BÁO CÁO THỰC HÀNH

**Môn học: Phương pháp học máy cho an toàn thông tin**

**Lab 3: Advanced Malware Detection**

*GVHD: Nguyễn Hữu Quyền*

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT522.O21.ATCL.2

Nhóm: 7

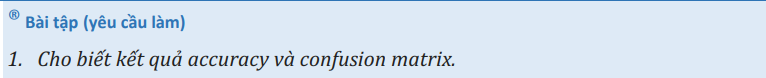
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Hồ Ngọc Thiện | 21522620 | 21522620@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Bài tập 1 | 100% |
| 2 | Bài tập 2 | 100% |
| 3 | Bài tập 3 | 100% |
| 4 | Bài tập 4 | 100% |
| 5 | Bài tập 5 | 100% |
| 6 | Bài tập 6 | 100% |
| 7 | Bài tập 7 | 100% |
| 8 | Bài tập 8 | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT



A screen shot of a computer program

Description automatically generated

-Code:

import os

from sklearn.feature\_extraction.text import HashingVectorizer, TfidfTransformer

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score, confusion\_matrix

from sklearn.pipeline import Pipeline

js\_path = "JavascriptSamples"

obfuscated\_js\_path = "JavascriptSamplesObfuscated"

corpus = []

labels = []

file\_types\_and\_labels = [(js\_path, 0), (obfuscated\_js\_path, 1)]

for files\_path, label in file\_types\_and\_labels:

files = os.listdir(files\_path)

for file in files:

file\_path = files\_path + "/" + file

try:

with open(file\_path, "r") as myfile:

data = myfile.read().replace("\n", "")

data = str(data)

corpus.append(data)

labels.append(label)

except:

pass

X\_train, X\_test, y\_train, y\_test = train\_test\_split( corpus, labels, test\_size=0.33, random\_state=42 )

text\_clf = Pipeline([

("vect", HashingVectorizer(input="content", ngram\_range=(1,3))),

("tfidf", TfidfTransformer(use\_idf=True,)),

("rf", RandomForestClassifier(class\_weight="balanced")),

])

text\_clf.fit(X\_train, y\_train)

y\_test\_pred = text\_clf.predict(X\_test)

print(accuracy\_score(y\_test, y\_test\_pred))

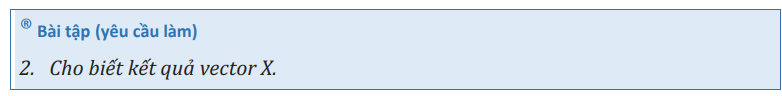
print(confusion\_matrix(y\_test, y\_test\_pred))

-Kết quả lần lượt là accuracy và ma trận confusion:

A screen shot of a computer

Description automatically generated

EX2



A screen shot of a computer program

Description automatically generated

-Code:

import subprocess

from os import listdir

def PDF\_to\_FV(file\_path):

"""Featurize a PDF file using pdfid."""

result = subprocess.run(["pdfid", file\_path], capture\_output=True, text=True)

out = result.stdout

out1 = out.split("\n")[2:-2]

return [int(x.split()[-1]) for x in out1]

PDFs\_path = "PDFSamples/"

X = []

files = listdir(PDFs\_path)

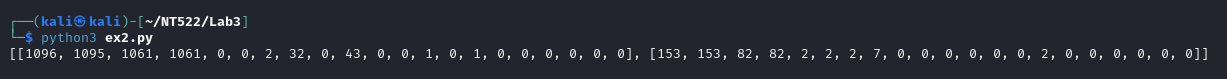
for file in files:

file\_path = PDFs\_path + file

X.append(PDF\_to\_FV(file\_path))

print(X)

-Kết quả vector x



EX3:

A blue rectangle with a blue border

Description automatically generated

-Code:

from os import listdir

from nltk import ngrams

import hashlib

directories = ["Benign 3", "Malware 3"]

N = 2

def read\_file(file\_path):

"""Reads in the binary sequence of a binary file."""

with open(file\_path, "rb") as binary\_file:

data = binary\_file.read()

return data

def byte\_sequence\_to\_Ngrams(byte\_sequence, N):

"""Creates a list of N-grams from a byte sequence."""

return ngrams(byte\_sequence, N)

def read\_file(file\_path):

"""Reads in the binary sequence of a binary file."""

with open(file\_path, "rb") as binary\_file:

data = binary\_file.read()

return data

def byte\_sequence\_to\_Ngrams(byte\_sequence, N):

"""Creates a list of N-grams from a byte sequence."""

return ngrams(byte\_sequence, N)

def hash\_input(inp):

"""Compute the MD5 hash of an input."""

return int(hashlib.md5(inp).hexdigest(), 16)

def make\_ngram\_hashable(Ngram):

"""Convert N-gram into bytes to be hashable."""

return bytes(Ngram)

def hash\_file\_Ngrams\_into\_dictionary(file\_Ngrams, T):

"""Hashes N-grams in a list and then keeps track of the counts in a dictionary."""

for Ngram in file\_Ngrams:

hashable\_Ngram = make\_ngram\_hashable(Ngram)

hashed\_and\_reduced = hash\_input(hashable\_Ngram) % B

T[hashed\_and\_reduced] = T.get(hashed\_and\_reduced, 0) + 1

B = 65521

T = {}

for dataset\_path in directories:

samples = [f for f in listdir(dataset\_path)]

for file in samples:

file\_path = dataset\_path + "/" + file

file\_byte\_sequence = read\_file(file\_path)

file\_Ngrams = byte\_sequence\_to\_Ngrams(file\_byte\_sequence, N)

hash\_file\_Ngrams\_into\_dictionary(file\_Ngrams, T)

K1 = 1000

import heapq

K1\_most\_common\_Ngrams\_Using\_Hash\_Grams = heapq.nlargest(K1, T)

def featurize\_sample(file, K1\_most\_common\_Ngrams\_Using\_Hash\_Grams):

"""Takes a sample and produces a feature vector. The features are the counts of the K1 N-grams we've selected."""

K1 = len(K1\_most\_common\_Ngrams\_Using\_Hash\_Grams)

fv = K1 \* [0]

file\_byte\_sequence = read\_file(file\_path)

file\_Ngrams = byte\_sequence\_to\_Ngrams(file\_byte\_sequence, N)

for Ngram in file\_Ngrams:

hashable\_Ngram = make\_ngram\_hashable(Ngram)

hashed\_and\_reduced = hash\_input(hashable\_Ngram) % B

if hashed\_and\_reduced in K1\_most\_common\_Ngrams\_Using\_Hash\_Grams:

index = K1\_most\_common\_Ngrams\_Using\_Hash\_Grams.index(hashed\_and\_reduced)

fv[index] += 1

return fv

X = []

for dataset\_path in directories:

samples = [f for f in listdir(dataset\_path)]

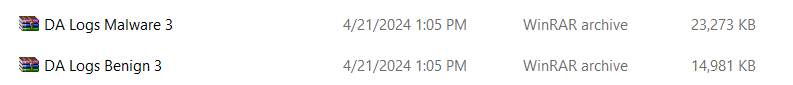
for file in samples:

file\_path = dataset\_path + "/" + file

X.append(featurize\_sample(file\_path, K1\_most\_common\_Ngrams\_Using\_Hash\_Grams))

print(X)

-Sử dụng 2 file dataset:



-Kết quả ta có được là chuỗi vector x

A black and white background with small dots

Description automatically generated

A screen shot of a computer screen

Description automatically generated

**A white rectangular object with blue border

Description automatically generated**

- Import các thư viện cần thiết và gán nhãn directory chứa các file JSON.Ở đây, ta chọn tập Benign 1 và tập Malware 1.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

- Hàm parse nhật ký JSON và trích xuất class, method và type từ API call.

**A screen shot of a computer code

Description automatically generated**

**-** Trích xuất dữ liệu và phân loại từ các tập dữ liệu có sẵn và gán nhãn tương ứng, nhằm tạo ra tập dữ liệu huấn luyện.

**A screenshot of a computer program

Description automatically generated**

**-** Chia tập dữ liệu thành tập train và tập test với corpus\_train và corpus\_test là tập dữ liệu train và test chứa các mẫu dữ liệu , y\_train và y\_test là nhãn tương ứng với tập train và tập test.

**A white background with black text

Description automatically generated**

**-** Hàm read\_file đọc nội dung của file ở dạng nhị phân và trả về giá trị nhị phân tương ứng.

- Hàm text\_to\_Ngrams tạo ra danh sách các N-grams .

- Cuối cùng hàm get\_Ngram\_counts tính toán tần suất xuất hiện các N-grams trong 1 văn bản đã cho.

**A screenshot of a computer program

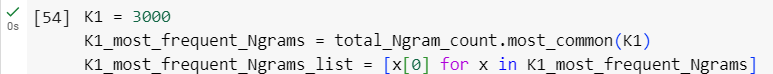
Description automatically generated**

**-** Tính tổng số lần xuất hiện các N-grams với số từ của 1 N-grams là 4 (N = 4) từ tập train.

**A screenshot of a computer

Description automatically generated**

**-** Tính toán danh sách các N-gram phổ biến nhất trong tập huấn luyện được sắp xếp theo thứ tự giảm dần số lần xuất hiện và có K1 = 3000 phần tử.

****

**-** Hàm chuyển đổi mỗi mẫu dữ liệu (văn bản) thành một vectơ đặc trưng, mà mỗi phần tử của vectơ đó đại diện cho số lần xuất hiện của một N-gram đã chọn trong mẫu dữ liệu.

**A screenshot of a computer program

Description automatically generated**

**-** Sử dụng hàm trên để tạo tập train và tập test.

**A screenshot of a computer code

Description automatically generated**

**-** Tạo ra một pipeline để chọn ra K2 đặc trưng tốt nhất sử dụng mutual information và sau đó huấn luyện một mô hình XGBoost.

**A screenshot of a computer program

Description automatically generated**

**-** Cuối cùng là huấn luyện pipeline và đánh giá hiệu suất.

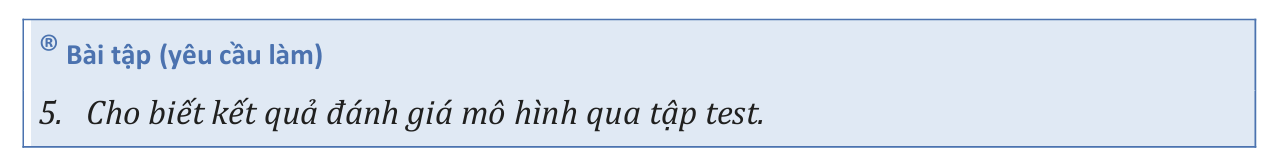
A screenshot of a computer code

Description automatically generated

\* Bài tập (yêu cầu làm)

**4. Cho biết kết quả đánh giá.**

Kết quả đánh giá : training accuracy khoảng 0.93577 và testing accuracy khoảng 0.82378.

****

- Import thư viên numpy để tính toán vector và tqdm để theo dõi tiền trình trong vòng lặp.

A close-up of black and purple text

Description automatically generated

- Định nghĩa hàm để chuyển byte thành vector.

A computer screen shot of a code

Description automatically generated

- Đọc các tin PE mẫu và dán nhãn cho chúng.

A screenshot of a computer program

Description automatically generated

- Hàm đọc chuỗi byte trong tập tin.

A computer code with black text

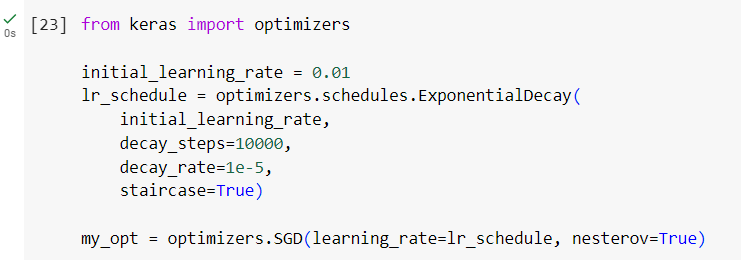
Description automatically generated with medium confidence

- Đặt độ dài tối đa, maxSize byte, để đọc cho mỗi mẫu, lấy tất cả byte của mẫu đưa vào X.

A screenshot of a computer code

Description automatically generated

- Thiết lập trình tối ưu.



- Sử dụng API của Keras để thiết lập màn thần kinh học sâu.

A screenshot of a computer program

Description automatically generated

- Biên dịch mô hình và chọn batch size.

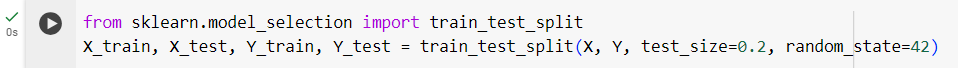
A white background with black text

Description automatically generated

A screenshot of a computer program

Description automatically generated

- Chia tập train, tập test.



- Huấn luyện mô hình.

A screenshot of a computer code

Description automatically generated

\* Bài tập (yêu cầu làm)

**5. Cho biết kết quả đánh giá mô hình qua tập test.**

- Kết quả đánh giá mô hình qua tập test.

**A screenshot of a computer code

Description automatically generated**

A close-up of a blue line

Description automatically generated

Tiến hành tạo đường dẫn đến Benign PE Samples UPX folder

A screenshot of a computer

Description automatically generated

Sau đó, ta kiểm tra hệ điều hành và tải bản UPX tương ứng

A screenshot of a computer program

Description automatically generated

Ta giải nén và kiểm tra xem file thực thi có hoạt động chính xác không

A screenshot of a computer

Description automatically generated

Sau đó tiến hành pack các tập tin lại

A screen shot of a computer

Description automatically generated

A close-up of a white rectangular object

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

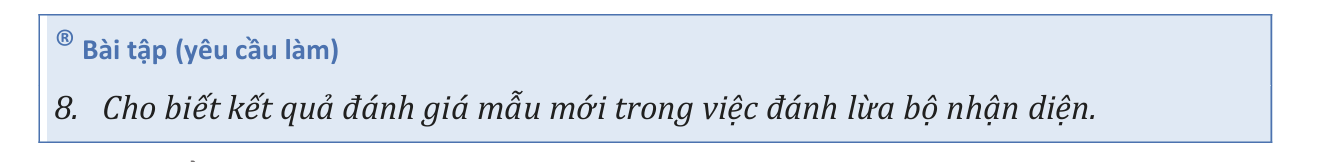
Description automatically generated

A screenshot of a computer program

Description automatically generated

A computer screen shot of a computer program

Description automatically generated



A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A close up of a text

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Đến bước này nhóm chúng em gặp lỗi, nhóm sẽ nghiên cứu thêm để sửa được lỗi này

A screenshot of a computer program

Description automatically generated

A white rectangular object with a black border

Description automatically generated

---

***Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này***

# **YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
* Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

**Báo cáo:**

* File .DOCX và .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
* Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: [Mã lớp]-ExeX\_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).

*Ví dụ: [*NT101.K11.ANTT*]-Exe01\_Group03.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài nộp.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá**:

* Hoàn thành tốt yêu cầu được giao.
* Có nội dung mở rộng, ứng dụng.

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)