Шифр: Threat Classification

КЛАСИФІКАЦІЯ ТА ВИЗНАЧЕННЯ ІМОВІРНОСТЕЙ РЕАЛІЗАЦІЇ КІБЕРЗАГРОЗ ЗА ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

АНОТАЦІЯ

Наукова робота під шифром «Threat Classification» присвячена дослідженню застосування методів штучного інтелекту, зокрема нейронних мереж, для класифікації та прогнозування ймовірності реалізації загроз інформаційної безпеки.

Актуальність зумовлена стрімким зростанням кількості кіберінцидентів і потребою у переході від реактивних до проактивних систем кіберзахисту.

Мета роботи: розроблення інтелектуальних моделей для кількісної оцінки ризиків реалізації загроз і їх автоматизованої класифікації за змістовими характеристиками.

Об’єкт дослідження: процеси оцінювання, прогнозування та класифікації загроз інформаційної безпеки.

Предмет дослідження: методи побудови та навчання нейронних мереж для класифікації загроз, моделювання ймовірностей їх реалізації і аналізу їх текстових описів.

Завдання наукової роботи полягає у створенні моделі прогнозування ймовірності загрози на основі багатошарового перцептрона, розробленні моделі класифікації загроз за текстами та інтеграції їх у єдину систему оцінювання ризиків.

Методика дослідження включає застосування методів машинного навчання, TF-IDF векторизації текстів, аугментації даних і алгоритмів багатовихідної класифікації.

За результатами дослідження досягнута точність класифікації загроз становила понад 94%, що підтверджує дієвість запропонованих підходів. Моделі було успішно інтегровано у прототип системи аналізу загроз, що підтверджує

можливість практичного використання виявлених підходів в системах кіберзахисту нового покоління.

Робота складається з 34 сторінок, містить 3 розділи, 5 рисунків, опис реалізації та список із 13 джерел.

Ключові слова: інформаційна безпека, нейронні мережі, класифікація загроз, прогнозування ризиків, машинне навчання, багатошаровий перцептрон, TF-IDF, аугментація даних, кіберзахист.

ЗМІСТ

ВСТУП

У сучасному цифровому світі інформація є одним із найцінніших ресурсів, тому її захист від несанкціонованого доступу, змін чи знищення має надзвичайно важливе значення для будь-яких організацій. Швидкий розвиток технологій та цифровізація бізнесу і повсякденного життя призводять до збільшення кількості кіберзагроз і розширення можливих напрямів атак. Ці загрози охоплюють усе – від шкідливих програм і фішингових листів до складних цілеспрямованих атак. Інструменти, створені для зручності чи підвищення ефективності, часто можуть використовуватися зловмисниками, що робить питання кібербезпеки ще більш актуальним.

Традиційні реактивні підходи до захисту інформації, спрямовані на усунення наслідків інцидентів, виявляються недостатніми в умовах динамічного ландшафту загроз. Бюрократичні процедури та необхідність окремої експертної оцінки кожного нового виду атак значно уповільнюють адаптацію систем безпеки до сучасних викликів. У результаті виникає нагальна потреба у впровадженні проактивних, гнучких та адаптивних рішень, здатних ефективно прогнозувати, оцінювати та класифікувати загрози в реальному часі.

Однією з центральних проблем теорії та практики інформаційної безпеки є точне визначення ймовірності реалізації загрози. Від цього залежить можливість обґрунтовано оцінювати ризики, оптимально розподіляти ресурси для їх мінімізації та приймати ефективні управлінські рішення. Проте ця задача ускладнюється багатофакторністю впливів – технічних, організаційних та людських – і часто обмеженою кількістю історичних даних. Традиційні якісні методи, такі як матриці ризиків чи експертні оцінки, страждають від суб’єктивності, тоді як кількісні статистичні підходи потребують значних обсягів достовірної інформації й не завжди враховують нелінійні взаємозв’язки між факторами.

У цьому контексті особливої уваги набувають сучасні методи штучного інтелекту, зокрема машинне навчання та штучні нейронні мережі. Вони здатні навчатися на даних, виявляти приховані закономірності, моделювати складні взаємозв’язки та адаптуватися до нових умов. Застосування таких технологій відкриває можливість переходу від інтуїтивних і якісних оцінок до об’єктивного, кількісного прогнозування ризиків, що є ключовим чинником у створенні ефективних, адаптивних систем кіберзахисту нового покоління.

1. ТЕОРЕТИЧНІ ОСНОВИ КЛАСИФІКАЦІЇ ТА ОЦІНЮВАННЯ ЗАГРОЗ ІЗ ВИКОРИСТАННЯМ НЕЙРОННИХ МЕРЕЖ

1.1. Класифікація та оцінювання загроз

Класифікація кіберзагроз є основою для побудови систем ризик-менеджменту в інформаційній безпеці. Вона передбачає поділ загроз за певними критеріями, нижче приведені ті, які є найпоширенішими.

Джерело походження: внутрішні (людські помилки, зловмисники всередині організації) та зовнішні (хакерські групи, державні суб’єкти, шкідливе ПЗ).

Тип впливу: загрози конфіденційності, цілісності, доступності, автентичності.

Рівень інфраструктури: від прикладного рівня до фізичного шару.

Характер впливу: цілеспрямовані, випадкові, техногенні, соціотехнічні тощо.

Одним із ключових завдань після класифікації є визначення ймовірності реалізації загрози, тобто математичної оцінки ризику того, що певна загроза може бути реалізована у конкретних умовах. Ця оцінка дозволяє розставити пріоритети при розробці захисних заходів.

У класичних методах визначення ймовірностей використовуються експертні оцінки, статистичні підходи або аналітичні моделі. Проте такі підходи мають низку недоліків: суб’єктивність оцінок, обмеженість даних, а також неврахування складних нелінійних зв’язків між параметрами. Саме тому останніми роками все більшої популярності набувають методи машинного навчання.

1.2. Штучний інтелект у задачах кібербезпеки

Штучний інтелект (ШІ) та його підгалузь – машинне навчання (ML) – відіграють дедалі важливішу роль у сфері кіберзахисту. Алгоритми машинного навчання здатні аналізувати великі обсяги даних, виявляти приховані залежності та автоматично вдосконалювати свої прогнози на основі нових даних.

У контексті кібербезпеки ШІ використовується для:

класифікації типів загроз та виявлення аномалій у мережевому трафіку;

прогнозування ймовірності реалізації загроз на основі історичних даних;

розпізнавання шкідливих дій та фішингових повідомлень;

аналізу поведінкових патернів користувачів для виявлення інсайдерських загроз.

Завдяки здатності ШІ моделювати складні взаємозв’язки між факторами, він відкриває нові можливості для автоматизованої оцінки ризиків і проактивного реагування на кіберінциденти.

1.3. Нейронні мережі в аналізі загроз

Серед різних підходів штучного інтелекту особливе місце посідають нейронні мережі (ШНМ). Вони імітують роботу людського мозку, складаючись із шарів штучних нейронів, які навчуються через оптимізацію вагових коефіцієнтів.

Залежно від завдання, використовуються різні архітектури нейронних мереж:

Багатошарові перцептрони (MLP) – для табличних даних, як у задачі оцінки ймовірності реалізації загрози.

Рекурентні мережі (RNN, LSTM) – для аналізу послідовностей, наприклад логів безпеки або часових рядів атак.

Трансформери та мовні моделі (BERT, GPT) – для обробки природної мови в описах загроз, класифікації текстових повідомлень чи звітів.

Використання нейронних мереж дозволяє виявляти нелінійні та багатовимірні залежності між характеристиками загроз, що є практично недосяжним для традиційних статистичних методів. У поєднанні з техніками аугментації даних, регуляризації та зважування класів, нейронні мережі забезпечують високу точність і стійкість моделей навіть за обмеженого обсягу даних.

1.4. Перспективи застосування нейронних мереж у кіберзахисті

Впровадження нейронних мереж у системи кібербезпеки має низку перспективних напрямів:

Автоматична класифікація загроз: аналіз текстових описів та визначення їх належності до певних категорій без участі експерта.

Прогнозування ймовірності реалізації: оцінювання ризику на основі множини технічних та поведінкових параметрів.

Інтелектуальні системи виявлення аномалій: побудова моделей нормальної поведінки системи для автоматичного виявлення відхилень.

Інтеграція у системи SOC (Security Operation Center): допомога аналітикам у пріоритизації інцидентів і формуванні звітів.

Попри значні досягнення, залишаються виклики – забезпечення пояснюваності моделей, робота з обмеженими або незбалансованими наборами

даних, а також потреба у формуванні стандартів застосування ШІ в інформаційній безпеці.

1.5. Висновок до розділу

Таким чином, сучасні підходи до кіберзахисту поступово переходять від статичних, експертно орієнтованих систем до динамічних, адаптивних моделей на основі штучного інтелекту. Нейронні мережі, завдяки своїй здатності виявляти складні закономірності та узагальнювати інформацію, стають ключовим інструментом для класифікації загроз і визначення ймовірностей їх реалізації. Поєднання цих підходів відкриває шлях до створення інтелектуальних систем кібербезпеки нового покоління.

2. МОДЕЛЬ ПРОГНОЗУВАННЯ ЙМОВІРНОСТІ РЕАЛІЗАЦІЇ ЗАГРОЗ НА ОСНОВІ БАГАТОШАРОВОГО ПЕРЦЕПТРОНА (MLP)

2.1. Підготовка та обробка даних

Якість результатів моделі машинного навчання, особливо нейронної мережі, значною мірою залежить від якості підготовки вхідних даних. Етап підготовки включає збір, очищення, перетворення та нормалізацію інформації для забезпечення коректного навчання. У цьому розділі описано етапи обробки даних, використані для побудови моделі, що передбачає ймовірність реалізації загроз інформаційної безпеки.

2.1.1. Початковий набір даних

У дослідженні використовувався набір даних "Probability_of_threat_realization_dataset.csv", який містив шість стовпців: п’ять вхідних ознак і одну цільову змінну.

Ознаки:

- Confidentiality – вплив загрози на конфіденційність;
- Integrity – вплив на цілісність;
- Availability – вплив на доступність;
- Authenticity – вплив на автентичність;
- Affiliation – ступінь зв’язку або асоційованості загрози.

Цільова змінна:

- Probability_of_threat_realization – дискретний числовий параметр із п’ятьма рівнями: 0.067, 0.133, 0.2, 0.267, 0.333 (від дуже низької до високої ймовірності).

Початковий набір даних містив лише 200 записів, що є недостатнім для навчання глибоких моделей, яким потрібно багато прикладів для формування стійких закономірностей і запобігання перенавчанню.

2.1.2. Дисбаланс класів і його наслідки

Аналіз показав, що клас із ймовірністю 0.333 мав значно більше прикладів, ніж інші. Такий дисбаланс класів може призвести до схильності моделі до домінуючого класу, ігнорування рідкісних випадків, які можуть бути критично важливими та перенавчання через обмежену різноманітність даних.

Для зменшення цих ризиків застосовано методи аугментації даних та зваженої функції втрат, що підвищило стійкість і баланс під час навчання.

2.1.3. Аугментація даних

Через невеликий обсяг і дисбаланс початкового набору було проведено аугментацію – штучне розширення вибірки.

Основні кроки:

1. Ресемплінг із поверненням. Із 200 початкових записів було згенеровано 20 000 нових зразків випадковим вибором із повторенням.

2. Додавання шуму. Унесено випадкові відхилення до ±20% діапазону кожної ознаки. Такий рівень шуму підвищив різноманітність, стійкість і здатність до узагальнення.

3. Об’єднання даних. Нові записи об’єднано з початковими, утворивши аугментований набір із 20 200 записів, збережений як augmented_threat_dataset_20000.csv.

Ця процедура дала змогу створити більший і різноманітніший набір, придатний для навчання нейронної мережі.

2.1.4. Перетворення даних для нейронної мережі

Після аугментації виконано кілька важливих кроків попередньої обробки.

Кодування цільової змінної

Значення (0.067, 0.133, 0.2, 0.267, 0.333) перетворено в цілочисельні класи (0-4) для сумісності з функцією втрат sparse_categorical_crossentropy і активацією softmax. Відповідність збережено для подальшої інтерпретації результатів.

Масштабування ознак (MinMaxScaler)

Щоб усі ознаки мали однакову шкалу й жодна не домінувала, використано MinMaxScaler із діапазоном [0, 1]. Масштабатор навчався лише на тренувальному наборі, щоб уникнути витоку даних.

Розподіл даних

Набір розділено на:

- Тренувальну вибірку – для навчання моделі;
- Валідаційну вибірку – 20% тренувальних даних для контролю перенавчання;
- Тестову вибірку – 20% усього набору для фінальної оцінки.

Використано параметри random_state і stratify для відтворюваності й збереження пропорцій класів.

2.1.5. Використання зважених втрат

Оскільки дисбаланс частково зберігся навіть після аугментації, під час навчання було застосовано Class Weights.

Функція class_weight.compute_class_weight із бібліотеки scikit-learn автоматично визначала ваги, обернено пропорційні частоті класів. Це забезпечувало більшу увагу до рідкісних класів, зменшуючи зміщення й покращуючи баланс.

Передача цих ваг у model.fit() підвищила точність для всіх рівнів ймовірності загрози, а не лише для найпоширенішого класу.

2.1.6. Висновок

У цьому розділі описано повний процес підготовки даних для побудови нейронної мережі. Визначено структуру набору даних, проведено аугментацію, масштабування й кодування, застосовано зважені втрати для компенсації дисбалансу.

Результатом став добре підготовлений, збалансований і розширений набір, який забезпечує ефективне та надійне навчання моделі прогнозування ймовірності реалізації загроз.

2.2. Розробка моделі та експериментальне дослідження

У цьому розділі описано побудову нейронної мережі для класифікації ймовірності реалізації загроз, вибір архітектури та гіперпараметрів, процес навчання й результати оцінки.

2.2.1. Архітектура нейронної мережі

Завдання розв’язувалося за допомогою багатошарового перцептрона (MLP) із прямим поширенням, реалізованого в TensorFlow/Keras. Цей підхід дозволяє виявляти нелінійні залежності між ознаками та цільовою змінною.

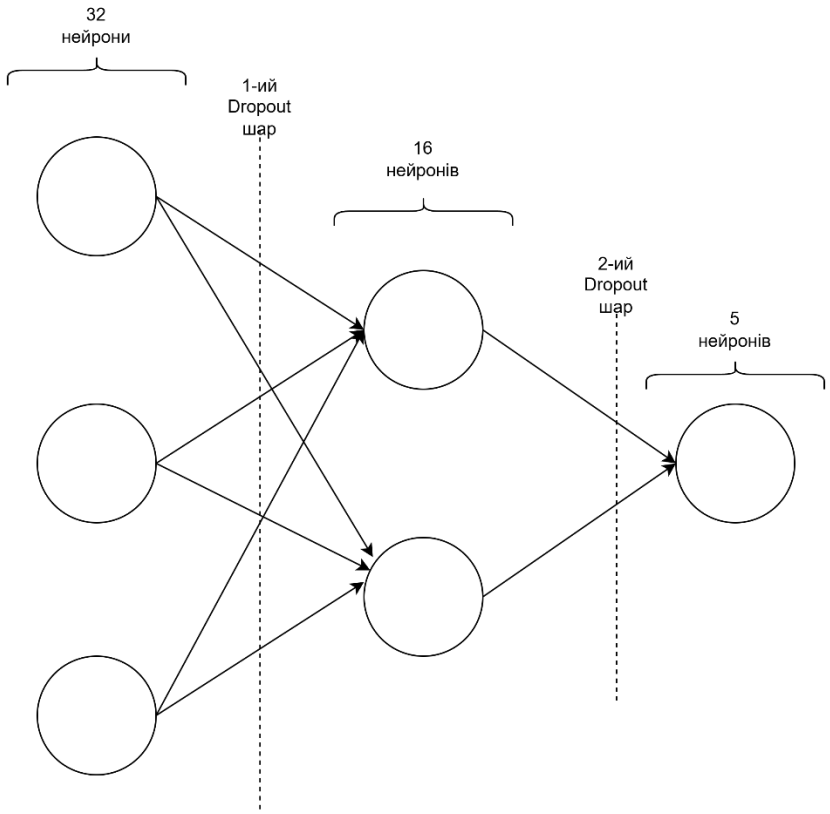

Рис. 2.1 – Архітектура моделі

Склад шарів:

- Вхід: 5 масштабованих ознак (Confidentiality, Integrity, Availability, Authenticity, Affiliation), input_shape = (5,).
- Dense(32, ReLU) – перший прихований шар для нелінійного відображення.
- Dropout(0.2) – регуляризація (вимикає 20% нейронів під час навчання).
- Dense(16, ReLU) – другий прихований шар меншої ширини для поступового узагальнення ознак.

- Dropout(0.2) – додаткова регуляризація.
- Dense(5, Softmax) – вихід для 5 класів ймовірності, сумарна ймовірність = 1.

2.2.2. Гіперпараметри

- Epochs: 50 (фактична кількість контролювалася EarlyStopping);
- Batch size: 64 – баланс між швидкістю та стабільністю градієнта;
- Optimizer: Adam – адаптивні швидкості навчання й швидка збіжність;
- Loss: sparse_categorical_crossentropy – для багатокласової класифікації з цілочисельними мітками (0-4);
- Metric: accuracy – для моніторингу навчання.

2.2.3. Навчання моделі

Навчання виконувалося за допомогою model.fit(X_train_scaled, y_train, ...) із двома ключовими механізмами.

Зважені втрати (Class Weights)

Через домінування класу 0.333 ваги обчислювалися функцією sklearn.utils.class_weight.compute_class_weight на основі розподілу y_train. Недопредставлені класи отримували більші ваги, словник ваг передавався у параметр class_weight функції model.fit().

Раннє припинення навчання (Early Stopping)

Моніторинг val_loss із параметрами patience = 10, restore_best_weights = True. Навчання зупинялося після 10 епох без покращення, після чого відновлювалися найкращі ваги.

Динаміка метрик (за підсумками 50 епох)

- accuracy (train): від ~0.2874 на 1-й епосі до >0.91 наприкінці – ознака ефективного навчання;
- loss (train): стабільно знижувалася – успішна оптимізація;
- val_accuracy: від ~0.5222 до стабілізації на рівні 0.95-0.954, вища, ніж тренувальна, через Dropout, застосований лише під час тренування;
- val_loss: стабільно знижувалася без зростання, EarlyStopping спрацював на 50-й епосі, найкращі ваги – 49-та епоха (val_loss = 0.1325).

Висновок: процес навчання був стабільним, регуляризація й зважування класів запобігли перенавчанню та покращили результати для рідкісних класів.

2.2.4. Оцінка на аугментованому тестовому наборі

Модель перевірялася на відкладеній частині аугментованих даних, не використаних під час навчання й валідації.

Test Accuracy: 0.9494 (94.94%) – високий рівень якості класифікації на згенерованих, але раніше невідомих моделі прикладах.

2.2.5. Тестування на реальному незалежному прикладі

Для якісної перевірки здатності до узагальнення модель протестували на одному реальному записі, відсутньому як у початковому, так і в аугментованому наборах.

Новий вектор ознак: [0.0096, 0.0085, 0.0087, 0.0084, 0.0092].

Масштабування виконано за допомогою того ж MinMaxScaler, навченного під час тренування моделі.

Прогноз моделі: клас 3 – ймовірність 0.267 – правильна класифікація для цього випадку.

Хоч один приклад не є статистично значущим, коректний результат свідчить, що модель не запам’ятала дані, а засвоїла стійкі закономірності. Це підтверджує ефективність аугментації з 20% шумом і зважуванням класів.

2.2.6. Висновок

Реалізовано архітектуру MLP із регуляризацією Dropout і оптимізатором Adam, проведено налаштування гіперпараметрів, використано зважені втрати та ранню зупинку. Модель досягла точності 94.94% на тестових даних і правильно класифікувала незалежний реальний приклад.

Результати демонструють високу здатність до узагальнення та практичну придатність для прогнозування ймовірності реалізації загроз.

3. МОДЕЛЬ КЛАСИФІКАЦІЇ ЗАГРОЗ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ НА ОСНОВІ ОБРОБКИ ТЕКСТОВИХ ОПИСІВ

3.1. Робота з даними

3.1.1 Структура даних

Для виконання поставленої задачі було використано дані з класифікатора у сфері кібербезпеки[1] – таблицю аналізу загроз із розрахованими класами та коефіцієнтами послуг безпеки. Дані стосуються 220 загроз банківського сектору, оцінених експертним шляхом, і були зведені в окремий файл зі сталою структурою.

Основні елементи структури:

- Кортеж загрози – рядок із восьми числових показників, розділених крапками, що відображають класифікацію за вісьмома критеріями.
- Опис загрози – короткий текстовий опис сутності загрози.
- Компоненти кортежу:
    - Рівень критичності (01-05)
    - Стан безпеки (01-03)
    - Тип послуги безпеки (01-05)
    - Характер спрямованості (01-03)
    - Рівень інфраструктури ISO/OSI (01-07)
    - Ознака соціальної інженерії (01-05)
    - Контур загрози (01-03)
    - Категорія об’єкта критичної інфраструктури (01-10)
- Також у наборі містяться коефіцієнти послуг безпеки, оцінки впливу та ваговий коефіцієнт, що характеризує частоту реалізації загрози.

3.1.2 Попередня обробка

Для виконання дослідження були залишені стовпці з елементами кортежу та описами загроз. Початковий набір містив кілька загроз під одним кортежем, тому його потрібно було переформатувати – тепер групування виконується за описом загрози.

Дані не містили пропусків або пошкоджених значень, однак не кожна загроза мала повний набір параметрів.

Для збереження відповідності між первинними та обробленими описами до таблиці додано стовпець “original”, який нумерує всі 220 унікальних загроз.

3.1.3 Аугментація даних

Загальний опис методики

Для ефективної роботи моделей машинного навчання необхідний достатньо великий і різноманітний набір даних.

Оскільки 220 записів було недостатньо, застосовано аугментацію даних[8] – штучне розширення набору шляхом створення модифікованих варіацій існуючих описів.

Основні методи аугментації текстів:

- синонімія – заміна слів на близькі за змістом;
- багаторазовий переклад – переклад на інші мови з поверненням до вихідної;
- маніпуляції з формою слів – зміна граматичних форм, введення помилок тощо.

У роботі застосовувались перші два методи. Усі створені варіанти зберігались зі збереженням ідентифікатора original, що дозволило простежити походження кожного опису.

Техніка перекладу

Техніка полягала у перекладі описів загроз іншими мовами з подальшим поверненням до української. Реалізацію здійснено за допомогою бібліотеки deep_translator, використовуючи різні перекладачі та кількість ітерацій.

Результати виявились неоднорідними: частина описів майже не змінювалась, інші – втрачали зміст через спотворення термінів та абревіатур. Це пояснюється тим, що мова описів має професійну специфіку, а перекладачі не завжди адекватно обробляють такі тексти.

У підсумку метод не забезпечив задовільного результату й не був використаний надалі.

Синонімія та парафразування

Другий метод – парафразування – показав набагато кращі результати.

Синонімія передбачала заміну окремих слів, а парафразування – глибше переформулювання речень зі зміною структури.

Для реалізації було застосовано кілька моделей із платформи Hugging Face: bart-paraphrase, paraphrase-MiniLM, Rephrase.

Отримані результати були якіснішими: більшість нових описів зберігали зміст, хоча траплялися дублікати та окремі спотворені варіанти.

Ручна перевірка

На фінальному етапі виконано ручну перевірку: дублікати видалено, частину описів відредаговано вручну.

Через появу ком у парафразованих текстах деякі записи взято в лапки для коректного зчитування у CSV.

ThreatDescription	CriticalityLevel	SecurityStatus	SecurityServices	ThreatDirection	ISOOSILevel	SocialEngineerir	ThreatContour	Original
Загроза заміни моделі машинного навчання	2	2	2	1	2	2	3	0
Загроза спотворення XML-схеми	2	2	2	1	2	2	3	1
Загроза модифікації моделі машинного навчання шляхом спотворення («отруєння») навчальних даних	2	2	2	1	7	3	3	2
Загроза порушення функціонування (обходу) засобів, що реалізують технології штучного інтелекту	2	2	2	1	6	4	2	3
Загроза розкрадання навчальних даних	2	2	1	1	7	3	3	4
Загроза розкриття інформації про модель машинного навчання	2	1	2	1	7	3	3	5
Загроза використання скомпрометованого довіреного джерела оновлень програмного забезпечення	1	2	2	1	1	1	3	6
Загроза отримання несанкціонованого доступу до програм, встановлених на Smart-картах	2	2	3	1	5	1	2	7
Загроза несанкціонованого доступу до системи за допомогою сторонніх сервісів	2	2	5	1	5	2	2	8
Загроза несвоєчасного виявлення та реагування компонентами інформаційної (автоматизованої) системи (у тому чис	3	2	3	1	7	2	1	9
Загроза обходу багатофакторної автентифікації	2	2	1	1	3	3	2	10
Загроза перехоплення управління інформаційною системою	1	2	1	1	7	3	1	11
Загроза використання неперевірених даних користувача при формуванні конфігураційного файлу, який використовує	3	2	3	2	3	3	1	12
Загроза порушення роботи інформаційної системи, спричиненого оновленням програмного забезпечення, що викори	3	2	3	1	3	1	1	13
Загроза несанкціонованого доступу до пам'яті, що захищається ядра процесора	3	2	2	1	1	1	3	14
Загроза нецільового використання обчислювальних ресурсів засобу обчислювальної техніки	3	2	5	1	1	1	3	15
Загроза несанкціонованого доступу до параметрів налаштування обладнання за рахунок використання «майстер-код	2	2	3	1	1	1	3	16
Загроза відмови у роботі обладнання через зміну геолокаційної інформації про нього	4	2	3	1	1	1	2	17
Загроза порушення роботи комп'ютера та блокування доступу до його даних через некоректну роботу встановлених	3	2	3	1	3	4	2	18
Загроза несанкціонованої зміни шкідливою програмою значень параметрів логічних контролерів, що програмуються.	3	2	3	1	1	1	2	19
Загроза витоку інформації з непідключених до Інтернету комп'ютерів	3	3	1	1	1	3	3	20
Загроза несанкціонованої установки програм на мобільні пристрої	2	2	1	1	2	1	2	21
Загроза витоку даних користувача при використанні функцій автоматичного заповнення автентифікаційної інформаці	2	1	1	1	2	5	2	22
Загроза розкрадання інформації з мобільного пристрою під час використання віртуальних голосових помічників	2	1	1	1	5	3	2	23
Загроза перехоплення керування мобільного пристрою під час використання віртуальних голосових помічників	3	2	1	1	1	4	2	24
Загроза прихованої реєстрації шкідливою програмою облікових записів адміністраторів	2	2	1	1	3	2	2	25
Загроза розкрадання автентифікаційної інформації з тимчасових файлів cookie	2	1	2	1	5	5	2	26
Загроза контролю шкідливої програми списку програм, запущених на мобільному пристрої	2	2	2	1	3	5	2	27
Загроза віддаленого запуску шкідливого коду в обхід механізмів захисту операційної системи	2	2	5	1	3	1	2	28
Загроза несанкціонованого використання привілейованих функцій мобільного пристрою	4	2	1	1	1	1	2	29
Загроза витоку інформації за рахунок застосування шкідливого програмного забезпечення алгоритмів шифрування т	2	2	3	1	5	5	2	30
Загроза використання вразливих версій програмного забезпечення	2	2	1	1	5	1	3	31
Загроза впровадження шкідливого коду у дистрибутив програмного забезпечення	3	2	1	1	2	1	3	32
Загроза незахищеного адміністрування хмарних послуг	3	2	1	1	2	1	3	33
Загроза впровадження шкідливого коду за рахунок відвідування заражених сайтів у мережі Інтернет	2	2	2	1	2	1	2	34
Загроза маскування дій шкідливого коду	3	2	4	1	3	1	2	35

Рис. 3.1 – Датасет після аугментації та редакції

Фінальний набір містить 1078 рядків і збережений у файлі full_threats_expanded_fixed.csv. Кожна загроза тепер має від 3 до 5 варіантів опису, що суттєво підвищує різноманітність даних.

3.1.4 Висновки

Проведена робота забезпечила якісну підготовку даних для подальшого навчання нейронної мережі.

Було реалізовано попередню обробку, очищення та аугментацію даних, що дозволило збільшити обсяг датасету майже у п’ять разів.

Доданий стовпець “original” забезпечив зв’язок між усіма варіаціями одного опису, а ручне редагування гарантувало цілісність змісту.

Таким чином, сформований набір став більш репрезентативним і придатним для моделювання, забезпечуючи достатню кількість різноманітних прикладів для аналізу загроз у банківській сфері.

3.2. Розробка нейронної мережі

3.2.1 Необхідні інструменти та підготовка середовищав

Підготовано робоче середовище Jupyter (ноутбук classification.ipynb). Бібліотеки імпортувалися в окремій комірці та додавалися за потреби.

Використані бібліотеки та їх призначення:

- pandas – обробка/аналіз табличних даних (датасет загроз).
- jupyterlab – інтерактивне середовище для написання, тестування й візуалізації коду.
- sklearn – моделювання, препроцесинг ознак, оцінювання якості.
- numpy – числові обчислення, масиви.
- matplotlib – графіки й візуалізація результатів.
- seaborn – статистичні графіки, теплокарти.
- joblib – збереження/завантаження моделей.
- pickle – серіалізація Python-об’єктів.
- re – регулярні вирази для очищення тексту.
- hashlib – хеші для унікальної ідентифікації та кешу.
- functools – кешування (наприклад, lru_cache).

3.2.2 Попередня обробка

Завантаження даних і цільові мітки

Завантажено full_threats_expanded_fixed.csv у DataFrame.

Ознаки: текстовий опис загрози.

Цілі: вісім стовпців кортежу (категоріальні мітки).

Поділ на тренувальні та тестові дані

Групування виконується за полем original (усі описи однієї загрози – одна група). Із кожної групи 1 опис відправляється в тестовий набір, решта – у тренувальний. Частка тесту становить приблизно 17-25%, зате у тесті присутні 100% унікальних загроз.

Кодування категоріальних ознак

Кожен клас у складових кортежу перетворюється у числовий діапазон 0…(n-1) (де n – кількість наявних класів). Виявлено дисбаланс: загрози розподілені нерівномірно, що ускладнює класифікацію окремих класів і підвищує ризик похибок.

3.2.3 Побудова моделі

Модель реалізовано як pipeline із трьох послідовних компонентів: підготовка тексту – векторизація – багатовихідна класифікація.

Рис. 3.2 – Структура моделі

Клас підготовки даних – TextCleaner.

Базова очистка: зниження регістру, видалення цифр, спецсимволів і пунктуації.

Лематизація: бібліотека Stanza[11] (підтримує українську, глибинні моделі для морфології, частин мови та залежностей). Перший запуск ініціалізує модель, далі працює кеш.

Кешування:

На диску: результати лематизації у lemma_cache.pkl.

У пам’яті: кеш для базової очистки, щоб не повторювати операції над однаковими рядками.

TF-IDF векторизатор: перетворює очищені описи на числові вектори за важливістю термів (TF) та рідкістю у корпусі (IDF). Застосовує українські стоп-слова (службова лексика вилучається до класифікації).

Багатовихідний класифікатор: використано MultiOutputClassifier для одночасного прогнозування 8 незалежних міток. Базовий алгоритм – RandomForestClassifier (добре працює з TF-IDF, стійкий до перенавчання відносно альтернатив).

3.2.4 Тренування та підбір гіперпараметрів

Застосовано GridSearchCV для пошуку найкращих налаштувань із крос-валідацією, діапазони параметрів коригувалися вручну за останніми результатами. У фінальному пайплайні зафіксовані найкращі значення для швидкого відтворення.

Додатково вручну зменшено вагу слів «загроза», «небезпека», «ризик», щоб ці універсальні терміни не домінували в рішенні моделі.

3.2.5 Оцінка моделі

Використано набір стандартних метрик і візуалізацій (класифікаційні звіти, матриці помилок, криві навчання).

Основні спостереження:

- Класифікаційні звіти демонструють зв’язок між дисбалансом класів і нерівномірністю precision/recall, місцями простежуються ознаки перенавчання.
- Матриці помилок показують, які саме класи плутаються найчастіше.
- Криві навчання: тренувальна точність дуже висока, крос-валідаційна – нижча, але зростає із додаванням даних. Це натякає на недостатню різноманітність і обсяг датасету.

Висновок щодо поліпшення: найефективніший шлях – розширення й урізноманітнення даних (нові експертні анотації, покращення покриття класів).

3.2.6 Висновки

Побудовано повний конвеєр: очистка + лематизація (Stanza) з кешем – TF-IDF – MultiOutput(RandomForest). Дані завантажено, цілі задано, поділ із використанням original гарантує присутність усіх загроз у тесті. Проведено GridSearchCV, зафіксовано найкращі параметри, відрегульовано ваги загальних термінів.

Оцінка показала гарну тренувальну точність із ознаками overfitting через дисбаланс та обмежений обсяг даних. Пріоритет подальшої роботи – поповнення датасету й балансування класів для підвищення узагальнювальної здатності моделі.

3.3. Реалізація системи класифікації

3.3.1 Структура повної системи класифікації

Розроблена система класифікації поєднує нейронну мережу з додатковим алгоритмом перевірки подібності введеного користувачем опису до вже наявних у базі загроз.

Загальна схема роботи подана на рисунку 3.2.

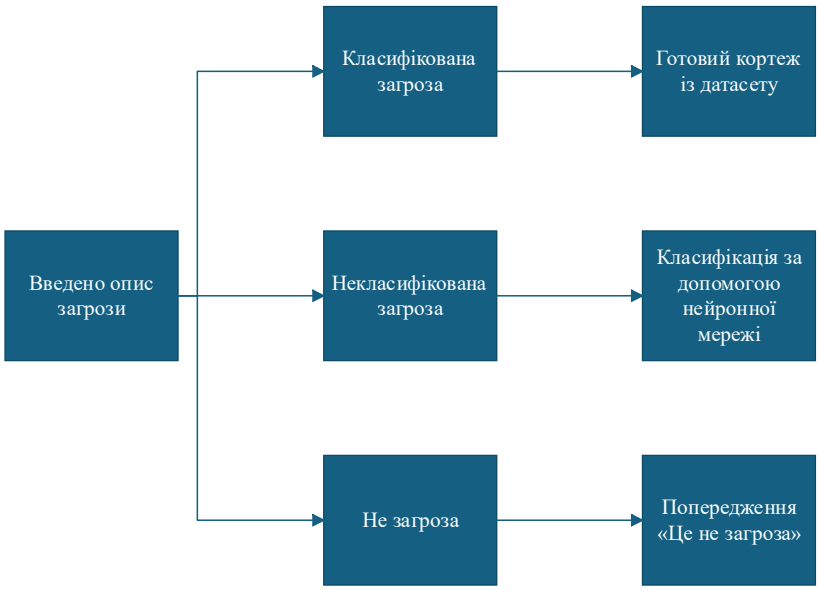

Рис. 3.3 – Схема роботи класифікатора

Якщо користувач вводить опис загрози, який повністю або майже збігається з уже відомим описом із датасету, модель не залучається – результат береться безпосередньо з попередньо проведеної експертної оцінки.

Якщо ж введений опис суттєво відрізняється від усіх відомих зразків, система повідомляє, що цей опис не належить до загроз інформаційної безпеки.

У проміжному випадку, коли подібність є помірною, класифікацію здійснює нейронна мережа, визначаючи клас і параметри загрози.

Такий підхід забезпечує швидку реакцію при повторних описах, знижує навантаження на модель і мінімізує кількість хибних класифікацій.

3.3.2 Програмна реалізація

Клас повної класифікації – ThreatClassifier. Основу реалізації становить клас ThreatClassifier, який виконує повний цикл обробки даних:

1. Завантаження збережених об’єктів – моделі, кодувальників, датафреймів ознак і міток. Це дозволяє відновити стан навченої системи без повторного тренування.

2. Передобробка введеного опису – очищення й лематизація тексту за допомогою попередньо кешованого лематизатора.

3. Векторизація – перетворення тексту у числовий вектор через TF-IDF векторизатор, ідентичний тому, що використовувався під час навчання моделі.

4. Оцінка подібності – обчислення косинусної подібності між вектором введеного опису та векторами відомих загроз.

Залежно від значення коефіцієнта подібності результат класифікації розподіляється на три категорії:

висока подібність – знайдена загроза з бази (нейронна мережа не викликається);

низька подібність – введений опис не є загрозою інформаційної безпеки;

середня подібність – опис передається нейронній мережі для повної класифікації.

Таким чином, ThreatClassifier є не лише інтерфейсом до моделі, а й контролюючою логікою, що вирішує, коли доцільно застосовувати глибоку класифікацію.

Користувацький інтерфейс

Для демонстрації роботи системи створено простий текстовий консольний інтерфейс, що дозволяє користувачу ввести опис загрози.

Після цього клас ThreatClassifier обробляє введення через метод predict_threat(), виконує векторизацію, оцінює подібність і, за потреби, передає дані в нейронну мережу.

У результаті користувач отримує:

- повідомлення про збіг із уже відомою загрозою;
- результат класифікації, якщо опис новий;
- або вказівку, що опис не є загрозою.

Завдяки модульній структурі інтерфейс легко адаптується до будь-якого типу застосунку – від настільних систем до веб-платформ. Приклад реалізації наведено на рисунку 3.3.

```
Система класифікації загроз готова. Введіть опис загрози.
Поріг подібності для визначення ймовірної загрози встановлено на: 0.40
Поріг подібності для використання 'істинного' прогнозу з датасету: 0.85

Введіть опис загрози інформаційної безпеки (або 'вихід' для завершення):  Загроза обходу багатофакторної автентифікації
Прогнозований клас загрози: Загроза обходу багатофакторної автентифікації -> 2.2.1.1.3.3.2.4
  Подібність: 1.00

Введіть опис загрози інформаційної безпеки (або 'вихід' для завершення):  загроза обходу двофакторки
Прогнозований клас загрози: загроза обходу двофакторки -> 2.2.1.1.2.1.3.4
  Подібність: 0.53

Введіть опис загрози інформаційної безпеки (або 'вихід' для завершення):  я люблю морозиво
Це, скоріш за все, не загроза інформаційної безпеки.
  Максимальна подібність до відомих загроз: 0.12 (нижче порогу 0.40)
  Якщо б це була загроза, прогнозований клас: я люблю морозиво -> 2.2.1.1.2.1.3.4

Введіть опис загрози інформаційної безпеки (або 'вихід' для завершення):  вихід
Завершення роботи.
```

Рис. 3.4 – Приклад користувацького інтерфейсу

3.3.3 Висновки

У цьому розділі реалізовано повноцінну систему класифікації загроз інформаційної безпеки, що поєднує інтелектуальні методи обробки текстів та перевірку подібності.

Центральним елементом став клас ThreatClassifier, який об’єднує передобробку тексту, TF-IDF векторизацію, аналіз косинусної подібності та роботу нейронної мережі.

Такий підхід робить систему:

- адаптивною – вона автоматично визначає, чи потрібне повне моделювання;
- надійною – зменшує кількість помилкових класифікацій;
- ефективною – економить ресурси, не перевантажуючи модель відомими даними.

Додатково реалізований консольний інтерфейс забезпечує просту взаємодію користувача з алгоритмом. Завдяки модульності рішення може бути легко інтегроване до будь-яких систем кіберзахисту як окремий компонент або ядро автоматизованого класифікатора кіберзагроз.

ВИСНОВОК

У ході дослідження було розроблено та реалізовано комплексний підхід до класифікації та прогнозування ймовірності реалізації загроз інформаційної безпеки із застосуванням методів штучного інтелекту, зокрема нейронних мереж. У роботі детально проаналізовано сучасний стан проблеми оцінювання ризиків у кіберзахисті, визначено недоліки традиційних статистичних і експертних методів, а також обґрунтовано доцільність переходу до гнучких, самонавчальних моделей, здатних працювати з реальними даними в умовах динамічного кіберсередовища.

У першій моделі реалізовано багатошаровий перцептрон (MLP) для прогнозування ймовірності реалізації кіберзагроз had основі п’яти ключових параметрів (Confidentiality, Integrity, Availability, Authenticity, Affiliation). Використання методів аугментації даних, масштабування, зважених втрат і регуляризації дозволило досягти високої точності (94,94%) та забезпечити стійкість моделі до дисбалансу даних.

Друга модель орієнтована на класифікацію текстових описів загроз. Вона поєднує TF-IDF векторизацію, багатовихідну класифікацію (MultiOutputClassifier) та RandomForest, що дозволило ефективно обробляти природну мову. Завдяки застосуванню лематизації українських текстів (Stanza) і методів парафразування було розширено базу даних майже у п’ять разів, підвищивши узагальнювальну здатність моделі.

Дані моделі було успішно інтегровано до класифікатора кібербезпеки[1], що є гарною демонстрацією можливостей їх впровадження.

Розроблена система має значний потенціал практичного впровадження в індустрії. Зокрема, її можна інтегрувати у:

- SOC (security operation center) – для автоматичного аналізу інцидентів та пріоритизації подій;

- банківські та фінансові установи – для виявлення та прогнозування загроз у критичних інформаційних системах;
- державні структури та корпоративні мережі – для створення адаптивних систем ризик-менеджменту;
- платформи кіберзахисту нового покоління – як модуль інтелектуальної класифікації та оцінки ризиків у режимі реального часу.

Поєднання двох підходів – аналітичного (кількісна оцінка) та семантичного (текстова класифікація) – створює основу для інтелектуальної системи кіберзахисту, здатної не лише реагувати на вже відомі загрози, але й проактивно прогнозувати нові ризики.

Таким чином, результати дослідження підтверджують ефективність застосування нейронних мереж у задачах класифікації та оцінювання ймовірностей реалізації загроз. Запропонована система може стати базовим компонентом для створення адаптивних, самонавчальних рішень у сфері інформаційної безпеки, що відповідають сучасним вимогам цифрової інфраструктури.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Класифікатор загроз [Електронний ресурс]. URL: https://skl.khpi.edu.ua/threat-analysis
2. Keras documentation [Електронний ресурс]. URL: https://www.tensorflow.org/guide/keras
3. Pandas documentation [Електронний ресурс]. URL: https://pandas.pydata.org/docs/
4. Class weights [Електронний ресурс]. URL: https://stackoverflow.com/questions/44716150/how-can-i-assign-a-class-weight-in-keras-in-a-simple-way
5. MinMaxScaler [Електронний ресурс]. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
6. Resample method [Електронний ресурс]. URL: https://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html
7. Neural networks for cybersecurity [Електронний ресурс]. URL: https://www.sciencedirect.com/science/article/pii/S0925231222007184
8. What is data augmentation? [Електронний ресурс]. URL: https://www.ibm.com/think/topics/data-augmentation
9. Hugging Face [Електронний ресурс]. URL: https://huggingface.co/
10. Scikit-learn: Machine Learning in Python [Електронний ресурс]. URL: https://scikit-learn.org/stable/index.html
11. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages [Електронний ресурс]. URL: https://arxiv.org/abs/2003.07082
12. Python Documentation [Електронний ресурс]. URL: https://docs.python.org
13. Ian Pointer. Programming PyTorch for Deep Learning. O`Reilly Media. 2019. 217 с.