УДК 004.77+0004.56

**А. В. Борусевич**
**Л. М. Куперштейн**

# МАШИННЕ НАВЧАННЯ У ЗАДАЧАХ ВИЯВЛЕННЯ ТИПУ ОПЕРАЦІЙНОЇ СИСТЕМИ ВІДДАЛЕНОГО ХОСТА

Вінницький національний технічний університет

***Анотація***

*У даній роботі розглянуто результати експерименту та моделювання методів машинного навчання для реалізації технології виявлення типу операційної системи пристрою. Наведено результати тестування навченої моделі та порівняно із існуючим засобом Nmap.*

**Ключові слова:** операційна система, виявлення, безпека, мережа.

***Abstract***

*This paper considers the results of experiment and modeling of machine learning methods for the implementation of technology for detecting the type of operating system of the device. The results of testing the trained model and in comparison with the existing Nmap tool are given.*

**Keywords**: operating system, detection, security, network.

## Вступ

Зі збільшенням кількості пристроїв у мережах зростає кількість загроз, що пов'язані із вразливостями певних операційних систем або їх версій. Зазвичай це старі версії операційних систем, що не містять останніх оновлень безпеки. Адміністраторам необхідно мати актуальну інформацію щодо версій операційних систем, що встановлені на пристроях в мережі, а також отримувати актуальну інформацію щодо наявних в мережі пристроях.

Метою роботи є підвищення ефективності методів та засобів виявлення типу операційної системи за рахунок використання методів машинного навчання із застосуванням власного підходу щодо збору даних для їх навчання.

## Методика збору та підготовки даних

Визначення типу операційної системи вузла відноситься до задач класифікації. Зазвичай такі задачі використовуються із застосування методів контрольованого навчання. Для застосування такого підходу необхідно мати набір даних для навчання. Проте в мережі Інтернет практично відсутній набір, який можна було б використати для навчання моделей, оскільки, як показав аналіз, більшість з наборів орієнтований на визначення сімейства операційної системи (Linux, MacOS, Windows), бо не має потрібних параметрів, які дозволяють розрізняти типи/версію між собою. Для цього було виконано експеримент, який включав у себе 6 етапів:

1. Генерація трафіку. Для цього етапу було виконано наступні дії на пристроях, де встановлені операційні системи: надсилання 20 ICMP пакетів з полем Type 8 Echo Request з операційної системи, що вивчається; перегляд відео на веб-ресурсі YouTube (тривалістю не менше ніж 30 секунд) з операційної системи, що вивчається; перегляд різних веб-сторінок з операційної системи, що вивчається; завантаження зображень з мережі Інтернет з операційної системи, що вивчається. Даний перелік дій дозволяє отримати необхідні пакети протоколів IP, TCP, ICMP [1].
2. Збір трафіку. У цьому етапі відбувалось безпосередній запис трафіку у локальній мережі. Для цього було використано безкоштовний додаток Wireshark, який дозволяє записати у файл пакети трафіку для подальшої обробки без необхідності бути підключеним до мережі.

3. Аналіз трафіку. Даний етап включає у себе аналіз параметрів протоколів у перехоплених пакетах. В результаті було отримано значення наступних параметрів: протокол IP – version, hdr_len, dsfield, dsfield_dscp, dsfield_ecn, len, id, flags, flags_rb, flags_df, flags_mf, frag_offset, ttl, proto, checksum, checksum_status; протокол ICMP – type, code, checksum, checksum_status, ident, seq, seq_le, data, data_data, data_len; протокол TCP – hdr_len, flags, flags_res, flags_ns, flags_cwr, flags_ecn, flags_urg, flags_ack, flags_push, flags_reset, flags_syn, flags_fin, flags_str, window_size_value, window_size, window_size_scalefactor, checksum, checksum_status; протокол DNS – id, flags, flags_response, flags_opcode, flags_truncated, flags_recdesired, flags_z, flags_checkdisable, count_queries, count_answers, count_auth_rr, count_add_rr, qry_name_len, count_labels, qry_type, qry_class; протокол HTTP – user_agent.

4. Попередня обробка. Етап включає в себе перетворення текстових значень параметрів протоколів у числові. Для назв операційних систем застосовувався метод «label encoding» (кожній операційній системі призначається відповідне число) [2], для деяких параметрів пакетів (IP: dsfield, flags; TCP: flags, flags_str; DNS: flags, qry_class) використовувався метод «one hot encoding» (кодування значень шляхом створення колонок для кожного значення параметру і встановлення значення 1 в разі співпадіння з колонкою і 0 в усіх інших колонках) [3].

5. Визначення важливості параметрів. Після етапу попередньої обробки необхідно було виконати зменшення розмірності параметрів для зменшення кількості ознак (кількість колонок зросла з 54 до 156). Даний крок дозволяє збільшити швидкість та якість навчання (зменшення появи явища «перенавчання» моделі) [4]. Під час даного процесу використовувався метод RFE (Recursive Feature Elimination) – рекурсивний метод виділення ознак. Бажаною кількістю ознак було встановлено 15. Після виконання зменшення розмірності було отримано наступні результати: IP – hdr_len, flags, flags_df, ttl, proto; ICMP – ident, seq, seq_le, data_data, data_len; TCP – hdr_len, window_size_value, window_size, window_size_scalefactor; DNS – count_labels.

6. Моделювання. Після визначення важливостей параметрів, усі дані були сформовані у сигнатури операційних систем. Далі сигнатури були надані моделям машинного навчання для навчання. Результати навчання моделей наведено в табл. 1.

Табл. 1 – Результати моделювання

| Назва класифікатора | Accuracy | Precision | Recall | F1 | FP | FN |
|---|---|---|---|---|---|---|
| Decision Tree | 1.0 | 1.0 | 1.0 | 1.0 | 0 | 0 |
| Multilayer Perceptron | 0.9980 | 0.9980 | 0.9981 | 0.9980 | 0 | 4 |
| Gaussian Naïve Bayes | 0.8580 | 0.9025 | 0.8602 | 0.8172 | 0 | 290 |
| K-Nearest Neighbors | 0.9828 | 0.9831 | 0.9829 | 0.9829 | 0 | 35 |
| Support Vector Machine | 0.9990 | 0.9990 | 0.9991 | 0.9990 | 0 | 2 |
| Logistic Regression | 0.9716 | 0.9742 | 0.9718 | 0.9719 | 0 | 58 |
| Random Forest | 1.0 | 1.0 | 1.0 | 1.0 | 0 | 0 |

Проаналізувавши результати, можна зробити наступні висновки: більшість помилок відбувались в межах однієї платформи операційної системи і вони стосувались версії; найкращими класифікаторами є дерево рішень (Decision Tree) та випадковий ліс (Random Forest).

Проте випадковий ліс має недолік, а саме – великий час навчання, оскільки відбувається навчання декількох дерев рішень. Тому було для реалізації модуля інтелектуального аналізу було обрано модель дерево рішень.

**Результати тестування**

На основі проведених експериментів було розроблено програмне забезпечення мовою Python. Було проведено тестування розробленого засобу і порівняно з результатами сканера Zenmap. Перше сканування виконувалось для вузла з операційною системою Windows 10 Corporate (192.168.1.170). Результати наведено на рис. 1 (зліва – власна інтелектуальна система, справа – Nmap). Результати показують, що Nmap не зміг визначити точно встановлену операційну систему. Також в результатах ми бачимо широкий спектр можливих операційних систем (Windows 10, 7, Windows Phone, Windows Server, FreeBSD), а ймовірність дорівнює 0,92 порівняно з 0,955 у інтелектуальної системи. Друге

сканування вузла з ОС Windows 10 Home 20h2 і адресою 192.168.43.60. Результати наведено на рис. 2. Nmap не надає результат і повідомляє, що хост має забагато збігів сигнатур, тоді як інтелектуальна система надає правильну відповідь. Третє сканування було виконано для орендованого хоста з білою IP-адресою. На ньому попередньо встановлена ОС Linux 5.4. Результати сканування наведено на рис. 3. Інтелектуальна система виявила ОС з імовірністю 0,987, Nmap виявив Windows 7.

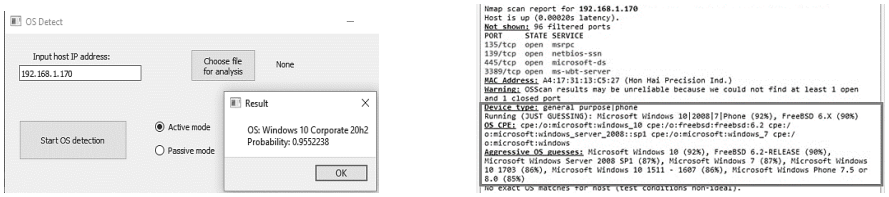

Рис. 1 – Перше сканування

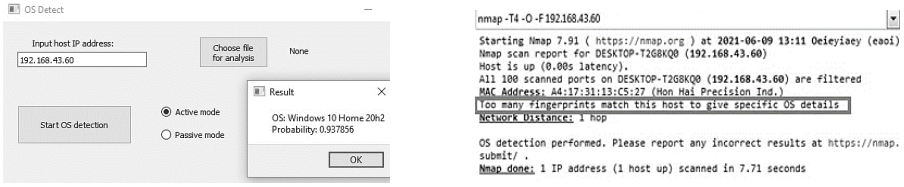

Рис. 2 – Друге сканування

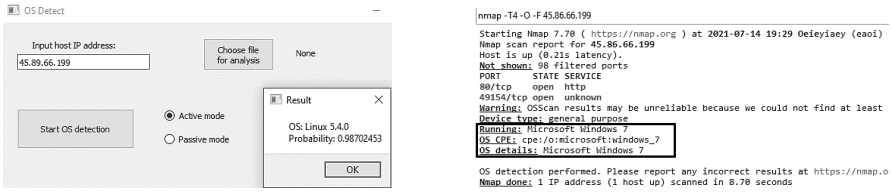

Рис. 3 – Третє сканування

## Висновки

Навчена модель машинного навчання може з високою точністю виявити сім версій ОС у кількох сімействах. Цей програмний інструмент дозволяє сканувати вузли як в автономному, так і в онлайн-режимі. Розроблений інструмент може бути використаний етичним хакером для тестування на вторгнення, мережевим адміністратором для аудиту, перевірки мережі на наявність нових невідомих пристроїв. Ви також можете використовувати цей інструмент для перевірки ефективності засобів захисту серверів від виявлення їх ОС. В результаті експерименту розроблене програмне забезпечення виявилося більш ефективним порівняно з Nmap. Однак Nmap дозволяє визначити набагато більше типів ОС. Тому подальші дослідження будуть пов’язані з масштабуванням моделі, а також розширенням функціональності програми. Однією з таких майбутніх функцій буде реалізація можливості надання списку ранжованих вразливостей, властивих певній ОС.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Смирнова Е., Пролетарский А., Ромашкина Е. Технологии TCP/IP в современных компьютерных сетях. Посібник. Москва : Издательство МГТУ им. Н.Э. Баумана, 2019. 638 с.

2. Label Encoding in Python Explained : веб-сайт. URL: https://www.mygreatlearning.com/blog/label-encoding-in-python/ (дата звернення 01.02.2022)

3. How to One Hot Encode Sequence Data in Python : веб-сайт. URL: https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/ (дата звернення 01.02.2022)

4. Васюра А.С., Мартинюк Т.Б., Куперштейн Л.М. Методи та засоби нейроподібної обробки даних для систем керування. Монографія. Вінниця: УНІВЕРСУМ−Вінниця, 2008. 175 с.

5. Борусевич А., Куперштейн Л. Аналіз методів та засобів виявлення типу операційної системи віддаленого вузла : веб-сайт. URL: https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12193 (дата звернення 01.02.2022).

***Борусевич Артур Вячеславович*** — студент групи 1БС-21М, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, e-mail: borusevych.av@gmail.com

***Куперштейн Леонід Михайлович —*** кандидат технічних наук, доцент кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця

***Borusevych Artur V.*** — Student of Information Technologies and Computer Engineering Department, Vinnytsia National Technical University, email : borusevych.av@gmail.com

***Kupershtein Leonid M.*** — PhD, Associated Professor of Information Protection Chair, Vinnytsia National Technical University, Vinnytsia, email: kupershtein.lm@gmail.com