КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (19), 2023
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2023.19.209225
УДК 004.89:[004.056:007]

Чичкарьов Євген
доктор технічних наук, професор, професор кафедри штучного інтелекту
Державний університет інформаційно-комунікаційних технологій, м. Київ, Україна
ORCID 0000-0002-4362-5129
chychkarovea@gmail.com

Зінченко Ольга
доктор технічних наук, доцент, завідувач кафедри штучного інтелекту
Державний університет інформаційно-комунікаційних технологій, м. Київ, Україна
ORCID 0000-0002-3973-7814
zinchenkoov@gmail.com

Бондарчук Андрій
доктор технічних наук, професор, директор навчально-наукового інституту інформаційних технологій
Державний університет інформаційно-комунікаційних технологій, м. Київ, Україна
ORCID 0000-0002-7309-4365
dekan.it@ukr.net

Асєєва Людмила
аспірант
Державний університет інформаційно-комунікаційних технологій, м. Київ, Україна
ORCID 0000-0001-5954-4211
aseewal@i.ua

ВИЯВЛЕННЯ МЕРЕЖЕВИХ ВТОРГНЕНЬ З ВИКОРИСТАННЯМ АЛГОРИТМІВ МАШИННОГО НАВЧАННЯ І НЕЧІТКОЇ ЛОГІКИ

Анотація. У дослідженні була запропонована модель системи виявлення вторгнень на основі машинного навчання з використанням вибору ознак у великих наборах даних на основі методів ансамблевого навчання. Для вибору необхідних ознак було використано статистичні тести та нечіткі правила. При виборі базового класифікатора було досліджено поведінку 8 алгоритмів машинного навчання. Запропонована система забезпечила скорочення часу виявлення вторгнень (до 60%) та високий рівень точності виявлення атак. Найкращі результати класифікації для усіх досліджених наборів даних забезпечили класифікатори на основі дерев: DecignTreeClassifier, ExtraTreeClassifier, RandomForestClassifier. При відповідному налаштуванні обрання Stacking або Bagging класифікатора для навчання моделі з використанням усіх наборів даних забезпечує невеличке підвищення точності класифікацій, але суттєво збільшує час навчання (більш ніж на порядок, в залежності від базових класифікаторів або кількості підмножин даних). При збільшенні кількості спостережень в наборі даних для навчання ефект зростання часу навчання стає більш помітним. Найкращі показники за швидкістю навчання забезпечив класифікатор VotingClassifier, побудований на базі алгоритмів з максимальною швидкістю навчання і достатньою точністю класифікації. Час навчання класифікатора з використанням FuzzyLogic практично не відрізняється від часу навчання вотуючого класифікатора (більше приблизно на 10–15%). Вплив кількості ознак на час навчання класифікаторів і ансамбля VotingClassifier залежить від поведінки базових класифікаторів. Для ExtraTreeClassifier час навчання слабко залежить від кількості ознак. Для DesignTree або KNeibors (і, як наслідок, для класифікатора Voting в цілому) час навчання помітно зростає зі збільшенням кількості ознак. Зменшення кількості ознак на усіх наборах даних впливає на точність оцінювання відповідно до критерію середнього зменшення помилок класифікації. Поки група ознак в наборі даних для навчання містить перши за списком ознаки з найбільшим впливом, точність моделі знаходиться на початковому рівні, але при виключенні з моделі хоча б однієї з ознак з великим впливом, точність моделі стрибкоподібно знижується.

Ключові слова: система виявлення вторгнень; машинне навчання; ансамблеве навчання; класифікатор; нечітка логіка; кібератака; кіберзахист з використанням машинного навчання; алгоритми обрання ознак.

ВСТУП

Виявлення вторгнень є важливою частиною мережевої безпеки боротьби з незаконним доступом до мережі чи зловмисними кібератаками. Щоб протидіяти цім атакам, було розроблено багато інструментів і механізмів для мережевої та хмарної безпеки, включаючи різні системи виявлення вторгнень (Intrusion Detection System — IDS).

Впровадження методів машинного навчання у розробку IDS широко вивчалося протягом останнього десятиріччя. Моделі машинного навчання (ML) показали перспективні результати в прогнозуванні або класифікації даних у багатьох областях досліджень, і в контексті IDS ML використовується для класифікації того, чи є трафік безпечним чи атакуючим [1].

Більшість досліджень використовують кілька моделей ML та кілька наборів даних для оцінки. Проте, в багатьох дослідженнях оцінка з використанням різних наборів даних била виконана окремо. Різні набори даних мають різні набори ознак, тому моделі для обробки різних наборів даних треба перенавчати. Для підвищення продуктивності IDS використовувались різні методи обрання ознак (Feature Selection — FS) і зменшення розмірности (Dimensionality Reduction — DR).

З метою підвищення надійності виявлення атак в складі IDS використовують ансамблеві класифікатори, які дозволяють об’єднати набір окремих алгоритмів класифікації і ухвалити найкраще рішення про класифікацію об’єкта, який з’явився на вході системи. Ансамблеве навчання спричинило значне поліпшення порівняно з індивідуальними класифікаторами, але результат залежить від різних факторів, таких як базові класифікатори, схеми голосування тощо [2]. Переважна більшість робіт з ансамблевого навчання користуються однорідним ансамблем, де random forest, bagging, boosting були найпоширенішими методами ансамблю [3]. Крім того, використовувалися мажоритарне голосування та стекова архітектура, особливо коли розглядалися різнорідні класифікатори. Більшість проаналізованих статей вважають за краще використовувати схему мажоритарного голосування. Існує великий інтерес до застосування для IDS класифікатора випадкового лісу, тому що його реалізація різноманітна і застосування неважке [3].

Постановка проблеми. Таким чином, залишається відкритим питання про найбільш ефективний варіант ансамблевого навчання стосовно побудови аналітичного блоку систем виявлення вторгнень, а також доцільність включення до складу базових класифікаторів ансамблю тих чи інших алгоритмів класифікації. Важливим питанням також є раціональний алгоритм обрання ознак і зменшення розмірності навчального набору даних.

Ця робота присвячена розробці вдосконаленої IDS, яка забезпечує високу точність за рахунок застосування методів вибору ознак та ансамблевого навчання.

Аналіз останніх досліджень і публікацій. Загалом під час використання IDS з моделями ML досягається точність понад 90%.

У ряді робіт зусилля дослідників були зосереджені на порівнянні декількох моделей з навчанням на різних наборах даних.

У роботі [4] було побудовано десять моделей для IDS, включаючи як

контрольовані, і неконтрольовані моделі машинного навчання. Автори оцінили кілька моделей машинного навчання, включаючи алгоритми, що базуються не нейронних мережах, класифікатори KNN та SVM. Вони представили детальне порівняльне. Результат дослідження з класифікації набору даних CIDDS-001 показав, що KNN, SVM, Decision Tree (DT), RandomForest (RF) та мережа глибокого навчання є найефективнішими моделями з рівнем точності вище 99,9%.

У роботі [5] узагальнено результати понад 40 робіт, в яких реалізовано глибоке навчання для IDS, та описано 35 відомих наборів даних в області IDS. Автори також реалізували сім моделей глибокого навчання та порівняли продуктивність з підходами NaiveBayes, нейронної мережи, SVM та RandomForest. У дослідженні були використані набори даних CSE-CIC-IDS2018 та Bot-IoT. На думку [5] моделі глибокого навчання досягли рівня виявлення 95%, що перевищує рівень виявлення 90% досягнутий іншими моделями.

У роботі [6] порівнювалася продуктивності SVM, KNN та DT з використанням декількох наборів даних (CSE-CIC-IDS2018, UNSW-NB15, ISCX-2012, NSLKDD та CIDDS-001). Результати дослідження показали, що точність моделей варіювалася від 95% до 100%, крім набору даних UNSW-NB15. DT незмінно показує найкращі результати серед усіх реалізованих моделей незалежно від набору даних.

Багато уваги дослідники систем виявлення вторгнень приділили методам вибору необхідних ознак та зменшення розмірності навчального набору даних.

У роботі [7] досліджено ефективні методи вибору ознак поліпшення виявлення вторгнень з допомогою методів машинного навчання. Для аналізу сукупних ознак використовувалися різні методи класифікації (SVM, Decision Tree, Naive Bayes) для вибору ознак з високим рейтингом, які об’єднувалися і передавалися в класифікатор штучної нейронної мережі для того, щоб розрізнити напад і нормальну поведінку. Результати експериментів показують високу точність і ефективність запропонованого методу.

У роботі [8] використано гібридний вибір ознак на основі правил та нейронна мережа глибокого навчання, досягнута точність розпізнавання склала 99,0% для NSL-KDD та 98,9% для UNSW-NB15.

У роботі [9] для зменшення розмірності даних були використані методи вибору ознак з урахуванням кореляції (Correlation-based Feature Selection - CFS) і класифікатор Naive Bayes (NB). Пропонована система виявлення вторгнень класифікувала атаки з використанням багаторівневого персептрона (MLP) та алгоритму навчання на основі екземплярів. Точність впровадженої IDS склала 99,87% та 99,82% за наявності всього 5 та 3 ознак з 78.

У роботі [10] запропоновано ефективну систему виявлення вторгнень (IDS) для хмарного середовища, яка використовує методи вибору та класифікації ансамблю ознак. На думку авторів [10] пропонований метод на основі ансамблю ефективно визначає, чи є поведінка мережного трафіку нормальним або атакуючим.

Таким чином, використання відомих алгоритмів класифікації за умови вибору результативного набору ознак дає хороші результати точності розпізнавання атак, але однозначна оцінка найбільш ефективного алгоритму класифікації та способу відбору ознак відсутня.

Одним із напрямків досліджень, що стосуються побудови IDS, є розробка систем з використанням нечіткої логіки.

На думку [11], методи штучного інтелекту, такі як дерева рішень, нейронні мережі та нечітка логіка, застосовуються для виявлення підозрілих дій у мережі, при цьому

система на основі нечіткої інформації забезпечує значні переваги порівняно з іншими методами штучного інтелекту.

У дослідженні [12] проведено дослідження можливостей використання методів інтерполяції нечітких правил (FRI) у сфері додатків IDS. Проведені авторами експерименти показали, що модель FRI-IDS, яка була побудована з використанням розрідженої нечіткої ідентифікації моделі, досягла прийнятного рівня виявлення DDOS-атак. Модель FRI-IDS може інтерполювати висновки навіть у тому випадку, якщо деякі спостереження не покриваються безпосередньо нечіткими правилами.

У роботі [13] розробили метод динамічної інтерполяції нечітких правил для підвищення загальних можливостей системи, а також для ефективного виявлення атак.

У роботі [14] запропоновано новий метод, який поєднує аналіз головних компонентів та алгоритм нечіткої кластеризації з технікою вибору ознак K-найближчих сусідів. Для перевірки надійності моделі був використаний відомий набір даних NSL-KDD. Використання розробленого алгоритму дозволило підвищити точність класифікації і знизити високий рівень помилкових спрацьовувань.

Таким чином, використання нечітких алгоритмів кластеризації та методів інтерполяції нечітких правил забезпечує підвищення загальних можливостей системи виявлення вторгнень.

Мета статті. Мета дослідження — встановити можливість і оцінити параметри аналітичного блоку IDS з використанням методів машинного навчання та нечіткої логіки.

Для досягнення цієї мети необхідно вирішити декілька задач, а саме:

- встановити можливості аналітичного блоку IDS, які можуть бути досягнуті за рахунок використання ансамблевої методології;
- дослідити продуктивність і можливості різних алгоритмів класифікації для виявлення мережевих вторгнень;
- встановити, який набір класифікаторів забезпечує необхідну точність оцінювання;
- обрати найкращий метод обрання результатів класифікації.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Системи виявлення вторгнень зосереджуються на виявленні поведінки для обробки аномалій або атак. Вона розпізнає будь-який порушений мережевий трафік за цим шаблоном, який може призвести до атаки на інфраструктуру. Раціональний підхід полягає в тому, щоб навчати систему на цьому шаблоні поведінки за допомогою кількох методів. Для комбінування різних методів класифікації та прогнозу в системах виявлення вторгнень досить широко використовують алгоритми ансамблевого навчання.

Метод ансамблевого навчання — це спосіб перегляду всіх методів навчання та алгоритмів одночасно, а не розгортання їх окремо [15]. Останнім часом методи ансамблевого навчання використовувались для вирішення кількох складних проблем. Ансамблеве навчання покладається на набір комбінованих класифікаторів або предикторів замість окремих класифікаторів, тому ці набори класифікаторів навчаються та вивчаються на основі проведених шаблонів для вирішення тієї самої проблеми та отримання кращих результатів [16].

Існують чотири різні типи технік ансамблевого навчання, як-от bagging, boosting, stacking, voting. Під час голосування потужність кількох окремих класифікаторів полегшує застосування правила комбінування для прийняття рішень.

Кожен алгоритм машинного навчання має різні обмеження, як-от низьке зміщення та незначна дисперсія. Ансамблеве навчання розглядає обмеження автономних методів машинного навчання. Парадигма такого навчання об'єднує різноманітні моделі (слабкі учні) для створення однієї оптимальної прогнозної моделі, яка дає більш точні результати, ніж одна модель [16]. Прогнози, зроблені кожною моделлю в складі ансамблю (базовим учнем), поєднуються за допомогою голосування або усереднення.

Ансамблеве навчання поділяється на два основні типи: гомогенне або гетерогене. Однородний тип ансамблевого навчання передбачає використання різних підмножин даних для навчання базових класифікаторів (слабких учнів). Результат кожного класифікатора агрегується для підвищення точності. Цей тип ансамблевого навчання підходить для великих наборів даних. Bagging і boosting є найпоширенішими типами однорідного ансамблевого навчання. Класифікатор Bagging — це ансамблевий мета-оцінювач, який встановлює кожен базовий класифікатор у випадкових підмножинах вихідного набору даних, а потім агрегує їхні індивідуальні прогнози (чи шляхом голосування, чи шляхом усереднення), щоб сформувати остаточний прогноз [17]. Таку метаоцінку зазвичай можна використовувати як спосіб зменшити дисперсію оцінки чорної скриньки (наприклад, дерева рішень), вводячи рандомізацію в процедуру її побудови, а потім створюючи з неї ансамбль.

У гетерогенних моделях ансамблевого навчання різні базові класифікатори використовуються для навчання на однакових даних. Ця техніка добре працює для невеликих наборів даних. Прикладом гетерогенної моделях ансамблевого навчання є stacking.

VotingClassifier базується на ідеї поєднання концептуально різні класифікаторів машинного навчання та використання більшості голосів або середніх прогнозованих ймовірностей (м'яке голосування) для прогнозування міток класу. Такий класифікатор може бути корисним для набору однаково ефективних моделей, щоб збалансувати їх окремі слабкі сторони [18].

Для організації ансамблевого навчання в цієї роботі було використано декілька алгоритмів, які забезпечили найкращі результати.

Класифікатор KNeiborClassifier

Класифікація на основі найближчих сусідів — це тип навчання на основі екземплярів або неузагальнюючого навчання: воно не намагається побудувати загальну внутрішню модель, а просто зберігає екземпляри навчальних даних. Класифікація обчислюється простою більшістю голосів найближчих сусідів кожної точки: точці запиту призначається клас даних, який має найбільше представників у найближчих сусідах точки [19].

Класифікація k-сусідів у KNeighborsClassifier є найпоширенішим методом. Оптимальний вибір значення k сильно залежить від даних: загалом більше пригнічує вплив шуму, але робить межі класифікації менш чіткими.

Дерево рішень (Decision Tree - DT)

DT є одним із найбільш широко використовуваних для класифікації та виявлення вторгнень. DT складається з трьох основних компонентів, а саме вузла рішення (ідентифікує тестовий атрибут), гілки (можливий вибір на основі значення тестового атрибута) і кінцевого вузла (класу, членом якого є екземпляр). Спочатку вивчається та моделюється набір даних, а потім у алгоритмі DT формується дерево. Коли тестові дані надаються DT, вони будуть класифіковані на основі процедури класифікації

попереднього набору даних. Виконується перевірка для класифікації з використанням значення тестового атрибута та процедури прийняття рішення (позначеної кореневим вузлом). Клас (звичайний, атакуючий) призначається тестовим даним, коли досягається кінцевий вузол. DT краще працює для великих наборів даних [20]. DT має такі переваги, як краща продуктивність виявлення, точність узагальнення тощо.

Випадковий ліс (Random Forest — RF)

RF або ліси випадкових рішень є методом ансамблевого навчання, у якому будується значна кількість декорельованих дерев, а потім усереднюється [21]. RF генерує ліс дерева рішень із довільно розділеного набору даних на вибірки. Для кожного атрибута створюється окреме дерево рішень залежно від незалежної випадкової вибірки. Для класифікації тестових даних отримують прогнози з кожного дерева, і, нарешті, клас призначається тестовим даним за допомогою більшості голосів або техніки усереднення.

Extremely Randomized Trees (Extra Trees)

Extra Tree — це набір моделей на основі ML, які поєднують класифікації з кількох невідрізаних DT на різних підвибірках цілі, щоб підвищити точність узагальнення, бути обчислювально ефективними та запобігти надмірній підгонці [22]. Весь навчальний екземпляр використовується для вирощування дерев, і вузли в кожному дереві розділяються шляхом абсолютного випадкового вибору точок розрізу. Ці прогнози зроблені за допомогою схеми голосування більшості для завдань класифікації або усереднення значень прогнозу для завдань регресії.

Використані набори даних

CSE-CIC-IDS2018 — це загальнодоступний набір даних про вторгнення [23]. Цей набір даних було створено з урахуванням недоліків попередніх наборів даних про вторгнення. CSE-CIC-IDS2018 — це один із найбільших наборів даних IDS із реальним мережевим трафіком і широким спектром атак. Він також містить звичайні дані та дані про вторгнення. CICIDS2018 включає сім різних сценаріїв атак: Brute-force (Web, XSS, FTP, SSH), SQL Injection і DDoS (HOIC, LOIC- UDP, LOIC-HTTP), Heartbleed, Botnet, DoS (Hulk, SlowHTTPTest, GoldenEye, Slowloris), DDoS (HOIC, LOIC- UDP, LOIC-HTTP), Web attacks, і проникнення в мережу зсередини.

Набір даних KDDCup99 [24] містить 41 ознаку та охоплює чотири основні категорії атак: атаки зондування (атаки зі збором інформації), атаки на відмову в обслуговуванні (DoS), атаки користувача на root (U2R), атаки віддаленого до локального (R2L), спостереження та інші зондування, наприклад, сканування портів (probing).

NSL-KDD — це поновлена версія набору даних KDDCup99 [25]. Це ефективний контрольний набір даних, який допоможе дослідникам порівняти різні методи виявлення вторгнень. Цей набір даних не має надлишкових записів, тому будь-яка модель, навчена на цьому наборі даних, не повинна бути схильна до повторних записів атак. Загалом у цьому наборі даних для одного запису є 43 ознаки. З 43 ознак 41 пов’язана з вхідним трафіком, а дві інші є мітками та балами вхідного трафіку. Цей набір даних має загалом чотири класи для різних атак: зондування, атак користувача на root (U2R), відмова в обслуговуванні (DoS) і віддалена локальна атака (R2L).

Набір даних UNSW-NB15 містить понад два мільйони записів, 48 ознак і дев’ять різних типів атак: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode та Worm [26].

Набір даних LITNET-2020 — це відносно новий набір даних, зібраний академічною

мережею LITNET (Литовська науково-освітня мережа) у мережевому трафіку Литви в режимі реального часу. Це реальний і сучасний мережевий набір даних на основі потоку [27], розроблений для тестування систем IDS. У цьому наборі даних було 85 функцій мережевого потоку та 12 типів атак.

МЕТОДИКА ДОСЛІДЖЕНЬ

Усі розрахунки в роботі було виконано на комп'ютері з процесором Intel Core i5 з 8 ГБ пам'яті під управлінням Windows 11. Було використано Python 3.11.6 та пакет Scikit-Learn 1.3.1.

Пропонована система

Сучасні IDS повинні відповідати вимогам і зростаючим потребам у вдосконаленні технологій. Для успішної роботи IDS необхідна високоефективна класифікація з використанням даних, які раніше не були відомі системі. IDS зазвичай обробляють досить великий обсяг даних, які містять різні надлишкові ознаки, що призводить до низького рівня точності та тривалого часу обробки [28]. Це робить вибір ознак для класифікації важливим питанням. Для скорочення часу навчання моделі класифікації та підвищення рівня її точності важливим питанням є вибір найважливіших ознак із набору даних [29]. У цьому дослідженні досліджувалися різні методи вибору ознак і ансамблю, щоб створити ефективну IDS з високою точністю розпізнавання загроз. На рис. 1 показана структура модуля оцінювання з вибором ознак та ансамблевим класифікатором.

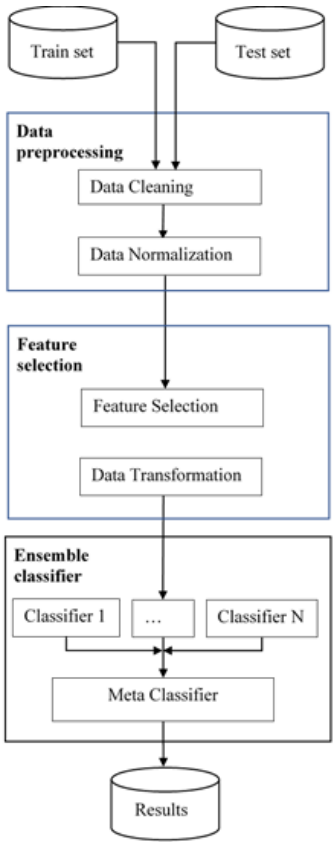


Рис. 1. Структура модуля оцінювання з вибором ознак та ансамблевим класифікатором

ISSN 2663 - 4023

Попередня обробка даних передбачає наступні операції:

- виявлення/усунення невідповідностей;
- виправлення помилок у даних;
- заповнення відсутніх значень;
- масштабування та нормалізація.

Після попередньої обробки були обрані найважливіші ознаки в наборі даних за допомогою відповідних алгоритмів. Час начання та класифікації та інші показники класифікатора оцінювались разом із точністю результатів.

Для вибору ознак були використані декілка варіантів обрання ознак:

- тест хі-квадрат та вибір долі ознак, яки відповідають критерію;
- вибір ознак на основі кореляції;
- рекурсивне виключення ознак (RFE) або модифікація з перехресною перевіркою для вибраних ознак (RFECV);
- оцінки на основі дерева;
- оцінки на основі нечітких множин.

В якості метакласифікатора також використовувались декілька варіантів:

- алгоритм RandomForest;
- алгоритм VotingClassifier;
- алгоритм нечіткого обрання результатів оцінювання;
- алгоритм StackingClassifier.

Для уточнення класифікації використовувались варианти:

- нечіткий класифікатор FuzzyKNN;
- алгоритм BaggingClassifier з використанням генерування випадкових наборів даних.

Підготовка даних

Деякі набори даних, які було використано, містили кілька окремих файлів з даними. Вони об’єднувались в один результуючий документ. Рядки з помилковими даними видалялись.

Стовпчики з категоріальними ознаками перетворювались на цифрові. Значення true, false, off і low були в результаті перетворені на нуль або одиницю.

Більшість алгоритмів класифікації функціонують більш ефективно, коли ознаки мають порівнянну величину, оскільки це допомагає зменшити зміщення в бік ознак із високими значеннями множинності в результатах прогнозування [30]. Отримані числові дані нормалізувалися за допомогою вбудованих можливостей scikit-learn (переважно StandardScaler).

Методи виділення ознак

Методи машинного навчання та інтелектуального аналізу даних широко використовуються для обробки та вилучення інформації з великомасштабних даних. Той факт, що ці методи застосовуються до великих обсягів даних, що містять нерелевантні та непотрібні ознаки, впливає на точність інформації та є дорогим з точки зору часу. Щоб запобігти цьому, у літературі використовуються багато алгоритмів вибору ознак, за допомогою яких непотрібні ознаки видаляються з набору даних для навчання моделі класифікації.

У межах цього дослідження були використані декілка варіантів обрання ознак:

- тест хі-квадрат, який базується на результатах перевірки наявності різниці між спостережуваною та очікуваною частотами;

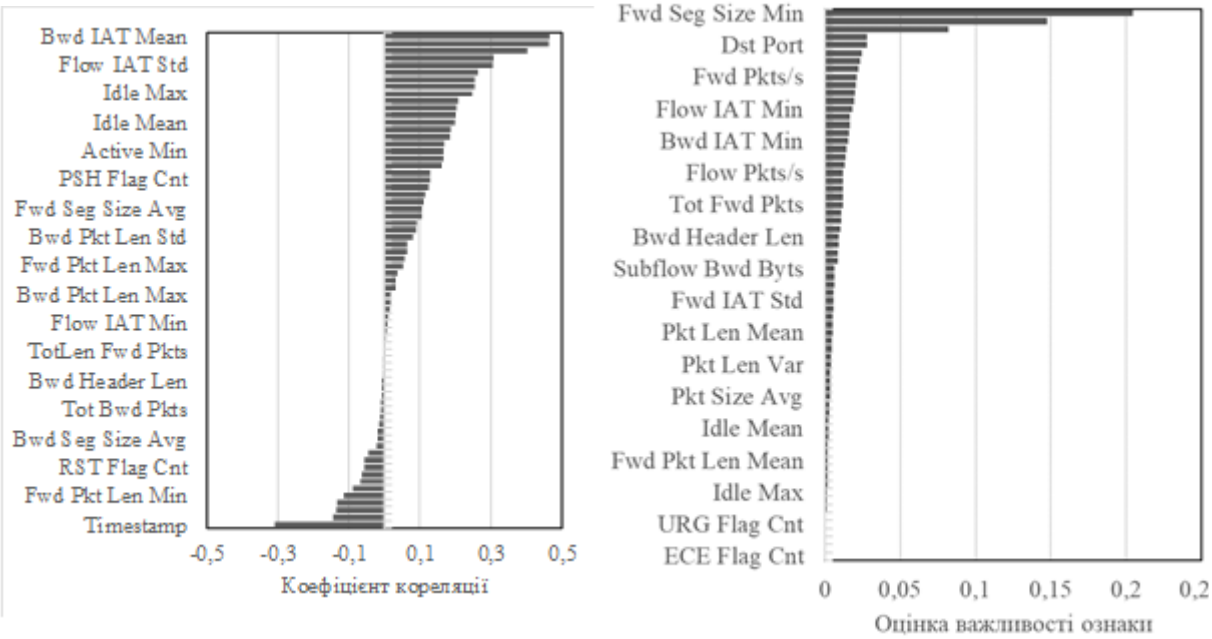
ISSN 2663 - 4023

– вибір ознак на основі кореляції, який базується на гіпотезі про те, що ефективні підмножини ознак складаються з ознак, які мають високу кореляцію з відповідним класом і низьку кореляцію між собою;
– рекурсивне виключення ознак (RFE) або модифікація з перехресною перевіркою для вибраних ознак (RFECV);
– оцінки на основі дерева для обчислення важливості ознак на основі impurity;
– оцінки на основі нечітких множин.

Приклад обчислення корреляції ознак набору даних CICIDS2018 і наявності атаки наведено на рис. 2а.

Після аналізу поведінки компонентів пропонованої системи було додано ще один варіант обрання ознак, більш адекватний саме для класифікаторів на основі дерев: вибір ознак за критерієм середнього зменшення помилок класифікації (Mean Decrease in Impurity — MDI).

Приклад обчислення вибору ознак за критерієм середнього зменшення помилок класифікації для набору даних CICIDS2018 наведено на рис. 2б.



а) Кореляційна діаграма наявності атаки і ознак набору даних CICIDS2018

б) Діаграма MDI для ознак набору даних CICIDS2018

Рис. 2. Варианти критеріїв обрання ознак для набору даних CICIDS2018

Як видно з наведених рисунків, кількість ознак у результуючому наборі даних для навчання класифікатора залежить від критерію вибору цих ознак. Те ж саме питання виникає й для обрання ознак за допомогою тестів chi-square або fuzzy logic. У подальшому дослідженні критерії вибору ознак вибиралися так, щоб обсяг результуючого набору даних складав 25-50% вихідного.

ISSN 2663 - 4023

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Основні результати роботи одиничних класифікаторів наведені в таблиці 1. Позначки в таблиці: KNN — класифікатор KNeibourClassifier (метод найближчих сусідів); DTC — класифікатор DecignTreeClassifier; ETC — класифікатор ExtraTreeClassifier; RFC — класифікатор RandomForestClassifier; MLP — класифікатор MLPClassifier (багатошаровий персептрон); ADA — класифікатор ADABoost; LR — логістична регресія.

Таблиця 1

Результати без недостатньої вибірки та вибору ознак (IDS2018)

Класифікатор	Точність, %	Показник F1	Час навчання, с
KNN	99.99	0,999	0.0945
DTC	100,00	1.00	1,836
ETC	100,00	1.00	0,214
RFC	100,00	1.00	22,01
MLP	99.5	0,995	206,67
ADA	68,0	0,716	37,48
LR	94,9	0,952	54,31

Результати показують, що алгоритми, засновані на дереві, є досить успішними при проведенні класифікації різних вибірок, які було використано для побудови класифікатора IDS. Оскільки метою дослідження є розробка ансамблевої моделі з набором даних, вибраних за ознаками, на наступному кроці всі алгоритми були використані з набором даних, після застосування RFE або MDI. Приклад результатів наведено в таблиці 2.

Таблиця 2

Результати для набору даних зі скороченой кількістю ознак

Класифікатор	Точність,%	Показник F1	Час навчання, с
KNN	99.99	0,999	0,0628
DTC	100,00	1.00	1,225
ETC	100,00	1.00	0,177
RFC	100,00	1.00	28,38
MLP	99.6	0,995	202,15
ABC	94,2	0,940	44,81
LR	93,4	0,936	61,95

Алгоритми LogisticRegression, AdaBoost, MultiLayerPerceptron довго навчаються і для досягнення високих показників точності потрібують налагодження. Коли час прогнозування та рівень точності оцінюються разом, було знайдено, що алгоритми дерева рішень (DTC), найближчих сусідів (KNN) та додаткових дерев (ETC) дають результати з високою точністю та швидким часом прогнозування. Тому ці три алгоритми були обрані для моделі ансамблю.

Показники точності та час роботи цих алгоритмів у результаті роботи з оригінальними наборами даних і наборами даних зі скороченою кількістю ознак наведені в таблиці 3. У цій таблиці представлені дані KDDCup99, хоча аналогічні результати були отримані й для інших наборів даних — LITNET, ISD2018.

Деяке підвищення точності оцінювання було досягнуто за рахунок використання метакласифікаторів (VotingClassifier, StackingClassifier, RandomForestClassifier). Було побудовано декілька моделей з використанням різних алгоритмів класифікації, які

навчались на підмножинах зі зменшеною кількістю ознак, або на наборах даних вихідного розміру.

Вибір результатів класифікації встановлювався переважно за допомогою алгоритму VotingClassifier. Його ідея полягає в тому, щоб поєднати концептуально різні класифікатори машинного навчання та використовувати більшість голосів або середні передбачені ймовірності (м'яке голосування) для прогнозування міток класу. Такий класифікатор може бути корисним для набору однаково ефективних моделей, щоб збалансувати їх окремі слабкі сторони.

Таблиця 3

Порівняння точності і часу навчання моделі на наборах даних з різною кількістю ознак (вихідний набор даних KDDCup99)

Параметр	**Класифікатор**		
	KNN	**DTC**	**ETC**
Точність, %			
41 ознака (вихідний)	99,9	100	100
21 ознака	99,9	100	100
Час навчання, с			
41 ознака (вихідний)	0,0945	1,836	0,214
21 ознака	0,0628	1,225	0,177

Результати навчання і використання моделі ансамблю наведено в таблиці 4.Як випливає з таблиці 4, точність класифікації при навчанні моделі на наборі даних лише з важливими ознаками практично не знижується.

Таблиця 4

Результати оцінювання точності різних ансамблевих класифікаторів і наборів даних

Набір даних	**Кількість ознак**	**Класифікатор**	**Точність, %**	**Час навчання, с**
KDDCup99	41	RF	99,55	22,55
	41	RF	100,0	12,57
	20	Voting	99,34	2.26
	20	Fuzzy	99,14	2,36
NSL-KDD	41	RF	99,99	9,53
	16	RF	100,0	9,18
	16	Voting	99,34	1,31
	16	Fuzzy	99,14	1,43
UNSW-NB15 (скорочений набір)	44	RF	99,99	17,66
	21	RF	99,99	12,40
	21	Voting	99,34	2,01
	21	Fuzzy	99,33	2,22
IDS 2018 (одна доба)	79	RF	99,99	172.1
	21	RF	99,99	121,4
	21	Voting	99,99	8,51
	21	Fuzzy	99,99	9,02

Час навчання Voting-класифікатора залежить від базових класифікаторів, які використано для його побудови. Наприклад, якщо для побудови вотуючого класифікатора користуватись RandomForest-Classifier (та ще два інші) і порівнювати швидкість навчання із безпосередньо RandomForestClassifier, то час навчання вотуючого класифікатора зростає на 10–15 %. Якщо порівнювати, наприклад, швидкість навчання вотуючого класифікатора на базі KNearestNeiborClassifier, ExtraTreeClassifier, DesignTreeClassifier з навчанням RandomForestClassifier, то час навчання вотуючого класифікатора значно менше (в залежності від набору даних, який обрано для навчання,

ISSN 2663 - 4023

але щонайменше на 60–70%). Час навчання класифікатора з використанням FuzzyLogic [33] практично не відрізняється від часу навчання вотуючого класифікатора (більше приблизно на 10–15%).

Вплив кількості ознак на час навчання класифікаторів і ансамбля VotingClassifier в цілому наведено на рис. 3. Для ExtraTreeClassifier час навчання практично не залежить від кількості ознак. Для DesignTree або KNeibors (і, як наслідок, для класифікатора Voting в цілому) час навчання помітно зростає зі збільшенням кількості ознак.

Зменшення кількості ознак на усіх наборах даних впливає на точність оцінювання досить неоднозначно (фактично, відповідно рис. 2б). Поки група ознак в наборі даних для навчання містить перші за списком ознаки з найбільшим впливом, точність моделі знаходиться на початковому рівні — більш ніж 99%. При виключенні з моделі хоча б однієї з ознак з великим впливом, точність моделі стрибкоподібно знижується.

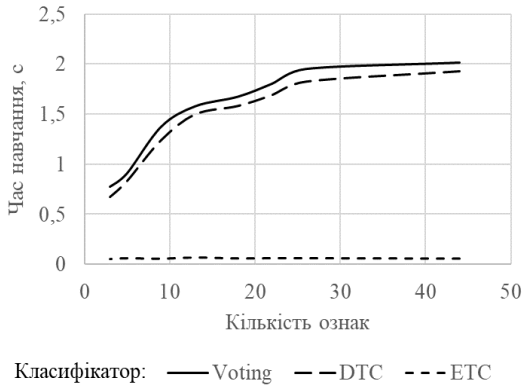


Рис. 3. Залежність часу навчання деяких класифікаторів ансамблю від кількості ознак (навчання на частині набору даних UNSW-NB15)

У деяких розрахунках визначення вагових коефіцієнтів для визначення результатів голосування здійснювалося за допомогою евристичного набору нечітких правил.

Для зменшення дисперсії базового оцінювача та підвищення надійності класифікації було використано алгоритм bagging, який агрегує індивідуальні прогнози за випадковими підмножинами початкового навчального набору для формування остаточного прогнозу.

При відповідному налаштуванні точність класифікацій трохи збільшується, але обрання Stacking або Bagging класифікатора як основи навчання моделі для усіх наборів даних збільшує час навчання більш ніж на порядок (у залежності від базових класифікаторів або кількості підмножин даних). При збільшенні кількості спостережень в наборі даних для навчання ефект зростання часу навчання стає більш помітним.

Таким чином, запропонована нова модель ансамблю дозволила підтримати високий рівень точності (краще 99%) і низький рівень помилок після навчання на наборі даних зі зменшеною кількістю ознак.

ВИСНОВКИ

У дослідженні була запропонована нова модель IDS з використанням методів ансамблевого навчання на скорочених за допомогою алгоритмів вибору ознак великих наборах даних.

Для вибору необхідних ознак було використано статистичні тести та нечіткі правила.

Найкращі результати класифікації для усіх досліджених наборів даних забезпечили класифікатори на основі дерев: DecignTreeClassifier, ExtraTreeClassifier, RandomForestClassifier.

Зменшення кількості ознак на усіх наборах даних впливає на точність оцінювання відповідно до критерію середнього зменшення помилок класифікації. Поки група ознак в наборі даних для навчання містить перши за списком ознаки з найбільшим впливом, точність моделі знаходиться на початковому рівні, але при виключенні з моделі хоча б однієї з ознак з великим впливом, точність моделі стрибкоподібно знижується.

Найкращі показники за швидкістю навчання забезпечив класифікатор VotingClassifier, побудований на базі алгоритмів з максимальною швидкістю навчання. Час навчання класифікатора з використанням FuzzyLogic практично не відрізняється від часу навчання вотуючого класифікатора (більше приблизно на 10–15%). За рахунок виключення із моделі несуттєвих ознак досягається помітне збільшення швидкості навчання (до 60–70%).

Для майбутньої роботи метою є подальше вдосконалення запропонованої моделі IDS в напрямках вдосконалення вибору класифікаторів для отримання оптимальних результатів, та налаштування параметрів вибраних класифікаторів, удосконалення стратегії узагальнення результатів окремих класифікаторів. Для запропонованої моделі істотний інтерес представляє можливість виявлення окремих типів атак з урахуванням багатокласового прогнозування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Chua, T.-H., & Salam, I. (2023). Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection Using Progressive Dataset. *Symmetry*, *15*(6), 1251. https://doi.org/10.3390/sym15061251
2. Aleesa, A. M., Zaidan, B. B., Zaidan, A. A., & Sahar, N. M. (2019). Review of intrusion detection systems based on deep learning techniques: Coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. *Neural Computing and Applications*, *32*(14), 9827–9858. https://doi.org/10.1007/s00521-019-04557-3
3. Tama, B. A., & Lim, S. (2021). Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. *Computer Science Review*, *39*, 100357. https://doi.org/10.1016/j.cosrev.2020.100357
4. Verma, Abhishek & Ranga, Virender. (2018). On Evaluation of Network Intrusion Detection Systems: Statistical Analysis of CIDDS-001 Dataset Using Machine Learning Techniques. *Pertanika Journal of Science and Technology*. *26*. 1307-1332. https://doi.org/10.36227/techrxiv.11454276.v1.
5. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, *50*, 102419. https://doi.org/10.1016/j.jisa.2019.102419
6. Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks, 188, 107840. https://doi.org/10.1016/j.comnet.2021.107840
7. Rahman, M. A., Asyhari, A. T., Wen, O. W., Ajra, H., Ahmed, Y., & Anwar, F. (2021). Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection. *Multimedia Tools and Applications*, *80*(20), 31381–31399. https://doi.org/10.1007/s11042-021-10567-y
8. Kocher, G., & Kumar, G. (2021). Analysis of machine learning algorithms with feature selec-tion for intrusion detection using UNSW-NB15 dataset. International Journal of Network Secu-rity & Its Applications, 13(1), 21–31. https://doi.org/10.5121/ijnsa.2021.13102
9. Kumar, K., & Singh, J. (2016). Network intrusion detection with feature selection techniques using machine-learning algorithms. International Journal of Computer Applications, 150(12), 1–13. https://doi.org/10.5120/ijca2016910764

10. Krishnaveni, S., Sivamohan, S., Sridhar, S. S., & Prabakaran, S. (2021). Efficient feature selec-tion and classification through ensemble method for network intrusion detection on cloud com-puting. Cluster Computing. https://doi.org/10.1007/s10586-020-03222-y
11. Shanmugavadivu, R. & Dr. Nagarajan,. N. (2011). Network Intrusion Detection System using Fuzzy Logic. Indian Journal of Computer Science and Engineering. 2. https://www.researchgate.net/publication/50417996_Network_Intrusion_Detection_System_using_Fuzzy_Logic
12. Almseidin, M., & Kovács, S. (2019). Intrusion Detection Mechanism Using Fuzzy Rule Interpo-lation. ArXiv, abs/1904.08790. https://api.semanticscholar.org/CorpusID:120430608
13. Naik, N., Diao, R., & Shen, Q. (2018). Dynamic fuzzy rule interpolation and its application to intrusion detection. IEEE Transactions on Fuzzy Systems, 26(4), 1878–1892. https://doi.org/10.1109/tfuzz.2017.2755000
14. Benaddi, H., Ibrahimi, K., & Benslimane, A. (2018). Improving the Intrusion Detection System for NSL-KDD Dataset based on PCA-Fuzzy Clustering-KNN. У 2018 6th international confer-ence on wireless networks and mobile communications (WINCOM). IEEE. https://doi.org/10.1109/wincom.2018.8629718
15. Rani, D., Gill, N. S., Gulia, P., & Chatterjee, J. M. (2022). An ensemble-based multiclass clas-sifier for intrusion detection using internet of things. Computational Intelligence and Neurosci-ence, 2022, 1–16. https://doi.org/10.1155/2022/1668676
16. Guo, G. (2021). A machine learning framework for intrusion detection system in iot networks using an ensemble feature selection method. In 2021 IEEE 12th annual information technology, electronics and mobile communication conference (IEMCON). IEEE. https://doi.org/10.1109/iemcon53756.2021.9623082
17. A. Subasi, S. Algebsani, W. Alghamdi, E. Kremic, J. Almaasrani, N. Abdulaziz, Intrusion de-tection in smart healthcare using bagging ensemble classifier, in International Conference on Medical and Biological Engineering, (2021), 164–171. https://doi.org/10.1007/978-3-030-73909-6_18
18. Khan, Muhammad Almas & Khattak, Muazzam & Latif, Shahid & Shah, Awais & Rehman, Mujeeb & Boulila, Wadii & Driss, Maha & Ahmad, Jawad. (2022). Voting Classifier-Based Intrusion Detection for IoT Networks. 10.1007/978-981-16-5559-3_26.
19. Cunningham, P., & Delany, S. J. (2021). K-Nearest neighbour classifiers - A tutorial. ACM Computing Surveys, 54(6), 1–25. https://doi.org/10.1145/3459665
20. J. Singh, M. J. Nene, A survey on machine learning techniques for intrusion detection systems, Int. J. Adv. Res. Comput. Commun. Eng., 2 (2013), 4349–4355.
21. N. Farnaaz, M. Jabbar, Random forest modeling for network intrusion detection system, Proce-dia Comput. Sci., 89 (2016), 213–217. https://doi.org/10.1016/j.procs.2016.06.047
22. Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. Mach Learn 63, 3–42 (2006). https://doi.org/10.1007/s10994-006-6226-1
23. IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018). https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv
24. Aggarwal, P., & Sharma, S. K. (2015). Analysis of KDD dataset attributes - class wise for intru-sion detection. Procedia Computer Science, 57, 842–851. https://doi.org/10.1016/j.procs.2015.07.490
25. NSL-KDD dataset. URL: http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html.
26. Moustafa, Nour & Slay, Jill. (2015). UNSW-NB15: a comprehensive data set for network intru-sion detection systems (UNSW-NB15 network data set). https://doi.org/10.1109/MilCIS.2015.7348942.
27. Damasevicius, R., Venckauskas, A., Grigaliunas, S., Toldinas, J., Morkevicius, N., Aleliunas, T., & Smuikys, P. (2020). LITNET-2020: An annotated real-world network flow dataset for network intrusion detection. Electronics, 9(5), 800. https://doi.org/10.3390/electronics9050800
28. Emanet S., Karatas Baydogmus G., Demir O. (2023) An ensemble learning based IDS using Voting rule: VEL-IDS. PeerJ Computer Science 9: e1553 https://doi.org/10.7717/peerj-cs.1553
29. Zhou, Z.H. (2021). Ensemble Learning. In: Machine Learning. Springer, Singapore. https://doi.org/10.1007/978-981-15-1967-3_8
30. Shushura, O. M., Asieieva, L. A., Nedashkivskiy, O. L., Havrylko, Y. V., Moroz, Y. O., Smailova, S. S., & Sarsembayev, M. (2022). SIMULATION OF INFORMATION SECURI-TY RISKS OF AVAILABILITY OF PROJECT DOCUMENTS BASED ON FUZZY LOG-IC. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, 12(3), 64–68. https://doi.org/10.35784/iapgos.3033

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (19), 2023
ISSN 2663 - 4023

Yevhen Chychkarov
doctor of Technical Sciences, professor, professor of the Artificial Intelligence Department
State University of Information and Communication Technologies, Kyiv, Ukraine
ORCID 0000-0002-4362-5129
chychkarovea@gmail.com

Olga Zinchenko
doctor of Technical Sciences, head of the Artificial Intelligence Department
State University of Information and Communication Technologies, Kyiv, Ukraine
ORCID 0000-0002-3973-7814
zinchenkoov@gmail.com

Andriy Bondarchuk
doctor of technical sciences, professor, director of the educational and scientific institute of information technologies
State University of Information and Communication Technologies, Kyiv, Ukraine
ORCID 0000-0001-5124-5102
dekan.it@ukr.net

Liudmyla Aseeva
postgraduate
State University of Information and Communication Technologies, Kyiv, Ukraine
ORCID 0000-0001-5954-4211
aseewal@i.ua

DETECTION OF NETWORK INTRUSIONS USING MACHINE LEARNING ALGORITHMS AND FUZZY LOGIC

Abstract. The study proposed a model of an intrusion detection system based on machine learning using feature selection in large data sets based on ensemble learning methods. Statistical tests and fuzzy rules were used to select the necessary features. When choosing a basic classifier, the behavior of 8 machine learning algorithms was investigated. The proposed system provided a reduction in intrusion detection time (up to 60%) and a high level of attack detection accuracy. The best classification results for all studied datasets were provided by tree-based classifiers: DesignTreeClassifier, ExtraTreeClassifier, RandomForestClassifier. With the appropriate setting, choosing Stacking or Bagging classifier for model training using all data sets provides a small increase in the classification accuracy, but significantly increases the training time (by more than an order of magnitude, depending on the base classifiers or the number of data subsets). As the number of observations in the training dataset increases, the effect of increasing training time becomes more noticeable. The best indicators in terms of learning speed were provided by the VotingClassifier, built on the basis of algorithms with maximum learning speed and sufficient classification accuracy. The training time of the classifier using FuzzyLogic practically does not differ from the training time of the voting classifier (approximately 10–15% more). The influence of the number of features on the training time of the classifiers and the VotingClassifier ensemble depends on the behavior of the base classifiers. For ExtraTreeClassifier, the training time is weakly dependent on the number of features. For DesignTree or KNeibors (and, as a result, for the Voting classifier in general), the training time increases significantly with the increase in the number of features. Reducing the number of features on all datasets affects the estimation accuracy according to the criterion of average reduction of classification errors. As long as the group of features in the training dataset contains the first in the list of features with the greatest influence, the accuracy of the model is at the initial level, but when at least one of the features with a large influence is excluded from the model, the accuracy of the model drops dramatically.

Keywords: intrusion detection system; machine learning; ensemble learning; classifier; fuzzy logic; cyber attack; cyber defense using machine learning; feature selection algorithms.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Chua, T.-H., & Salam, I. (2023). Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection Using Progressive Dataset. *Symmetry*, *15*(6), 1251. https://doi.org/10.3390/sym15061251
2. Aleesa, A. M., Zaidan, B. B., Zaidan, A. A., & Sahar, N. M. (2019). Review of intrusion detection systems based on deep learning techniques: Coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions. *Neural Computing and Applications*, *32*(14), 9827–9858. https://doi.org/10.1007/s00521-019-04557-3
3. Tama, B. A., & Lim, S. (2021). Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. *Computer Science Review*, *39*, 100357. https://doi.org/10.1016/j.cosrev.2020.100357
4. Verma, Abhishek & Ranga, Virender. (2018). On Evaluation of Network Intrusion Detection Systems: Statistical Analysis of CIDDS-001 Dataset Using Machine Learning Techniques. *Pertanika Journal of Science and Technology*. *26*. 1307-1332. https://doi.org/10.36227/techrxiv.11454276.v1.
5. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, *50*, 102419. https://doi.org/10.1016/j.jisa.2019.102419
6. Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks, 188, 107840. https://doi.org/10.1016/j.comnet.2021.107840
7. Rahman, M. A., Asyhari, A. T., Wen, O. W., Ajra, H., Ahmed, Y., & Anwar, F. (2021). Effective combining of feature selection techniques for machine learning-enabled IoT intrusion detection. *Multimedia Tools and Applications*, *80*(20), 31381–31399. https://doi.org/10.1007/s11042-021-10567-y
8. Kocher, G., & Kumar, G. (2021). Analysis of machine learning algorithms with feature selec-tion for intrusion detection using UNSW-NB15 dataset. International Journal of Network Secu-rity & Its Applications, 13(1), 21–31. https://doi.org/10.5121/ijnsa.2021.13102
9. Kumar, K., & Singh, J. (2016). Network intrusion detection with feature selection techniques using machine-learning algorithms. International Journal of Computer Applications, 150(12), 1–13. https://doi.org/10.5120/ijca2016910764
10. Krishnaveni, S., Sivamohan, S., Sridhar, S. S., & Prabakaran, S. (2021). Efficient feature selec-tion and classification through ensemble method for network intrusion detection on cloud com-puting. Cluster Computing. https://doi.org/10.1007/s10586-020-03222-y
11. Shanmugavadivu, R. & Dr. Nagarajan,. N. (2011). Network Intrusion Detection System using Fuzzy Logic. Indian Journal of Computer Science and Engineering. 2. https://www.researchgate.net/publication/50417996_Network_Intrusion_Detection_System_using_Fuzzy_Logic
12. Almseidin, M., & Kovács, S. (2019). Intrusion Detection Mechanism Using Fuzzy Rule Interpo-lation. ArXiv, abs/1904.08790. https://api.semanticscholar.org/CorpusID:120430608
13. Naik, N., Diao, R., & Shen, Q. (2018). Dynamic fuzzy rule interpolation and its application to intrusion detection. IEEE Transactions on Fuzzy Systems, 26(4), 1878–1892. https://doi.org/10.1109/tfuzz.2017.2755000
14. Benaddi, H., Ibrahimi, K., & Benslimane, A. (2018). Improving the Intrusion Detection System for NSL-KDD Dataset based on PCA-Fuzzy Clustering-KNN. У 2018 6th international confer-ence on wireless networks and mobile communications (WINCOM). IEEE. https://doi.org/10.1109/wincom.2018.8629718
15. Rani, D., Gill, N. S., Gulia, P., & Chatterjee, J. M. (2022). An ensemble-based multiclass clas-sifier for intrusion detection using internet of things. Computational Intelligence and Neurosci-ence, 2022, 1–16. https://doi.org/10.1155/2022/1668676
16. Guo, G. (2021). A machine learning framework for intrusion detection system in iot networks using an ensemble feature selection method. In 2021 IEEE 12th annual information technology, electronics and mobile communication conference (IEMCON). IEEE. https://doi.org/10.1109/iemcon53756.2021.9623082
17. A. Subasi, S. Algebsani, W. Alghamdi, E. Kremic, J. Almaasrani, N. Abdulaziz, Intrusion de-tection in smart healthcare using bagging ensemble classifier, in International Conference on Medical and Biological Engineering, (2021), 164–171. https://doi.org/10.1007/978-3-030-73909-6_18
18. Khan, Muhammad Almas & Khattak, Muazzam & Latif, Shahid & Shah, Awais & Rehman, Mujeeb & Boulila, Wadii & Driss, Maha & Ahmad, Jawad. (2022). Voting Classifier-Based Intrusion Detection for IoT Networks. 10.1007/978-981-16-5559-3_26.

19. Cunningham, P., & Delany, S. J. (2021). K-Nearest neighbour classifiers - A tutorial. ACM Computing Surveys, 54(6), 1–25. https://doi.org/10.1145/3459665
20. J. Singh, M. J. Nene, A survey on machine learning techniques for intrusion detection systems, Int. J. Adv. Res. Comput. Commun. Eng., 2 (2013), 4349–4355.
21. N. Farnaaz, M. Jabbar, Random forest modeling for network intrusion detection system, Proce-dia Comput. Sci., 89 (2016), 213–217. https://doi.org/10.1016/j.procs.2016.06.047
22. Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. Mach Learn 63, 3–42 (2006). https://doi.org/10.1007/s10994-006-6226-1
23. IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018). https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv
24. Aggarwal, P., & Sharma, S. K. (2015). Analysis of KDD dataset attributes - class wise for intru-sion detection. Procedia Computer Science, 57, 842–851. https://doi.org/10.1016/j.procs.2015.07.490
25. NSL-KDD dataset. URL: http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html.
26. Moustafa, Nour & Slay, Jill. (2015). UNSW-NB15: a comprehensive data set for network intru-sion detection systems (UNSW-NB15 network data set). https://doi.org/10.1109/MilCIS.2015.7348942.
27. Damasevicius, R., Venckauskas, A., Grigaliunas, S., Toldinas, J., Morkevicius, N., Aleliunas, T., & Smuikys, P. (2020). LITNET-2020: An annotated real-world network flow dataset for network intrusion detection. Electronics, 9(5), 800. https://doi.org/10.3390/electronics9050800
28. Emanet S., Karatas Baydogmus G., Demir O. (2023) An ensemble learning based IDS using Voting rule: VEL-IDS. PeerJ Computer Science 9: e1553 https://doi.org/10.7717/peerj-cs.1553
29. Zhou, Z.H. (2021). Ensemble Learning. In: Machine Learning. Springer, Singapore. https://doi.org/10.1007/978-981-15-1967-3_8
30. Shushura, O. M., Asieieva, L. A., Nedashkivskiy, O. L., Havrylko, Y. V., Moroz, Y. O., Smailova, S. S., & Sarsembayev, M. (2022). SIMULATION OF INFORMATION SECURI-TY RISKS OF AVAILABILITY OF PROJECT DOCUMENTS BASED ON FUZZY LOG-IC. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, 12(3), 64–68. https://doi.org/10.35784/iapgos.3033