ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

кваліфікаційна робота

Метод виявлення мережних аномалій з використанням машинного навчання

Виконала:
студентка гр. СПм-23-5
Глоба Є.Ю.

Керівник:
доц. каф. ЕОМ,
к.т.н. Федорченко В.М.

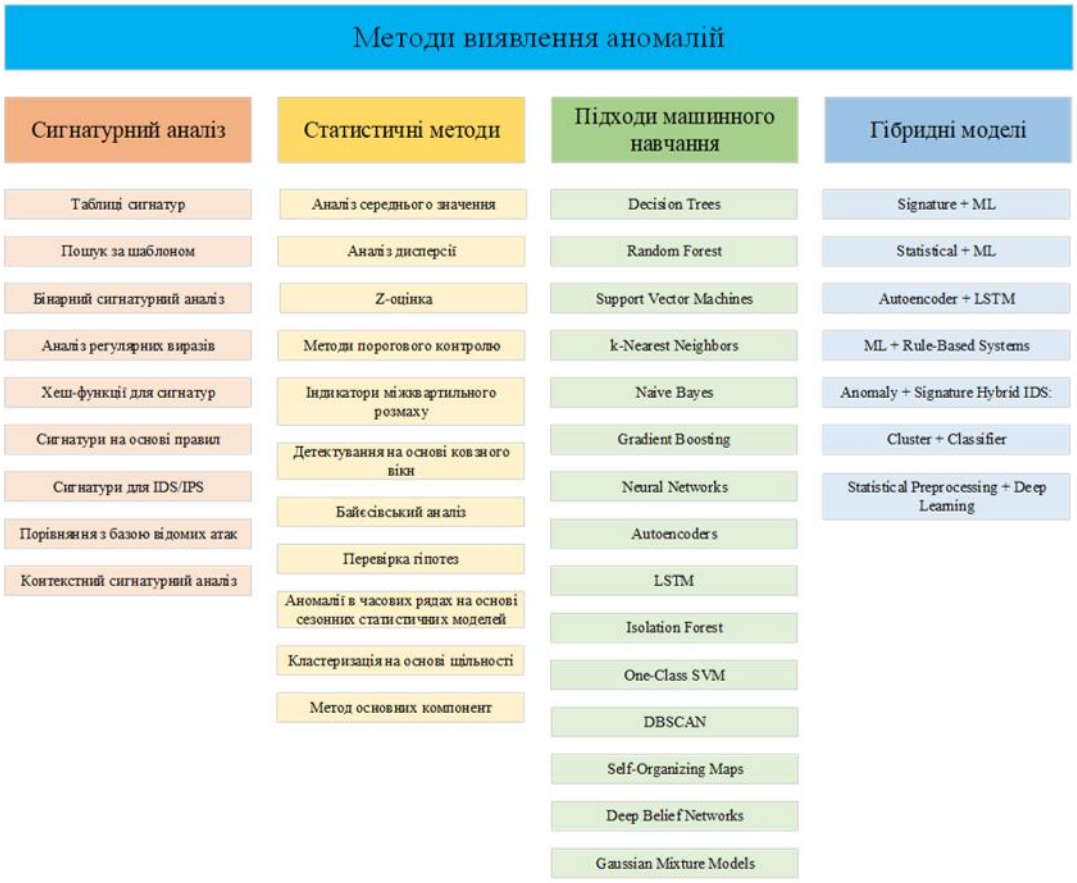
Мета роботи та завдання 2

Метою роботи є розробка, реалізація та експериментальне обґрунтування удосконаленого методу виявлення аномалій у корпоративній мережі, що базується на гібридному нейромережевому підході та здатен забезпечити високу точність, швидкодію й адаптивність в умовах динамічного мережевого середовища.

Об'єктом дослідження є мережевий трафік корпоративної інформаційної системи, який аналізується з метою виявлення аномальної поведінки, що може свідчити про потенційні порушення інформаційної безпеки, такі як кіберзагрози, технічні збої або несанкціоновані дії всередині мережі..

Задачі:

- проаналізувати сучасні методи виявлення аномалій, включаючи статистичні, сигнатурні, евристичні та алгоритми машинного навчання, з метою виявлення їхніх сильних і слабких сторін;
- сформулювати вимоги до ефективного методу виявлення аномалій з урахуванням специфіки корпоративного трафіку, масштабованості та реального часу обробки;
- розробити алгоритм виявлення аномалій на основі гібридної нейромережевої архітектури з урахуванням особливостей попередньої обробки мережевих даних;
- реалізувати програмний прототип запропонованого методу з використанням сучасних інструментів і бібліотек (зокрема Python, TensorFlow або PyTorch);
- створити тестове середовище для моделювання нормального та аномального трафіку з метою перевірки ефективності реалізованого рішення;
- провести експериментальну оцінку методу за критеріями точності, чутливості, швидкодії та витрат ресурсів.



3

Порівняльний аналіз існуючих методів

4

Метод	**Переваги**
Сигнатурний аналіз	Висока точність для відомих атак; швидкість реагування; простота реалізації
Статистичні методи	Можливість виявляти невідомі аномалії; інтерпретованість результатів; незалежність від баз загроз
Машинне навчання	Адаптивність до змін мережевого середовища; здатність до виявлення прихованих закономірностей
Гібридні моделі	Поєднання переваг сигнатурних та інтелектуальних методів; висока гнучкість і ефективність
Метод	**Недоліки**
Сигнатурний аналіз	Неможливість виявлення нових або обфускованих атак; потреба в постійному оновленні бази сигнатур
Статистичні методи	Низька ефективність у динамічному середовищі; чутливість до вибору статистичних порогів
Машинне навчання	Високі обчислювальні витрати; складність налаштування і потреба у великих масивах даних
Гібридні моделі	Складна інтеграція; необхідність глибокої експертизи для супроводу; збільшення системних вимог
Метод	**Потреба в навчанні**
Сигнатурний аналіз	Немає
Статистичні методи	Потрібна початкова побудова статистичних моделей
Машинне навчання	Необхідне навчання на великій кількості даних
Гібридні моделі	Необхідне навчання для інтелектуальної компоненти
Метод	**Здатність до виявлення нових атак**
Сигнатурний аналіз	Немає
Статистичні методи	Обмежена
Машинне навчання	Висока
Гібридні моделі	Висока

Метод виявлення аномалій. Загальна структура 5

На першому етапі здійснюється **збір вхідних даних**. Джерелами інформації можуть бути сирі пакети мережевого трафіку, NetFlow-записи, журнали доступу, логи систем безпеки, проксі-серверів або SIEM-платформ. Дані мають охоплювати широкий спектр параметрів, зокрема IP-адреси, порти, протоколи, часові мітки, обсяги переданої інформації, кількість з'єднань тощо.

Далі відбувається **попередня обробка**, яка включає очищення від шумів і пошкоджених записів, нормалізацію масштабів числових значень, обчислення похідних характеристик і агрегацію даних до рівня потоків або з'єднань. У цьому ж модулі формується вектор ознак, придатний для подачі на вхід моделі машинного навчання.

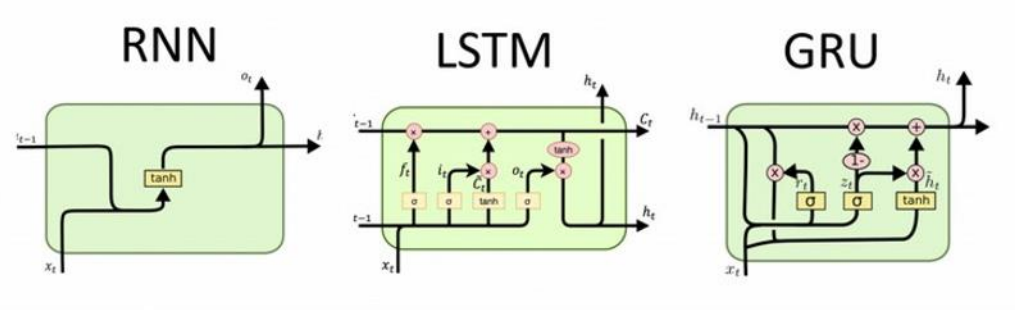
Наступний етап передбачає **побудову профілю "нормальної" поведінки** на основі історичних або довірених даних. За допомогою навчання без учителя (unsupervised learning), як-от автоенкодерів або кластеризації, створюється модель, яка відтворює характерну структуру типового трафіку. У разі гібридного підходу до цієї моделі можуть бути додані рекурентні компоненти (LSTM), які враховують часову залежність подій.

У процесі **реального функціонування системи** нові вхідні дані аналізуються на відповідність побудованій моделі. Визначальною ознакою аномалії є значення похибки реконструкції або класифікаційна оцінка, що виходить за межі допустимого порогу. При перевищенні цього порогу трафік вважається підозрілим, формується інцидент, який може бути переданий до системи реагування або для ручного перегляду.

Додатково реалізується **модуль логування та моніторингу**, який фіксує виявлені аномалії, записує статистику роботи, генерує звітність та забезпечує візуалізацію результатів.

Фінальним етапом є **динамічне оновлення поведінкового профілю**, що базується на повторному навчанні моделі із застосуванням нових даних, які підтверджено як "безпечні". Це дозволяє системі адаптуватися до змін у користувацьких шаблонах, зміни графіку роботи, появи нових сервісів або технологій доступу без втрати чутливості до загроз.

Вибір підходу на основі гібридної нейромережевої архітектури 6



7

Критерії ефективності методу

Критерій	Опис
Точність класифікації	Загальна здатність правильно класифікувати трафік як нормальний або аномальний
Чутливість	Здатність виявити всі істинні аномалії (мінімізує хибнонегативні спрацювання)
Прецизійність	Точність серед виявлених аномалій (мінімізує хибнопозитивні спрацювання)
F1-міра	Гармонійне середнє між показниками точності та повноти
AUC-ROC	Узагальнена здатність моделі відокремлювати класи незалежно від порогу
Час реакції	Час між появою аномалії та її виявленням
Пропускна здатність	Кількість оброблених зразків за одиницю часу
Масштабованість	Можливість масштабування моделі до більших обсягів даних
Ресурсна ефективність	Здатність моделі працювати на обмежених ресурсах
Стійкість до змін	Здатність адаптуватися до нових умов без перенавчання
Узагальнювальна здатність	Стійкість моделі до нових сценаріїв і здатність узагальнювати знання

8

Архітектура системи виявлення аномалій у корпоративній мережі

9

Вибір програмних засобів

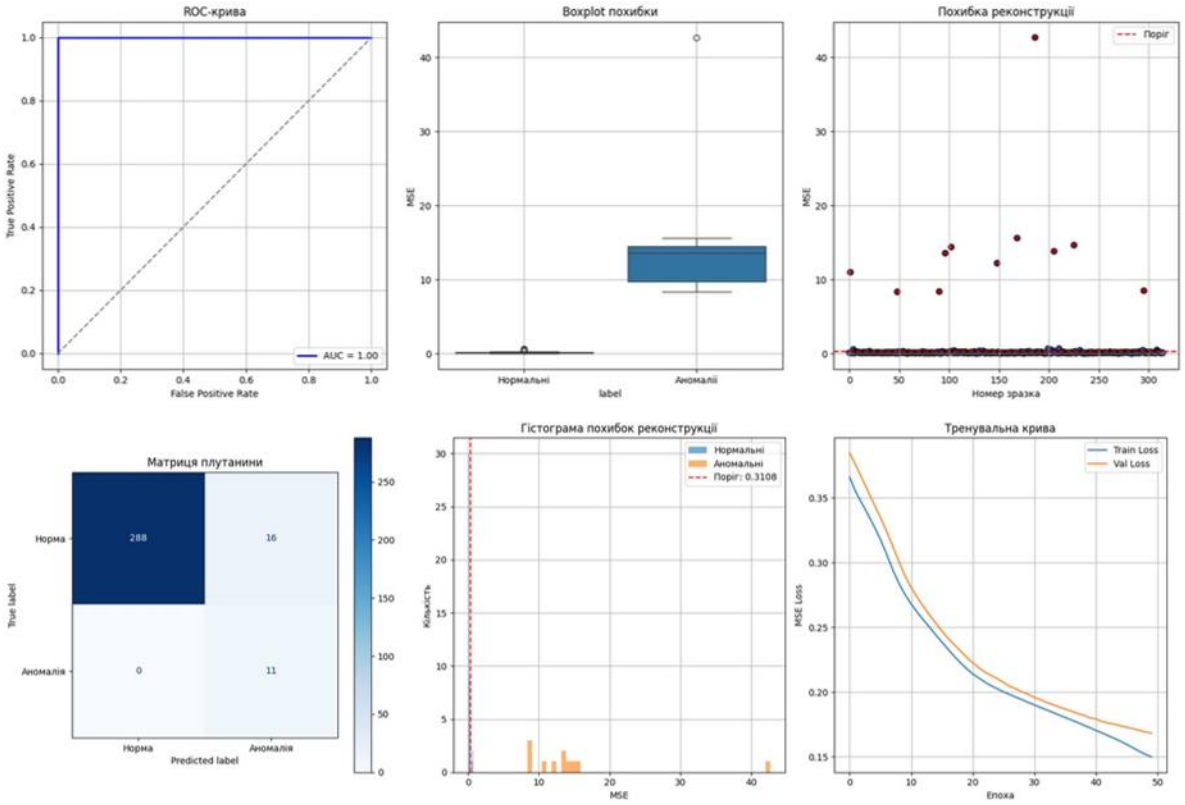
10

Фрагмент коду програми. Процес навчання

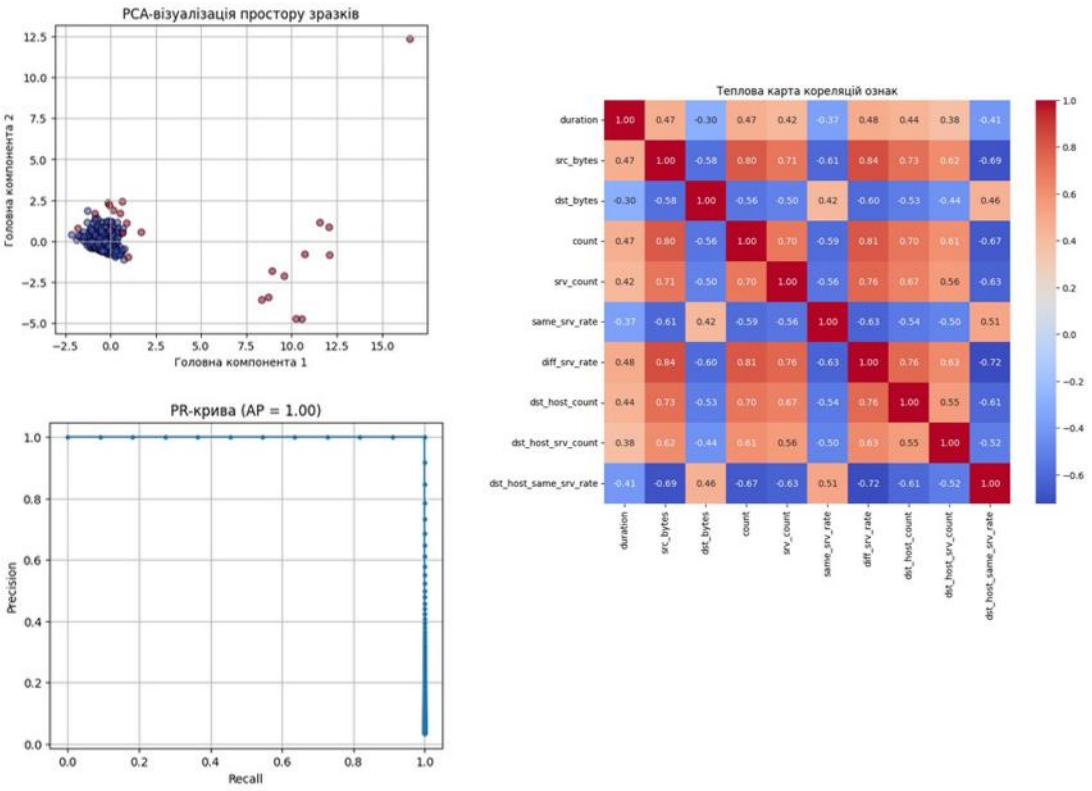
```python
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, auc, confusion_matrix, ConfusionMatrixDisplay
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt
import seaborn as sns

# Генерація даних
def generate_data():
    np.random.seed(42)
    n_normal = 1000
    n_anomalous = 50
    normal_data = {
        "duration": np.random.exponential(2.0, n_normal),
        "src_bytes": np.random.normal(300, 50, n_normal),
        "dst_bytes": np.random.normal(250, 60, n_normal),
        "count": np.random.randint(1, 20, n_normal),
        "srv_count": np.random.randint(1, 15, n_normal),
        "same_srv_rate": np.random.uniform(0.5, 1.0, n_normal),
        "diff_srv_rate": np.random.uniform(0.0, 0.2, n_normal),
        "dst_host_count": np.random.randint(10, 100, n_normal),
        "dst_host_srv_count": np.random.randint(5, 100, n_normal),
        "dst_host_same_srv_rate": np.random.uniform(0.6, 1.0, n_normal),
        "label": [0]*n_normal
    }
    anomalous_data = {
        "duration": np.random.exponential(10.0, n_anomalous),
        "src_bytes": np.random.normal(1000, 200, n_anomalous),
        "dst_bytes": np.random.normal(30, 15, n_anomalous),
        "count": np.random.randint(30, 100, n_anomalous),
        "srv_count": np.random.randint(20, 60, n_anomalous),
        "same_srv_rate": np.random.uniform(0.0, 0.3, n_anomalous),
        "diff_srv_rate": np.random.uniform(0.7, 1.0, n_anomalous),
        "dst_host_count": np.random.randint(200, 255, n_anomalous),
        "dst_host_srv_count": np.random.randint(100, 255, n_anomalous),
        "dst_host_same_srv_rate": np.random.uniform(0.0, 0.4, n_anomalous),
        "label": [1]*n_anomalous
    }
    df = pd.concat([pd.DataFrame(normal_data), pd.DataFrame(anomalous_data)], ignore_index=True)
    return df.sample(frac=1.0, random_state=42).reset_index(drop=True)

# 1. Дані
df = generate_data()
X = df.drop(columns=["label"]).values
y = df["label"].values
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
X_train_norm = X_train[y_train == 0]

# 2. Автоенкодер
input_dim = X_train.shape[1]
input_layer = Input(shape=(input_dim,))
encoded = Dense(8, activation='relu')(input_layer)
encoded = Dense(4, activation='relu')(encoded)
decoded = Dense(8, activation='relu')(encoded)
output_layer = Dense(input_dim, activation='linear')(decoded)
autoencoder = Model(inputs=input_layer, outputs=output_layer)
autoencoder.compile(optimizer=Adam(learning_rate=0.001), loss='mse')
history = autoencoder.fit(X_train_norm, X_train_norm, epochs=50, batch_size=32,
```

```
Epoch 26/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1970 - val_loss: 0.2075
Epoch 27/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1991 - val_loss: 0.2044
Epoch 28/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.2036 - val_loss: 0.2024
Epoch 29/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1998 - val_loss: 0.2001
Epoch 30/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1873 - val_loss: 0.1983
Epoch 31/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1878 - val_loss: 0.1958
Epoch 32/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1834 - val_loss: 0.1941
Epoch 33/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1845 - val_loss: 0.1919
Epoch 34/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1839 - val_loss: 0.1903
Epoch 35/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1828 - val_loss: 0.1885
Epoch 36/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - loss: 0.1813 - val_loss: 0.1869
Epoch 37/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1800 - val_loss: 0.1852
Epoch 38/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1748 - val_loss: 0.1835
Epoch 39/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1753 - val_loss: 0.1822
Epoch 40/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1718 - val_loss: 0.1801
Epoch 41/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1751 - val_loss: 0.1792
Epoch 42/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1724 - val_loss: 0.1772
Epoch 43/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1630 - val_loss: 0.1760
Epoch 44/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1615 - val_loss: 0.1751
Epoch 45/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1722 - val_loss: 0.1739
Epoch 46/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1586 - val_loss: 0.1727
Epoch 47/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1604 - val_loss: 0.1716
Epoch 48/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1541 - val_loss: 0.1705
Epoch 49/50
20/20 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 0.1516 - val_loss: 0.1691
Epoch 50/50
```

Результати роботи 11



Результати роботи 12



13

АПРОБАЦІЯ

Є. Ю. Глоба[1], В. Р. Смірнов[1], М.С. Нараєвський[1]

[1] Харківський національний технічний університет радіоелектроніки, Харків, Україна

МЕТОД ВИЯВЛЕННЯ АНОМАЛІЙ В КОРПОРАТИВНІЙ МЕРЕЖІ

Анотація. Актуальність. На сучасному етапі розвитку інформаційних технологій та цифрової економіки корпоративні мережі є критично важливими елементами бізнес-інфраструктури. Ефективне функціонування корпоративних мереж забезпечує безперебійну роботу підприємств, високий рівень продуктивності працівників та збереження конфіденційної інформації. Однак із зростанням складності та масштабу мережевих структур зростає й вразливість загроз, серед яких особливу небезпеку становлять аномальні події, що можуть бути спричинені кібератаками, витоком даних, технічними збоїв або іншими потенційно шкідливими ситуаціями. Актуальність дослідження полягає в тому, що існуючі методи виявлення аномалій нерідко демонструють недостатню точність та оперативність у реальних умовах експлуатації. Тому необхідно розробляти та вдосконалювати методи, які здатні ефективно адаптуватися до мінливих умов сучасних корпоративних мереж і надавати точні результати у режимі реального часу. **Об'єкт дослідження** є мережевий трафік корпоративної інформаційної системи, який аналізується з метою виявлення аномальної поведінки, що потенційно свідчить про кібератаки, порушення політик безпеки або внутрішні загрози. **Мета статті** розробка, реалізація та експериментальне обґрунтування методу виявлення аномалій у корпоративній мережі на основі моделі автоенкодера, який дозволяє ефективно ідентифікувати відхилення від нормального мережевого трафіку без потреби в попередньо мічених даних. **Результати дослідження.** У процесі моделювання було згенеровано дані, які імітують типову та аномальну активність у мережі. Після навчання нейронної мережі на нормальних зразках спостерігалося чітке розділення класів за критерієм помилки реконструкції. Гістограми та boxplot включають на те, що нормальні зразки характеризуються низькими і стабільними значеннями середньоквадратичної помилки, тоді як аномалії – навпаки виражені відхилення. ROC-крива з AUC = 1.00 підтверджує, що модель здатна безпомилково розрізняти обидві категорії. Матриця плутанини показала практично ідеальне співпадіння між передбаченими та фактичними мітками, що підтверджує високу точність і чутливість моделі. Тренувальна крива свідчить про стабільне навчання без перенавчання, а обраний аномальний поріг дозволив знизити рівень хибнопозитивних спрацювань до мінімуму. Загалом, результати підтверджують, що використання автоенкодера для побудови поведінкової моделі трафіку дозволяє точно та надійно виявляти аномалії в корпоративній мережі, навіть за відсутності мічених прикладів. Запропонований підхід може бути інтегрований у системи моніторингу безпеки для виявлення потенційних активностей у режимі реального часу. **Висновки.** На основі моделі автоенкодера було реалізовано підхід, який дозволяє будувати поведінковий профіль нормального мережевого трафіку та виявляти потенційні відхилення без потреби у мічених даних. Проведені експерименти продемонстрували високу точність класифікації, чітке розділення нормальних і аномальних зразків та помилкову реконструкцію, а також стабільність навчання моделі. Отримані результати підтверджують доцільність використання глибокого навчання для автоматизованого моніторингу мережевої безпеки, а запропонований метод може бути успішно застосований у реальних умовах корпоративної IT-інфраструктури для виявлення інцидентів безпеки в режимі реального часу.

Ключові слова: корпоративна мережа, виявлення аномалій, корпоративна мережа, мережевий трафік, автоенкодер, машинне навчання, кібербезпека, реконструктивна похибка, навчання без вчителя, нейронна мережа, профілювання поведінки, інформаційна безпека.

Вступ

Постановка проблеми. Сучасні корпоративні мережі є невід'ємною складовою цифрової інфраструктури будь-якої компанії, незалежно від її розміру та сфери діяльності. Вони забезпечують ключові процеси, такі як передача інформації, підтримка операційних та управлінських систем, комунікація між підрозділами, а також зовнішню взаємодію з клієнтами й партнерами. В умовах постійного зростання обсягу переданих і оброблюваних даних, а також ускладнення топології мереж, зростають ризики виникнення небезпечних ситуацій, зокрема

...ких, несанкціонованих доступом), так і з різного роду технічними проблемами (відмовами обладнання, помилками програмного забезпечення, неефективністю налаштувань чи конфігурацій). Особливість таких аномалій полягає у тому, що їх важко своєчасно розпізнати за допомогою стандартних інструментів моніторингу й аналізу трафіку, оскільки вони часто не вписуються у попередньо задані моделі та правила, і можуть залишатися прихованими у великих обсягах даних.

Загрози, що виникають у результаті аномальних подій, мають серйозні наслідки, серед яких особливо небезпечними є витоки конфіденційних да-

...вості адаптації до змінних умов експлуатації, а також недостатня ефективність при роботі з великими обсягами даних у реальному часі. Ці недоліки стимулюють пошук нових підходів і методів, що дозволять ефективно виявляти та класифікувати аномалії в корпоративних мережах.

Аналіз останніх досліджень і публікацій.

У сучасному науково-практичному середовищі проблема виявлення аномалій у корпоративних мережах стала однією з найбільш обговорюваних у сфері кібербезпеки та інтелектуального аналізу даних. Розвиток мережевої інфраструктури, зростання обсягів трафіку, поширення хмарних сервісів і віддаленої роботи значно ускладнили завдання виявлення загроз у реальному часі. Саме тому останніх років стали свідками стрімкого зростання інтересу до методів виявлення аномалій на основі штучного інтелекту, машинного навчання, гібридних та контекстно-орієнтованих підходів.

Базові методи, наприклад, сигнатурний аналіз, добре зарекомендували себе у виявленні відомих загроз, однак вони нездатні розпізнавати нові або обфусковані атаки. У відповідь на це, з'явилися численні публікації, в яких обґрунтовується перехід до статистичних методів та автоматичної побудови профілю нормальної поведінки.

Стаття [1] присвячена огляду сучасних методів навчання без вчителя для виявлення аномалій у часових рядах. У ній розглядаються дослідження в галузі збору та аналізу даних, які привели до накопичення великої кількості послідовних спостережень. Стаття підкреслює важливість відображення знань із таких даних, зокрема виявлення викидів або аномалій, що можуть сигналізувати про помилки, порушення чи значущі події.

В роботі [2] автори проводять порівняння класичних алгоритмів машинного навчання і роблять висновок, що гібридні методи (машинне навчання + фільтрація ознак) перевищують по точності базові евристики.

Особливу увагу привертають методи глибокого навчання, зокрема автоенкодери, LSTM-модель та рекурентні нейронні мережі. Вони дозволяють моделювати складну часову структуру мережевого трафіку та виявляти аномалії без необхідності міток. У дослідженні [3] описано Kitsune – систему, що використовує ансамбль автоенкодерів для виявлення аномалій на основі flow-даних. Вона працює в реальному часі з низькою обчислювальною складністю, що робить її придатною для корпоративних мереж.

Стаття [4] присвячена огляду сучасних підходів

...підходів, які використовуються для виявлення таких аномалій. Зокрема, розглядаються методи машинного навчання, глибокого навчання, математичного моделювання.

Також залишається проблема оцінки моделей виявлення аномалій в роботі [5] автори критикують популярні набори даних за низьку репрезентативність і пропонують створення нових сценаріїв на основі реального корпоративного трафіку.

Метою роботи є розробка удосконаленого методу виявлення аномалій у корпоративних мережах, який базується на сучасних підходах у сфері навчання та штучного інтелекту. Запропонований метод має забезпечити високу точність, швидкодію, адаптивність до динамічно змінних умов і низький рівень хибних спрацювань, що є критично важливо для оперативного прийняття рішень та забезпечення надійності функціонування інформаційних систем підприємств

Основний матеріал

Проблема виявлення аномалій у корпоративних мережах уже давно є предметом активного дослідження в галузі інформаційної безпеки. У різних підходах до її вирішення застосовуються як класичні методи аналізу даних, так і сучасні алгоритми машинного навчання та штучного інтелекту [6]. Основна ідея таких методів полягає у тому, щоб вчасно виявити нетипову поведінку в мережевому трафіку, яка потенційно може свідчити про загрозу, або про критичні відхилення в роботі системи. У цьому контексті важливою є здатність системи відокремлювати нормальну активність від аномальної, навіть за відсутності явної сигнатури загрози.

Традиційні методи, зокрема сигнатурний аналіз [7], орієнтовані на виявлення вже відомих шаблонів атак, що зберігаються у базі даних сигнатур. Вони ефективні у виявленні відомих загроз, однак повністю безсилі перед новими, раніше невідомими формами атак, особливо так званими атаками нульового дня. Крім того, сигнатурні системи вимагають постійного оновлення баз даних і часто створюють великі навантаження на систему безпеки через значну кількість перевірок.

У протилежу сигнатурним методам, аномаліє-орієнтовані методи ґрунтуються на побудові моделей «нормальної» поведінки мережі, з якими порівнюється поточна активність. Порушення встановленого шаблону вважається підозрілим і може бути маркером аномалії. Такі методи вимагають етапу навчання або збору статистики, проте вони здатні виявляти

Глоба Є. Ю., Смірнов В. Р., Нараєвський М. С. Метод виявлення аномалій в корпоративній мережі. Системи управління, навігації та зв`язку, вип3. Полтава, 2025. С. 154-158.

14

ВИСНОВКИ

У результаті виконаного дослідження було розроблено та експериментально апробовано новий підхід до виявлення аномалій у корпоративному мережевому середовищі, що базується на гібридній нейромережевій архітектурі з використанням автоенкодера, доповненого компонентами LSTM. Такий підхід дозволяє ефективно моделювати нормальну поведінку мережевого трафіку, виявляючи відхилення на основі похибки реконструкції без необхідності маркованих даних. Ретельний аналіз існуючих методів від сигнатурного аналізу до глибоких моделей машинного навчання дозволив обґрунтувати вибір саме гібридного підходу як найбільш придатного для роботи в умовах динамічного й неоднорідного середовища сучасних інформаційних систем.

Було створено архітектуру програмного прототипу, що включає модулі збору, обробки, аналізу, моніторингу та адаптації, реалізовані мовою Python із використанням сучасних бібліотек, зокрема Scapy, Pandas, Scikit-learn, TensorFlow та Loguru. Особливу увагу було приділено аспектам адаптації до змін у мережі, зокрема через механізми оновлення профілю поведінки на основі довірених даних та використання потокової обробки трафіку. Експериментальне тестування прототипу, проведене на стилізованому наборі даних, продемонструвало високу точність моделі в задачах розпізнавання аномальної активності. Графічні результати, зокрема ROC-крива, гістограми та матриця плутанини, засвідчили, що система має відмінну здатність до відокремлення нормальної та підозрілої поведінки навіть за умов слабкої апріорної інформації та відсутності маркування. Висока площа під ROC-кривою (AUC = 1.00), низька частка хибнопозитивних спрацювань та стабільність навчання свідчать про добру узагальнювальну здатність та надійність розробленої моделі.

Таким чином, розроблений метод підтвердив свою ефективність як у теоретичному обґрунтуванні, так і в практичній реалізації. Він поєднує точність виявлення, адаптивність до нових умов, низьку залежність від маркованих даних і здатність до роботи в реальному часі. Це робить його перспективним рішенням для впровадження в інфраструктуру кіберзахисту корпоративних мереж, де актуальною є потреба у безперервному моніторингу, ранньому виявленні загроз і мінімізації ризиків безпеки. Подальші дослідження можуть бути зосереджені на вдосконаленні методів самонавчання, інтеграції з SIEM-платформами, а також на масштабуванні архітектури для роботи з великими розподіленими системами в реальному середовищі.

ДОДАТОК Б

Програмний код

Б.1 Встановлення бібліотек

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_curve, auc, confusion_matrix,
ConfusionMatrixDisplay
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.optimizers import Adam
import matplotlib.pyplot as plt
import seaborn as sns
```

Б.2 Генерація даних

```
def generate_data():
    np.random.seed(42)
    n_normal = 1000
    n_anomalous = 50
    normal_data = {
        "duration": np.random.exponential(2.0, n_normal),
        "src_bytes": np.random.normal(300, 50, n_normal),
        "dst_bytes": np.random.normal(250, 60, n_normal),
        "count": np.random.randint(1, 20, n_normal),
        "srv_count": np.random.randint(1, 15, n_normal),
        "same_srv_rate": np.random.uniform(0.5, 1.0, n_normal),
        "diff_srv_rate": np.random.uniform(0.0, 0.2, n_normal),
        "dst_host_count": np.random.randint(10, 100, n_normal),
        "dst_host_srv_count": np.random.randint(5, 100,
n_normal),
        "dst_host_same_srv_rate": np.random.uniform(0.6, 1.0,
n_normal),
        "label": [0]*n_normal
    }
    anomalous_data = {
        "duration": np.random.exponential(10.0, n_anomalous),
        "src_bytes": np.random.normal(1000, 200, n_anomalous),
        "dst_bytes": np.random.normal(30, 15, n_anomalous),
        "count": np.random.randint(30, 100, n_anomalous),
        "srv_count": np.random.randint(20, 60, n_anomalous),
        "same_srv_rate": np.random.uniform(0.0, 0.3,
n_anomalous),
```

```
        "diff_srv_rate": np.random.uniform(0.7, 1.0,
n_anomalous),
        "dst_host_count": np.random.randint(200, 255,
n_anomalous),
        "dst_host_srv_count": np.random.randint(100, 255,
n_anomalous),
        "dst_host_same_srv_rate": np.random.uniform(0.0, 0.4,
n_anomalous),
        "label": [1]*n_anomalous
    }
    df = pd.concat([pd.DataFrame(normal_data),
pd.DataFrame(anomalous_data)], ignore_index=True)
    return df.sample(frac=1.0,
random_state=42).reset_index(drop=True)

# 1. Дані
df = generate_data()
X = df.drop(columns=["label"]).values
y = df["label"].values
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.3, random_state=42)
X_train_norm = X_train[y_train == 0]

# 2. Автоенкодер
input_dim = X_train.shape[1]
input_layer = Input(shape=(input_dim,))
encoded = Dense(8, activation='relu')(input_layer)
encoded = Dense(4, activation='relu')(encoded)
decoded = Dense(8, activation='relu')(encoded)
output_layer = Dense(input_dim, activation='linear')(decoded)
autoencoder = Model(inputs=input_layer, outputs=output_layer)
autoencoder.compile(optimizer=Adam(learning_rate=0.001),
loss='mse')
history = autoencoder.fit(X_train_norm, X_train_norm, epochs=50,
batch_size=32,
                          shuffle=True, validation_split=0.1,
verbose=1)
```

Б.3 Виявлення аномалій та візуалізація

```
X_test_pred = autoencoder.predict(X_test)
mse = np.mean(np.square(X_test - X_test_pred), axis=1)
threshold = np.percentile(mse[y_test == 0], 95)
predictions = (mse > threshold).astype(int)
fpr, tpr, _ = roc_curve(y_test, mse)
roc_auc = auc(fpr, tpr)
error_df = pd.DataFrame({'MSE': mse, 'label': y_test})

plt.figure(figsize=(18, 12))
```

```
plt.subplot(2, 3, 1)
plt.plot(fpr, tpr, color='blue', lw=2, label=f'AUC =
{roc_auc:.2f}')
plt.plot([0, 1], [0, 1], color='gray', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC-крива')
plt.legend()
plt.grid(True)

plt.subplot(2, 3, 2)
sns.boxplot(x='label', y='MSE', data=error_df)
plt.xticks([0, 1], ['Нормальні', 'Аномалії'])
plt.title("Boxplot похибки")
plt.grid(True)

plt.subplot(2, 3, 3)
plt.scatter(range(len(mse)), mse, c=y_test, cmap='coolwarm',
edgecolors='k')
plt.axhline(y=threshold, color='red', linestyle='--',
label='Поріг')
plt.title("Похибка реконструкції")
plt.xlabel("Номер зразка")
plt.ylabel("MSE")
plt.legend()
plt.grid(True)

plt.subplot(2, 3, 4)
cm = confusion_matrix(y_test, predictions)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=["Норма", "Аномалія"])
disp.plot(ax=plt.gca(), cmap='Blues', values_format='d')
plt.title("Матриця плутанини")
plt.grid(False)

plt.subplot(2, 3, 5)
plt.hist(mse[y_test == 0], bins=50, alpha=0.6,
label="Нормальні")
plt.hist(mse[y_test == 1], bins=50, alpha=0.6,
label="Аномальні")
plt.axvline(threshold, color='red', linestyle='--',
label=f'Поріг: {threshold:.4f}')
plt.title("Гістограма похибок реконструкції")
plt.xlabel("MSE")
plt.ylabel("Кількість")
plt.legend()
plt.grid(True)

plt.subplot(2, 3, 6)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title('Тренувальна крива')
```

```
plt.xlabel('Епоха')
plt.ylabel('MSE Loss')
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.show()
```

Б.4 Додатковий аналіз результатів

```
# --- PCA-візуалізація простору ознак ---
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_test)
plt.figure(figsize=(6, 5))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=predictions,
cmap='coolwarm', alpha=0.6, edgecolors='k')
plt.title("PCA-візуалізація простору зразків")
plt.xlabel("Головна компонента 1")
plt.ylabel("Головна компонента 2")
plt.grid(True)
plt.show()
# --- Precision-Recall крива ---
from sklearn.metrics import precision_recall_curve,
average_precision_score

precision, recall, _ = precision_recall_curve(y_test, mse)
avg_precision = average_precision_score(y_test, mse)

plt.figure(figsize=(6, 5))
plt.plot(recall, precision, marker='.')
plt.title(f'PR-крива (AP = {avg_precision:.2f})')
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.grid(True)
plt.show()
# --- Теплова карта кореляцій ознак ---
plt.figure(figsize=(10, 8))
sns.heatmap(pd.DataFrame(X, columns=df.columns[:-1]).corr(),
annot=True, fmt=".2f", cmap="coolwarm")
plt.title("Теплова карта кореляцій ознак")
plt.show()
```