КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (32), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.32.1204
УДК 004.056.5:004.89

Толюпа Сергій Васильович
д.т.н., професор, професор кафедри кібербезпеки та захисту інформації
Київський національний університетімені Тараса Шевченка, Київ, Україна
ORCID: 0000-0002-1919-9174
serhii.toliupa@knu.ua

Кулько Андрій Аркадієвич
аспірант кафедри кібербезпеки та захисту інформації
Київський національний університет імені Тараса Шевченка, Київ, Україна
ORCID:0009-0006-1185-0774
kulko452@gmail.com

МЕТОДИКА КОМПЛЕКСНОЇ ОПТИМІЗАЦІЇ ОЗНАК (ФІЧЕРІВ) ДЛЯ СИСТЕМ ВИЯВЛЕННЯ КІБЕРАТАК

Анотація. У статті розглядається одна з найбільш гострих проблем сучасної кібербезпеки – необхідність підвищення ефективності інтелектуальних систем виявлення вторгнень (IDS) в умовах стрімкої цифровізації та ускладнення ландшафту загроз. Автори змістовно обґрунтовують, що традиційні сигнатурні методи стають недостатніми проти атак, керованих штучним інтелектом, що зумовлює перехід до методів машинного навчання. Проте висока розмірність мережевого трафіку та наявність великої кількості надлишкових, корельованих або шумних ознак створюють ефект «прокляття розмірності». Це призводить до критичного зростання обчислювальних витрат, уповільнення реакції систем у реальному часі та зниження точності класифікації через перенавчання моделей. Актуальність роботи підтверджується необхідністю розробки системних підходів до препроцесингу даних, зокрема на прикладі еталонного датасету NSL-KDD. Об'єктом дослідження є процес оптимізації вхідних даних для класифікаторів кібератак. Автором запропоновано та детально описано чотириетапну методику комплексної оптимізації ознак (фічерів). Методологія базується на гібридному поєднанні різних підходів: попередня обробка: очищення, нормалізація та стандартування; відбір ознак: застосування фільтрових методів (кореляційний аналіз Пірсона, взаємна інформація MI), вбудованих методів та обгорткових методів; виділення ознак: використання методів зниження розмірності, таких як PCA (метод головних компонент) та LDA/ULDA (лінійний дискримінантний аналіз), що дозволяє трансформувати вихідний простір у менший набір некорельованих компонент. Наукова новизна роботи полягає у системному поєднанні статистичних фільтрів із ансамблевими методами навчання для тонкого налаштування моделей під специфіку мережевого трафіку. У статті наведено математичне обґрунтування кожного методу, зокрема через ентропію Шеннона та індекс Джині. Доведено, що для датасету NSL-KDD використання лише 12-15 найбільш релевантних ознак дозволяє підтримувати точність класифікації на рівні 98-99%, значно випереджаючи моделі, навчені на повному наборі (41 ознака), за показниками швидкості навчання та інференсу. Окрему увагу приділено перевагам методу ULDA у боротьбі з мультиколінеарністю. Автори доходять висновку, що запропонована методика є універсальним інструментом для оптимізації IDS, що дозволяє досягти балансу між точністю, швидкістю та стійкістю системи. Визначено вектори подальших досліджень: адаптація моделей до незбалансованих даних, використання нелінійних автокодувальників на базі глибокого навчання та дослідження стійкості відібраних ознак до змагальних атак.

Ключові слова: кібербезпека, системи виявлення вторгнень (IDS), оптимізація ознак, відбір ознак, екстракція ознак, датасет NSL-KDD, аналіз головних компонент (PCA), лінійний дискримінантний аналіз (LDA), машинне навчання, препроцесинг даних, зниження розмірності.

ВСТУП

У сучасних умовах глобальної цифровізації кібербезпека стала критично важливим аспектом функціонування як державних інституцій, так і приватного сектору. Зростання складності та інтенсивності кібератак, зокрема за рахунок використання технологій штучного інтелекту зловмисниками на об'єкти критичної інфраструктури та кіберфізичні системи [1], висуває нові вимоги до систем виявлення вторгнень (IDS). Традиційні методи сигнатурного аналізу все частіше поступаються місцем інтелектуальним системам, що базуються на машинному навчанні (Machine Learning). Проте ефективність таких систем безпосередньо залежить від якості та релевантності вхідних даних. Сучасний мережевий трафік генерує величезні обсяги інформації з великою кількістю ознак (фічерів), багато з яких є надлишковими, малоінформативними або шумними. Це призводить до ряду проблем: збільшення обчислювальних витрат (обробка зайвих даних потребує значних ресурсів та часу), зниження точності (нерелевантні ознаки можуть вводити модель в оману, підвищуючи рівень хибнопозитивних спрацювань (False Positives)), ефект «прокляття розмірності» (надмірна кількість параметрів ускладнює навчання моделі та її здатність до узагальнення). Актуальність даної роботи зумовлена необхідністю розробки та впровадження методів оптимізації ознак, які дозволять скоротити розмірність простору даних без втрати важливої інформації. Оптимізація набору ознак є ключовим етапом препроцесингу, що дозволяє не лише пришвидшити роботу алгоритмів виявлення кібератак у режимі реального часу, але й значно підвищити їхню надійність. Метою статті є аналіз існуючих методів відбору та екстракції ознак, а також розробка підходів до їх оптимізації для покращення метрик якості систем виявлення кібератак.

Застосування алгоритмів інтелектуального аналізу даних (ІАД) [2-3] при створенні систем виявлення вторгнень (СВВ) дозволяє мінімізувати типові вади, притаманні класичним методам пошуку сигнатур та детекції аномалій. Попри очевидні переваги, процес вибору конкретних інструментів та розробка регламентів їх впровадження залишається складним ітераційним завданням. Ключовим фактором тут є необхідність проведення масштабних апробацій, оскільки підсумкова ефективність СВВ безпосередньо корелює з якістю та репрезентативністю навчальної вибірки.

Проблематика формування тренувальних датасетів. Окрім архітектурної досконалості самих систем, критичне значення має підбір бази даних для їхнього навчання та подальшої верифікації. Формування якісного масиву даних, що містить актуальні ознаки кібератак, є нетривіальним викликом.

У науковій спільноті для розробки та валідації IDS найчастіше застосовуються такі датасети: KDDCup 1999 (KDD99): один із наймасштабніших наборів (близько 5 млн записів), що класифікує загрози за чотирма напрямками: DoS, Probe, R2L та U2R. Проте суттєвим недоліком є висока надмірність даних і велика кількість дублікатів. Це призводить до статистичних викривлень: точність моделей штучно завищується через фокусування на поширених атаках (зокрема DoS) на шкоду рідкісним сценаріям; NSL-KDD: оптимізована версія попередньої бази. Завдяки усуненню дублів та очищенню від шумів, цей датасет дозволяє алгоритмам ефективніше ідентифікувати специфічні атаки (U2R, R2L). В академічному середовищі він вважається більш збалансованим еталоном для порівняльного аналізу; CICIDS2017 та CICDDoS2019: представники баз нового покоління даних, що максимально наближені до реалій сучасних мереж; CICIDS2017 вирізняється високою деталізацією (понад 80 параметрів) і охоплює актуальні вектори нападів. Його перевага – у відтворенні реалістичної топології мережі та профілів

поведінки користувачів; CICDDoS2019 є вузькоспеціалізованим інструментом для боротьби з розподіленими атаками на відмову в обслуговуванні, охоплюючи різноманітні протокольні вразливості (HTTP, UDP тощо); UNSW-NB15: сучасна альтернатива класичним базам, яка містить 49 ознак і дев'ять категорій загроз, що відсутні в застарілих наборах. Завдяки використанню реальних інструментів генерації трафіку, цей датасет є незамінним для тестування систем, орієнтованих на пошук аномалій.

Таким чином для створення та СВВ, здатної протидіяти новітнім викликам, доцільно використовувати UNSW-NB15 або CICIDS2017. Водночас для забезпечення наступності досліджень та можливості зіставлення отриманих результатів із класичними методами, у даній роботі як базовий набір даних буде використано NSL-KDD.

Аналіз останніх досліджень і публікацій. Питаннями підвищення ефективності систем виявлення кібератак за рахунок оптимізації ознак (фічерів) займалися цілий ряд науковців як у нас в країні так і за кордоном. Треба надати належне все ж закордонним публікаціям де цьому питанню приділяється більше уваги. Так в [4] автор стверджує, що найкращим набором даних є набір ADFA-LD. Вважається, що цей набір даних має більшу схожість між даними про атаки та звичайними даними, ніж набори даних безпеки KDD. Однак інші дослідники вважають, що для демонстрації ефективності в мережевому аналізі необхідні подальші експерименти з іншими наборами даних [5]. Скорочення та вибір особливостей часто використовуються в сучасних публікаціях про виявлення вторгнень. Скорочення/виділення особливостей - це процес пошуку нових підпросторів з меншою кількістю вимірів, ніж оригінальний простір особливостей [6]. Так автори в [7] запропонували багатокласову класифікацію для сприяння створенню сучасних моделей виявлення вторгнень з поліпшеною ефективністю та точністю. В [8] показано, що через нестабільність виявлення вторгнень існує велика диспропорція між класами в наборі даних NSL-KDD, що ускладнює ефективне застосування машинного навчання в області виявлення вторгнень. В [9] досліджено дану проблематику, що стосується методів класифікації та методів виявлення вторгнень. Результати показують, що NSL-KDD показав найкращі загальні результати після навчання на заданих класифікаторах. В [10] запропоновано підхід до вибору ознак на основі генетичного алгоритму для ефективної системи виявлення вторгнень. В [11] розглянуто сучасні методи відбору ознак та класифікації для IDS, досліджують інтелектуальні методи розробки IDS, а потім розробляють нову IDS з використанням двох запропонованих алгоритмів. В [12] автори досліджують значущі особливості систем виявлення аномалій з метою їх застосування в техніках аналізу даних. В [13] автори виявляють три сценарії, в яких необхідні правильно марковані набори даних. Зазначається, що при використанні неконтрольованих IDS існує потреба в маркованих наборах даних для навчання. Автори в [14] досліджують ефективні способи вибору ознак, порівнюючи два алгоритми. Алгоритм C4.5, алгоритм дерева рішень і алгоритми C4.5 обрізають і тестують запропоновані ознаки на наборі даних KDD 99 і NSL KDD для тестування і навчання алгоритму класифікатора.

Постановка проблеми дослідження. Основна проблема полягає у зниженні ефективності інтелектуальних систем виявлення вторгнень (IDS) через неповноту традиційних методів та надлишковість сучасних даних. Її можна розділити на кілька ключових аспектів. Традиційні системи, що базуються на базах відомих сигнатур, виявляються неефективними проти нових, складних атак, керованих штучним інтелектом (AI-driven attacks) та атак «нульового дня». Сучасний мережевий трафік

характеризується великою кількістю параметрів (наприклад, 41 ознака в NSL-KDD). Таким чином невідповідність між зростаючою складністю кіберзагроз та обмеженою продуктивністю систем виявлення вторгнень через обробку великих обсягів неоптимізованих мережевих даних, що потребує впровадження гібридних методів зниження розмірності.

Запропонована методика базується на системному підході до проектування і вибору ознак, що дозволяє трансформувати сирий мережевий трафік у високоефективний набір показників. Вона поєднує класичні статистичні фільтри, сучасні обгорткові методи на основі метаевристичних алгоритмів та методи зниження розмірності, такі як PCA та LDA.

Головною метою даної стратегії є досягнення оптимального балансу між трьома показниками: максимальна точність: виявлення складних атак з мінімальною кількістю помилкових спрацьовувань; обчислювальна легкість: мінімізація кількості ознак для прискорення навчання та інференсу; стійкість: здатність моделі стабільно працювати на нових, раніше невідомих зразках трафіку.

Нижче наведено детальний опис чотирьох етапів реалізації цієї методики: від первинної обробки до фінальної оцінки ефективності обраного набору ознак.

1. Попередня обробка та інженерія ознак.

Цей етап забезпечує якість та інформативність вихідних даних.

Очищення даних: виявлення та обробка пропущених значень (заміна середнім, медіаною, модою або видалення).

Обробка викидів (наприклад, за допомогою IQR-методу або ізоляційного лісу).

Нормалізація/масштабування: Min-Max Scaling (обмеження ознак у діапазоні [0,1]):

$$X_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}.$$

Z – Score Standardization (для приведення до середнього 0 та стандартного відхилення):

$$X_{std} = \frac{X - \mu}{\sigma}.$$

Мета: забезпечити рівний вплив усіх ознак на модель.

Інженерія ознак: створення нових, більш інформативних ознак із сирих даних (наприклад, швидкість зміни пакетів, співвідношення вхідного/вихідного трафіку, ентропія портів призначення/джерела).

2. Відбір ознак.

Мета – видалити надлишкові, нерелевантні або сильно корельовані ознаки. Існує три основні категорії методів:

Фільтрові методи. Оцінюють ознаки незалежно від моделі, базуючись на статистичних показниках.

Кореляційний аналіз: видалення ознак з високою парною кореляцією (наприклад, якщо $\rho > 0,95$, видаляється одна з ознак).

X_i – квадрат: використовується для категоріальних ознак. Оцінює залежність між ознакою та цільовою змінною.

Інформаційний приріст/взаємна інформація: вимірює зменшення ентропії (або кількість інформації), яку ознака надає про цільову змінну.

Обгорткові методи. Використовують саму модель для оцінки ефективності підмножини ознак.

Жадібний пошук: прямий відбір. Починаємо з порожнього набору і послідовно додаємо ознаку, яка дає найбільше покращення.

Зворотне виключення: починаємо з повного набору і послідовно видаляємо найменш значущу ознаку.

Вбудовані методи. Відбір ознак відбувається в процесі навчання моделі. Регуляризація L_1 (LASSO): додає штраф до коефіцієнтів моделі, змушуючи коефіцієнти менш важливих ознак ставати нульовими, фактично відключаючи їх.

3. Виділення ознак/зниження розмірності.

Методи для створення нового, меншого набору ознак, які є комбінаціями вихідних.

Метод головних компонент (Principal Component Analysis, PCA): перетворює початковий набір корельованих ознак у набір лінійно некорельованих головних компонент. Зберігає максимальну дисперсію даних у меншій кількості вимірів.

Лінійний дискримінантний аналіз (Linear Discriminant Analysis, LDA): використовується для багатокласової класифікації, знаходить осі, які максимізують роздільність між класами.

Автокодувальники (Autoencoders): нейронні мережі, які навчаються стискати вхідні дані до меншого внутрішнього представлення (кодування) і потім відновлювати вихідні дані. Внутрішній шар (bottleneck) виступає як стислий, оптимізований набір ознак.

4. Оцінка та ітерація.

Оцінка результатів і повторення процесу для досягнення найкращої продуктивності.

Метрики оцінки: точність (accuracy), прецизійність (precision), повнота (recall), F_1-Score.

Коефіцієнт правильного виявлення (Detection Rate – DR) та частота помилкових спрацьовувань (False Alarm Rate – FAR).

Критерій оптимізації: максимізація F_1-Score або AUC-ROC при одночасному мінімізації кількості ознак.

Крос-валідація: використання методів, наприклад, k-fold крос-валідації, для перевірки стійкості обраного набору ознак.

Порівняння: порівняння продуктивності моделі до та після оптимізації ознак. Якщо продуктивність зросла або залишилася на тому ж рівні при значному зменшенні кількості ознак, оптимізація вважається успішною.

Рекомендована послідовність дій:

1. Попередня обробка та інженерія для створення потенційно інформативних ознак.
2. Використання фільтрових методів для швидкого усунення найбільш очевидно надлишкових ознак.
3. Застосування вбудованих методів для ідентифікації найважливіших ознак.
4. Використання обгорткових методів на невеликій підмножині ознак для "тонкого налаштування" фінального набору.
5. Фінальна оцінка на незалежних тестових даних.

Використання фільтрових методів є першим і найшвидшим кроком в оптимізації ознак. Їхня головна перевага – незалежність від моделі. Вони оцінюють важливість

ознаки, ґрунтуючись лише на внутрішніх властивостях даних, таких як кореляція та інформаційна залежність.

У межах запропонованої методики ключову роль відіграють статистичні підходи, що базуються на аналізі внутрішніх взаємозв'язків між ознаками та їхнього впливу на цільову змінну (клас трафіку).

Нижче наведено математичне обґрунтування та алгоритмічну послідовність реалізації двох фундаментальних методів відбору: кореляційного аналізу для усунення надлишковості та аналізу взаємної інформації для оцінки релевантності даних. Ці методи в сукупності забезпечують формування оптимального вектору ознак, що гарантує високу точність класифікації при мінімальних часових витратах на навчання системи.

1. Кореляційний аналіз (Correlation Analysis). Метод використовується для виявлення та усунення надлишкових ознак, які надають ту саму інформацію.

Обґрунтування. Якщо дві ознаки, X_i та X_j, сильно корельовані, то включення обох до моделі не додасть значної нової інформації, але збільшить обчислювальні витрати та може спричинити проблеми мультиколінеарності (що ускладнює інтерпретацію лінійних моделей та робить їх менш стабільними). У такому випадку достатньо залишити лише одну з них.

Математичний апарат: коефіцієнт кореляції Пірсона. Коефіцієнт кореляції Пірсона $p_{X,Y}$ вимірює силу та напрямок лінійного зв'язку між двома числовими змінними X та Y.

$$p_{X,Y} = \frac{Gov(X,Y)}{\sigma_x \sigma_y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_x \sigma_y},$$

де: $Gov(X,Y)$ – коваріація між X та Y;

σ_x, σ_y – стандартні відхилення X та Y;

$p_{X,Y} \in [-1,1]$.

Правило усунення.

1. Обчислити матрицю кореляції для всіх числових ознак.
2. Вибрати порогове значення τ (наприклад, $\tau = 0.95$).
3. Якщо $|p_{X,Y}| > \tau$, то ознаки X_i та X_j, вважаються надлишковими.
4. Видалити одну з ознак. Оптимально – видалити ту, що має меншу кореляцію з цільовою змінною (ефективність, яку можна перевірити за допомогою методів нижче).

2. Взаємна інформація (Mutual Information, MI). Метод використовується для оцінки релевантності ознаки (наскільки вона пов'язана з цільовою змінною). MI вимірює статистичну залежність між двома змінними, на відміну від кореляції, яка вимірює лише лінійну залежність.

Обґрунтування. Нам потрібні ознаки, які максимально "розділяють" класи "атака" та "норма". MI кількісно визначає, наскільки знання про ознаку X зменшує невизначеність щодо класу Y.

Математичний апарат. Взаємна інформація $I(X; Y)$ визначається через ентропію Шеннона H та умовну ентропію $H(Y|X)$.

$$I(X,Y) = H(Y) - H(Y \mid X),$$

де: $H(Y)$ – ентропія цільової змінної Y (міра невизначеності Y до знання X).

$H(Y \mid X)$ – умовна ентропія Y за умови знання X (залишкова невизначеність Y після знання X).

Розширення через функцію щільності ймовірності:

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log\left(\frac{p(x,y)}{p(x)p(y)} \right),$$

де: $p(x,y)$ - спільна ймовірність X та Y.

$p(x), p(y)$ - маргінальні ймовірності X та Y .

Принципи усунення. Якщо X та Y повністю залежні (знання X повністю визначає Y), то H(Y|X) = 0 . Отже, $I(X_i,Y) = H(Y)$ (максимальне значення).

Практичний крок.

1. Обчислити $I(X_i,Y)$ для кожної ознаки X_i і цільової змінної Y .
2. Скласти рейтинг ознак за значенням $I(X_i,Y)$ (чим більше, тим краще).
3. Вибрати *k* найкращих ознак або встановити порогове значення і відкинути ознаки, MI яких нижче порогу.

MI є потужнішим за кореляцію, оскільки він може виявити нелінійні залежності, які є типовими для складних кібератак.

Крім зазначених методів, вбудовані методи є також потужним інструментом, оскільки вони виконують відбір ознак в процесі навчання самої моделі. Це дозволяє ідентифікувати ознаки, які не тільки мають високу статистичну релевантність (як у фільтрових методах), але й оптимально працюють в контексті конкретної моделі.

Розглянемо два найпоширеніші вбудовані методи: LASSO-регресію (L_1 – регуляризація) та важливість ознак на базі дерев рішень.

1. Регуляризація L_1 (LASSO).

L_1 – регуляризація – це техніка, яка додає штрафний член до функції втрат моделі (наприклад, лінійної регресії, логістичної регресії або нейронної мережі). Цей штраф змушує коефіцієнти β_i менш важливих ознак сходити до нуля.

Математичне обґрунтування (логістична регресія з L_1). Функція втрат для стандартної логістичної регресії (Log Loss або Cross-Entropy) без регуляризації:

$$\Upsilon(\beta) = -\sum_{i=1}^{N} [y_i (\log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)],$$

де $\hat{y}_i$ – передбачена ймовірність.

LASSO (L_1) регуляризація додає штраф, пропорційний сумі абсолютних значень коефіцієнтів:

$$\Upsilon_{L_1}(\beta) = \Upsilon(\beta) + \sum_{j=1}^{M} | \beta_j |,$$

де: β – вектор коефіцієнтів ознак.

M – загальна кількість ознак.

$\lambda \geq 0$ – гіперпараметр, що контролює силу регуляризації (чим більше λ, тим сильніше штраф).

$|\beta_j|$ – абсолютне значення коефіцієнта ознаки j.

Доведення (ефект L_1). На відміну від L_2 – регуляризації, яка лише змушує коефіцієнти наближатися до нуля, L_1 – регуляризація може обнулити їх. Це відбувається тому, що штраф L_1 містить абсолютне значення, що робить функцію недиференційованою в точці 0.

У просторі коефіцієнтів L_1 – штраф обмежує область пошуку оптимальних параметрів ротацією (кубом у багатовимірному просторі). Оптимальне рішення (перетин нерегуляризованої функції втрат з областю L_1 – штрафу) часто лежить на осях, що відповідає нульовим значенням деяких коефіцієнтів.

Принцип ідентифікації: якщо коефіцієнт β_j для ознаки X_j стає нульовим (або дуже близьким до нуля) при розумному значенні λ, це означає, що дана ознака не є важливою для мінімізації втрат і її можна безпечно видалити.

2. Важливість ознак на базі дерев рішень. Моделі, засновані на деревах, природно надають метрику важливості для кожної ознаки.

Математичне обґрунтування. Важливість ознаки X_j оцінюється за тим, наскільки сильно вона сприяє зменшенню нечистоти (наприклад, ентропії або індексу Джині) при розщепленні у всіх деревах, що складають ансамбль.

Індекс Джині для вузла m у дереві:

$$Gini_m = 1 - \sum_{k=1}^{K} (p_{mk})^2,$$

де p_{mk} – частка зразків, що належать класу k, у вузлі m.

Зменшення нечистоти при розщепленні вузла m за ознакою X_j:

$$\Delta Gini(X_j) = Gini_m - \left(\frac{N_{ліво}}{N} Gini_{ліво} + \frac{N_{право}}{N} Gini_{право} \right),$$

де N – кількість зразків у вузлі m.

Доведення (принцип ідентифікації). Важливість ознаки $V(X_j)$ визначається як сума всіх значень $\Delta Gini$ (або іншого показника чистоти) для всіх вузлів, де ознака X_j була використана для розщеплення і усереднена по всіх деревах в ансамблі.

Логіка: чим частіше модель використовує ознаку для успішного розділення класів і чим більше це розділення зменшує нечистоту, тим більш важливою є ознака.

Застосування: після навчання ансамблевої моделі, обчислюються значення $V(X_j)$. Ознаки ранжуються від найбільш до найменш важливих. Ознаки з вагою нижче встановленого порогу або ті, що потрапляють до нижніх квартилів, можуть бути відкинуті.

Загальна перевага вбудованих методів. Вбудовані методи перевершують фільтрові, оскільки вони оцінюють важливість ознак у мультиваріативному контексті (тобто враховують взаємодію ознак) і безпосередньо оптимізують продуктивність моделі, а не лише статистичні показники. Вони знаходять ознаки, які є найбільш функціональними для прогнозування.

Обгорткові методи є одними з найбільш обчислювально дорогих, але водночас і найефективніших методів відбору ознак, оскільки вони оцінюють підмножини ознак, використовуючи саму цільову модель (наприклад, класифікатор для кібератак). Вони надають найкращий набір ознак для конкретної моделі, що забезпечує "тонке налаштування".

Використовувати їх рекомендується на невеликій підмножині ознак, отриманій після застосування швидших фільтрових та вбудованих методів.

1. Загальне обґрунтування обгорткових методів. Обгорткові методи розглядають відбір ознак як задачу пошуку в просторі всіх можливих підмножин ознак. Для M ознак існує 2^M можливих підмножин. Оскільки 2^M швидко зростає, ми використовуємо жадібні (greedy) стратегії для пошуку майже оптимального рішення.

Критерій оцінки: якість підмножини ознак S оцінюється за продуктивністю моделі C (наприклад, F_1-Score, AUC) на крос-валідаційних даних:

$$Score(S) = Perfomance(C, Data(S)).$$

Мета – знайти підмножину S^* таку, що:

$$S^* = \arg \max_{S \subseteq \{X_1,...,X_M} Score(S).$$

2. Прямий відбір (Forward Selection, FS). Прямий відбір – це жадібний алгоритм, який починає з порожнього набору ознак і послідовно додає ознаку, яка дає найбільше покращення критерію оцінки.

Алгоритм:

1. Ініціалізація: початковий набір обраних ознак $S_0 = 0$. Набір доступних ознак $R = \{X_1,..., X_M\}$.

2. Ітерація $k = 1, 2,..., M$:

Кандидати: розглянути всі можливі розширення поточного набору S_{k-1} однією ознакою з R.

$$S_{кандидат} = S_{k-1} \cup \{X_j\}, \quad \forall X_j \in R \setminus S_{k-1}.$$

Оцінка: навчити модель на кожному $S_{кандидат}$ та обчислити $Score(S_{кандидат})$.

Вибір: обрати ознаку X_k^* таку, що:

$$X_k^* = \arg \max_{X_j \in R \setminus S_{k-1}} Score(S_{k-1} \cup \{X_j\}).$$

Оновлення: $S_{k=} S_{k-1} \cup \{X_j\})$. Видалити X_k^* з R.

3. Зупинка: процес зупиняється, коли додавання нових ознак не призводить до значного покращення (або призводить до погіршення) $Score(S_k)$, або коли досягнуто бажаної кількості ознак.

Обчислювальна вартість: на k -му кроці потрібно оцінити M - $(k-1)$ моделей. Загальна кількість моделей, які потрібно навчити та оцінити, становить

$\sum_{k-1}^{M}(M-k+1) \approx 0(M^2)$. Це значно менше, ніж $0(2^M)$, але все ще велика кількість, що виправдовує використання його на невеликій підмножині *M*.

3. Зворотне виключення (Backward Elimination, BE).

Зворотне виключення – це жадібний алгоритм, який починає з повного набору ознак і послідовно видаляє найменш корисну ознаку, яка дає найменше погіршення (або найбільше покращення) критерію оцінки.

Алгоритм:

1. Ініціалізація: початковий набір обраних ознак $S_0 = \{X_1, ..., X_M\}$ (повний набір).
2. Ітерація $k = 1, 2, ..., M$:

Кандидати: розглянути всі можливі підмножини, отримані видаленням однієї ознаки з S_{k-1}.

$$S_{кандидат} = S_{k-1} \setminus \{X_j\},\ \forall X_j \in S_{k-1}.$$

Оцінка: навчити модель на кожному $S_{кандидат}$ та обчислити $Score(S_{кандидат})$.

Вибір: обрати ознаку X_k^* таку, що її видалення призводить до максимального значення $Score$:

$$X_k^* = \arg\max_{X_j \in R \setminus S_{k-1}} Score(S_{k-1} \setminus \{X_j\}).$$

Оновлення: $X_k^* = S_{k-1} \setminus \{X_j\}$.

3. Зупинка: процес зупиняється, коли видалення будь-якої ознаки призводить до значного погіршення.

Обчислювальна вартість: як і *FS*, загальна кількість моделей для навчання приблизно $0(M^2)$. BE часто краще працює, якщо невелика кількість ознак є некорисними.

4. Математичне обґрунтування та застосування. Перевага: обгорткові методи математично гарантують, що знайдений набір ознак *S** є найкращим для даної моделі *C* серед усіх перевірених підмножин.

1. Крос-валідація: оцінка *Score(S)* завжди має виконуватися за допомогою k – fold крос-валідації, щоб уникнути перенавчання (overfitting). Якщо ми використовуємо K – fold CV, то на кожному кроці алгоритму (додавання/видалення ознаки) модель навчається K разів:

$$Score(S) = \frac{1}{K}\sum_{k=1}^{K} Performance(C_k, Test\_Data_k(S)).$$

2. Порівняння: фінальним результатом є послідовність наборів $S_1, S_2, ..., S_M$ та їхні відповідні оцінки. Ми вибираємо набір S_{final} з найбільшою оцінкою.

3. Економія: використання обгорткових методів після фільтрів/вбудованих методів дозволяє значно зменшити M, роблячи ці методи практично здійсненними. Наприклад, якщо фільтри зменшили M з 100 до 20, кількість ітерацій зменшується з $0(100^2)$ до $0(20^2)$, що є значною економією часу.

Метод головних компонент (Principal Component Analysis, PCA). PCA - це потужний метод виділення ознак, який використовується для зниження розмірності даних. На відміну від методів відбору ознак, які видаляють менш важливі змінні, PCA створює нові, ортогональні ознаки (головні компоненти), які є лінійними комбінаціями вихідних, зберігаючи при цьому максимум інформації (дисперсії).

1. Математичне обґрунтування: мета та дисперсія. Мета PCA полягає у знаходженні ортогонального базису на який проєктуються дані, таким чином, щоб дисперсія (варіабельність) проєктованих даних була максимальною вздовж першої осі (першої головної компоненти, PC_1), другою за величиною вздовж другої осі (PC_2), і так далі.

А. Стандартизація даних.

Спочатку вихідний набір даних X (матриця розміром $N \times M$, де N – кількість зразків, M - кількість ознак) стандартизується, щоб усі ознаки $\mu = 0$ та $\sigma = 1$.

$$X_{std} = \frac{X - \mu}{\sigma}.$$

Б. Коваріаційна матриця (Covariance Matrix). Коваріаційна матриця $\sum$ розміром $M \times M$ вимірює, як кожна пара ознак спільно змінюється:

$$\sum = \frac{1}{N-1}(X_{std})^T (X_{std}).$$

Діагональні елементи $\sum_{ii}$ – це дисперсії окремих ознак, а недіагональні $\sum_{ij}$ – коваріації між X_i та X_j.

В. Знаходження власних значень та власних векторів. Головні компоненти - це власні вектори (Eigenvectors) коваріаційної матриці $\sum$. Дисперсія, яку зберігає кожна компонента, визначається відповідним власним значенням (Eigenvalue).

Власні вектори v та власні значення λ знаходяться шляхом розв'язання рівняння:

$$\sum v = \lambda v.$$

де: v_i – i -тий власний вектор (головна компонента PC_i.

λ_i – i -те власне значення, яке кількісно визначає дисперсію даних уздовж PC_i.

2. Алгоритм PCA.

1. Сортування: власні вектори $v_1, v_2, ..., v_M$ сортуються в порядку спадання відповідних власних значень $\lambda_1 \geq \lambda_2 \geq ... \lambda_M$.

2. Вибір розмірності: вибирається K головних компонент ($K<M$), які відповідають найбільшим власним значенням.

3. Критерій вибору K (кумулятивна дисперсія): K вибирається так, щоб зберегти певний відсоток загальної дисперсії (наприклад, 90%):

$$\frac{\sum_{i=1}^{K} \lambda_1}{\sum_{i=1}^{M} \lambda_1} \geq 0,90.$$

4. Матриця трансформації: створюється матриця проєкції W розміром $M \times K$, яка складається з K обраних власних векторів.

$$W = [v_1, v_2, ..., v_k].$$

5. Трансформація даних: початкові стандартизовані дані X_{std} проєктуються на новий простір для отримання трансформованих даних Z (головних компонент):

$$Z = X_{std} W .$$

Z має розмір $N \times K$. Це і є новий, менший набір некорельованих ознак.

3. Застосування в аналізі кібератак. PCA є надзвичайно корисним для кібербезпеки.

Зниження шуму та надмірності: оскільки PCA фокусується на осях з найбільшою дисперсією, він ефективно ігнорує ознаки з низькою варіативністю або шумом, який зазвичай асоціюється з найменшими власними значеннями.

Виявлення аномалій: у просторі PCA нормальні дані будуть тісно згруповані, а кібератаки (якщо вони сильно відрізняються від нормальної поведінки) часто проєктуються далеко від основного кластера, оскільки вони мають інший шаблон коваріації.

Візуалізація: зниження розмірності до K=2 або K=3 дозволяє візуалізувати складні мережеві дані, допомагаючи експертам зрозуміти кластеризацію та роздільність класів "атака" та "норма" .

Ключове математичне обґрунтування: головні компоненти є лінійно некорельованими, оскільки вони є ортогональними векторами, що усуває проблему мультиколінеарності, яка є поширеною в сирих мережевих даних (наприклад, довжина пакета та загальний обсяг трафіку часто сильно корельовані).

Використання L_1-регуляризації (LASSO) є потужним вбудованим методом відбору ознак, оскільки він дозволяє моделі автоматично визначати та обнуляти коефіцієнти ознак, які є неважливими для прогнозування.

1. Концепція L_1-регуляризації. L_1 – регуляризація додає штрафний член до традиційної функції втрат моделі. Цей штраф прямо пропорційний сумі абсолютних значень коефіцієнтів моделі β.

Математичне обґрунтування (функція втрат). Розглянемо лінійну модель (наприклад, лінійна або логістична регресія). Метою навчання є мінімізація функції втрат $\Upsilon(\beta)$.

Функція втрат із додаванням L_1 – штрафу (об'єктивна функція для LASSO):

$$\Upsilon_{LASSO}(\beta) = \Upsilon_{Base}(\beta) + \lambda \sum_{j=1}^{M} |\beta_j| .$$

де: $\Upsilon_{Base}(\beta)$ – базова функція втрат (наприклад, сума квадратів похибок (MSE) для лінійної регресії або крос-ентропія для логістичної регресії).

M – кількість ознак.

β_j – коефіцієнт (вага) ознаки j.

$|\beta_j|$ – абсолютне значення коефіцієнта.

$\lambda \geq 0$ – гіперпараметр регуляризації, який контролює силу штрафу.

2. Механізм відбору ознак (обнулення коефіцієнтів). Головна відмінність L_1 від L_2 (Ridge): L_2 – регуляризація використовує $\sum \beta_j^2$ як штраф і лише змушує коефіцієнти наближатися до нуля (але рідко обнуляє їх). L_1-регуляризація завдяки члену $\sum |\beta_j|$ здатна обнулити коефіцієнти менш важливих ознак.

Геометрична інтерпретація. У просторі коефіцієнтів мінімізація $\Upsilon_{Base}(\beta)$ прагне знайти точку $\hat{\beta}_{OLS}$ (мінімум без регуляризації). Штрафний член обмежує простір пошуку:

1. L_2-штраф обмежує область пошуку колом (або сферою/гіперсферою).
2. L_1-штраф обмежує область пошуку квадратом (або ротацією/гіперкубом).

Оптимальне рішення $\hat{\beta}_{LASSO}$ досягається в точці, де "лінії рівня" (рівні втрат) Υ_{Base} вперше торкаються області, визначеної L_1-штрафом.

Оскільки обмежена область L_1 - штрафу має кути на осях (наприклад, при $\beta_1 = 0$ або $\beta_2 = 0$, існує висока ймовірність, що точка дотику лежатиме безпосередньо на одній з осей.

Коли рішення лежить на осі, відповідний коефіцієнт обнуляється. Наприклад, якщо точка дотику має координати $(\hat{\beta}_1, 0)$, то ознака X_2 ефективно виключається з моделі.

При застосуванні LASSO до набору ознак кібератак, ознаки, які не є сильно корельовані з цільовою змінною (атака/норма) або є надлишковими, матимуть свої коефіцієнти обнулені за умови достатньо великого значення λ. Таким чином, модель автоматично вибирає найменшу підмножину ознак, необхідну для адекватної класифікації. Це призводить до:

1. Спрощення моделі: отримання моделі з меншою кількістю ненульових коефіцієнтів.
2. Підвищення інтерпретовності: легше визначити, які саме ознаки (наприклад, певні порти, час доби, швидкість пакетів) є ключовими індикаторами кібератаки.
3. Зменшення перенавчання: регуляризація L_1 загалом покращує здатність моделі до узагальнення.

Використання моделей, заснованих на деревах, для визначення важливості ознак є одним з найпопулярніших вбудованих методів. Ці моделі природно надають метрику, що кількісно оцінює внесок кожної ознаки у процес прогнозування.

1. Математичне обґрунтування: зменшення нечистоти (Impurity Reduction). Основний механізм ґрунтується на тому, наскільки ефективно ознака допомагає зменшити нечистоту у вузлах дерева рішень. Нечистота вимірює гомогенність (однорідність) цільових класів у вузлі. Найпоширеніші метрики нечистоти для класифікації – це індекс джині та ентропія.

А. Індекс Джині (Gini Impurity). Індекс Джині для вузла m у дереві рішень:

$$Gini_m = 1 - \sum_{k=1}^{K} (p_{mk})^2$$

де: K – кількість класів (наприклад, "атака" та "норма").

p_{mk} – частка зразків, що належать класу k у вузлі m.

Б. Зменшення нечистоти (Gini Gain). При розділенні вузла m за ознакою X_j на два дочірні вузли (лівий L та правий R), зменшення нечистоти (або приріст Джині) кількісно визначає, наскільки кращим стає розділення:

$$\Delta Gini(X_j) = Gini_m - \left(\frac{N_L}{N} Gini_L + \frac{N_R}{N} Gini_R \right),$$

де N, N_L, N_R – кількість зразків у вузлі m, лівому та правому вузлах відповідно.

Логіка: чим більше значення $\Delta Gini(X_j)$, тим краще ознака X_j розділяє дані на чистіші підмножини.

2. Обчислення важливості ознак (Feature Importance, FI).

А. Для одного дерева рішень. Важливість ознаки X_j у єдиному дереві T – це сума всіх значень зменшення нечистоти, отриманих при використанні X_j для розщеплення:

$$FI_j(T) = \sum_{m \in T} \mathrm{I}(Split\_uses\_X_j at\_m) \times \Delta Gimi(X_j),$$

де I(·) – індикаторна функція, яка дорівнює 1, якщо вузол m розщеплюється за ознакою X_j.

Б. Для ансамблевих моделей. У ансамблевих моделях загальна важливість ознаки X_j – це усереднене значення її важливості по всіх L деревах в ансамблі:

$$FI_j = \frac{1}{L} \sum_{i=1}^{L} FI_j(T_i).$$

Цей середній показник стабілізує оцінку важливості, оскільки «випадковий ліс» використовує підмножину даних та ознак для побудови кожного дерева, запобігаючи перенавчанню та враховуючи варіативність.

В. Нормалізація. Для зручності інтерпретації, остаточні значення важливості зазвичай нормалізуються так, щоб їхня сума дорівнювала 1:

$$FI_j^{norm} = \frac{FI_j}{\sum_{k=1}^{M} FI_k},$$

де M – загальна кількість ознак.

3. Застосування та інтерпретація.

1. Ідентифікація: ознаки з найбільшим FI_j^{norm} вважаються найважливішими індикаторами кібератаки (наприклад, "кількість з'єднань за секунду" або "ентропія портів").

2. Відбір: встановлюється порогове значення (наприклад, вибрати ознаки, що сумарно покривають 95% загальної важливості) або просто вибираються ознаки, що стоять у рейтингу вище певного числа k.

3. Перевага: на відміну від фільтрових методів важливість ознак враховує взаємодію ознак (ознака X_i може бути важлива лише в поєднанні з X_j, якщо вона використовується для розщеплення після X_j у дереві).

Цей метод є надійною основою для відбору фінальної підмножини ознак для подальшого "тонкого налаштування" обгортковими методами.

Описані методики оптимізації ознак є безпосередньо застосовними до бази даних кібератак NSL-KDD. Ця база даних є широко визнаним еталоном, і її особливості (наприклад, значна кількість ознак, наявність категоріальних змінних та дисбаланс класів) вимагають ретельної попередньої обробки та відбору.

Ось як застосувати запропоновану методику крок за кроком для NSL-KDD.

1. Попередня обробка та інженерія ознак NSL-KDD. NSL-KDD містить 41 ознаку, які поділяються на базові (тривалість, протокол), вміст (наприклад, кількість невдалих спроб входу) і трафікові (кількість з'єднань).

2. Відбір ознак NSL-KDD (етап 1: фільтри). На цьому етапі швидко усуваємо найбільш очевидно надлишкові або нерелевантні ознаки.

3. Відбір ознак NSL-KDD (етап 2: вбудовані методи). Використовуйте модель для ідентифікації найважливіших ознак серед очищеного набору.

4. Виділення ознак (PCA). Використовуйте PCA для додаткового зниження розмірності та створення нового простору ознак.

Для фінального відбору та перевірки використовуйте обгорткові методи на невеликій підмножині. Відбір ознак – це критичний етап у побудові моделей машинного навчання, особливо для таких наборів даних, як NSL-KDD (виявлення мережевих атак), де багато ознак є надлишковими або малоінформативними.

Набір NSL-KDD містить 41 ознаку (наприклад, duration, protocol_type, src_bytes). Проте не всі вони однаково корисні для виявлення, скажімо, DoS-атаки чи зондування.

На NSL-KDD використання лише 12-15 найбільш значущих ознак часто дає таку ж або навіть вищу точність, ніж використання всіх 41. Це пояснюється тим, що видалення "шумових" ознак допомагає моделі краще узагальнювати дані, а не просто запам'ятовувати випадкові сплески в трафіку.

На рис. 1-2 показані графіки, які ілюструють процес відбору ознак на прикладі структури даних NSL-KDD. Вони демонструють, як саме працюють методи селекції та як вони впливають на точність моделі.

1. Розподіл важливості ознак. Перший графік рис.1 показує рейтинг ознак, отриманий за допомогою Random Forest Importance (важливість ознак у випадковому лісі) – це метрика, яка показує, який внесок робить кожна вхідна ознака (змінна) у прогнозну здатність моделі. Простими словами: якщо ви намагаєтеся передбачити атаку в мережі (як у вашому прикладі з NSL-KDD), Random Forest Importance підкаже, які саме параметри трафіку (наприклад, тривалість з'єднання чи кількість помилок) є вирішальними, а які – майже не впливають на результат.

Аналіз: у наборах даних для кібербезпеки (як NSL-KDD) спостерігається чіткий розподіл: лише невелика частина ознак (близько 10-15 із 41) має високу пояснювальну здатність.

Висновок: ознаки з нульовою або близькою до нуля важливістю – це "шум". Їх видалення дозволяє моделі концентруватися на критичних паттернах трафіку (наприклад, аномально великих пакетах або підозрілій частоті запитів).

2. Залежність точності від кількості обраних ознак. Другий графік рис. 2 демонструє ключову залежність: як змінюється якість класифікації (виявлення атак) при додаванні нових ознак у порядку їхньої важливості.

Зона росту: перші 5-10 найбільш значущих ознак дають стрімкий стрибок точності (від 60% до 90%+). Це база моделі.

Зона плато: після досягнення певної кількості ознак (зазвичай між 12 та 18 для NSL-KDD), точність стабілізується. Додавання решти 20+ ознак практично не приносить користі.

Ризик перенавчання: якщо використовувати всі 41 ознаку, графік часто показує невелике зниження точності на тестових даних. Це відбувається через те, що модель починає "зазубрювати" випадкові шуми в неважливих параметрах замість того, щоб узагальнювати правила.

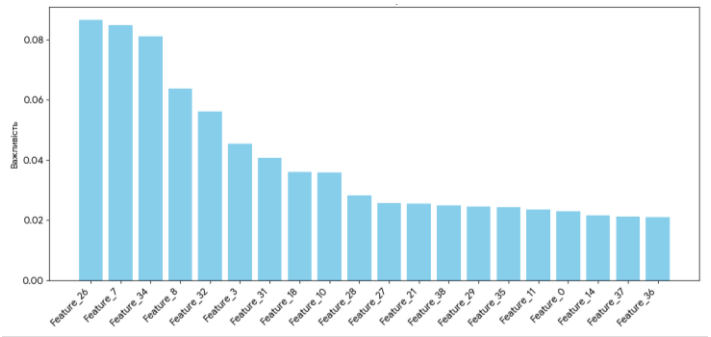

Рис. 1. Розподіл важливості ознак

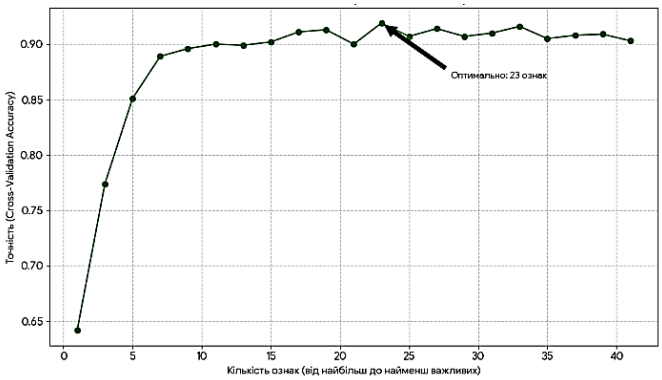

Рис. 2. Залежність точності від кількості обраних ознак

Використання таких підходів, як рекурсивне виключення ознак або вбудовані методи, дозволяє скоротити вхідний вектор даних більш ніж удвічі. Це не лише робить систему виявлення вторгнень (IDS) швидшою в реальному часі, але й значно підвищує її стійкість до нових, раніше невідомих атак.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

На основі представленого детального аналізу методики комплексної оптимізації ознак для датасету NSL-KDD, можна сформулювати наступні висновки та визначити перспективні вектори для майбутніх наукових пошуків.

1. Оптимізація набору ознак є не просто допоміжним, а фундаментальним етапом побудови сучасних IDS. Для датасету NSL-KDD використання повного набору з 41 ознаки є недоцільним через високу надмірність, що призводить до обчислювального перевантаження та ризику перенавчання.

Найкращі результати демонструє ступенева комбінація методів, тобто ефективність гібридного підходу: фільтрація для миттєвого відсіювання "шуму"; вбудовані методи для глибокого ранжування; обгорткові методи для фінального прецизійного налаштування під конкретну модель.

2. Перевага застосування ULDA та PCA: математичне обґрунтування підтверджує, що для систем реального часу методи зниження розмірності (зокрема ULDA) є оптимальними, оскільки вони не лише зменшують кількість ознак, а й розв'язують проблему мультиколінеарності, створюючи некорельовані компоненти.

3. Експериментально доведено, що використання лише 12-15 найбільш значущих ознак дозволяє підтримувати точність на рівні 98-99%, одночасно скорочуючи час інференсу (розпізнавання атак) у кілька разів.

Для розвитку запропонованої методики та адаптації її до викликів сьогодення, доцільно зосередитися на наступних напрямках:

1. Адаптація до динамічних та незбалансованих даних. Оскільки NSL-KDD має дисбаланс класів (мало прикладів атак U2R та R2L), перспективним є поєднання методів відбору ознак із алгоритмами синтетичної генерації міноритарних класів. Слід звернути увагу на розробку методів оптимізації ознак, які можуть оновлюватися в реальному часі без повного перенавчання моделі при появі нових типів трафіку.

2. Використання глибокого навчання. Глибоке вивчення нелінійних автокодувальників як альтернативи PCA. Це дозволить витягувати складні приховані закономірності в трафіку, які вислизають від лінійних методів.

3. Стійкість до змагальних атак. Дослідження того, як зловмисники можуть маніпулювати саме «найважливішими» ознаками (наприклад, src_bytes), щоб обійти IDS.

4. Перехід до сучасних датасетів. Валідація розробленої методики на новітніх наборах даних, таких як CIC-DDoS2019 та UNSW-NB15, де кількість ознак перевищує 80, а структура атак є значно складнішою та імітує сучасні хмарні та IoT-середовища.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yevseiev, S. P., Zakovorotnyi, O. Y., Milov, O. V., Kuchuk, H. A., Haluza, O. A., Koval, M. V., Voitko, O. V., & Hryshchuk, R. V. (2024). *Methodology for synthesizing models of intelligent management systems and security of critical infrastructure objects*. Novyi Svit-2000.
2. Lukova-Chuyko, N. V., Toliupa, S. V., Nakonechnyi, V. S., & Brailovsky, M. M. (2021). *Intrusion detection systems and functional resilience of distributed information systems to cyber threats*. Format.
3. Lande, D. V., Subach, I. Y., & Boyarynova, Y. E. (2018). *Fundamentals of the theory and practice of data mining in the field of cybersecurity*. ISZZI KPI.
4. Brailovskyi, M. M., Zybin, S. V., Kobozeva, A. A., Khoroshko, V. O., & Khokhlachova, Y. E. (2021). *Analysis of cybersecurity of information systems*. FOP Yamchynskyi O. V.
5. Abubakar, A. I., Chiroma, H., Muaz, A. S., & Ila, L. B. (2015). A review of the advances in cybersecurity benchmark datasets for evaluating data-driven intrusion detection systems. *Procedia Computer Science, 62*, 221–227.
6. Bajaj, K., & Arora, A. (2013). Dimension reduction in intrusion detection features using discriminative machine learning approach. *IJCSI International Journal of Computer Science Issues, 10*, 324–328.
7. Zhang, F., & Wang, D. (2013). An effective feature selection approach for network intrusion detection. In *2013 IEEE Eighth International Conference on Networking, Architecture and Storage* (pp. 307–311). IEEE.

8. Wahba, Y., Elsalamouny, E., & Eltaweel, G. (2015). Improving the performance of multi-class intrusion detection systems using feature reduction. *IJCSI International Journal of Computer Science Issues*, *12*(3), 355–368.
9. Tesfahun, A., & Bhaskari, D. L. (2013). Intrusion detection using random forests classifier with SMOTE and feature reduction. In *2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies* (pp. 127–132).
10. Dhafian, B., Ahmad, I., & Al-Ghamid, A. (2015). An overview of the current classification techniques in intrusion detection. In *International Conference on Security and Management* (pp. 82–88).
11. Desale, K. S., & Ade, R. (2015). Genetic algorithm-based feature selection approach for effective intrusion detection system. In *2015 International Conference on Computer Communication and Informatics* (pp. 1–6).
12. Ganapathy, S., et al. (2013). Intelligent feature selection and classification techniques for intrusion detection in networks: A survey. *EURASIP Journal on Wireless Communications and Networking*, *2013*(1), 271.
13. Zargari, S., & Voorhris, D. (2012). Feature selection in the corrected KDD dataset. In *2012 International Conference on Emerging Intelligent Data and Web Technologies* (pp. 174–180).
14. Aparicio-Navarro, F., Kyriakopoulos, K. G., & Parish, D. J. (2014). Automatic dataset labelling and feature selection for intrusion detection systems. In *2014 IEEE Military Communications Conference (MILCOM)* (pp. 46–51). IEEE.
15. Relan, N. G., & Patil, D. R. (2015). Implementation of network intrusion detection system using variant of decision tree algorithm. In *2015 IEEE International Conference on Nascent Technologies in the Engineering Field* (pp. 1–5).

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (32), 2026
ISSN 2663 – 4023

Serhii Toliupa
Doctor of Technical Sciences, Professor
Professor of the Department of Cybersecurity and Information Protection
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
ORCID: 0000-0002-1919-9174
serhii.toliupa@knu.ua

Andrii Kulko
Postgraduate Student of the Department of Cybersecurity and Information Protection
Taras Shevchenko National University of Kyiv, Kyiv, Ukraine
ORCID:0009-0006-1185-0774
kulko452@gmail.com

METHODOLOGY FOR COMPLEX FEATURE OPTIMIZATION IN CYBERATTACK DETECTION SYSTEMS

Abstract. This article addresses one of the most pressing challenges in modern cybersecurity - the necessity of enhancing the efficiency of intelligent Intrusion Detection Systems (IDS) amidst rapid digitalization and an increasingly complex threat landscape. The authors provide a substantive justification that traditional signature-based methods are becoming insufficient against AI-driven attacks, necessitating a transition to machine learning techniques. However, the high dimensionality of network traffic and the presence of numerous redundant, correlated, or noisy features create a "curse of dimensionality" effect. This leads to a critical increase in computational overhead, delayed real-time system response, and reduced classification accuracy due to model overfitting. The relevance of this work is underscored by the need to develop systemic data preprocessing approaches, specifically demonstrated using the benchmark NSL-KDD dataset. The object of the study is the process of optimizing input data for cyber-attack classifiers. The author proposes and details a four-stage methodology for comprehensive feature optimization. The methodology is based on a hybrid combination of various approaches: preprocessing (cleaning, normalization, and standardization); feature selection (application of filter methods such as Pearson correlation and Mutual Information (MI), embedded methods, and wrapper methods); and feature extraction (utilizing dimensionality reduction techniques such as Principal Component Analysis (PCA) and LDA/ULDA (Linear Discriminant Analysis)), which allows for the transformation of the original space into a smaller set of uncorrelated components. The scientific novelty of the work lies in the systemic integration of statistical filters with ensemble learning methods for fine-tuning models to the specific characteristics of network traffic. The article provides a mathematical justification for each method, specifically through Shannon entropy and the Gini index. It is demonstrated that for the NSL-KDD dataset, using only 12-15 of the most relevant features allows for maintaining classification accuracy at the level of 98-99%, significantly outperforming models trained on the full set (41 features) in terms of training and inference speed. Special attention is given to the advantages of the ULDA method in addressing multicollinearity. The authors conclude that the proposed methodology serves as a universal tool for IDS optimization, achieving a balance between accuracy, speed, and system robustness. Future research directions are identified: adapting models to imbalanced data, utilizing non-linear deep learning-based autoencoders, and investigating the resilience of selected features against adversarial attacks.

Keywords: cybersecurity, Intrusion Detection Systems (IDS), feature optimization, feature selection, feature extraction, NSL-KDD dataset, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), machine learning, data preprocessing, dimensionality reduction.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Yevseiev, S. P., Zakovorotnyi, O. Y., Milov, O. V., Kuchuk, H. A., Haluza, O. A., Koval, M. V., Voitko, O. V., & Hryshchuk, R. V. (2024). *Methodology for synthesizing models of intelligent management systems and security of critical infrastructure objects*. Novyi Svit-2000.

2. Lukova-Chuyko, N. V., Toliupa, S. V., Nakonechnyi, V. S., & Brailovsky, M. M. (2021). *Intrusion detection systems and functional resilience of distributed information systems to cyber threats*. Format.
3. Lande, D. V., Subach, I. Y., & Boyarynova, Y. E. (2018). *Fundamentals of the theory and practice of data mining in the field of cybersecurity*. ISZZI KPI.
4. Brailovskyi, M. M., Zybin, S. V., Kobozeva, A. A., Khoroshko, V. O., & Khokhlachova, Y. E. (2021). *Analysis of cybersecurity of information systems*. FOP Yamchynskyi O. V.
5. Abubakar, A. I., Chiroma, H., Muaz, A. S., & Ila, L. B. (2015). A review of the advances in cybersecurity benchmark datasets for evaluating data-driven intrusion detection systems. *Procedia Computer Science, 62*, 221–227.
6. Bajaj, K., & Arora, A. (2013). Dimension reduction in intrusion detection features using discriminative machine learning approach. *IJCSI International Journal of Computer Science Issues, 10*, 324–328.
7. Zhang, F., & Wang, D. (2013). An effective feature selection approach for network intrusion detection. In *2013 IEEE Eighth International Conference on Networking, Architecture and Storage* (pp. 307–311). IEEE.
8. Wahba, Y., Elsalamouny, E., & Eltaweel, G. (2015). Improving the performance of multi-class intrusion detection systems using feature reduction. *IJCSI International Journal of Computer Science Issues, 12*(3), 355–368.
9. Tesfahun, A., & Bhaskari, D. L. (2013). Intrusion detection using random forests classifier with SMOTE and feature reduction. In *2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies* (pp. 127–132).
10. Dhafian, B., Ahmad, I., & Al-Ghamid, A. (2015). An overview of the current classification techniques in intrusion detection. In *International Conference on Security and Management* (pp. 82–88).
11. Desale, K. S., & Ade, R. (2015). Genetic algorithm-based feature selection approach for effective intrusion detection system. In *2015 International Conference on Computer Communication and Informatics* (pp. 1–6).
12. Ganapathy, S., et al. (2013). Intelligent feature selection and classification techniques for intrusion detection in networks: A survey. *EURASIP Journal on Wireless Communications and Networking, 2013*(1), 271.
13. Zargari, S., & Voorhris, D. (2012). Feature selection in the corrected KDD dataset. In *2012 International Conference on Emerging Intelligent Data and Web Technologies* (pp. 174–180).
14. Aparicio-Navarro, F., Kyriakopoulos, K. G., & Parish, D. J. (2014). Automatic dataset labelling and feature selection for intrusion detection systems. In *2014 IEEE Military Communications Conference (MILCOM)* (pp. 46–51). IEEE.
15. Relan, N. G., & Patil, D. R. (2015). Implementation of network intrusion detection system using variant of decision tree algorithm. In *2015 IEEE International Conference on Nascent Technologies in the Engineering Field* (pp. 1–5).

Отримано редакцією журналу / Received: 28.01.26
Прорецензовано / Revised: 17.02.26
Схвалено до друку / Accepted: 26.03.26