УДК 004.852:519.6 DOI: 10.18372/2073-4751.82.20369

Мельниченко П.І.,
orcid.org/0000-0003-3746-1241,
e-mail: polina.melnychenko@npp.kai.edu.ua

РОЗРОБКА МОДЕЛІ ПРОГНОЗУВАННЯ ВРАЗЛИВОСТЕЙ ВЕБ ЗАСТОСУНКІВ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Державний університет «Київський авіаційний інститут»

Вступ

Сучасні системи захисту інформації вимагають автоматичного аналізу великих масивів даних про інциденти. Моделі машинного навчання дозволяють не тільки аналізувати, але й прогнозувати потенційні загрози. Дерева рішень та ентропія як міра невизначеності грають ключову роль в задачах класифікації загроз, оцінки пріоритетів та виявлення закономірностей у вразливостях.

Аналіз останніх досліджень та публікацій

Аналізуючи наукові матеріали останніх років, можна побачити активне використання методів інтелектуального аналізу даних у сфері інформаційної безпеки. Наприклад, у роботі [1] авторами запропоновано інформаційну технологію оцінки кількості, невизначеності та достатності інформації за її ентропією в системах підтримки прийняття рішень. У роботі [2] автори проаналізували інтеграцію інформаційно-ентропійних методів в системи управління безпекою. У [3] авторами було досліджено алгоритми для побудови та оптимізації дерев рішень. Робота [4] присвячена створенню системи виявлення кіберзагроз на основі аналізу даних мережевого трафіку вебресурсів. У досліджених джерелах не розглядається проблема прогнозування вразливостей методами машинного навчання, що обмежує можливості своєчасного реагування на потенційні загрози. Тому актуальною є розробка моделі, яка дозволяє здійснювати автоматизований аналіз журналів вразливостей, та передбачати найбільш ймовірні типи вразливостей відповідно до класифікації *OWASP*.

Основна частина

Ентропія як міра невизначеності є важливим інструментом для оцінки розподілу ознак у наборах даних. Розрахунок ентропії дозволяє визначити наскільки кожна ознака (рівень ризику) сприяє зменшенню невизначеності у прогнозі майбутніх вразливостей. Розрахунок ентропії виконується для відбору ознак перед навчанням моделі щоб виключити не релевантні та надлишкові ознаки.

Визначення ентропії в контексті теорії інформації було вперше застосовано Клодом Шенноном в 1948 році у роботі «*A Mathematical Theory of Communication*» [5]. Ентропія розраховує середню кількість інформації, необхідну для опису випадкової величини, тобто міру невизначеності. Чим вище ентропія, тим більш різноманітні та непередбачувані значення випадкової величини.

Для дискретної випадкової величини X з можливими значеннями $x_1, x_2, \dots, x_n$ та відповідними ймовірностями $p_1, p_2, \dots, p_n$ ентропія визначається наступним чином [5]:

$$H(X) = -\sum_{i=1}^{n} p_i \cdot log_2(p_i) \qquad (1)$$

де $H(X)$ – ентропія (в бітах); p_i – ймовірність настання події i; логарифм з основою 2 використовується для визначення в бітах.

Властивості ентропії:

- ентропія дорівнює 0, якщо результат повністю передбачуваний;
- ентропія дорівнює 1 коли існує максимальна невизначенність;
- використовується для визначення оптимальних поділів в алгоритмах побудови дерев рішень: обирається та

ознака, при розбитті якої ентропія є найменшою.

Для аналізу було використано датасет вразливостей за 2024 рік. Даний датасет містить наступні рівні загроз: *Low*, *Medium*, *High*, *Critical*. Для рівня загроз вразливостей ентропія дозволяє зрозуміти наскільки рівномірно розподілені вразливості та наскільки складно передбачити рівень загроз.

Використовуючи формулу ентропії Шеннона (1) розраховано ентропію для кожного рівня загроз.

Рівень Low:

$$p_{Low} = \frac{257}{648} \approx 0{,}3966.$$

$$log_2(0{,}3966) \approx -1{,}3342.$$

$$-p \cdot log_2(p) = -(0{,}3966 \cdot -1{,}3342) \approx 0{,}5291.$$

Рівень Medium:

$$p_{Medium} = \frac{181}{648} \approx 0{,}2793.$$

$$log_2(0{,}2793) \approx -1{,}837.$$

$$-p \cdot log_2(p) = -(0{,}2793 \cdot (-1{,}837)) \approx 0{,}5130.$$

Рівень High:

$$p_{High} = \frac{162}{648} = 0{,}25.$$

$$log_2(0{,}25) = -2.$$

$$-p \cdot log_2(p) = -\big(0{,}25 \cdot (-2)\big) = 0{,}5.$$

Рівень Critical:

$$p_{Critical} = \frac{48}{648} \approx 0{,}0741.$$

$$log_2(0{,}0741) \approx -3{,}755.$$

$$-p \cdot log_2(p) = -(0{,}0741 \cdot (-3{,}755) \approx 0{,}2782.$$

Розрахунок ентропії:

$$H = 0{,}5130 + 0{,}5291 + 0{,}5 + 0{,}2782 = 1{,}8203 \text{ біт.}$$

Отриманий результат (1,8203 біт) говорить про помірну невизначеність, тобто вибірка складається як з поширених, так і з не поширених категорій. Згідно з теорією інформації, максимальна ентропія досягається у випадку рівноймовірного розподілу всіх можливих подій (тобто коли всі рівні загроз представлені однаково), і свідчить про максимальну невизначеність. В даному випадку значення ентропії менше максимально можливого, що означає переважання деяких рівнів загроз. Це свідчить про те, що для подальшого аналізу необхідно побудувати дерево рішень, яке буде зменшувати невизначеність на основі додаткових ознак.

Дерево рішень – це один з базових алгоритмів машинного навчання, який використовується як для задач класифікації, так і для задач регресії. Даний метод базується на покроковому розбитті початкових даних на підмножини з урахуванням значень ознак, що дозволяє створювати інтерпретовану модель, схожу з логічною схемою прийняття рішень [6].

За своєю структурою дерева рішень складаються з:

- корінь – початок вузла, що охоплює усю навчальну вибірку;
- внутрішні вузли – містять вимоги до розбиття за ознаками;
- листки – кінцеві вузли, в яких модель дає підсумковий прогноз (клас чи значення).

На кожному етапі дерево обирає ознаку, яка дозволяє найкращим чином розділити дані, тобто мінімізувати невизначеність у класах цільової змінної. Найбільш частим критерієм вибору є ентропія Шеннона.

Після розбиття за ознакою A, зростання інформації розраховується за формулою:

$$IG(S, A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v) \quad (2)$$

де S_v – підмножина об'єктів, які мають значення v ознаки A.

Ціль алгоритму – обрати таку ознаку та поріг, при яких приріст інформації є максимальним.

В задачах захисту інформації дерева рішень можуть використовуватись для класифікації вразливостей за типом та рівнями небезпеки, прогнозування ймовірностей появи нових вразливостей, аналізу поведінки зловмисників (наприклад, в *SIEM*-системах), підтримки прийняття рішень в системах реагування на інциденти.

Переваги дерева рішень:

- інтерпретованість – кожне рішення можна відстежити покроково;
- швидке навчання та прогнозування – підходить для систем, що працюють у реальному часі;
- можливість візуалізації – зручність для аудиту та аналізу.

До недоліків дерева рішень відносяться:

- ймовірність перенавчання на шумних даних (необхідно обмежувати глибину дерева);
- чутливість до нерівномірного розподілу класів;
- необхідність попередньої обробки та нормалізації ознак.

Для побудови дерева рішень було необхідно попередньо дослідити датасет вразливостей та нормалізувати дані. Датасет складається з наступних колонок:

- *vulnerabilities* – тип вразливості;
- *corrected* – дані про виправлення вразливості;
- *date_of_correction* – дата виправлення;
- *date_of_discovery* – дата виявлення;
- *level* – рівень загрози;
- *reward* – нагорода.

Використовуючи бібліотеки *pandas* та *matplotlib* мови програмування *Python* було визначено числові характеристики та побудовано графік динаміки появи вразливостей за 2024 рік, що зображені на рис. 1 та рис. 2.

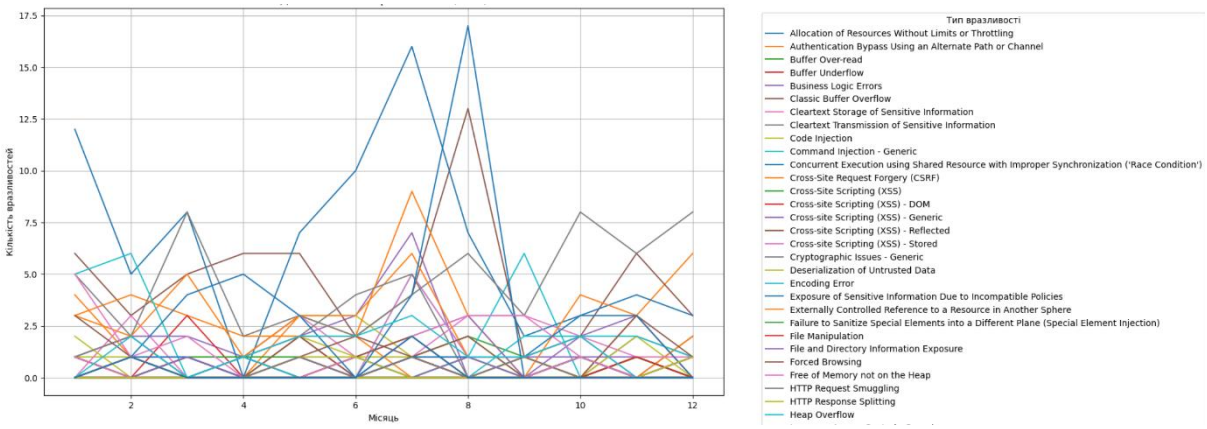

Рис. 1. Динаміка появи вразливостей по місяцям

Загальна кількість унікальних вразливостей, що зустрічаються у датасеті, дорівнює 91. Для побудови дерева рішень така велика кількість класів буде давати малу точність прогнозованої моделі.

Для вирішення проблеми надмірної кількості класів необхідно ввести додаткову змінну, щоб зменшити кількість класів. Авторським колективом запропоновано розділити вразливості за категоріями *OWASP Top* 10 2021. Це дає змогу зменшити кількість класів до десяти, та отримати більшу точність прогнозованих даних.

```
Improper Access Control - Generic: 77
Cross-site Scripting (XSS) - Reflected: 57
Information Disclosure: 57
Insecure Direct Object Reference (IDOR): 31
Improper Authentication - Generic: 31
Privilege Escalation: 23
Business Logic Errors: 23
Uncontrolled Resource Consumption: 22
Cross-Site Request Forgery (CSRF): 20
Misconfiguration: 20
Path Traversal: 18
Cross-site Scripting (XSS) - Stored: 18
Server-Side Request Forgery (SSRF): 17
Violation of Secure Design Principles: 14
Code Injection: 14
Command Injection - Generic: 13
SQL Injection: 11
Open Redirect: 9
Improper Input Validation: 9
Cross-site Scripting (XSS) - DOM: 8
Cross-site Scripting (XSS) - Generic: 7
Cleartext Transmission of Sensitive Information: 7
Deserialization of Untrusted Data: 7
```

Рис. 2. Числова характеристика вразливостей

OWASP Top 10 – це методологія оцінки вразливостей, які найчастіше зустрічаються у вебзастосунках [7]. Вона

складається з наступних класифікацій, представлених у таблиці 1.

Першим кроком необхідно створити нову колонку, в якій вказана категорія *OWASP* до кожної вразливості. Для того, щоб знайти відповідності було розроблено словник ключових слів до кожної категорії (Рис. 3).

Таблиця 1. Перелік категорій *OWASP Top* 10 2021

Назва	Опис
*A*01:2021 *Broken Access Control*	Порушення контролю доступу - вразливість, при якій користувачі отримують доступ до ресурсів чи функцій, на які в них не має дозволу
*A*02:2021 *Cryptographic Failures*	Збій криптографії – це помилки в реалізації та використанні криптографічних методів, що призводять до компрометації даних
*A*03:2021 *Injection*	Впровадження коду через вразливості в обробці вхідних даних, наприклад *XSS* або *SQL*-ін'єкції
*A*04:2021 *Insecure Design*	Небезпечний дизайн – недоліки в архітектурі та проєктуванні системи, що створюють потенційні загрози безпеці
*A*05:2021 *Security Misconfiguration*	Неправильна конфігурація – помилки в налаштуваннях системи, що можуть відкрити доступ зловмисникам
*A*06:2021 *Vulnerable and Outdated Components*	Використання вразливих та застарілих компонентів та бібліотек, що мають велику кількість вразливостей
*A*07:2021 *Identification and Authentication Failures*	Помилки ідентифікації та аутентифікації – вразливості в процесах перевірки особистості користувача, що дозволяють обходити захист
*A*08:2021 *Software and Data Integrity Failures*	Порушення цілісності програмного забезпечення та даних
*A*09:2021 *Security Logging and Monitoring Failures*	Збій реєстрації та моніторингу безпеки – недостатній контроль та запис подій безпеки, що ускладнюють виявлення атак
*A*10:2021 *Server-Side Request Forgery*	Підробка запитів на стороні серверу – атака, при якій зловмисник змушує сервер виконувати небажані запити до інших ресурсів

```
# Словник з категоріями OWASP Top 10 2021 та ключовими словами для пошуку
owasp_categories = {
    "A01:2021 Broken Access Control": ["Access Control", "Privilege", "Authorization", "IDOR", "Access"],
    "A02:2021 Cryptographic Failures": ["Cryptographic", "Encryption", "Key", "Certificate", "Cryptography", "Cleartext"],
    "A03:2021 Injection": ["Injection", "SQL", "Command Injection", "Code Injection", "OS Command", "Deserialization"],
    "A04:2021 Insecure Design": ["Design", "Logic", "Business Logic", "Violation of Secure Design"],
    "A05:2021 Security Misconfiguration": ["Misconfiguration", "Configuration", "Improper Validation", "Improper Handling"],
    "A06:2021 Vulnerable and Outdated Components": ["Components", "Known Vulnerabilities", "Outdated", "Use After Free"],
    "A07:2021 Identification and Authentication Failures": ["Authentication", "Credential", "Login", "Authorization Bypass",
    "A08:2021 Software and Data Integrity Failures": ["Integrity", "Verification", "Tampering", "Deserialization"],
    "A09:2021 Security Logging and Monitoring Failures": ["Logging", "Monitoring", "Session Expiration", "Race Condition"],
    "A10:2021 Server-Side Request Forgery (SSRF)": ["Server-Side Request Forgery", "SSRF", "Request Forgery"]
```

Рис. 3. Словник ключових слів категорій *OWASP Top* 10

Використовуючи цикл *for* було виконано перевірку , чи міститься одне або декілька ключових слів у вразливості. Якщо виконується умова, то вразливість потрапляє до даної категорії. Якщо вразливість не відноситься до жодної з категорій, то позначається як «*Not categorized*». В результаті було отримано новий датасет з колонкою *Owasp_Top_Ten* (Рис. 4).

Даний датасет містить наступні унікальні значення категорій *OWASP*, зображені на числовій гістограмі (Рис. 5).

	Vulnerabilities	Corrected	Date_of_Correction	Date_of_Discovery	Level	Reward	month	Owasp_top_ten	Owasp_Top_Ten
0	Information Disclosure	True	2024-12-30T14:40:35.426Z	2024-12-30 14:40:35.177000+00:00	Medium	NaN	12	A03:2021 Injection	A03:2021 Injection
1	Information Disclosure	True	2024-12-30T14:31:30.023Z	2024-12-30 14:31:29.747000+00:00	Low	NaN	12	A03:2021 Injection	A03:2021 Injection
2	Improper Input Validation	True	2024-12-28T18:09:56.940Z	2024-12-28 18:09:56.763000+00:00	Medium	500.0	12	A03:2021 Injection	A03:2021 Injection
3	Cross-site Scripting (XSS) - Reflected	True	2024-12-28T08:57:12.674Z	2024-12-28 08:57:12.458000+00:00	Medium	100.0	12	A03:2021 Injection	A03:2021 Injection
4	Incorrect Privilege Assignment	True	2024-12-27T14:40:27.431Z	2024-12-27 14:40:27.234000+00:00	Medium	NaN	12	A01:2021 Broken Access Control	A01:2021 Broken Access Control
5	Information Disclosure	True	2024-12-27T12:42:23.327Z	2024-12-27 12:42:23.462000+00:00	Medium	NaN	12	A03:2021 Injection	A03:2021 Injection

Рис. 4. Оновлений датасет вразливостей

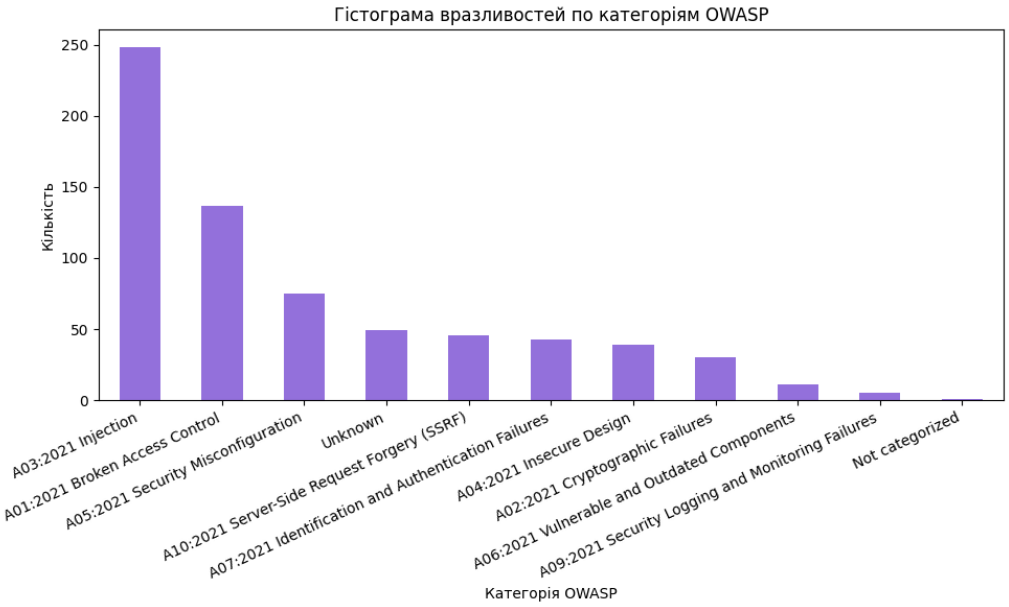

Рис. 5. Гістограма категорій *OWASP Top* 10

Наступним кроком було побудувано дерево рішень використовуючи колонку *Owasp_Top_Ten* в якості цільової змінної для класифікації. Для цього було необхідно нормалізувати датасет, тобто привести дані у необхідний числовий вигляд, що представлено у лістингу 1.

Лістинг 1

```
# Цільова змінна
target = "Owasp_Top_Ten"

# Видалення стовбців, що не мають ознак
# Обираємо лише числові та категоріальні дані
X = df.drop(columns=[target])

# Перетворення категоріальних змінних у числові
X = pd.get_dummies(X)

# Цільова змінна (категорії OWASP)
y = df[target]

# Умова: якщо цільова змінна string, то кодувати у числовому вигляді
label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)
```

Дані необхідно розділити на навчальну та тестову вибірки. Навчальна вибірка – це дані, на яких модель буде навчатись. Модель оцінює вхідні ознаки *X-train* та відповідні їм правильні відповіді *y-train* щоб знайти закономірності. Тестова вибірка – це нові дані, які ще не були опрацьовані моделлю, на них буде оцінюватись точність моделі. Моделі надаються вхідні ознаки *X_test* та відбувається порівняння передбачуваних відповідей з реальними *y_test*.

Якщо не розділювати дані, модель може лише вивчити навчальну вибірку та погано працювати на нових даних. Оцінка на тестових даних дозволяє зрозуміти, як модель буде себе поводити в реальних умовах. Для того, щоб розділити дані, використовується параметр *test_size*.

Лістинг 2

```
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)
```

В даному випадку модель було розділено на: 80 % - навчальна вибірка, 20 % - тестова вибірка.

Далі відбувається навчання дерева рішень. Точність моделі залежить від того, яку глибину дерева буде вказано. Для визначення найкращої глибини дерева рішень було виконано підбір за допомогою функції *GridSearchCV* [8].

Лістинг 3

```
from sklearn.model_selection import GridSearchCV
```

```
    params = {'max_depth': range(3, 20)}
    grid = GridSearchCV(DecisionTreeClassifier(random_state=42), params, cv=5)
    grid.fit(X_train, y_train)

    print(f"Найкраща глибина: {grid.best_params_['max_depth']}")
    print(f"Найкраща точність (кросс-валідація): {grid.best_score_:.2f}")
```

Методом підбору було визначено, що найкраща глибина дерева рішень 19, при якій досягається точність у 82 %. Даний результат можна вважати оптимальним та достатнім для подальшого аналізу та практичного використання, оскільки занадто висока точність може говорити про те, що модель перенавчилась.

Наступним кроком було створено дерево рішень.

Лістинг 4

```
    clf = DecisionTreeClassifier(max_depth=19, random_state=42)
    clf.fit(X_train, y_train)
```

DecisionTreeClassifier – це алгоритм класифікації, що базується на побудові дерева, яке ділить дані на підмножини на основі найкращих ознак, мінімізуючи помилку класифікації.

random_state=42 – це параметр, що задає початкове значення генератора випадкових чисел (при повторному виконанні з тими ж даними та параметрами дерево буде побудовано так само).

Метод *.fit()* – навчає модель на тренувальній вибірці.

Часткова візуалізація перших чотирьох рівнів дерева рішень представлена на рис. 6.

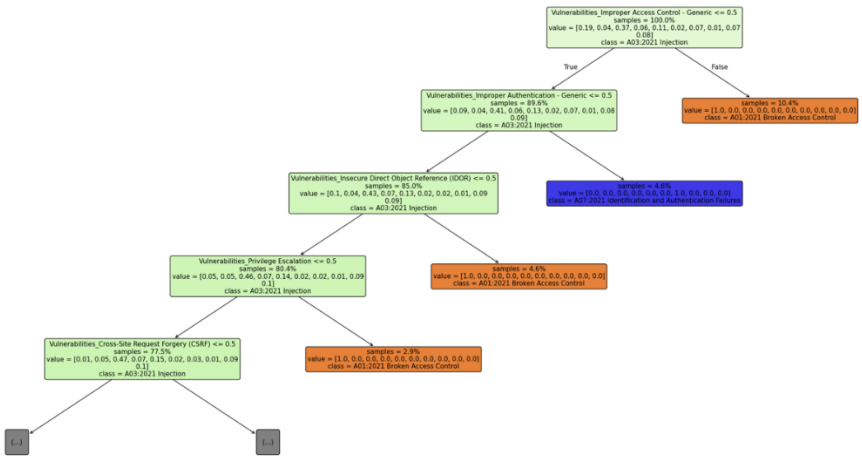

Рис. 6. Візуалізація чотирьох рівнів дерева рішень

Загальна візуалізація дерева рішень представлена на рис. 7. У верхньому рядку кожного блоку вказана умова, за якою виконується поділ вузла (*Vulnerabilities_Improper Access Control - Generic* <= 0.5), це означає, що вузол ділиться на два піддерева.

Індекс Джині (в першому вузлі *gini* = 0.797) – це критерій неоднорідності вибірки [9]. Він приймає значення від 0 (повністю чистий вузол в якому всі об'єкти відносяться до однієї категорії) до 1 (максимальна різноманітність). Чим нижче значення індексу, тим достовірніше визначено клас об'єктів у вузлі.

Параметр *samples* показує яка кількість записів потрапила у даний вузол.

Параметр *value* – це масив, в якому кожне число показує, яка кількість об'єктів кожного класу (категорії *OWASP*) знаходиться у вузлі. Кожен індекс масиву відповідає класу вразливостей *OWASP*.

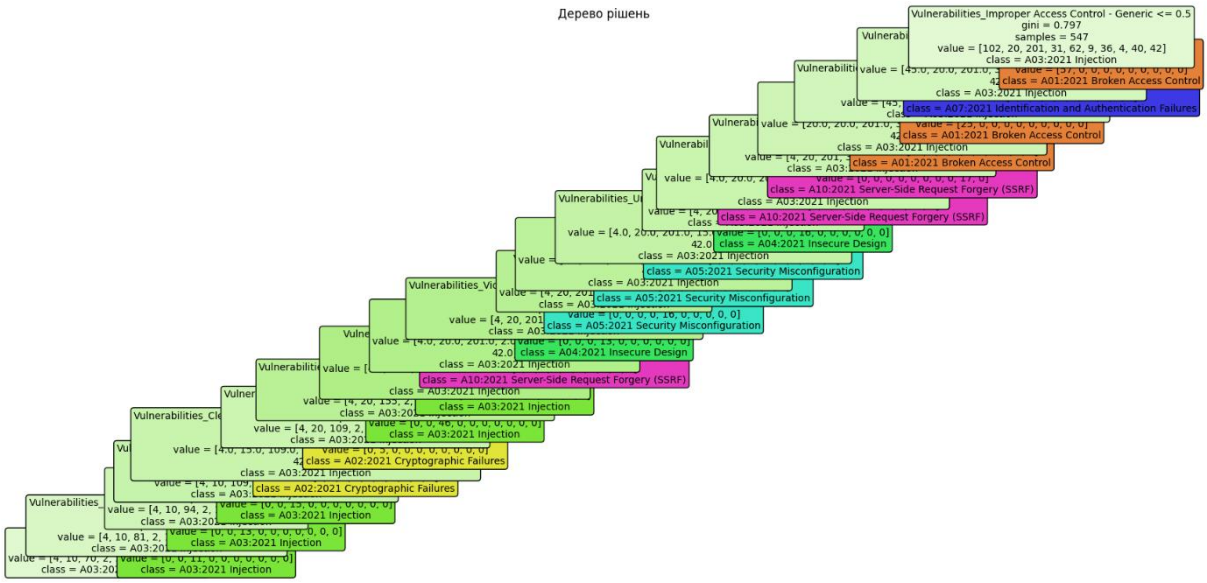

Рис. 7. Візуалізація дерева рішень

Розрахувавши частоту передбачених категорій було отримано наступні результати, що зображено на рис. 8.

```
Частота передбачених OWASP категорій:
A07:2021 Identification and Authentication Failures: 6
A03:2021 Injection: 74
A01:2021 Broken Access Control: 33
A10:2021 Server-Side Request Forgery (SSRF): 6
A04:2021 Insecure Design: 8
A05:2021 Security Misconfiguration: 8
A02:2021 Cryptographic Failures: 2

Найчастіша категорія: A03:2021 Injection (74 рази)
```

Рис. 8. Розрахунок частоти категорій вразливостей

Отриманий результат демонструє, що модель дерева рішень найчастіше передбачає категорію *A*03:2021 *Injection*. Це може свідчити про високий рівень розповсюдження вразливостей типу *SQL*-ін’єкцій, *XSS* та командних ін’єкцій. Серед інших передбачених категорій також є *A*01:2021 *Broken Access Control*, *A*04:2021 *Insecure Design* та *A*05:2021 *Security Misconfiguration*.

Висновки

Розроблена модель дерева рішень може ефективно використовуватись для автоматизованої обробки журналів вразливостей та попередньої класифікації нових інцидентів за категоріями *OWASP Top* 10. Це дозволяє пришвидшити аналіз інцидентів, виявляти найбільш розповсюджені типи загроз та визначати пріоритетність заходів щодо усунення вразливостей.

Дана модель може бути інтегрована у систему моніторингу безпеки або використовуватись для оцінки ризиків, особливо в умовах великих об’ємів даних, де ручна класифікація є неефективною.

Література

1. Катеринчук І. С., Овчар М. М. Інформаційна технологія оцінки кількості, невизначеності та достатності інформації в системах підтримки прийняття рішень. *Наука і техніка сьогодні. Серія «Техніка»*. 2022. № 9(9). С. 49–61. DOI: 10.52058/2786-6025-2022-9(9)-49-61.

2. Безсонний В. Л., Третьяков О. В., Доронін Є. В. Оптимізація систем захисту критичної інфраструктури на основі інформаційної ентропії. Проблеми інформатизації : тези доповідей одинадцятої міжнародної науково-технічної конференції, Баку, Азербайджаан, Харків, України, Бельсько-Бяла, Пóльща, 16–17 листопада 2023 / НУО АР, НТУ “ХПІ”, ХНУРЕ, НАУ "ХАІ", УТіГН, 2023. Т. 2. С. 79–83. URL: https://repository.hneu.edu.ua/bitstream/123456789/32294/1/Konf_pi_11_vol_2_30%2079.pdf.

3. Lamrini B. Contribution to Decision Tree Induction with Python: A Review. *Data Mining – Methods, Applications and Systems* / ed. by D. Birant. 2020. DOI: https://doi.org/10.5772/intechopen.92438.

4. Прокопов В. В. та ін. Розробка системи виявлення кіберзагроз на основі аналізу даних з веб-ресурсів на мові програмування Python. *Системи управління, навігації та зв'язку*. 2022. Том 2, № 68 С. 79–84. DOI: 10.26906/SUNZ.2022.2.079.

5. Shannon C. E. A Mathematical Theory of Communication. *Bell System Technical Journal*. 1948. Vol. 27. P. 379–423.

6. Quinlan J. R. Induction of Decision Trees. *Machine Learning*. 1986. Vol. 1(1). P. 81–106.

7. OWASP Foundation. (2021). OWASP Top Ten Web Application Security Risks. URL: https://owasp.org/Top10/.

8. Pedregosa F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*. 2011. Vol. 12. P. 2825–2830. URL: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

9. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. New York : Springer, 2009. 745 p.

10. Tan P.-N., Steinbach M., Kumar V. Introduction to Data Mining. 2nd ed. New York : Pearson, 2018. 864 p.

11. Терейковський І. А., Корченко А. О. Інтелектуалізовані методи захисту інформації: нейронні мережі в захисті інформації : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2022. 176 с.

12. Cover T. M., Thomas J. A. Elements of Information Theory. 2nd ed. Hoboken : John Wiley & Sons, Inc, 2006. 774 p.

Мельниченко П.І.,

РОЗРОБКА МОДЕЛІ ПРОГНОЗУВАННЯ ВРАЗЛИВОСТЕЙ ВЕБ ЗАСТОСУНКІВ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Зі зростанням кількості кіберзагроз оцінка ризиків на основі даних про вразливості стає важливим завданням. Одним з основних підходів є використання методів ентропії та дерева рішень, що використовуються для оцінки ступеня невизначеності та класифікації інформації. В роботі розглянуто метод ентропії Шеннона та алгоритм побудови дерев рішень для аналізу та прогнозу вразливостей. Побудована модель дерева, яка прогнозує майбутні вразливості на основі датасету вразливостей. Також розраховано ентропію на основі рівня загроз, щоб кількісно оцінити невизначеність даних.

***Ключові слова**: ентропія; метод дерева рішень; OWASP; вразливості; оцінка ризиків; захист інформації.*

Melnychenko P.I.

DEVELOPMENT OF A MODEL FOR FORECASTING WEB APPLICATION VULNERABILITIES USING MACHINE LEARNING METHODS

With the increasing number of cyber threats, risk assessment based on vulnerability data has become a critical task. One of the main approaches involves the use of entropy methods and decision trees, which are applied to evaluate the degree of uncertainty and classify information. This study explores Shannon entropy and the decision tree construction algorithm for analyzing and predicting vulnerabilities. A decision tree model was developed to forecast future vulnerabilities based on a vulnerability dataset. Entropy was also calculated based on threat levels to quantitatively assess the uncertainty in the data.

***Keywords**: entropy; decision tree method; OWASP; vulnerabilities; risk assessment; information security.*