідн. наук. співроб.	(16.12.2022)	Москаленко В. В. (вступ, висновки, розділ 1, розділ 2)

Відповідальний виконавець,

канд. техн. наук, старш. наук. співроб.	(16.12.2022)	Москаленко А. С. (підрозділ 1.4, 2.4)

Виконавці:

канд. техн. наук, старш. наук. співроб.	(16.12.2022)	Бойко О. В. (підрозділ 2.2)
аспірант, мол. наук. співроб.	(16.12.2022)	Мироненко М. І. (підрозділ 1.3)
канд. техн. наук, старш. наук. співроб.	(16.12.2022)	Нагорний В. В. (підрозділ 2.2)
канд. техн. наук, старш. наук. співроб.	(16.12.2022)	Коробов А. Г. (підрозділи 1.3 та 2.4)
мол. наук. співроб.	(16.12.2022)	Зарецький М. О. (підрозділ 1.3)

РЕФЕРАТ

Звіт про НДР: 117 с., 10 табл., 22 рис., 109 джерел.

ВІДМОВИ, ВИТОНЧЕНА ДЕГРАДАЦІЯ, ДРЕЙФ КОНЦЕПЦІЙ, КЛАСИФІКАТОР ЗОБРАЖЕНЬ, МАШИННЕ НАВЧАННЯ, ПРОТИБОРЧІ АТАКИ, РЕЗІЛЬЄНТНІСТЬ СИСТЕМИ, РОБАСТНІСТЬ СИСТЕМИ, СИСТЕМА ШТУЧНОГО ІНТЕЛЕКТУ.

Об’єкт дослідження – процеси інтелектуального аналізу даних в середовищі кібер-фізичних систем в умовах комплексного впливу деструктивних збурень. Мета роботи – розроблення моделей та методів забезпечення резільєнтності систем штучного інтелекту до деструктивних впливів, притаманних кіберфізичним системам різного призначення. Методи дослідження – аналіз літературних джерел, методи теорії надійності і безпеки складних систем, методи нейромережевого та інформаційно-екстремального моделювання.

Проаналізовано вразливості та джерела загроз для систем штучного інтелекту, а також підходи щодо підвищення їх резільєнтності. Запропоновано критерії та методи оцінювання резільєнтності систем інтелектуального аналізу даних до інжекції несправностей, протиборчих атак та дрейфу концепцій. Розроблено критерії та методи оцінювання резільєнтності інтелектуальної системи аналізу даних. Запропоновано множину принципів побудови резільєнтних систем класифікації зображень та досліджено вплив архітектурних рішень та параметрів на резільєнтність.

Результати виконання роботи впроваджено в навчальний процес під час підготовки лекційних курсів з навчальних дисциплін «Мови програмування інтелектуальних систем» і «Наука про дані» та під час написання двох магістерських робіт. Також отриманий досвід використано для надання послуг “Розроблення програмного забезпечення модуля машинного зору для дистанційно-керованої мобільної платформи” в рамках виконання договору № 52.17-2022.СП/01 від 22.11.2022.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

СШІ – системи штучного інтелекту;

ІЕІ-технологія – інформаційно-екстремальна інтелектуальна технологія;

КФЕ – критерій функціональної ефективності;

ВСТУП

Кіберфізичні системи проникають у всі сфери життєдіяльності людини: виробництво, будівництво, транспорт, енергетику, медицину, оборону тощо, де забезпечують нові функціональні можливості. При цьому технології штучного інтелекту стають невід'ємною їх частиною і виконують функції кіберзахисту, механізмів адаптації, функціональної діагностики, предиктивного обслуговування тощо. Проте самі технології штучного інтелекту мають ряд вразливостей і їх використання в складі об'єктів критичної інфраструктури в умовах впливу деструктивних збурень може призвести до значних матеріальних збитків і людських жертв. Метою проєкту є розроблення моделей та методів забезпечення резільєнтності систем штучного інтелекту (СШІ) до деструктивних впливів, притаманних кіберфізичним системам різного призначення. Під резільєнтністю СШІ розуміється їх здатність стабільно надавати свої послуги в надійний спосіб навіть при зовнішніх та внутрішніх змінах.

У відомих дослідженнях під резільєтністю, як правило, розуміють окремі її аспекти. Найчастіше під резільєнтністю СШІ мають на увазі її робастність і саме робастність вимірюють та верифікують. Однак в рамках такого вузького розуміння резільєнтності ігноруються такі важливі аспекти резільєнтності як здатність до витонченої деградації, здатність до швидкого відновлення і удосконалення під впливом деструктивних факторів. Досі у працях дослідників не представлено комплексний та системний погляд на забезпечення резільєнтності СШІ. Більшість праць розглядають стійкість до окремих збурювальних деструктивних факторів та реалізацію окремих механізмів резільєтності. Тому розроблення методів оцінювання та забезпечення резільєнтності СШІ до деструктивних факторів є актуальною задачею для розвитку теорії і практики резільєнтних інтелектуальних систем.

Пропонована розробка має практичну цінність для роботизованих та безпілотних систем військового призначення, оскільки дозволяє здійснювати оцінювання рівня автономності і живучості за умов інформаційних та ресурсних

обмежень. Крім того запропоновані алгоритми дозволяють здійснювати оптимізацію резільєнтності і надавати певні ймовірнісні гарантії щодо робастності і швидкості відновлення ефективності, що важливо і для інфокомунікаційних систем загального призначення, оскільки сприяє зниженню накладних витрат на експлуатацію сервісу аналізу даних.

Проміжний звіт складається із вступу, двох розділів, висновків і переліку посилань.

Перший розділ присвячено аналізу проблеми забезпечення резільєнтності СШІ до деструктивних збурень. Розглянуто сутність поняття резільєнтності та його розуміння у контексті СШІ. Проаналізовано сучасний стан та тенденції розвитку технологій штучного інтелекту. Також у розділі детально розглядаються типи деструктивних збурень та існуючі підходи до забезпечення резільєнтності до них.

Другий розділ присвячено реалізації інформаційної технології оцінювання та забезпечення резільєнтності СШІ. Розглянуто основні показники резільєнтності та запропоновано інтегральний показник резільєнтності, що є функціоналом від функціональної ефективності системи, що змінюється в часі. Запропоновані інформаційні критерії функціональної ефективності і їх диференційовані версії для задач класифікаційного аналізу даних. Розглянуто критерії оптимізації резільєнтності за умов ресурсних обмежень. Визначено набір принципів щодо забезпечення резільєнтності класифікатора зображень, як найбільш вразливої СШІ. Досліджено залежність резільєнтності системи класифікаційного аналізу до впливу інжекції несправностей, протиборчих атак та дрейфу концепцій залежно від архітектурних рішень і параметрів системи.

Результати наукових досліджень, одержаних виконавцями проекту, опубліковано в працях [1]–[8].

1 АНАЛІЗ ПРОБЛЕМИ ЗАБЕЗПЕЧЕННЯ РЕЗІЛЬЄНТНОСТІ СИСТЕМ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

1.1 Сутність концепції резільєнтних систем

Слово «резільєнтність» походить від латинського слова «resiliere», яке означає «відскочити назад». Термін резильєнтність вперше використали фізики для опису здатності твердих та пружних тіл відновлювати свою форму після механічної деформації. Психологи цим терміном часто називають здатність людини успішно функціонувати в умовах стресу, здатність людини розвиватися в умовах негативного життєвого досвіду [9]. Поняття резільєнтності також набуло поширення в екологічних, соціологічних та економічних науках, де використовується для позначення феномену взаємодії і балансу між факторами ризику і захисними факторами, а також як ресурсна адаптація до змінюваних обставин та непередбачуваних умов середовища. Організація ResilienceAlliance визначає резільєнтність як здатність соціо-природньої системи протистояти збуренням і стресорам, вчасно та з мінімальними втратами адаптуючи до них свою структуру і процеси, при збереженні власної ідентичності, функцій і зворотніх зв’язків [9]. Поняття резільєнтності набуло поширення в системній інженерії і відповідна властивість активно досліджується в технічних системах. Резільєнтність в даному контексті розширює поняття гарантоздатності технічних систем, акцентуючи необхідність створення систем, які є гнучкими та адаптивними [10]. Фахівці з кібербезпеки формулюють резільєнтність як здатність передбачати, протистояти, відновлюватись та пристосовуватися до несприятливих умов, зовнішніх впливів, атак чи порушення нормального функціонування системи [11].

Починаючи з 2005 року було запропоновано багато визначень поняття резільєнтності системи. У праці [12] резільєнтність системи була сформульована як її здатність підтримувати свої функції та структуру в умовах внутрішніх і зовнішніх змін і керовано деградувати, коли це необхідно. У праці [13] резільєнтність визначається як здатність системи протистояти значним

збуренням у межах прийнятних параметрів деградації та відновлюватися за прийнятний час зі зваженими витратами та ризиками. У праці [14] автори розглядають властивість резильєнтності системи до збуюючої події (подій) як здатність системи ефективно зменшувати величину і тривалість відхилень від цільових рівнів продуктивності системи під впливом даної події (даних подій). Інші дослідники [15] формулюють резільєнтність як здатність системи підтримувати функціональність та відновлюватися від втрат, спричинених екстремальними подіями. У праці [16] під резільєнтністю системи розуміють внутрішню здатність системи корегувати своє функціонування до, під час і після змін чи збурень, або протягом змін чи збурень задля підтримання необхідних операцій як в очікуваних, так і в неочікуваних умовах. У більш новій праці [17] під резільєнтністю розуміють здатність сконструйованої системи автономно сприймати та реагувати на несприятливі зміни в функціональному стані, протистояти подіям збоїв і відновлюватися від наслідків цих непередбачуваних подій. Деякі дослідники [18] формулюють резільєнтність більш коротко: здатність системи забезпечувати протистояння стресорам. В праці [19] резільєнтність було визначено як здатність системи пристосовуватися до мінливих умов, витримувати збурення і відновлюватися після них. У звіті Національної академії наук США в 2012 році про резільєнтність до стихійних лих було визначено, що резільєнтне функціонування системи полягає в реалізації 4-х основних етапів обробки збурень та загроз (рис. 1.1): 1) планування і підготування системи; 2) поглинання (абсорбування) збурення; 3) відновлення системи; 4) адаптація системи [19].

На етапі планування і підготовки до деструктивних збурень резільєнтна система може виконати такі дії:

– оцінювання ризиків шляхом здійснення аналізу системи та симуляції деструктивних збурень;

– впровадження методів детектування деструктивних збурень;

– усунення відомих вразливостей та впровадження множини заходів захисту системи від деструктивних збурень;

– забезпечення відповідних стратегій резервування та відновлення.

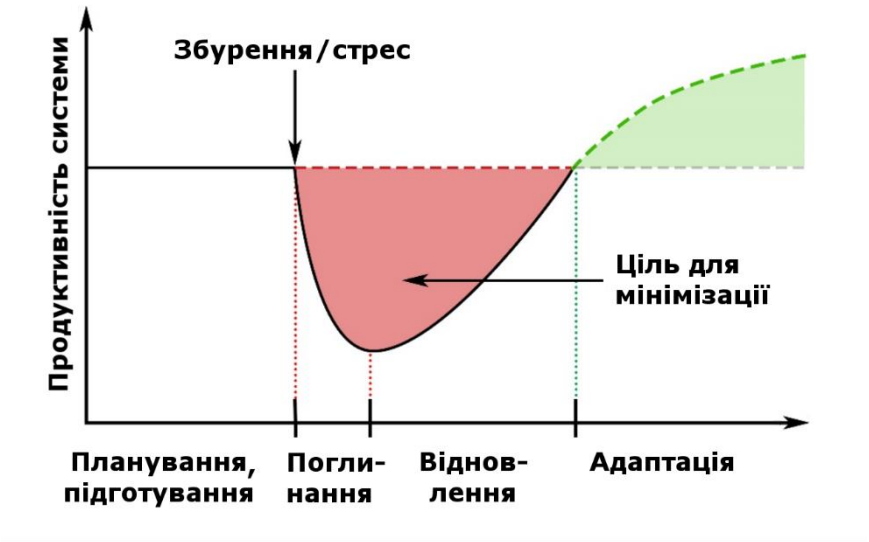


Рисунок 1.1 – Етапи резільєнтного функціонування

Етап поглинання (абсорбції) призначений для реалізації непередбачуваних змін базової архітектури системи, залежно від того, що саме зазнає деструктивного впливу. Механізми поглинання можуть мати багатошарову структуру, реалізуючи захист в глибину, коли система визначає який механізм потрібно використати, якщо не вдалося поглинути загрозу на даному рівні. Якщо уникнути деградації неможливо, то реалізується механізм керованої деградації (graceful degradation), коли основні операції системи мають пріоритет над несуттєвими послугами якомога довше. Система може бути попередньо налаштована з впорядкованим набором менш функціональних станів, які представляють прийнятні компроміси між продовженням функціональності та, продуктивністю та економічною ефективністю.

Етап відновлення включає заходи спрямовані на відновлення втраченої функціональності і продуктивності якомога швидше, дешевше і ефективніше. Етап адаптації зосереджується на здатності системи змінюватися, щоб краще справлятися з майбутніми загрозами.

Дослідниками та інженерами, що займаються розробкою резільєнтних систем, було сформульовано ряд евристик і принципів, на які потрібно спиратися при проектуванні таких систем [17]:

– функційна надлишковість або диверсність, що полягає в існуванні альтернативних способів виконання певної функції [15];

– апаратна надлишковість, що полягає в резервуванні обладнання для захисту від апаратних збоїв;

– можливість самореструктиризації у відповідь на зовнішні зміни;

– прогнозованість поведінки автоматизованої системи задля гарантування довіри і уникнення частого втручання людини;

– уникнення надлишкової складності, викликаної поганими практиками проектування;

– здатність системи функціонувати в найбільш ймовірних і найгірших сценаріях природнього і техногенного характеру;

– керована деградація, що полягає у здатності системи продовжувати роботу під впливом непередбачуваного деструктивного фактору шляхом переходу в стан меншої функціональності чи продуктивності [13];

– реалізація механізму контролю і корегування дрейфу системи до нефункціонального стану шляхом прийняття відповідних компромісів та своєчасних привентивних дій [11];

– забезпечення переходу до “нейтрального” стану для запобігання подальшому пошкодженню під впливом невідомого деструктивного збурення, доки проблема не буде ретельно діагностована;

– ревізійність системи, що полягає у наданні можливості необхідного втручання людини, не потребуючи від неї небгрунтованих припущень;

– людина повинна бути в курсі справи, коли є потреба у «швидкому осмисленні» ситуації та формуванні творчих варіантів;

– реалізація можливості заміни чи підстахування автоматики людьми, коли відбувається зміна контексту, до якої автоматизація не підготована, але достатньо часу для людського втручання;

– реалізація принципу поінформованості про наміри, коли система і люди повинні підтримувати спільну модель намірів для підтримання один одного, коли в цьому є необхідність;

– навчання і адаптація, тобто переналаштування, оптимізація і розвиток системи на основі постійно отримуваних нових знань з середовища [10].

Таким чином, концепція резільєнтних систем полягає в реалізації механізмів підготовки, поглинання, відновлення і адаптації задля забезпечення стабільного надання послуг в надійний спосіб за умов внутрішніх та зовнішніх змін і впливів. Концепція резільєнтності спирається на досить загальні ідеї і принципи і може бути розвинена для складних систем різного типу з урахуванням специфіки притаманних їм загроз і можливостей захисту.

1.2 Сучасний стан і тенденції розвитку технологій інтелектуального аналізу даних

Моделі і методи регресійного, класифікаційного та кластер- аналізу даних різної топології і розмірності, а також пошукові алгоритми і методи навчання з підкріпленням активно досліджуються ще з 60-х років 20 століття [20]. Однак найбільшого прогресу в галузі інтелектуального аналізу даних було досягнуто в останнє десятиліття в результаті розвитку моделей і методів ознакового подання даних і їх оптимізації для вирішення конкретних задач. Дослідниками в галузі аналізу даних сформульовано ряд універсальних принципів щодо формування ознакового опису даних [21]:

– ієрархічна організація пояснювальних факторів;

– гладкість (smoothness) функції, що описує модель подання даних;

– множинність пояснювальних факторів;

– навчання з частковим залученням вчителя;

– гіпотеза багатовидів та принцип інформаційного пляшкового горла;

– спільність факторів під час розв’язання різних задач;

– гіпотеза про природну кластеризацію;

– просторово-часова зв’язаність;

– розрідженість ознакового подання;

– простота залежності факторів високорівневого подання.

Поняття, використовувані для опису спостережень, можуть бути визначені в термінах інших більш абстрактних понять, тобто ієрархічно [20]. Тобто високорівневі ознаки формуються методом композиції низькорівневих ознак (рис. 1.2). Тобто результуюча функція f для екстракції ознак може бути описана у вигляді композиції n шарів трансформації простору ознак $f = f_1 \circ f_2 \circ \dots \circ f_n$.

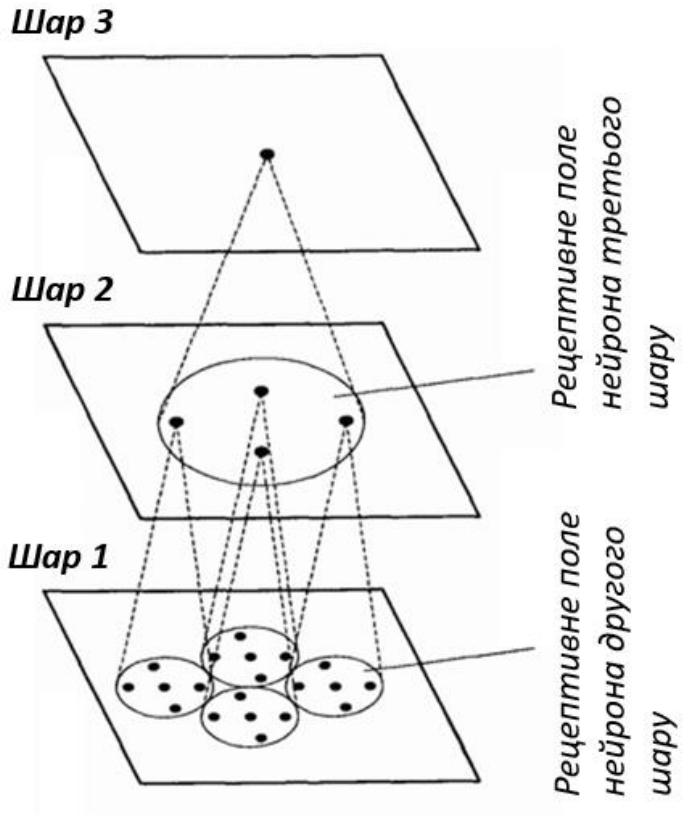


Рисунок 1.2 – Приклад ієрархічної композиції ознак

Варто відмітити, що як неієрархічними так і ієрархічними моделями можна апроксимувати будь-яку функцію з однаковою точністю, проте ієрархічні моделі потребують меншу кількість параметрів для цього [21]. Ієрархічні моделі мають більшу інформаційну ємність і потенційно можуть бути навчені значно більшим абстраціям ознак на верхніх рівнях. Тому збільшення обсягу навчальних даних для ієрархічних моделей забезпечує деталізований аналіз образів з урахуванням

різних контекстів при прийнятній кількості параметрів. Точністні характеристики традиційних моделей машинного навчання перестають зростати після обробки певного обсягу навчальних даних (рис. 1.3). Подальше зростання продуктивності потребує досить суттєвого збільшення кількості параметрів і обчислювальних ресурсів.

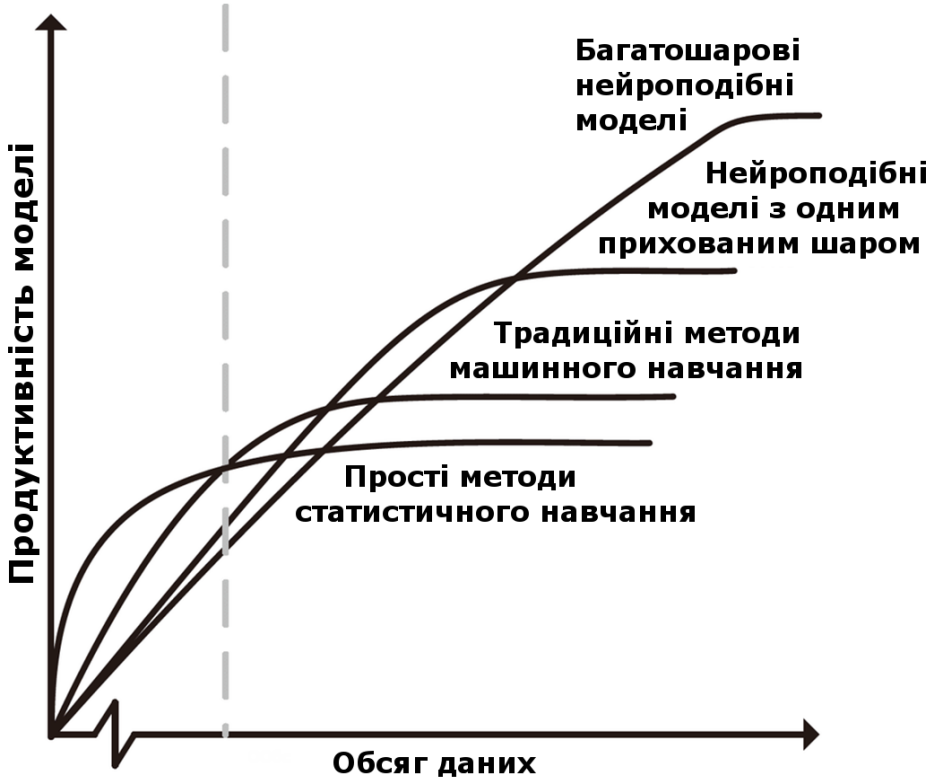


Рисунок 1.3 – Залежність продуктивності аналізу даних від обсягу даних для різних підходів щодо побудови моделей аналізу даних

За допомогою ієрархічного подання можна ефективно здійснювати або екстракцію інваріантних ознак (invariant features), або розділення пояснювальних факторів (disentangle the latent factors) різного рівня абстрактності. Навчання інваріанним ознакає корисне з точки зору навчання з учителем для виконання конкретної вузької задачі. Розділення пояснювальних факторів може бути корисним під час навчання без вчителя, коли сформований ознаковий опис може повторно використовуватися для вирішення різних завдань.

Гладкість функції $f(x)$, що описує модель подання даних можна виразити як $f(x_1) \approx f(x_2)$ за умови $x_1 \approx x_2$. Ця властивість лежала в основі багатьох класичних алгоритмів, що спиралися на принцип локального узагальнення тобто локальної інтерполяції між сусідніми навчальними зразками. Характерними представниками даних алгоритмів є метод найближчих сусідів та ядерні машини з фіксованим локально-чутливим ядром, наприклад Гаусовим ядром [21]. Дослідження показали, що не слід повністю покладатися на дане припущення, особливо у випадку необроблених даних чи обмеженого їх обсягу. Жодна модель не буде повністю гладкою, проте можна приймати міри щодо підвищення гладкості.

Гладкість моделі означає, що модель повинна прогнозувати однакові мітки для схожих зразків, в тому числі для їх масштабованих, обрізаних та викривлених версій. Навчання з аугментацією даних використовують для посилення гладкості $f(x)$, що одночасно забезпечує підвищення узагальнюючої здатності моделі. У випадку навчання з учителем одним із шляхів підвищення гладкості $f(x)$, окрім аугментації, є заміна оригінальних цільових міток зразків (labels) на їх вбудований варіант (Label Embedding), що може бути отриманий в процесі машинного навчання (Label Representation learning). В рамках даного підходу схожі за змістом мітки матимуть схожі вбудовані подання, що полегшує задачу забезпечення властивості гладкості. У працях [22] по аналогії з методом ECOC (Error-correcting output codes) пропонується здійснювати кодування цільових міток завадозахищеними кодами, що виправляють помилки. В цьому випадку помилки в окремих бітах прогнозу не змінюватимуть результат прогнозу, якщо їх кількість менша за порогове значення. Це також забезпечує властивість гладкості в рамках заданого порогу.

Ефект згладжування може бути досягнутий за рахунок використання ансамблювання, в основі якого лежить принцип деверсності (diversity). На цьому принципі базується стекінг множини різнорідних моделей для формування високорівневого ознакового опису. Існує багато визначень диверсності та

способів її оцінювання. Один з підходів до оцінювання диверсності полягає в аналізі ймовірностей одночасної незгоди моделей ансамблю під час прийняття рішень. В інших підходах диверсність моделей ансамблю розглядають як оцінку структурної різноманітності в ансамблі, що може бути виміряна ентропійними та інформаційними мірами [23]. При цьому одним із методів забезпечення диверсності є використання в ансамблі моделей, що основані на різних принципах. Однак також популярним підходом до забезпечення диверсності є введення елементу рандомізації:

– маніпуляції з навчальними вибірками моделей (формування випадкових підвибірок, ініціалізація вагових коефіцієнтів навчальних зразків);

– вибір різних підпросторів (випадкова вибірка ознак);

– маніпуляції з параметрами навчання (ініціалізація нейронної мережі випадковими числами, введення регуляризуючого штрафу за кореляцію результатів мережі з результатами інших мереж ансамблю);

– маніпуляції з поданням вихідних значень моделей (подання класів кодами, що виправляють помилки, або випадкові зміни міток класів в деяких навчальних зразках).

У багатошарових нейронних мережах аналогічну поведінку до ансамблювання можна отримати шляхом введення мульти-шляхів (multi-path connections), в тому числі з'єднань з пропуском шарів (skip connections), або міжшарових з’єднань (cross-layer connectivity). Дані архітектурні рішення використовуються в мережах залишкових зв’язків (residual deep networks), в мережах з щільними зв’язками (densely connected deep networks) та магістральних глибоких нейромережах (highway deep network) і їх модифікаціях. Як показали дослідження в даних мережах навчання альтернативних шляхів відбувається одночасно і незалежно один від одного [24]. Зі збільшенням кількості альтернативних шляхів згладжуючі властивості отриманої моделі зростають повільніше, подібно до сповільнення зростання згладжучих властивостей ансамблю при збільшення його розміру.

Нестабільність виходу навченої моделі під впливом невеликих збурень на вході є ознакою недостатньої гладкості функції моделі. Дослідниками з Google було запропоновано додавати регуляризаційну складову $\|f(x)-f(x')\|_2^2$ до функції втрат, де $f(x)$ – функція, що описує модель ознакового опису, x' – збурений варіант оригінального вхідного зразка x. Для моделювання багатьох видів збурень використовується некорельований Гаусовий шум. Так само регуляризаційна складова може бути введена і для моделі класифікаційного аналізу у вигляді дивергенції Кульбака-Лейблера між розподілами на виході класифікатора до і після збурення. Однак останнім часом з'являється багато нових видів збурень і атак на моделі аналізу даних, що знаходять вразливості попри покращення алгоритмів навчання [25]. Покриття всіх можливих збурень під час стабілізаційного навчання може потребувати дуже великого обсягу ресурсів і часу.

Вхідні дані є результатом взаємодії багатьох пояснюючих факторів. Тому навчання моделі новому фактору приводить до його узагальнення в конфігураціях інших факторів. Саме ця ідея лежить в основі розподіленого подання даних. Кожен параметр може повторно бути задіяним для кодування різних вхідних спостережень чи частин вхідного спостереження. При цьому ці спостереження можуть навіть не бути близькми сусідами. В розподіленому поданні даних експоненційно більша кількість ознак або прихованих змінних можуть бути активовані вхідним сигналом, в той час як в алгоритмах з локальним узагальненням різні частини вхідного простору асоціюються тільки зі своїм персональним набором параметрів [20]. Таким чином, розподілене подання даних може кодувати більшу кількість різноманітних вхідних конфігурацій ніж подання з використанням локального узагальнення.

Для будь-якого даного спостереження x тільки мала частина з усіх можливих факторів є значимою. Більша частина виділених ознак повинна бути нечутлива до малих змін спостереження x. Тобто більша частина детектованих ознак повинна бути нульовою. Цю властивість називають розрідженістю

(sparsity). Дана властивість може бути до-сягнута за рахунок різноманітних технік, основаних на ефекті редукції причини (explaining away). Редукція причини полягає у зв'зуванні двох апріорно не зв'язаних причин події, якщо з'являється спостереження даної події. В цих техніках можуть використовуватися спеціальної форми приховані змінні h, більшість з яких прямують до нуля, або спеціальна нелінійність, значення якої лежать в основному біля нуля, або обмеження матриці Якобіана (або похідних функції) перетворення вхідних даних в обране подання [21]. При цьому знаходження апостеріорної ймовірності розподілу для активації прихованих факторів (причин) h, p(h | x), яке часто використовується як базис для екстракції ознак, виявляється складною задачею. У випадку дискретного h задача взагалі може не мати розв’язку.

Важливе практичне значення має встановлення статистичного взаємозв’язку між навчанням без вчителя і навчанням з учителем. Це дозволяє ефективно використати нерозмічені навчальні дані, які отримати простіше і дешевше, для формування інформативного ознакового опису. Саме в цьому і полягає ідея навчання з частковим залученням учителя (semi-supervised learning). Різні алгоритми навчання з частковим залученням учителя спираються на різні припущення.

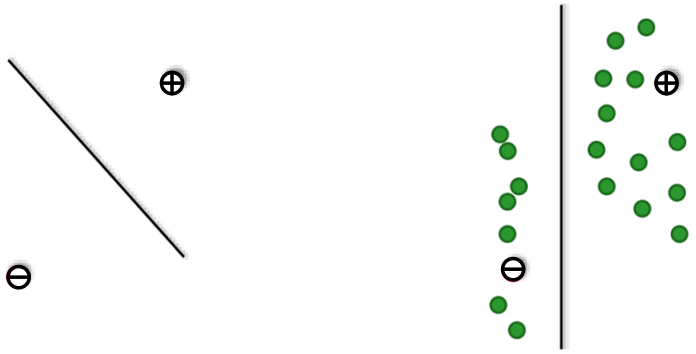
Перше припущення полягає в тому, що найбільш впевнені прогнози моделі, попередньо навченої з учителем, для нерозмічених даних можна вважати правильними і використовувати як псевдо-мітки (psevdo-labels) для подальшого навчання з учителем (self-training). На даному припущенні будувалися класичні алгоритми з частковим залученням учителя, які мали ітераційну структуру і виконувалися доки всі нерозмічені дані не будуть впевнено розпізнаватися [21]. Друге припущення полягає у проходженні роздільної гіперповерхні через області простору з низькою щільністю ймовірності, що дозволяє уточнити межі категорій (рис. 1.4). На цьому припущенні основано так звану трансдуктивну машину опорних векторів (Transductive Support Vector Machines), де після максимізації зазору між роздільною гіперплощиною і опорними векторами в

розширеному просторі ознак здійснюється уточнення зазору з урахуванням відстані нерозмічених даних до роздільної гіперплощини [26].

Друге припущення спирається на користь генеративних моделей (рис. 1.5*а*) і полягає в тому, що ознакове подання z для вхідних даних x, яке зручне для обчислення ймовірнісного розподілу p(x), є зручним і для навчання p(y|x), де y – цільова змінна. Наприклад, варіаційний автокодувальник (Variational Autoencoder) формує варіаційну апроксимацію $q(z \mid x)$ для апостеріорного розподілу p(x|z), що дозволяє визначити сумісний розподіл p(x, z) = p(z)p(x|z), де p(z) – апріорний розподіл прихованих змінних z [21]. Відповідно функція втрат L_{VAE} для варіаційного автокодувальника має вигляд

$$L_{VAE} = -E_{q(z|x)}(-\log(p(x \mid z)) + KL(q(z \mid x) \| p(z)),$$

де $KL(\cdot)$ – функція Кульбака-Лейблера.



а *б*

Рисунок 1.4 – Ілюстрація до навчання з частковим залученням учителя:
а – роздільна гіпеповерхня для розмічених навчальних зразків;
б – роздільна гіперповерхня для розмічених навчальних зразків з урахуванням щільності розподілу нерозмічених даних

Умовний варіаційний кодувальник (Conditional Variational Autoencoder) визначає варіаційну апроксимацію $q(z \mid y, x)$ для апостеріорного розподілу $p(z \mid y, x)$, що дозволяє визначити сумісний розподіл $p(x \mid y, z)$. Для цього функція втрат варіаційного автокодувальника містить окрім помилки реконструкції

регуляризаційну компоненту, що відповідає за приведення розподілу в просторі прихованих змінних до стандартного нормального розподілу і забезпечує гладкість і неперервність функції кодувальника і декодувальника (генератора).

Класичний EM-алгоритм (expectation-maximization) розглядає дані як суміш розподілів. Якщо кількість змішаних компонентів з апріорною ймовірністю p(y) та умовним розподілом p(x|y) є корректними, то можна встановити дійсний зв'язок між розподілом немаркованих даних і мітками категорій за формулою p(x, y) = p(y)p(x|y), де p(x|y) оптимізується за EM-алгоритмом. EM-алгоритм часто поєднують з глибокими архітектурами, якщо розмірність оригінального простору ознак дуже висока [26].

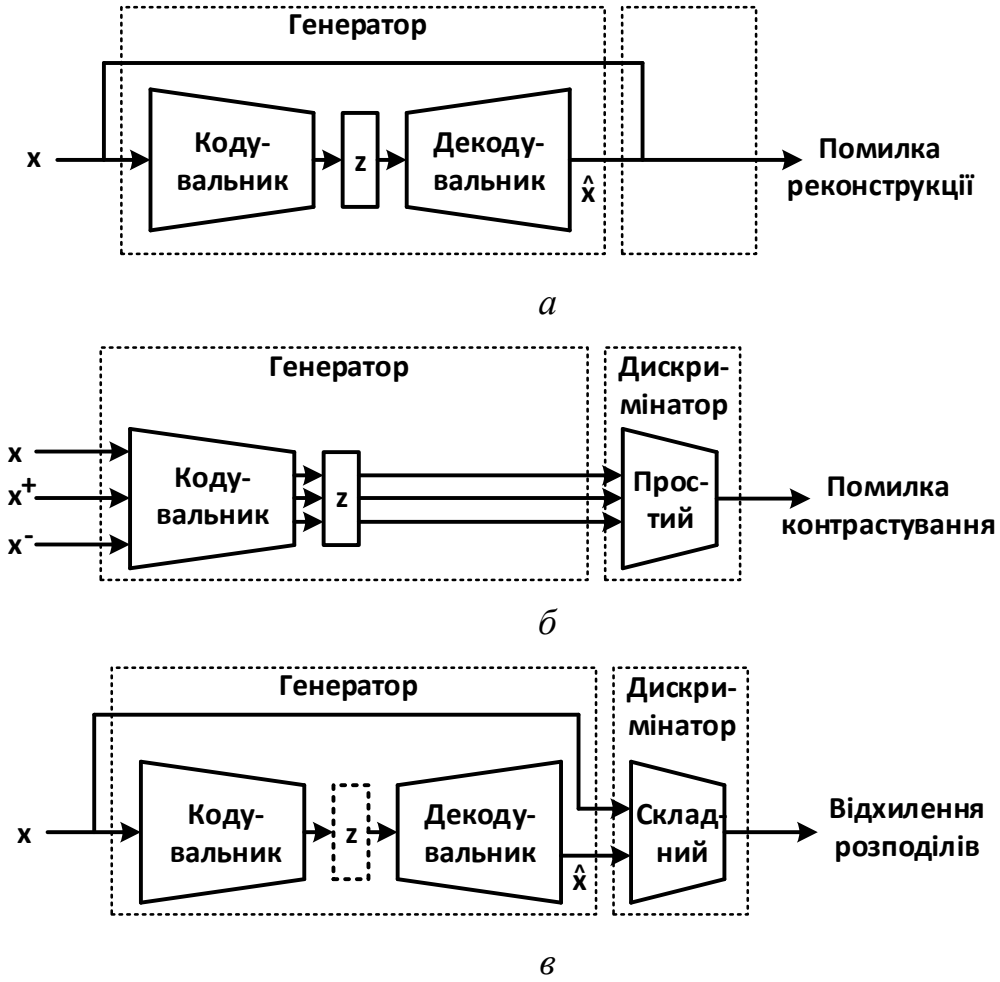


Рисунок 1.5 – Схеми основних методів навчання без вчителя для підготовки ознакового подання z: *а* – генеративні моделі; *б* – контрастні моделі; *в* – генеративно-контрастні моделі

Більшість методів навчання без вчителя полегшують задачу навчання з учителем забезпечуючи ефективну ініціалізацію вагових коефіцієнтів. У випадку вирішення класифікаційних задач найбільш успішним підходом до попереднього навчання без вчителя вважається контрастне навчання [21]. Однак існують модифікації контрастних методів самонавчання і для задача регресії. В рамках даного підходу аугментація даних використовується для генерації різних варіантів кожного вхідного навчального зразка, які розглядаються як позитивні пари, а негативні пари формуються з інших вхідних зразків (рис. 1.5*б*). Контрастне навчання забезпечує максимізацію близькості ознакового подання для позитивних пар і максимізацію віддаленості ознакового подання для негативних пар. Функція втрат для контрастного навчання реалізує шумове контрастне оцінювання за формулою [27]

$$J = E_{x,x^+,x^-}[-\log\left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + e^{f(x)^T f(x^-)}}\right)], \qquad (1.2.1)$$

де x^+ – позитивна пара для x;

x^- – негативна пара для x.

Якщо є банк пам’яті з ознаковим поданням K негативних пар для x, то можна оцінювати взаємну інформацію між позитивною парою і відповідним набором негативних пар. Функція втрат в цьому випадку називається інформаційним шумовим контрасним оцінюванням (InfoNCE), що обчислюється за формулою [27]

$$J = E_{x,x^+,x^k}[-\log\left(\frac{e^{f(x)^T f(x^+)}}{e^{f(x)^T f(x^+)} + \sum_{k=1}^{K} e^{f(x)^T f(x^k)}}\right)]. \qquad (1.2.2)$$

Мета контрастного навчання природньо відповідає задачам класифікаційного аналізу даних на відміну від генеративних моделей. Крім того, контрастне навчання обчислювально ефективніше, оскільки відпадає необхідність у використанні декодера. Дослідниками було розроблено такі відомі методи констрастного самонавчання як MoCo, SimCLR, BYOL та SwAV, які продемонстрували покращення результатів наступного навчання з учителем на різних бенчмарках машинного зору [27]. Одним із недоліків контрастних методів навчання є необхідність у підборі архітектури та ємності екстрактора ознак, а також гіперпараметрів навчання задля уникнення збіжності до константного ознакового подання для всіх зразків.

Генеративно-контрастні методи навчання без вчителя основані на генеративно-змагальних мережах (Generative Adversarial Networks). Навчання генеративно-змагальних мереж полягає у змаганні двох підмереж, одну з яких називають генератором і позначають як G, а іншу – дискримінатором і позначають як D. Генератор навчається формувати фіктивні зразки даних, які вводять в оману дискримінатор, а дискримінатор навчається відрізняти фіктивні зразки від справжніх. Навчання закінчується коли генератор почне формувати дані з того ж розподілу, що й оригінальні дані. Формалізовано мета навчання полягає у вирішенні мінімаксної задачі

$$\min_G \max_D E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(x)} \log[1 - D(G(z))]. \qquad (1.2.3)$$

На відміну від варіаційного аавтокодувальника, де задано розподіл для прихованих змінних $p_z(z)$ у явному вигляді, у змагально-генеративних моделях розподіл $p_z(z)$ моделюється приховано всередині моделі генератора (рис. 1.5*в*).

У мережах Bi-GAN та ALI пропонується архітектура так званого змагального автокодувальника (рис. 1.6), що має такі складові [28]: кодувальник, E, що здійснює відображення реального зразка x в подання $z' = E(x)$; генератор (декодувальник), G, що генерує вихідні зразки $x' = G(z)$ на основі вхідного

подання z; дискримінатор, D, що приймає випадковий вектор z' з обраного розподілу (реальний розподіл) і приховане ознакове подання на виході кодувальника z. Регуляризуюча складова, що додається для навчання кодувальника і дискримінатора в рамках змагальної конфігурації має такий вигляд

$$L_{Disc} = CrossEntropy(q(z), p(z)) \tag{1.2.4}$$

Перевага змагального автокудвальника і його модифікацій перед варіаційним автоенкодером полягає в тому, що існує можливість підгонки q(z) до p(z) без необхідності доступу до функціональної форми апріорного розподілу. Для змагального навчання в даній конфігурації достатньо формувати вибіркові дані з апріорного розподілу p(z). Така конфігурація спрощує вбудовування інформації про мітки даних для уточнення і стилізації форми розподілу даних q(z) шляхом асоціювання кожного класу з певною модою суміші Гаусіан. Розвиток даного підходу забезпечує реалізацію навчання з неповним залученням учителя для ефективного використання як розмічених, так і нерозмічених навчальних даних.

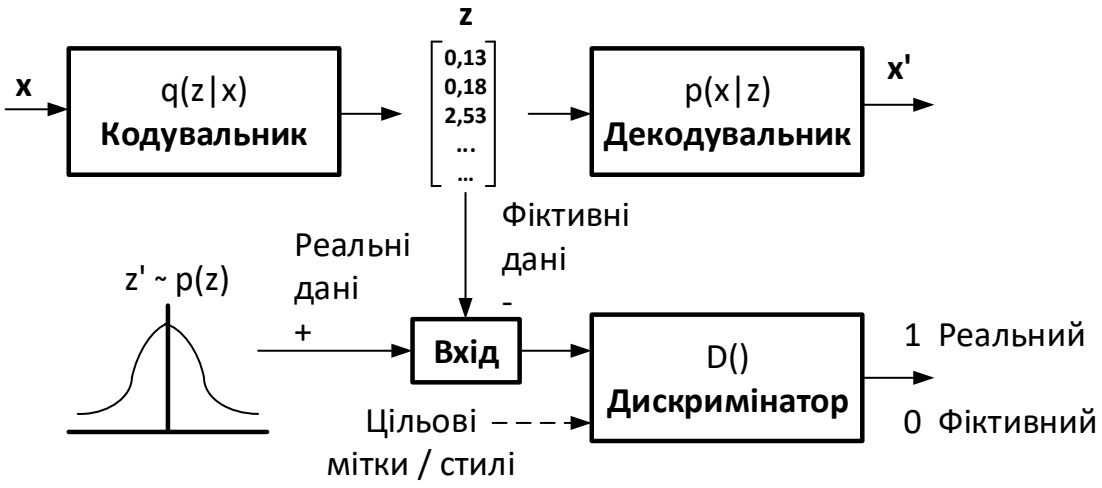


Рисунок 1.6 – Структура змагального автоенкодера

Популярним підходом до навчання без вчителя для полегшення навчання з учителем також є використання методів, у яких псевдомітки даних розробляються на основі внутрішньої інформації нерозмічених даних. Задачу використання внутрішньої інформації даних для визначення їх псевдорозмітки називають передтектом (pretext task). Найбільший розвиток даний підхід набув в галузі аналізу зображень, де до передтекстових задач відносяться: контекстне кодування (Context Encoder) для заповнення видалених патчів (Inpainting), збирання пазлу (Jigsaw puzzle), визначення повороту (Rotation prediction) та кольору зображень (re-colorization), відтворення роздільної здатності зображень (super resolution recover) [27]. Як правило, подібні задачі вирішуються з використанням принципів змагально-генеративних мереж та їх умовних (Conditional GAN) варіантів. При цьому псевдорозмітка використовується як дискримінатором, так і генератором. Як модель генератора часто використовується архітектора автокодувальника з латеральними різномасштабними зв’язками (multi-scale skip connections) типу U-Net або архітектура знешумлюючого автокодувальника (Denoising Autoencoder) [28].

Під час проектування алгоритмів навчання з частковим залученням учителя також інколи використовують графові структури, що покладаються на геометрію даних, утворену як розміченими, так і нерозміченими зразками. В даному випадку геометрію подають у вигляді графу G = (V, E), де вузли V представляють навчальні точки даних з |V| = n, а ребра E представляють схожість (відстань) між точками. Використовуючи графову структуру даних, можна навчитися розповсюджувати інформацію за допомогою дуже малої кількості міток [21]. Наприклад, в методі поширення міток (Label propagation) [29] здійснюється прогнозування міток нерозмічених даних на основі вузлів розмічених даних. Кожна мітка вузла поширюється до своїх сусідів відповідно до схожості. На кожному кроці поширення вузла кожен вузол оновлює свою мітку відповідно до інформації про мітки своїх сусідів. В даному методі мітка розмічених даних фіксована, тому поширення міток відбувається тільки на нерозмічені дані. Метод поширення мітки можна застосувати і до глибокого

навчання, де всі вузли кодуються певним високорівневим поданням, що відображає як роль самого вузла так і структурну інформацію про сусідство.

Останні успіхи в галузі глибокого машинного навчання пов’язані з використання все більшого обсягу розмічених та нерозмічених реальних навчальних даних. При цьому останні дослідження в галузі машинного зору поставили під сумнів необхідність саме реальних даних для навчання глибоких нейронних мереж [30]. Було визначено, що важливим є не реальність даних, а їх натуралістичність, тобто з фіксацією певних структурних властивостей реальних даних, а також їх різноманітність. Багато з цих властивостей можна відобразити в простих моделях структурованого шуму. Тому у працях [30] було досліджено ефективність генерації зразків на основі генеративної змагальної моделлі StyleGAN, що замість навчання ініціалізується різними джерелами шуму. Розглянуто ініціалізацію згорткових фільтрів StyleGAN шляхом семплінгу вейвлетів, Лапласового шуму або фрактальних структур. В результаті StyleGAN перетворювався в генератор даних для контрастного самонавчання екстрактора ознак. Було показано, що згенеровані таким чином дані забезпечують ефективне навчання екстрактора ознак для багатьох задач машинного зору.

Можливість зниження розмірності ознакового подання ґрунтується на гіпотезі багатовидів (manifolds). Згідно з даною гіпотезою основна щільність ймовірності даних зосереджена в регіонах, що мають набагато меншу розмірність, ніж оригінальний простір вхідних даних. Дана гіпотеза покладена в основу класичного алгоритму головних компонент та в основу розробки різних варіантів автоенкодерів. Питання стиснення ознакового подання для конкретних задач на кінці мережі найбільш теоретично обгрунтовано у працях професора Н. Тішбі і його послідовників в рамках принципу інформаційного пляшкового горла [31]. Згідно даного принципу необхідно знайти баланс між компресією даних та збереженням максимальної предиктивності щодо цільової змінної. Зниження розмірності відбувається архітектурно за рахунок стекування шарів і операторів агрегації. При цьому алгоритм навчання забезпечує збереження інформації про цільову змінну на кожному з шарів, що приводить до

забезпечення інваріантності ознак і розділення пояснювальних факторів (рис. 1.7). Тобто реалізація принципу інформаційного пляшкового горла полягає у пошуку компактного ознакового подання $\tilde{X}$ при збереженні максимальної кількості інформації про цільову змінну Y шляхом мінімізації наступної цільової функції

$$J = I(\tilde{X}; X) - \beta I(\tilde{X}; Y), \qquad (1.2.5)$$

де $I(\square)$ – взаємна інформація між двома змінними;

X, $\tilde{X}$ – дані та їх стиснене ознакове подання;

$\beta > 0$ – параметр компромісу між складністю ознакового подання та кількістю збереженої актуальної для задачі інформації.

Для стиснення ознакового подання можна зменшувати як його розмірність, так і множину можливих значень. У працях [32] було виявлено регуляризуючі та мета-регуляризуючі властивості дискретного ознакового подання даних. При цьому в одних підходах дискретизація здіснюється на одному з етапів навчання, а в інших – реалізується на кожному кроці навчання [31, 32]. Якщо у виразі (1.2.5) прийняти обмеження $I(\tilde{X}; X) \le \gamma$, то принцип інформаційного пляшкового горла зводитиметься до максимізації взаємної інформації між ознаковим поданням та вихідними цільовими мітками даних $\max\limits_{\tilde{X}} I(\tilde{X}; Y)$.

У багатозадачних моделях аналізу даних може досягатися вищий рівень узагальнення в навчанні внаслідок підсилення статистичного взаємозв’язку між задачами, що використовують спільні пояснювальні фактори (рис. 1.8). Навчання моделі на декількох задачах, що мають спільні фактори є цінним для реалізації передачі знань (Transfer Learning) та їх адаптації під нову задачу чи умови (Domain Adaptation). Останнім часом багато досліджень і розробок пов’язано з реалізацією концепції неперервного навчання (Continual Learning, Continual Lifelong Learning), в рамках якої передбачається постійна адаптація

моделі до змін середовища і задач, повторне використання вже накопичених знань, набуття нових знань та розширення функціональних можливостей. При цьому передбачається уникнення катастрофічного забування (Catastrophic forgetting) під впливом нових знань, для чого часто реалізують різні механізми регуляризації та нагадування, в тому числі з використанням багатозадачного виходу моделі [33]. Винятком є задача адаптації до дрейфу концепцій (Concept Drift), коли частину застарілого досвіду модель всеж таки повинна забути чи заблокувати задля коректного функціонування [33].

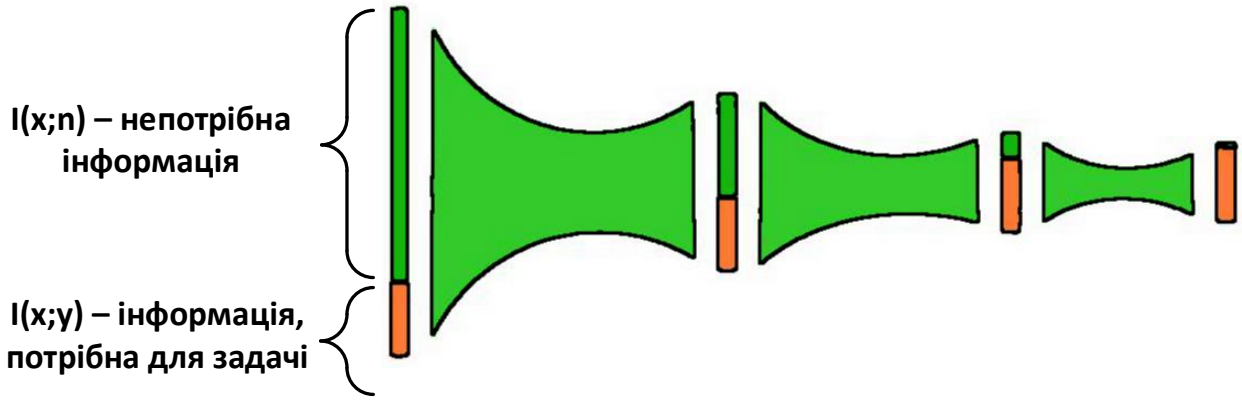


Рисунок 1.7 – Стекування шарів для узагальнення важливої і фільтрації не важливої для задачі інформації

Ефективне використання накопиченого у моделі досвіду дозволяє реалізувати навчання розпізнавати нові класи маючи один чи декілька розмічених зразків на клас. Такі задачі навчання називають навчанням з одного пострілу (One Shot Learning) та навчанням з декількох пострілів (Few Shot Learning) відповідно [34]. Також дані задачі у літературі часто називають класифікацією на N шляхів з K пострілів (N-way-K-Shot-classification), де N позначає кількість класів, а K – кількість зразків на клас. Як правило, ефективне швидке донавчання на обмеженій кількості розмічених даних реалізують з використанням різних варіантів мета-навчання (Meta-Learning), яке полягає в

навчанні навчанню (Learning-to-Learn) [35]. Узагальнена схема процесу мета-навчання показана на рис. 1.9.

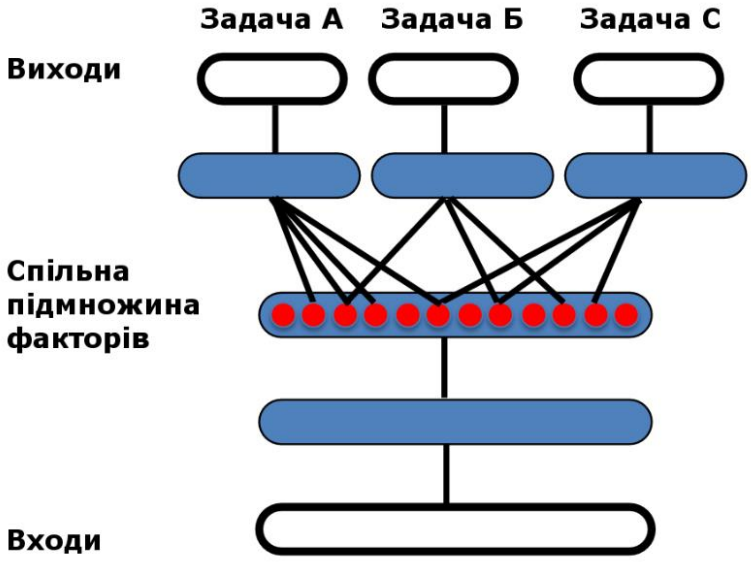


Рисунок 1.8 – Ілюстрація спільних факторів багатозаданої моделі, що сприяють підвищенню рівня узагальнення даних

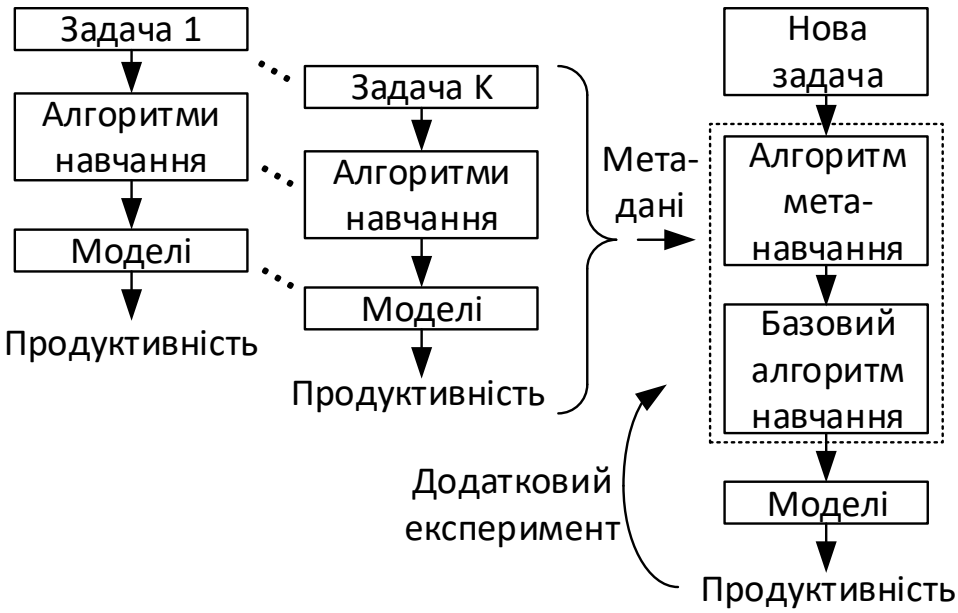


Рисунок 1.9 – Ілюстрація концепції мета-навчання

Техніка мета-навчання полягає в аналізі процесу навчання моделі для вирішення різних задач з метою навчання навчатися на нових задачах швидше і ефективніше. Найбільшого поширення набув незалежний від моделі алгоритм мета-навчання (Model-Agnostic Meta-Learning, MAML) та інспіровані ним методи Reptile, LEO та інші [34]. MAML-алгоритм здійснює навчання будь-якої стандартної моделі таким чином, щоб підготувати модель до швидкої адаптації, тобто сформоване ознакове подання даних краще переноситиметься на нові задачі.

Дистиляція знань є популярним методом стиснення великої моделі, яку ще називають вчителем, в невелику модель, яку називають студентом. Ідея методу полягає у заміні чітких міток (hard targets) даних на м’які мітки (soft targets) даних [36]. Вектор прогнозу на виході великої моделі даних використовується як цільова мітка даних для моделі студента. Така м’яка цільова мітка містить набагато більше інформації ніж чітка мітка, оскільки м’яка мітка вказує на подібність прогнозованого класу на інші класи, взаємозв’язки між класами. Велика модель може забезпечити ефективну екстракцію ознак і виявлення відношень між об’єктами, що дозволить підготувати більш концентровану інформацію для навчання моделі меншого розміру. Дистиляційна функція втрат (Distillation Loss) основана на дивергенції Кульбака-Лейблера, оскільки призначена для оцінювання подібності двох розподілів ймовірності [36]. При цьому результуюча функція втрат моделі студента містить і складову звичайної крос-ентропійної функції втрат з чіткими цільовими мітками.

На ефективність ознакового подання даних може впливати також архітектура і мікроархітектура нейронної мережі. Ефективність тієї чи іншої моделі залежить від типу і топології вхідних даних. Останнім часом розвивається моделі і методи автоматичного машинного навчання (Auto-ML), що здійснють пошук найбільш ефективних архітектур і мікроархітектур з точки розу співвідносшення точності і обчислювальної складності. Наприклад, компанія Google використала метод нейронного пошуку архітектури (Neural Architecture Search) для винайденя ефективної згорткової мережі EfficientNet [24]. Однак

подібні підходи є дуже ресурсозатратними і не завжди виправданими. У працях [37] було запропоновано різноманітні алгоритми автоматичного навчання, основані на еволюційних пошукових алгоритмах, однак вони мають велику кількість гіперпараметрів.

Під час синтезу моделей аналізу даних варто враховувати обмеження на зміну «крізь час і простір». Тобто спостереження, сформовані у сусідніх областях простору чи отримані послідовно в часі, повинні прагнути асоціюватися з однаковими значеннями відповідної категорії понять, чи приводити до невеликого руху по поверхні багатовиду високої щільності. Деякі дослідники вводили до-даткову регуляризуючу складову до функціоналу якості, яка враховує різницю значень ознак в різні моменти часу. Інші використовують апріорні знання про топологічну структуру даних для використання локальних рецептивних полів нейронів і операторів агрегації відгуку нейронів на сусідні ділянки вхідного простору в одне компактне подання. В обох випадках було досягнуто підвищення ефективності моделі. Крім цього, просторово-часова зв’язаність спостережень обгрунтовує спосіб розширення навчальних даних, що оснований на застосуванні невеликих випадкових деформацій образів в існуючих навчальних даних. При цьому деформуючі зміни повинні бути обмежені, щоб зберегти відношення згенерованих зразків до тієї ж категорії, що й оригінальний зразок.

В ефективних високорівневих поданнях даних фактори зв’язані один з одним через прості залежності. Тому синтез вихідних шарів варто здійснювати в рамках підходів, що характеризуються найбільшою ефективністю з точки зору обчислень та інтерпретації. Найбільшого поширення набули лінійні та радіально-базисні вихідні шари. Лінійні залежності найбільш поширені в природі, що можна побачити з більшості фізичних законів. Радіально-базисні функції можуть інтерпретуватися з точки зору нечіткої логіки задаючи функцію належності до категорії чи підкатегорії [22]. У випадку радіально-базисних функцій з дискретним ознаковим поданням вихідний шар може реалізувати принципи кодів, що виправляють помилки [22].

Таким чином, основні досягнення в галузі штучного інтелекту пов’язані з удосконаленням методів формування високорівневого ознакового подання та обробки великих обсягів розмічених і нерозмічених даних. Дослідниками в галузі аналізу даних сформульовано основні принципи, яких необхідно дотримуватимя при синтезі архітектур моделей і алгоритмів навчання для забезпечення кращого співвідношення точності і обчислювальної ефективності. Однак питання синтезу і операційної підтримки життєвого циклу інтелектуальних алгоритмів, що неперервно навчаються і удосконалюються утворюють активну область досліджень.

1.3 Аналіз збурюючих факторів, що впливають на системи штучного інтелекту

У загальному випадку системи штучного інтелекту функціонують в неідеальних умовах, і можуть піддаватися впливу різних деструктивних чинників. Деструктивні чинники можуть впливати як на обчислювальне середовище розгортання, так і на дані в режимі навчання чи екзамену. Деструктивні впливи можуть бути зловмисними, а можуть мати природній характер. До деструктивних впливів можна віднести такі чинники:

– апаратні несправності;

– шум та змагальні атаки.

– дрейф концепцій;

– новизна в тестових даних;

– пропуски і помилки в даних.

Апаратні несправності (Faults) в обчислювальній системі породжують помилки (Errors). Помилкою вважається такий прояв несправності в системі, що призводить до відхилення актуального стану елемента системи від очікуваного [38]. Якщо несправність не викликає помилку, то таку несправність називають сплячою. Внаслідок помилок виникають збої (Failures), тобто стан нездатності системи виконувати свою передбачувану функціональність чи поведінку. На рис. 1.10 проілюстровано причинно-наслідковий зв'язок між

апаратною несправністю, помилкою і збоєм, при яких відбувається порушення передбачуваної поведінки нейронної мережі, яка розгорнута для виконання своєї задачі в обчислювальному середвищі.

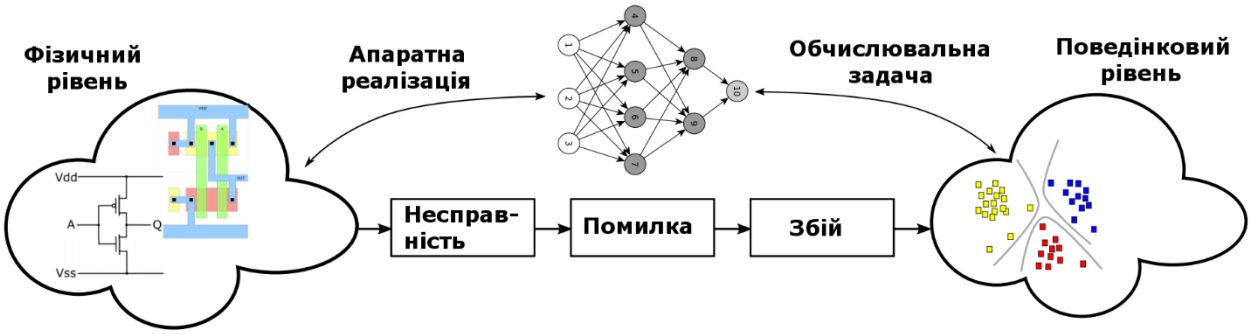


Рисунок 1.10 – Поширення несправності з фізичного рівня обчислювального середовища до поведінкового рівня нейромережевого додатку

Несправності можна класифікувати за їх часовими характеристиками (рис. 1.11) [38]:

– постійна несправність – безперервна і стабільна в часі, що в основному є результатом незворотного фізичного пошкодження;

– непостійна несправність – може зберігатися лише протягом короткого періоду часу і часто є результатом зовнішніх збоїв.

Постійні типи несправностей можуть моделювати багато дефектів в транзисторах та з'єднувальних структурах на логічному рівні з досить високою точністю. Найбільш поширеною моделлю постійних дефектів є так зване «застрягання», що полягає у збереженні виключно високого (stuck-at-1) чи низького (stuck-at-0) стану на лініях даних чи управління. Також для оцінювання відмовостійкості в обчислювальних системах, в літературі розглядаються несправності типу «застрягання у відкритому» або «застрягання у замкнутому» стані, щоб описати випадки, коли «плаваюча» лінія має високу ємність і зберігає свій заряд протягом значного часу в сучасних напівпровідникових технологіях.

Рисунок 1.11 – Типи апаратних несправностей в обчислювальному Середовищі

Непостійні несправності охоплюють переважну більшість несправностей, які виникають у цифрових обчислювальних системах, побудованих за сучасною напівпровідниковою технологією. Очікується, що технології майбутнього будуть ще більше схильні до непостійних несправностей через більшу чутливість до впливу зовнішнього середовища та високу напругу матеріалів в носіях з високим рівнем мініатюризація. Непостійні несправності, що повторюються з деякою частотою, як правило виникають через граничну або нестабільну роботу пристрою, і їх складніше виявити, ніж постійні. Тимчасові несправності пов’язані з впливом на параметри схем, що визначають часові характеристики, а не на структуру ланцюгів. До тимчасових несправностей відносять непердбачувану затримку поширення сигналу, випадкові перемикання бітів в регістрах пам’яті, імпульсні зміни в логічних схемах.

Існує кілька фізичних методів ін’єкції несправностей зі зловмисними намірами. На практиці інцієкцію несправностей реалізують за рахунок збою

системного лічильника тобто синхронізації схем, просадки живлення до певного рівня, електромагнітний вплив на напівпровідники, опромінення важкими іонами, впливу на пам'ять лазерного променю, а також програмних rowhammer-атак на біти пам’яті.

Лазерний промінь може інжектувати помилку в статичну пам'ять із довільним доступом (SRAM-пам’ять). Під час впливу лазерного променю на кремній в діелектрику утворюється тимчасовий провідний канал, який, у свою чергу, змушує транзистор перемикати стан точним і контрольованим способом [38]. Ретельно регулюючи параметри лазерного променя, наприклад його діаметр, випромінювану енергію та координату впливу, зловмисник може точно змінити будь-який біт в SRAM-пам’яті. Раніше лазерний промінь широко використовувався разом з диференціальним аналізом несправностей для вилучення приватного ключа мікросхем шифрування [38].

Rowhammer-атаки можуть спричиняти помилки в DRAM-пам’яті. Цей вид атак використовує особливості електричної взаємодії між сусідніми комірками пам’яті [39]. Швидко і багаторазово звертаючись до певної ділянки фізичної пам’яті, біт у сусідній ділянці може інвертуватися. Профілюючи шаблони інвертування бітів у модулі DRAM-пам`яті та зловживаючи функціями керування пам’яттю, rowhammer може надійно інвертувати один біт за будь-якою адресою в стеку програмного забезпечення. Rowhammer-атаки використовувався часто для зламу ізоляції пам’яті у системах віртуалізації та отримання root-прав в системі Android.

Лазерний промінь та Rowhammer-атаки можуть інжектувати помилку в пам’ять з надзвичайно високою точністю. Однак, для інжекції багатьох помилок, лазерний промінь повинен переналаштовуватися, а Rowhammer-атака потребує переміщення цільових даних в пам’ять. Переналаштування лазерного променю, як і переміщення даних вимагають певних накладних витрат, тому проектування нейромережевих алгоритмів повинно забезпечувати стійкість до певного рівня інвертованих бітів щоб зробити дані атаки непридатними з практичної точки зору.

Інжекція несправностей і помилок в цифрову систему розгортання штучного інтелекту може здійснюватися адаптивним чином з урахуванням зворотного зв’язку (рис. 1.12). У даному випадку успішність атаки контролюється на виході нейромережі, а сама атака може виконуватися наприклад за алгоритмом одиночних зміщень (Single Bias Attack, SBA) або алгоритмом градієнтного спуску (Gradient Descent Attack, GDA) [38, 40].

Вихідні дані нейронних мереж сильно залежать від зміщень у вихідному шарі, тому SBA реалізується шляхом збільшення лише одного значення зміщення, що відповідає нейрону, що відповідає цільовому класу розпізнавання. SBA розроблений для тих випадків, коли прихованість атаки не потрібна. Для випадків, коли прихованість важлива, використовується GDA, де градієнтний спуск здійснює пошук набору параметрів, які потрібно змінити, для успішності атаки. Деякі дослідники реалізували приховану атаку на основі методу множників змінного напрямку (Alternating Direction Method of Multipliers, ADMM), що гарантує атаку для вхідного зображення і мінімізацію вплив під час обробки інших зображень [39].

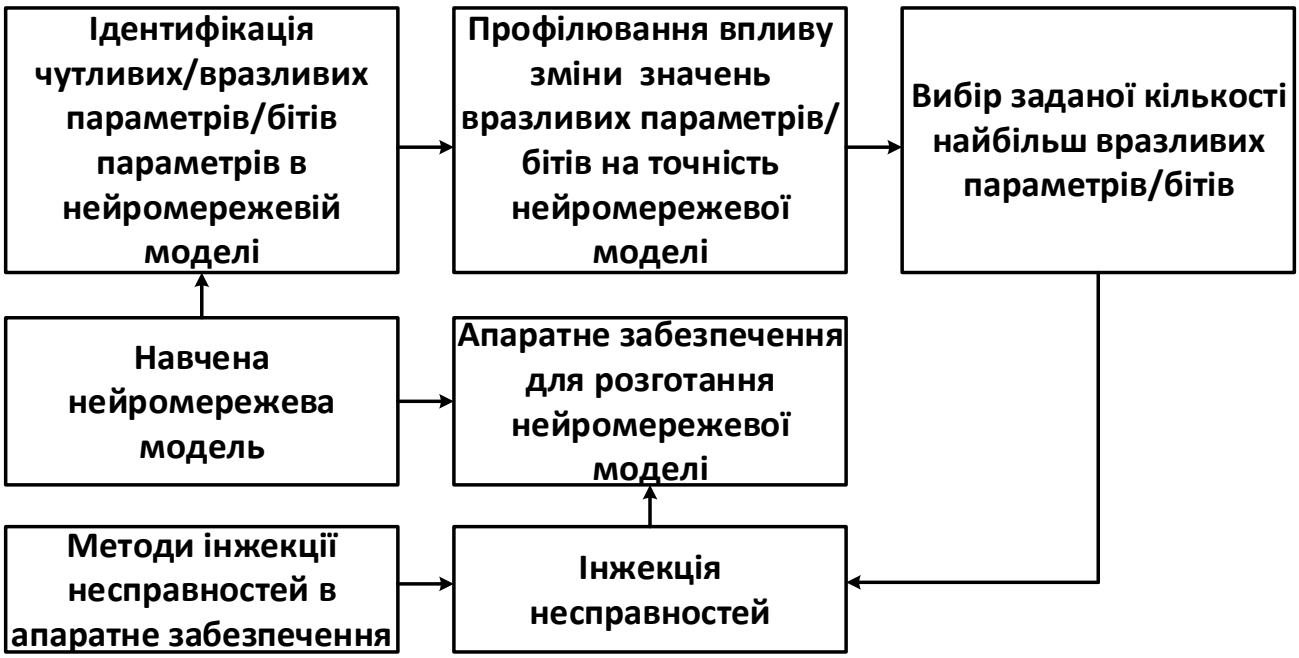


Рисунок 1.12 – Блок-схема методології інжекції несправностей і помилок до цифрового пристрою

Актуальність захисту від адаптивних алгоритмів інжекції несправностей і помилок пов’язана з тенденцією до переміщення інтелектуальних обчислень в реальному темпі часу на крайові пристрої (edge devices). Ці пристрої з більшою імовірністю доступні зловмиснику фізично, що збільшує можливості щодо введення пристроїв в оману. Сучасні кіберфізині системи та інтернет речей є типовими платформами, на яких можуть розгортатися інтелектуальні алгоритми, що потребують захисту від інжекції помилок.

Дослідниками в галузі штучного інтелекту було виявлено, що нейромережеві алгоритми чутливі до так званих змагальних атак (Adversarial attacks), які полягають у додаванні до навчальних чи тестових даних викривлень малої амплітуди, що призводить до помилкових рішень [41]. При цьому було відмічено, що змагальним атакам притаманні наступні властивості:

– непомітність (imperceptibility), тобто існують способи такої мінімальної модифікації даних, що ця модифікація не помітна для людині, але призводить до неадекватної роботи штучного інтелекту;

– існує можливість цільової атаки (Targeted Manipulation) на вихід нейромережі, наприклад, на певний клас розпізнавання, тобто є можливість маніпуювання системоою для власної користі і отримання виграшу, а не просто порушувати її нормальне функціонування;

– властивість переносимості (transferability), коли змагальні зразки, створені для введення в оману однієї моделі, здатні до такого ж введення в оману інших моделей з іншою архітектурою, якщо вони були навчені для виконанння однієї і тієї ж задачі. Ця властивість дозволяє зловмисникам використовувати сурогатну модель як наближення для генерування атак для цільової моделі, яку називають оракулом).

– відсутність загальноприйнятих теоретичних моделей щодо того, чому змагальні атаки такі ефективні. Було висунуто кілька гіпотез, таких як лінійність, неінваріантність та ненадійність ознак, що привело до розроблення кількох захисних механізмів, але жоден з них не діє як панацея для створення надійних моделей та стійких засобів захисту.

Ціллю атак можуть бути модель машинного навчання, середовище розгортання та джерело генерації даних. За метою атаки можна розпізнити на три основні типи:

– атака на доступність, яка призводить до непридатності використання моделі кінцевим користувачем;

– атака цілісності, яка призводить до неправильної класифікації вхідного спостереження (цілеспрямована атака цілісності змушує модель виробляти конкретне неправильне рішення);

– атака конфіденційності, де метою зловмисника є перехоплення комунікації між двома сторонами і отримання приватної інформації. Як правило це атаки на алгоритми штучного інтелекту, що задіяні у забепеченні кіберзахисту певної системи.

За стратегією змагальні атаки можна класифікувати на: атаки ухилення (evasion), атаки отруєння (poisoning) та атаки оракула [42]. Атаки ухилення – це атаки на систему в режимі прийняття нею рішень, тобто це пошукова атак, що має на меті заплутати модель машинного навчання. Дана атака передбачає оптимізаційний процес пошуку маленького збурення, шо викличе неправильне рішення. За частотою оновлення та оптимізації змагальних зразків їх поділяють одноразові (One-shot attacks) та ітеративні (Iterative attacks). Ітеративні атаки формують більш жорсткі змагальні зразки, однак вони обчислювально дуже складні. Атаки отруєння – це пошкодження даних або логіки моделі, щоб погіршити результат навчання. Атака оракула полягає у використанні зловмисником доступу до програмного інтерфейсу, щоб створити заміну, тобто сурогатну модель, яка зберігає значну частину функціональних можливостей оригінальної моделі. Це робиться для більш кропіткого пошуку атаки ухилення для моделі сурогату, яка переноситься на оригінальну модель. Атаки оракула поділяють на: екстракцію, інверсію і виведення. Метою атаки екстракції є вилучення архітектурних деталей моделі зі спостережень вихідних прогнозів та ймовірностей класів. Інверсійні атаки виникають, коли противник намагається

відновити дані навчання. Атака виводу дозволяє противнику ідентифікувати конкретні точки даних з розподілу навчального набору даних.

За нашими знаннями стосовно моделі аналізу даних атаки можна класифікувати на:

– атаки білої шухляди (white-box attacks), для формування яких звломисник має повне знання про дані, модель і алгоритм навчання. Цим методом скоріш за все може користуватися сам розробник системи для аугментації даних і оцінювання робастності моделі;

– атаки сірої шухляди (gray-box attacks), для формування яких зловмисник має часткову інформацію, але достатню для атаки на систему;

– атаки чорної шухляди (black-box attacks), для формування яких у розпоряджені зловмисника є інтерфейс надсилання даних і відповідь у вигляді розпізнаного класу чи ймовірностей розпізнавання класів. Даний вид атак має найбільшу загрозу на практиці.

Основною складовою змагальної атаки є змагальний зразок даних x', що утворюється шляхом незначного збурення (додавання шумової складової) $x' = x + \varepsilon$, що призводить до значної зміни на виході мережі, що описується функцією $f(x)$, тобто $f(x') \neq f(x)$.

У випадку атак білого ящика вагові коєфіцієнти θ нейромережі розглядаються як фіксовані і оптимізації підлягає лише вхідний тестовий зразок x з метою генерації змагального зразка $x' = x + \varepsilon$. Популярними методами вирішення даної оптимізаційної задачі є ітераційний метод числової оптимізації Бройдена-Флетчера-Гольдфарба-Шанно (L-BFGS-алгоритм), метод швидкого знаку градієнта (Fast gradient sign, FGSM), базового ітераційного методу (Basic Iteractive Method, BIM), алгоритм проекційного градієнтного спуску (projected gradient descent, PGD), атака Карліна і Вагнера (Carlini and Wagner, C&W) та їх удосконалені версії.

Цільова функція мінімізації за алгоритмом L-BFGS для модифікації вхідного зразка x, якому відповідає мітка y, з метою генерації змагального зразка x' може мати вигляд

$$c\|x - x'\|_p + J(\theta,\ x',\ y'),$$

де y' – змагальна цільова мітка, $y' \neq y$;

$\|x - x'\|_p$ – p-норма викривлення вхідного зразка для формування змагального зразка;

c – гіперпараметр оптимізаційної задачі;

J – функція втрат, що обчислюється на виході моделі аналізу даних.

Генерація змагальних зразків з амплітудою ε за FGSM-алгоритмом може бути виконана в один крок, що збільшує функцію втрат J за процедурою

$$x' = x + \varepsilon \cdot sign[\nabla_x J(\theta, x, y)].$$

У випадку таргетованої атаки зі змагальною міткою y', де $y' \neq y$, процедура FGSM-алгоритму дещо модифікується для зменшення крос-ентропії J між реальним розподілом ймовірності на виході мережі і бажаним для зловмисника розподілом

$$x' = x - \varepsilon \cdot sign[\nabla_x J(\theta, x, y')].$$

Алгоритм PGD є удосконаленням FGSM за рахунок реалізації ітераційної процедури з T кроків і ампілітудою модифікації вхідного зразка $\alpha = \varepsilon / T$. З урахуванням обмеження амплітуди результуючої модифікації функцією $Clip$ процедура PGD матиме наступний вигляд

$$x'_{t+1} = Clip\left\{x'_t + \alpha \cdot sign[\nabla_x J(\theta, x'_t, y)]\right\}.$$

Для підвищення надійності змагальних атак в C&W-атаках мета оптимізації полягає в ітеративному пошуці змагальних зразків, утворених найменшим збуренням, яке викликає найбільшу ймовірність помилкового

рішення. Для досягнення даної мети в ітераційній процедурі функція втрат замінена на сурогатну-функцію $\hat{f}(\cdot)$

$$x'_{t+1} = Clip\left\{x'_t + \alpha \cdot sign[\nabla_x \hat{f}(x')]\right\},$$

$$\hat{f}(x') = \max\left(z_y(x'_t, \theta) - z_{y_{\max} \neq y}(x'_t, \theta), -k\right),$$

де z_y – логіт-функція класу y (вихід нейромережі до softmax-нормалізації);

$z_{y_{\max} \neq y}$ – максимальне значення логіт-функції серед інших класів;

k – параметр, що регулює впевненість атаки.

Для формування змагальних атак чорної шухляди використовують різноманітні методи побудови сурогатної моделі, методи оцінювання градієнту та різноманітні евристичні алгоритми.

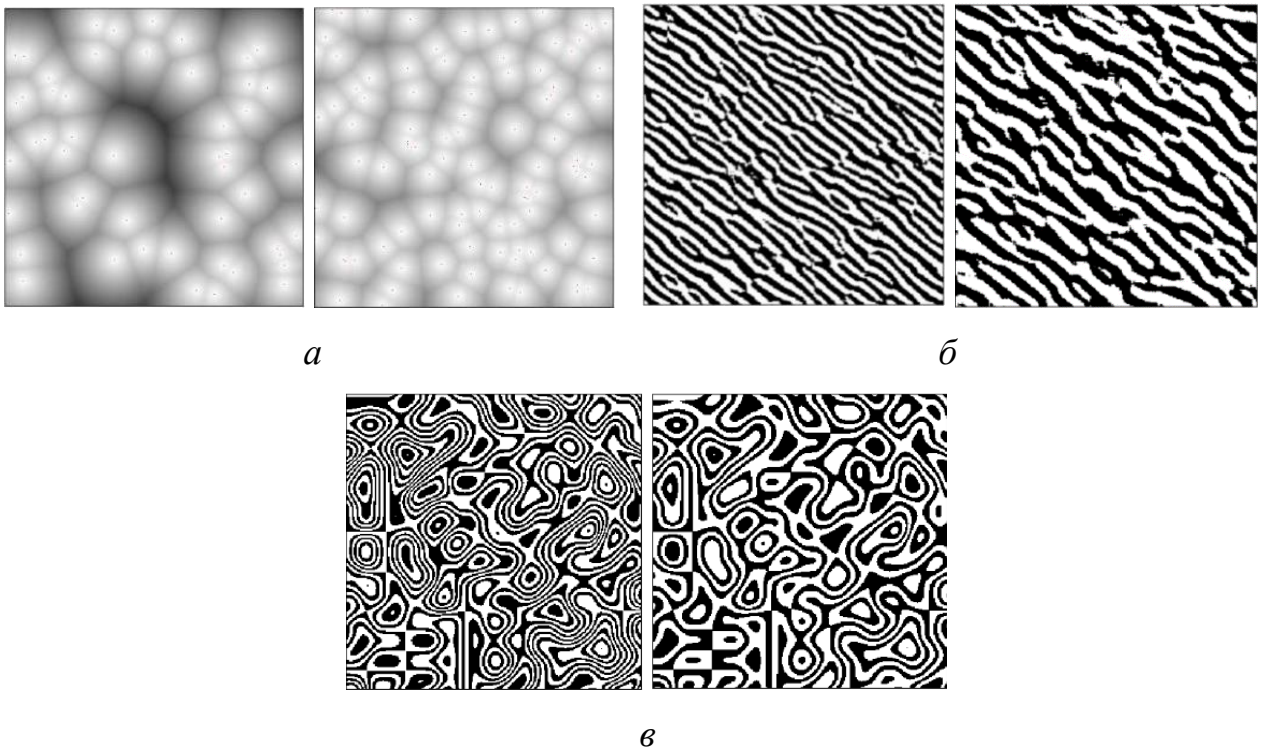
Для побудови сурогатної моделі зловмисник подає дані на цільову модель і використовує вихідні значення моделі для розмітка даних [43]. Отримані розмічені дані можуть використовуватися для тонкої настройки сурогатної моделі чи дистиляції знань в сурогатну модель. Після навчання сурогатної моделі можна використовувати методи білого ящика для генерації змагальних зразків. Однак ефективність подібних атак значною мірою залежить від якості навчання сурогатної моделі. Перенос змагальних атак, отриманих для сурогатної моделі, на оригінальну модель має низький рівень успіху в областях, де фігурують великі обсяги даних (наприклад, задача ImageNet), оскільки важко отримати якісний сурогат.

Інший підхід до формування атак чорної шухляди полягає в оцінюванні градієнту да допомогою оптимізації нульового порядку (zeroth-order optimization, ZOO) [43]. В даному випадку здійснюються запити до цільової моделі і обчислюється оцінка градієнту щодо відповідних вхідних спостережень, яка використовується для генерації атак за C&W-методом. Однак цей метод

досить обчислювально трудомісткий і потребує великої кількості запитів на одну ітерацію для точного оцінювання градієнту. З метою зниження обчислювальної складності у працях було запропоновано здійснювати оцінювання градієнту на основі жадібного алгоритму локального пошуку, де на кожній ітерації збуренню піддається лише невелика підмножина вхідних компонент. Однак в рамках даного підходу не вдається мінімізувати амплітуду збурення, що збільшує помітність змагальної атаки. Тому у працях [43] досліджуються більш реалістичні моделі змагальних атак шляхом визначення параметрів обмеження на запити до цільової моделі та параметри часткової інформації. В рамках даного підходу було розроблено методи атак, що основані на стратегіях природньої еволюції (Natural Evolutionary Strategies) та апроксимації Монте-Карло (Monte Carlo approximation).

Останнім часом на додаток до вищезгаданих підходів у формуванні змагальних атак було досліджено велику кількість евристичних алгоритмів. Найбільш простим серед них є метод атаки на межу рішень (Decision Boundary Attack). Даний метод починає пошук змагального збурення з дуже великих амплітуд, що спричиняє помилковість рішення на виході нейромережі. Потім здійснюється зменшення амплітуди збурення (тобто мінімізація спотворення) в процесі випадкового блукання на межі між правильним і неправильним рішенням, але залишаючись бути змагальним (таким, що призводить до помилкового рішення). Недоліками даного підходу є висока обчислювальна складність внаслідок необхідності у великій кількості запитів та відсутність гарантій щодо збіжності. У працях [44] автори формулюють дану задачу як задачу непервної дійсночислової оптимізації, що дозволяє отримати результат за обмежену кількість запитів на основі алгоритму оптимізації нульового порядку (zeroth-order optimization). Аналогічним шляхом в праці [45] було запропоновано пошук універсального (незалежно від вхідного спостереження) збурення (universal perturbation) для атаки чорної шухляди. У працях [46] було запропоновано формувати універсальні змагальні атаки чорної шухляди на

основі процедруного шуму, наприклад шуму Ворлі (Worley noise), шуму Габора (Gabor noise) або шуму Перліна (Perlin noise) (рис. 1.13).



а *б* *в*

Рисунок 1.13 – Шаблони процедурного шуму для зображень:
а – шум Ворлі; б – шум Габора; в – шум Перліна

Змагальні атаки, що основані на процедурному шумі, як правило, мають невелику кількість параметрів, які можуть бути оптимізовані за алгоритмом Баєсівської оптимізації для формування успішних універсальних атак. Експериментальні дослідження показали надзвичайно високу ефективність змагальних атак на основі процедурного шуму [46]. Проте даний підхід дозволяє формувати лише нетаргетовані атаки.

Методи формування атак постійно удосконалюються і комбінуються, випереджаючи методи захисту. Існує багато методів оцінювання робастності моделей щодо змагальних атак, однак сформовані оцінки подаються в метриках L0, L1, L2 або L∞ норми, що ускладнює порівняння методів і поточного стану проблеми. Крім того кожен метод оцінювання має свої обмеження, винятки і компроміси, що ускладнює задачу достовірного оцінювання робастності. При

цьому більшість змагальних атак розроблені для сценарію білого ящика, що не дозволяє аналізувати гібридні і нестандартні моделі аналізу даних.

Середовища реального світу є нестаціонарними і в них відбуваються непередбачувані зміни, які суперечать тому, що модель аналізу вивчила до цього. З часом продуктивність навченої моделі аналізу даних знижується внаслідок зміни статистичних властивостей даних. Ці зміни прийнято називати дрейфом концепцій, який можна класифікувати на такі основні типи [47]:

– реальний дрейф концепцій (Real Concept Drift);
– віртуальний дрейф концепцій (Virtual Concept Drift);
– зміна апріорних ймовірностей концепцій (Class Prior Concept Drift).

Реальний дрейф (або зсув, або зміна) концепцій полягає в зміні умовної ймовірності $P(y|x)$ без помітного впливу на $P(x)$. Це відбувається коли змінюється залежність між вхідними даними і цільовою змінною, тобто $P_t(y|x) \neq P_{t+1}(y|x)$ за умови $P_t(x) = P_{t+1}(x)$, де $P_t(y|x)$ є ймовірністю $P(Y = y|X = x)$ в момент часу t (рис. 1.14,*б*). У зв'язку з цим змінюються межі рішення, що знижує продуктивність моделі. Наприклад, у рекомендаційних системах настрій користувача може змінитися (змінна контексту), що призведе до того, що користувач буде вибирати інші рекомендовані елементи (мітки).

Віртуальний дрейф концепцій виникає коли відбуваються зміни в умовній ймовірності $P(X|y)$ без впливу на апостеріорну ймовірність $P(y|X)$. При цьому межа рішень залишається незмінною, оскільки вплив відбувається на розподіл даних в межах класу розпізнавання (рис. 1.14, *в*). Зміна в апріорних ймовірностях класів $P(y)$ може призводити до різних ефектів: незбалансованість класів (рис. 1.14, *г*), поява спостережень нового класу (рис. 1.14, г), злиття зразків класів (рис. 1.14, *д*).

З точки зору швидкості дрейфу концепцій їх можна класифікувати на раптові (abrupt, sudden), поступові (gradual), повторювані (recurring) та сплескові (blip) [47].

Раптовий дрейф концепцій полягає у швидкій зміні старої концепції на нову. При цьому раптово знижується ефективність моделі, існує потреба у швидкому навчанні новій концепції для відновлення продуктивності.

У випадку поступового дрейфу тривалість зміни концепції є відносно великою порівняно з раптовим дрейфом. Існує дві варіації цього типу дрейфу: повільний поступовий дрейф і нормальний поступовий дрейф. Наприклад, поступовий дрейф спостерігається, коли ринкові явища змінюються внаслідок інфляції або рецесії. Цей вид дрейфу має перекриваючу концепцію, і через деякий період часу нова концепція стає стабільною.

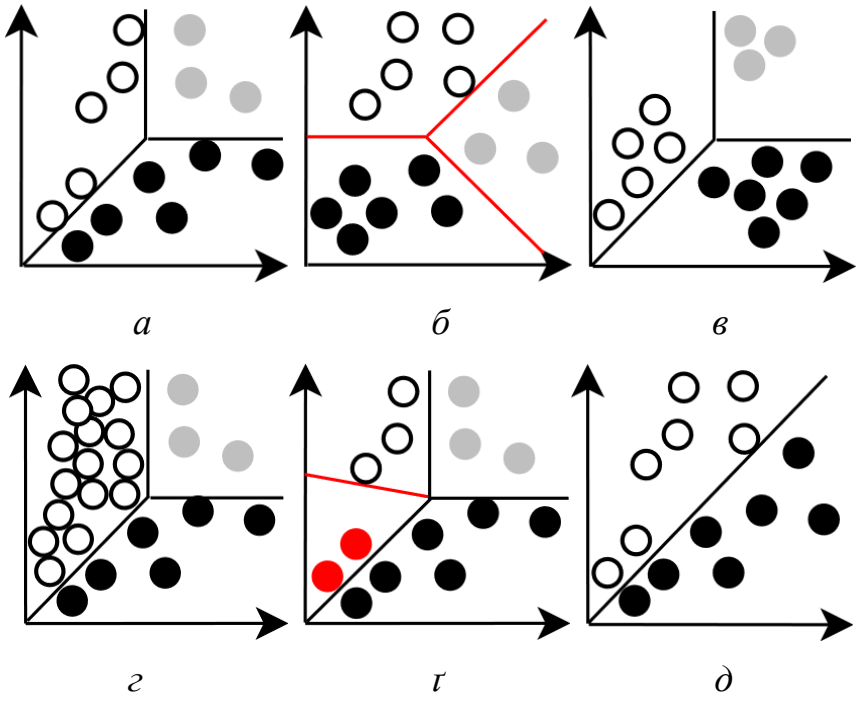


Рисунок 1.14 – Візуальна ілюстрація різних типів дрейфу концепцій на прикладі класифікатора на три класи: *а* – оригінальний розподіл даних і роздільна межа класифікаційних рішень; *б* – реальний дрейф концепцій; *в* – віртуальний дрейф концепцій; *г* – незбалансованість класів; *ґ* – новий клас; *д* – злиття зразків класів

У повторюваному виді дрейфу концепція знову з’являється через тривалий період часу, тобто в потоці відбувається повторювана зміна концепції. Такий дрейф може мати циклічну та ациклічну поведінку. Циклічний дрейф

відбувається в ситуації, коли відбуваються сезонні коливання. Наприклад, через літній сезон збільшується продаж холодних речей.

Ациклічний феномен спостерігається, коли ціна електроенергії збільшується через підвищення ціни на бензин і в звичайній ситуації вона повертається до попередньої ціни.

Сплексовий дрейф є дуже швидкою зміною в концепції або рідкісною подію, тому розглядається як викид (outlier) в стаціонарному розподілі. Тобто в загальному випадку сплесковий дрейф, як правило, навіть не вважається дрейфом концепцій.

Для адаптації до дрейфу концепцій необхідно детектувати дрейф, ідентифікувати його тип, донавчатися на нових даних. Для періодичного дрейфу необхідно вживати мір щодо уникнення катастрофічного забування старих концепцій (catastrophic forgetting). При цьому варто зауважити, що детектування і навчання як правило відбувається за рахунок наявності достовірних міток даних, але мітки можуть надходити зі значним запізненням, їх може бути замало чи можуть бути взагалі недоступними.

Будь-яка модель аналізу даних зіштовхується з алеаторичною (aleatoric) та епістемічною (epistemic) невизначеностями. У машинному навчанні алеаторна невизначеність — це невизначеність, що виникає через природну стохастичність даних, її можна порівняти зі здатністю моделі ефективно інтерполювати. З іншого боку, епістемічна невизначеність виникає через те, що навчальні дані недостатньо відображають дані, які спостерігаються моделлю. Тобто модель зустрічається з незнайомими їй даними [48]. Це може бути пов’язано з недостатністю навчальних даних у цій області, або це може бути через те, що дані повністю поза навчальним розподілом, і модель не змогла ефективно екстраполювати. Епістемічна невизначеність може приймати кілька форм – “відоме невідоме” та “невідоме невідоме”. “Відоме невідоме” характеризує невизначеність на відомих заздалегідь об'єктах, спостереження яких поза навчальним розподілом (out-of-distribution). “Невідоме невідоме” характеризує невизначеність на спостереженнях невідомих об’єктів поза навчальним

розподілом. Оброблення невизначеності типу “Невідоме невідоме” є більш складною проблемою, оскільки набагато складніше підготувати модель до чогось невідомого.

Небезпечним з практичної точки зору є непередбачуваність реакції моделі аналізу даних на новизну в даних. Коли моделі машинного навчання стикаються з даними, які виходять за межі розподілу, на якому вони були навчені, вони мають тенденцію до надмірної впевнеості у помилкових прогнозах. Така поведінка матиме катастрофічні наслідки для реальних систем машинного навчання. Вирішення даної проблеми полягає у застосуванні методів детектування аномалій (викидів) та змагальних атак і керованій деградації у випадках невизначеності. Проте питання керованої деградації в галузі штучного інтелекту покищо знаходиться на рівні постановки задачі.

Не менш шкідливим дестабілізуючим чинником для система машинного начання є пошкодження, пропуски і помилки в навчальних і тестових даних [49]. Пропуски в ознаках призводять до втрат інформації, а помилки в цільових мітках даних призводять до дезінформації і зниженні ефективності навчання.

Таким чином, до деструктивних факторів інтелетуальних систем належать інжекція апаратних несправностей, змагальні атаки, дрейф концепцій, новизна (дані поза навчальним розподілом), пропуски і помилки в даних. Існує багато видів і підвидів кожного з них. Кожен деструктивний фактор було досліджено у числених наукових працях, однак відсутні дослідження щодо комплексного їх впливу та ефективні методи захисту від такого сценарію.

1.4 Аналіз підходів щодо забезпечення резільєнтності систем штучного інтелекту

Основні принципи резільєнтності систем до деструктивних збурень сформульовані в роботах [50, 51, 52]. Вони передбачають наявність механізмів поглинання збурень, виявлення збурень, виточченої деградації, відновлення продуктивності та вдосконалення. В роботах [53, 54, 55] досліджено вразливість алгоритмів штучного інтелекту і визначено наступні деструктивні впливи: шум

та протиборчі атаки, збої та ін'єкція збоїв в середовищі розгортання інтелектуального алгоритму, дрейф концепції та поява новизни, тобто тестових прикладів, що знаходяться поза розподілом навчальних даних.

Здатність поглинати деструктивні збурення називається робастністю. Існує багато методів і підходів до підвищення робачтності до протиборчих атак [56, 57, 58]. Деякі дослідники поділяють методи забезпечення робастності до протиборчих атак на такі категорії: методи маскування градієнту, методи оптимізації робастності та методи виявлення протиборчих зразків [56, 59]. Маскування градієнту включає деякі методи попередньої обробки вхідних даних (стиснення jpeg, випадкове заповнення та зміна розміру [60], дискретне атомарне стиснення [61]), захисну дистиляцію [62], випадковий вибір моделі з набору моделей або використання відсіву [63], а також використання генеративних моделей (тобто PixelDefend [64] та Defense-GAN [65]). Проте в роботі [66] продемонстровано неефективність методів маскування градієнту. Підхід оснований на оптимізації робастності включає в себе протиборче навчання [67] та методи регуляризації, які мінімізують вплив малих збурень вхідних даних (наприклад, якобіанська регуляризація або L2-відстань між ознаковими поданнями для природної та збуреної вибірок) [68, 69], а також доказові засоби захисту (наприклад, алгоритм Reluplex [70]). До оптимізаційного підходу також відносяться методи подання ознак на основі розрідженого кодування, які забезпечують ефект низькочастотної фільтрації. Ці методи в основному реалізуються на основі L0-регуляризації, L1-регуляризації або подібних альтернатив [71]. Однак оптимізація робастності, як правило, вимагає значних витрат обчислювальних ресурсів для отримання хорошого результату. Нарешті, ще один підхід полягає в розробці детектора протиборчих зразків для відкидання таких зразків на вході основної моделі [72, 73, 74]. Однак, Карліні та Вагнер [75] довели, що властивості протиборчих зразків важко та ресурсоємно виявляти. В роботах [59, 76, 77] запропоновано розділити методи захисту від ворожих атак на дві групи, що реалізують два окремих принципи: методи підвищення внутрішньокласової компактності та міжкласового розділення векторів ознак і

методи маргіналізації або видалення неробастних ознак зображення. Потенціал подальшого розвитку цих фундаментальних принципів та їх комбінування з урахуванням додаткових вимог та обмежень висвітлено в роботах [78, 79].

Існує три основні підходи до забезпечення робастності до ін'єкції несправностей в обчислювальне середовище, де розгортаються нейронні мережі: введення явної надмірності [80, 81], модифікація алгоритму навчання [82] та оптимізація архітектури [83]. Під несправностями розуміють випадкові або навмисні інверсії бітів у пам'яті, яка зберігає вагові коефіцієнти або вихідне значення нейрона. Введення явної надмірності досягається, як правило, дублюванням критичних нейронів і синапсів, рівномірним розподілом синаптичних ваг і видаленням неважливих ваг і нейронів. Також можна підвищити робастність нейронної мережі до ін'єкції несправностей на етапі машинного навчання шляхом додавання шуму, збурень або ін'єкції прямих помилок під час навчання [82]. Цього ж можна досягти шляхом включення регуляризаційного (штрафного) компонента в функцію втрат для опосередкованого включення помилок в звичайні алгоритми. Оптимізація архітектури для підвищення робастності означає мінімізацію максимальної помилки на виході нейронної мережі при заданій кількості інвертованих бітів в пам'яті, де зберігаються ваги або результати проміжних обчислень. Деякі автори [83] вирішували цю проблему за допомогою еволюційних алгоритмів пошуку. Однак оптимізація архітектури традиційно є дуже ресурсоємним процесом.

В роботах [84, 85] пропонуються методи рандомізації домену та протибочого розширення домену, які підвищують робастність моделі при обмежених зміщеннях розподілу даних. Рандомізація області – це генерація синтетичних даних з достатньо великою кількістю варіацій, щоб реальні дані розглядалися як просто ще одна варіація домену [84]. В цьому випадку під час синтезу даних може застосовуватися рандомізація кутів огляду, текстур, форм, шейдерів, ефектів камери, масштабування та багатьох інших параметрів. Протиборче розширення доменів створює кілька доповнених доменів з

вихідного домену, використовуючи протиборче навчання з послабленим обмеженням розбіжності доменів на основі автокодера Вассерштейна [85]. Навчання з перенесенням і навчання багатозадачних або багатоджерельних доменів також посилюють стійкість до збурень, пов'язаних з викидом поза з розподіл навчальних даних [86]. Однак, якщо в потоці даних відбувається реальний дрейф концепцій, виникає необхідність виявлення такої ситуації та впровадження реактивних механізмів адаптації [87]. Існують дослідження щодо адаптації до реального дрейфу концепцій, але проблемою залишається відсутність міток для тестових даних або значна затримка в їх отриманні. Одним з успішних підходів до зменшення вимог до кількості даних та підвищення узагальнюючої здатності моделі є передача інформації про мітки через гетерогенні домени, але міждоменна інформація може бути недоступною для деяких додатків [88, 89].

Протиборчі атаки, ін'єкція несправностей, дрейф концепцій та зразки даних поза навчальним розподілом не завжди можуть бути поглинуті, тому актуальною залишається розробка механізмів реактивної резільєнтності, а саме поступової деградації, відновлення та покращення [50, 56]. В роботі [90] запропоновано механізми вкладеного навчання та ієрархічної класифікації, які є особливо корисними для реалізації механізму поступової деградації. Однак реалізація цих механізмів часто пов'язана з необхідністю виявлення збурення. Найбільш успішні методи виявлення протиборчих зразків та зразків поза розподілом і дрейфу концепцій базуються на аналізі високорівневого простору ознак з використанням дистанційної довірчої оцінки або класифікатора на основі прототипів [74, 91, 92]. В роботах [93, 94] для виявлення зміни ваг нейронної мережі під впливом несправностей в пам'яті пропонується використовувати контрольну суму та низькоколізійну хеш-функцію. В роботі [95] механізм виявлення несправностей, що впливають на режим екзамену, базується на обчисленні еталонного значення контрастної функції втрат на тестових діагностичних вибірках даних за відсутності несправностей. Для виявлення несправностей поточне значення функції контрастних втрат для діагностичних

даних порівнюється з еталонним значенням. Також відновлення пошкоджених ваг нейронної мережі може бути реалізовано шляхом точного налаштування [95, 96].

В роботах [97, 98] розглядаються алгоритми адаптації моделей до деструктивних збурень, де використовуються принципи активного навчання або контрастного навчання для прискорення адаптації за рахунок зменшення вимоги до кількості мічених даних. В роботі [99] запропоновано методи напівкерованого навчання для одночасного використання як мічених, так і немічених даних з метою прискорення адаптації до дрейфу концепцій. Методи навчання впродовж життєвого циклу, які дозволяють безперервно накопичувати знання з різних завдань і вдосконалюватися, а також різні механізми нагадування, що допомагають уникнути катастрофічної проблеми забування, розглянуті в [54, 100]. Різні підходи до реалізації мета-навчання для підвищення ефективності адаптації висвітлено в [53, 100]. В роботі [101] з метою підвищення ефективності навчання нижніх шарів нейронних мереж, окрім покращення продуктивності моделі, розглядається принцип само-дистиляції для навчання нейронних мереж, який дозволяє реалізувати адаптивні обчислення та прискорити режим виводу (екзамену). Ця властивість покращення виводу не розглядалася в контексті впливу збурень, але потенційно може покращити резільєнтність алгоритму.

У працях [56, 52] показано, що найбільш вразливими до деструктивних збурень є моделі класифікаційного аналізу даних. При цьому класифікаційний аналізв відбувається як в детекторах об’єктів так і сегментаторах зображень. В табл. 1.1 наведено можливості різних підходів до побудови моделей та алгоритмів навчання для забезпечення резільєнтності класифікатора даних до протиборчих атак, інжекції несправностей та дрейфу концепцій.

Аналіз літературних джерел дозволяє зробити висновок, що більшість досліджень присвячено окремим принципам резільєнтності СШІ, але практично відсутні роботи, в яких розглядається їх одночасне поєднання для забезпечення синергетичного ефекту. Аналіз показує, що відомі підходи, які реалізують окремі

властивості резільєнтності, не враховують принципи раціональної резільєнтності [51], що є актуальним в умовах обмежених ресурсів.

Таблиця 1.1 – Огляд підходів щодо забезпечення резільєнтності систем штучного інтелекту

Мета	Підхід	Можливості	Слабкі сторони	Алгоритм
Резільєнтність до протиборчих атак	Маскування градієнту	Поглинання збурення	Вразливість до атак на основі градієнтної апроксимації або оптимізації "чорного ящика" зі стратегіями еволюції	Недиференційовані трансформації даних на вході моделі [60, 61]
				Захисна дистиляція [62]
				Вибір моделі випадковим чином із сімейства моделей [63]
				Генеративні моделі PixelDefend або Defense-GAN [64, 65]
	Оптимізація робастності	Поглинання збурень та відновлення ефективності	Значне споживання обчислювальних ресурсів для отримання прийнятних результатів	Протиборче навчання [67]
				Стабілізаційне навчання [68]
				Якобіанова регіляризація [69]
				Подання даних, основане не розрідженому кодуванні [71]
				Регуляризація основана на забезпеченні внутрі-класової компактності та міжкласової розділимості [76, 77, 78]
				Доказовий захист з Reluplex-алгоритмом [70]
	Детектування протиборчих зразків	Витончена деградація	Недостатня надійність	Спрощене Байсовське уточнення [72]
				Використання графу прихованого сусідства [73]
				Аналіз відстаней в просторі ознак [74]

Продоження таблиці 1.1

Мета	Підхід	Можливості	Слабкі сторони	Алгоритм
Резільєнт-ність до несправ-ностей	Надлиш-ковість та маскування помилок	Поглинання збурень	Обчислювально затратний процес синтезу моделі та режиму виведення	Явна надлишковість [80]
				Подання ваг кодами, що коректують помилки [81]
				Відмовостійкісне навчання на основі ін'єкції відмов під час навчання [82]
				Алгоритми пошуку нейро-мережевої архітектури [83]
	Детекту-вання відмов	Витончена деградація та відновлення шляхом завантаження неушкодженої копії ваг	Модель не покращується і інформація з вразливих ваг не розподіляється між іншими нейронами	Кодування вразливих ваг моделі з використанням хеш функцій з низьким рівнем колізій [93]
				Алгоритм на основі контрольних сум, що обчислює низьковимірний двійковий підпис для кожної групи ваг [94]
	Активне відновлення	Відновлення ефективності	Невисока надійність відновлення	Контрастне точне на-лаштування на діаг-ностичних даних [95]
				Механізм зсуву ваг в картах нейронів, що самоорганізуються [96]
Резільєнт-ність до дрейфу концепцій	Узагаль-нення поза межами домену	Поглинання збурень	Застосовується лише для протидії віртуальному дрейфу концепцій і є малоефективним у випадку реального дрейфу концепцій	Рандомізація домену [84]
				Протиборче розши-рення набору даних [85]
				Інваріантний до доме-ну ознаковий опис [86]
				Поширення знань ге-терогенних доменів [88, 89]

Продовження таблиці 1.1

Мета	Підхід	Можли-вості	Слабкі сторони	Алгоритм
Резільєнт-ність до дрейфу концепцій	Детекту-вання дрейфу	Витон-чена дегра-дація	Збільшення ефективності детектування дрейфу за рахунок зменшення здат-ності до швидкої адаптації	Вбудовування обмежень [91]
				Мета-навчання для виявлення дрейфу концепцій [92]
	Неперервна адаптація	Відновлення ефективності та покращення	Необхідність у реалізації комплексних механізмів для запобігання катастрофічному забуванню і прискорення адаптації	Адаптивний підбір диверсного ансамблю [87]
				Неперервне навчання [97]
				Активне навчання [98]
				Навчання з частковим залученням учителя [99]
				Неперервне мета-навчання [100]

Таким чином, існує нагальна потреба в розробці нових моделей і методів оцінювання та забезпечення резільєнтності моделей до відомих збурень з урахуванням ресурсних обмежень.

2 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОЦІНЮВАННЯ ТА ЗАБЕЗПЕЧЕННЯ РЕЗІЛЬЄНТНОСТІ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

2.1 Показники резільєнтності систем штучного інтелекту

Існують різноманітні підходи щодо формування показників резільєнтності [9, 12]. Проте більша частина досліджень присв'ячена аналізу кривих резільєнтності, побудованих у координатах часу і показника продуктивності системи, що описують реакцію резільєнтної системи на деструктивний збурюючий вплив. (рис. 2.1). У працях [9, 14, 15] було запропоновано такі основні показники резільєнтності системи:

– час відгуку (реагування) T_{res} (Response Time);

– час відновлення T_{rec} (Recovery Time);

– просідання продуктивності A (Performance Attenuation);

– втрата продуктивності L (Performance Loss);

– робастність R (Robustness);

– швидкість відновлення θ (Rapidity);

– надлишковість (Redundancy);

– винахідливість (Resourcefulness);

– інтегральна міра резільєнтності Re (Integrated measure of resilience).

Час відгуку (T_{res}) характеризує своєчасність реагування на деструктивний збурюючий вплив. Системи з коротким часом відгуку краще пом'якшують вплив, зменшуючи ослаблення продуктивності, викликане збурюючим впливом.

Час відновлення (T_{rec}) – це період, необхідний для відновлення функціональності системи до бажаного рівня, при якому система може функціонувати так само, близько або краще, ніж до збурюючого впливу.

Просідання продуктивності (A) описує максимальне зменшення продуктивності системи в результаті збурюючого впливу, в той час як показник втрати продуктивності (L) характеризує загальні втрати продуктивності на

етапах реагування та відновлення. Втрата продуктивності представлена площею, що виділена темнішим (зеленим), на рис. 2.1.

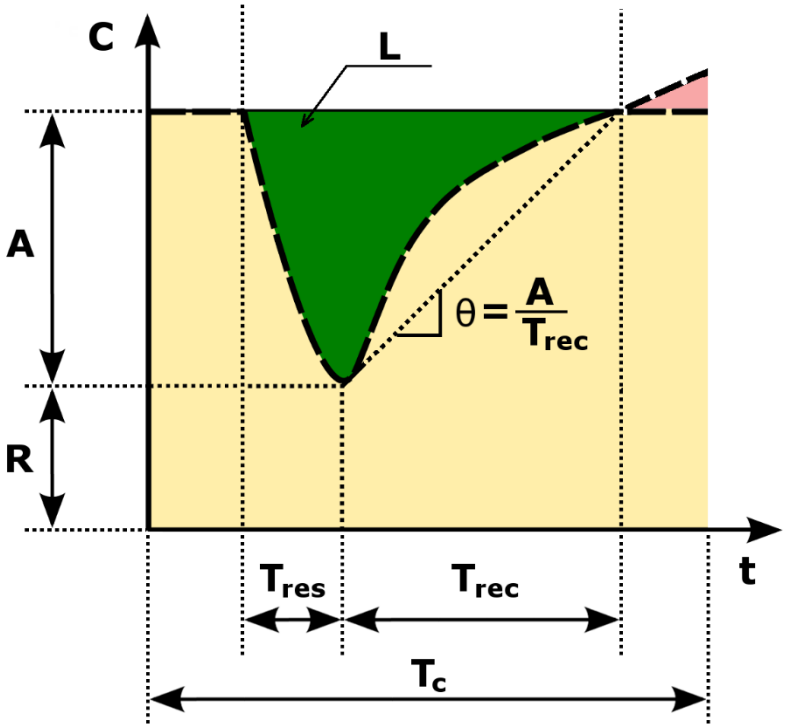


Рисунок 2.1 – Ілюстрація кривої резільєнтності
та основних її показників

Робастність (R) характеризує здатність системи витримувати певний рівень стресу, зберігаючи функціональність, тобто без значного погіршення чи втрати продуктивності. Робастність дозволяє системі поглинати й протистояти деструктивним впливам. Система з високим ступенем робастності збереже основну частину функціональних характеристик протягом впливу деструктивних факторів. Робастність можна визначити як залишкову функціональність після впливу екстремального деструктивного збурення і обчислити за наступною формулою

$$R = 1 - \tilde{A}(m_A, \sigma_A), \qquad (2.1.1)$$

де $\tilde{A}$ – випадкова змінна, виражена як функція від середнього значення m_A та середньоквадратичного відхилення σ_A для показника просідання продуктивності.

Швидкість відновлення (θ) – це здатність своєчасно відновлювати функціональність, обмежуючи втрати і уникаючи майбутніх збоїв. Математично швидкість відновлення є нахилом кривої продуктивності протягом періоду відновлення (рис. 2.1), що обчислюється за формулою

$$\theta = \frac{dC(t)}{dt}, \qquad (2.1.2)$$

де d/dt – оператор диференціювання;

$C(t)$ – функція, що описує залежність продуктивності від часу.

Усереднену оцінку швидкості відновлення можна визначити за такою формулою

$$\theta = \frac{A}{T_{rec}}. \qquad (2.1.3)$$

Надлишковість характеризує наявність альтернативних ресурсів на стадії відновлення, коли первинні ресурси недостатні. Також надлишковість визначають як міру наявності альтернативних шляхів у структурі системи, за допомогою яких можна передавати підтримуючі сили для забезпечення стабільності після відмови будь-якого елементу. Структурна надлишковість передбачає доступність множини підтримуючих компонентів, які зможуть витримати додаткові навантаження у разі відмови окремих основних компонентів. Тобто, якщо один чи декілька компонентів виходять з ладу, то структура, що залишилася, здатна перерозподілити навантаження і запобігти виходу з ладу всієї системи.

Винахідливість системи полягає у здатності діагностувати проблеми, розставляти пріоритети та ініціювати вирішення проблем шляхом виявлення та мобілізації матеріальних, грошових, інформаційних, технологічних та людських ресурсів (рис. 2.2).

Винахідливість і надлишковість тісно взаємопов'язані, наприклад, винахідливість може створити надлишковості, яких раніше не існувало. Крім того, винахідливість і надмірність можуть вплинути на швидкість і час відновлення. Як показано на рис. 2.2, де додається третя вісь для врахування винахідливості, додавання ресурсів може скоротити час відновлення порівняно з очікуваним за стандарних умов.

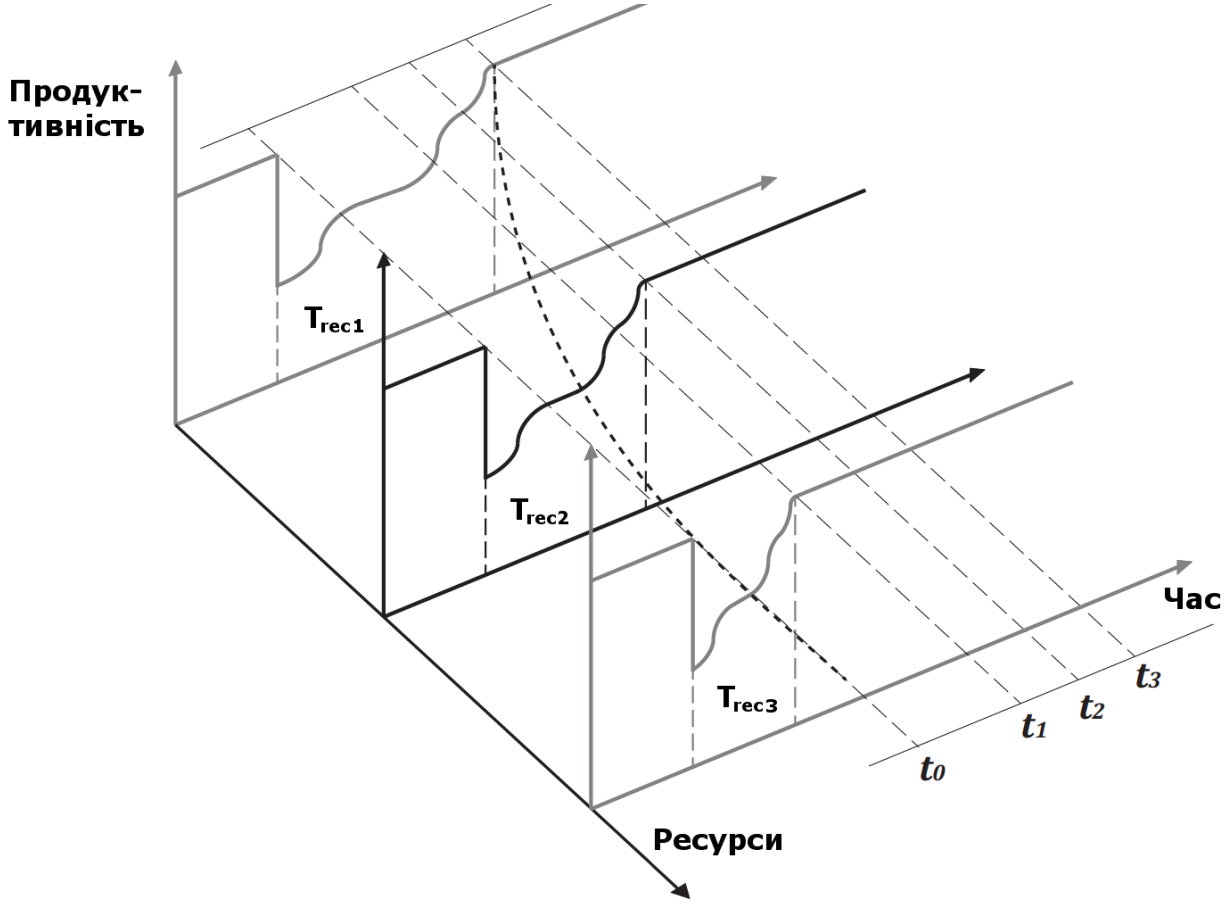


Рисунок 2.2 – Залежність часу відновлення від обсягу задіяних механізмом винахідливості додаткових ресурсів

Теоретично, якби були доступні нескінченні ресурси, то час відновлення асимптотично б наближався до нуля. На практиці, навіть за наявності величезних фінансових і трудових можливостей існує деякий мінімальний час відновлення. Проте час відновлення може бути досить високим навіть за наявності великого обсягу ресурсів через неадекватне планування, організаційні невдачі чи нефективну політику [16]. Винахідливість та робастність також пов’язані між собою. Можна стверджувати, що інвестування в обмеження початкових втрат (підвищення робастності) в деяких випадках може бути кращим підходом для підвищення резільєнтності, оскільки це автоматично призводить до подальшого скорочення часу відновлення; інвестиції в модернізацію – це інвестиції, які приносять вигоди по обом осям.

Під час проектування та експлуатації резільєнтних систем з урахуванням ресурсних обмежень часто звертаються до принципів раціональної резільєнтності (Affordable Resilience) [51]. Раціональна резільєнтність передбачає досягнення ефективного балансу між вартістю життєвого циклу та технічними характеристиками резільєнтності системи. Під час розгляду життєвого циклу щодо раціональної резільєнтності потрібно враховувати не лише ризики та проблеми, пов’язані з відомими та невідомими збуреннями в часі, але й можливості пошуку виграшу у відомих і невідомих майбутніх середовищах. Для того шоб досягти раціональної резільєнтності, як це показано на рис. 2.3., потрібно збалансувати приведену вартість затрат та приведену цінність отриманої резільєнтності.

Після визначення раціональних рівнів резільєнтності для кожного ключового показника продуктивності, пріоритети цих показників можуть бути визначені на основі теорії багатокритеріальної корисності (Multi-attribute Utility Theory) [12], або методом обробки аналітичних ієрархій (Analytical Hierarchy Process) [13]. Як правило, пріоритети показників продуктивності резільєнтної системи залежать від прикладної доменної області.

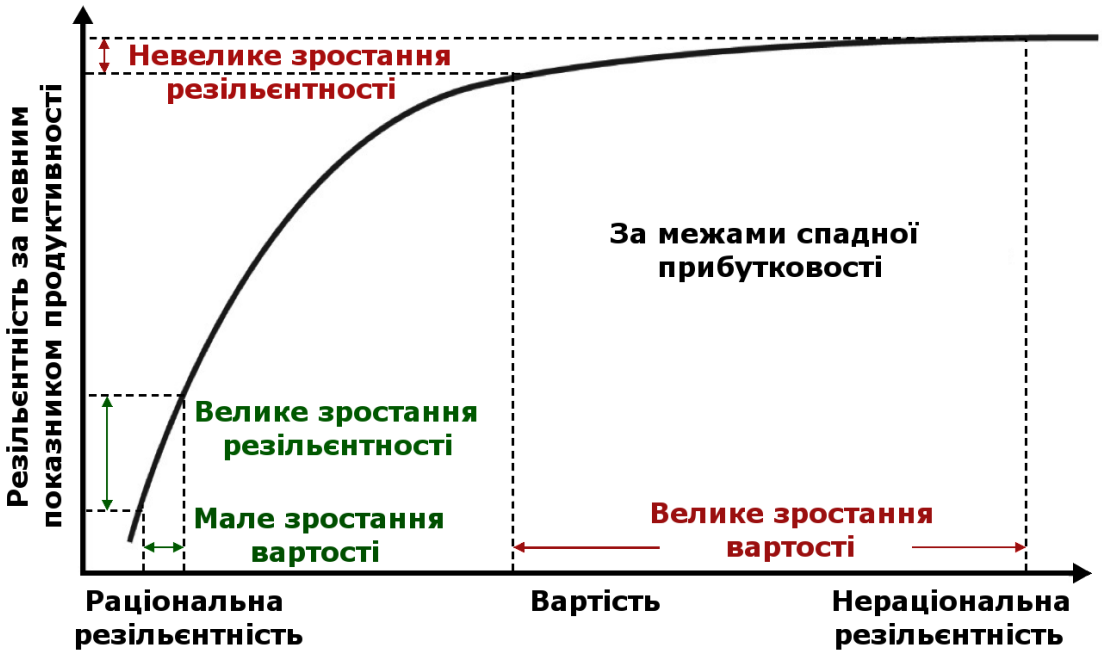


Рисунок 2.3 – Крива вартості забезпечення резільєнтності системи

Для одночасного врахування змінних часу і продуктивності під час оцінювання резільєнтності системи були розроблені різноманітні варіанти інтегральних показників [17, 18]. Ці показники, як правило, характеризують різницю або відношення номінальної продуктивності та втрати продуктивності в часі внаслідок збурюючих впливів. Для зручності інтегральний показник можна виразити у нормалізованій формі

$$R \equiv \frac{\frac{1}{|E|}\sum_{E}\int_{t=0}^{T_c} C(t)dt}{\int_{t=0}^{T_c} C^{nominal}(t)dt}. \qquad (2.1.4)$$

$C(t)$ – залежність поточного значення продуктивності або функціональності системи від часу;

$C^{nominal}(t)$ – нормальний (номінальний) функціональний стан, що вводиться до формули для відображення значень інтегрального показника резільєнтності в інтервал [0, 1];

T_c – контрольний час, що обирається за результатами попереднього оцінювання усередненого інтервалу між подіями деструктивних впливів;

E – множина деструктивних подій за інтервал контрольного періоду [0, Tc].

Для оцінювання та сертифікації показників резільєнтності системи можуть бути використані аналітичні, імітаційні та тестові методи [15, 17]. Аналітичні методи потребують декомпозиції системи і аспектів її функціонування, глибокий аналіз її структури і елементів для отримання аналітичного виразу резільєнтності. Метод симуляції полягає у заміні системи, що вивчається, на імітаційну модель, що з достатньою точністю описує поведінку системи. Саме з такою моделлю проводяться експерименти. Багато складних систем дуже важко або неможливо описати аналітично, або існуючі аналітичні моделі не дозволяють отримати стійке рішення, тому метод симуляції, є виграшним. Метод тестування найпростіший, але вимагає доступного прототипу системи, і може проводитися за принципами прозорої чи чорної шухляди. У багатьох випадках тестування реального протипу системи є або небезпечним або дуже дорогим. Тому ефективність всіх методів оцінювання резільєнтності залежить від прикладної області, від специфіки самої системи.

Таким чином, основними показниками резільєнтних систем є робастність, швидкість відновлення, надлишковість, винахідливість та інтегральні міри резільєнтності. При цьому їх оцінювання здійснюється на основі аналітичного, імітаційного або тестувального методів.

В роботах [1, 2] для оптимізації параметрів та гіперпараметрів g системи з урахуванням ресурсних обмежень пропонується знаходити компроміс між критерієм ефективності J в нормальних умовах та інтегральним показником резільєнтності системи R в умовах дії збурень, тобто

$$g^* = arg\max_{G}\left\{\eta\bar{J}(g) + (1-\eta)R(g)\right\} \qquad (2.1.5)$$

де η – коефіцієнт, що регулює компроміс між критерієм функціональної ефективності та інтегральним критерієм резільєнтності протягом котрольованого періоду.

2.2 Критерії функціональної ефективності

Центральним питанням інформаційного синтезу систем інтелектуального аналізу даних є оцінка функціональної ефективності процесу навчання моделі, яка визначає максимальну асимптотичну достовірність рішень, що приймаються на екзамені. Як критерій функціональної ефективності (КФЕ) можуть використовуватися різні валідаційні метрики, що обчислюються на навчальній та тестовій вибірках.

Валідаційні критерії функціональної ефективності класифікаційної моделі обчислюються на основі підрахунку результатів статистичних тестів. Для одного тестового зразка можливі 4-ри можливі результати статистичного тесту [1, 2, 5]:

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, частота помилок першого та другого роду, прецизійність, точність 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

$$\text{Sensitivity} = \text{Recall} = TP / (TP+FN). \qquad (2.2.1)$$

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

$$\text{Specificity} = \text{TNR} = \text{TN} / (\text{FP}+\text{TN}). \quad (2.2.2)$$

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

$$\text{FPR} = \text{FP}/(\text{FP}+\text{TN}). \quad (2.2.3)$$

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

$$\text{FNR} = \text{FN}/(\text{FN}+\text{TP}). \quad (2.2.4)$$

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

$$\text{Precision} = \text{TP}/(\text{TP}+\text{FP}). \quad (2.2.5)$$

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

$$\text{Accuracy} = (\text{TP}+\text{TN}) / (\text{TP}+\text{TN}+\text{FP}+\text{FN}) \quad (2.2.6)$$

Однак точність може бути адекватною метрикою тільки якщо кількість зразків кожного класу однакова, тобто класи є збалансованими. Для випадку незбалансованості класів більш ефективним є використання міра F1. Міра F1 (F) – гармонічне середнє між точністю та чутливістю, значення якої лежить в діапазоні [0, 1], тобто

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}. \quad (2.2.7)$$

Головна мета машинного навчання в системах штучного інтелекту це побудова моделей, які в режимі екзамену зменшують невизначеність щодо спостережуваного процесу. Тобто у загальному випадку КФЕ повинен бути мірою зняття невизначеності, тобто мати інформаційну природу. Інформаційні КФЕ повинні задовольняти таким властивостям:

– 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.

Використання інформаційних мір під час оптимізації параметрів моделі аналізу даних найбільшого розвитку набуло в рамках так званої інформаційно-екстремальної інтелектуальної технології (ІЕІ-технології), де використовуються статистичні логарифмічні інформаційні міри виражені як функціонали точністних характеристики моделі [1, 6, 78]. В рамка ІЕІ-технології розглядають нормовані модифікації інформаційних мір. Наприклад, нормований ентропійний КФЕ навчання системи розпізнавати реалізації класу X_m^o має вигляд:

$$J_m^{(k)} = \frac{I_m^{(k)}}{I_{max}^{(k)}} = \frac{H_m^{(k)} - H_m^{(k)}(\gamma)}{H_m^{(k)}}, \quad (2.2.8)$$

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

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

$$H_m^{(k)} = -\sum_{l=1}^{M} p(\gamma_{l,k})\, log_2\, p(\gamma_{l,k}) \qquad (2.2.9)$$

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

$$H_m^{(k)}(\gamma) = -\sum_{l=1}^{M}\sum_{m=1}^{M} p(\gamma_{l,k}) p(\mu_{m,k}/\gamma_{l,k})\, log_2\, p(\mu_{m,k}/\gamma_{l,k}) \qquad (2.2.10)$$

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

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

$p(\gamma_{l,k})$– 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 (2.2.9) і (2.2.10) у вираз (2.2.8) та заміни відповідних апостеріорних ймовірностей на апріорні за формулою Байєса ентропійний критерій набуває вигляду:

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

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

$$+\frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d)+D_{2,m}^{(k)}(d)}\,log_2\frac{D_{2,m}^{(k)}(d)}{\alpha_m^{(k)}(d)+D_{2,m}^{(k)}(d)}\Bigg), \qquad (2.2.11)$$

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

$\beta_m^{(k)}(\mathrm{d})$ – ймовірність помилок другого роду;

$D_{1,m}^{(k)}(\mathrm{d})$ – перша достовірність (чутливість);

$D_{2,m}^{(k)}(\mathrm{d})$ – друга достовірність (специфічність).

Для оптимізації параметрів глибоких нейронних мереж використовують градієнтні алгоритми і функцію втрат, що враховує відмінність прогнозів і очікуваних (цільових) значень виходів моделі, а також регуляризаційні складові. Однак справжня мета навчання, це отримання найкращих результатів на валідаційних критеріях, які зручніше інтерпретувати і аналізувати. Нелінійний зв'язок між функцією втрат, що мінімізується, та валідаційними критеріями, що максимізуються, робить процес навчання більш тривалим і нестабільним. Виходом з цієї ситуації є використання одних і тих самих критеріїв чи їх наближень як для оптимізації, так і для валідації. Функція втрат на основі використання інформаційної міри може мати наступний вигляд:

$$L_{INF} = 1 - \bar{J}\ , \qquad (2.2.12)$$

де $\bar{J}$ – усереднене за алфавітом класів значення інформаційного критерію ефективності (2.1.16).

Основне ускладнення прямого використання функції втрат (2.2.12) є її недиференційованість, оскільки процедури обчислення статистичних тестів є недиференційованими. Проте можна скористатися згладженими (smoothed) версіями результатів статистичних тестів [31]:

$$TP \approx \sum_{i=1}^{n} \hat{y}_i \odot y_i, \tag{2.2.13}$$

$$FP \approx \sum_{i=1}^{n} \hat{y}_i \odot (1 - y_i), \tag{2.2.14}$$

$$FN \approx \sum_{i=1}^{n} (1 - \hat{y}_i) \odot y_i, \tag{2.2.15}$$

$$TN \approx \sum_{i=1}^{n} (1 - \hat{y}_i) \odot (1 - y_i), \tag{2.2.16}$$

де $\odot$ – оператор поелементного множення (добуток Адамара);

$y_i = \{y_{i,k} |\ k = \overline{1,K}\}$ – мітки класів для і-го зразка в one-hot форматі [2];

$\hat{y}_i = \{relu(\mu_{i,k}) |\ k = \overline{1,K}\}$ – значення функції належності після функції relu() для і-го зразка.

Для уникнення невизначеностей при діленні на нуль чи взятті логарифму з нуля у наступних розрахунках точністні характеристики потрібно модифікувати шляхом введення в формулу невеликиого невід'ємного числа ε ($\varepsilon = 10^{-6}$) наступним чином:

$$D_{1,k} = \frac{TP_k}{TP_k + FN_k + \varepsilon} + \varepsilon, \tag{2.2.17}$$

$$D_{2,k} = \frac{TN_k}{TN_k + FP_k + \varepsilon} + \varepsilon, \tag{2.2.18}$$

$$\alpha_k = \frac{FN_k}{FN_k + TP_k + \varepsilon} + \varepsilon, \tag{2.2.19}$$

$$\beta_k = \frac{FP_k}{FP_k + TN_k + \varepsilon} + \varepsilon, \tag{2.2.20}$$

Робоча область критерію (2.2.11) обмежена нерівностями $D_{1,k} \geq 0{,}5$ і $D_{2,k} \geq 0{,}5$, або $\beta_k < 0{,}5$ і $\alpha_k < 0{,}5$, то для зручності оптимізації методом

зворотного поширення помилки пропонується виконати наступні операції під час обчислення функції втрат [2]:

$$D_{1,k} = max(D_{1,k},\ 0{,}5), \tag{2.2.21}$$

$$D_{2,k} = max(D_{2,k},\ 0{,}5), \tag{2.2.22}$$

$$\alpha_k = min(\alpha_k,\ 0{,}5), \tag{2.2.23}$$

$$\beta_k = min(\beta_k,\ 0{,}5). \tag{2.2.24}$$

До функції втрат, що кодує інформаційний критерій ефективності, можуть бути додані компоненти регуляризації, спрямовані на підвищення стійкості до деструктивних збурюючих впливів і прискорення процесу оптимізації параметрів.

2.3 Моделі та алгоритми оцінювання та сертифікації резільєнтності

Різні ваги моделі мають різну важливість та вплив на продуктивність моделі. Крім того, помилка в старших розрядах значення тензора призводить до більшого спотворення результатів, ніж помилка в молодших розрядах. Аналогічно, ефективність протиборчих атак з однаковим амплітудним обмеженням може сильно відрізнятися в залежності від просторового розподілу збурених пікселів. Тому для оцінки та порівняння резільєнтності моделі до пошкоджених тензорів або збурених зображень слід використовувати статистичні характеристики. Статистичні характеристики отримуються з великої кількості експериментів, де біти та тензори для інверсії вибираються випадковим чином з рівномірного розподілу або протиборчих зображень, що генеруються оптимізатором чорного ящика з рандомізованою стратегією еволюції. Для спрощення можна розглядати медіанне значення (MED) та інтерквартильне значення (IRQ) точності (Acc) або прецизійності (Prec) класифікатора при

збуреннях та необхідну кількість кроків відновлення працездатності (Rec_steps), розраховану після 1000 експериментів.

Для тестування моделі на відмовостійкість та стійкість рекомендується використовувати бібліотеку TensorFI2, яка здатна емулювати програмні та апаратні відмови [106]. В експериментах пропонується розглянути вплив найбільш складного для поглинання типу відмов – ін'єкції випадкової інверсії бітів в кожному шарі моделі, з випадково вибраною фіксованою часткою тензора (частотою відмов) та одним випадково вибраним бітом для інверсії. Діагностичні дані додаються до кожної моделі для діагностики та відновлення разом з тестовими даними на вході моделі. Для простоти моделювання процесу відновлення передбачається, що інжекція несправностей генерується заново перед подачею кожного навчального міні-пакету даних. Діагностичні дані вибираються з додаткового набору даних, кількість яких дорівнює розміру 128 зображень. Процес відновлення припиняється, коли функція втрат на діагностичних даних не зменшується протягом послідовних 5 ітерацій, або коли різниця між минулим та поточним значеннями функції втрат стає меншою за β=0,001.

Для тестування моделі на стійкість до шуму та протиборчих атак пропонується не покладатися на специфічні особливості архітектури моделі та алгоритму навчання, такі як градієнти. Замість цього тестування буде базуватися на атаках з використанням чорного ящика. При цьому розглядатимуться два типи атак, які дають найбільш розбіжні результати – "сильні" атаки на один/декілька пікселів та "слабкі" атаки на всі пікселі. Формування обох атак буде реалізовано на основі алгоритму пошуку стратегії еволюції адаптації коваріаційної матриці (CMA-ES) [107, 108]. Для першого типу атак обмеження на амплітуду збурення (th) задається L0-нормою, а для другого – L∞-нормою.

Появу нового класу можна розглядати як один з підвидів дрейфу концепцій, оскільки при цьому повинні змінюватися межі між класами. Для перевірки здатності адаптуватися до появи нових класів пропонується спочатку навчити модель на неповній множині класів, а потім подати марковані вибірки

класів, які були видалені раніше, для доналаштування навченої моделі. Тестування здатності адаптації до дрейфу концепцій пропонується здійснювати шляхом подачі вибірки класів, мітки яких поміняні місцями, на доналагодження навченої моделі. Успішною адаптацією вважається відновлення продуктивності, тобто досягнення не менше 95% від продуктивності на незбурених даних. Адаптація припиняється, якщо точність не покращується протягом послідовних 10 кроків (міні-пакетів). Передбачається, що реальний дрейф концепцій буде виявлятися автоматично при додаванні до навчального міні-пакету мічених вибірок з бази даних зі зміненими мітками, які потім включатимуться в чергу останніх мічених даних. Поріг виявлення реального дрейфу понять встановлено на рівні 50 зразків одного класу, що хибно розпізнаються як інший клас.

На рис. 2.4 показано запропоновану схему емпіричного тестування системи класифікаційного аналізу зображень на резільєнтність до несправностей, протиборчих атак та реального дрейфу концепцій.

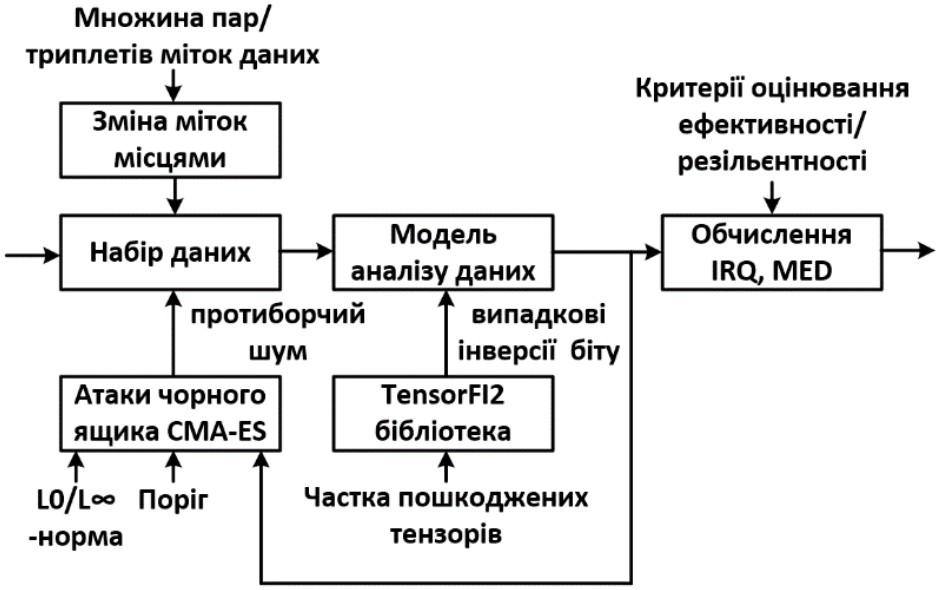


Рисунок 2.4 – Схема тестування системи аналізу даних на резільєнтність

У представленій на рис. 2.4 схемі тестування модель аналізу даних розглядається як чорна шухляда, тобто використовується лише інформація про входи і виходи моделі, ігнорується інформація про градієнти на проміжних

шарах моделі. Тестування відбувається шляхом формування збурення і спостереження за результатом поглинання збурення для оцінювання робастності та адаптації до збурення для оцінювання швидкості відновлення ефективності. Однак для емпіричне тестування більш придатне для порівняльного аналізу, як допоміжний інструмент у оцінюванні резільєтності інтелектуальної системи. Емпіричне тестування не дає якихось точних гарантій, для підвищення його ефективності існує необхідність у забезпеченні тестового покриття. Емпіричне тестування не дає чіткої кількісної оцінки, наскільки ми можемо бути впевнені, що бажана властивість є істинною після тестування.

Для надання суворих гарантій результатів тестування використовуються методи формальної верифікації. У працях, що пов’язані з формальною верифікацією глибоких нейронних мереж, розглядаються різні варіанти кодування моделі у форму, що зручна для розв’язувача, і реалізується у відповідності до обраної теорії. Найбільшого поширення набули такі підходи щодо формальної верифікації: підходи, що основані на вирішувачі обмежень (constraint solvers), де нейромережа кодується у вигляді множини обмежень [70]; підходи, що основані на обчисленні наближення границь (approximate bound), де операції наближення застосовуються до простору входів, виходів чи функції нейронних шарів з метою спрощення пошуку гарантованих меж [107]; підходи, основані на обчисленні збіжних границь (converging bounds), де реалізується пошук та ітеративне уточнення гарантованих меж [70]. Дані підходи розраховані на забезпечення якісної верифікації (qualitative verification), тобто розраховані на детерміновані результати і характеризуються високою обчислювальною складністю. Однак стохастичний характер навчання, поява даних з невідомого розподілу в режимі екзамену (inference), розвиток ймовірнісних нейронних мереж (Probabilistic Neural Network*)* та рандомізованих архітектур моделей звужують можливості та ефективність якісної верифікації. Тому розробляються ймовірнісні (статистичні) методи верифікації та сертифікації, що є найбільш загальними і обчислювально ефективними. Прикладом є метод оцінювання сталої Ліпшиця за допомогою теорії екстремальних значень, однак подібні

методи мають проблему надійності. У переважній більшості праць розглядається сертифікація робастності до збурень. При цьому приділяється недостатня увага поведінці моделі в режимі відновлення продуктивності. Швидкість відновлення після збурення досі не є об'єктом верифікації. Тому про пропонується метод ймовірнісної верифікації (сертифікації) резільєнтності зі зменшеними вимогами до кількості тестів.

На рис. .2.5 показано блок-схему обчислювально ефективного алгоритму ймовірнісної сертифікації резільєнтності класифікатора зображень до протиборчих атак та ушкодження тензорів до інжекції несправностей в пам'яті, шуму протиборчих атак та дрейфу концепцій, що дозволяє працювати з інтелектуальним алгоритмом як з чорним ящиком та зменшити кількість тестів, необхідних для задоволення заданих довірчих меж. Резільєнтність до кожної вибіркової реалізації збурюючого впливу оцінюється на по кривій резільєнтності, що будується протягом наперед заданого інтервалу T, і обчислюється за формулою (2.1.4). Так само до початку тестування потрібно задати розмір міні-пакету даних.

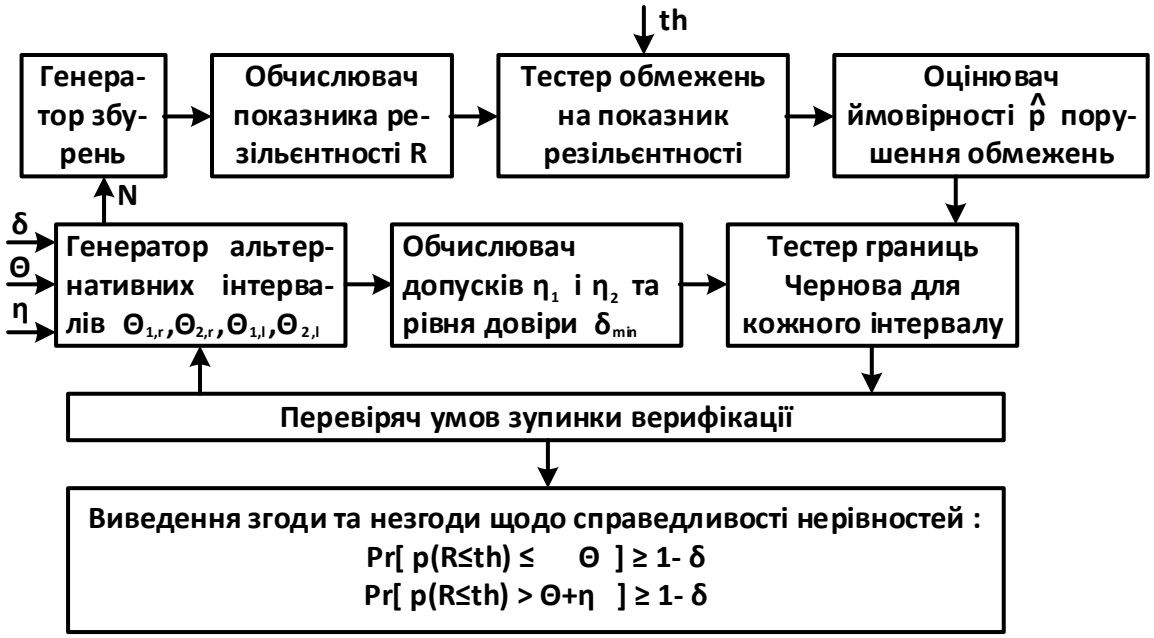


Рисунок 2.5 – Функціональна схема реалізації методу сертифікації резільєнтності інтелектуальної системи

Сертифікації підлягає четвірка обраних параметрів th, θ, η та δ, де th – задане значення порогу на мінімально допустиме значення показника резільєнтності, θ – задане значення порогу на ймовірність прояву недостатньої резільєнтності, η – задане значення допустимої помилки оцінювання ймовірності, δ – рівень статистичної значущості, що, як правило, обирається з множини {0,1; 0,05, 0,01; 0,001}.

Згода алгоритму сертифікації полягатиме у перевірці справедливості двох нерівностей:

$$Pr\,[p(R \le th) \le \theta] \ge 1 - \delta; \tag{2.3.1}$$

$$\Pr[p(R \le th) > \theta + \eta] \ge 1 - \delta, \tag{2.3.2}$$

де Pr – рівень довіри до оцінки ймовірності успіху та неуспіху під час тестування алгоритму на прояв недостатньої резільєнтності до деструктивного збурення;

R – інтегральний показник резільєнтності інтелектуального алгоритму;

$p(R \le th)$ – оцінка ймовірності прояву недостатньої резільєнтності.

Алгоритм верифікації резільєнтності оснований на лемі про границю Чернова [7]. Згідно даної леми для перевірки справедливості нерівностей (2.3.1) та (2.3.2) не обхідно виконати N тестів, де

$$N = \frac{12}{\eta^2} ln \frac{1}{\delta}. \tag{2.3.3}$$

У праці [109] для зменшення кількості тестів під час перевірки границі Чернова у випадку значної різниці між реальною ймовірністю p(R <= th) та пороговим значенням θ розглядається можливість зниження кількості тестів шляхом перевірки серії альтернативних гіпотез. Замість перевірки нерівностей

(2.3.1) та (2.3.2) пропонується перевіряти серію альтернативних нерівностей, перевірка яких вимагає меншу кількість тестів для раннього прийняття рішення

$$Pr\ [p(R \leq th) \leq \theta_1] \geq 1 - \delta_{min}; \tag{2.3.4}$$

$$\Pr[p(R \leq th) > \theta_2] \geq 1 - \delta_{min}, \tag{2.3.5}$$

де θ_1 та θ_2 межі альтернативного інтервалу замість інтервалу $[\theta, \theta + \eta]$;

δ_{min} – рівень статистичної значущості для одного з n альтернативних інтервалів, причому

$$\delta_{min} = \frac{\delta}{n}. \tag{2.3.6}$$

Загальна кількість альтернативних інтервалів включає в себе максимальну кількість інтервалі зліва n_l від θ, максимальну кількість інтервалів справа n_r від $\theta + \eta$, і якщо не відбудеться прийняття рішення на альтернативних інтервалах, тестування відбудеться і для інтервалу $[\theta,\ \theta + \eta]$:

$$n_l \leq 1 + log\frac{\theta}{\eta}, \tag{2.3.7}$$

$$n_r \leq 1 + log\frac{1-\theta-\eta}{\eta}, \tag{2.3.8}$$

$$n = 3 + \max\left(0, log_2\left(\frac{\theta}{\eta}\right)\right) + \max\left(0, log_2\left(\frac{1-\theta-\eta}{\eta}\right)\right). \tag{2.3.9}$$

Інтервали, що обираються зліва від θ, можна назвати підтвержуючими, оскільки підтвердження нерівностей (2.3.4) та (2.3.5) на будь-якому з цих інтервалів припиняє виконання алгоритму з позитивним результатом. Інтервали справа від $\theta + \eta$ можна назвати спростовуючими, оскільки негативний результат перевірки нерівностей (2.3.4) та (2.3.5) на будь-якому з цих інтервалів припиняє

виконання алгоритму з негативним результатом. При цьому для прискорення алгоритму пропонується формування інтервалів робити від максимального їх розміру до мінімального, де кожен наступний інтервал формується поділом навпіл ширини попереднього інтервалу (рис. 2.6).

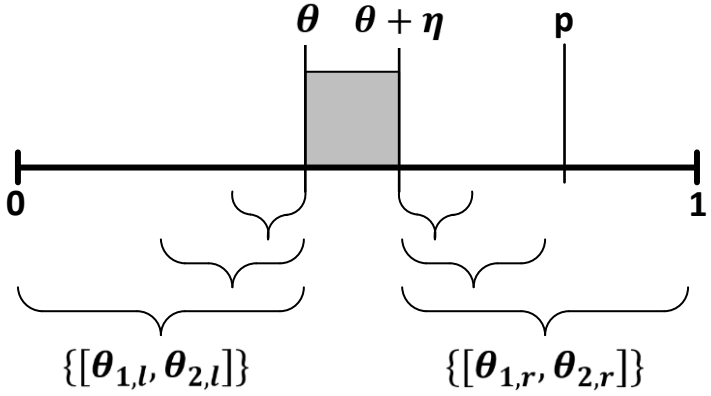


Рисунок 2.6 – Альтернативні інтервали для тестування границі Чернова

Для тестування кожного альтернативного інтервалу необхідно N тестів

$$N = \frac{\left(\sqrt{3\theta_1}+\sqrt{2\theta_2}\right)^2}{(\theta_2-\theta_2)^2} ln \frac{1}{\delta_{min}}, \qquad (2.3.10)$$

де η_1, η_2 – допустимі помилки оцінювання ймовірності $p(R \leq th)$ на вибірці тестів обмеженого обсягу N зліва та справа, що обчислюються за формулами

$$\eta_1 = (\theta_2 - \theta_1)\left(1 + \sqrt{\frac{2}{3}\frac{\theta_2}{\theta_1}}\right)^{-1}, \qquad (2.3.11)$$

$$\eta_2 = \theta_2 - \theta_1 - \eta_1. \qquad (2.3.12)$$

Тест на альтернативному інтервалі вважаться успішним якщо оцінка $\hat{p}$ ймовірності $p(R \leq th)$ менша або рівна за $\theta_1 + \eta_1$. Тест на альтернативному інтервалі вважаться неуспішним якщо оцінка $\hat{p}$ ймовірності $p(R \leq th)$ більша за $\theta_2 - \eta_2$. Якщо жодна з цих умов не виконується, то це свідчить про необхідність

продовження генерації нових інтервалів і продовження процесу тестування. Проте процес може зупинитися також якщо час спливе допустимий час або ресурс, що виділено на тестування.

2.4 Принципи побудови моделі та алгоритму навчання резільєнного класифікатора зображень

Найбільш вразливими до деструктивних збурень є системи класифікаційного аналізу зображень. Тому в першу чергу варто розглянути принципи забезпечення резільєнтності таких системи, тим більше, що більшість сформульованих принципів забезпечуватимуть резільєнтність інших типів систем інтелектуального аналізу даних. Принципи забезпечення резільєнтності систем інтелектуального аналізу можна поділити на принципи, що стосуються архітектури моделі, та принципи, що стосуються алгоритму навчання.

Під час побудови моделі аналізу даних необхідно намагатися створити основу для реалізації таких механізмів резільєнтності як робастність, витончена деградація, відновлення ефективності та удосконалення. Синтез моделей класифікаційного аналізу пропонується здійснювати опираючись на такі основні принципи [2, 3]:

– ієрархічна класифікація і відповідно ієрархічне маркування даних для реалізації механізму витонченої деградації шляхом огрублення прогнозу більш абстрактним класом з достатньою впевненістю, коли класи внизу ієрархії розпізнаються з низьким рівнем впевненості;

– поєднання механізмів самодистиляції знань та вкладеного навчання для підвищення робастності моделі за рахунок збільшення інформативності зворотного зв'язку для нижніх шарів на етапі навчання та прискорення режиму екзамену за рахунок пропуску високорівневих шарів для простих зразків даних;

– формування прототипу та компактного сферичного контейнера для кожного класу для спрощення виявлення зразків, що виходять за межі розподілу, та дрейфу концепцій;

– використання в пам'яті FIFO-черги обмеженого розміру для зберігання маркованих та немаркованих даних з відповідними значеннями функції втрат, отриманими в результаті виведення, для реалізації механізму діагностики та відновлення.

Ці принципи повинні забезпечити ресурсоефективність, оскільки модель матиме гілки для проміжних рішень, що вносить мінімальну надлишковість, оскільки основна частина тіла екстрактора ознак розподіляється між проміжними класифікаторами. Крім того, розмір черг даних для діагностичних вибірок з відповідними значеннями функції втрат, або маркованих та немаркованих вибірок з результатами їх розпізнавання може бути встановлені на прийнятну з точки зору ресурсних обмежень ємність.

На рис. 2.7 показано архітектуру резільєнтного класифікатора зображень, що реалізує запропоновані принципи [3]. Дана схема ілюструє секційну структуру глибокого нейронного класифікатора. Секції складаються з ResBlocks відомої архітектури ResNet50. Архітектура ResNet50 також послужила натхненням для модуля Bottleneck, який служить для пом'якшення впливу між картами ознак верхнього і нижнього рівня в рамках механізм самодистиляції знань.

Для класифікаційного аналізу ознакового подання на виході кожної секції будується набір векторів-прототипів. Вектори-прототипи не є фіксованими, вони налаштовуються в процесі навчання разом з вагами екстрактора ознак. Для реалізації принципу витонченої деградації прототипи класифікатора можуть належати до різних рівнів в ієрархії відповідно до ієрархії маркування. Для підвищення завадостійкості та реалізації інформаційного пляшкового горла ознакове подання стискається (апроксимується) до дискретного вигляду, для чого на виході екстрактора ознак кожної ділянки використовується сигмоїдний (Sigmoid) шар та відповідна регуляризація в алгоритмі навчання.

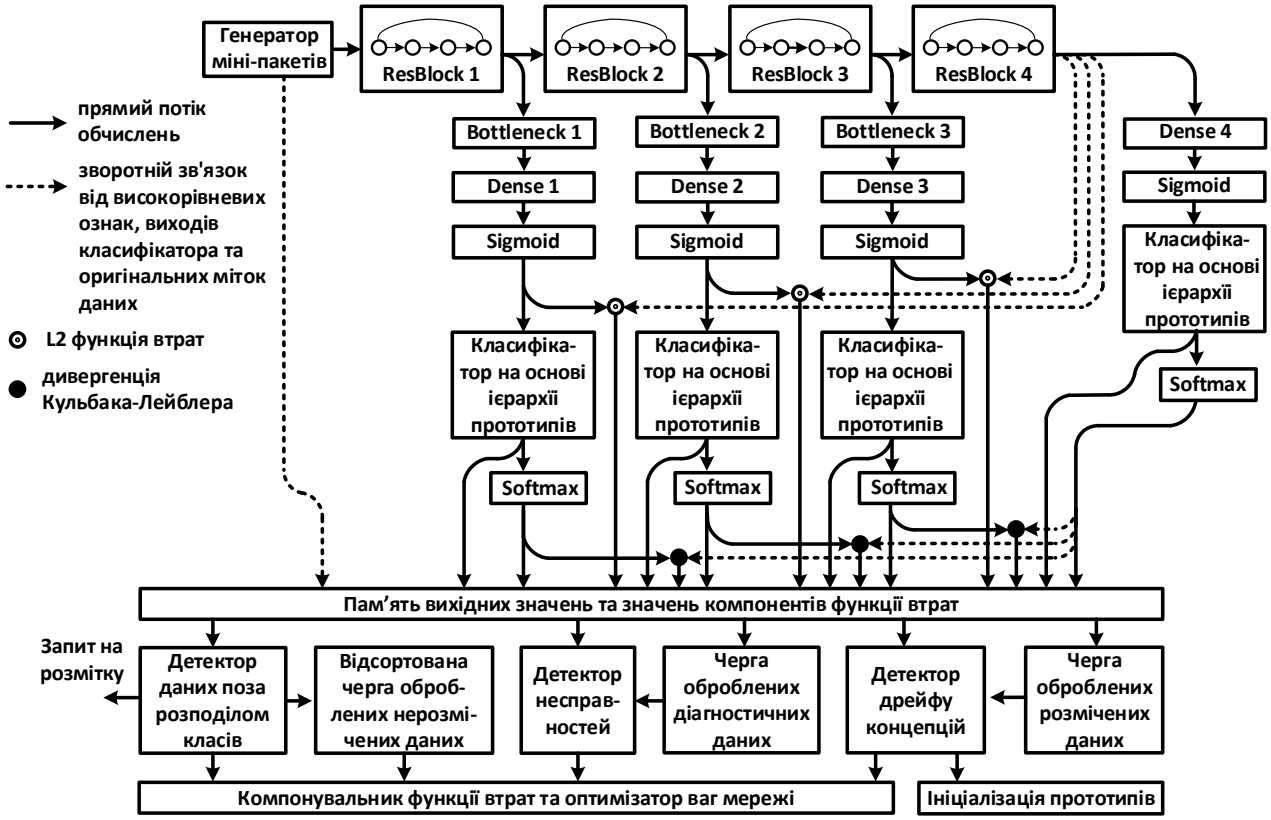


Рисунок 2.7 – Архітектура моделі резільєнтного класифікатора зображень

Радіус гіперсферичних контейнерів класів оптимізуються для кожного прототипу класифікатора. Радіуси контейнерів зберігаються в пам'яті для виявлення високих рівнів невизначеності при прийнятті рішень. Тестові зразки поза контейнерами класів стають кандидатами для напівконтрольованого настроювання та для ручного маркування (активного навчання), яке буде виконано на більш пізньому етапі. Контроль для зразків поза контейнером класу також може бути використаний для виявлення реального дрейфу концепції і виявлення новизни в даних, тобто даних поза навчальним розподілом.

Після оновлення ваг і параметрів моделі діагностичний набір даних і відповідне значення функції втрат повинні бути збережені (або оновлені) в пам'яті. Після цього підмножина діагностичних даних повинна передаватися для прогнозування разом з тестовими зразками в кожному міні-пакеті даних. Це дозволить порівняти минулі та поточні значення функції втрат для виявлення несправностей у пам'яті ваг нейронної мережі. Якщо різниця між минулим та поточним значенням функції втрат перевищує певний поріг α=0,01, необхідно

ініціювати алгоритм точної настройки нейронної мережі з використанням діагностичних даних для приведення цієї різниці до порогу β=0,001.

Багатосекційна структура моделі з проміжними класифікаторами дозволяє реалізувати адаптивні обчислення, що дозволяє прискорити розпізнавання простих образів. При цьому, в міру навчання моделі, вона стає швидшою за рахунок підвищення достовірності прогнозування класифікаторів нижніх секцій. Це, в свою чергу, дозволить передчасно пропускати решту високорівневих секцій моделі. Пропонуються наступні правила проведення класифікаційного аналізу в рамках адаптивних обчислень [1, 2, 3]:

– обчислення нейромережі виконуються послідовно, секція за секцією;

– секції високого рівня можуть бути пропущені, якщо на виході поточної секції максимальне значення функції належності до певного класу нижчого ієрархічного рівня перевищує довірчий поріг T;

– якщо максимальне значення функції належності жодного з ієрархічних рівнів класифікатора на виході поточної секції не збільшилося порівняно з попередньою секцією, то подальші розрахунки можна пропустити;

– якщо виконується будь-яка з умов пропуску наступних секцій або розглянутий класифікатор є останнім класифікатором в моделі і максимальне значення функції належності нижнього ієрархічного рівня не перевищує довірчого порогу, то перевіряється вищий рівень в ієрархії;

– якщо клас з достатнім рівнем довірчої ймовірності не виявлено, відбувається відмова прийняття рішення, формується запит на ручне маркування, а відповідна вибірка позначається як придатна для напівконтрольованого налаштування.

Модуль ієрархічного класифікатора на основі прототипів складається з прототипів класів, параметрів гіперсферичного контейнера та параметра регуляризації, призначеного для стиснення (дискретизації) представлення ознак та прототипів. При цьому достовірність прогнозу належності i-го зразка до k-го класу визначається наступною функцією належності [3]

$$\mu_k(z_i) = 1 - \frac{\mathrm{dist}(z_i, \bar{z}_k)}{N \cdot r_k}, \qquad (2.4.1)$$

де z_i – бінарне представлення ознак і -го прикладу на виході екстрактора ознак;

$\bar{z}_k$ – прототип k-го класу, що навчається;

N – розмірність вхідного вектора ознак z_i;

r_k – навчальний масштабний коефіцієнт для радіуса гіперсферичної границі рішення (контейнера) k-го класу, $r_k \in (0; 1)$;

dist(·) – евклідова квадратна відстань.

Якщо максимальне значення функції (2.4.1) для вхідного немаркованого зразка z_i менше нуля, то такому прогнозу не слід довіряти і таку вибірку слід додати до черги немаркованих даних, що відповідає зразкам поза навчальним розподілом. Якщо вхідна немаркована вибірка потрапляє в один з контейнерів класів розпізнавання (на будь-якому з рівнів), то вона повинна бути додана в буфер немаркованих даних, що відповідає зразкам всередині навчального розподілу. Буфери немаркованих вибірок можуть бути використані для навчання з псевдомаркуванням, м'яким маркуванням або для регуляризації узгодженості.

У випадку, коли модель навчена, але в черзі нових маркованих даних виявлено появу n зразків c-го класу, що потрапляють до контейнера іншого k-го класу, система розпізнає реальний дрейф концепцій.

Для уникнення катастрофічного забування в умовах дрейфу концепцій або появи нового класу розпізнавання неявно реалізована функція нагадування. Така функція базується на немаркованих чергах даних та векторах-прототипах у просторі ознак, які повільно змінюються. Цій же меті слугує і механізм дистиляції знань верхніх шарів.

Дані з немаркованої черги даних можуть бути переміщені до маркованої черги даних після отримання зворотного зв'язку про їх фактичну належність до класів. Пріоритетність конкретних зразків, рекомендованих для ручного маркування, залежить від значення функції належності (2.4.1).

Під час розроблення алгоритму навчання ми так само ставимо за мету реалізацію механізмів і властивостей робастності, виточченої деградації, відновлення та вдосконалення. З цією метою алгоритм навчання буде базуватися на наступних принципах [2, 3]:

– врахування ієрархії маркування даних та ієрархії прототипів класів шляхом обчислення функції втрат окремо для кожного рівня ієрархії для забезпечення витонченої деградації в режимі екзамену;

– реалізація самодистиляції знань, тобто перегонки знань з високорівневого шару (секції) моделі на більш низькі шари (секції) моделі у вигляді додаткових компонентів регуляризації для підвищення робастності та забезпечення адаптивних розрахунків в режимі екзамену;

– збільшення компактності розподілу класів та буферної зони між класами для підвищення стійкості до шуму, викидів та протиборчих атак за рахунок додаткових компонентів регуляризації, що враховує відстані між зразками і прототипами в просторі ознак;

– дискретизація ознакового подання з метою реалізації інформаційного пляшкового горла та підвищення робастності ознакового подання шляхом введення відповідного компоненту регуляризації;

– можливість ефективного використання як мічених, так і немічених вибірок даних для прискорення адаптації при обмеженій кількості мічених даних, які зазвичай надходять з часовим лагом;

– уникнення катастрофічного забування під час адаптації до змін та протиборчих атак без повного перенавчання за рахунок реалізації механізму нагадування з використанням буферів даних та дистиляційного зворотного зв'язку верхніх шарів.

Запропонований метод навчання складається з наступних етапів:

– попереднє самонавчання моделі з використанням екземпляр-прототипної контрастної функції втрат L_{ICPL} [77, 78];

– ініціалізація прототипу та радіусу контейнера для кожного класу;

– навчання з учителем з функцією втрат L_S, яка включає звичайну кросентропію та додаткові компоненти для самодистиляції знань і регуляризації;

– вибір даних для діагностики несправностей (обираються випадковим чином або з урахуванням значення функції втрат);

– екзамен (inference) на нових та діагностичних даних;

– запит на ручне маркування складних зразків даних;

– навчання з учителем на основі функції втрат L_S з урахуванням результатів запиту на ручне маркування або точне настроювання з частковим залученням учителя з додатковим компонентом L_{SSIN} або L_{SSOUT} в залежності від результату екзамену;

– оновлення діагностичних даних.

За наявності великого обсягу немаркованих даних підготовку моделі слід починати з самонавчання екстрактора ознак. Для цього пропонується використовувати екземпляр-прототипну контрастну функцію втрат, що характеризується обчислювальною ефективністю та узагальнюючою здатністю, близькою до біологічного прототипу [77, 3]. Екземпляр-прототипна контрастна функція втрат розраховується за формулою

$$L_{ICPL} = -\log \frac{\exp\left(-\mathrm{dist}(z_i, \bar{z}_i)/\tau\right)}{\exp(-\mathrm{dist}(z_i, \bar{z}_i)/\tau) + \sum_{b=1}^{B} \exp\left(-\mathrm{dist}(z_i, z_b)/\tau\right)}, \quad (2.4.2)$$

де z_i – представлення ознак на виході екстрактора ознак $z_i = f(x_i)$ для вхідного зразка x_i з міні-пакету;

$\bar{z}_i$ – усереднене представлення ознак для аугментованої версії вхідного прикладу і розглядається як позитивна пара для вхідного прикладу x_i,

$$\bar{z}_i = \frac{1}{n_a} \sum_{j=1}^{n_a} f(a_j(x_i)), \quad (2.4.3)$$

де a_j – оператор аугментації, наприклад, випадкове обрізання, зміна масштабу, випадкове перевертання по горизонталі/вертикалі, випадкове коригування

відтінку, насиченості, контрастності та яскравості, випадкове перетворення в градації сірого;

B – кількість попередньо оброблених прикладів, що розглядаються як від'ємні пари, ознакові подання яких зберігаються в неіндексованій черзі FIFO (з довжиною черги B=1024) і оновлюються після обробки кожного міні-пакету;

τ – температурний параметр, який контролює динамічний діапазон функції подібності.

Функція втрат L_S, що використовується у навчанні з учителем включає окрім звичайної перехресної ентропії L_{SCE}, обчисленої за допомогою міток і прогнозів, компоненти самодистиляції знань на рівні класів L_{CSD} та на рівні ознак L_{FSD}. До функції втрат також може бути додана компонента регуляризації L_D для досягнення стиснення ознакового подання, реалізуючи принцип інформаційного пляшкового горла. Контрасно-центрована функція втрат L_{CCL} може бути використана як компонент регуляризації для покращення оптимізації робастності за рахунок збільшення внутрішньокласової компактності та міжкласової відокремлюваності. Таким чином, для навчання з учителем пропонується наступна комбінована функція втрат

$$L_S = \lambda_{SCE}\bar{L}_{SCE} + \lambda_{CCL}\bar{L}_{CCL} + \lambda_{CSD}\bar{L}_{CSD} + \lambda_{FSD}\bar{L}_{FSD} + \lambda_D\bar{L}_D, \qquad (2.4.4)$$

де $\bar{L}_{SCE}$, $\bar{L}_{CCL}$ – значення відповідних функцій втрат L_{SCE} та L_{CCL}, усереднені за S-секціями та H-рівнями ієрархії класів класифікаційної моделі;

$\bar{L}_{FSD}$, $\bar{L}_{CSD}$ – значення відповідних функцій втрат L_{FSD} та L_{CSD}, усереднені по (S-1)-секціях та H-рівнях ієрархії класів класифікаційної моделі;

$\bar{L}_D$ – усереднене за S-виходами (секціями) моделі класифікації, включаючи вихід останнього шару, значення функції втрат L_D;

λ_{SCE}, λ_{CCL}, λ_{FSD}, λ_{CSD}, λ_D – коефіцієнти регулювання впливу складових результуючої функції втрат.

Модуль класифікатора пропонується будувати на основі прототипів класів, які є центрами гіперсферичних контейнерів класів. При цьому на виході класифікатора розраховується крос-ентропійна функція втрата L_{SCE} за формулою

$$L_{SCE} = CE(q(z_i, \tau = 1), y_i) \quad (2.4.5)$$

де CE(·) – кросентропійна функція втрат;

y_i – вектор цільової змінної з унітарним кодуванням (one-hot encoded) для і-го вхідного зразка даних;

q(·) – оцінка ймовірності належності ознакового подання для і-го вхідного зразка до контейнера кожного з класів, яка для k-го класу обчислюється за формулою

$$q_k(z_i, \tau) = \text{softmax}(\text{leaky\_relu}(\mu(z_i)/\tau))_k = \frac{\exp(\text{leaky\_relu}(\mu_k(z_i)/\tau))}{\sum_{c=1}^{K} \exp(\text{leaky\_relu}(\mu_c(z_i)/\tau))}, \quad (2.4.6)$$

де K – розмір множини класів;

leaky_relu – покращена версія функції ReLU з малим нахилом (alpha =0,02) для від'ємних значень.

Контрастно-центрова функція втрат L_{CCL}, яка впливає на ефективність оптимізації кожного прототипу класу, розраховується на маркованій навчальній множині за формулою [2, 3]

$$L_{CCL} = \frac{\text{dist}(z_i, \bar{z}_{y_i})}{\sum_{k=1,\, k\neq y_i}^{K} \text{dist}(z_i, \bar{z}_k)+1} \quad (2.4.7)$$

Компонент самодистиляції знань на рівні ознак L_{FSD} у вигляді L_2 функції та на рівні класів L_{CSD} у вигляді дивергенції Кульбака-Лейблера розраховуються між S-м (останнім) виходом моделі та s-м (проміжним) виходом моделі:

$$L_{FSD} = \mathrm{dist}(z_i^s,\ z_i^S), \tag{2.4.8}$$

$$L_{CSD} = \mathrm{KL}(q(z_i^s, \tau),\ q(z_i^S, \tau)), \tag{2.4.9}$$

де z_i^s, z_i^S – ознакові подання на виході s-ї секції (проміжний вихід) моделі та на виході S (останній вихід) моделі.

Компонент регуляризації L_D реалізує інформаційне пляшкове горло шляхом штрафу за похибку дискретизації подання ознак

$$L_D = z_i{}^T(e - z_i), \tag{2.4.10}$$

де e – одиничний вектор.

Для прискорення адаптації до змін в регуляризації узгодженості можуть бути використані немарковані приклади даних [99, 100]. У цьому випадку немарковані дані поділяються на дві групи: немарковані приклади, які потрапляють у контейнери класів та немарковані приклади, які знаходяться поза межами всіх контейнерів класів.

Немарковані дані, які потрапляють у контейнери класів, використовуються для обчислення компоненти регуляризації L_{SSIN} за наступною формулою

$$L_{SSIN} = \mathrm{CE}\big(q(z_i', \tau = 1),\ q(z_i'', \tau = 1)\big) \tag{2.4.11}$$

де z_i', z_i'' – ознакові подання двох аугментованих версій вхідного зразка x_i.

Певна частина γ ($\leq$10%) немаркованих даних, які потрапляють у контейнери класів і мають максимальні значення $q(z_i)$, можуть бути

псевдомарковані відповідними класами. Такі псевдомічені дані можуть бути включені в кожен міні-пакет під час навчання.

Немарковані зразки даних, які випадають за межі всіх контейнерів класів, можуть бути реалізаціями невідомих класів або результатом дрейфу концепцій. У цьому випадку в компоненті регуляризації узгодженості L_{SSOUT} слід використовувати м'яке маркування $q_k^{dist}(z_i)$ на основі відстаней до прототипу відомих класів:

$$L_{SSOUT} = CE\big(q(z_i, \tau = 1), q^{dist}(z_i)\big), \tag{2.4.12}$$

$$q_k(z_i) = \text{softmax}(-\text{dist}\,(z_i, \bar{z}\,))_k = \frac{\exp\,(-\text{dist}(z_i,\ \bar{z}_k))}{\sum_{c=1}^{K} \exp\,(-\text{dist}(z_i, \bar{z}_c))}. \tag{2.4.13}$$

Початкові значення параметрів прототипів класів нижнього рівня ініціалізуються на основі матриці Хадамара [103] з використанням принципу згладжування міток (label smoothing). Для цього спочатку визначається розмірність матриці Хадамара $N_{Hadamard} = 2^{\text{ceil}(\log_2(N))}$, де ceil() – функція округлення числа до більшого цілого значення. Всі значення, менші за 0, замінюються на 0, тобто $Z = \max\big(0,\ \text{Hadamard}(N_{Hadamard})\big)$.

Для реалізації принципу зглажування міток значення коодинат ініціалізованих прототипів класів модифікуються за формулою $Z' = Z \cdot 0{,}7 + 0{,}15$, в результаті чого одиниці перетворяться на 0,85, а нулі на 0,15. Потім з отриманої матриці вибираються K перших векторів, усічених за N першими ознаками, тобто $\bar{z} = Z'[1{:}\,K,\ 1{:}\,N]$. Масштабуючий коефіцієнт r_k для радіуса гіперсферичної границі рішень (контейнера) k-го класу ініціалізується значенням половини границі Плоткіна, поділеної на розмірність простору ознак [76, 103]:

$$r_k \leftarrow \left(\frac{1}{2}\frac{N}{2}\frac{K}{K-1}\right)\frac{1}{N} = \frac{K}{4(K-1)}. \tag{2.4.14}$$

Поява зразка з міткою, що вказує на новий (К + 1)-й клас нижчого рівня, зумовлює необхідність формування нового прототипу для класу Z_{K+1} з відповідними початковими значеннями масштабуючого коефіцієнта r_{K+1}. Це досягається шляхом вибору найближчого вектора з решти невикористаних рядків модифікованої матриці Хадамара Z', де близькість визначається на основі евклідової квадратичної відстані. Початкове значення масштабуючого коефіцієнта радіусу контейнера для нового класу також визначається за формулою (2.4.14), але з урахуванням нової кількості класів.

Кожна координата прототипу верхнього ієрархічного рівня ініціалізується копіюванням відповідної координати одного з прототипів нижнього рівня, обраного випадковим чином. Початковий радіус класу верхнього ієрархічного рівня визначається за формулою (2.4.14) з урахуванням кількості класів на цьому рівні.

При розпізнаванні реального дрейфу понять прототипи дрейфуючих класів ініціалізуються випадковими числами з діапазону [0; 1].

Діагностичні набори даних, сформовані шляхом вибірки маркованих прикладів, які потрапляють у свої контейнери класів за результатами екзамену. Значення функції втрат (2.4.4), розраховане для діагностичних даних, зберігається в пам'яті для порівняння. При виявленні розбіжності між поточним значенням і раніше розрахованим значенням функції втрат ініціюється точне настроювання. Точне настроювання припиняється, якщо різниця зменшується більш ніж в 10 разів.

2.5 Аналіз впливу параметрів і архітектурних рішень на резільєтність інтелектуального класифікатора зображень

Для проведення експериментів було обрано набори даних CIFAR-10 та CIFAR-100, оскільки вони є загальнодоступними, а їх зображення мають невеликий розмір, що прискорює проведення експериментальних досліджень [104, 105]. Класи набору даних CIFAR-10 можна розташувати в ієрархічній структурі. Наприклад, першим суперкласом (класом верхнього

рівня) буде клас тварин, який включає підкласи “bird”, “cat”, “deer”, “dog”, “frog” та “horse”. Другим суперкласом буде клас транспортних засобів, який включає підкласи літак, автомобіль, судно та вантажівка. Набір даних CIFAR-10 складається з 50 000 навчальних зображень та 10 000 тестових зображень розміром 32x32, рівномірно розподілених між 10 класами. 100 класів в CIFAR-100 згруповані в 20 суперкласів на верхньому рівні [105]. CIFAR-100 складається з тієї ж кількості зображень, що і CIFAR-10, але має 500 навчальних зображень на клас і 100 тестових зображень на клас. Для навчання базової моделі будемо використовувати 70% навчальних даних, а решту 30% використовуватимемо для формування додаткового навчального набору даних.

У випадку набору даних CIFAR-10 на виході класифікатора кожної секції моделі буде використано 12 векторів прототипів, з яких 2 для прототипів суперкласу та 10 для прототипів нижчого рівня. Для набору даних CIFAR-100 на виході класифікатора кожної секції буде використано 120 векторів прототипів відповідно.

Для всіх експериментів обраний довірчий поріг, який вважається достатнім для прийняття рішення і становить T=0,8. Навчання проводилося на основі оптимізатора Adam зі швидкістю навчання 0,0003. Розмірність ознакового подання становить N=64. Пропонуються наступні значення коефіцієнтів, що використовуються у функції втрат (2.4.4), за замовчуванням: $\lambda_{\mathrm{SCE}} = 1{,}0$, $\lambda_{\mathrm{CCL}} = 1{,}0$, $\lambda_{\mathrm{CSD}} = 0{,}1$, $\lambda_{\mathrm{FSD}} = 0{,}01$, $\lambda_{\mathrm{D}} = 0{,}0001$. Крім того, використовуються компоненти напівнагляду λ_{SSIN} та λ_{SSOUT} з коефіцієнтами $\lambda_{\mathrm{SSIN}} = 0.1$ та $\lambda_{\mathrm{SSOUT}} = 0{,}1$ відповідно. Попереднє навчання з функцією втрат (2.4.2) пропонується проводити на всіх навчальних даних без урахування міток. Розмір міні-пакету пропонується встановити рівним 128 зображень.

У табл. 2.1 наведено залежність точності моделей (Acc) до та після відновлення від частки несправностей для рівня класів та суперкласів незалежно від того, вихід якої секції моделі було обрано як кінцевий [3]. У таблиці також наведено кількість кроків (Rec_steps), виконаних для відновлення в кожному

випадку. У всіх експериментах наявність несправностей виявляється зі 100% точністю.

Таблиця 2.1 – Порівняння точності в умовах впливу збурень та необхідної кількості кроків відновлення для різних наборів даних на різних ієрархічних рівнях класифікатора

Набір даних	Частка пошкоджених тензорів	MED (Acc) під впливом інжекції несправностей	MED (Acc) після відновлення	MED (Acc) для рівня супер-класів під впливом інжекції несправностей	MED (Acc) для рівня супер-класів після відновлення	MED (Rec_steps)	MED (Rec_steps) для рівня супер-класів
CIFAR-10	0.0	0,993	-	0,980	-	-	-
CIFAR-10	0,1	0,985	0,991	0,975	0,979	12	12
CIFAR-10	0,2	0,932	0,976	0,930	0,961	29	21
CIFAR-10	0,3	0,852	0,971	0,870	0,932	32	32
CIFAR-10	0,4	0,801	0,921	0,790	0,914	49	54
CIFAR-10	0,5	0,713	0,882	0,730	0,893	63	71
CIFAR-10	0,6	0,532	0,851	0,721	0,881	81	80
CIFAR-100	0,0	0,890	-	0,970	-	-	-
CIFAR-100	0,1	0,879	0,889	0,962	0,970	35	25
CIFAR-100	0,2	0,871	0,881	0,961	0,970	55	51
CIFAR-100	0,3	0,790	0,880	0,926	0,961	59	50
CIFAR-100	0,4	0,600	0,870	0,910	0,958	62	64
CIFAR-100	0,5	0,551	0,851	0,890	0,929	70	71
CIFAR-100	0,6	0,357	0,758	0,665	0,910	80	77

Аналіз табл. 2.1 показує, що класифікатор більш високого ієрархічного рівня краще поглинає несправності і краще відновлюється на діагностичних даних, хоча класифікатор більш високого ієрархічного рівня несе менше інформації. Це є корисною властивістю для механізму витонченої деградації. В той же час, при наявності одиничної випадкової інверсії бітів у 10% випадково вибраних тензорів глибокої моделі точність класифікатора нижнього та верхнього ієрархічного рівня помітно не змінювалась, що є проявом властивості робастності. Подальше збільшення інтенсивності відмов призводить до помітного зниження точності, але механізм відновлення відновлює більше половини падіння точності.

Також помічено, що зі збільшенням частки ушкоджених відмовами тензорів збільшується кількість ітерацій відновлення. IRQ розрахованих значень точності не перевищує 0,05, а IRQ кількості кроків відновлення не перевищує 9.

При цьому різниця в кількості класів в наборах даних CIFAR10 та CIFAR100 не мала суттєвого впливу на поведінку класифікатора.

В табл. 2.2 наведено залежність точності (Acc) та прецизійності (Prec) моделі від максимального рівня збурень (th) тестових даних за нормою L_0 та L_∞ для верхнього та нижнього ієрархічних рівнів [3].

Таблиця 2.2 – Порівняння точності та прецизійності під впливом протиборчих атак з різним рівнем збурення

Набір даних	Рівень збурення (th)	MED(Acc) на збурених тестових даних		MED(Acc) на збурених тестових даних для рівня суперкласів		MED(Prec) на збурених тестових даних		MED(Prec) на збурених тестових даних для рівня суперкласів	
		L_0-атаки	L_∞-атаки	L_0-атаки	L_∞-атаки	L_0-атаки	L_∞-атаки	L_0-атаки	L_∞-атаки
CIFAR-10	0	0,981	0,981	0,995	0,995	0,981	0,981	0,995	0,995
CIFAR-10	1	0,975	0,967	0,980	0,970	0,979	0,978	0,991	0,991
CIFAR-10	2	0,941	0,853	0,965	0,881	0,978	0,977	0,991	0,990
CIFAR-10	3	0,851	0,762	0,880	0,811	0,977	0,975	0,989	0,984
CIFAR-10	4	0,831	0,744	0,875	0,771	0,977	0,974	0,985	0,980
CIFAR-10	5	0,801	0,711	0,871	0,741	0,963	0,955	0,985	0,979
CIFAR-10	6	0,781	0,680	0,841	0,711	0,950	0,949	0,973	0,970
CIFAR-100	0	0,890	0,890	0,970	0,970	0,930	0,930	0,980	0,980
CIFAR-100	1	0,885	0,883	0,970	0,967	0,930	0,926	0,978	0,971
CIFAR-100	2	0,881	0,880	0,942	0,941	0,910	0,910	0,972	0,968
CIFAR-100	3	0,833	0,829	0,910	0,900	0,905	0,900	0,970	0,941
CIFAR-100	4	0,741	0,745	0,902	0,871	0,898	0,889	0,960	0,941
CIFAR-100	5	0,692	0,701	0,820	0,812	0,891	0,884	0,920	0,905
CIFAR-100	6	0,642	0,603	0,780	0,750	0,890	0,883	0,820	0,831

Аналіз табл. 2.2 показує, що верхній ієрархічний рівень краще поглинає збурення від протиборчих атак, а точність більше знижується під впливом L_∞-атак порівняно з L_0-атаками. Аналіз значень точності показує, що навчений класифікатор досить стійкий до атак з максимальною амплітудою th=1. Найбільш різке зниження точності відбувається при th>3. Аналіз значень точності показує, що збільшення рівня збурення в основному призводить до збільшення кількості відмов від прийняття рішень, а не помилкових спрацьовувань. Високу прецизійність можна розглядати як одну з форм витонченої деградації.

Відновлення працездатності в умовах впливу протиборчих збурень відбувається безперервно шляхом навчання з частковим залученням учителя. Черга останніх маркованих даних може оновлюватися за допомогою механізмів активного навчання. Для моделювання механізму активного навчання використовується додатковий набір збурених даних, що містить 10% мічених і 90% немічених прикладів. У табл. 2.3 наведено залежність кількості кроків налаштування (Rec_steps), необхідних для відновлення принаймні 95% продуктивності до збурення [3].

Таблиця 2.3 – Порівняння необхідної кількості ітерацій для відновлення продуктивності після впливу протиборчих атак з різним рівнем збурення

Набір даних	Рівень збурення (th)	MED(Rec_steps)		MED(Rec_steps) для рівня суперкласів		IRQ(Rec_steps)		IRQ(Rec_steps) для рівня суперкласів	
		L_0-атаки	L_∞-атаки	L_0-атаки	L_∞-атаки	L_0-атаки	L_∞-атаки	L_0-атаки	L_∞-атаки
CIFAR-10	1	12	18	10	13	1	2	2	2
CIFAR-10	2	21	29	20	21	1	1	3	3
CIFAR-10	3	32	45	31	35	2	3	2	5
CIFAR-10	4	39	50	39	42	3	3	4	4
CIFAR-10	5	50	68	41	44	2	6	3	7
CIFAR-10	6	91	111	59	52	4	5	5	6
CIFAR-100	1	34	37	20	22	3	3	4	4

Продовження таблиці 2.3

Набір даних	Рівень збурення (th)	MED(Rec_steps)		MED(Rec_steps) для рівня суперкласів		IRQ(Rec_steps)		IRQ(Rec_steps) для рівня суперкласів	
		L_0-атаки	L_∞-атаки	L_0-атаки	L_∞-атаки	L_0-атаки	L_∞-атаки	L_0-атаки	L_∞-атаки
CIFAR-100	2	39	41	42	42	5	3	3	4
CIFAR-100	3	45	45	44	45	4	5	5	5
CIFAR-100	4	46	49	49	50	2	4	4	4
CIFAR-100	5	68	71	70	70	6	5	5	6
CIFAR-100	6	100	99	80	85	7	6	5	7

Аналіз табл. 2.3 показує, що відносно невеликої кількості збурених маркованих даних у поєднанні зі збуреними немаркованими даними достатньо для відновлення продуктивності та отримання робастності до заданого типу та рівня збурень. Набір даних CIFAR-100 потребує дещо більшої кількості ітерацій відновлення продуктивності порівняно з набором даних CIFAR-10. Це може бути пов'язано з великою кількістю сусідніх класів в обмеженому просторі ознак.

В табл. 2.4 приведено найгірші результати щодо швидкості адаптації нижнього ієрархічного рівня класифікатора серед результатів всіх комбінацій дрейфу концепцій. Аналіз табл. 2.4 показує, що відновлення продуктивності після появи нових класів відбувається швидко, вимагаючи до 56 кроків (тюнінгових міні-пакетів) з набором даних CIFAR-10 та до 62 кроків з набором даних CIFAR-100 [2, 3]. Однак для відновлення продуктивності після реального дрейфу концепції знадобилося до 95 кроків з набором даних CIFAR-10 та 97 кроків з набором даних CIFAR-100. IRQ табличних значень менше 8, тобто для відновлення продуктивності потрібно більш ніж у 32 рази менше кроків (міні-партій), ніж для навчання з нуля. Поведінка алгоритму на обох наборах даних схожа, але для випадку CIFAR-100 потрібно дещо більше ітерацій для навчання та відновлення.

Запропонована модель класифікатора має багатосекційну структуру, призначену для реалізації адаптивних розрахунків та підвищення

узагальнюючих можливостей моделі за рахунок самодистиляції знань. Проведено порівняння точності в умовах впливу збурень і швидкості відновлення ефективності для моделі, яка використовує виходи проміжних секцій, і для моделі, яка використовує тільки останній вихід (останній шар моделі) для виявлення впливу багатосекційної структури на резільєнтність моделі.

Згідно з концепцією, резільєнтний класифікатор може бути побудований з модулів, які мають різну мікроархітектуру, але властивість резільєнтності системи повинна зберігатися за рахунок запропонованої макроархітектури та методу навчання. Однак, в задачах класифікаційного аналізу, поряд зі згортковими структурними елементами, широкого розповсюдження набувають трансформаторні структурні елементи. Тому цікаво розглянути поведінку запропонованого класифікатора з екстрактором ознак (backbone), побудованим на основі відомих блоків Swin Transformer [104]. Крім того, традиційний підхід до побудови класифікаційної головки моделі полягає у використанні повнозв’язного (Dense) шару та нормалізації виходу функцією Softmax. Тому варто перевірити, як вплине заміна класифікатора-прототипу на повнозвязний (Dense) шар на резільєнтність класифікатора.

Таблиця 2.4 – Швидкість відновлення продуктивності після додавання нових класів або взаємного дрейфу класів на нижньому ієрархічному рівні

Збурення	Кількість кроків для навчання з нуля		Кількість кроків для відновлення шляхом навчання з учителем (максимальне значення серед експериментів)	
	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Додавання одного нового класу	2400	2800	33	38
Додавання двох нових класів	2400	3000	56	62
Реальний дрейф концепцій між парою класів	2800	3000	73	75
Реальний дрейф концепцій між трійкою класів	2800	3200	95	97

В табл. 2.5 наведено порівняння моделі, яка використовує виходи окремих секцій та моделі, яка використовує лише один вихід на останньому шарі за їх точністю в умовах впливу збурень з урахуванням різної реалізації екстрактора ознак та класифікаційного модуля. В табл. 2.6 наведено порівняння швидкості відновлення в залежності від наведених вище модифікацій. Для спрощення використано лише набір даних CIFAR-10 та два різних типи збурень – інжекція помилок з fault_rate=0,3 та протиборчої L∞-атаки з th=3. IRQ значень точності з табл. 2.5 не перевищує 0,03, а IRQ кількості кроків відновлення з табл. 2.6 не перевищує 5.

Аналіз табл. 2.5 і табл. 2.6 показує, що всі розглянуті модифікації не перевершують запропоновані варіанти на рис. 2.7. Однак точність класифікатора та швидкість відновлення ефективності помітно вищі у разі використання проміжних виходів моделі. При цьому Swin Transformer Blocks забезпечують дещо вищі значення точності у порівнянні з ResNet Blocks, особливо в умовах впливу протиборчих атак. Однак відновлення ефективності класифікатора з естрактором ознак на основі Swin Transformer Blocks потребує більшої кількості кроків точного настроювання порівняно з використанням ResNet Blocks.

Таблиця 2.5 – Порівняння точності модифікацій моделі в умовах впливу збурень

Виходи проміж-них секцій врахову-ються ?	Збурення	MED(Acc) після збурення			
		Екстрактор ознак, основа-ний на ResNet блоках		Екстрактор ознак основа-ний на Swin блоках	
		Класифікатор на основі прототипів	Класифікатор на основі Dense-шару	Класифікатор на основі прототипів	Класифікатор на основі Dense-шару
Так	Інжекція несправ-ностей (fault_rate=0.3)	0,852	0,831	0,849	0,841
Ні	Інжекція несправ-нотей (fault_rate=0.3)	0,802	0,792	0,810	0,800

Продовження таблиці 2.5

Виходи проміж-них секцій врахову-ються ?	Збурення	MED(Acc) після збурення			
		Екстрактор ознак, основа-ний на ResNet блоках		Екстрактор ознак основа-ний на Swin блоках	
		Класифікатор на основі прототипів	Класифікатор на основі Dense-шару	Класифікатор на основі прототипів	Класифікатор на основі Dense-шару
Так	Протиборча L_∞-атака (th=3)	0,762	0,712	0,782	0,722
Ні	Протиборча L_∞-атака (th=3)	0,723	0,685	0,754	0,709

Таблиця 2.6 – Порівняння необхідної кількості кроків для відновлення ефективності для різних модифікацій моделі в умовах впливу збурень

Виходи проміж-них секцій врахову-ються ?	Збурення	MED (Rec_steps)			
		Екстрактор ознак, осно-ваний на ResNet блоках		Екстрактор ознак оснований на Swin блоках	
		Класифікатор на основі прототипів	Класифікатор на основі Dense-шару	Класифікатор на основі прототипів	Класифікатор на основі Dense-шару
Так	Інжекція несправ-ностей (fault_rate= 0.3)	25	45	55	95
Ні	Інжекція несправ-нотей (fault_rate= 0.3)	151	277	240	297
Так	Протиборча L_∞-атака (th=3)	41	83	95	173
Ні	Протиборча L_∞-атака (th=3)	270	450	403	489

Аналогічно можна розглянути, як впливає відсікання виходів проміжних секцій на швидкість відновлення після впливу дрейфу концепцій. З табл. 2.7 видно, що найменша швидкість адаптації нижнього ієрархічного рівня класифікатора відповідає ситуації відсікання виходів проміжних секцій.

Аналіз табл. 2.7 і табл. 2.4 показує, що використання виходів всіх секцій моделі збільшує швидкість відновлення ефективності більш ніж на 20%, тобто для досягнення того ж результату потрібна на 20% менша кількість маркованих даних.

Таблиця 2.7 – Швидкість відновлення продуктивності після додавання нових класів або взаємного дрейфу класів на нижньому ієрархічному рівні з відсіканням виходів проміжних секцій

Збурення	Кількість кроків для навчання з нуля		Кількість кроків для відновлення ефективності шляхом навчання з учителем (максимальне значення серед експерементів)	
	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Додавання одного нового класу	3600	5600	41	47
Додавання двох нових класів	2600	4600	68	72
Реальний дрейф концепцій між парами класів	3600	4600	88	88
Реальний дрейф концепцій між трійками класів	3600	4800	130	145

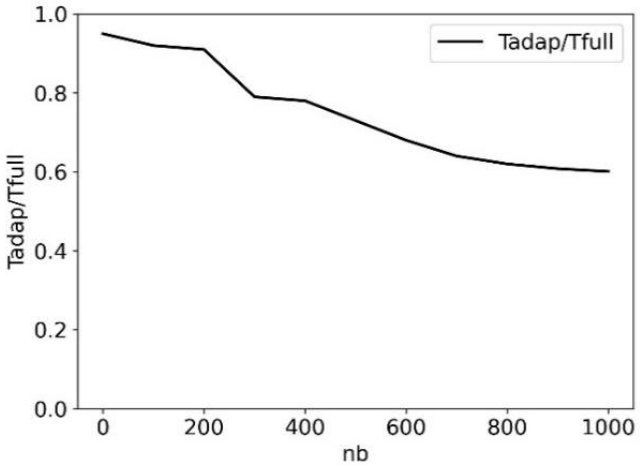
Передбачається, що в міру навчання багатосекційної архітектури моделі покращується її обчислювальна ефективність виводу (екзамену) за рахунок

економії ресурсів на простих зразках даних без збурень. На рис. 2.8 показано залежність відношення середнього часу роботи в адаптивному режимі T_adap до часу виведення через всю мережу T_full від кількості навчальних міні-пакетів (рис. 2.8*а*), максимального рівня збурення від протиборчої L_∞-атаки (рис. 2.8*б*) та fault_rate (рис. 2.8*в*) [2, 3].

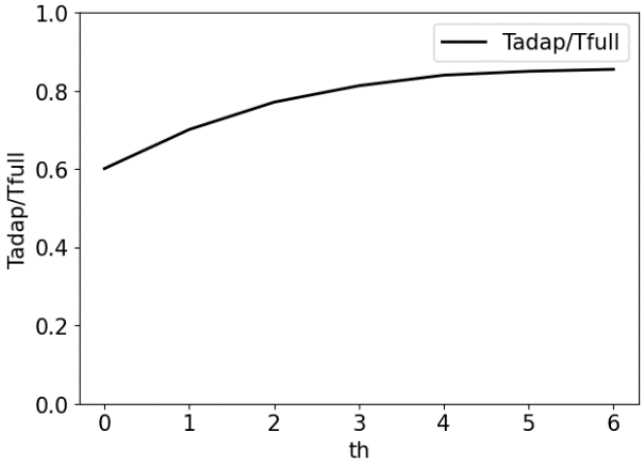
Аналіз рис. 2.8 підтверджує гіпотезу про те, що середній час виведення зменшується в міру навченості багатосекційної нейромережі. Також середній час виведення збільшується у випадку збільшення рівня збурення від протиборчої атаки або у випадку збільшення частки ушкоджених несправностями тензорів.

Таким чином, запропонований класифікатор може поглинати певний рівень збурень, виявляти зразки поза навчальним розподілом та дрейфом концепції, забезпечувати витончену деградацію, відновлення та покращення характеристик. І він працює краще, ніж традиційний підхід. Недоліком запропонованого підходу можна вважати необхідність використання додаткових черг для діагностичних, маркованих та немаркованих останніх оброблених даних. Також відсутні механізми вимірювання та покращення продуктивності безпосередньо класифікатора. Якщо покращення і відбувається, то воно може розглядатися як побічний ефект відновлення продуктивності.

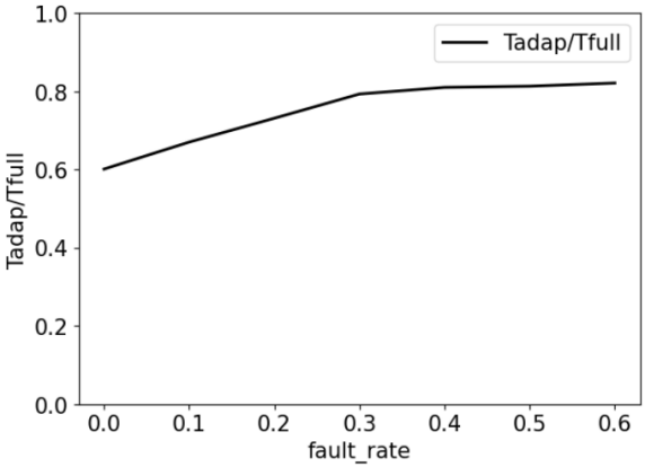
Вибір параметрів моделі потрібно здійснювати з урахуванням компромісу між резільєнтністю та продуктивністю моделі (2.1.5) [1]. Одним з важливих гіперпараметрів моделі класифікатора є розмірність ознакового опису. У табл. 2.8 показано залежність показника резільєнтності (2.1.4) та ефективності класифікатора $\overline{J}$ після відновлення, що обчислюється як усереднене за алфавітом класів значення інформаційного критерію (2.2.11), від частки ушкодженим відмовами тензорів для розмірності простору ознак N=64 та N=128 [2]. У табл. 2.5.9 показано залежність показника резільєнтності (2.1.4) та ефективності класифікатора $\overline{J}$ після відновлення від рівня збурення зображення th для розмірності простору ознак (N=64 та N=128).



a



b



c

Рисунок 2.8 – Залежність відношення середнього часу виведення в адаптивному режимі до часу виведення через всю мережі від фактору впливу : *а* – кількість навчальних міні-партій; *б* – максимальний рівень протиборчної L_∞-атаки; *в* – частка ушкоджених несправностями тензорів

Таблиця 2.8 – Результати відновлення для різної частки пошкоджених тензорів та різної розмірності ознакового опису на нижньому ієрархічному рівні класифікатора

fault_ rate	R		$\bar{J}$	
	N=64	N=128	N=64	N=128
0,1	0,978	0,979	0,981	0,980
0,3	0,945	0,951	0,970	0,975
0,5	0,881	0,880	0,952	0,961
0,6	0,674	0,511	0,652	0,651

Таблиця 2.9 – Результати відновлення для різного рівня L∞-збурення та різної розмірності ознакового опису на нижньому ієрархічному рівні класифікатора

th	R		$\bar{J}$	
	N=64	N=128	N=64	N=128
1	0,980	0,988	0,988	0,980
3	0,955	0,965	0,988	0,979
5	0,885	0,905	0,975	0,965
10	0,793	0,843	0,932	0,930

Аналіз табл. 2.8 показує, що немає однозначної залежності між розмірністю ознакового опису, резільєнтністю до несправностей та інформаційним критерієм ефективності після відновлення. Аналіз табл. 2.9 показує, що розмірність N=128 хоч і поступається за інформаційним критерієм ефективністю після відновлення, але забезпечує краще компромісне рішення згідно формули (2.1.5) за рахунок значного покращення резільєнтності моделі.

Таким чином, підтверджено перевагу у використанні згорткового багатосекційного екстрактора ознак з модулем класифікації на основі прототипів порівняно з односекційним екстрактором ознак і модулем класифікації на основі повнозв’язних шарів. Продемонстровано здатність запропонованого алгоритму навчання і екзамену забезпечувати реалізацію механізмів керованої деградації та відновлення ефективності. При цьому експерементально виявлено вплив розмірності ознакового подання на резільєнтність до протиборчих атак.

ВИСНОВКИ

Аналіз сучасного стану та тенденцій розвитку систем штучного інтелекту показав, що моделі систем штучного інтелекту будують на основі багатошарового екстрактора ознак та чутливого до задачі вихідного модуля. Вихідний модуль може вирішувати задачу класифікації, регресії, кластер-аналізу, детектування, сегментації, або генерації (синтезу) образів з певного розподілу. При цьому основні досягнення в галузі штучного інтелекту пов’язані з обробкою великих обсягів розмічених і нерозмічених даних. Питання синтезу і операційної підтримки життєвого циклу інтелектуальних алгоритмів, що неперервно навчаються і удосконалюються, утворюють активну область наукових досліджень.

Дослідження за останні 10 років показали, шо СШІ вразливі до численних деструктивних збурюючих факторів. До деструктивних факторів інтелектуальних систем належать інжекція апаратних несправностей, протиборчі атаки, дрейф концепцій, новизна (дані поза навчальним розподілом), пропуски і помилки в даних. Кожен деструктивний фактор було досліджено у числених наукових працях, однак відсутні дослідження щодо комплексного їх впливу та ефективні методи захисту від такого сценарію. Крім цього аналіз літературних джерел показує, що відомі підходи, які реалізують окремі властивості резільєнтності і не враховують принципи раціональної резільєнтності, що є актуальним в умовах обмежених ресурсів. Тому існує потреба у реалізації таких механізмів резільєнтності, як підготовка, поглинання, витончена деградація, відновлення і адаптація до комплексного впливу різнотипних деструктивних збурень.

У науково-дослідній роботі розв’язано важливу науково-технічну прикладну задачу оцінювання і забезпечення резільєнтності СШІ. Головні наукові та практичні результати роботи полягають у такому:

– запропоновано критерій оптимізації для забезпечення резільєнтності СШІ до деструктивних збурень, який дозволяє знаходити компроміс між інтегральним показником резільєнтності та ефективністю СШІ, що функціонує в режимі без деструктивних збурень;

– удосконалено алгоритм обчислення інформаційного критерію ефективності класифікаційної СШІ, що дозволяє обчислювати його градієнти, обмежувати робочу область його функції та використовувати його в складі функції втрат нейронної мережі;

– розроблено метод оцінювання резільєнтності СШІ до інжекції несправностей в пам’яті, шуму протиборчих атак та дрейфу концепцій, що дозволяє працювати з СШІ як з чорним ящиком та врахувати як робастність, так і швидкість відновлення ефективності СШІ;

– розроблено алгоритм ймовірнісної сертифікації резільєнтності до інжекції несправностей в пам’яті, шуму протиборчих атак та дрейфу концепцій, що дозволяє працювати з СШІ як з чорним ящиком та зменшити кількість тестів, необхідних для задоволення заданих довірчих меж;

– запропоновано множину принципів для побудови моделі та алгоритму навчання класифікатора, поєднання яких дозволяє підвищити характеристики резільєнтності до інжекції несправностей в пам’яті, шуму протиборчих атак та дрейфу концепцій з раціональним використання ресурсів середовища розгортання;

– встановлено вплив розмірності простору ознак, багатосекційності моделі і самодистиляції знань, типу класифікаційної головки та типу модулів екстрактора ознак на основні характеристики резільєнтності моделі класифікатора зображень.

Практична значимість отриманих результатів полягає в розробленні алгоритмів, що дозволяють здійснювати оцінювання рівня автономності і живучості за умов інформаційних та ресурсних обмежень, що має практичну цінність для безпілотних систем військового призначення. Запропоновані алгоритми дозволяють здійснювати оптимізацію резільєнтності і надавати певні ймовірнісні гарантії щодо робастності і швидкості відновлення ефективності, що важливо і для інфокомунікаційних систем загального призначення, оскільки сприяє зниженню накладних витрат на експлуатацію сервісу аналізу даних.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Moskalenko V. Neural network based image classifier resilient to destructive perturbation influences – architecture and training method / V. Moskalenko, A. Moskalenko // Radioelectronic and Computer Systems. – 2022. – No. 3. – P. 95–109. – DOI: https://doi.org/10.32620/reks.2022.3.07. (Scopus, Q3)

2. Image classifier resilient to adversarial attacks, fault injections and concept drift – model architecture and training algorithm / V. V. Moskalenko [et al.] // Radio Electronics, Computer Science, Control. – 2022. – No. 3. – P. 86. – DOI: https://doi.org/10.15588/1607-3274-2022-3-9 (WoS, Q4)

3. Model and Training Method of the Resilient Image Classifier Considering Faults, Concept Drift, and Adversarial Attacks / V. Moskalenko [et al.] // Algorithms. – 2022. – Vol. 15, no. 10. – P. 384. – DOI: https://doi.org/10.3390/a15100384 (Scopus та WoS, Q2)

4. Nahornyi V. Prediction of the vibration moment of mount etna based on electromagnetic signal monitoring / V. Nahornyi, V. Straser, D. Cataldi // MM Science Journal. – 2022. – Vol. 2022, no. 3. – P. 5943–5948. – DOI: https://doi.org/10.17973/mmsj.2022_10_2022054 (Scopus, Q3)

5. Moskalenko V.V. Robust Model And Training Method For Malware Recognition In IoT Devices. / V.V. Moskalenko // Proceedings of International Conference on Next Generation Cybersecurity Systems and Applications, 14-15 July, 2022, Kyiv, Ukraine. – 2022. – 17 p. (Scopus)

6. Information-Extreme Machine Learning of an On-board Ground Object Recognition System with a Choice of a Base Recognition Class / I. Naumenko, V. Piatachenko, M. Myronenko, T. Savchenko // Proceedings of 6th International Conference on Computational Linguistics and Intelligent Systems, May 12-13, 2022, Gliwice, Poland. – 2022. – 10 p. (**Scopus**)

7. Свідоцтво про реєстрацію авторського права на твір «Програма для ймовірнісної сертифікації резільєнтності класифікатора зображень до протиборчих атак і пошкодження тензорів нейромережі» № 114795 Україна /

Москаленко В. В., Москаленко А. С., Зарецький М. О., Коробов А. Г.; СумДУ; заяв. 2022-09-14; опубл. 2022-10-24.

8. Свідоцтво про реєстрацію авторського права на твір «Програма машинного навчання класифікатора зображень з підвищеною робастністю до шуму і змагальних атак» № 113206 Україна / Москаленко В.В., Москаленко А.С., Зарецький М.О, Коробов А. Г.; СумДУ; заяв. 2022-06-06; опубл. 2022-07-29.

9. Yodo N. Engineering Resilience Quantification and System Design Implications: A Literature Survey / Nita Yodo, Pingfeng Wang // Journal of Mechanical Design. – 2016. – Vol. 138, no. 11. – DOI: https://doi.org/10.1115/1.4034223.

10. Поночовний Ю. Л. Методологія забезпечення гарантоздатності інформаційно-керуючих систем з використанням багатоцільових стратегій обслуговування / Ю. Л. Поночовний, В. С. Харченко // Radioelectronic and Computer Systems. – 2020. – № 3. – С. 43–58. – DOI: https://doi.org/10.32620/reks.2020.3.05.

11. Резильєнтність комп’ютерних систем в умовах кіберзагроз: таксономія та онтологія / С. М. Лисенко [та ін.] // Radioelectronic and Computer Systems. – 2020. – № 1. – С. 17–28. – DOI: https://doi.org/10.32620/reks.2020.1.02.

12. Allenby B. Toward Inherently Secure and Resilient Societies / B. Allenby // Science. – 2005. – Vol. 309, no. 5737. – P. 1034–1036. – DOI: https://doi.org/10.1126/science.1111534

13. Haimes Y. Y. On the Definition of Resilience in Systems / Yacov Y. Haimes // Risk Analysis. – 2009. – Vol. 29, no. 4. – P. 498–501. – DOI: https://doi.org/10.1111/j.1539-6924.2009.01216.x.

14. A Framework for Assessing the Resilience of Infrastructure and Economic Systems / Eric D. Vugrin, Drake E. Warren, Mark A. Ehlen, R. Chris Camphouse / Sustainable and Resilient Critical Infrastructure Systems. – 2010. – P. 77–116. – DOI: 10.1007/978-3-642-11405-2_3.

15. Cimellaro G. P. Framework for analytical quantification of disaster resilience / G. Paolo Cimellaro, A. M. Reinhorn, M. Bruneau // Engineering Structures.

– 2010. – Vol. 32, no. 11. – P. 3639–3649. – DOI: https://doi.org/10.1016/j.engstruct.2010.08.008

16. Hollnagel E. Resilience engineering and the built environment / Erik Hollnagel // Building Research & Information. – 2013. – Vol. 42, no. 2. – P. 221–228. – DOI: https://doi.org/10.1080/09613218.2014.862607

17. Yodo N. Engineering Resilience Quantification and System Design Implications: A Literature Survey / Nita Yodo, Pingfeng Wang // Journal of Mechanical Design. – 2016. – Vol. 138, no. 11. – DOI: https://doi.org/10.1115/1.4034223.

18. Brtis J. S. Resilience Requirements Patterns / John S. Brtis, Michael A. McEvilley, Michael J. Pennock // INCOSE International Symposium. – 2021. – Vol. 31, no. 1. – P. 570–584. – DOI: https://doi.org/10.1002/j.2334-5837.2021.00855.x

19. Defining resilience analytics for interdependent cyber-physical-social networks / Kash Barker [et al.] // Sustainable and Resilient Infrastructure. – 2017. – Vol. 2, no. 2. – P. 59–67. – DOI: https://doi.org/10.1080/23789689.2017.1294859

20. Bengio Y. Representation Learning: A Review and New Perspectives / Y. Bengio, A. Courville, P. Vincent // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2013. – Vol. 35, no. 8. – P. 1798–1828. – DOI: https://doi.org/10.1109/tpami.2013.50

21. Clustering-Based Representation Learning through Output Translation and Its Application to Remote-Sensing Images / Qinglin Li [et al.] // Remote Sensing. – 2022. – Vol. 14, no. 14. – P. 3361. – DOI: https://doi.org/10.3390/rs14143361

22. Construction of Error Correcting Output Codes for Robust Deep Neural Networks Based on Label Grouping Scheme / Hwiyoung Youn [et al.] // 2021 7th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC), Beijing, China, 17–19 November 2021. – 2021. – DOI: https://doi.org/10.1109/ic-nidc54101.2021.9660486.

23. Gong Z. Diversity in Machine Learning / Zhiqiang Gong, Ping Zhong, Weidong Hu // IEEE Access. – 2019. – Vol. 7. – P. 64323–64350. – DOI: https://doi.org/10.1109/access.2019.2917620

24. A survey of the recent architectures of deep convolutional neural networks / Asifullah Khan [et al.] // Artificial Intelligence Review. – 2020. – Vol. 53, no. 8. – P. 5455–5516. – DOI: https://doi.org/10.1007/s10462-020-09825-6

25. Improving the Robustness of Deep Neural Networks via Stability Training / Stephan Zheng [et al.] // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. – 2016. – DOI: https://doi.org/10.1109/cvpr.2016.485.

26. Active semi-supervised expectation maximization learning for lung cancer detection from Computerized Tomography (CT) images with minimally label training data / Phuong Nguyen [et al.] // Computer-Aided Diagnosis, Houston, United States, 15–20 February 2020 / ed. by H. K. Hahn, M. A. Mazurowski. – 2020. – DOI: https://doi.org/10.1117/12.2549655.

27. A Survey on Contrastive Self-Supervised Learning / Ashish Jaiswal [et al.] // Technologies. – 2020. – Vol. 9, no. 1. – P. 2. – DOI: https://doi.org/10.3390/technologies9010002.

28. Improved Bidirectional GAN-Based Approach for Network Intrusion Detection Using One-Class Classifier / Wen Xu [et al.] // Computers. – 2022. – Vol. 11, no. 6. – P. 85. – DOI: https://doi.org/10.3390/computers11060085.

29. Wang H. Combining Graph Convolutional Neural Networks and Label Propagation / Hongwei Wang, Jure Leskovec // ACM Transactions on Information Systems. – 2022. – Vol. 40, no. 4. – P. 1–27. – DOI: https://doi.org/10.1145/3490478.

30. Learning to See by Looking at Noise / M. Baradad, J. Wulff, T. Wang, P. Isola, A. Torralba // 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia. – 2021. – DOI: https://doi.org/10.48550/arXiv.2106.05963

31. On the information bottleneck theory of deep learning / Andrew M. Saxe [et al.] // Journal of Statistical Mechanics: Theory and Experiment. – 2019. – Vol. 2019, no. 12. – P. 124020. – DOI: https://doi.org/10.1088/1742-5468/ab3985.

32. Discrete Infomax Codes for Supervised Representation Learning / Yoonho Lee [et al.] // Entropy. – 2022. – Vol. 24, no. 4. – P. 501. – DOI: https://doi.org/10.3390/e24040501.

33. Concept drift detection and adaptation for federated and continual learning / Fernando E. Casado [et al.] // Multimedia Tools and Applications. – 2021. – DOI: https://doi.org/10.1007/s11042-021-11219-x.

34. Generalizing from a Few Examples / Yaqing Wang [et al.] // ACM Computing Surveys. – 2020. – Vol. 53, no. 3. – P. 1–34. – DOI: https://doi.org/10.1145/3386252.

35. Meta-Learning in Neural Networks: A Survey // T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey / IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2022. – Vol. 44, no 9 P. 5149-5169. – DOI: 10.1109/TPAMI.2021.3079209.

36. Knowledge Distillation: A Survey / Jianping Gou [et al.] // International Journal of Computer Vision. – 2021. – Vol. 129, no. 6. – P. 1789–1819. – DOI: https://doi.org/10.1007/s11263-021-01453-z.

37. Doke A. Survey on Automated Machine Learning (AutoML) and Meta learning / Ashwini Doke, Madhava Gaikwad // 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021. –2021. – DOI: https://doi.org/10.1109/icccnt51525.2021.9579526.

38. Torres-Huitzil C. Fault and error tolerance in neural networks: a review / Cesar Torres-Huitzil, Bernard Girau // IEEE access. – 2017. – Vol. 5. – P. 17322–17341. – DOI: https://doi.org/10.1109/access.2017.2742698.

39. Practical Fault Attack on Deep Neural Networks / Jakub Breier [et al.] // CCS '18: 2018 ACM SIGSAC Conference on Computer and Communications

Security, Toronto Canada. – New York, NY, USA, 2018. – DOI: https://doi.org/10.1145/3243734.3278519.

40. Fault injection attack on deep neural network / Yannan Liu [et al.] // 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Irvine, CA, 13–16 November 2017. –2017. – DOI: https://doi.org/10.1109/iccad.2017.8203770.

41. A taxonomy and survey of attacks against machine learning / Nikolaos Pitropakis [et al.] // Computer Science Review. – 2019. – Vol. 34. – P. 100199. – DOI: https://doi.org/10.1016/j.cosrev.2019.100199.

42. Review of Artificial Intelligence Adversarial Attack and Defense Technologies / Shilin Qiu [et al.] // Applied Sciences. – 2019. – Vol. 9, no. 5. – P. 909. – DOI: https://doi.org/10.3390/app9050909.

43. Adversarial Attack and Defense: A Survey / Hongshuo Liang [et al.] // Electronics. – 2022. – Vol. 11, no. 8. – P. 1283. – DOI: https://doi.org/10.3390/electronics11081283.

44. Efficient Decision-Based Black-Box Adversarial Attacks on Face Recognition / Yinpeng Dong [et al.] // 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019. – 2019. – DOI: https://doi.org/10.1109/cvpr.2019.00790.

45. Hayes J. Learning Universal Adversarial Perturbations with Generative Models / Jamie Hayes, George Danezis // 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, 24 May 2018. – 2018. – DOI: https://doi.org/10.1109/spw.2018.00015.

46. Procedural Noise Adversarial Examples for Black-Box Attacks on Deep Convolutional Networks / Kenneth T. Co [et al.] // CCS '19: 2019 ACM SIGSAC Conference on Computer and Communications Security, London United Kingdom. – New York, NY, USA, 2019. – DOI: https://doi.org/10.1145/3319535.3345660.

47. Agrahari S. Concept drift detection in data stream mining: a literature review / Supriya Agrahari, Anil Kumar Singh // Journal of king saud university - computer and information sciences. – 2021. – DOI: https://doi.org/10.1016/j.jksuci.2021.11.006.

48. Anomalous example detection in deep learning: a survey / Saikiran Bulusu [et al.] // IEEE access. – 2020. – Vol. 8. – P. 132330–132347. – DOI: https://doi.org/10.1109/access.2020.3010274.

49. A survey on missing data in machine learning / Tlamelo Emmanuel [et al.] // Journal of Big Data. – 2021. – Vol. 8, no. 1. – DOI: https://doi.org/10.1186/s40537-021-00516-9.

50. Fraccascia L. Resilience of Complex Systems: State of the Art and Directions for Future Research / Luca Fraccascia, Ilaria Giannoccaro, Vito Albino // Complexity. – 2018. – Vol. 2018. – P. 1–44. – DOI: https://doi.org/10.1155/2018/3421529.

51. Madni A. Affordable Resilience / A. Madni // Transdisciplinary Systems Engineering. – 2017. – P. 133-159. – DOI: https://doi.org/10.1007/978-3-319-62184-5_9.

52. Graceful degradation and related fields - ePrints Soton // Welcome to ePrints Soton - ePrints Soton. – Mode of access: https://eprints.soton.ac.uk/455349 (date of access: 18.12.2022). – Title from screen.

53. Hospedales T. Meta-Learning in Neural Networks: A Survey. / T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2021. – P. 5149-5169. – DOI: https://doi.org/10.1109/TPAMI.2021.3079209.

54. Continual lifelong learning with neural networks: A review / German I. Parisi [et al.] // Neural Networks. – 2019. – Vol. 113. – P. 54–71. – DOI: https://doi.org/10.1016/j.neunet.2019.01.012.

55. Towards Resilient Artificial Intelligence: Survey and Research Issues / Oliver Eigner [et al.] // 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 26–28 July 2021. – 2021. – DOI: https://doi.org/10.1109/csr51186.2021.9527986..

56. Olowononi F. O. Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS / F. O. Olowononi, D. B. Rawat, C. Liu // IEEE Communications Surveys &

Tutorials. – 2020. – Vol. 23, no. 1. – P. 524–552. – DOI: https://doi.org/10.1109/comst.2020.3036778.

57. Zhang L. Self-Distillation: Towards Efficient and Compact Neural Networks / Linfeng Zhang, Chenglong Bao, Kaisheng Ma // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2021. – Vol. 44, no. 8. – P. 4388-4403. – DOI: https://doi.org/10.1109/tpami.2021.3067100.

58. Marquez E. S. Deep Cascade Learning / Enrique S. Marquez, Jonathon S. Hare, Mahesan Niranjan // IEEE Transactions on Neural Networks and Learning Systems. – 2018. – Vol. 29, no. 11. – P. 5475–5485. – DOI: https://doi.org/10.1109/tnnls.2018.2805098

59. Leslie N S. A useful taxonomy for adversarial robustness of Neural Networks / Smith Leslie N // Trends in Computer Science and Information Technology. – 2020. – P. 037–041. – DOI: https://doi.org/10.17352/tcsit.000017.

60. A. Mitigating Adversarial Effects Through Randomization. / C. Xie, J. Wang, Z. Zhang, Z. Ren, A. Yuille // Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017. – 2017. – P. 1–16. https://doi.org/10.48550/arXiv.1711.01991.

61. Digital Image Representation by Atomic Functions: The Compression and Protection of Data for Edge Computing in IoT Systems / Viktor Makarichev [et al.] // Sensors. – 2022. – Vol. 22, no. 10. – P. 3751. – DOI: https://doi.org/10.3390/s22103751.

62. N. Papernot. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks / N. Papernot, P. McDaniel, X. Wu, S. Jha, A. Swami // Proceedings of the IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20-24 May 2016. – 2016. – P. 582-597. – DOI: https://doi.org/10.1109/SP35280.2016.

63. MulDef: Multi-model-based Defense Against Adversarial Examples for Neural Networks / S. Srisakaokul [et al.] // arXiv.org e-Print archive. – Mode of access: https://arxiv.org/pdf/1809.00065.pdf (date of access: 18.12.2022). – Title from screen.

64. PixelDefend: Leveraging Generative Models to Understand and Defend against Advers arial Examples / Y. Song [et al.] // Proceedings of the International Conference on Learning Representations, Vancouver CANADA, 30 Apr – 3 May 2018. – P. 1–20. – DOI: https://doi.org/10.48550/arXiv.1710.10766.

65. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models // arXiv.org. – Mode of access: https://arxiv.org/abs/1805.06605 (date of access: 18.12.2022). – Title from screen.

66. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples // arXiv.org. – Mode of access: https://arxiv.org/abs/1802.00420 (date of access: 18.12.2022). – Title from screen.

67. Kwon H. Diversity Adversarial Training against Adversarial Attack on Deep Neural Networks / Hyun Kwon, Jun Lee // Symmetry. – 2021. – Vol. 13, no. 3. – P. 428. – DOI: https://doi.org/10.3390/sym13030428

68. J. Laermann. Achieving Generalizable Robustness of Deep Neural Networks by Stability Training. / J. Laermann, W. Samek, N. Strodthoff // Proceedings of the 41st DAGM German Conference, Dortmund, Germany, 10–13 September 2019. – 2019. – P. 360–373. DOI: https://doi.org/10.1007/978-3-030-33676-9_25.

69. Jakubovitz D. Improving DNN Robustness to Adversarial Attacks using Jacobian Regularization / D. Jakubovitz, R. Giryes // Proceedings of the European Conference on Computer Vision, Munich, Germany, 8-14 Sept. 2018. –2018. – P. 1-16. – DOI: https://doi.org/10.48550/arXiv.1803.08680.

70. Reluplex made more practical: Leaky ReLU / Jin Xu [et al.] // 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes, France, 7–10 July 2020. – 2020. – DOI: https://doi.org/10.1109/iscc50000.2020.9219587.

71. Image Classification With Tailored Fine-Grained Dictionaries / Xiangbo Shu [et al.] // IEEE Transactions on Circuits and Systems for Video Technology. – 2018. – Vol. 28, no. 2. – P. 454–467. – DOI: https://doi.org/10.1109/tcsvt.2016.2607345

72. LiBRe: A Practical Bayesian Approach to Adversarial Detection / Zhijie Deng [et al.] // 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021. – 2021. – DOI: https://doi.org/10.1109/cvpr46437.2021.00103.

73. Adversarial Example Detection Using Latent Neighborhood Graph / Ahmed Abusnaina [et al.] // 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021. – 2021. – DOI: https://doi.org/10.1109/iccv48922.2021.00759.

74. Adversarial Examples Detection in Features Distance Spaces / Fabio Carrara [et al.] // Lecture Notes in Computer Science. – Cham, 2019. – P. 313–327. – DOI: https://doi.org/10.1007/978-3-030-11012-3_26.

75. Carlini N. Adversarial Examples Are Not Easily Detected / Nicholas Carlini, David Wagner // CCS '17: 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas Texas USA. – New York, NY, USA, 2017. – DOI: https://doi.org/10.1145/3128572.3140444.

76. Deep representation learning with target coding. / S. Yang [et al.] // Proceedings of the AAAI15: Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin Texas, USA, 25–30 January, 2015. – 2015. – P 3848–3854.

77. Багатоетапний метод глибинного навчання з попереднім самонавчанням для класифікаційного аналізу дефектів стічних труб / В. В. Москаленко [та ін.] // Radioelectronic and Computer Systems. – 2021. – № 4. – С. 71–81. – DOI: https://doi.org/10.32620/reks.2021.4.06

78. Moskalenko V. Neural network based image classifier resilient to destructive perturbation influences – architecture and training method / V. Moskalenko, A. Moskalenko // Radioelectronic and Computer Systems. – 2022. – No. 3. – P. 95–109. – DOI: https://doi.org/10.32620/reks.2022.3.07.

79. Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey // arXiv.org. – Mode of access: https://arxiv.org/abs/2007.00753 (date of access: 18.12.2022). – Title from screen.

80. Huang K. Functional Error Correction for Robust Neural Networks / Kunping Huang, Paul H. Siegel, Anxiao Jiang // IEEE Journal on Selected Areas in Information Theory. – 2020. – Vol. 1, no. 1. – P. 267–276. – DOI: https://doi.org/10.1109/jsait.2020.2991430..

81. Jeong Y.-S. Improvement of Handoff-state and QOS in Wireless Environment / You-Sun Jeong, U.-Gin Choe // Journal of information and communication convergence engineering. – 2010. – Vol. 8, no. 1. – P. 1–5. – DOI: https://doi.org/10.6109/jicce.2010.8.1.001.

82. Hoang L.-H. TRe-Map: Towards Reducing the Overheads of Fault-Aware Retraining of Deep Neural Networks by Merging Fault Maps / Le-Ha Hoang, Muhammad Abdullah Hanif, Muhammad Shafique // 2021 24th Euromicro Conference on Digital System Design (DSD), Palermo, Italy, 1–3 September 2021. – 2021. – DOI: https://doi.org/10.1109/dsd53832.2021.00072.

83. FTT-NAS: Discovering Fault-Tolerant Neural Architecture / Wenshuo Li [et al.] // 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), Beijing, China, 13–16 January 2020. – 2020. – DOI: https://doi.org/10.1109/asp-dac47756.2020.9045324.

84. Valtchev S. Z. Domain randomization for neural network classification / Svetozar Zarko Valtchev, Jianhong Wu // Journal of Big Data. – 2021. – Vol. 8, no. 1. – DOI: https://doi.org/10.1186/s40537-021-00455-5

85. Generalizing to unseen domains via adversarial data augmentation. / R.Volpi [et al.] // Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal CANADA, 2-8 December 2018. – 2018. – P. 1-11. https://doi.org/10.5555/3327345.3327439.

86. DIRL: Domain-Invariant Representation Learning for Generalizable Semantic Segmentation / Qi Xu [et al.] // Proceedings of the AAAI Conference on Artificial Intelligence. – 2022. – Vol. 36, no. 3. – P. 2884–2892. – DOI: https://doi.org/10.1609/aaai.v36i3.20193

87. Museba T. ADES: A New Ensemble Diversity-Based Approach for Handling Concept Drift / Tinofirei Museba, Fulufhelo Nelwamondo, Khmaies

Ouahada // Mobile Information Systems. – 2021. – Vol. 2021. – P. 1–17. – DOI: https://doi.org/10.1155/2021/5549300.

88. Generalized Deep Transfer Networks for Knowledge Propagation in Heterogeneous Domains / Jinhui Tang [et al.] // ACM Transactions on Multimedia Computing, Communications, and Applications. – 2016. – Vol. 12, no. 4s. – P. 1–22. – DOI: https://doi.org/10.1145/2998574.

89. Weakly-Shared Deep Transfer Networks for Heterogeneous-Domain Knowledge Propagation / Xiangbo Shu [et al.] // MM '15: ACM Multimedia Conference, Brisbane Australia. – New York, NY, USA, 2015. – DOI: https://doi.org/10.1145/2733373.2806216.

90. Achddou R. Nested Learning for Multi-Level Classification / Raphael Achddou, J. Matias Di Martino, Guillermo Sapiro // ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021. –2021. – DOI: https://doi.org/10.1109/icassp39728.2021.9415076.

91. Castellani A. Task-Sensitive Concept Drift Detector with Constraint Embedding / Andrea Castellani, Sebastian Schmitt, Barbara Hammer // 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021. –2021. – DOI: https://doi.org/10.1109/ssci50451.2021.9659969.

92. Self-assembled aluminum oxyhydroxide nanorices with superior suspension stability for vaccine adjuvant / Shisheng Bi [et al.] // Journal of Colloid and Interface Science. – 2022. – Vol. 627. – P. 238–246. – DOI: https://doi.org/10.1016/j.jcis.2022.07.022.

93. Javaheripi M. HASHTAG: Hash Signatures for Online Detection of Fault-Injection Attacks on Deep Neural Networks / Mojan Javaheripi, Farinaz Koushanfar // 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), Munich, Germany, 1–4 November 2021. – 2021. – P. 1-9. – DOI: https://doi.org/10.1109/iccad51958.2021.9643556.

94. RADAR: Run-time Adversarial Weight Attack Detection and Accuracy Recovery / Jingtao Li [et al.] // 2021 Design, Automation & Test in Europe Conference

& Exhibition (DATE), Grenoble, France, 1–5 February 2021. – 2021. – P. 790-795. – DOI: https://doi.org/10.23919/date51398.2021.9474113.

95. Detection and recovery against deep neural network fault injection attacks based on contrastive learning / Wang C. [et al.] // Proceedings of the 3rd Workshop on Adversarial Learning Methods for Machine Learning and Data Mining at KDD, Singapore, 14 Aug. 2021. – 2021. – P. 1-5 p.

96. Girau B. Fault tolerance of self-organizing maps / Bernard Girau, Cesar Torres-Huitzil // Neural Computing and Applications. – 2018. – Vol. 32, no. 24. – P. 17977–17993. – DOI: https://doi.org/10.1007/s00521-018-3769-6.

97. CLEAR: Contrastive-Prototype Learning with Drift Estimation for Resource Constrained Stream Mining / Zhuoyi Wang [et al.] // WWW '21: The Web Conference 2021, Ljubljana Slovenia. – New York, NY, USA, 2021. – DOI: https://doi.org/10.1145/3442381.3449820.

98. Active Learning by Acquiring Contrastive Examples / K. Margatina [et al.] // Proceedings of the Conference on Empirical Methods in Natural Language Processing, Dominican Republic, 7–11 Nov. 2021. – 2021. – P. 1-14. – DOI: https://doi.org/10.48550/arXiv.2109.03764.

99. Chen Y. Semi-Supervised Contrastive Learning for Few-Shot Segmentation of Remote Sensing Images / Y. Chen [et al.] // Remote Sensing – 2022. – Vol. 14. – P. 1–17. – DOI: https://doi.org/10.3390/rs14174254.

100. Online fast adaptation and knowledge accumulation (OSAKA): a new approach to continual learning. / M. Caccia [et al.] // Proceedings of the 34th International Conference on Neural Information Processing Systems, Canada, 6-12 December 2020. –2020. – P. 16532–16545.

101. Інформаційно-аналітична система оцінювання відповідності сучасним вимогам навчального контенту спеціальності кібербезпека / А.С. Довбиш [та ін.] // Radioelectronic and Computer Systems. – 2021. – № 1. – С. 70–80. – DOI: https://doi.org/10.32620/reks.2021.1.06.

102. Konkle T. A self-supervised domain-general learning framework for human ventral stream representation / T. Konkle, G. Alvarez // Nature

Communications. – 2020. – P. 1–13. – DOI: https://doi.org/10.1101/2020.06.15.153247.

103. Verma G. Error correcting output codes improve probability estimation and adversarial robustness of deep neural networks / G. Verma, A. Swami // Proceedings of the Advances in Neural Information Processing Systems, Vancouver, Canada, 8-14 December 2019. – 2019. – P. 8646–8656.

104. Wu J. Supervised Contrastive Representation Embedding Based on Transformer for Few-Shot Classification / J. Wu, X. Tian, G. Zhong // Journal of Physics: Conference Series. – 2022. – Vol. 2278, no. 1. – P. 012022. – DOI: https://doi.org/10.1088/1742-6596/2278/1/012022

105. Doon R. Cifar-10 Classification using Deep Convolutional Neural Network / Raveen Doon, Tarun Kumar Rawat, Shweta Gautam // 2018 IEEE Punecon, Pune, India, 30 Nov. – 2 Dec. 2018. – 2018. – P. 1-5. – DOI: https://doi.org/10.1109/punecon.2018.8745428.

106. Li G. TensorFI: A Configurable Fault Injector for TensorFlow Applications / Guanpeng Li, Karthik Pattabiraman, Nathan DeBardeleben // 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Memphis, TN, 15–18 October 2018. –2018. – DOI: https://doi.org/10.1109/issrew.2018.00024.

107. Kotyan S. Adversarial robustness assessment: Why in evaluation both L0 and L∞ attacks are necessary / Shashank Kotyan, Danilo Vasconcellos Vargas // PLOS ONE. – 2022. – Vol. 17, no. 4. – P. e0265723. – DOI: https://doi.org/10.1371/journal.pone.0265723.

108. Kotyan S. Adversarial robustness assessment: Why in evaluation both L0 and L∞ attacks are necessary / Shashank Kotyan, Danilo Vasconcellos Vargas // PLOS ONE. – 2022. – Vol. 17, no. 4. – P. e0265723. – DOI: https://doi.org/10.1371/journal.pone.0265723

109. Scalable Quantitative Verification for Deep Neural Networks / Teodora Baluta [et al.] // 2021 IEEE/ACM 43rd International Conference on Software

Engineering: Companion Proceedings (ICSE-Companion), Madrid, ES, 25–28 May 2021. – 2021. – DOI: https://doi.org/10.1109/icse-companion52605.2021.00115.