УДК 004.08

А. О. Нікітенко; Є. О. Башков, д-р техн. наук, проф.

ОПТИМІЗАЦІЯ СИСТЕМИ ВИЯВЛЕННЯ МЕРЕЖЕВИХ ВТОРГНЕНЬ НА ОСНОВІ ГЛИБОКОГО НАВЧАННЯ ІЗ ВИКОРИСТАННЯМ МЕТОДУ ЗМЕНШЕННЯ РОЗМІРНОСТІ ТА МЕТАЕВРИСТИЧНИХ АЛГОРИТМІВ

Системи виявлення мережевих вторгнень (NIDS) є невід'ємною складовою сучасної кібербезпеки, оскільки забезпечують моніторинг, аналіз та ідентифікацію загроз у режимі реального часу. Зі зростанням складності кібератак та мережевого трафіку потреба в ефективних механізмах виявлення аномалій стає критичною. У статті проаналізовано підходи до оптимізації NIDS, спрямовані на досягнення оптимального балансу між точністю класифікації та обчислювальною ефективністю шляхом усунення надлишкових ознак без втрати критично важливої інформації. Розглянуто методи зменшення розмірності та метаевристичні алгоритми, зокрема метод головних компонент (PCA), генетичний алгоритм (GA), оптимізацію рою частинок (PSO) та багатоцільову воронкову оптимізацію (MVO). Проведено експериментальне дослідження на наборі даних CSE-CIC-IDS-2018, який містить широкий спектр сучасних атак і нормального трафіку. Оцінено вплив різних алгоритмів на точність класифікації, час тренування моделей та вимоги до обчислювальних ресурсів. Виконано порівняльний аналіз ефективності GA, PSO та MVO у контексті оптимізації ознак для NIDS. Виявлено, що GA демонструє найкращий баланс між швидкістю обчислень і точністю класифікації, тоді як PSO та MVO є ефективними альтернативами для задач реального часу. Запропонований підхід дозволяє значно скоротити час тренування моделі, забезпечуючи оптимальний баланс між продуктивністю та результативністю. Використання метаевристичних алгоритмів та методів зменшення розмірності є перспективним напрямом для підвищення ефективності NIDS, що дозволяє забезпечити швидке та точне виявлення кібератак із мінімальними витратами ресурсів. Отримані результати сприяють подальшому розвитку адаптивних NIDS, здатних ефективно функціонувати в умовах реальних мереж із високою варіативністю трафіку.

Ключові слова: *мережеві вторгнення, зменшення розмірності, метаевристичні алгоритми, кібербезпека, глибоке навчання.*

Вступ

На сьогодні NIDS є ключовим інструментом у забезпеченні кібербезпеки сучасних інформаційних систем. У міру зростання складності та кількості кібератак розробка ефективних рішень для виявлення загроз стає важливішою, ніж будь-коли. Глибокі нейронні мережі, завдяки своїм потужним можливостям у задачах класифікації та розпізнавання шаблонів, стали основою для багатьох сучасних NIDS. Проте значні обчислювальні витрати та проблеми, пов'язані з високою розмірністю вхідних даних, обмежують їхнє широкомасштабне використання.

Висока розмірність даних створює додаткове навантаження на обчислювальні ресурси та може негативно впливати на продуктивність моделей через наявність нерелевантних або надлишкових особливостей. Таким чином, зменшення розмірності особливостей стає необхідним кроком у розробці оптимізованих і швидкодійних NIDS.

Аналіз літератури

На основі наведених аргументів стає очевидним, що оптимізація NIDS на основі глибокого навчання є складним завданням, яке вимагає пошуку ефективних підходів для збереження високої точності класифікації при зниженні обчислювальних витрат. У межах цього дослідження проведено ґрунтовний аналіз сучасної наукової літератури з метою

вивчення методів зменшення розмірності даних та метаевристичних алгоритмів для вибору ознак та оцінки їхнього впливу на продуктивність NIDS. Незважаючи на значну кількість публікацій у цій галузі, проблема оптимізації залишається відкритою, і науковці продовжують розробляти нові методи, спрямовані на підвищення ефективності таких систем. Нижче наведені основні роботи, які розкривають тему дослідження.

У роботі [1] запропоновано моделі вибору ознак для NIDS, що базуються на алгоритмах PSO, GWO, FFA та GA. Для оцінювання вибраних ознак із набору даних UNSW-NB15 застосовано класифікатори SVM та J48, а також проведено аналіз ефективності за метриками accuracy, precision, sensitivity, f-measure, TNR, FPR, FNR та TPR. Найкращі результати отримано з використанням алгоритму J48, де точність склала 90,48 %, precision — 84,13 %, sensitivity — 97,14 %, f-measure — 90,17 %, а FPR — 2,859 %. До недоліків роботи можна віднести відсутність аналізу важливих ознак, відібраних алгоритмами, неврахування часу, необхідного для навчання моделей, а також відсутність порівняння результатів із сучасними дослідженнями.

У роботі [2] представлено гібридну систему зменшення розмірності, що поєднує селекцію ознак і екстракцію ознак, зменшуючи 41 вхідну ознаку до 15 компонентів у наборі даних UNSW-NB15. Ефективність системи оцінювали за допомогою ансамблю класифікаторів SVC, KNC і DNN, використовуючи метрики точності, частоти виявлення, частоти хибних спрацьовувань, f1-балів та площі під кривою. Результати показали, що скорочення ознак не суттєво вплинуло на продуктивність класифікації, досягнувши 94,34 % точності, 93,92 % виявлення, 5,23 % хибних спрацьовувань, 94,32 % f1-міри та 94,34 % площі під кривою з використанням 15 компонентів. Недоліком роботи є відсутність неврахування часу, необхідного для навчання моделі та аналіз обраних ознак.

У роботі [3] представлено модель виявлення вторгнень, яка використовує GA для вибору ознак та алгоритми оптимізації для градієнтного спуску. По-перше, метод на основі GA використовується для вибору набору висококорельованих ознак з набору даних NSL-KDD, потім нейронна мережа зі зворотним поширенням (BPNN) навчається за допомогою методу HPSOGWO, гібридної комбінації алгоритмів оптимізації рою частинок (PSO) та оптимізації сірого вовка (GWO). Нарешті, гібридний алгоритм HPSOGWO-BPNN використовується для розв'язання задач бінарної та багатокласової класифікації на наборі даних NSL-KDD. Результати експерименту демонструють, що запропонована модель досягає кращих результатів, ніж інші методи, з точки зору точності, з меншим рівнем помилок і кращою здатністю виявляти різні типи атак. Недоліком є відсутність загального результату за мультикласовою класифікацією.

У роботі [4] розроблено моделі для виявлення кібератак із мінімальною кількістю ознак через ретельний відбір. Використано методи штучної бджолиної колонії (ABC), алгоритму запилення квітів (FPA) та оптимізації мурашиних колоній (ACO) для вибору 5, 9 та 10 ознак. Моделі показали точність понад 98,8 % (ACO), 98,7 % (FPA) та 98,6 % (ABC). Використано набір даних CSE-CIC-IDS2018 для проведення експериментальних досліджень. Недоліком роботи є відсутність результатів за мультикласовою класифікацією.

Проведений аналіз показує, що основні виклики оптимізації NIDS пов'язані з вибором ознак для тренувальних і тестових наборів даних, а також ефективністю передобробки. Неправильний відбір ознак може знизити точність класифікації: їх нестача призводить до втрати інформації, а надмірність — до ускладнення моделі та збільшення часу обробки.

Метаевристичні алгоритми, такі як GA і PSO, активно використовуються для зменшення часу тренування, оскільки дозволяють ефективно знаходити глобальні оптимуми. Однак ці методи іноді спричиняють втрату точності через перебільшену чи недостатню селекцію ознак. Тому важливо балансувати між швидкістю тренування та якістю моделі, враховуючи специфіку задачі.

Метою статті є розробка ефективного підходу до оптимізації NIDS, який дозволяє

знизити обчислювальні витрати та підвищити ефективність системи без суттєвого зниження точності класифікації.

Методологія оптимізації NIDS для підвищення продуктивності

Запропонована методологія спрямована на підвищення ефективності раніше розробленої авторської моделі (CNN-BiGRU-Attention). Останніми роками численні дослідники активно застосовують методи інтелектуального аналізу даних та машинного навчання для вирішення проблем і оптимізації продуктивності NIDS. Проте ключовим етапом у процесах глибокого навчання є попередня обробка даних. Цей етап є критично важливим для очищення даних від шуму та пропусків, нормалізації й стандартизації, зменшення вимірності, обробки категоріальних даних, балансування класів та зменшення обчислювальної складності. Застосування таких підходів дозволяє суттєво покращити якість даних, прискорити процес навчання моделей та забезпечити їхню стійкість. На основі проведеного аналізу літературних джерел встановлено, що більшість досліджень реалізують етап зменшення розмірності після попередньої обробки даних. У цій роботі використовується інший підхід: спочатку проводиться ретельна попередня обробка даних, яка включає всі перелічені аспекти, а вже потім — оптимізація розмірності вхідних ознак. Це дозволяє зберегти релевантність даних і мінімізувати втрати важливої інформації. Архітектура запропонованого фреймворку представлена на рис. 1. У наступних підрозділах докладно розглянуто етапи методології, що забезпечують підвищення ефективності та продуктивності моделі.

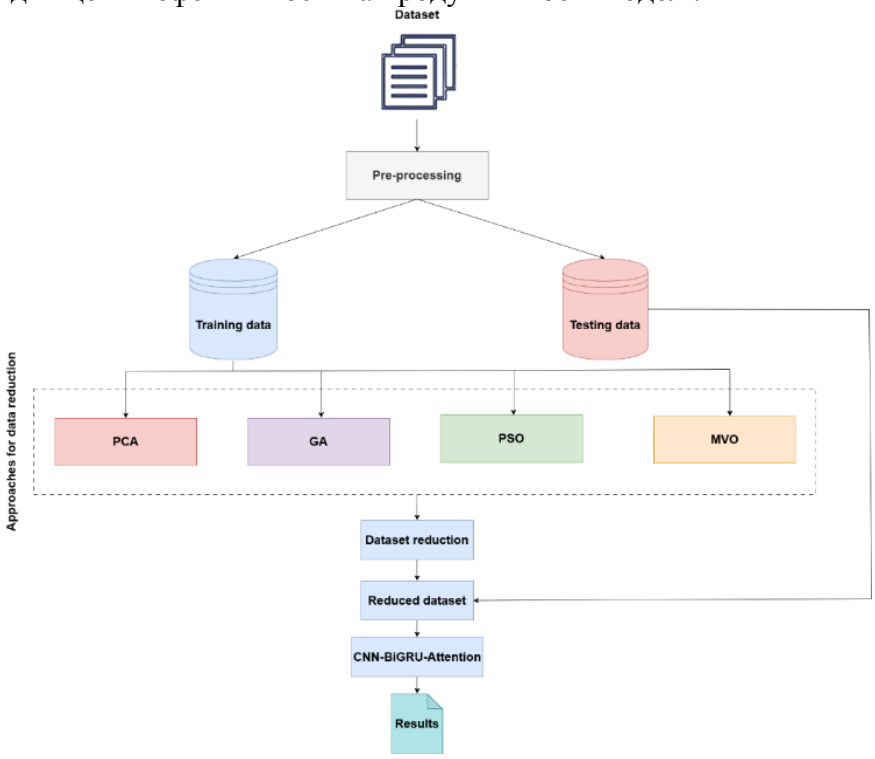

Рис. 1. Архітектура запропонованого фреймворку

Набори даних

Вибір відповідного набору даних для тренування та тестування моделі NIDS є ключовим фактором для забезпечення її ефективності та надійності. Правильно підібраний набір даних впливає на здатність моделі розпізнавати різноманітні типи атак, а також на її загальну продуктивність у реальних умовах експлуатації. Нижче коротко описано 3 набора даних, які використано у роботі:

1) Набір даних NSL-KDD [5], є широко використовуваним еталоном для досліджень у галузі виявлення вторгнень. Він вирішує проблему передискретизації, властиву набору даних KDD Cup 1999, завдяки збалансованішому розподілу категорій. Це дозволяє уникнути

зміщення моделей класифікації в бік частіших класів, забезпечуючи точніше виявлення менш поширених атак. NSL-KDD складається з навчального (KDDTrain+) та тестового (KDDTest+) наборів даних, які містять записи нормального трафіку та чотирьох типів атак: DoS, Probe, R2L та U2R.

2) Набір даних UNSW-NB15 [6] створено в Лабораторії кіберполігону UNSW у Канберрі за допомогою IXIA PerfectStorm для моделювання сучасного мережевого трафіку, що поєднує реальну звичайну активність і синтетичні атаки. Захоплення 100 ГБ сирого трафіку виконано за допомогою tcpdump, включаючи Pcap-файли. Набір містить дев'ять типів атак: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode та Worms. Для обробки даних використано Argus, Bro-IDS і дванадцять спеціальних алгоритмів, які генерують 49 ознак з мітками класу.

3) Набір даних CSE-CIC-IDS2018 [7] є вдосконаленою версією попереднього набору 2017 року, із залученням більшої кількості пристроїв для реалістичнішого моделювання атак. Інфраструктура включає мережу зловмисників із 50 машинами, мережу жертв із 420 машинами та 30 серверами, поділену на шість відділів: R&D, управління, технічний, секретаріат і операції, IT та серверні кімнати. У відділах встановлено ОС Windows (8.1 і 10), в IT-відділі – Ubuntu, а на серверах – Windows Server 2012/2016. Топологія відображає реальні мережі та охоплює 7 типів атак: Brute Force, Heartbleed, ботнет, DoS, DDoS, веб-атаки та інфільтрації.

Етап попередньої обробки даних

Попередня обробка включає в себе очистку, one-hot кодування та нормалізацію даних.

- *Очистка даних*. Для забезпечення цілісності даних усі рядки були ретельно перевірені на предмет виявлення пропущених та дублюваних значень. Цей процес є стандартною практикою при підготовці даних, оскільки допомагає зменшити вибірку від «шумних» даних. Для набору даних CSE-CIC-IDS2018, а саме одного з файлів «02-20-2018.csv» проведено скорочення з 84 ознак до 80 ознак, вилучивши перші чотири «Flow ID", "Src IP", "Src Port" та "Dst IP". В подальшому всі файли було об'єднано шляхом гомогенізації до 80 ознак. Після кореляційної фільтрації ознак, які включали в себе постійні нульові значення кількість ознак було скорочено до 69, зокрема було видалено «Bwd PSH Flags», «Fwd URG Flags», «CWE Flag Count», «Fwd Byts/b Avg», «Fwd Pkts/b Avg», «Fwd Blk Rate Avg», «Bwd Pkts/b Avg» і «Bwd Blk Rate Avg», "Bwd URG Flags".
- *One-hot кодування даних.* Набір даних NSL-KDD містить 38 числових ознак і 3 нечислові ознаки. Оскільки вхідним значенням авторської моделі має бути числова матриця, потрібно перетворити деякі нечислові ознаки, такі як «protocol_type», «service» та «flag» у числову форму. Наприклад, характеристика «protocol_type» має три типи атрибутів: «tcp», «udp» та «icmp», і її числові значення кодуються як двійкові вектори (1,0,0), (0,1,0) та (0,0,1). Аналогічно, характеристика «service» має 70 різних атрибутів, а характеристика «flag» – 11 типів атрибутів. Таким чином, 41-вимірний набір даних після перетворення перетворюється на 122-вимірний. Набір даних UNSW-NB15 містить 3 символьні (нечислові) ознаки та 39 неперервних (числових) ознак. Символічні ознаки не можуть бути безпосередньо використані як вхідні дані для моделі, тому три символьні ознаки "proto", "service", "state" перетворюються на числові, що розширює дані з 42 ознак до 196.
- *Нормалізація даних.* Значення вхідних даних можуть бути занадто великими, що призводить до таких проблем, як «великі числа, що їдять десяткові знаки», переповнення при обробці даних, непослідовність ваг тощо. По-перше, за деякими ознаками в наборі даних NSL-KDD, такими як 'duration [0,58329]', 'src_bytes [0,1.3 × 109]' та 'dst_bytes [0,1.3 × 109]', де різниця між максимальним та мінімальним значеннями має дуже великий розмах, застосовано логарифмічне перетворення, щоб отримати діапазони 'duration [0,4.77]', 'src_bytes [0,9.11]' і 'dst_bytes [0,9.11]' [8]. Так само на стовпчиках "dur", "sbytes", "dbytes",

"sload", "dload", "spkts", "stcpb", "dtcpb", "smeansz", "dmeansz", "djit", "sjit" застосовано логарифмічне перетворення для набору даних UNSW-NB15. По-друге використано Min-maxscaler [9] для нормалізації неперервних даних до діапазону (0, 1), Процедура нормалізації усуває вплив одиниці виміру на навчання моделі і робить результат навчання більш залежним від характеристик самих даних.

Після проведення всіх етапів перетворення, набір даних CSE-CIC-IDS2018 містить 7 214 096 екземплярів та 69 ознак для подальших експериментальних досліджень. Набір даних NSL-KDD складається з 148 517 екземплярів та 122 ознак. Набір даних UNSW-NB15 містить 2 540 047 екземплярів та 196 ознак.

Підходи до оптимізації моделі

Зменшення розмірності – це операція перетворення високорозмірного представлення даних у низькорозмірне представлення. Зі значним зростанням обсягів даних високої розмірності використання різних методів зменшення розмірності стало популярним у багатьох додатках. Більше того, постійно з'являються нові сучасні підходи. Методи зменшення розмірності трансформують вихідний набір даних високої розмірності і перетворюють його в новий набір даних низької розмірності, зберігаючи при цьому якомога більше вихідної розмірності. Низька розмірність вихідних даних допомагає вирішити проблему прокляття розмірності. Низькорозмірні дані можна легко аналізувати, обробляти та візуалізувати [10]. Загалом, методи зменшення розмірності можна розділити на дві основні групи, або, іншими словами, зменшення розмірності досягається двома різними методами: відбором ознак (Feature Selection – FS) та вилученням ознак (Feature Extraction – FE). При відборі ознак інформація може бути втрачена, оскільки деякі ознаки повинні бути виключені при відборі підмножини ознак, що зменшує кількість інформації. Однак, шляхом вилучення ознак можна зменшити розмірність без втрати значної частини початкового набору ознак. Вибір ознак для NIDS може бути здійснений за допомогою декількох підходів. Одним з таких підходів є метаевристичні алгоритми.

Метаевристичні алгоритми – це алгоритми, засновані на певних фізичних та біологічних стандартах. Вони поділяються на два типи: алгоритми на основі популяцій та алгоритми на основі одного рішення. Популяційні детектори вважаються більш придатними, ніж алгоритми на основі одного рішення [11]. До популяційних метаевристичних алгоритмів, що використовуються в цьому дослідженні, належать GA, PSO, MVO.

Метод головних компонент. PCA – це складний статистичний підхід, який використовується в аналізі даних і машинному навчанні для скорочення складних наборів даних. Його основна мета – зменшити кількість характеристик або вимірів у наборі даних, зберігаючи при цьому критично важливу інформацію. PCA робить це, змінюючи вихідні змінні на новий набір змінних, відомих як головні компоненти. Ці компоненти, які є лінійними комбінаціями вихідних ознак, навмисно робляться некорельованими, щоб охопити максимальну варіацію даних. PCA дозволяє науковцям і фахівцям з даних ефективніше аналізувати багатовимірні дані, виявляти закономірності та максимізувати продуктивність алгоритмів машинного навчання, обираючи головні компоненти, які прояснюють найбільшу варіативність [12]. PCA, по суті, спрощує як інтерпретацію, так і обробку даних, конденсуючи інформацію в більш зрозумілий і глибокий формат.

Генетичний алгоритм. GA – це метод еволюційного пошуку, який застосовується для вирішення оптимізаційних задач на основі методу природного відбору. GA кодує набір рішень для вирішення проблеми оптимізації. Ці рішення генеруються випадковим чином для формування популяції. Потім GA оцінює цю популяцію за допомогою функції пристосованості. Вибирається найкраще рішення, виходячи з задачі, що вирішується. Воно оцінюється з точки зору точності, середньоквадратичної похибки, F1-міри або площі під кривою. Особини, що підходять, були відібрані для набору операцій з розмноження, а саме

кросинговеру та мутації. Ці операції повторюються до тих пір, поки не будуть відповідати критерію припинення. Це призводить до формування набору поколінь [1].

Оптимізація рою частинок. PSO розроблена на основі простої концепції, отриманої з руху пташиних зграй і рибних косяків. Вона була розроблена після кількох інтерпретацій за допомогою комп'ютерного моделювання. PSO використовує різноманітні агенти (частинки), які утворюють рій. Цей рій подорожує в просторі пошуку, щоб знайти рішення, яке вважається найкращим. Щодо кожної частинки в просторі пошуку, вона змінює свій «політ» відповідно до свого досвіду польоту та досвіду польоту інших частинок [1]. PSO запускається випадково згенерованими частинками та їхньою швидкістю, яка вказує на швидкість пошуку. Потім, подібно до алгоритму GA, частинки оцінюються з точки зору придатності. За такою оцінкою слідують два основні тести. Перший тест порівнює досвід частинки з її власним, який називається особистим найкращим результатом (pbest). Другий тест порівнює придатність частинки з досвідом всього рою. Він називається глобальним найкращим (gbest). Виконання цих двох тестів призводить до збереження найкращої частинки. Після цього виконується критерій припинення.

Оптимізатор мультивсесвіту. MVO є метаевристичним алгоритмом, натхненним концепціями з фізики, зокрема теорією мультивсесвіту. Цей підхід базується на принципі, що різні можливі рішення (всесвіти) мають різну рівновагу [13]. В алгоритмі MVO рішення репрезентуються як всесвіти, що відображають розподіл "матерії", що відповідає рівню оптимальності рішень.

Процес оптимізації в MVO здійснюється через три основні операції: білий отвір, чорний отвір та потік матерії. Ці механізми імітують переміщення матерії між всесвітами, де більш оптимальні рішення обмінюються даними з менш оптимальними для досягнення глобального балансу. Алгоритм MVO характеризується здатністю уникати локальних мінімумів та високою ефективністю при вирішенні складних задач оптимізації.

Алгоритм починається з випадкової ініціалізації популяції всесвітів, після чого кожен всесвіт оцінюється за функцією пристосованості. Операції білих та чорних дір дозволяють покращувати рішення, водночас зберігаючи різноманітність популяції. Процес завершується, коли досягнуто заданої кількості ітерацій або встановлених критеріїв припинення. MVO активно використовується для задач вибору ознак, маршрутизації в мережах, енергетичного балансування та багатьох інших задач оптимізації.

Цей підхід демонструє високу продуктивність завдяки використанню фізичних метафор, що забезпечують широкий простір пошуку та ефективне знаходження глобальних оптимумів [3].

Експериментальне тестування та результати

У процесі виконання дослідження було проведено серію експериментів на кожному з наборів даних щодо вибору ознак за допомогою різних алгоритмів зменшення розмірності для тренування авторської моделі. Вибір ознак та зменшення розмірності виконувались як для мультикласової класифікації так і для бінарної класифікації з використанням наступних алгоритмів: PCA, GA, PSO, MVO. Для алгоритму GA були обрані такі параметри: коефіцієнт схрещування = 0,5, коефіцієнт мутації = 0,2, кількість генерацій = 10, кількість популяції = 20. Для PSO встановлено інерційний коефіцієнт = 0,5, розмір рою = 10, кількість ітерацій = 10. Для MVO визначено кількість всесвітів = 10, кількість ітерацій = 5, мінімальна й максимальна ймовірність існування кротової нори (wormhole) = 0,2 та 1,0 відповідно. Для PCA кількість компонент = 50.

Експериментальне середовище. У межах проведеного дослідження експериментальні обчислення здійснювалися у хмарному середовищі Google Colab Pro, яке забезпечує доступ до високопродуктивних обчислювальних ресурсів, що робить його оптимальним вибором для виконання ресурсоємних завдань. Усі експерименти виконувалися з використанням

Python версії 3.10.12, TensorFlow версії 2.15.0 та Keras версії 2.15.0. Для оптимізації гіперпараметрів використовувалася бібліотека Optuna.

Метрики оцінювання. Механізми уваги оцінюють за допомогою метрик точності (accuracy), правильності (precision), фактичних спостережень, що передбачені правильно (recall) та F1-міра (F1-score) [14]. Accuracy визначає частку правильно ідентифікованих об'єктів у загальному обсязі даних. Precision вимірює точність позитивних прогнозів, зроблених моделлю, і розраховується як відношення кількості правильно передбачених позитивних спостережень до загальної кількості передбачених позитивних результатів. Recall вимірює здатність моделі вловлювати всі позитивні результати, розраховуючи відношення кількості правильно передбачених позитивних спостережень до фактичних позитивних спостережень. F1-міра виражає середнє гармонійне значення точності та відтворюваності, що є збалансованою мірою, яка враховує як хибнопозитивні, так і хибнонегативні результати. Також додатковою метрикою виступає коефіцієнт помилкових спрацьовувань (FalsePositiveRate –FPR) оцінює відсоток записів, які помилково ідентифікуються як аномалії [3].

У таблиці 1 представлено результати роботи алгоритмів вибору ознак для кожного з раніше описаних наборів даних, призначених для задач мультикласової класифікації. Як свідчать наведені дані, всі розглянуті алгоритми зменшили загальну кількість ознак приблизно вдвічі, залишивши лише ті, що мають вагомий вплив на побудову моделі. Крім того, алгоритми обрали приблизно однакову кількість ознак, що дозволяє припустити їхню високу значущість для забезпечення ефективності виявлення вторгнень.

Таблиця 1

Вибрані ознаки з наборів даних

Алгоритм	Обрані ознаки	Кількість
NSL-KDD		
GA	f1,f4,f7,f8,f9,f10,f11,f15,f16,f17,f18,f19,f20,f21,f25,f28,f29,f34,f35,f40,f41,f43,f44,f48,f50,f51,f52,f53,f57,f58,f59,f60,f61,f62,f63,f64,f65,f68,f69,f70,f71,f72,f73,f74,f75,f76,f77,f79,f81,f82,f83,f85,f86,f88,f89,f91,f96,f97,f100,f103,f105,f106,f108,f109,f111,f117,f118,f119,f121	69
PSO	f1,f2,f3,f5,f6,f8,f9,f12,f15,f18,f24,f25,f26,f28,f29,f33,f34,f35,f37,f39,f40,f43,f44,f46,f48,f49,f50,f51,f52,f55,f59,f61,f62,f67,f68,f69,f71,f74,f75,f76,f77,f79,f84,f86,f91,f92,f93,f94,f95,f101,f102,f104,f106,f107,f108,f110,f111,f113,f114,f117,f118,f120	62
MVO	f2,f3,f4,f6,f8,f10,f11,f12,f14,f15,f16,f17,f18,f20,f21,f22,f23,f25,f27,f28,f29,f30,f31,f32,f34,f35,f36,f37,f39,f42,f49,f50,f53,f54,f55,f56,f57,f58,f61,f62,f66,f68,f69,f72,f75,f77,f78,f82,f83,f86,f89,f91,f93,f95,f96,f98,f108,f109,f112,f113,f118	61
UNSW-NB15		
GA	f1,f2,f4,f5,f8,f11,f12,f14,f16,f17,f19,f20,f21,f22,f23,f24,f26,f29,f31,f33,f37,f38,f39,f41,f42,f43,f45,f48,f49,f50,f52,f54,f56,f57,f58,f59,f63,f65,f67,f69,f70,f71,f74,f75,f76,f88,f90,f93,f94,f97,f99,f101,f102,f104,f105,f106,f107,f113,f114,f116,f117,f119,f120,f123,f124,f126,f127,f129,f136,f137,f138,f140,f142,f144,f145,f146,f149,f150,f151,f152,f153,f154,f155,f157,f160,f161,f162,f163,f164,f166,f170,f173,f174,f177,f178,f180,f186,f187,f188,f189,f190,f192,f196	103
PSO	f2,f3,f4,f5,f6,f8,f10,f11,f12,f13,f17,f18,f21,f22,f23,f24,f26,f29,f30,f33,f39,f40,f42,f47,f49,f50,f52,f53,f54,f56,f57,f62,f64,f65,f66,f72,f76,f78,f80,f84,f85,f88,f89,f90,f92,f93,f96,f99,f100,f101,f104,f105,f106,f109,f110,f111,f114,f115,f117,f120,f126,f128,f129,f131,f132,f133,f135,f137,f138,f139,f140,f141,f142,f143,f145,f148,f150,f151,f152,f153,f154,f156,f159,f160,f161,f163,f164,f170,f176,f179,f180,f182,f183,f184,f186,f187,f188,f189,f193,f196	100
MVO	f1,f2,f3,f5,f6,f7,f8,f13,f14,f15,f17,f21,f22,f23,f26,f29,f30,f37,f39,f43,f45,f46,f47,f48,f55,f58,f59,f60,f61,f65,f68,f69,f70,f71,f72,f76,f78,f79,f82,f84,f85,f88,f89,f90,f93,f95,f97,f98,f101,f103,f107,f109,f111,f115,f117,f119,f120,f122,f123,f124,f126,f127,f128,f129,f130,f132,f134,f136,f137,f138,f139,f142,f143,f145,f146,f149,f153,f155,f156,f158,f159,f160,f162,f163,f165,f167,f168,f170,f171,f172,f173,f176,f179,f182,f183,f185,f186,f191,f192,f195,f196	101
CSE-CIC-IDS2018		
GA	f1,f2,f3,f8,f11,f12,f14,f17,f19,f20,f21,f22,f23,f25,f26,f27,f28,f29,f30,f32,f34,f36,f40,f41,f44,f49,f51,f52,f55,f58,f60,f61,f63,f65,f66,f69	36
PSO	f2,f3,f4,f5,f7,f8,f10,f12,f13,f15,f16,f17,f22,f26,f27,f29,f31,f35,f36,f37,f39,f40,f41,f44,f45,f46,f47,f49,f50,f51,f52,f53,f59,f60,f61,f62,f68	37
MVO	f1,f2,f3,f5,f6,f7,f8,f9,f11,f13,f14,f18,f21,f22,f23,f25,f26,f28,f30,f31,f34,f35,f36,f37,f39,f42,f43,f44,f46,f47,f48,f49,f51,f53,f54,f55,f56,f57,f58,f59,f62	40

Незважаючи на те, що всі алгоритми вибрали приблизно однакову кількість ознак, спостерігаються значні відмінності у складі обраних характеристик. Наприклад, для набору даних NSL-KDD спільними ознаками є f86, f28, f50, f62, f75, f25, f29, f77, f34, f8, f61, f15, f18, f68, f69, f118, f35, f91, f108, які можна розглядати як ключові для задачі класифікації мережевих вторгнень. Для набору UNSW-NB15 спільними ознаками є: f129, f186, f160, f5, f17, f22, f196, f29, f76, f88, f101, f163, f2, f8, f170, f90, f120, f126, f93, f39, f137, f117, f65, f21, f26, f23, f138, f142, f153, f145. Для набору CSE-CIC-IDS2018 спільні ознаки включають: f29, f93, f2, f8, f39, f50, f69, f75, f113, f49, f12, f37. Ці ознаки можуть бути визначені як флагмани для розробки сучасних NIDS. Враховуючи різні стратегії вибору ознак, можна зробити висновок, що алгоритми застосовують специфічні підходи до вибору характеристик: GA схиляється до обрання більш широкого спектра ознак, тоді як PSO та MVO орієнтовані на відбір більш специфічних характеристик, що відповідають їхнім стратегіям оптимізації.

Далі розглянуто ефективність авторської моделі при використанні оптимізованих наборів даних NSL-KDD (табл. 2 – 3), UNSW-NB15 (табл. 4 – 5) та CSE-CIC-IDS2018 (табл. 6 – 7) в контексті мультикласової та бінарної класифікації.

Таблиця 2

Порівняння алгоритмів зменшення розмірності основані на наборі даних NSL-KDD (мультикласова класифікація)

Алгоритм	Accuracy	Precision	Recall	F1-score	FPR
Training					
PCA	99.56	99.52	99.56	99.54	0,0036
GA	99.82	99.82	99.82	99.82	0,0014
PSO	99.78	99.78	99.78	99.78	0,0016
MVO	97.42	97.40	97.42	97.41	0,017
Testing					
PCA	68.97	55.45	68.97	61,47	0,1941
GA	82.46	84.17	82.46	83,31	0,1126
PSO	80.98	83.75	80.98	82,34	0,1296
MVO	80.35	81.08	80.35	80,71	0,1248

Таблиця 3

Порівняння алгоритмів зменшення розмірності основані на наборі даних NSL-KDD (бінарна класифікація)

Алгоритм	Accuracy	Precision	Recall	F1-score	FPR
Training					
PCA	99.59	99.59	99.59	99.59	0,0044
GA	99.72	99.72	99.72	99.72	0,0029
PSO	99.54	99.54	99.54	99.54	0,0049
MVO	99.51	99.51	99.51	99.51	0,0052
Testing					
PCA	81.29	83.94	81.29	82.59	0,1578
GA	86.98	88.85	86.98	87.91	0,1087
PSO	82.82	86.08	82.82	84.42	0,1393
MVO	85.51	87.79	85.51	86,63	0,1174

Результати, отримані на наборі даних NSL-KDD, вказують на те, що більшість алгоритмів демонструють задовільну ефективність, при цьому GA досягає найкращих результатів за всіма метриками оцінки. У той же час, PCA продемонстрував хороші результати на тренувальних даних, однак на тестових даних спостерігалося значне погіршення показників, зокрема в контексті мультикласової класифікації. Це погіршення можна пояснити дисбалансом класів у тестових даних, зменшення кількості яких негативно позначився на результатах PCA. Варто зазначити, що цей ефект не спостерігався для інших алгоритмів.

Таблиця 4

Порівняння алгоритмів зменшення розмірності основані на наборі даних UNSW-NB15 (мультикласова класифікація)

Алгоритм	Accuracy	Precision	Recall	F1-score	FPR
Training					
PCA	97.43	97.14	97.43	97.28	0,023
GA	97.86	97.95	97.86	97.90	0,0302
PSO	97.88	97.71	97.88	97.79	0,036
MVO	97.84	97.84	97.84	97.84	0,029
Testing					
PCA	97.29	97.07	97.29	97,18	0,073
GA	97.89	97.89	97.89	97.89	0,030
PSO	97.86	97.63	97.86	97.74	0,032
MVO	97.89	97.82	97.89	97.85	0,028

Таблиця 5

Порівняння алгоритмів зменшення розмірності основані на наборі даних UNSW-NB15 (бінарна класифікація)

Алгоритм	Accuracy	Precision	Recall	F1-score	FPR
Training					
PCA	99.20	99.20	99.20	99.20	0.0392
GA	99.22	99.21	99.22	99.21	0,0358
PSO	99.15	99.15	99.15	99.15	0,0376
MVO	99.18	99.19	99.18	99.18	0,043
Testing					
PCA	99.11	99.11	99.11	99.11	0,040
GA	99.21	99.20	99.21	99.20	0,0364
PSO	99.17	99.16	99.17	99.16	0,0367
MVO	99.16	99.15	99.16	99.15	0,046

Результати, отримані на наборі даних UNSW-NB15, свідчать про те, що більшість алгоритмів демонструють задовільну ефективність. На відміну від попереднього набору даних, GA показав гірші результати лише на тренувальних даних у контексті мультикласової класифікації. Незважаючи на це, GA продовжує мати позитивний вплив на загальну ефективність моделі. Водночас, PCA знову продемонстрував найгірші результати серед усіх розглянутих алгоритмів.

Таблиця 6

Порівняння алгоритмів зменшення розмірності основані на наборі даних CSE-CIC-IDS2018 (бінарна класифікація)

Алгоритм	Accuracy	Precision	Recall	F1-score	FPR
Training					
PCA	99.10	99.09	99.11	99.10	0.0123
GA	99.41	99.45	99.41	99.43	0,0035
PSO	99.36	99.42	99.36	99.38	0,00299
MVO	98.97	99.00	98.97	98.98	0,0074
Testing					
PCA	98.94	98.91	98.94	98.93	0.0134
GA	99.41	99.45	99.41	99.42	0,0034
PSO	99.35	99.41	99.35	99.37	0.0128
MVO	98.94	98.91	98.94	98.93	0.0134

Таблиця 7

Порівняння алгоритмів зменшення розмірності основані на наборі даних CSE-CIC-IDS2018 (бінарна класифікація)

Алгоритм	Accuracy	Precision	Recall	F1-score	FPR
Training					
PCA	99.38	99.38	99.38	99.38	0,0086
GA	99.51	99.51	99.51	99.51	0,00606
PSO	99.40	99.40	99.40	99.40	0,0076
MVO	99.49	99.49	99.49	99.49	0,0072
Testing					
PCA	97.73	97.75	97.25	97.50	0,043
GA	99.50	99.50	99,50	99.50	0,00621
PSO	99.39	99.39	99.39	99.39	0,0077
MVO	99.48	99.48	99.48	99.48	0,0075

Результати, отримані на наборі даних CSE-CIC-IDS2018, свідчать про те, що більшість алгоритмів демонструють задовільну ефективність у вирішенні задачі класифікації. У випадку застосування GA, він знову продемонстрував найкращі результати. Водночас, PCA знову виявився найменш ефективним серед усіх розглянутих методів, продемонструвавши найгірші показники за всіма метриками на цьому наборі даних.

Для комплексного аналізу позитивного впливу оптимізації набору даних на процес навчання моделі було проведено додаткове порівняння часу тренування моделі з використанням кожного з алгоритмів оптимізації та без них, при фіксованих значеннях гіперпараметрів. Це порівняння здійснено в контексті мультикласової класифікації, і результати представлені в таблиці 8. Основні фіксовані параметри: розмір батча = 512, кількість епох = 5, швидкість навчання = 0,001.

Таблиця 8

Порівняння часу тренування моделі з наборами даних

Алгоритм	Час тренування, сек.
NSL-KDD	
Модель без оптимізації	18
PCA	17
GA	17
PSO	17
MVO	17
UNSW-NB15	
Модель без оптимізації	431
PCA	428
GA	415
PSO	412
MVO	422
CSE-CIC-IDS2018	
Модель без оптимізації	1473
PCA	1492
GA	1322
PSO	1337
MVO	1346

Результати дослідження демонструють, що зменшення розмірності ознак суттєво скорочує час тренування моделі, причому цей ефект є більш вираженим на великих наборах даних, таких як CSE-CIC-IDS2018, порівняно з невеликими наборами, наприклад, NSL-KDD, де такі

зміни є менш помітними. Крім того, спостерігається зменшення обсягу ресурсів, необхідних для зберігання даних, що використовуються під час тренування та тестування моделі.

У таблиці 9 представлені результати аналізу сучасних розробок, спрямованих на оптимізацію NIDS. Ці дослідження обґрунтовані необхідністю створення більш ефективних та надійних методів протидії кібератакам і зосереджені на різноманітних аспектах кібербезпеки. Зокрема, вони охоплюють виявлення нових загроз, аналіз вразливостей, удосконалення алгоритмів машинного навчання для ідентифікації аномальної мережевої активності та підвищення ефективності механізмів реагування на інциденти безпеки. Порівняльний аналіз із цими роботами дозволяє оцінити, наскільки запропонована модель відповідає сучасним вимогам галузі, та визначити потенційні напрями її вдосконалення для подальшого розвитку наявних підходів.

Таблиця 9

Порівняння оптимізованої моделі з сучасними дослідженнями у контексті бінарної класифікації

Назва моделі	Набір даних	Accuracy	Precision	Recall	F1-score	FPR
PSO [15]	NSL-KDD	99,32	99,37	N/A	99,31	0,62
XGBoost+PCA [16]	CSE-CIC-IDS2018	99,77	92,07	98,87	94,93	N/A
MOB-EBATMLP [17]	UNSW-NB15	97,63	N/A	98,18	N/A	N/A
Авторська модель (CNN-BiGRU-Attention)	NSL-KDD	99,85	99,85	99,85	99,85	N/A
Авторська модель (CNN-BiGRU-Attention)	UNSW-NB15	99,20	99,20	99,20	99,20	N/A
Оптимізована модель	NSL-KDD	99,72	99,72	99,72	99,72	0,0029
Оптимізована модель	UNSW-NB15	99,22	99,21	99,22	99,21	0,0358
Оптимізована модель	CSE-CIC-IDS2018	99,51	99,51	99,51	99,51	0,00621

Порівняння запропонованої моделі з сучасними дослідженнями свідчить про її підвищену ефективність у певних аспектах. Зокрема, порівняно з попередньою авторською моделлю, точність для набору даних NSL-KDD дещо зменшилася, проте було досягнуто значного скорочення часу навчання та обсягу використаної пам'яті для зберігання й тренування даних. Це свідчить про успішне досягнення ключових цілей дослідження — оптимізації продуктивності та ресурсоефективності моделі.

Додатково, використання метрики FPR, яка не завжди застосовується у сучасних роботах, забезпечує більш комплексну оцінку результатів і може слугувати корисним орієнтиром для майбутніх досліджень у цій галузі. Такий підхід підкреслює прагнення до створення практичних та універсальних рішень у сфері виявлення мережевих вторгнень.

Висновки

На основі проведеного огляду та критичного аналізу сучасних досліджень у галузі оптимізації NIDS встановлено, що питання зменшення розмірності даних залишається актуальним і важливим для підвищення ефективності моделей. Дослідники активно застосовують широкий спектр алгоритмів, зокрема метаевристичні та гібридні підходи, які поєднують кілька методів зменшення розмірності для досягнення більш точних і швидких рішень. Крім того, значна увага приділяється попередній обробці даних, яка відіграє ключову роль в оптимізації моделі, забезпечуючи її стійкість до надмірності або недостатності ознак.

Виконано попередню обробку наборів даних NSL-KDD, UNSW-NB15 та CSE-CIC-IDS2018, що забезпечило належну якість даних для подальшого аналізу**.** Для зменшення

розмірності було застосовано алгоритми PCA, GA, PSO та MVO. У результаті кількість ознак зменшилась приблизно вдвічі, що значно оптимізувало процес тренування моделей. Додатково було визначено спільні релевантні ознаки, які, ймовірно, мають істотний вплив на якість класифікації та ефективність моделей для виявлення мережевих вторгнень та можуть бути використаними у майбутніх дослідженнях.

Проведено експериментальні дослідження з використанням оптимізованих наборів даних та авторської моделі. Аналіз результатів виявив, що GA забезпечує найвищі показники за основними метриками оцінювання ефективності. Порівняння отриманих результатів із сучасними дослідженнями вказує на те, що оптимізована модель демонструє значні переваги з точки зору ефективного використання обчислювальних ресурсів та часу тренування. Проте це супроводжується деяким зниженням точності за метриками оцінювання, що вказує на необхідність подальшого вдосконалення підходів до оптимізації для досягнення кращого балансу між ефективністю та якістю класифікації.

Хоча проведене дослідження демонструє задовільні результати, воно не є остаточним і має певні обмеження. Зокрема, одним із недоліків є складність інтерпретації рішень авторської моделі, що може обмежувати її практичне застосування в реальних умовах, де важлива прозорість роботи NIDS. Також варто відзначити невелике падіння точності після оптимізації моделі. У майбутніх дослідженнях планується вирішити ці обмеження, зокрема шляхом дотренування моделі з використанням більш сучасних та репрезентативних наборів даних, що дозволить врахувати нові типи кіберзагроз. Особливу увагу буде приділено подальшій оптимізації моделі із застосуванням сучасних методів зменшення розмірності, таких як генетичний алгоритм, для досягнення кращого балансу між ефективністю використання ресурсів і точністю класифікації.

СПИСОК ЛІТЕРАТУРИ

1. Almomani O. A Feature Selection Model for Network Intrusion Detection System Based on PSO, GWO, FFA and GA Algorithms. *Symmetry*. 2020, Vol. 12, № 6. P. 1046. https://doi.org/10.3390/sym12061046.

2. Ghani H., Salekzamankhani S., Virdee B. A Hybrid Dimensionality Reduction for Network Intrusion Detection. *J. Cybersecur. Privacy*. 2023. Vol. 3, № 4. P. 830–843. https://doi.org/10.3390/jcp3040037.

3. Sheikhi S., Kostakos P. A Novel Anomaly-Based Intrusion Detection Model Using PSOGWO-Optimized BP Neural Network and GA-Based Feature Selection. *Sensors*. 2022. Vol. 22, № 23. P. 9318. https://doi.org/10.3390/s22239318.

4. Najafi Mohsenabad H., Tut M. Optimizing Cybersecurity Attack Detection in Computer Networks: A Comparative Analysis of Bio-Inspired Optimization Algorithms Using the CSE-CIC-IDS 2018 Dataset. *Applied Sciences*. 2024. Vol. 14, № 3. P. 1044. https://doi.org/10.3390/app14031044.

5. NSL-KDD dataset. URL: https://www.unb.ca/cic/datasets/nsl.html. (дата звернення 15.01.2025).

6.The UNSW-NB15 Dataset. URL: https://research.unsw.edu.au/projects/unsw-nb15-dataset. (дата звернення 15.01.2025).

7. CIC-IDS-2018 onAWS. URL: https://www.unb.ca/cic/datasets/ids-2018.html (дата звернення 15.01.2025).

8. Yin C., Zhu Y., Fei J., He X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Acces*s. 2017. Vol. 5. P. 21954–21961. https://doi.org/10.1109/access.2017.2762418.

9. BAT: Deep Learning Methods on Network Intrusion Detection Using NSL-KDD Dataset / T. Su et al. *IEEE Access*. 2020. Vol. 8. P. 29575–29585. https://doi.org/10.1109/access.2020.2972627.

10. A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction / R. Zebari et al. *J. Appl. Sci. Technol. Trends*. 2020. Vol. 1, № 2. P. 56–70. https://doi.org/10.38094/jastt1224.

11. Almomani O. A Hybrid Model Using Bio-Inspired Metaheuristic Algorithms for Network Intrusion Detection System. *Computers, Materials& Continua*. 2021. Vol. 68, № 1. P. 409–429. https://doi.org/10.32604/cmc.2021.016113.

12. Nskh P., Varma M., Naik R. Principle component analysis based intrusion detection system using support vector machine. *2016 IEEE Int. Conf. Recent Trends Electron., Inf. Communication Technol. (RTEICT). 2016.* P. 1344–1350. https://doi.org/10.1109/rteict.2016.7808050.

13. Mirjalili S., Mirjalili M., Hatamlou A. Multi-Verse Optimizer: a nature-inspired algorithm for global optimization. *Neural Comput. Appl.* 2015. Vol. 27, № 2. P. 495–513. https://doi.org/10.1007/s00521-015-1870-7.

14. Network Intrusion Detection Technology Based on Convolutional Neural Network and BiGRU / B. Cao et al. *Computational Intelligence and Neuroscience*. 2022. https://doi.org/10.1155/2022/1942847.

15. Kunhare N., Tiwari R., Dhar J. Particle swarm optimization and feature selection for intrusion detection system. *Sādhanā*. 2020. Vol. 45. https://doi.org/10.1007/s12046-020-1308-5.

16. Songma S., Sathuphan T., Pamutha T. Optimizing Intrusion Detection Systems in Three Phases on the CSE-CIC-IDS-2018 Dataset. *Computers*. 2023. Vol. 12, № 12. P. 245. https://doi.org/10.3390/computers12120245.

17. Cyber Intrusion Detection System Based on a Multi-objective Binary Bat Algorithm for Feature Selection and Enhanced Bat Algorithm for Parameter Optimization in Neural Networks / W. A. H. M. Ghanem et al. *IEEE Access*, 2022. Vol. 10. P. 76318–76339. https://doi.org/10.1109/access.2022.3192472

Стаття надійшла до редакції 11.03.2025.
Стаття пройшла рецензування 15.03.2025.

Нікітенко Андрій Олександрович – аспірант кафедри прикладної математики та інформатики, e-mail: andrii.nikitenko@donntu.edu.ua.

Башков Євген Олександрович – д-р техн. наук, професор, професор кафедри прикладної математики та інформатики.

Донецький національний технічний університет.