Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп’ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

Магістр
(назва освітнього ступеня)

на тему: Дослідження застосування генетичних алгоритмів для змагальних атак на багатозадачні глибокі нейронні мережі

Виконав(ла): студент(ка) VI курсу, групи СНнм-61
спеціальності *122 Комп’ютерні науки*
(шифр і назва спеціальності)

Коломийчук Д.С.

Керівник _____ Дуда О.М.
(підпис) (прізвище та ініціали)

Нормоконтроль _____ _____
(підпис) (прізвище та ініціали)

Завідувач кафедри _____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

Рецензент _____ _____
(підпис) (прізвище та ініціали)

Тернопіль
2025

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Кафедра комп’ютерноих наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)
«____» _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістр
(назва освітнього ступеня)

за спеціальністю 122 Комп’ютерні науки
(шифр і назва спеціальності)

Студенту Коломийчуку Даниїлу Андрійовичу
(прізвище, ім’я, по батькові)

1. Тема роботи Дослідження застосування генетичних алгоритмів для змагальних атак на багатозадачні глибокі нейронні мережі

Керівник роботи Липак Галина Ігорівнакандидат наук із соціальних комунікацій (PhD), доцент кафедри комп’ютерних наук
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» 11 2024 року № 4/7-1139

2. Термін подання студентом завершеної роботи

3. Вихідні дані до роботи

4. Зміст роботи (перелік питань, які потрібно розробити)
Вступ
Аналітична частина
Проєктна частина
Спеціальна частина
Охорона праці та безпека в надзвичайних ситуаціях
Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень, слайдів)
Титульний слайд
Актуальність роботи
Завдання роботи
Основна частина
Висновки

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці			
Безпека в надзвичайних ситуаціях			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	04.12.2024	*Виконано*
2.	Підбір джерел по темі роботи	15.12.2024-31.11.2024	*Виконано*
3.	Опрацювання публікацій та збір даних по	15.01.2025-25.02.2025	*Виконано*
	темі роботи		
4.	Виконання роботи згідно мети	26.02.2025-07.04.2025	*Виконано*
5.	Оформлення першого та другого розділів	15.04.2025-18.04.2025	*Виконано*
6.	Оформлення третього розділу	19.04.2025-25.04.2025	*Виконано*
7.	Оформлення четвертого розділу	26.04.2025-02.05.2025	*Виконано*
8.	Підготовка презентації	03.05.2025-10.05.2025	*Виконано*
9.	Підготовка доповіді	11.05.2025-14.05.2025	*Виконано*
10.	Оформлення кваліфікаційної роботи	15.05.2025-16.05.2025	*Виконано*
11.	Нормоконтроль	15.05.202-16.05.2025	*Виконано*
12.	Перевірка на плагіат	17.05.2025	*Виконано*
13.	Попередній захист кваліфікаційної роботи	21.05.2025	*Виконано*
14.	Захист кваліфікаційної роботи		

Студент _____ Коломийчук Д.А.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Дуда О.М.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Дослідження застосування генетичних алгоритмів для змагальних атак на багатозадачні глибокі нейронні мережі // Кваліфікаційна робота освітнього рівня «Магістр» // Коломийчук Даниїл Андрійович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СНнм-61 // Тернопіль, 2025 // С. __, рис. – __, табл. – __, кресл. – __, додат. – __, бібліогр. – __.

Ключові слова: нейромережа, машинне навчання, алгоритм, змагання.

У роботі запропоновано нову цільову багатоцільову змагальну модель атаки під назвою MAT та новий алгоритм навчання під назвою GAMAT, заснований на генетичних алгоритмах, для створення змагальних зображень, здатних цілеспрямовано впливати на MT-DNN. На основі симуляційних експериментів, розроблених з використанням набору даних Taskonomy, продемонстровано перевагу запропонованої атаки з точки зору метрики відстані пристосованості та кількості кадрів за секунду, на які впливає атака. Крім того, алгоритм навчання GAMAT має потенціал стати дуже ефективним алгоритмом навчання. Тому майбутня робота включатиме навчання та тестування варіантів GAMAT на більших наборах даних та більш віддалених POF, а також розробку масштабованих варіацій, які можуть краще використовувати переваги графічних процесорів.

ANNOTATION

Research on the application of genetic algorithms for adversarial attacks on multi-task deep neural networks // The educational level "Master" qualification work // Kolomyychuk Daniil Andriyovych // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SNnm-61 group// Ternopil, 2025 // P. __, fig. – __, tab. – __, drawing – __, append. – __, ref. – __.

Keywords: neural network, machine learning, algorithm, competition.

The paper proposes a new target multi-objective adversarial attack model called MAT and a new learning algorithm called GAMAT, based on genetic algorithms, for creating adversarial images capable of purposefully influencing MT-DNN. Based on simulation experiments developed using the Taskonomy dataset, the superiority of the proposed attack is demonstrated in terms of the fitness distance metric and the number of frames per second affected by the attack. Furthermore, the GAMAT training algorithm has the potential to become a very efficient training algorithm. Therefore, future work will include training and testing variants of GAMAT on larger datasets and more distant POFs, as well as developing scalable variations that can better take advantage of GPUs.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

MT-DNN (англ. Multi-task deep neural networks) – багатозадачні глибокі нейронні мережі.

HPS (англ. hard parameter sharing) – спільний доступ до жорстких параметрів.

FGSM (англ. Fast Gradient Sign Method) – метод швидкого градієнтного знаку.

PGD (англ. Projected Gradient Descent) - прогнозований градієнтний спуск.

MIM (англ. Momentum Iterative Methods) - ітераційні методи імпульсу.

MAT (англ. Multi-objective adversarial ATtack) – багатоцільова змагальна атака.

POF (англ. Pareto-Optimal Front) - Паретто-оптимальний фронт.

FE (англ. Fitness evaluation) – фізичні вимірювання.

GAMAT (англ. Genetic algorithm Multi-objective adversarial ATtack) – генетичні алгоритми для багатоцільової змагальної атаки

ВСТУП

Багатозадачні глибокі нейронні мережі (MT-DNN) стали кращим вибором у критично важливих для безпеки застосуваннях, таких як медична візуалізація та кіберфізичні системи (наприклад, автономне водіння) завдяки своїй здатності одночасно навчатися кільком пов'язаним завданням, що призводить до підвищення ефективності та точності моделей [1, 2]. MT-DNN продемонстрували здатність навчатися переносимим та узагальнюваним представленням для різних завдань, що має вирішальне значення у критично важливих для безпеки застосуваннях, де надійність та стійкість є обов'язковими. Крім того, завдяки спільному навчанню кількох завдань, MT-DNN можуть ефективно використовувати спільні структури та залежності між завданнями, що призводить до кращої продуктивності порівняно з одностаточними моделями. Отже, MT-DNN здобули популярність у різних областях, де одночасно необхідно виконувати кілька пов'язаних завдань, а точність та надійність є вирішальними факторами. Крім того, MT-DNN виконують два або більше завдань зору на одному зображенні або відео в програмах комп'ютерного зору, таких як виявлення об'єктів [3-5], сегментація зображень [6-9] та класифікація зображень [10-13], що робить їх добре придатними для таких завдань, як самокеровані автомобілі та автономні багатозадачні роботи для приміщень.

1. АНАЛІТИЧНА ЧАСТИНА

1.1. Характеристика багатозадачних глибоких нейронних мереж

Було розроблено різні архітектури MT-DNN, щоб дозволити моделям глибоких нейронних мереж вивчати спільні риси в різних завданнях та використовувати ці спільні знання для кращого виконання кожного завдання. Однак у комп'ютерному зорі найчастіше використовуваною архітектурою є спільний фреймворк кодера-декодера.

Спільний фреймворк кодера-декодера — це відома архітектура для багатозадачного навчання.ування в глибокому навчанні [1]. У цій структурі для обробки вхідних даних для всіх завдань використовується один кодер, і кожне завдання пов'язане з окремим декодером, який генерує відповідні виходи. Спільний кодер дозволяє моделі вивчати спільні представлення в усі завдання. Такий спосіб обміну параметрами називається жорстким обміном параметрами (HPS) [14]. Водночас, специфічні для завдання декодери дозволяють моделі вивчати конкретні вихідні формати, необхідні для кожного завдання (рис. 1.1). Цей підхід є практичним для різних сценаріїв багатозадачного навчання, таких як численні завдання комп'ютерного зору [15, 16] та обробка природної мови [17].

Зростаючу увагу до MT-DNN у застосуваннях технічного зору можна пояснити покращеною продуктивністю багатозадачного навчання завдяки спільному використанню параметрів між завданнями, що призводить до кращої продуктивності узагальнення для кожного завдання; замість проектування кількох нейронних мереж, кожна з яких виконує навчання для одного окремого завдання [18, 19].

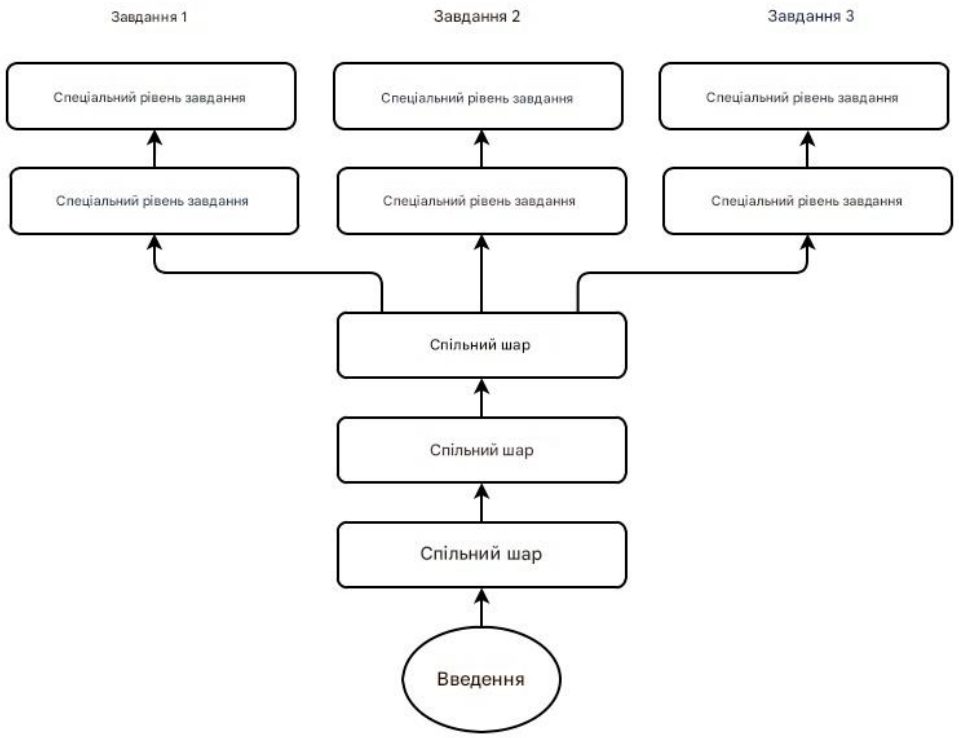

Рисунок 1.1 – Спільне використання жорстких параметрів у фреймворку кодера-декодера

Однак, MT-DNN, як правило, вразливі до зусилля, які спрямовані на максимальне погіршення продуктивності певної функції втрат. Використання взаємозалежності між кількома завданнями [16] може дозволити розробку стійких багатоцільових змагальних збурень, які можуть суттєво поставити під загрозу надійність критично важливих для безпеки систем. Крім того, розробка таких збурень сприяє розробці більш стійких та надійних систем MT-DNN.

1.2 Огляд робіт щодо багатоцільових змагальних атак

Існує мало робіт з багатоцільових змагальних атак, особливо для програм комп'ютерного зору. Традиційно, проектування змагальних вхідних даних розглядається як одноцільова задача оптимізації шляхом агрегування функцій втрат, специфічних для задачі, певним чином. Один з таких підходів полягає в обчисленні емпіричного середнього значення всіх втрат в одній задачі для отримання єдиної мети [20, 21] та проектуванні змагальних вхідних даних з використанням традиційних підходів, таких як метод швидкого градієнтного

знака (FGSM) [22], ітеративний FGSM (I-FGSM) [23], проектований градієнтний спуск (PGD) [24] та ітеративні методи імпульсу (MIM) [25]. Однак такі одноцільові змагальні атаки на MT-DNN можуть бути непрактичними, коли зловмисник не може об'єднати втрати в окремих задачах через обмеження обчислювальних ресурсів зловмисника. Наприклад, алгоритм Multitask-PGD [20] призначає однакові ваги як високовразливим, так і менш вразливим завданням, оцінюючи спільний градієнт як суму градієнтів від кожного із завдань. Така стратегія може бути дорогою для зловмисника з обмеженими ресурсами, який може віддати перевагу атакам лише на невелику підмножину високовразливих завдань. Більше того, в деяких випадках MT-DNN стали стійкими до атак на основі Multitask-PGD шляхом обфускації градієнтів [26] (окремий випадок маскування градієнта [27]) та коригування ваг завдань для захисту чутливих завдань [15]. Пізніше були розроблені адаптивні одноцільові збройні атаки, такі як Auto-PGD (APGD) [28] та Weighted Gradient Descent (WGD) [21], щоб ще більше погіршити продуктивність адаптивних MT-DNN. APGD поступово зменшує розмір кроку в PGD, щоб поступово переходити від використання цілого можливо встановити локальну оптимізацію [28]. З іншого боку, WGD зважує внесок кожного завдання під час обчислення збурення, щоб забезпечити вплив атаки на всі завдання [21]. Однак, як APGD, так і WGD атаки також об'єднують усі функції втрат в одну функцію втрат, що призводить до лінійної одноцільової задачі оптимізації. Іншими словами, такі змагальні атаки здійсненні лише тоді, коли зловмисник знає ваги, що використовуються для об'єднання втрат окремих завдань. Без таких знань природним підходом для зловмисника є використання принципу домінування, як це відбувається в багатоцільовій оптимізації.

Основний внесок роботи має три складові. Було використано модель генератора ResNet [29], який є першою генеративною моделлю атаки, що використовує принципи багатоцільової оптимізації для генерації прикладів змагання. Отже, MAT може використовувати будь-який нелінійний фронт Парето, включаючи несуміжні фронти Парето. Крім того, MAT пропонує генерацію вхідних даних змагання в режимі реального часу за швидкий, один

прямий прохід по навченій мережі, на відміну від традиційних ітеративних генераторів вхідних даних змагання. По-друге, запропоновано новий алгоритм навчання, заснований на генетичних алгоритмах [30] (скорочено, GA, наприклад, алгоритм NSGA-II [31]), який називається GAMAT, і який можна виконувати на кількох графічних процесорах для швидшої конвергенції навчання. По-третє, продуктивність MAT перевіряється шляхом навчання за допомогою GAMAT на еталонному наборі даних під назвою Taskonomy, найбільшому загальнодоступному багатозадачному наборі даних. Очевидне домінування MAT щодо деградації втрат, часу генерації вхідних даних противника та конвергенції до бажаного фронту Парето за різних сценаріїв спостерігається у порівнянні з сучасними моделями атак.

1.3 Модель загрози

Розглянемо нейронну мережу $\boldsymbol{y} = f(X;\theta)$ (як показано на рисунку 1.2), яка виконує висновок (або прогнозування) для M завдань для створення міток.

$$\begin{aligned} y_1 &= b_1(c(X;\theta_0);\theta_1), \\ &\vdots \\ y_M &= b_M(c(X;\theta_0);\theta_M), \end{aligned} \tag{1.1}$$

де $c(X;\theta_0)$ позначає вихід спільного ядра (магістралі) мережі виводу з параметром θ_0, а $b_m(X;\theta_m)$ – це гілка мережі з параметрами θ_m, що відповідає задачі m^{th}, для заданого вхідного значення $X \in \mathcal{X}$. Мережа виводу оцінюється за допомогою коефіцієнта функції втрат $\ell_m(y_m, y_m^*|X)$, що відповідає задачі m^{th}, де y_m^* – справжня мітка для заданого вхідного значення *X*, що відповідає завданню m^{th}.

Розглянемо зловмисника, метою якого є розробка нейронної мережі $\tilde{X} = g(X;\tau)$ з параметри моделі τ, що генерує змагальний вхідний сигнал $\tilde{X}$

шляхом збурення початкового вхідного сигналу X з мінімальними зусиллями. Іншими словами, зловмисник має доступ до моделі мережі логічного висновку f для виконання висновку на основі будь-яких вхідних даних $X \in \mathcal{X}$. Нехай $\mathcal{S} = \{(X_1, \boldsymbol{y}_1), \cdots, (X_K, \boldsymbol{y}_K)\}$ позначає множину K навчальних зображень. Потім мережу генераторів g навчають таким чином, щоб багатозадачна продуктивність мережі виводу $\boldsymbol{\ell}(\mathcal{S}, f, g) = \{l_1, \cdots, l_M\}$, усереднене по всьому навчальному набору S, лежить у бажаній для атакуючого області $\mathbb{L}^*$, де

$$l_i = \frac{1}{K} \sum_{k=1}^{K} \ell_i(b_i(c(\bar{X}_k; \theta_0); \theta_i), y^*_{k,i}) \tag{1.2}$$

– це емпірична середня втрата, досягнута в завданні i^{th}. Без обмеження загальності, якщо бажана область зловмисника $\mathbb{L}^*$ представлена нерівністю $h(\ell_1, \cdots, \ell_N) \geq 0$, наша мета – спроектувати нейронну мережу зловмисника таким чином, щоб для будь-якого вхідного сигналу X згенероване збурення $\tilde{X}$ задовольняло умову $h(\boldsymbol{\ell}(f(\tilde{X}; \theta), \boldsymbol{y}^*)) \geq 0$. Через багатоцільовий характер вихідної нейронної мережі f, ця модель загрози позначається як багатоцільова змагальна атака (MAT). Для спрощення бажана область позначається як $\mathbb{L}^*$ з її Парето-оптимальним фронтом (POF) $h(\ell_1, \cdots, \ell_N) = 0$.

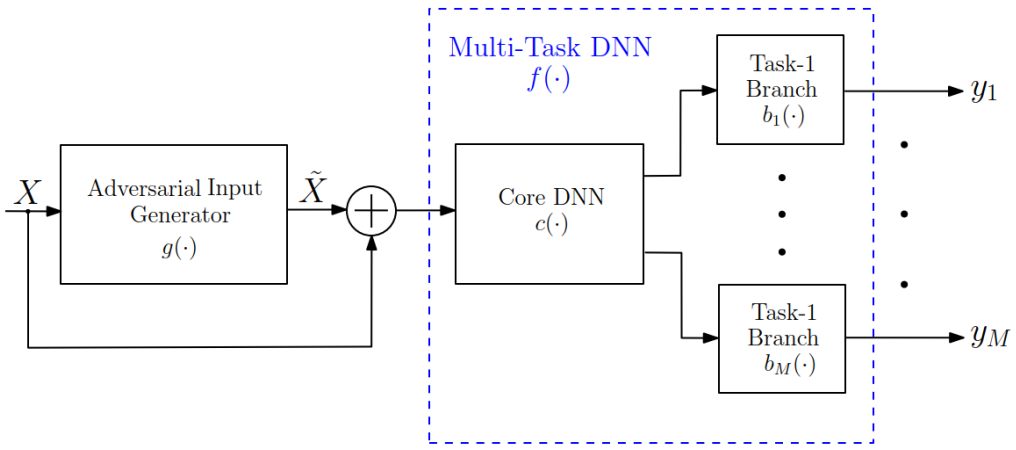

Рисунок 1.2 - MAT-генератор змагальних атак для атаки MT-DNN

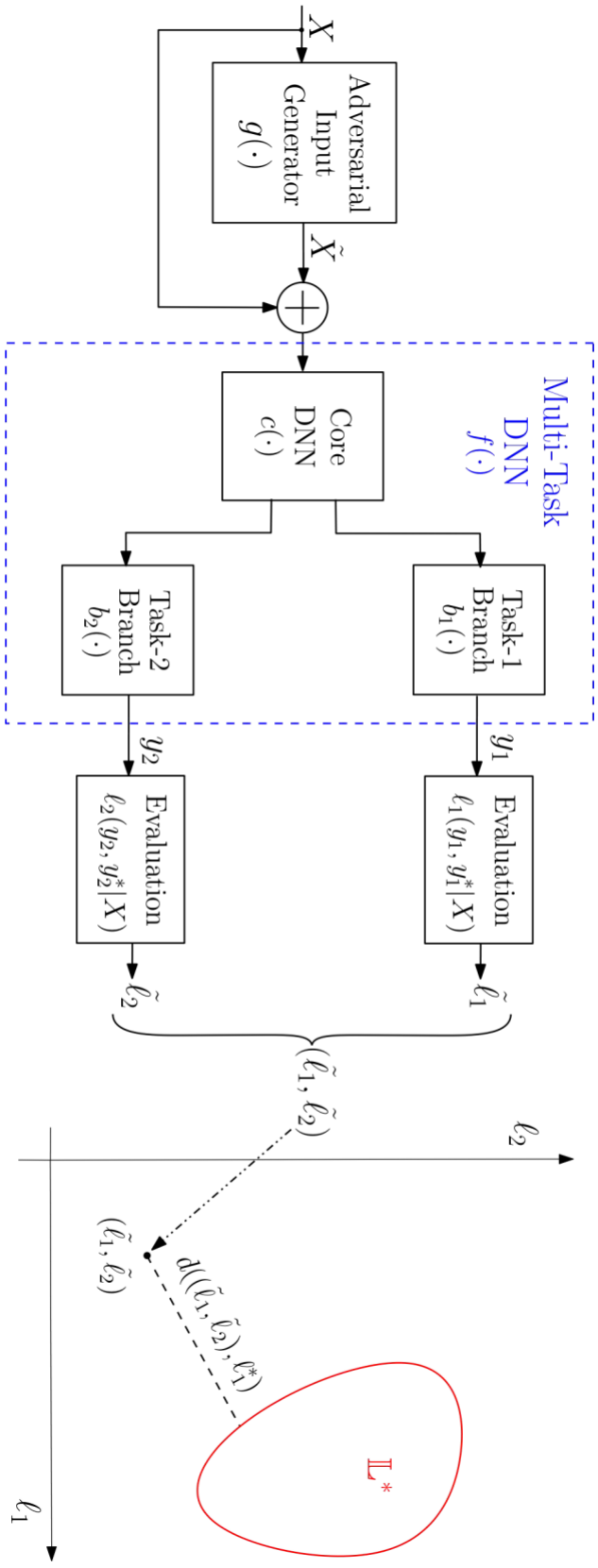

Рисунок 1.3 - Структура MAT-атак

Для кращої візуалізації описаного середовища розглянемо вхідне зображення X, генератор вхідних даних зловмисника $g(\cdot)$ та MT-DNN $f(\cdot)$, що

складається з ядра $c(\cdot)$ та двох гілок, де гілка $b_1(\cdot)$ виконує оцінку глибини, а гілка $b_2(\cdot)$ виконує оцінку нормалі до поверхні, два завдання, що зазвичай використовуються в самонавігованих машинах (рис. 1.3). Після оцінки виходу кожної гілки, результуючі значення втрат формують точку втрат після виводу $(\tilde{\ell}_1, \tilde{\ell}_2)$ у просторі втрат, де можна спостерігати мінімальну відстань між $(\tilde{\ell}_1, \tilde{\ell}_2)$ та бажаною областю зловмисника $\mathbb{L}^*$, позначеною як $d((\tilde{\ell}_1, \tilde{\ell}_2), \mathbb{L}^*)$. Отже, головною метою генератора $g(\cdot)$ є створення моделі, яка зменшить відстань d та задовольнятиме нерівність $h(\ell_1, \ell_2) \geq 0$.

Зауважте, що MAT має дві суттєві переваги над сучасними моделями змагальних атак. По-перше, генератор змагальних атак на основі нейронної мережі в MAT пропонує роботу в режимі реального часу, продуктивність шляхом перерозподілу ітераційних обчислень на етапі навчання. По-друге, MAT – єдина відома модель загрози, яка має здатність використовувати будь-яку нелінійну POF. MAT має таку перевагу, оскільки всі багатоцільові Моделі загроз розроблені для максимізації лінійної комбінації втрат окремих завдань, що еквівалентно лінійній POF. Більше того, область інтересу не обов'язково є суміжною, тобто $\mathbb{L}^*$ також може бути об'єднанням непересічних областей.

2. ПРОЄКТНА ЧАСТИНА

2.1. Навчання багатоцільової змагальної атаки на основі генетичних алгоритмів

Враховуючи, що MAT розроблено для атаки на MT-DNN, мережа генераторів зловмисника $g(\cdot;\tau)$ вимагає багатоцільового оптимізатора під час навчання. Тому в цьому розділі пропонується новий підхід до навчання на основі генетичних алгоритмів, який називається GAMAT, для проектування бажаного генератора MAT g. GAMAT складається з семи основних етапів, як показано на рис. 2.7, і бере POF $h(\cdot)$, а також T цільових точок $\boldsymbol{\ell}_1^*, \cdots, \boldsymbol{\ell}_T^*$ на POF (тобто, $h(\boldsymbol{\ell}_t^*) = 0$ для всіх $t = 1, \cdots, T$) як його вхідні атрибути. Ці T цілі (також називаються частинками здобичі) на POF включені в GAMAT для покращення різноманітності кандидатів у моделі (також називають хижаками, оскільки вони переслідують здобич протягом поколінь потомства), що виникають під час фази навчання.

2.2. Фізичні вимірювання

Для будь-якого MAT-генератора $g(\cdot;\tau)$ (який надалі називатиметься частинкою τ), його придатність (продуктивність) буде оцінюватися на кожному етапі шляхом обчислення відстані $d(\tau)$ як норма ℓ_2 між вектором втрат $\boldsymbol{l}(\tau)$ та POF $h(\cdot)$, тобто евклідова відстань між точкою втрат, згенерованою після висновку, у просторі втрат та бажаною областю атакуючого. Зауважте, що така метрика придатності оцінює здатність MAT керувати продуктивністю MT-DNN у бажаній для атакуючого області $\mathbb{L}^*$. Чим менша придатність завдання, тим ближче точка втрат, згенерована після висновку, до бажаної області POF та $\mathbb{L}^*$.

2.3. Ініціалізація популяції

На першому етапі GAMAT ініціалізує генератор атак g, вибираючи випадкову популяцію параметрів моделі з N_0. Нехай $\Omega_0 = \{\tau_1, \cdots, \tau_{N_0}\}$ позначає набір початкових параметрів моделі, вибраних за допомогою різних методів ініціалізації, таких як Glorot-Xavier-Uniform та Glorot-Xavier-Normal. За заданого вхідного значення X кожна частинка $\tau_n \in \Omega_0$ генерує різне змагальне збурення $\tilde{X}_n = g(X; \tau_n)$ для частинки n^{th} у популяції Ω_0. Після пропускання кожного збуреного вхідного сигналу $X + \tilde{X}_n$ через мережу логічного висновку f отримується сукупність результатів $Y_0 = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{N_0}\}$, де

$$\boldsymbol{y}_n = f(X + \tilde{X}_n; \theta) = f\Big(X + g(X; \tau_n); \theta\Big) \tag{2.1}$$

Нехай Ω_k позначає популяцію моделей (частинок), що розглядаються в ітерації k^{th}. Кожну частинку $\tau \in \Omega_k$ можна оцінити за допомогою емпіричного вектора середніх втрат $\boldsymbol{l}(\tau) = \{l_1(\tau), \cdots, l_M(\tau)\}$, де кожен $l_i(\tau)$ – емпіричне середнє значенняВтрата завдання i^{th} з використанням рівняння (1.2) після підстановки $\tilde{X} = g(X; \tau)$ для кожного зображення X у навчальному наборі S.

2.3. Покоління нащадків

На етапі генерації потомства використовуються чотири методи для генерації частинок-нащадків, щоб отримати новий набір $\tilde{\Omega}_{k+1}$; кожен з них детально обговорюється як окремий підетап нижче:

Етап 2.3.1 – Вибіркове оновлення на основі градієнта. Оскільки не всі частинки в Ω_k варті оновлення, найближчі до r хижаки в Ω_k вибираються для кожної t^{th} жертви $\boldsymbol{\ell}_t^*$ для всіх $t = 1, \cdots, T$, на основі відстані $d(\tau, \boldsymbol{\ell}_t^*)$ між хижаками та $\boldsymbol{\ell}_t^*$. Нехай вибрані частинки хижака для здобичі t^{th} позначимо як P_t. Потім для кожної частинки хижака $\tau \in P_t$ на зображеннях у навчальному наборі S

обчислюються градієнти втрат $\nabla d(\tau, \ell_t^*)$. Зокрема, навчальний набір додатково розбивається на міні-пакети. Використовуючи інструментарій CUDA, кожен міні-пакет надсилається на інший графічний процесор для обчислення емпіричних оцінок градієнта з використанням алгоритм зворотного просування, а потім агрегування.

Ці градієнти втрат потім використовуються для оновлення вибраних хижаків у P_t для всіх $t = 1, \cdots, T$, використовуючи стохастичний градієнтний спуск (SGD) з прискоренням Нестерова на основі імпульсу [32] та швидкістю навчання η. Нове потомство, згенероване за допомогою цього методу, включається в Ω_k для отримання нового набору частинок під назвою $\tilde{\Omega}_{k+1}$. Нарешті, придатність кожної частинки $\tau \in \tilde{\Omega}_{k+1}$ оцінюється за допомогою методу скінченних елементів.

Наприклад, розглянемо сценарій, де MT-DNN $f(\cdot)$ має дві гілки $b_1(\cdot)$ та $b_2(\cdot)$. Після початкового прямого проходження вхідного зображення через генератор $g(\cdot)$ та MT-DNN $f(\cdot)$, згенерований результат дорівнює N_0 сукупності параметрів моделі для $g(\cdot)$. Кожен параметр моделі, розміщений у $g(\cdot)$, дає різну точку втрат, згенеровану після виведення, у просторі втрат, яка в цьому прикладі містить точку втрат з двома точками жертви $\ell_1^* *$ та ℓ_2^*. Водночас, кількість хижаків на одну жертву встановлюється як $r = 3$ (рис. 3.1). Тому, під час виконання оновлення на основі селективного градієнта, r кількість найближчих до жертви хижаків ℓ_1^* вибрані (позначені як множина P_1 на рис. 3.1) для оновлення на основі кроку градієнта.

Той самий прцес повторюється для здобичі ℓ_2^*, де вибрані точки для градієнтного селективного оновлення позначаються як P_2. Нарешті, параметри кожної моделі оновлюються за допомогою SGD-оптимізатора на основі відстані d між вибраним хижаком і здобиччю.

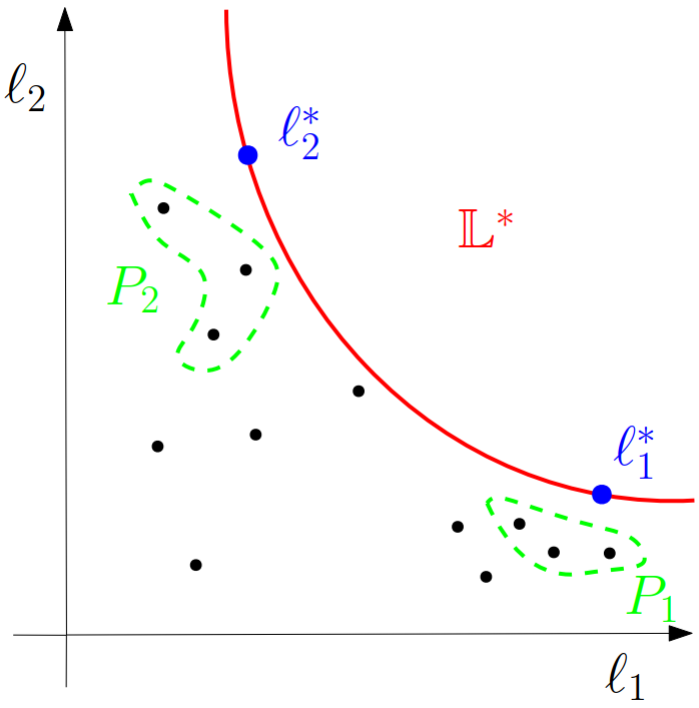

Рисунок 2.1 - Оновлення на основі вибіркового градієнта

Етап 2.3.2 – Розмноження. На цьому етапі використовується новий метод розмноження запропоновано на основі статистичної вибірки логіт-ймовірностей, які побудовані з використанням оцінки придатності $d(\tau)$ для кожного $\tau \in \tilde{\Omega}_{k+1}$. Логітна імовірність ч i^{th} астинки τ_i в $\tilde{\Omega}_{k+1}$ обчислюється як

$$p_i = \mathbb{P}(\tau_i) = \frac{\exp\big(-\lambda \cdot d(\tau_i)\big)}{\sum_{\tau_j \in \tilde{\Omega}_{k+1}} \exp\big(-\lambda \cdot d(\tau_j)\big)} \tag{2.2}$$

де $\lambda \in [0, \infty)$ можна інтерпретувати як настроюваний параметр різноманітності, який забезпечує розподіл вибіркових частинок у всіх напрямках. Зауважте, що коли $\lambda = 0$, $\boldsymbol{p}$ зводиться до рівномірного розподілу. З іншого боку, коли $\lambda \to \infty$, $\boldsymbol{p}$ зводиться до оператора min на оцінках відстані, тобто розподіл завжди вибірково використовуватиме частинку з найменшою оцінкою відстані. Враховуючи логістичний розподіл $\boldsymbol{p}$, вибірково використаємо U частинок та додамо їх до $\tilde{\Omega}_{k+1}$.

Етап 2.3.3 – Кросовер. Для будь-яких двох частинок $\tau_i, \tau_j \in \tilde{\Omega}_{k+1}$ мета будь-якого методу кросовера полягає у створенні пари нащадків $\tau_{j,i}, \tau_{j,i}$, які мають спільні характеристики обох батьківських частинок.

Зокрема, в межах кожного згорткового шару τ_i та τ_j, кожен інший вектор ваг, отриманий шляхом розщеплення кожного ядра по другому виміру кожного відповідного ядра, перетинається або міняється місцями, утворюючи дві нові частинки, як показано на рис. 2.2. Аналогічно, кожна інша вага міняється місцями в кожному пакетному шарі нормалізації τi та τj. Цей процес повторюється K разів, де K – гіперпараметр. Всі нові покоління, що генеруються на цьому етапі, включається до $\tilde{\Omega}_{k+1}$, а потім оцінюється за допомогою FE методу. Крім того, вищезгаданий метод схрещування також доповнено двома методами відбору батьків. Таким чином, на основі двох методів відбору батьків, у цій дисертації запропоновано два нових кросовери: схрещування антиподів та схрещування на основі POF.

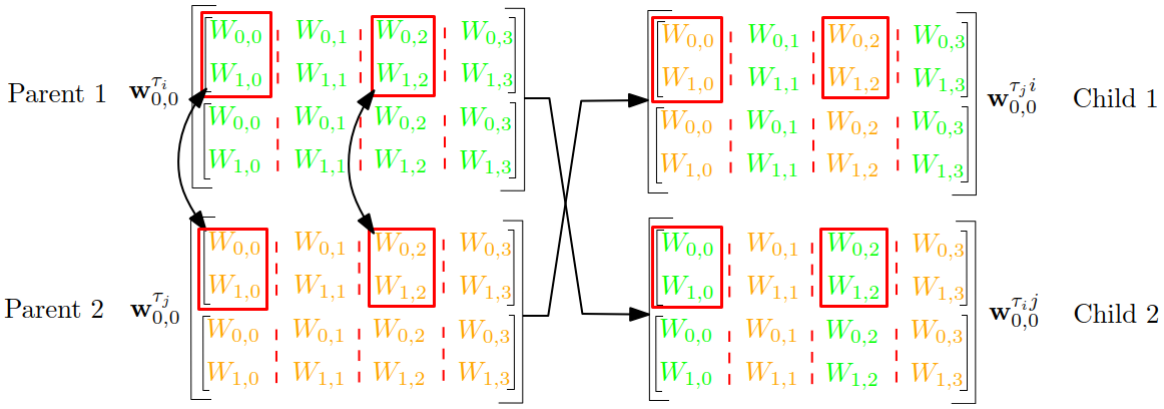

Рисунок 2.2 - Етап кросовера – для спрощення, кросовер представлено між двома 3D-тензорами ваг по виміру з індексом 2.

Антиподи-кросовер. У методі антиподного схрещування та сама процедура відбору, що й на етапі 2.3.1, використовується для визначення K хижаків для кожної з двох найбільш розділених цільових точок (антиподальних жертв) серед $\boldsymbol{\ell}_1^*, \cdots, \boldsymbol{\ell}_T^*$. У жадібному варіанті Антиподів-Кросовер, K найближчі точки хижаків до обох точок жертви сортуються у порядку зростання на основі їхньої відстані від жертви. Метод схрещування виконується між обома наборами

хижаків, дотримуючись порядку зростання точок, щоб створити потомство розміром K. Ілюстративний приклад методу жадібного відбору в антиподному схрещуванні показано на рис. 2.3, де $K = 3$, і є дві точки здобичі ℓ_1^* та ℓ_2^*. Вибрані частинки двох точок здобичі позначені як множини P_1 та P_2. Пари хижаків, вибрані для процедури схрещування, вказуються за допомогою двонаправленої стрілки. З іншого боку, метод нежадібного відбору в антиподному схрещуванні випадковим чином вибирає хижака з кожного набору P. Вибрані хижаки потім використовуються в процедурі схрещування. Метод нежадібного відбору антиподного схрещування показано на рис. 2.4 з використанням ідентичної схеми, як на рис. 2.3.

Кросовер на основі POF. Метод вибору кросовера на основі POF відрізняється від методу вибору антиподного кросовера тим, що $2K$ частинки для процедури кросовера вибираються рівномірно випадковим чином з $\tilde{\Omega}_{k+1}$ (рис. 2.5).

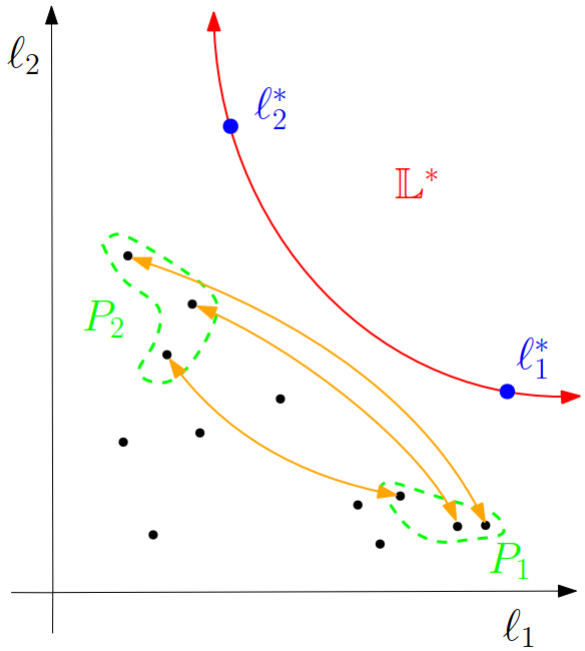

Рисунок 2.3 - Жадібний варіант методу відбору антиподів-схрещування

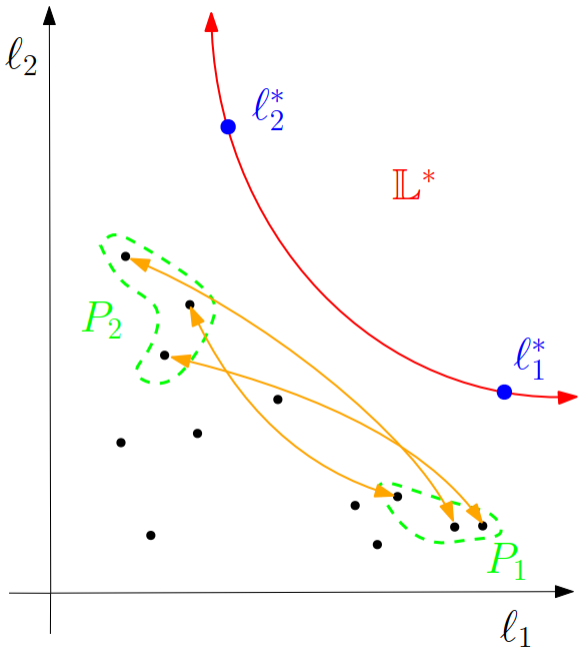

Рисунок 2.4 - Нежадібний варіант методу антиподного схрещування

Однак, у своєму жадібному варіанті, вибрані точки обираються на основі мінімальної відстані d від точки позначення точки (POF), де для процедури кросовера будуть вибрані дві найближчі точки до POF, потім наступні дві найближчі точки тощо (рис. 2.6)

Стадія 2.3.4 – Мутація. Для цього етапу пропонуються два підходи до мутації у цій роботі. Для будь-якого $\tau \in \tilde{\Omega}_{k+1}$ запропонований метод мутації замінює ваги t випадкові шари в τ з (i) випадковими числами, рівномірно дискретизованими по всій області $[-1, 1]$ у першому методі, або (ii) ваги інвертуються. Метод мутації застосовується до γ кількість шарів Г частинок з поточної популяції $\tilde{\Omega}_{k+1}$. Нове потомство, отримане будь-яким із запропонованих методів мутації, оцінюється за допомогою методу FE, а потім додається до $\tilde{\Omega}_{k+1}$. Крім того, обидва методи мутації доповнені методами батьківського відбору, запропонованими на етапах 2.3.1 та 2.3.2.

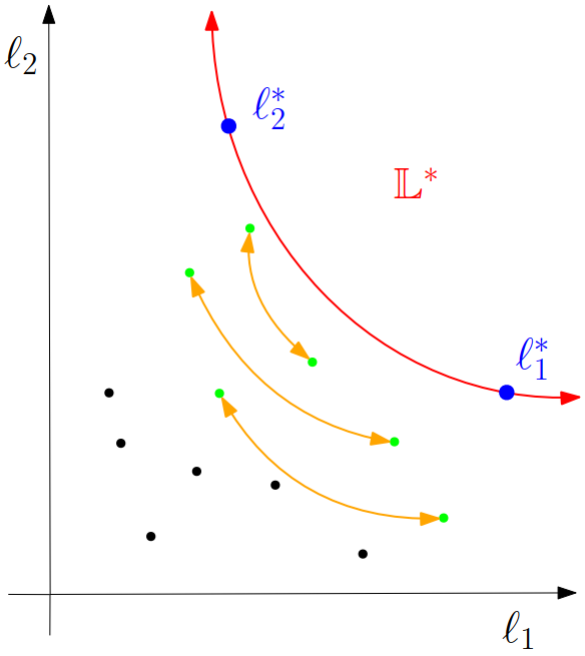

Рисунок 2.5 - Жадібний варіант методу кросоверного відбору на основі POF

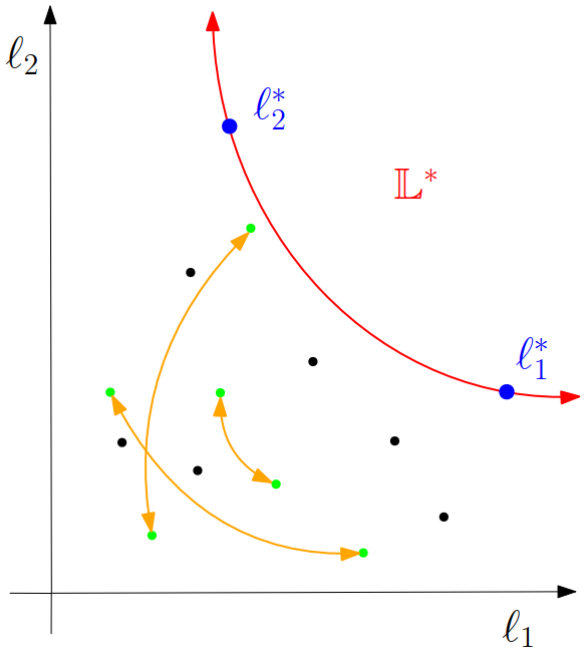

Рисунок 2.6 - Нежадібний варіант методу кросоверного відбору на основі POF

2.4. Елітизм, заснований на рангах

На цьому етапі всі частинки в $\tilde{\Omega}_{k+1}$ сортуються у порядку зростання залежно від їхньої придатностіоцінки (тобто їхня відстань від POF). Потім частинки з найвищим значенням I у цьому відсортованому списку зберігаються в Ω_{k+1}, списку частинок, що зберігаються для наступної ітерації алгоритму.

2.5. Критерії зупинки

Завершення алгоритму навчання базується на наявності принаймні одного $\tau \in \Omega_{k+1}$ такого, що $\boldsymbol{l}(\tau) \in \mathbb{L}^*$, i.e., $h(\boldsymbol{l}(\tau)) = 0$. У найгіршому випадку, коли навчання ніколи не завершується, алгоритм змушений завершитися після деякої великої, але фіксованої кількості ітерацій.

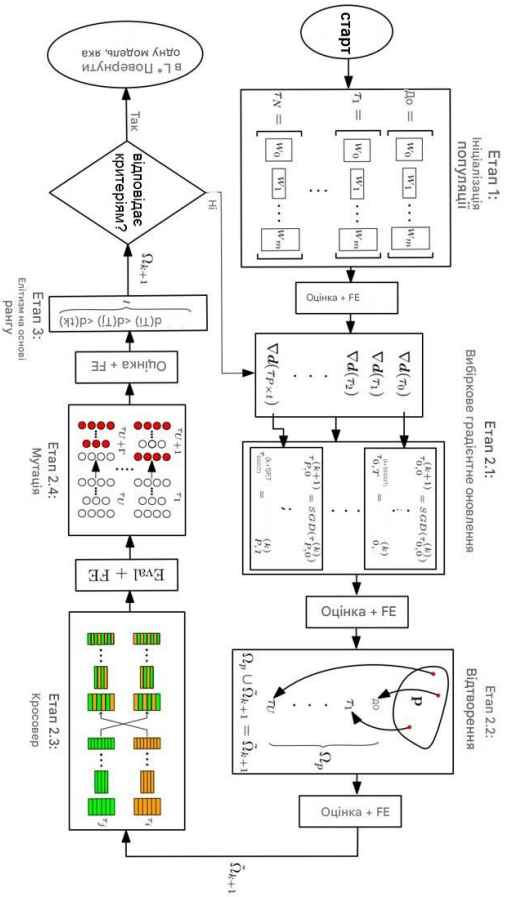

Рисунок 2.7 - Блок-схема MAT

3 СПЕЦІАЛЬНА ЧАСТИНА

3.1 Налаштування експерименту

Розглянемо сценарій, у якому робот використовує глибоку нейронну мережу для виконання кількох завдань під час навігації в невідомому приміщенні. Приклади цих завдань включають оцінку глибини для виявлення близькості до навколишніх об'єктів, оцінку нормалі до поверхні для визначення просторової орієнтації та нахилу землі, виявлення ключових точок для визначення важливих ознак на зображенні та виявлення країв для визначення країв перешкод та інших перешкод. Набір даних Taskonomy є найбільшим загальнодоступним багатозадачним набором даних для таких умов. Він містить 4,5 мільйона зображень внутрішніх сцен приблизно з 600 будівель з роздільною здатністю 1024 × 1024 [15], кожне з яких містить мітки, що відповідають 26 завданням. У цій роботі, через обмежені ресурси, для експериментів використовується «Tiny» версія набору даних Taskonomy (так звана Tiny-Taskonomy), яка містить 9464 зображення з 1500 кімнат у 30 будівлях. Зображення в Tiny-Taskonomy додатково зменшуються до 256 × 256, як у випадку з [20]. Набір даних розділяється у співвідношенні 80%-20% для навчання та тестування моделей. Набір даних курується таким чином, щоб не було перекриття кімнат, які з'являються як у навчальному, так і в тестовому наборі даних.

Попередньо навчену модель під назвою Xception, яка містить один кодер та власні декодери для кожного завдання, було отримано з [16] та використано як модель MT-DNN за замовчуванням у моїх експериментах. Ця архітектура була обрана в першу чергу через низький час виведення, щоб можна було перевірити продуктивність запропонованого алгоритму в реальному часі. Для отримання додаткової інформації про архітектуру MT-DNN зацікавлені читачі можуть звернутися до [16]. Ці експерименти розроблені для двох завдань, а саме: оцінка глибини Z-буфера (позначена (d), як у [16]) та оцінка нормальної поверхні (позначена (n), як у [16]), кожне з яких оцінюється за допомогою втрати $L1$. Для

детального опису цих завдань, будь ласка, зверніться до додаткових матеріалів, наданих у статті про набір даних Taskonomy [15].

У цій роботі використовується генератор ResNet [29, 33, 34] як модель змагального вхідного генератора для MT-DNN на основі Xception. Після навчання генератора ResNet за допомогою запропонованого алгоритму навчання, вхідне зображення, X, пропускається через навчений генератор ResNet для отримання збурення $g(X)$. Потім збурення $g(X)$ статистично нормалізується та масштабується з коефіцієнтом $min(1, \frac{\epsilon}{\|g(X)\|_\infty})$ так, щоб збурення було ледь помітним для людського ока на збуреному зображенні. Тут ϵ вибрано таким чином, що норма збурення дорівнює $L_\infty = 10$. Це результуюче збурення потім обрізається до розміру вихідного вхідного зображення, яке потім агрегується з вихідним вхідним зображенням і подається на MT-DNN. Продуктивність кожного завдання в MT-DNN на основі Xception оцінюється за допомогою функції втрат ℓ_1. Придатність результуючого вектора втрат потім оцінюється за допомогою норми ℓ_2, як обговорюється в методі скінченних елементів у розділі 2.

В експериментах цієї дисертації численні моделі навчаються за допомогою алгоритму навчання GAMAT таким чином, що для кожної навченої моделі використовуються різні гіперпараметри GAMAT. Точніше, чотири моделі навчаються таким чином, що метод кросовера, що використовується в GAMAT, відрізняється для кожної моделі. Методи кросовера:

1. Жадібні антиподи - кросовер,
2. Нежадібні антиподи-кросовери,
3. Жадібний кросовер на основі POF,
4. Нежадібний кросовер на основі POF.

Потім увага експерименту переходить на вплив кількості частинок-хижаків та кількості точок здобичі на POF під час процесу навчання. Отже, інший набір із чотирьох вищезгаданих моделей потім навчається для кожного з наступних сценаріїв:

- дві жертви з двома хижаками на жертву,

- три жертви з трьома хижаками на жертву,
- дві жертви з п'ятьма хижаками на кожну жертву,
- п'ять жертв, по два хижаки на жертву.

Зрештою, всі експерименти проводяться на наступних POF:

$$h(\ell_{(d)}, \ell_{(n)}) = 1.8 \times \ell_{(d)} \times \ell_{(n)} - 1 = 0 \quad (3.1)$$

Наступні гіперпараметри залишалися незмінними для всіх навчених моделей: кількість початкових моделей (10), кількість моделей, згенерованих шляхом відтворення (2), швидкість навчання при градієнтному оновленні (0,01), λ у розподілі ймовірностей p (1,0), кількість моделей, згенерованих шляхом мутації (4), кількість мутованих вагових шарів для кожної мутації (2), функція активації (relu) та кількість моделей, перенесених у наступне покоління (10).

Зрештою, всі експерименти в цій роботі проводились з використанням графічних процесорів V-100.

3.2. Результати та їх обговорення

Результати розділені на кілька розділів цього розділу. По-перше, представлено еволюцію частинок протягом етапів GAMAT в межах однієї ітерації. Така презентація дає читачеві краще уявлення про те, як кожен етап алгоритму впливає на простір пошуку алгоритму навчання GAMAT. Другий експеримент представляє еволюцію частинок протягом десяти епох під час навчання на одному зображенні в кожній епосі. Така презентація краще показує вплив алгоритму GAMAT протягом епох. По-третє, порівняння на основі придатності моделі, обчисленої за допомогою методу скінченних елементів, проводиться серед моделей, навчених за допомогою різних методів кросовера, описаних на етапі 2.3 алгоритму навчання GAMAT. По-четверте, проводиться аналіз втрат завдань моделей протягом процесу навчання та валідації. Після

цього проводиться аналіз гіперпараметрів. Нарешті, в останньому розділі читач побачить порівняння між багатьма моделями MAT та сучасними атаками.

3.3. Еволюція через гаматичні стадії

Заради огляду та уявлення про ефект, який можуть мати етапи GAMATУ просторі пошуку алгоритму розглядається завдання MT-DNN, де двома завданнями є оцінка глибини (позначена (d)) та оцінка нормальності поверхні (позначена (n)). Модельований зловмисник має бажану область з нелінійною POF.

$$h\left(\ell_{(d)}, \ell_{(n)}\right) = 1.8\ell_{(d)}\ell_{(n)} - 1 = 0 \qquad (3.2)$$

який буде використано в оцінці придатності та на етапі 2.1 з використанням дисциплінованого опуклого програмування, як це застосовується в оптимізаційному розв'язувачі під назвою cvxpy. Оскільки це невеликий експеримент, гіперпараметри вибрано так, щоб вони дещо відрізнялися від раніше визначених у розділі 4. Точніше, гіперпараметри GAMAT збільшуються до кращого розпізнавання впливу кожної стадії. Таким чином, кількість частинок, що генеруються на стадії розмноження, становить 8, кількість хижаків на одну точку здобичі – 7, кількість частинок, що генеруються в результаті кросовера, залишається незмінною (чверть поточної популяції), а кількість частинок, що генеруються після мутації, становить 10, де кожна частинка має чотири мутовані вагові шари. Нарешті, для кожної стадії з таким збільшенням використовується жадібна версія, а використаний метод кросовера – це жадібне кросовера на основі POF та п'яти точок здобичі на POF.

Точки втрат після виводу, що генеруються після того, як вхідні зображення збурюються генератором g, вагові параметри якого не змінилися з попередньої епохи та пройшли через MT-DNN f, позначені чорним кольором на рис. 3.1. Аналогічно, точки втрат після виводу, що генеруються після етапу градієнтного

відбору, позначені блакитним кольором (рис. 3.1). Градієнтний відбір у цьому випадку показав дуже хороші результати, оскільки деякі частинки досягли POF, що призвело до значної різноманітності серед частинок, що є критично важливим для етапу кросовера.

Наступний етап, етап відтворення, позначений червоним кольором на рис. 3.2. Природно, що вибрані частинки знаходяться найближче до POF, оскільки етап відтворення може бути жадібним у виборі частинок. Далі, частинки, згенеровані етапом кросовера, позначені фіолетовим кольором (рис. 3.3). Через природу етапу кросовера можна припустити, що моделі, згенеровані етапом кросовера, будуть знаходитися десь між його батьками, оскільки потомство складається з вагових параметрів батьків.

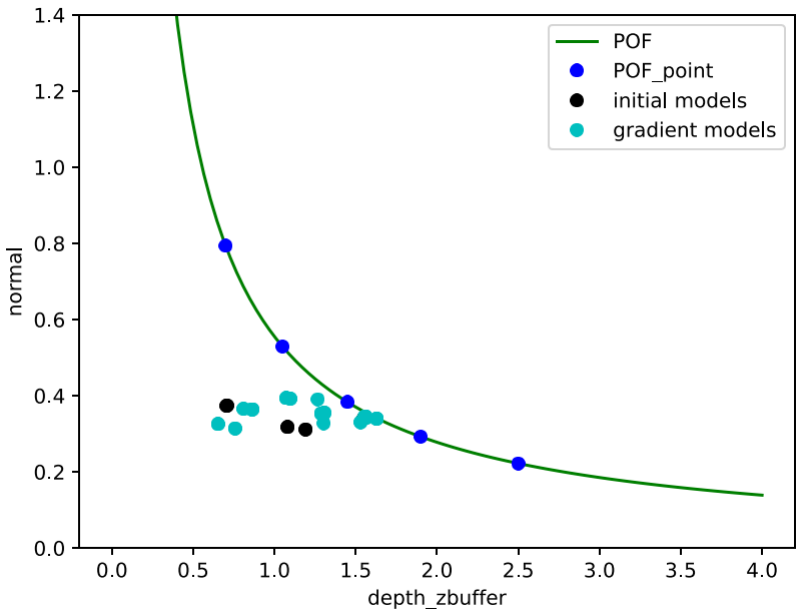

Рисунок 3.1 - Початкові втрати та частинки відбору на основі градієнта

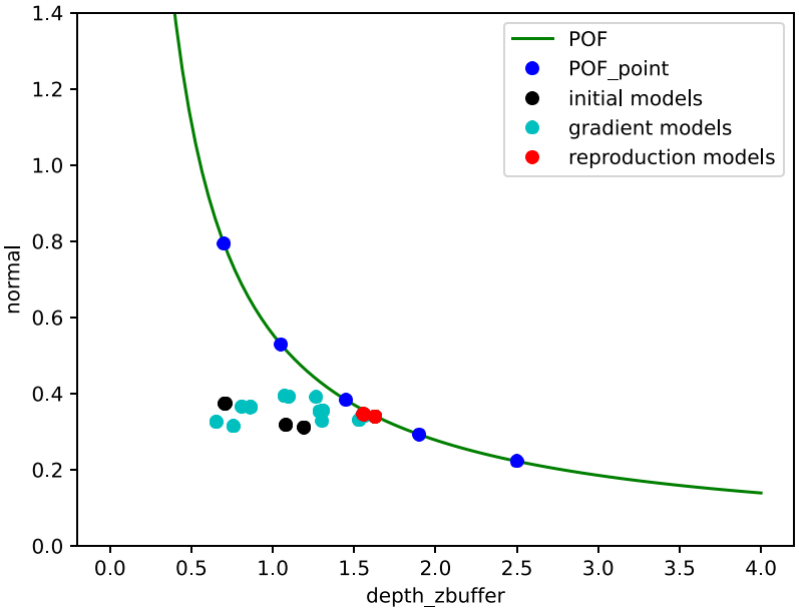

Рисунок 3.2 - Стадія розмноження

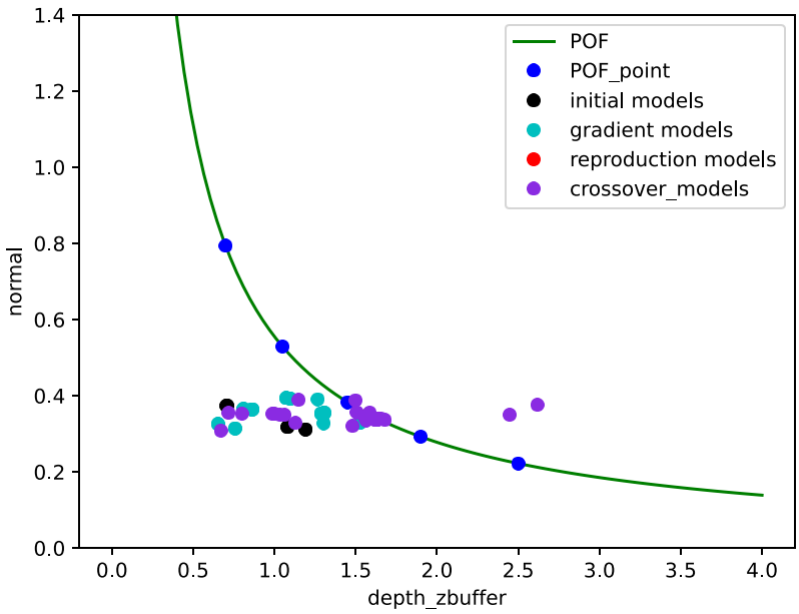

Рисунок 3.3 - Стадія кросовера

Однак, це не завжди так, як можна помітити на рис. 3.3. Такий сценарій показує можливість створення на етапі кросовера моделей, які є відмінними, точки втрат яких після виводу лежать в областях, віддалених від їхніх батьківських моделей. Тому етап кросовера може мати потужний вплив на простір пошуку алгоритму. Нарешті, на рисунку 3.4 можна спостерігати моделі,

згенеровані за допомогою оператора жадібної мутації, точки яких після виводу позначені помаранчевим кольором.

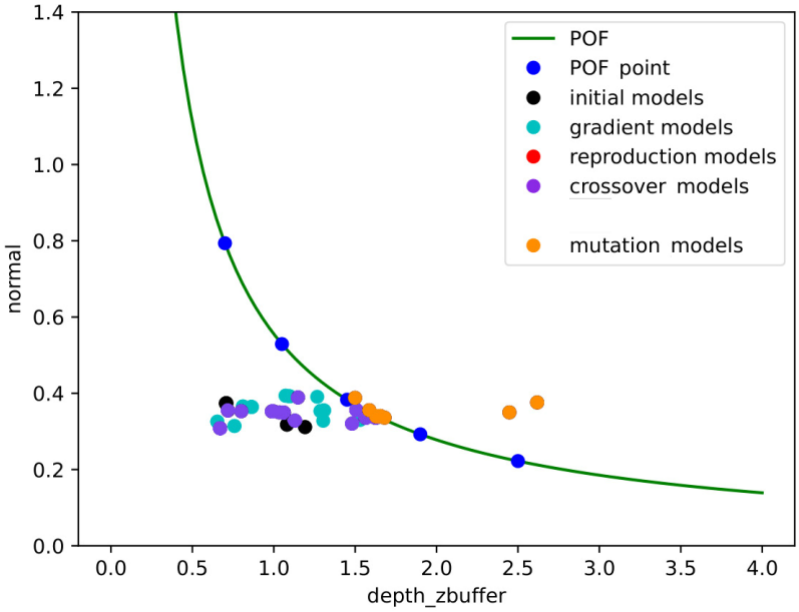

Рисунок 3.4 - Стадія мутації

Як можна помітити, точка після виводу моделі, згенерованої мутацією, вносить лише незначну різницю у відстані від її батьківської моделі. Така поведінка очікується через дуже велику кількість вагових параметрів у генераторі g та лише частину цих вагових параметрів, що мутуються.

3.4 Еволюція точок втрат після висновку протягом епох

У цьому невеликому експерименті партія розміром один використовується під час ітерації протягом десяти епох у процесі навчання. Крім того, партія розміром один обрана для кращого розуміння впливу згенерованих моделей на зображення. Вибрані завдання для MT-DNN та гіперпараметрів залишаються такими ж, як і в розділі 3.3. З рис. 3.5 по рис. 3.10 точки втрат моделей після виводу (позначені чорним кольором) показують значну різноманітність, яка пропонує різноманітні моделі.

Однак різноманітність точок втрат після виводу на одному зображенні не гарантує різноманітності на іншому вхідному зображенні. Такий сценарій можна

побачити на рис. 3.10 та 3.11. Нарешті, з рис. 3.11 по рис. 3.14 різноманітність точок менша, ніж на попередніх рисунках. Такий сценарій очікується, оскільки використана партія має розмір один, а зображення можуть мати суттєво різні характеристики. Нарешті, можна помітити, що алгоритм сходився для семи з десяти зображень за одну епоху, що може натякнути читачеві на час збіжності. Конвергенція обговорюється в наступному розділі.

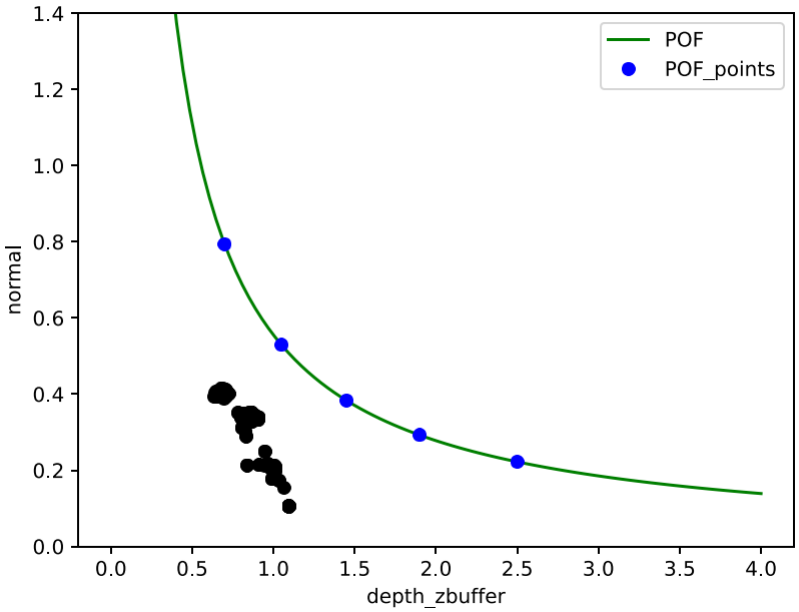

Рисунок 3.5 - Еволюція точок втрат після виведення протягом епох – Епоха 1.

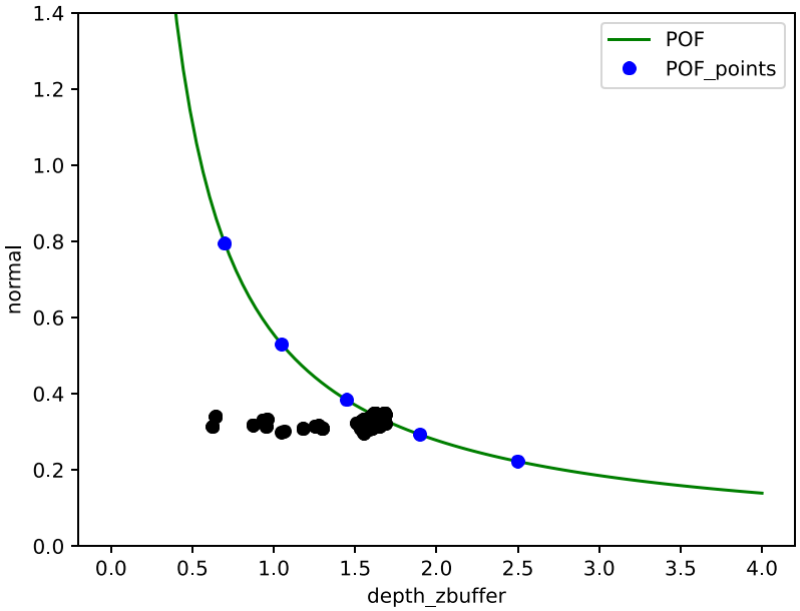

Рисунок 3.6 - Еволюція точок втрат після виведення протягом епох – Епоха 2.

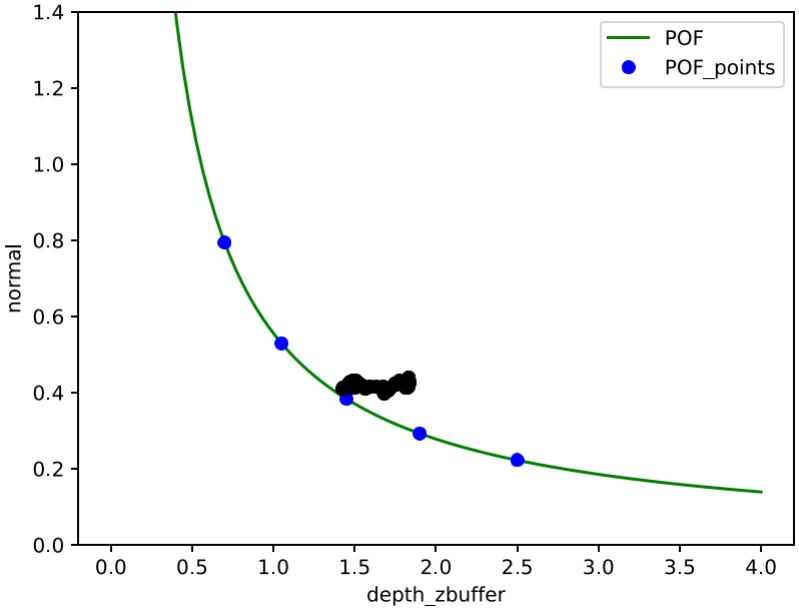

Рисунок 3.7 - Еволюція точок втрат після виведення протягом епох – Епоха 3.

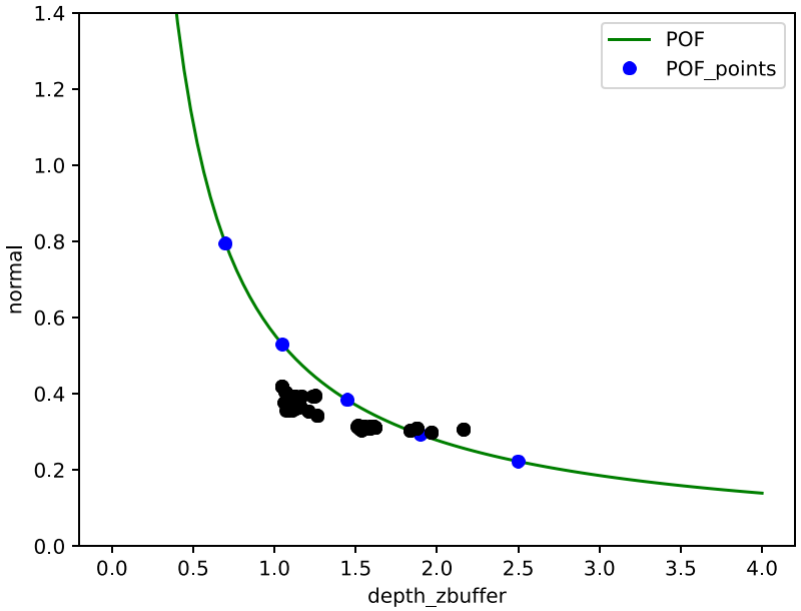

Рисунок 3.8 - Еволюція точок втрат після виведення протягом епох – Епоха 4

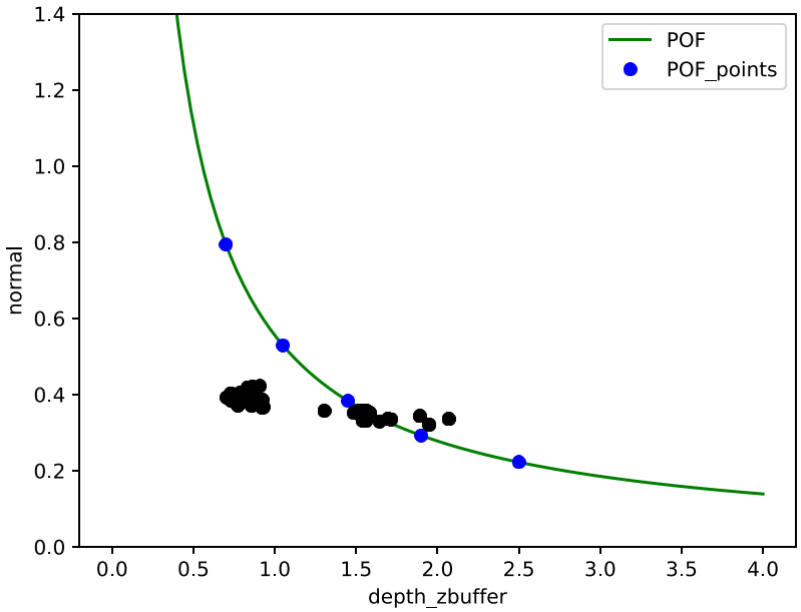

Рисунок 3.9 - Еволюція точок втрат після виведення протягом епох – Епоха 5

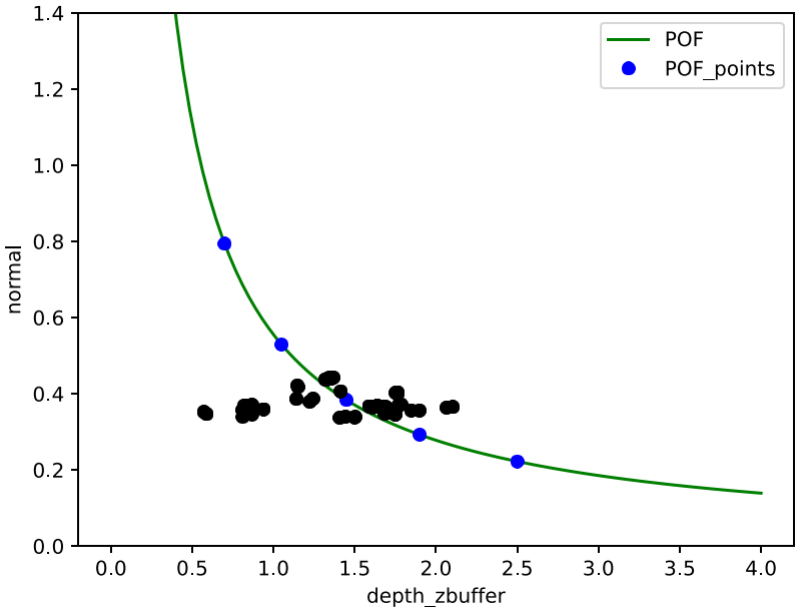

Рисунок 3.10 - Еволюція точок втрат після виведення протягом епох – Епоха 6

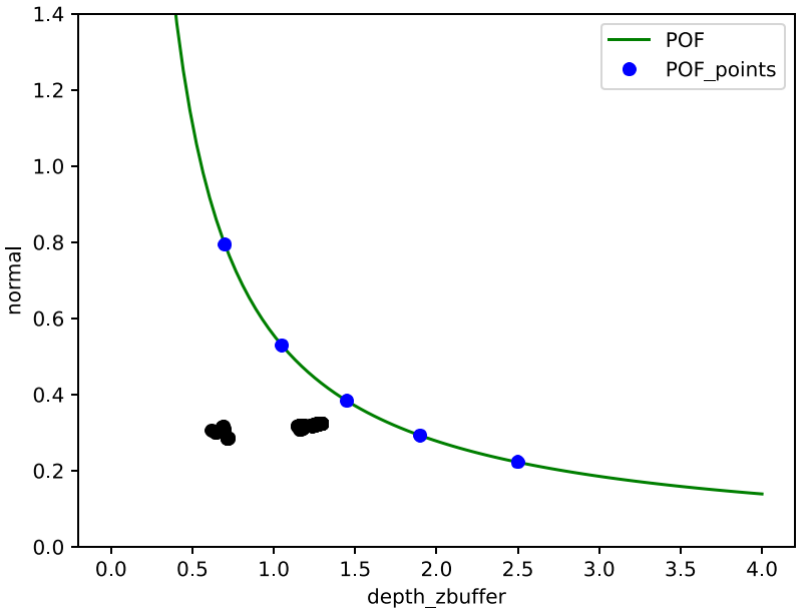

Рисунок 3.11 - Еволюція точок втрат після виведення протягом епох – Епоха 7

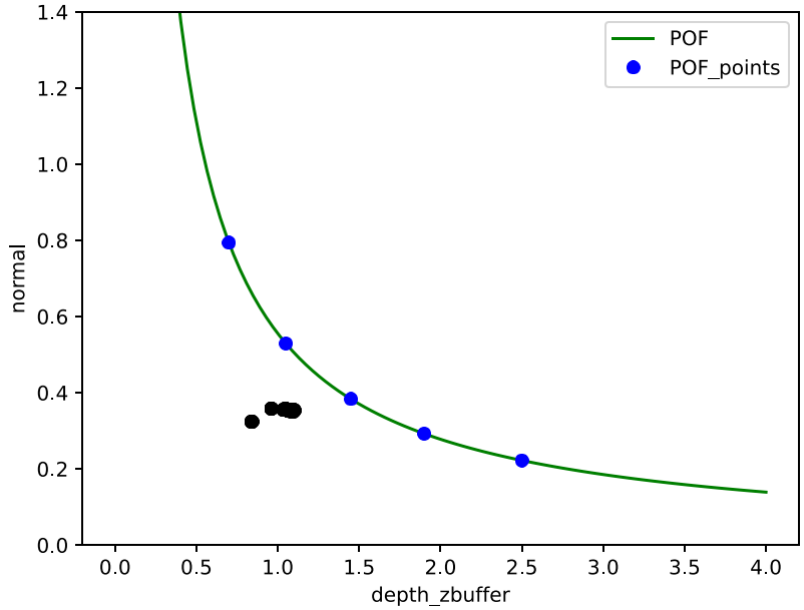

Рисунок 3.12 - Еволюція точок втрат після виведення протягом епох – Епоха 8

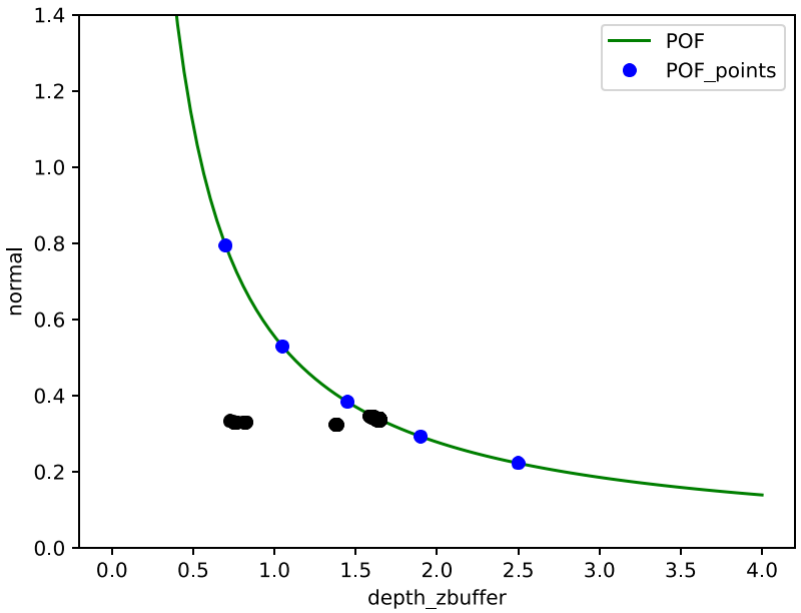

Рисунок 3.13 - Еволюція точок втрат після виведення протягом епох – Епоха 9

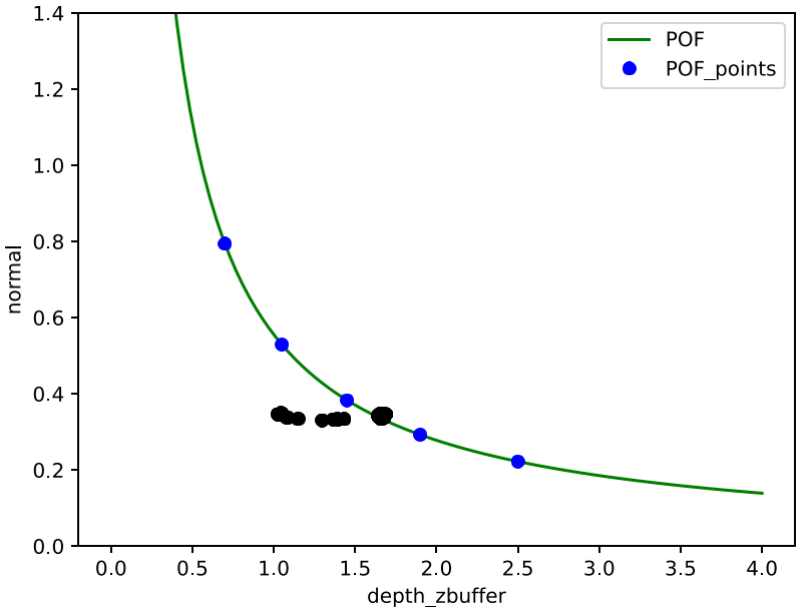

Рисунок 3.14 - Еволюція точок втрат після виведення протягом епох – Епоха 10

3.5. моделі MAT – навчання та перевірка

Навчання.

У цьому експерименті цієї роботи всі моделі навчаються протягом десяти епох, з розміром пакету чотири епохи. Причиною вибору розміру чотири є те, що менший пакет може призвести до швидшої збіжності та кращої продуктивності

узагальнення моделі завдяки частішим оновленням ваг та зміщень [35]. Спочатку розглянемо $h(\ell_{(d)}, \ell_{(n)}) = 1.8 \times \ell_{(d)} \times \ell_{(n)} - 1 = 0$, у чотирьох різних сценаріях базуються на кількості жертв та хижаків. Рис. 3.15–3.18 представляють процес навчання чотирьох моделей з різними методами схрещування у всіх сценаріях жертва-хижак. Моделі зазвичай сходяться до шостої епохи, що очікується, оскільки моделі навчаються з використанням невеликого розміру партії, а відстань від точки спостереження за точкою спостереження (POF) не є надзвичайно великою. Однак на рис. 5.17 можна помітити, що модель, що містить жадібну версію схрещування антиподів, сходиться набагато повільніше, ніж інші моделі. Те саме стосується моделі, що містить нежадібну версію схрещування на основі POF у рис. 3.18. При використанні невеликого розміру пакету шум в оцінках градієнта оптимізатора може бути вищим, оскільки градієнт розраховується на меншій підмножині даних, а не на всьому наборі даних. Отже, процес оптимізації може бути більш нестабільним через потенційно неправильний напрямок градієнта, що може призвести до збіжності моделі до неоптимального рішення або коливань навколо оптимального рішення [36].

Протягом перших трьох епох придатність до валідації двох згаданих моделей зменшується, що означає, що моделі швидко знаходять «досить гарне» рішення. Однак після третьої ітерації моделі стагнують, що можна інтерпретувати як спробу моделей знайти краще рішення через те, що модель застрягла в локальному мінімумі або в області плато, де градієнт близький до нуля, і модель не може далі покращуватися. Дивно, але придатність до валідації моделі знову зменшується, починаючи з шостої та восьмої епох відповідно. Повторне зниження являє собою те, що моделі знаходять спосіб вийти з локального мінімуму та продовжити оптимізацію до глобального мінімуму. Отже, узагальнення моделей продовжує покращуватися. За припущення, що навчання продовжуватиметься, конвергенція буде можливою, оскільки придатність моделей демонструватиме тенденцію до зниження. Така можливість залишається частиною майбутньої роботи.

Валідація.

Як зазначалося раніше, 20% набору даних використовується для перевірки навчених моделей. Рис. 3.19–3.26 впорядковані попарно таким чином, що пара рисунків показує результати тестування моделі з певним методом кросовера за епохами та середню точку втрат після виведення кожної моделі по всій навчальній частині набору даних для кожної епохи відповідно. Зверніть увагу, що для кожного результату змагальної придатності існує «чистий» результат. «Чистий» результат являє собою результат тієї ж моделі MT-DNN без атаки. Чисті результати базуються на тих самих даних тестування, що й для навчання моделей. Таке представлення дозволяє читачеві краще зрозуміти масштаб впливу MAT-атаки.

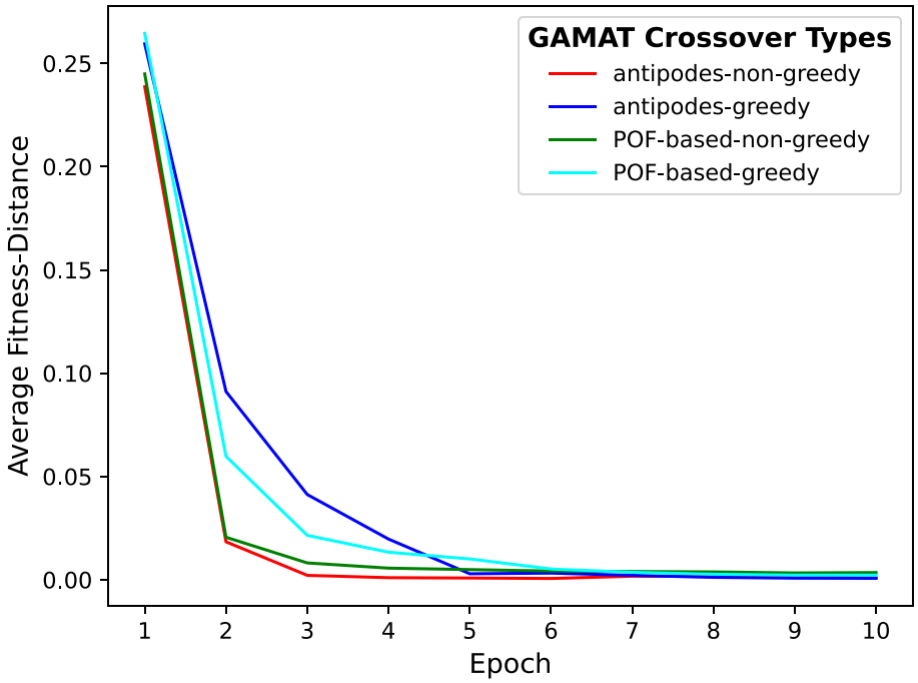

Рисунок 3.15 - Навчання за сценарієм: дві жертви з двома хижаками на жертву

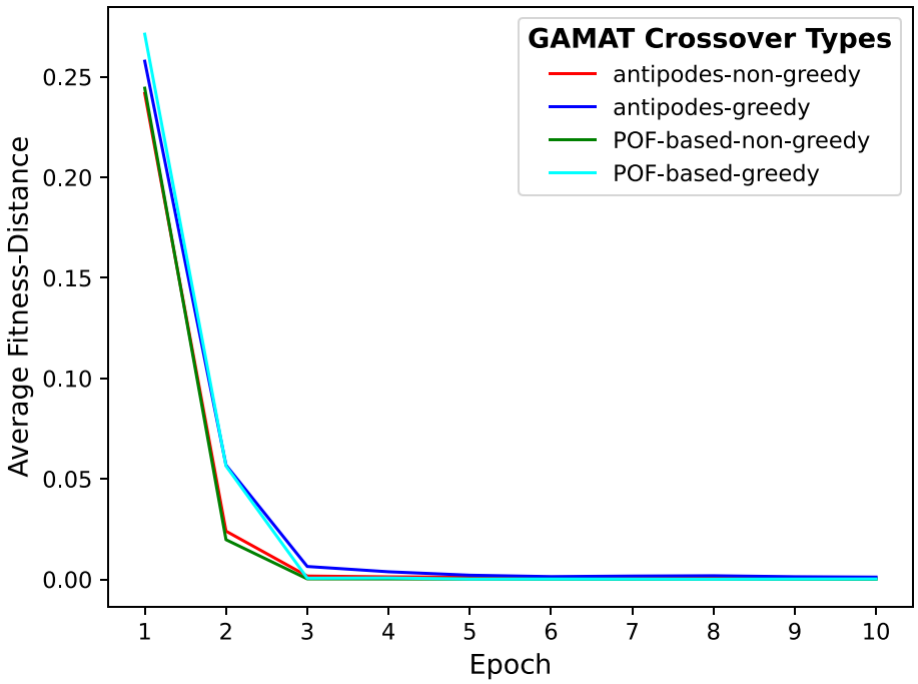

Рисунок 3.16 - Навчання за сценарієм: дві жертви з п'ятьма хижаками на кожну жертву

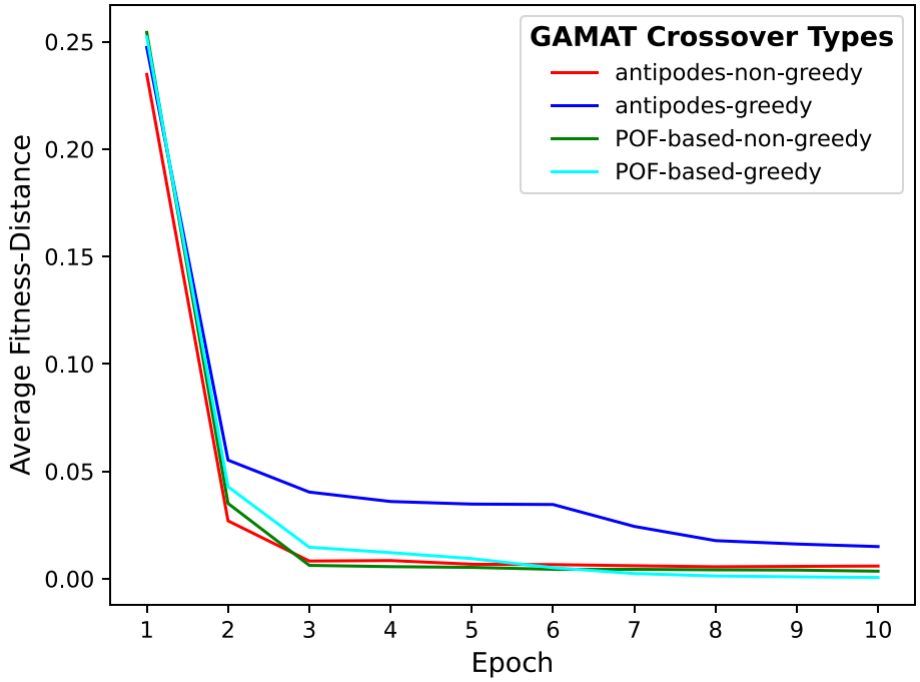

Рисунок 3.17 - Навчання за сценарієм: п'ять жертв з двома хижаками на жертву

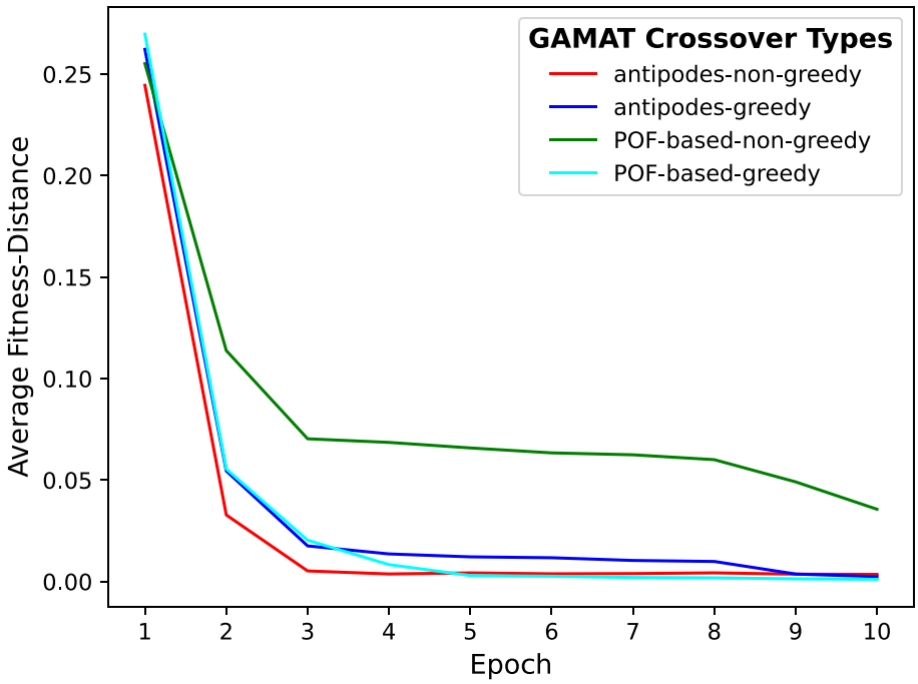

Рисунок 3.18 - Навчання за сценарієм: три жертви з трьома хижаками на жертву

Наприклад, на рис. 3.19 та 3.20 зображено сценарій навчання, де GAMAT використовує дві точки жертви та двох хижаків на кожну точку жертви. На рис. 5.19 показано, що всі моделі були навчені таким чином, що результати валідації мають значення придатності 0. Цікавим моментом щодо цього результату є те, що валідація досягла відстані придатності 0 лише за дві-чотири ітерації залежно від моделі. Тому подальше навчання після збіжності не потрібне, і навчання можна припинити. Такий метод також відомий як раннє зупинення [37]. Під час реалізації цього експерименту стан моделі записується після кожної епохи, що дозволяє використовувати стан моделі будь-якої епохи. Слідуючи за лінією відстані придатності на рис. 5.19, читач може уявити собі еволюцію середніх точок втрат після висновку протягом епох на рис. 5.20. Точки втрат після висновку моделі з жадібною версією кросовера на основі POF розташовані глибоко в області $\mathbb{L}^*$, що задовольняє нерівність $h(\ell_{(d)}, \ell_{(n)}) = 1.8 \times \ell_{(d)} \times \ell_{(n)} - 1 = 0$. Отже, дана модель добре навчена для цього конкретного POF.

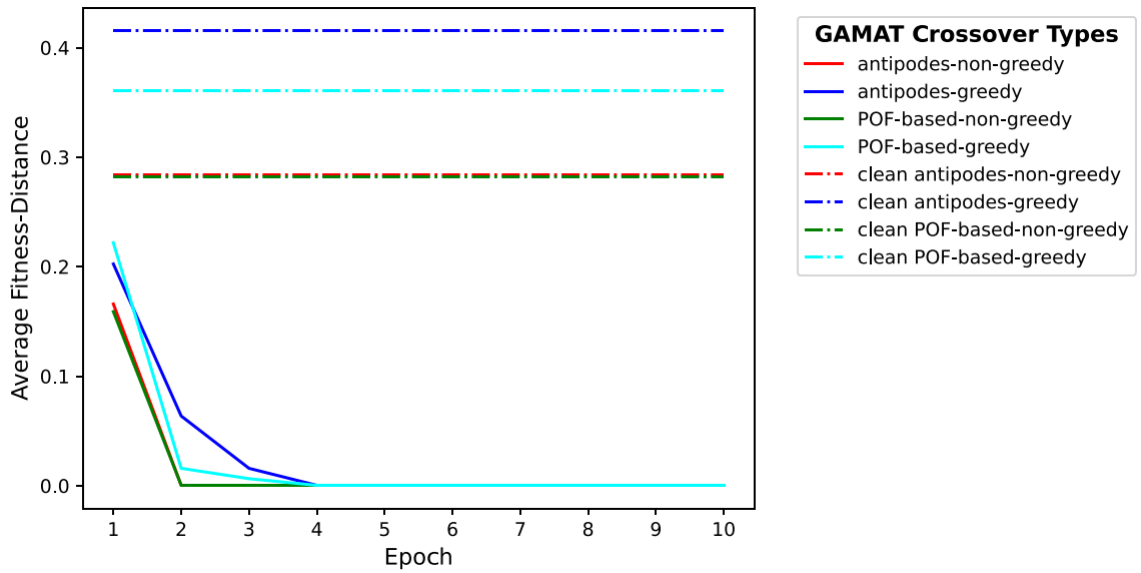

Рисунок 3.19 - Сценарій: дві жертви з двома хижаками на жертву - Перевірка придатності протягом епох

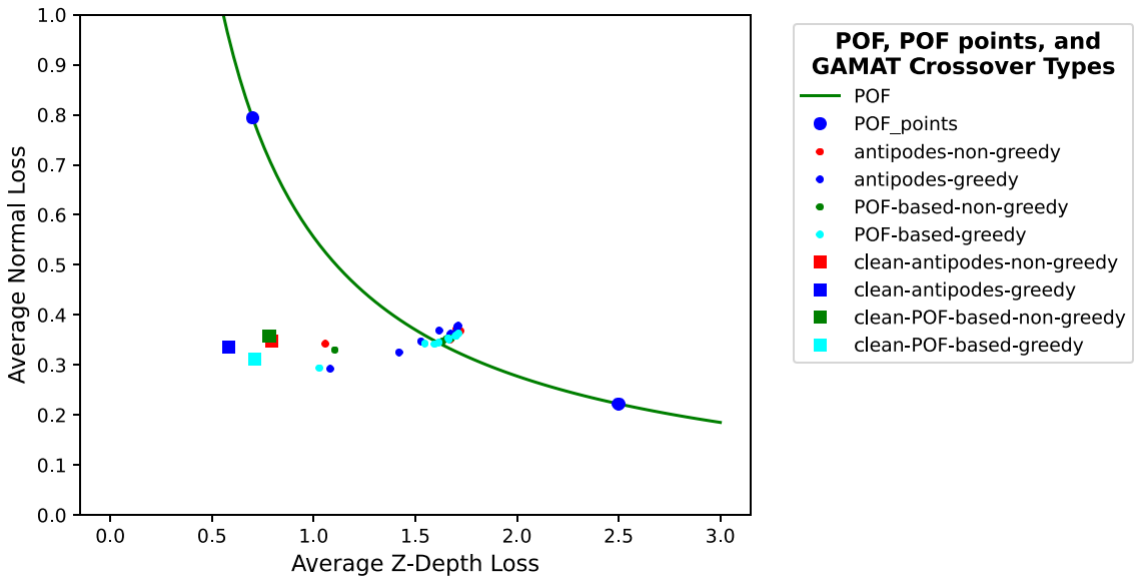

Рисунок 3.20 - Сценарій: дві жертви з двома хижаками на жертву - Точки втрат після висновку

На рис. 3.21 та 3.22 зображено сценарій навчання, де GAMAT використовує дві точки пошуку та п'ять хижаків на кожну точку пошуку. На рис. 3.21 показано, що всі моделі зійшлися лише після двох епох. Така висока швидкість збіжності вказує на те, що цей сценарій, зі збільшеною кількістю

хижаків на кожну здобич, забезпечує кращу продуктивність усіх варіантів алгоритму GAMAT. Як зазначалося раніше, стани моделей знову зберігаються. Однак швидка збіжність моделей піднімає питання для майбутньої роботи: навчання моделей на POF, далі від «чистої втрати» MT-DNN. Такий експеримент краще продемонстрував би ефективність алгоритму навчання. Знову ж таки, щоб краще візуалізувати еволюцію простору пошуку кожної моделі, читач може звернутися до рис. 3.22.

На рис. 3.23 та 3.24 зображено сценарій навчання, де GAMAT використовує п'ять точок здобичі та двох хижаків на кожну точку здобичі. Усі моделі сходяться. Однак модель, навчена GAMAT з використанням жадібного схрещування антиподів, сходиться до десятої епохи, тоді як інші моделі сходяться до моменту досягнення третьої епохи. Крива втрат валідації моделі на рис. 3.23 вказує на те, що модель або застрягла в локальному мінімумі, або на плато з нульовими градієнтами, доки не знайшла шлях до глобального мінімуму.

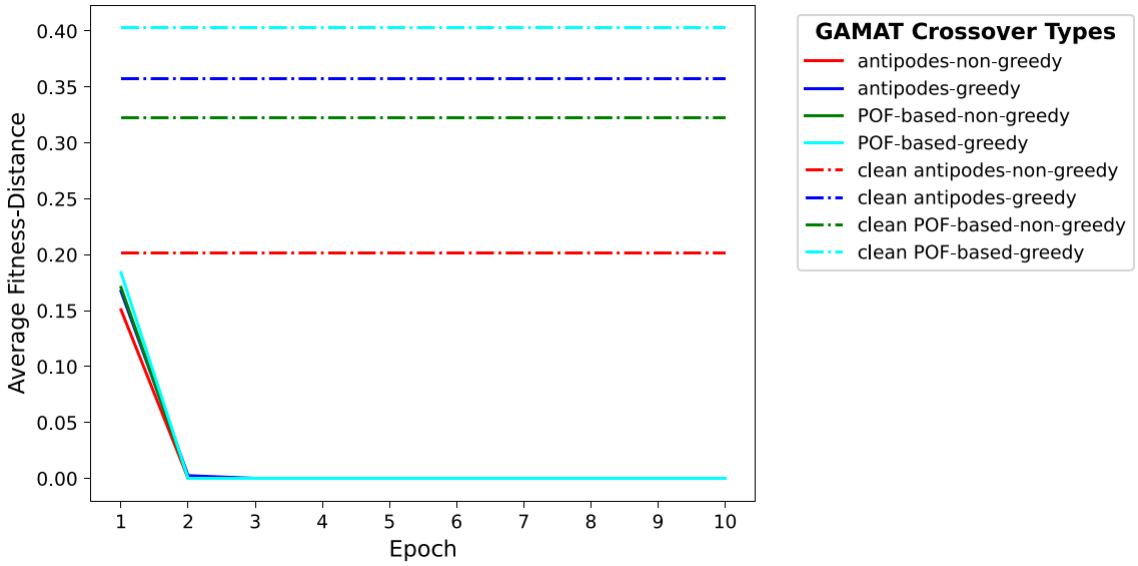

Рисунок 3.21 - Сценарій: дві жертви з п'ятьма хижаками на жертву - Перевірка придатності протягом епох

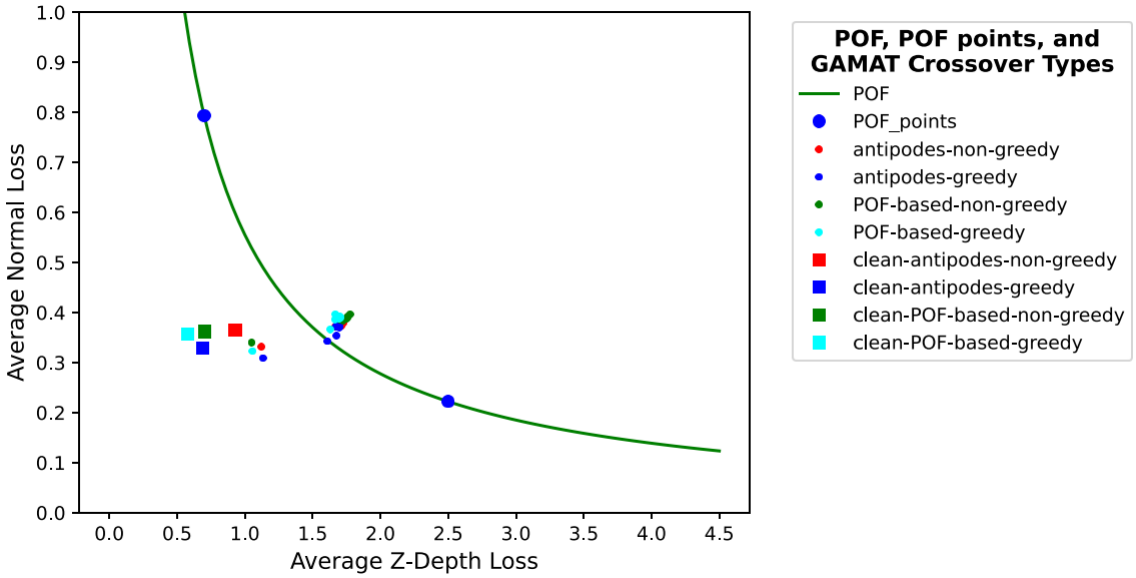

Рисунок 3.22 - Сценарій: дві жертви з п'ятьма хижаками на жертву - Точки втрат після висновку

Отже, сценарій з п'ятьма жертвами та двома хижаками на точку жертви видається дещо менш ефективним, ніж попередні два сценарії у випадку GAMAT з жадібною версією схрещування антиподів. Водночас інші моделі демонструють подібну швидкість конвергенції.

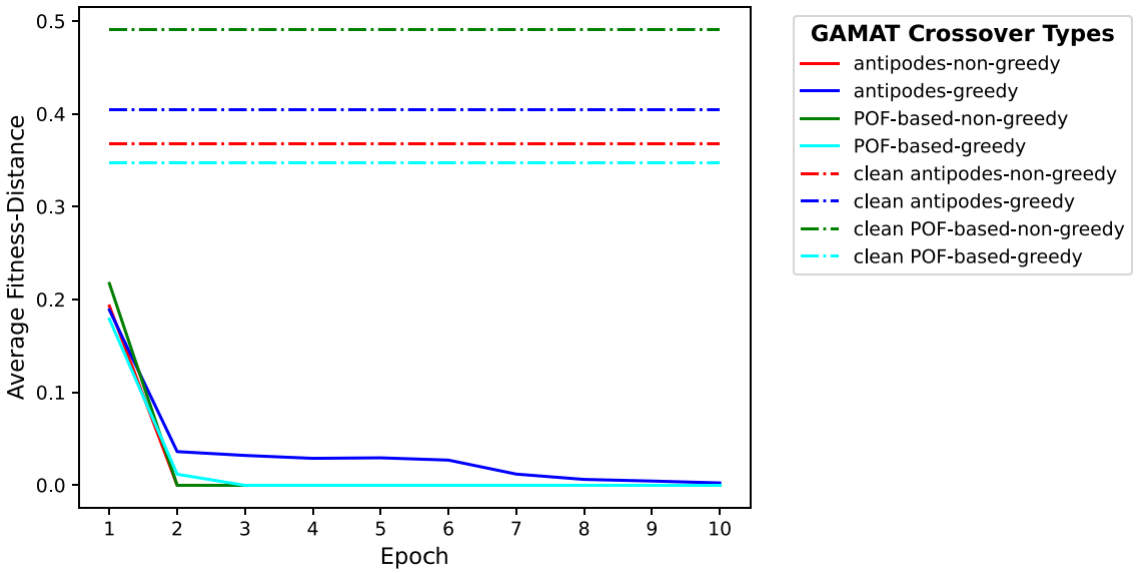

Рисунок 3.23 - Сценарій: п'ять жертв з двома хижаками на жертву - Перевірка придатності протягом епох

Зрештою, на рис. 3.25 та 3.26 зображено сценарій навчання, де GAMAT використовує п'ять точок жертви та двох хижаків на кожну точку жертви. Усі моделі, крім моделі, навченої GAMAT з нежадібним кросовером на основі POF, сходяться протягом десяти епох. Однак, навіть модель, навчена GAMAT з нежадібним кросовером на основі POF, демонструє потенціал до збіжності після десятої епохи через зменшення дистанції пристосованості після восьмої епохи, перед якою вона досягає плато в просторі втрат, де градієнт дорівнює нулю, або застрягає в локальному мінімумі, доки не знайде шлях до подальшого просування до глобального мінімуму. Тому, враховуючи більше епох для навчання, всі моделі можуть зійтися в такому сценарії. Однак, як і в попередньому сценарії, цей сценарій менш ефективний, ніж інші сценарії щодо моделі, навченої GAMAT з нежадібною версією кросовера на основі POF.

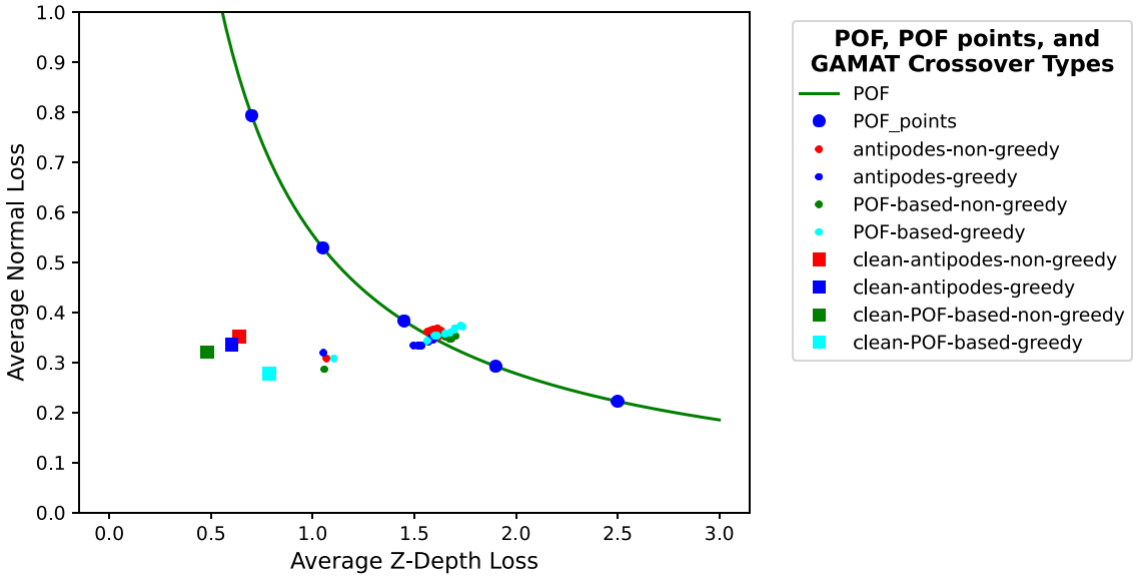

Рисунок 3.24 - Сценарій: п'ять жертв з двома хижаками на жертву - Точки втрат після висновку

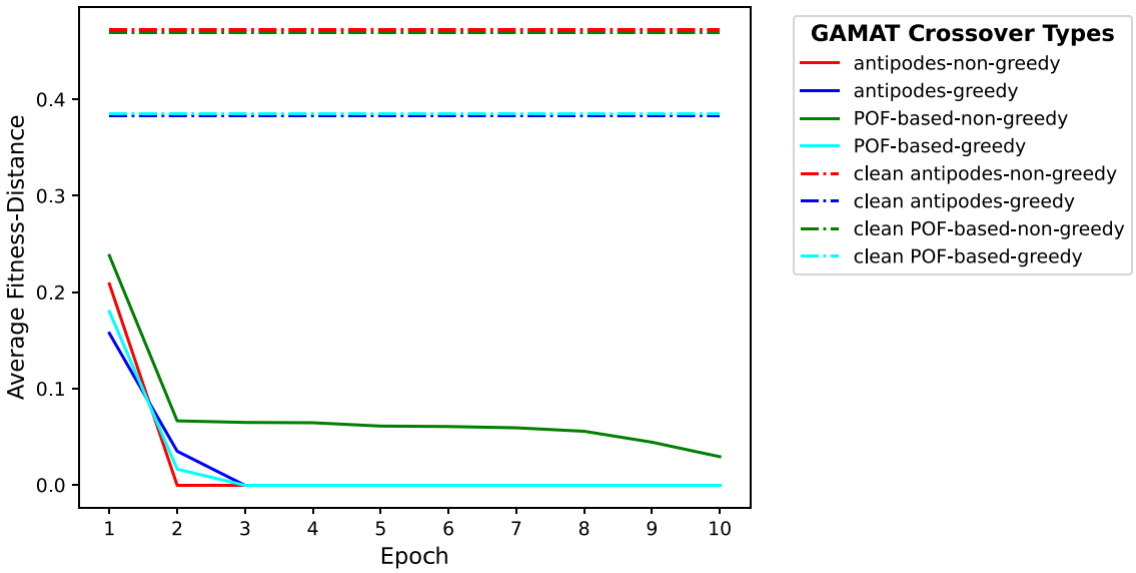

Рисунок 3.25 - Сценарій: три жертви з трьома хижаками на жертву - Перевірка придатності протягом епох

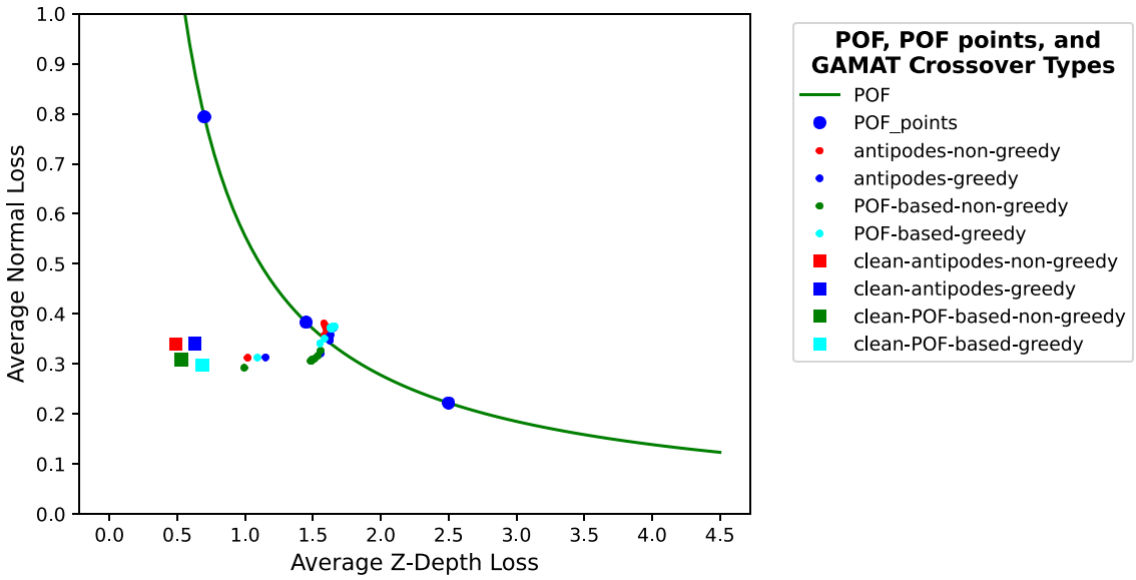

Рисунок 3.26 - Сценарій: три жертви з трьома хижаками на жертву - Точки втрат після висновку

3.4. Кросоверний аналіз

Щоб краще зрозуміти, за якого сценарію конкретні моделі працюють найкраще, у цій дисертації представлено ще один експеримент. На рис. 3.27 представлена модель, навчена за допомогою GAMAT з нежадібним схрещуванням на основі антиподів у всіх сценаріях, для яких модель була

навчена. Модель добре працює в усіх сценаріях, оскільки вона сходиться протягом двох епох у кожному сценарії. Тому GAMAT з нежадібним схрещуванням на основі антиподів варто продовжити в майбутній роботі.

На рис. 3.28 представлено модель, навчену за допомогою GAMAT з жадібним схрещуванням на основі антиподів у всіх сценаріях, для яких модель була навчена. Модель, навчена за допомогою GAMAT з жадібним схрещуванням на основі антиподів, сходиться у всіх сценаріях, але з дещо нижчою швидкістю збіжності, ніж моделі, навчені за допомогою GAMAT з нежадібним схрещуванням антиподів, що вказує на те, що GAMAT з жадібним схрещуванням на основі антиподів є менш ефективним, ніж GAMAT з нежадібним схрещуванням антиподів.

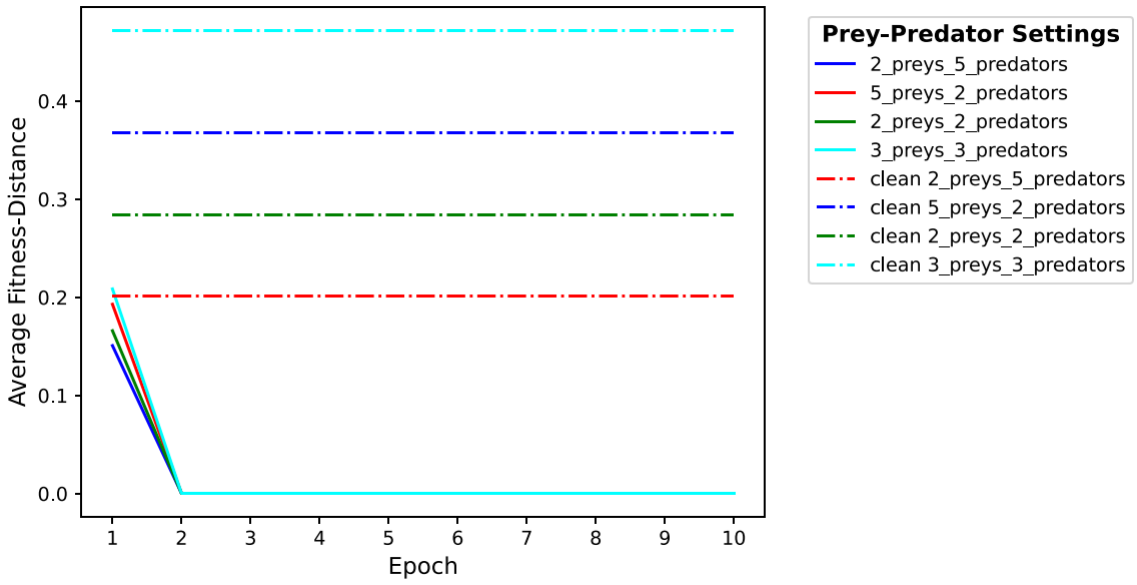

Рисунок 3.27 - Перевірка придатності моделі, навченої за допомогою GAMAT, з нежадібними антиподами - схрещування у всіх умовах взаємодії жертв та хижаків

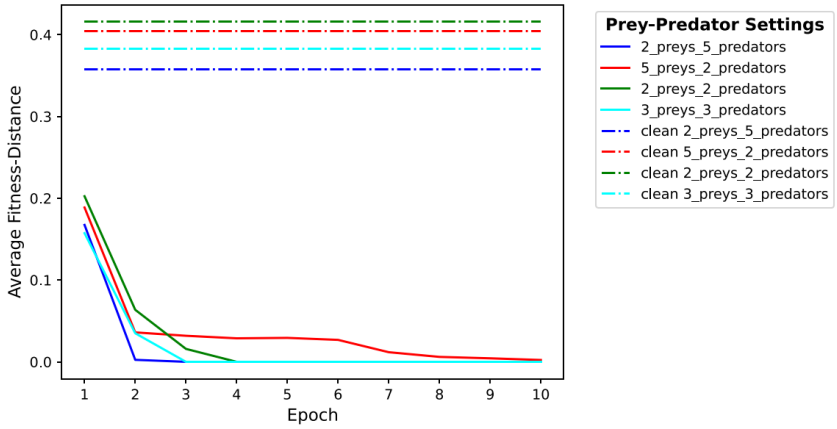

Рисунок 3.28 - FE валідація моделі, навченої за допомогою GAMAT, з використанням жадібних антиподів - схрещування у всіх умовах взаємодії жертви та хижаки.

На рис. 3.29 модель, навчену GAMAT з нежадібним кросовером на основі POF, розглядається в тих самих чотирьох сценаріях. Аналогічно, як і у випадку з GAMAT з жадібним кросовером антиподів, усі моделі, навчені GAMAT з нежадібним кросовером на основі POF, збігаються.протягом десяти епох, а одна. Модель, яка не зійшлася протягом заданих десяти епох, була навчена за сценарієм з трьома жертвами та трьома хижаками на жертву. Хоча модель має потенціал для збіжності після десятої епохи, можливо, не варто розглядати цю модель у майбутній роботі через низький коефіцієнт збіжності. Нарешті, модель, згенерована за допомогою

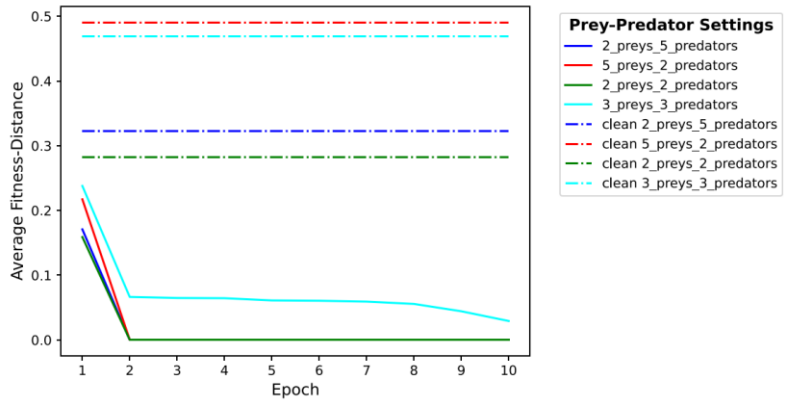

Рисунок 3.29 - Перевірка придатності моделі, навченої за допомогою GAMAT, з нежадібним схрещуванням на основі POF у всіх умовах співвідношення жертви та хижаки

GAMAT з жадібним кросовером на основі POF представлено на рис. 5.30 для тих самих чотирьох сценаріїв. Результати показують, що ця модель сходиться у всіх сценаріях протягом чотирьох епох, що все ще менш ефективно, ніж моделі, навчені GAMAT з нежадібним кросовером антиподів. Однак, моделі, навчені GAMAT з жадібним кросовером на основі POF, все ще варто розглянути в майбутній роботі.

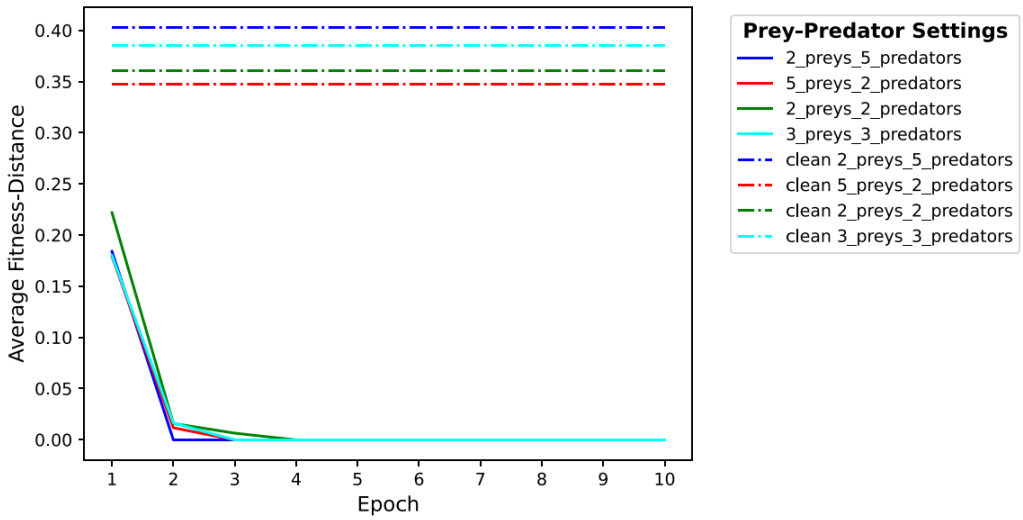

Рисунок 5.30 – FE валідація моделі, навченої за допомогою GAMAT, з жадібним кросовером на основі POF у всіх умовах співвідношення жертви та хижаки.

3.5. Гіперпараметричний аналіз

Наявність кількох моделей з лише однією відмінністю в процесі навчання, а також повільніша збіжність деяких моделей за інші, піднімає питання про те, як кожен метод кросовера впливає на процес навчання моделі. На жаль, на це питання важко відповісти, виходячи лише з чотирьох параметрів та однієї точки функціонування (POF). Тому це питання залишається ще одним викликом у майбутній роботі. З іншого боку, час, необхідний для навчання моделі, безпосередньо пов'язаний з кількістю точок жертви та кількістю хижаків на кожну точку жертви. Чим вищий добуток точок жертви та кількості хижаків на

кожну точку жертви, тим більше часу потрібно для завершення процесу навчання. Причиною взаємозв'язку між ними є етап оновлення на основі градієнта. Ваги моделі необхідно оновлювати за допомогою градієнтного градієнта після прямого та зворотного проходу для кожного хижака. Чим більша кількість хижаків, тим більша кількість прямих та зворотних проходів, що призводить до збільшення часу навчання. Однак час навчання не є проблемою в цьому випадку.

Ще одне питання, яке варто розглянути, полягає в тому, як співвідношення між кількістю точок жертви та кількістю хижаків на точку жертви впливає на процес навчання та валідації. Згідно з розділом 3.3, можна помітити різницю між моделями, навченими за сценарієм з двома жертвами та двома хижаками на жертву, та сценарієм з двома жертвами та п'ятьма хижаками на жертву. Різниця полягає в коефіцієнті конвергенції. Усі моделі, навчені за сценарієм з двома жертвами та п'ятьма хижаками на жертву, сходяться у другій епосі. Для порівняння, деякі моделі, навчені за сценарієм з двома жертвами та двома хижаками на жертву, не сходяться до четвертої епохи. Тим не менш, сценарій з двома жертвами та п'ятьма хижаками на жертву має таку ж перевагу над усіма іншими протестованими сценаріями. Отже, висновок полягає в тому, що більша кількість жертв не обов'язково призводить до кращого коефіцієнта конвергенції. Навпаки, кількість хижаків на жертву обернено пропорційна кількості епох, необхідних для конвергенції моделі. Чим більша кількість хижаків на жертву, тим менша кількість епох, необхідна для конвергенції.

Очікується швидка збіжність моделей через те, що розмір пакету встановлено на чотири. Навпаки, одна з моделей не збіглася під час фази навчання; найімовірнішою причиною є малий розмір пакету. Тому збільшення розміру пакету є справедливим врахуванням для майбутніх експериментів. Крім того, збільшення розміру пакету зменшить можливий шум в оцінках градієнтів та покращить узагальнення під час навчання. Однак недоліком більшого розміру пакету є довший час збіжності та потреба в більших ресурсах.

Зрештою, всі моделі в цьому експерименті навчаються на десяти епохах. Це число обрано з урахуванням невеликого розміру групи та очікування відносно

швидкої збіжності моделей. Однак, був випадок, коли модель застрягла в локальному мінімумі та зрештою знайшла вихід, наближаючись до кінця процесу навчання. Після цього крива навчання моделі продовжувала знижуватися, і вона не збігалася через брак епох навчання. Тому варто розглянути можливість навчання моделей на більш ніж десяти епохах.

3.6. Порівняння з найсучаснішими моделями

У цьому розділі наведено детальне порівняння запропонованої моделі атаки з кількома сучасними атаками здібностей, розробленими для багатозадачних нейронних мереж, такими як зважений градієнтний спуск (WGD) [21], атака проектованого градієнтного спуску (PGD) [24] та ітеративний проектований градієнтний спуск імпульсу (MI-PGD) [25]. По-перше, час виведення кожної атаки розраховується як середній час атаки для одного вхідного зображення протягом десяти ітерацій, де обчислення виконуються на відеокарті NVIDIA TITAN RTX. У таблиці 3.1 порівнюється час виведення атаки MAT та час виведення згаданих вище атак, виміряний у секундах, кількість кадрів за секунду, на які може вплинути кожна атака, та метрика придатності до кожної атаки з POF, обраного для експерименту. WGD, PGD та MI-PGD є ітеративними атаками; тому атака MAT порівнюється з кожною ітераційною атакою, де кількість кроків коливається від одного до двадцяти. Оскільки MAT-атака базується на навченій офлайн-моделі, для створення збурення потрібен лише один прямий прохід через навчений генератор. З іншого боку, сучасні атаки для багатозадачних нейронних мереж є ітеративними та вимагають зворотного поширення на кожному кроці атаки. Таким чином, MAT-атака значно перевершує сучасні атаки за часом виведення. Час, необхідний атаці для створення збуреного зображення, має бути якомога ближчим до «реального часу» (25-30 кадрів на секунду). Така продуктивність атаки дасть зловмиснику можливість впливати майже на кожен, якщо не на кожен, кадр вхідного відео в автономних машинах, що дає системі MT-DNN дуже мало або взагалі не дає часу

для виправлення неправильного рішення, прийнятого після обробки збуреного зображення. Як показано в таблиці 3.1, MAT-атака досягає 66,67 кадрів на секунду (FPS), тоді як інші атаки досягають 2,16 FPS у найкращому випадку.

Крім того, у таблиці 3.1 показано продуктивність кожної атаки, де кожна ітеративна атака має кількість кроків, необхідних для досягнення відповідної продуктивності. Порівняння атак на основі дистанції пристосованості для всіх експериментальних сценаріїв можна спостерігати з рис. 3.31 - 3.34. Зі збільшенням кількості кроків під час ітеративних атак середня відстань пристосованості зменшується, а час, необхідний для генерації збурення, збільшується, що зменшує кількість кадрів за секунду, на які впливає процес. Тому досягнення бажаної продуктивності відстані пристосованості є дороговартісним під час використання ітеративних атак. З іншого боку, бажана продуктивність відстані пристосованості для MAT-атаки не пов'язана з часом, необхідним для створення збурення, оскільки модель MAT-атаки навчається офлайн, і для створення збурення потрібен один прямий прохід через генератор. Таким чином, MAT-атака є швидкою та ефективною порівняно з сучасними атаками на системи MT-DNN, і вона є еталоном для створення більш надійних MT-DNN.

Таблиця 3.1 - Порівняння часу виконання (у секундах) для обчислення вхідних даних зловмисника, кількості кадрів за секунду (FPS) та середньої відстані пристосованості для найсучасніших атак

Adversarial Attack	Perturbation Time (s)	Frames Per Second (FPS)	Average Fitness Distance
MAT-BEST (10 epochs)	0.015	66.67	0.00
MAT-WORST (10 epochs)	0.015	66.67	0.04
WGD (1 step)	0.462	2.16	0.12
WGD (2 steps)	0.777	1.29	0.00
WGD (4 steps)	1.309	0.76	0.00
WGD (10 steps)	2.776	0.36	0.00
WGD (20 steps)	5.688	0.18	0.00
MI-PGD (1 step)	0.984	1.02	0.12
MI-PGD (2 steps)	1.813	0.55	0.00
MI-PGD (4 steps)	3.1996	0.31	0.00
MI-PGD (10 steps)	5.706	0.17	0.00
MI-PGD (20 steps)	11.919	0.08	0.00
PGD (1 step)	0.930	1.08	0.12
PGD (2 steps)	1.887	0.53	0.00
PGD (4 steps)	3.8698	0.26	0.00
PGD (10 steps)	8.367	0.12	0.00
PGD (20 steps)	19.657	0.05	0.00

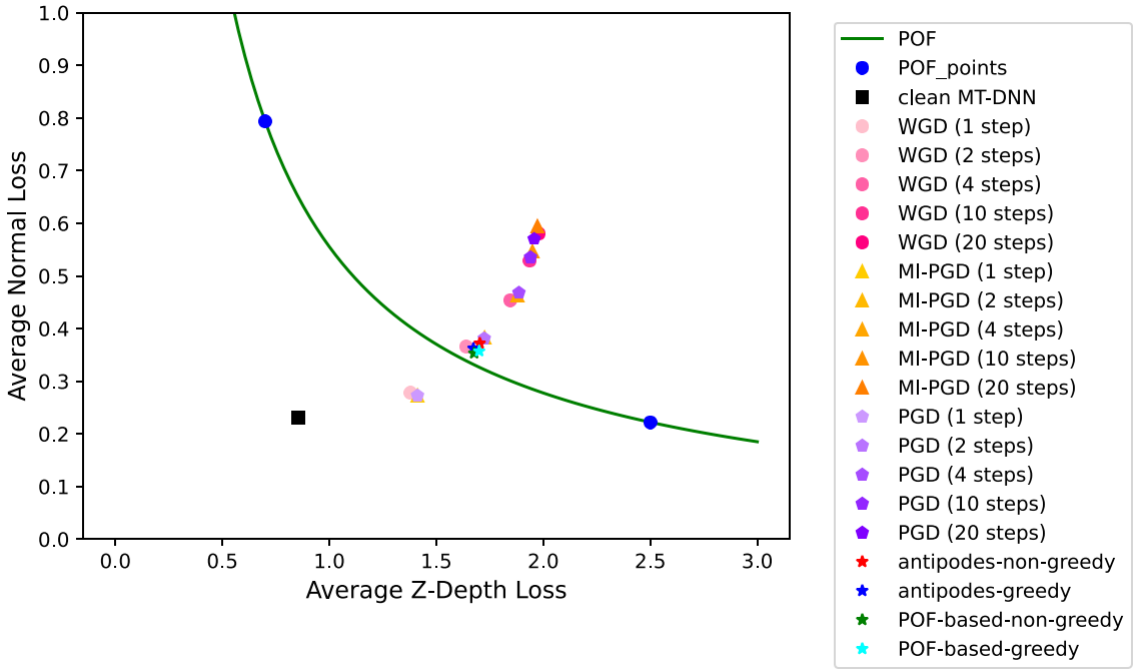

Рисунок 3.31 - Порівняння атаки MAT та сучасних атак у просторі втрат – сценарій з двома жертвами та двома хижаками на жертву

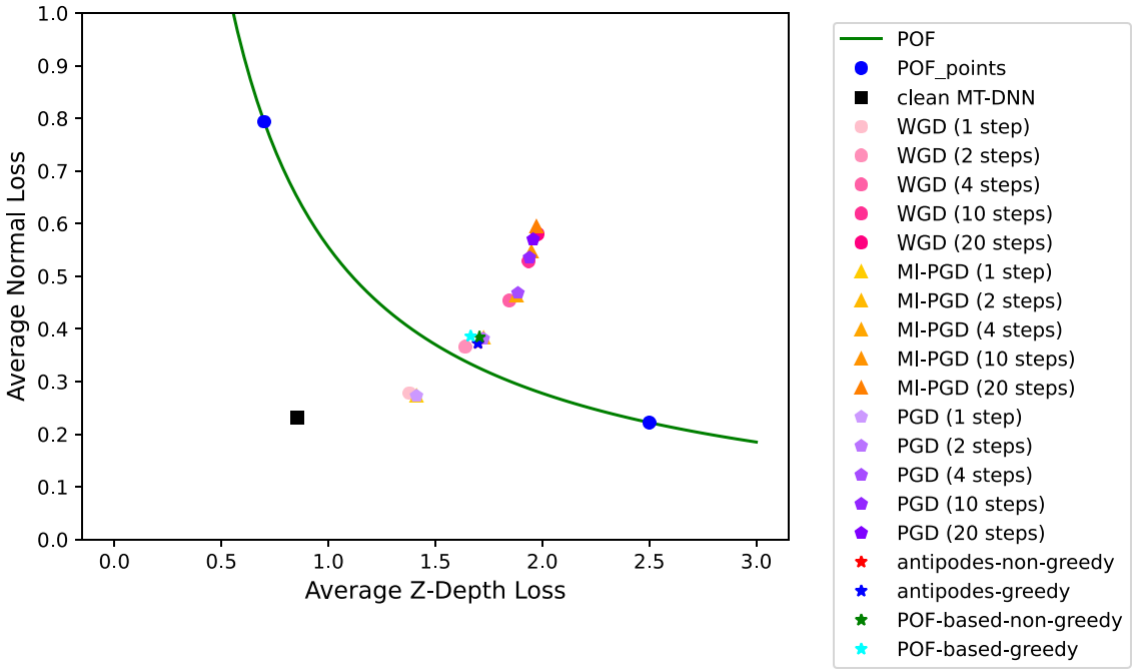

Рисунок 3.32 - Порівняння атаки MAT та сучасних атак у просторі втрат – сценарій з двома жертвами та п'ятьма хижаками на жертву

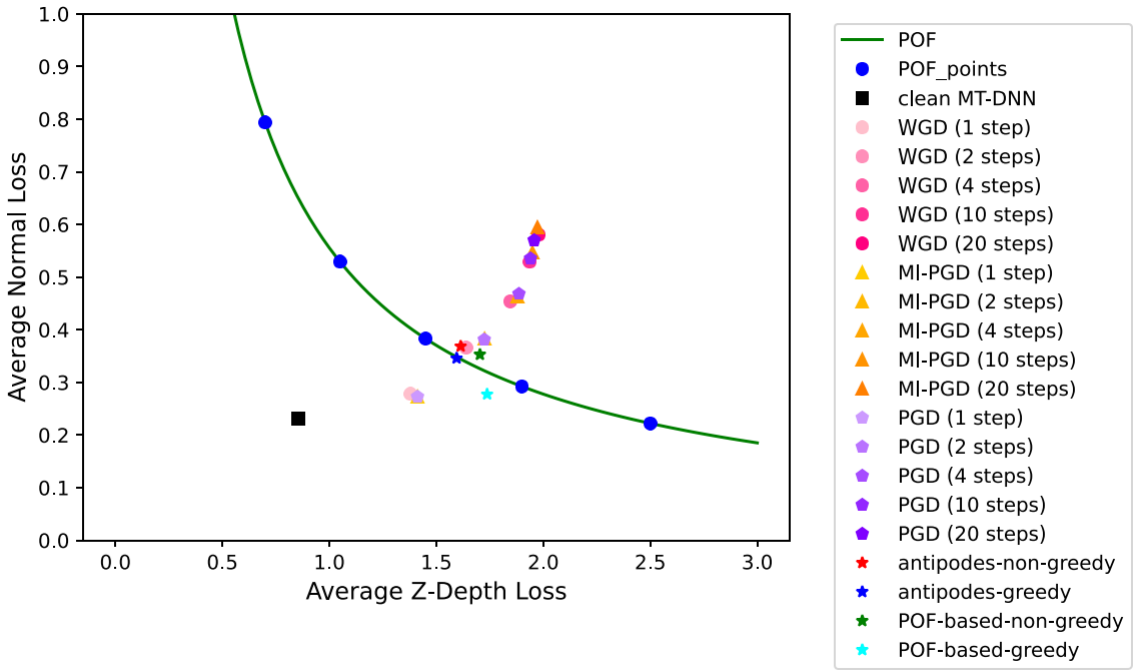

Рисунок 3.33 - Порівняння атаки MAT та сучасних атак у просторі втрат – сценарій з п’ятьма жертвами та двома хижаками на жертву

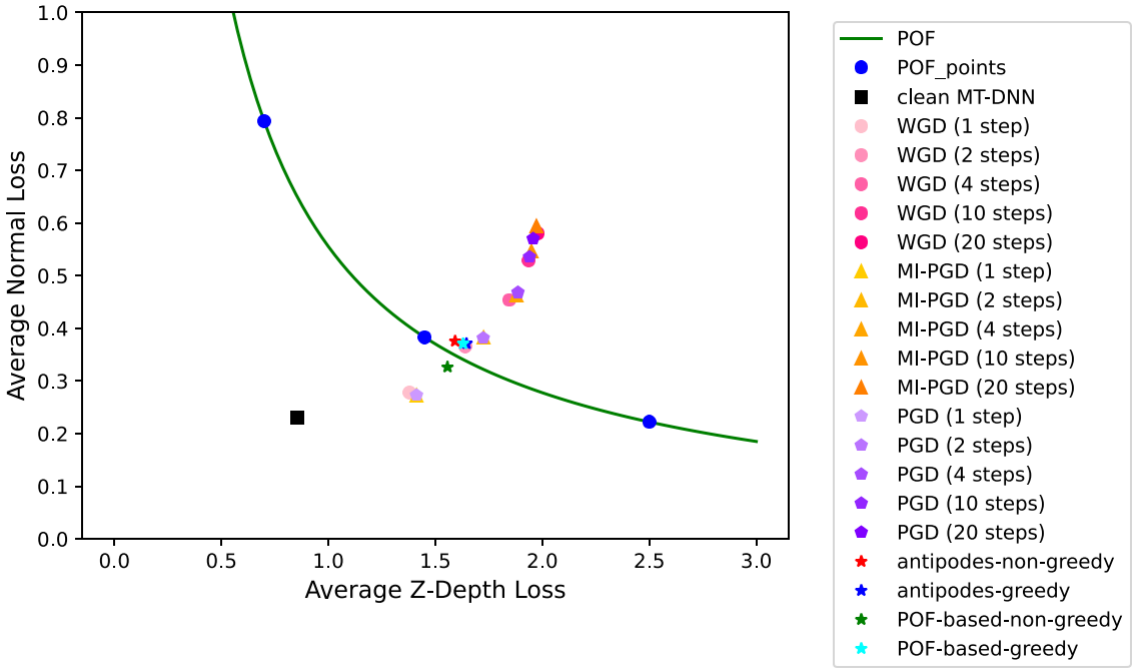

Рисунок 3.34 - Порівняння атаки MAT та сучасних атак у просторі втрат – сценарій з трьома жертвами та трьома хижаками на жертву

MAT-атака значно випереджає сучасні ітеративні атаки, такі як WGD, PGD та MI-PGD, у швидкості виконання. Завдяки навченій офлайн-моделі вона досягає 66,67 кадрів за секунду, тоді як інші методи працюють із продуктивністю 2,16 FPS у найкращому випадку. Ітеративні атаки потребують багато часу через зворотне поширення на кожному етапі, тоді як MAT-атака створює збурене зображення за один прямий прохід через генератор. Це робить її ефективною та небезпечно швидкою, що може впливати на кожен кадр вхідного відео, залишаючи системам MT-DNN мінімальний шанс на корекцію. Отже, MAT-атака є еталоном швидкості та ефективності, демонструючи необхідність розробки більш надійних захисних механізмів.

4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

4.1 Підвищення стійкості роботи промислових підприємств у воєнний час

Ефективність економіки держави залежить від того, наскільки окремі галузі господарства здатні стійко працювати не тільки у звичайних умовах, а й в умовах НС мирного та воєнного часу.

Значні руйнування та втрати серед населення, викликані наслідками НС, у даному випадку наслідку воєнного часу, можуть стати причиною різкого скорочення випуску промислової та сільськогосподарської продукції, а отже і зниження економічного потенціалу держави. Виникає потреба завчасного вживання заходів щодо забезпечення стійкої роботи промислових об’єктів на випадок виникнення НС.

Знання можливих НС, характерних для даної місцевості та виробництва, дозволяє диференційовано і цілеспрямовано розробляти та здійснювати заходи, які можуть запобігти аваріям, катастрофам та стихійним лихам або пом’якшити їх наслідки.

Стійкість роботи об’єкта господарської діяльності – це здатність його в умовах НС випускати продукцію у запланованому обсязі та визначеної номенклатури, а у разі слабких та середніх руйнувань або порушення матеріального постачання – відновлювати виробництво власними силами у короткий термін [67]. На стійкість роботи об’єкта впливають такі фактори:

• захищеність робітників та службовців від уражальних факторів у НС;

• здатність інженерно-технічного комплексу об’єкта (будівель, споруд,

• обладнання та комунально-енергетичних мереж) протистояти руйнівній дії уражальних факторів аварій, катастроф, стихійного лиха та сучасної зброї;

• надійність постачання об’єкта електроенергією, водою, паливом;

• комплектуючими та сировиною;

• підготовленість об’єкта до проведення аварійно-рятувальних та відновлюваних робіт;

• оперативність управління виробництвом та здійсненням заходів ЦЗ у НС.

Підвищення стійкості об’єкта досягають проведенням комплексу інженерно-технічних, технологічних, організаційних заходів. До інженерно-технічних заходів належать роботи, що забезпечують стійкість виробничих будівель і споруд, обладнання та комунально-енергетичних систем. Технологічні заходи забезпечують підвищення стійкості об’єкта спрощенням технологічного процесу виробництва кінцевої продукції та виключенням або обмеженням розвитку аварій. Організаційні заходи передбачають розробку ефективних дій керівного складу, служб та формувань ЦЗ, спрямованих на захист виробничого персоналу, проведення рятувальних та інших невідкладних робіт, а також відновлення виробництва. У останні роки спостерігається зростаюча увага до питань забезпечення екологічної безпеки та стійкості об'єктів господарської діяльності (ОГД) під час надзвичайних ситуацій. Важливість цих питань стає особливо актуальною в умовах складних виробничих процесів та високої взаємозалежності між галузями промисловості.

Забезпечення стійкості роботи ОГД під час воєнного часу має критичне значення для забезпечення безперебійного функціонування економіки та збереження виробничого потенціалу. Це особливо важливо для збереження виробничих ланцюжків та забезпечення необхідної продукції для цивільного населення та військових потреб. Стійкість роботи ОГД в умовах надзвичайних ситуацій визначається кількома факторами, серед яких надійність захисту персоналу, здатність технічних систем протистояти впливу вражаючих факторів, надійність постачання необхідних ресурсів, ефективне управління

виробництвом та готовність до відновлення порушеного виробництва. Розв'язання цих завдань передбачає комплексний підхід до організації виробництва та систематичну підготовку об'єктів господарської діяльності до можливих надзвичайних ситуацій.

Тільки такий підхід може забезпечити ефективний захист персоналу та надійність функціонування ОГД в умовах надзвичайних обставин. Важливою складовою загальної стійкості підприємства також є фінансова політика та захист працівників [68]. Для об’єктів, що не виробляють матеріальні цінності (транспорт, зв’язок), під стійкістю їх роботи розуміють здатність виконувати свої функції у НС. Для об’єктів економіки держави в цілому під стійкістю функціонування розуміють здатність забезпечити життєдіяльність країни, виробництво продукції (промислової, сільського господарської), роботу енергетики, транспорту, зв’язку у воєнний час та у будь яких надзвичайних ситуаціях.

Таким чином, стійкість об’єктів народного господарства – це здатність всього комплексу, тобто будівель, обладнання, складів, комунікацій, технологічне обладнання, транспорту протистояти руйнівним діям вражаючих факторів. Основні шляхи підвищення стійкості роботи промислових об’єктів у надзвичайних умовах мирного і воєнного часів:

• забезпечення надійного захисту робітників і службовців від ЗМУ (засобів масового ураження);

• захист виробничих фондів від вражаючих факторів ЗМУ, в тому числі і від вторинних;

• підвищення надійності і оперативності управлінням виробництвом і ЦЗ;

• забезпечення стійкості постачання підприємства електроенергією, газом, водою та інше;

• підготовка об’єкта до проведення відновлювальних робіт.

Підвищення стійкості роботи промислових підприємств в умовах НС мирного і воєнного часів досягається завчасним проведенням комплексу інженерно-технічних, технологічних і організаційних заходів. Інженерно-технічні заходи (ІТЗ) включають комплекс робіт по підвищенню міцності і надійності будинків, споруд комунально-енергетичних систем, матеріально-технічних запасів.

Технологічні заходи спрямовані на підвищення стійкості виробництва шляхом заміни існуючого технологічного режиму роботи на такий, що виключає можливість виникнення вторинних вражаючих факторів. Організаційні заходи передбачають розробку і планування дій в умовах НС керівного складу об’єкту, штабу, служб та невоєнізованих формувань ЦЗ по захисту робітників і службовців, проведення рятувальних робіт та відновлення порушеного виробництва. Підвищення стійкості об'єкта досягається посиленням найбільш слабких (вражаючих) елементів і ділянок об'єкта. Для цього на кожному ОГД завчасно на основі досліджень планують і проводять відповідні організаційні й інженерно- технічні заходи. Досягнення науки і техніки дозволяють реалізувати такі рішення, при яких підприємство буде стійке до впливу дуже значних надлишкових тисків, однак це пов'язано з великими витратами засобів і матеріалів і може бути виправдано лише при захисті унікальних, особливо важливих елементів об'єкта.

Заходи будуть економічно обґрунтовані, якщо вони максимально узгоджені із завданнями, які розв'язуються в мирний час для забезпечення безаварійної роботи, поліпшення умов праці, удосконалювання виробничого процесу. Підвищення характеристик міцності проводять, якщо: • окремі особливо важливі будинки і спорудження значно слабші за інші і їхню міцність доцільно довести до прийнятої для даного підприємства межі стійкості; • необхідно зберегти деякі важливі ділянки (цехи), які можуть самостійно функціонувати при виході з ладу інших і забезпечать випуск особливо цінної продукції.

При проектуванні і будівництві нових цехів підвищення стійкості може бути досягнуто застосуванням для несучих конструкцій високоміцних і легких матеріалів (легованих сталей, алюмінієвих сплавів). При будівництві і реконструкції промислових споруд необхідно використовувати стійкі матеріали, що збільшать стійкість споруди. Загалом, стійкість роботи підприємств є важливим фактором для економіки країни та забезпечення її існування та розвитку.

4.2. Заходи, що покращують умови праці оператора

Трудовим законодавством передбачено, що роботодавець зобов'язаний щорічно забезпечувати реалізацію заходів щодо поліпшення умов і охорони праці, в тому числі розроблених за результатами атестації робочих місць за умовами праці та оцінки рівнів професійних ризиків [69].

З метою реалізації цієї норми існує перелік щорічно реалізованих роботодавцем заходів щодо поліпшення умов і охорони праці та зниження рівнів професійних ризиків. У перелік включені наступні заходи:

• Проведення в установленому порядку робіт з атестації робочих місць за умовами праці, оцінці рівнів професійних ризиків.

• Реалізація заходів щодо поліпшення умов праці, в тому числі розроблених за результатами атестації робочих місць за умовами праці, та оцінки рівнів професійних ризиків.

• Впровадження систем (пристроїв) автоматичного та дистанційного керування і регулювання виробничим обладнанням, технологічними процесами, підйомними і транспортними пристроями.

• Придбання та монтаж засобів сигналізації про порушення нормального функціонування виробничого устаткування, засобів аварійної зупинки, а також пристроїв, що дозволяють виключити виникнення небезпечних

ситуацій при повному або частковому припиненні енергопостачання і подальшому його відновленні.

• Влаштування огорож елементів виробничого устаткування від впливу рухомих частин, а також розлітаються предметів, включаючи наявність фіксаторів, блокувань, герметизуючих та інших елементів.

• Пристрій нових і (або) модернізація наявних засобів колективного захисту працівників від впливу небезпечних і шкідливих виробничих факторів.

• Нанесення на виробниче обладнання, органи управління і контролю, елементи конструкцій, комунікацій і на інші об'єкти сигнальних кольорів і знаків безпеки.

• Впровадження систем автоматичного контролю рівнів небезпечних і шкідливих виробничих факторів на робочих місцях.

• Впровадження та (або) модернізація технічних пристроїв, що забезпечують захист працівників від ураження електричним струмом.

• Встановлення запобіжних, захисних і сигнальних пристроїв (пристосувань) з метою забезпечення безпечної експлуатації та аварійного захисту парових, водяних, газових, кислотних, лужних, расплавних та інших виробничих комунікацій, обладнання і споруд.

• Механізація і автоматизація технологічних операцій (процесів), пов'язаних із зберіганням, переміщенням (транспортуванням), заповненням і спорожненням пересувних та стаціонарних резервуарів (судин) з отруйними, агресивними, легкозаймистими і горючими рідинами, використовуваними у виробництві. Конкретний перелік заходів щодо поліпшення умов і охорони праці та зниження рівнів професійних ризиків визначає роботодавець виходячи із специфіки його діяльності [70]. Фінансування заходів щодо поліпшення умов і охорони праці роботодавцями (за винятком державних унітарних підприємств та федеральних установ) повинно здійснюватися у розмірі не менше 0,2 відсотка суми витрат на виробництво продукції (робіт,

послуг). Працівник не несе витрат на фінансування заходів щодо поліпшення умов і охорони праці.

4.3 Значення автоматизації виробничих процесів в питаннях охорони праці.

Процес автоматизації виробництва є одним із головних напрямів технічного прогресу вже пів сторіччя. У зв'язку з розвитком автоматики з'явилася можливість звільнити людину від безпосередньої участі у виробничому процесі, що покращує умови її праці, дало час для розвитку та вирішення інших завдань. Через процес автоматизації машини не лише замінюють фізичну працю людини, а й виконують функції управління виробництвом. На рисунку 4.1 показано 70 промисловий робот як приклад автоматизації. При розробці цих роботів, що застосовуються в автоматизованих системах, та засобів захисту працівників та обслуговуючого персоналу повинні враховуватися специфічні властивості залучених у виробничий процес промислових роботів, пов'язані з особливостями конструкції, виконуваних функцій, динаміки та алгоритмів управління переміщенням робочих органів. Якщо ж технологія буде зроблена неправильно, то її корисність у плані безпеки буде дорівнювати нулю і така технологія буде завдавати більше шкоди, ніж користі. Самі засоби захисту повинні бути розроблені з урахуванням необхідності знаходження обслуговуючого персоналу в робочому просторі промислових роботів. Слід врахувати, що обслуговуючий персонал бере участь у включенні, програмуванні, а також обслуговуванні та контролі промислових роботів та автоматизованого виробництва загалом.

Рисунок 4.1 – Промисловий робот

При цьому процеси отримання, перетворення, передачі та використання енергії, матеріалів та інформації відбуваються в автоматичному режимі. В автоматизованому виробництві обслуговуючий персонал займається налагодженням механізмів та систем управління. Автоматизація виробництва підготовлена всім попереднім розвитком науки, техніки, технології та є закономірним продовженням механізації виробничих процесів. Водночас автоматизація – це якісно новий етап розвитку виробництва.

Внаслідок автоматизації збільшується продуктивність обладнання, знижується собівартість, скорочується брак виробів та підвищується безпека праці, покращується санітарний стан виробничих підрозділів тощо. Розвиток автоматизованих систем впливає на технічний прогрес, бо наразі всі питання створення нової техніки вирішуються комплексно. Автоматичне управління широко застосовується на складних виробництвах.

Автоматичні системи управління виробництвом здійснюють безперервний контроль і точне регулювання параметрів процесу, таких як швидкість руху стрічкового конвеєра, що забезпечує високу якість продукції, що випускається. Не менш важливою є цифровізація при покращенні процесів [71].

ВИСНОВКИ

У роботі запропоновано нову цільову багатоцільову змагальну модель атаки під назвою MAT та новий алгоритм навчання під назвою GAMAT, заснований на генетичних алгоритмах, для створення змагальних зображень, здатних цілеспрямовано впливати на MT-DNN. На основі симуляційних експериментів, розроблених з використанням набору даних Taskonomy, продемонстровано перевагу запропонованої атаки з точки зору метрики відстані пристосованості та кількості кадрів за секунду, на які впливає атака. Крім того, алгоритм навчання GAMAT має потенціал стати дуже ефективним алгоритмом навчання. Тому майбутня робота включатиме навчання та тестування варіантів GAMAT на більших наборах даних та більш віддалених POF, а також розробку масштабованих варіацій, які можуть краще використовувати переваги графічних процесорів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Caruana, R. Multitask learning. Machine learning, 1997, vol. 28, no. 1, pp. 41–75.

2. Figueroa, H., Wang, Y., Giakos, G.C. Adversarial attacks in industrial control cyber physical systems. IEEE International Conference on Imaging Systems and Techniques (IST), 2022, pp. 1–6. DOI: 10.1109/IST55454.2022.9827763.

3. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C. SSD: Single shot multibox detector. European conference on computer vision. Springer, 2016, pp. 21–37.

4. Girshick, R., Donahue, J., Darrell, T., Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation, 2014.

5. Girshick, R. Fast R-CNN. Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

6. Badrinarayanan, V., Kendall, A., Cipolla, R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 2017, vol. 39, no. 12, pp. 2481–2495.

7. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE transactions on pattern analysis and machine intelligence, 2017, vol. 40, no. 4, pp. 834–848.

8. Long, J., Shelhamer, E., Darrell, T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

9. Yu, F., Koltun, V. Multi-scale context aggregation by dilated convolutions. arXiv preprint, arXiv:1511.07122, 2015.

10. Simonyan, K., Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint, arXiv:1409.1556, 2014.

11. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

12. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818–2826.

13. He, K., Zhang, X., Ren, S., Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

14. Ruder, S. An overview of multi-task learning in deep neural networks. arXiv preprint, arXiv:1706.05098, 2017.

15. Zamir, A.R., Sax, A., Shen, W., Guibas, L.J., Malik, J., Savarese, S. Taskonomy: Disentangling task transfer learning. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3712–3722.

16. Standley, T., Zamir, A., Chen, D., Guibas, L., Malik, J., Savarese, S. Which tasks should be learned together in multi-task learning? International Conference on Machine Learning. PMLR, 2020, pp. 9120–9132.

17. Zhang, Y., Yang, Q. A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering, 2021.

18. Vandenhende, S., Georgoulis, S., De Brabandere, B., Van Gool, L. Branched multi-task networks: deciding what layers to share. arXiv preprint, arXiv:1904.02920, 2019.

19. Mao, C., Gupta, A., Nitin, V., Ray, B., Song, S., Yang, J., Vondrick, C. Multitask learning strengthens adversarial robustness. European Conference on Computer Vision. Springer, 2020, pp. 158–174.

20. Ghamizi, S., Cordy, M., Papadakis, M., Le Traon, Y. Adversarial robustness in multi-task learning: promises and illusions. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, vol. 36, pp. 697–705.

21. Goodfellow, I.J., Shlens, J., Szegedy, C. Explaining and harnessing adversarial examples. arXiv preprint, arXiv:1412.6572, 2014.

22. Kurakin, A., Goodfellow, I., Bengio, S. Adversarial machine learning at scale. arXiv preprint, arXiv:1611.01236, 2016.

23. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A. Towards deep learning models resistant to adversarial attacks. arXiv preprint, arXiv:1706.06083, 2017.

24. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J. Boosting adversarial attacks with momentum. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 9185–9193.

25. Athalye, A., Carlini, N., Wagner, D. Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. International conference on machine learning. PMLR, 2018, pp. 274–283.

26. Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B., Swami, A. Practical black-box attacks against machine learning. Proceedings of the 2017 ACM on Asia conference on computer and communications security, 2017, pp. 506–519.

27. Croce, F., Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. International conference on machine learning. PMLR, 2020, pp. 2206–2216.

28. Poursaeed, O., Katsman, I., Gao, B., Belongie, S. Generative adversarial perturbations. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4422–4431.

29. Deb, K. Multi-objective optimization using evolutionary algorithms. Wiley-Interscience, 2001.

30. Ghamizi, S., Cordy, M., Papadakis, M., Le Traon, Y. Adversarial robustness in multi-task learning: promises and illusions. AAAI Conference on Artificial Intelligence, 2022, pp. 697–705.

31. Deb, K. Multi-objective optimization using evolutionary algorithms. Wiley-Interscience, 2001.

32. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002, vol. 6, no. 2, pp. 182–197. DOI: 10.1109/4235.996017.

33. Gitman, I., Lang, H., Zhang, P., Xiao, L. Understanding the role of momentum in stochastic gradient methods. arXiv preprint, arXiv:1904.00000, 2019.

34. Johnson, J., Alahi, A., Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. arXiv preprint, arXiv:1603.08155, 2016.

35. Zhu, J.-Y., Park, T., Isola, P., Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2223–2232.

36. Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. MIT Press, 2016. URL: http://www.deeplearningbook.org..

37. Ruder, S. An overview of gradient descent optimization algorithms. arXiv preprint, arXiv:1609.04747, 2016.

38. Prechelt, L. Early stopping—but when? Neural Networks: Tricks of the Trade. Springer, 2002, pp. 55–69.

39. Holland, J.H. Adaptation in natural and artificial systems. University of Michigan Press, 1975.

40. Goldberg, D.E. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, 1989.

41. Kim, D.-W., Kim, J.-H. A novel initialization method for genetic algorithms using K-means clustering. Expert Systems with Applications, 2011, vol. 38, no. 5, pp. 6241–6247.

42. Whitley, D. A genetic algorithm tutorial. Statistics and Computing, 1994, vol. 4, no. 2, pp. 65–85.

43. Deb, K. Multi-objective optimization using evolutionary algorithms. John Wiley & Sons, 2001.

44. Sivanandam, S.N., Deepa, S.N. Introduction to genetic algorithms. Springer Science & Business Media, 2008.

45. Dinu, C.M., Dinu, N., Dumitrescu, D. Hybrid rank-based and tournament selection genetic algorithm for constrained optimization. International Conference on Numerical Analysis and Approximation Theory, Springer, 2018, pp. 122–127.

46. Deb, K. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evolutionary Computation, 2001, vol. 9, no. 3, pp. 257–280.

47. Fonseca, C.M., Fleming, P.J. Multi-objective genetic algorithms. International Conference on Evolutionary Multi-Criterion Optimization, 1995, pp. 105–124.

48. Rechenberg, I. Evolution strategy: Optimization technique from ICLR and parallel search. ICLR, 1973.

49. Goldberg, D.E. Alleles, loci, and the traveling salesman problem. Proceedings of the First International Conference on Genetic Algorithms and Their Applications, 1985, pp. 154–159.

50. Grefenstette, J.J., Baker, J.E. Genetic algorithms for the traveling salesman problem. Handbook of Combinatorial Optimization, 1989, vol. 3, pp. 315–344.

51. Purshouse, R.C., Fleming, P.J. An evolutionary approach to the linearisation of nonlinear systems. International Journal of Control, 2003, vol. 76, no. 1, pp. 48–62.

52. Mitchell, M. An introduction to genetic algorithms. MIT Press, 1998.

53. Fogel, D.B., Hays, T.J., Hahn, S. Evolving artificial intelligence. Morgan Kaufmann, 1999.

54. Lopes, H.S., de Carvalho, A.C.P.L., Lorena, A.C. Multi-objective evolutionary algorithms for feature selection in data mining: A survey. Proceedings

of the Eighth International Conference on Machine Learning and Applications, IEEE, 2009, pp. 918–923.

55. Duan, R., Zhang, W., Han, M. An adaptive mutation operator for evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 2012, vol. 16, no. 1, pp. 37–50.

56. Do, H.D., Li, X., Yang, S., Zhu, W., Li, J., Yang, J. Fuzzy logic-based adaptive mutation operator in differential evolution. Information Sciences, 2013, vol. 238, pp. 272–287.

57. Gao, Y., Cao, X., Lin, Y. A hybrid genetic algorithm with adaptive mutation strategy for feature selection in breast cancer diagnosis. Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2019, pp. 2608–2613.

58. Rahman, M.M., Hossain, M.S., Alam, M.R., Haque, M.R. Multi-objective genetic algorithm with adaptive mutation and crossover operators for optimizing the parameters of a solar water heater system. Applied Energy, 2018, vol. 223, pp. 312–328.

59. Harik, G.R. Learning real heuristics in the traveling salesman problem. Proceedings of the 1999 Congress on Evolutionary Computation, IEEE, 1999, vol. 1, pp. 318–325.

60. Eiben, A.E., Smith, J.E. Introduction to evolutionary computing. Springer, 2015.

61. Mühlenbein, H., Schlierkamp-Voosen, D. Parallel genetic algorithms, population genetics, and combinatorial optimization. Lecture Notes in Computer Science, 1991, vol. 496, pp. 416–425.

62. Goldberg, D.E., Deb, K. A comparison of selection schemes used in genetic algorithms. Foundations of Genetic Algorithms, 1991, vol. 1, no. 1, pp. 69–93.

63. Jin, Y. A multi-objective evolutionary algorithm using Gaussian process-based inverse modeling. IEEE Transactions on Evolutionary Computation, 2005, vol. 9, no. 4, pp. 365–385.

64. Huang, J., Li, C., Li, W. A hybrid genetic algorithm for the redundancy allocation problem. Journal of Mechanical Science and Technology, 2015, vol. 29, no. 11, pp. 4785–4792.

65. Rios, L.M., Sahinidis, N.V. Sparse Gaussian process regression for calibration of computer models. Engineering Optimization, 2013, vol. 45, no. 5, pp. 529–552.

66. Goldberg, D.E. Genetic algorithms with sharing for multimodal function optimization. Innovation in Evolutionary Algorithm, 1987, vol. 1, pp. 41–53.

67. Стефанович, Іван Станіславович, Павло Іванович Стефанович, and Тетяна Володимирівна Курбанова. "ПІДВИЩЕННЯ СТІЙКОСТІ ОБ'ЄКТІВ ГОСПОДАРЮВАННЯ." The 6th International scientific and practical conference "Modern directions of scientific research development"(November 24-26, 2021) BoScience Publisher, Chicago, USA. 2021. 1153 p. 2021.

68. Захист цивільного населення під час війни. URL: https://wiki.legalaid.gov.ua/index.php/Захист_цивільного_населення_під_час_війни

69. Про охорону праці. URL: https://zakon.rada.gov.ua/laws/show/2694-12

70. Дідур, Катерина. "Економічна ефективність впровадження заходів з охорони праці." (2020).

71. Охорона праці в галузі комп'ютингу. URL: https://library.kre.dp.ua/Books/2-4%20kurs/Охорона%20праці%20в%20галузі%205.05010201/OP12.pdf

ДОДАТКИ

Додаток А

Публікація 2 тез на конференції «Фундаментальні та прикладні проблеми сучасних технологій»

Міністерство освіти і науки України
Тернопільська обласна військова адміністрація
Тернопільський національний технічний університет імені Івана Пулюя
Фізико-механічний інститут ім. Г.В. Карпенка НАН України
Інститут прикладних проблем механіки і математики
ім. Я. С. Підстригача НАН України
Інститут електродинаміки НАН України
Інститут фізики напівпровідників ім. В.Є. Лашкарьова НАН України
Наукове товариство імені Шевченка
Vilnius Gediminas Technical University (Литва)
Warsaw University of Technology Lighting Technology Division (Польща)
Iskenderun Technical University (Туреччина);
Lublin University of Technology (Польща)
Technical University of Košice (Словаччина)
Асоціація випускників ТНТУ

МАТЕРІАЛИ

Міжнародної науково-технічної конференції

«ФУНДАМЕНТАЛЬНІ ТА ПРИКЛАДНІ ПРОБЛЕМИ СУЧАСНИХ ТЕХНОЛОГІЙ»

присвячена

180-річчю з дня народження Івана Пулюя та 65-річчю з дня заснування Тернопільського національного технічного університету імені Івана Пулюя

28-29 травня 2025 року
Тернопіль

2. Бойко І.В., Петрик М.Р., Цуприк Г.Б. Моделювання та видобуток даних (високопродуктивні обчислення у великих та числових системах, комбінаторному аналізі): навчальний посібник. Тернопіль: : ТНТУ 2019 – 62 с.

3. Information System for Design of Thin Multilayer Film Processes Parameters Management based on Diffusion M.R., Petryk, Mykhaylo R., V., Chyzh, Vitalii, H.B., Tsupryk, H. B., O.Y., Petryk, Oksana Yu ITTAP'2024: 4th International Workshop on Information Technologies: Theoretical and Applied Problems, October 23- 25, 2024, Ternopil, Ukraine, Opole, Poland, 2024, pp. 486-493 (https://ceur-ws.org/Vol-3896/)

4. Methods of constructing algorithms for comparative test statistical verification of mathematical models of bioobject responses to low-intensity stimuli / Bohdan Yavorskyy, Evhenia Yavorska, Halyna Tsupryk, Roman Kinash // Scientific Journal of TNTU. — Tern.: TNTU, 2023. — Vol 112. — No 4. — P. 82–90.

УДК 004.8

Р.З. Золотий, к.т.н., доцент, Д.А. Коломийчук
Тернопільський національний технічний університет імені Івана Пулюя

ЕФЕКТИВНІСТЬ АТАК МЕРЕЖ МАШИННОГО НАВЧАННЯ

R. Zolotyi, Ph.D., D. Kolomyichuk
Ternopil Ivan Puluj National Technical University

EFFECTIVENESS OF MACHINE LEARNING NETWORK ATTACKS

Було проведено огляд атак змагального навчання, зосереджений, зокрема, на захисті від атак на класифікатори глибоких нейронних мереж. Після ознайомлення з відповідною термінологією, цілями та діапазоном можливих знань як зловмисників, так і захисників, було розглянуто роботи з ухилення під час тестування (TTE), отруєння даних (DP), бекдор-DP та атак зворотного компілювання (RE), а також, зокрема, захист від них. При цьому розрізняли робустну класифікацію від виявлення аномалій (AD), ненавчений від контрольованого, а також захист на основі статистичних гіпотез від тих, що не мають явної нульової гіпотези (відсутність атаки).

Для побудови ефективного захисту необхідно визначити гіперпараметри, які вимагає певний метод, його обчислювальну складність, а також показники продуктивності, за якими оцінюється якість.

Також можна для аналізу атак застосовувати інноваційні методи: 1) робустна класифікація при визначенні аномалій, як стратегія захисту; 2) переконання, що успіх атаки зростає з силою атаки, яка ігнорує вразливість до аномалій; 3) невеликі збурення для атак ухилення від тестування: помилка чи вимога?; 4) обґрунтованість універсального припущення, що зловмисник TTE знає клас істинності для прикладу, що атакується; 5) атаки чорної, сірої або білої коробки як стандарт для оцінки захисту; 6) вразливість RE на основі запитів до захисту AD. Ми також обговорюємо атаки на конфіденційність навчальних даних.

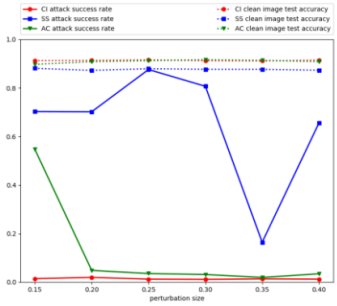

Рисунок 1 - Коефіцієнт успішності та точність атаки на чистому тестовому наборі перенавчених нейронних мереж для трьох захистів

Література.

1. Міллер Д. Дж., Сян Чж., Кесідіс Г. Adversarial Learning in Statistical Classification: A Comprehensive Review of Defenses Against Attacks // Proceedings of the IEEE. 2020. URL: https://arxiv.org/abs/1904.06292.

УДК 004.41

Д.В. Коваль, Г.Б. Цуприк, к.т.н., доц..
Тернопільський національний технічний університет імені Івана Пулюя, Україна

МОДУЛЬНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ ФІНАНСОВОГО КОНТРОЛЮ НА ANDROID ІЗ ЗАСТОСУВАННЯМ ПРИНЦИПІВ ІНВЕРСІЇ УПРАВЛІННЯ ТА ШАБЛОНУ MVVM

D.V. Koval, H.B. Tsupryk, Ph.D., Assoc. Prof.

MODULAR IMPLEMENTATION OF FINANCIAL CONTROL FUNCTIONALITY ON ANDROID USING INVERSION OF CONTROL PRINCIPLES AND THE MVVM PATTERN

Мобільні застосунки для фінансового контролю набули значної популярності серед користувачів, оскільки вони допомагають ефективно управляти особистими фінансами. Такі програми дозволяють стежити за доходами та витратами, створювати бюджети, аналізувати фінансові звіти та досягати фінансових цілей. Однак для забезпечення надійної та ефективної роботи таких застосунків важливо правильно організувати їх внутрішню структуру, щоб забезпечити гнучкість, масштабованість і легкість у підтримці.

Одним з основних підходів до розробки мобільних застосунків є використання модульної архітектури, що передбачає поділ програми на окремі компоненти. Кожен з цих компонентів має свою чітко визначену функцію і може бути розроблений, тестований і оновлений незалежно від інших частин програми. Наприклад, в фінансовому застосунку один модуль може відповідати за облік доходів і витрат, інший — за категоризацію витрат, а ще один — за аналіз бюджету чи формування фінансових цілей.

Такий підхід до структуризації дає можливість працювати з кожною частиною програми окремо, що значно спрощує роботу розробників. Модульність також дає можливість розподіляти завдання між кількома розробниками, зменшуючи час розробки і підвищуючи ефективність роботи над проєктом. Крім того, модульність дозволяє простіше додавати нові функції в застосунок без потреби переписувати вже готовий код, що робить програму більш гнучкою і стійкою до змін.

4. Valstar M. et al. Affect recognition in HCI. In Human-Centered AI. Springer, 2020.
5. Huang J. et al. Real-time AI-enhanced UX analytics. IJHCS, 2023.

УДК 004.8
Г.І. Липак, к.н.с.к., доцент, Д.А. Коломийчук
Тернопільський національний технічний університет імені Івана Пулюя

ВПЛИВ БЕКДОР АТАК НА МОЖЛИВОСТІ НАВЧАННЯ НЕРОННИХ МЕРЕЖ

H. Lypak, Ph.D., D. Kolomyichuk
Ternopil Ivan Puluj National Technical University

THE IMPACT OF BACKDOR ATTACKS ON THE LEARNING CAPABILITY OF NEURAL NETWORKS

У роботі було проаналізовано непомітні атаки зловмисників на дані часових рядів у рекурентних нейронних мережах, щоб вивчити як безпеку глибоких рекурентних нейронних мереж, так і зрозуміти властивості навчання в глибоких рекурентних нейронних мережах.

Оскільки глибокі нейронні мережі широко використовуються в прикладних областях, існує можливість зниження точності та безпеки за допомогою методів зловмисників. Метод зловмисників - це отруєння даних через бекдор, коли зловмисник збиває навчальні вибірки невеликим збуренням, щоб неправильно класифікувати вихідний клас до цільового класу. При зміні даних через бекдор зловмисник має доступ до підмножини навчальних даних з мітками, можливість впливати на навчальні вибірки та можливість змінити мітку вихідного класу на мітку цільового класу. Зловмисник не має доступу до класифікатора під час навчання або знання процесу навчання. Також було досліджено захист від зміни даних через бекдор після навчання, розглядаючи ітераційний метод для визначення пари вихідного та цільового класів у такій атаці.

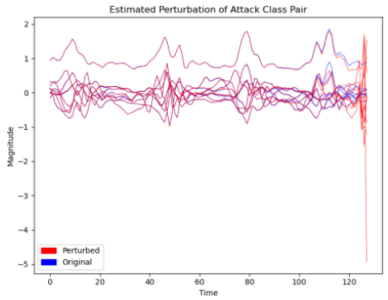

Рисунок 1 - Очікуване збурення пари класів атаки, LSTM

Методи атак через бекдор, успішно обманюють класифікатор LSTM, не знижуючи точності тестових зразків, без наявності шаблону бекдору. По-друге, метод захисту успішно визначає пару вихідних класів в такій атаці. По-третє, отруєння через бекдор у LSTM вимагає або більше навчальних зразків, або більшого збурення, ніж стандартна мережа прямого зв'язку.

Література.

2. B. Tran, J. Li and A. Madry. Spectral Signatures in Backdoor Attacks // NeurIPS. 2018. URL: https://arxiv.org/abs/1811.00636.