**Hướng dẫn thực nghiệm Phát hiện DDos trên SDN**

1. **Mở các máy ảo trên VM Virtual Box**

* Mở Ubuntu (nơi chứa [ryu-controller](https://ryu.readthedocs.io/en/latest/getting_started.html))
* Mở Mininet VM [mininet](https://github.com/mininet/mininet/releases/)
* Connect ssh 2 máy ảo trên trên mobaXterm

1. **Clone source từ github**
   1. Trên máy Ubuntu ryu

* cd ryu/ryu/controller
* git clone <https://github.com/tranthanhhuy19032001/DACNTT2_DDos_ML.git>

username: tranthanhhuy19032001

password: ghp\_dJ13ANgAhoPo1DMLIQECFZGixaRluE1VT6Vc

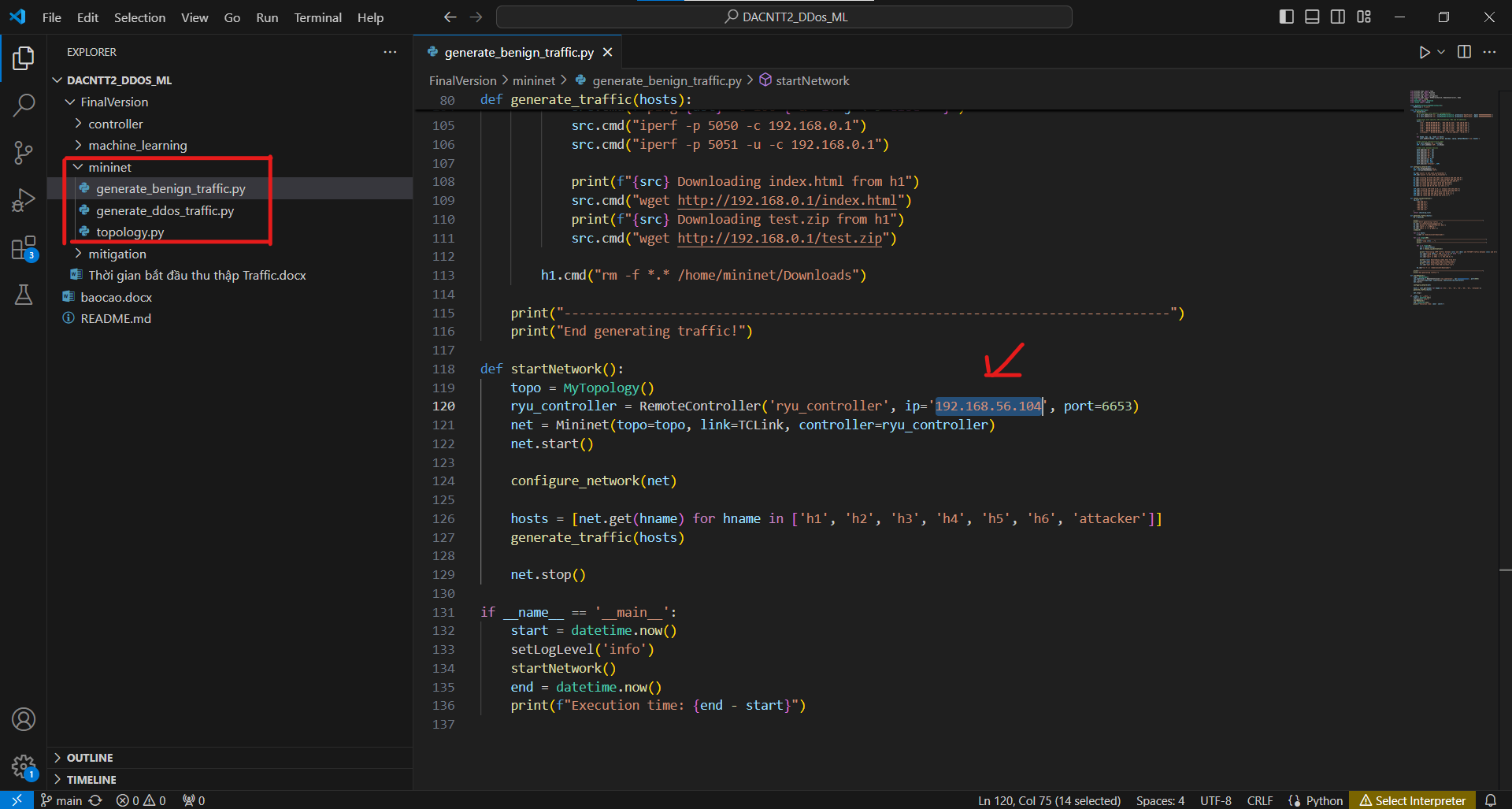
* cd DACNTT2\_DDos\_ML/FinalVersion/controller
  1. Trên Mininet VM
* cd mininet/mininet
* git clone <https://github.com/tranthanhhuy19032001/DACNTT2_DDos_ML.git>

username: tranthanhhuy19032001

password: ghp\_dJ13ANgAhoPo1DMLIQECFZGixaRluE1VT6Vc

* cd DACNTT2\_DDos\_ML/FinalVersion/mininet

Đổi địa chỉ ip trong 3 file generate\_benign\_traffic.py, generate\_benign\_traffic.py, topology.py thành ip máy ryu của bạn



* **Lệnh để mở được xterm trển mobaXterm**
* sudo -i
* export DISPLAY=localhost:10.0
* xauth add $(xauth -f ~mininet/.Xauthority list|tail -1)

1. **Generate Dataset**

Chạy các lệnh sau:

* Trên máy ryu

ryu-manager collect\_traffic.py

* Trên máy Mininet

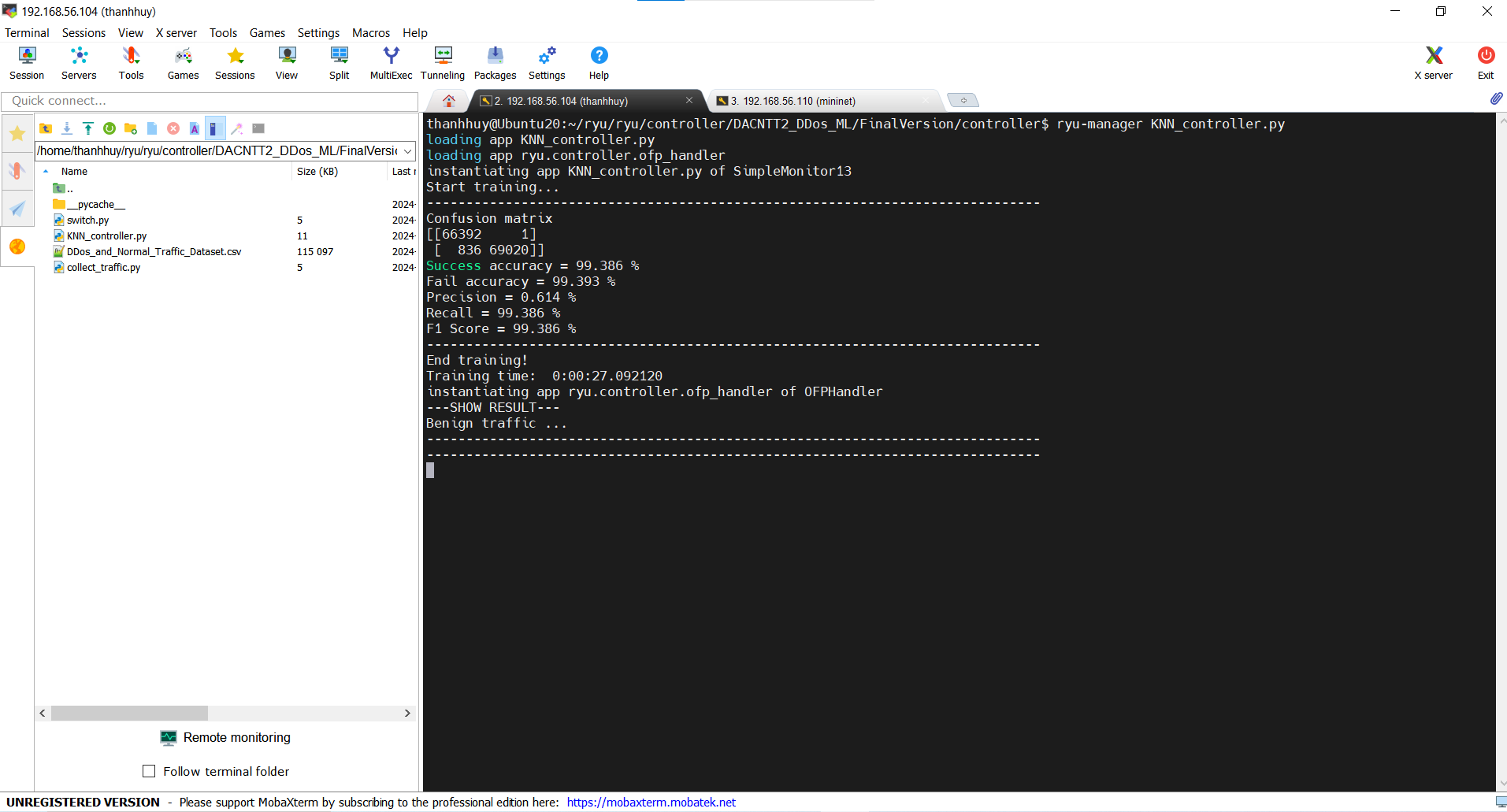
sudo python generate\_benign\_traffic.py (tạo normal traffic)

sudo python generate\_ddos\_traffic.py (tạo DDos traffic)

1. **Thực nghiệm phát hiện DDos**
   1. Trên máy RYU

ryu-manager KNN\_controller.py

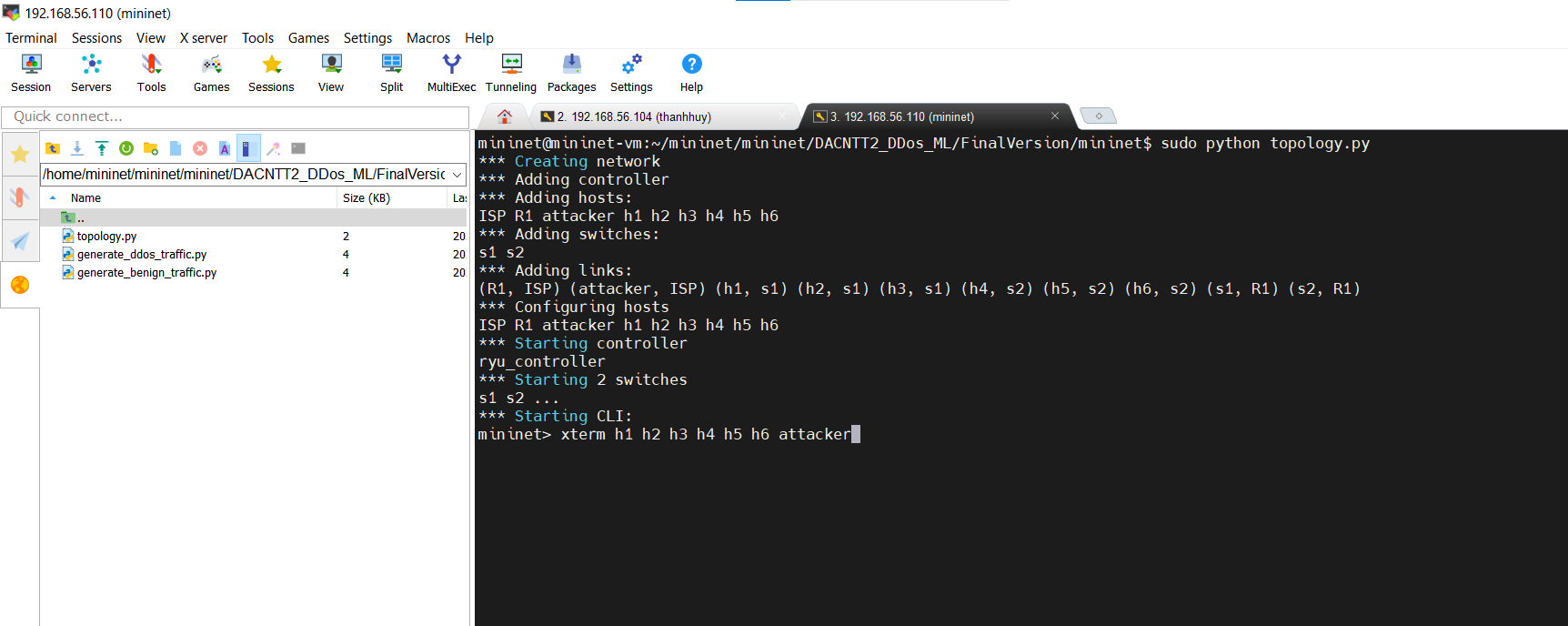
Chạy lệnh trên để khởi động controller được tích hợp model học máy KNN



* 1. Trên Mininet VM

sudo python topology.py

Chạy lệnh trên để start hệ thống mạng trên mininet



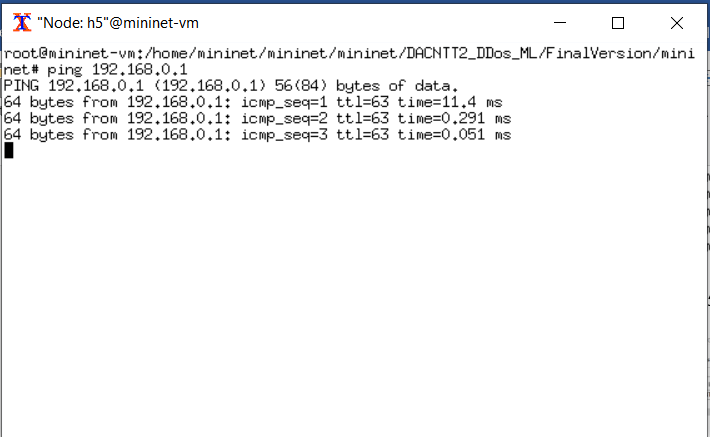
xterm h1 h2 h3 h4 h5 h6 attacker

Lệnh này dùng để mở CLI của các node trong topology

* 1. Các bước thực nghiệm và kết quả
     1. Thực hiện các lệnh ping bình thường để kiểm tra traffic bình thường

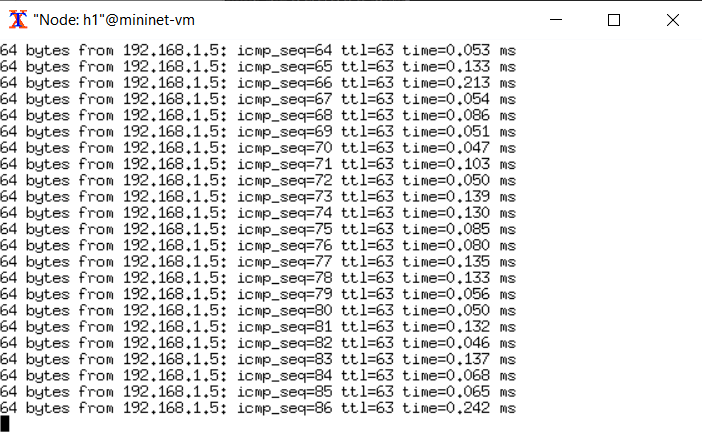
ping 192.168.0.1

Trên host h5 ping tới Web server host h1

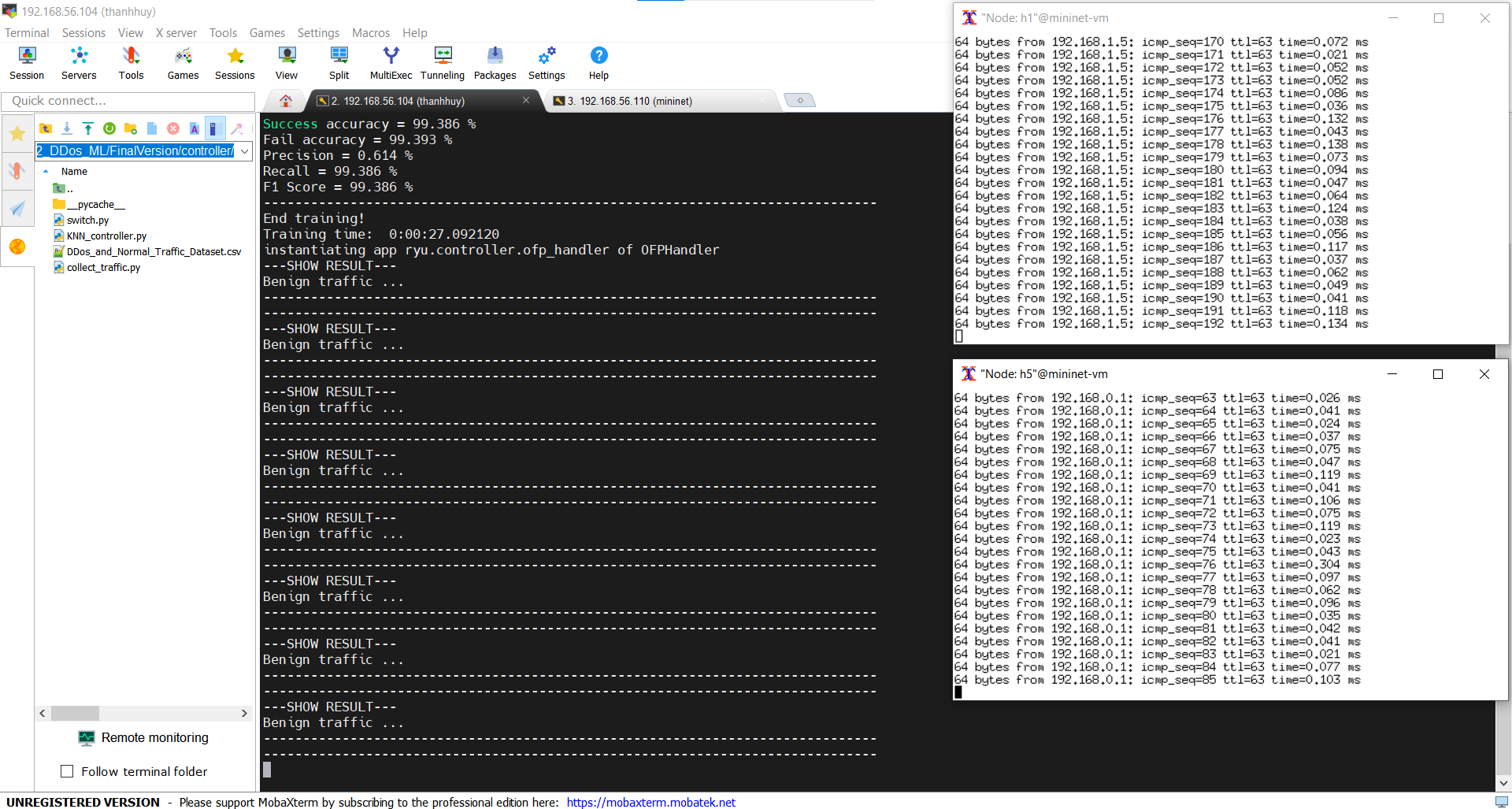


ping 192.168.1.5

Trên h1 ping tới host h5



Hình bên dưới show results predict của các traffic bình thường vừa ping.

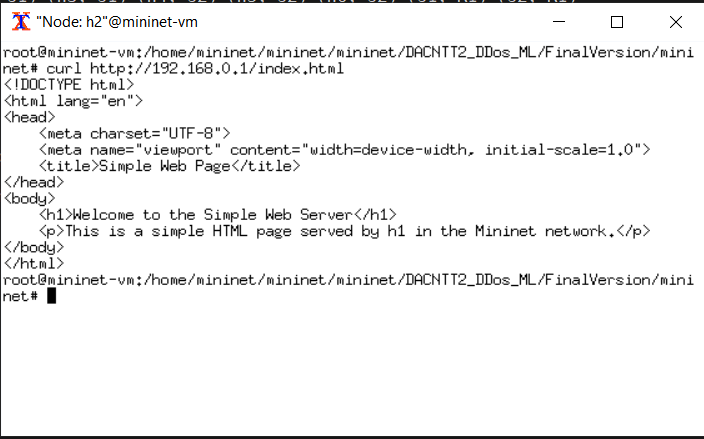


* + 1. Kiểm tra truy cập tới Web server

curl http:192.168.0.1/index.html

Gửi request GET tới trang web index.html trên Web Server

Hình bên dưới cho thấy đã truy cập thành công trang web trên Web Server

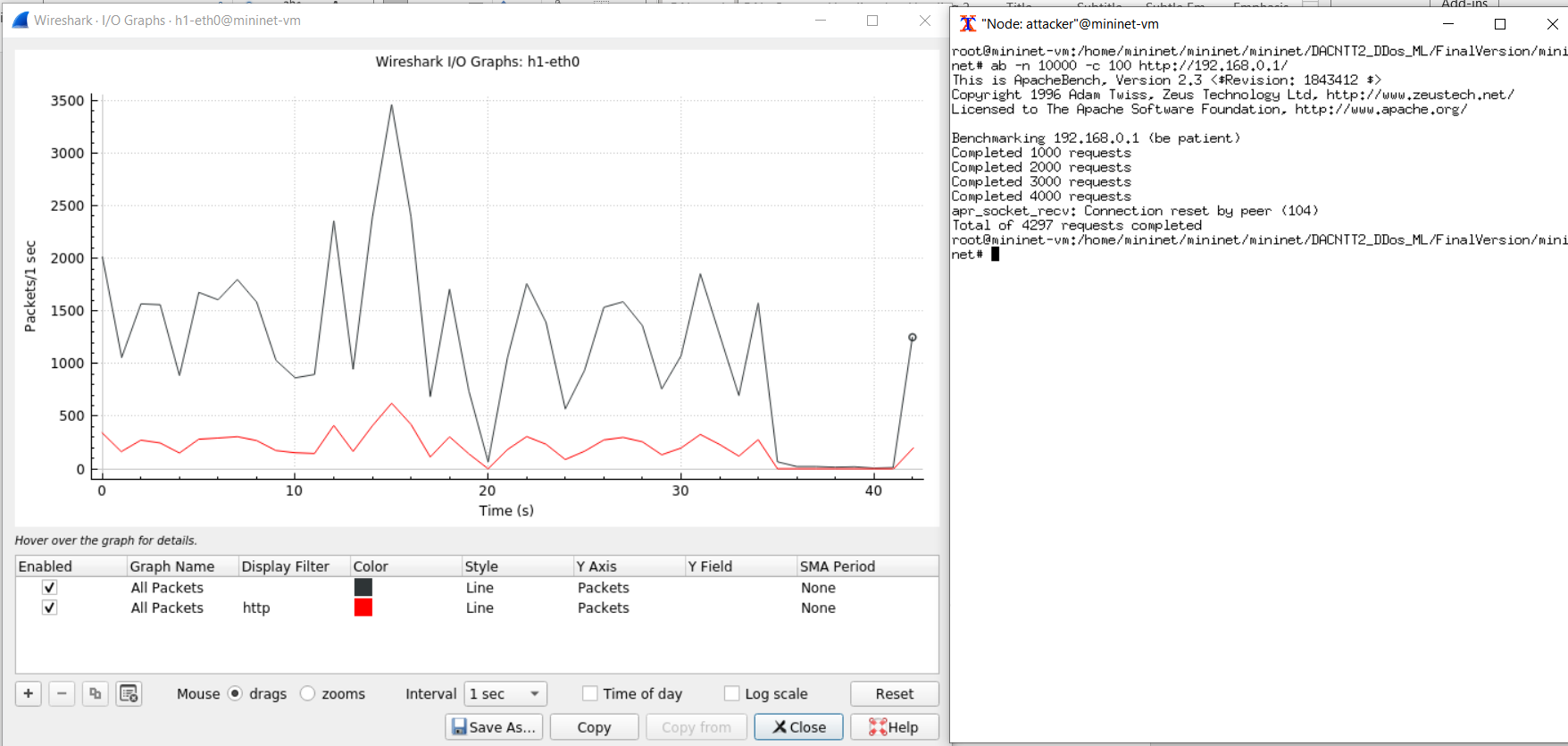


* + 1. Thực hiện tấn công HTTP Flood tới Web Server

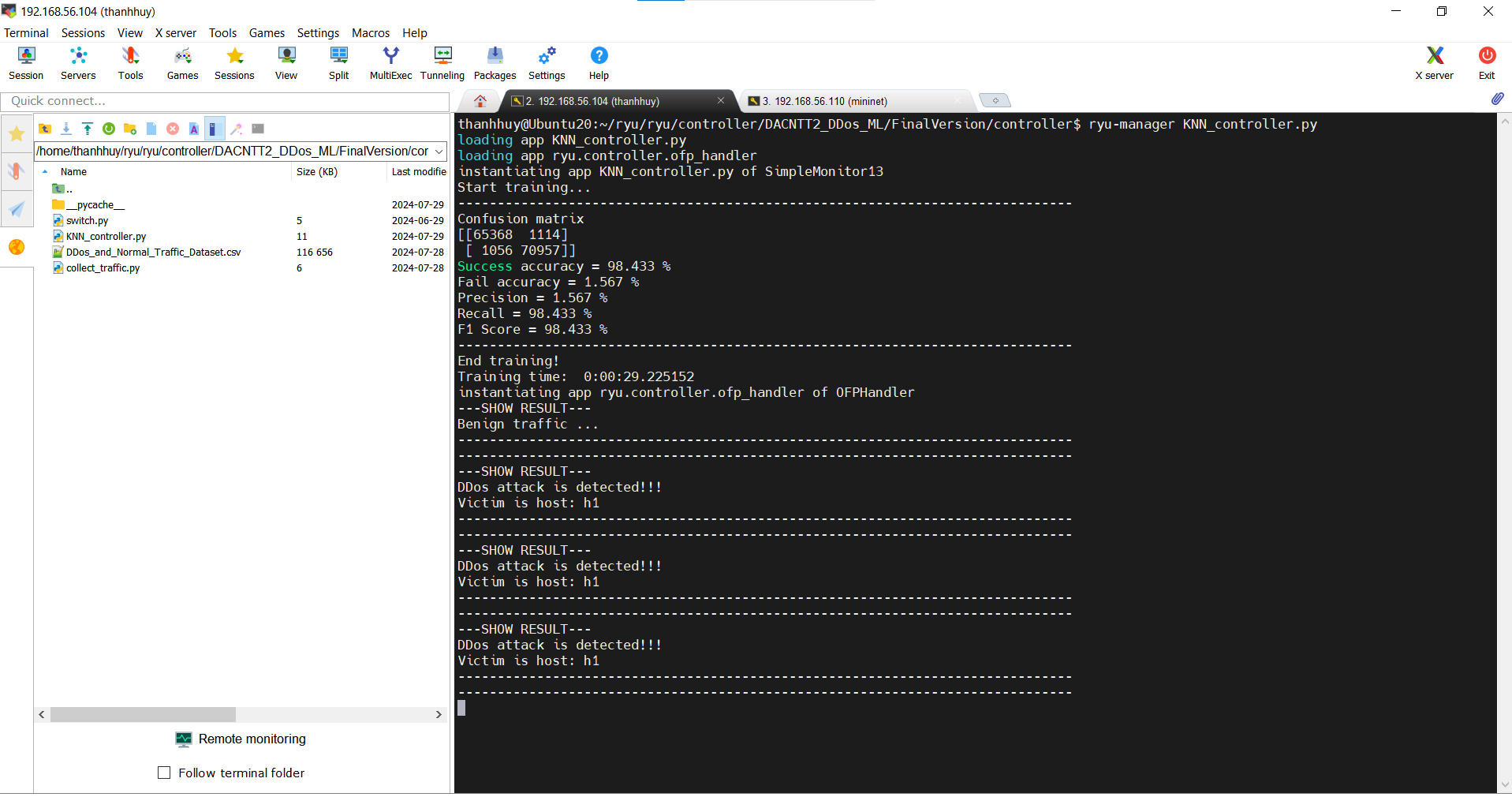
ab -n 10000 -c 100 <http://192.168.0.1/>

Lệnh trên tạo ra 10000 request với 100 kết nối đồng thời được gửi Web server (có thể tùy chỉnh các thông số)

Biểu đồ dưới hiển thị các requests đã được gửi đến Web Server trong thời gian lệnh HTTP Flood được thực thi



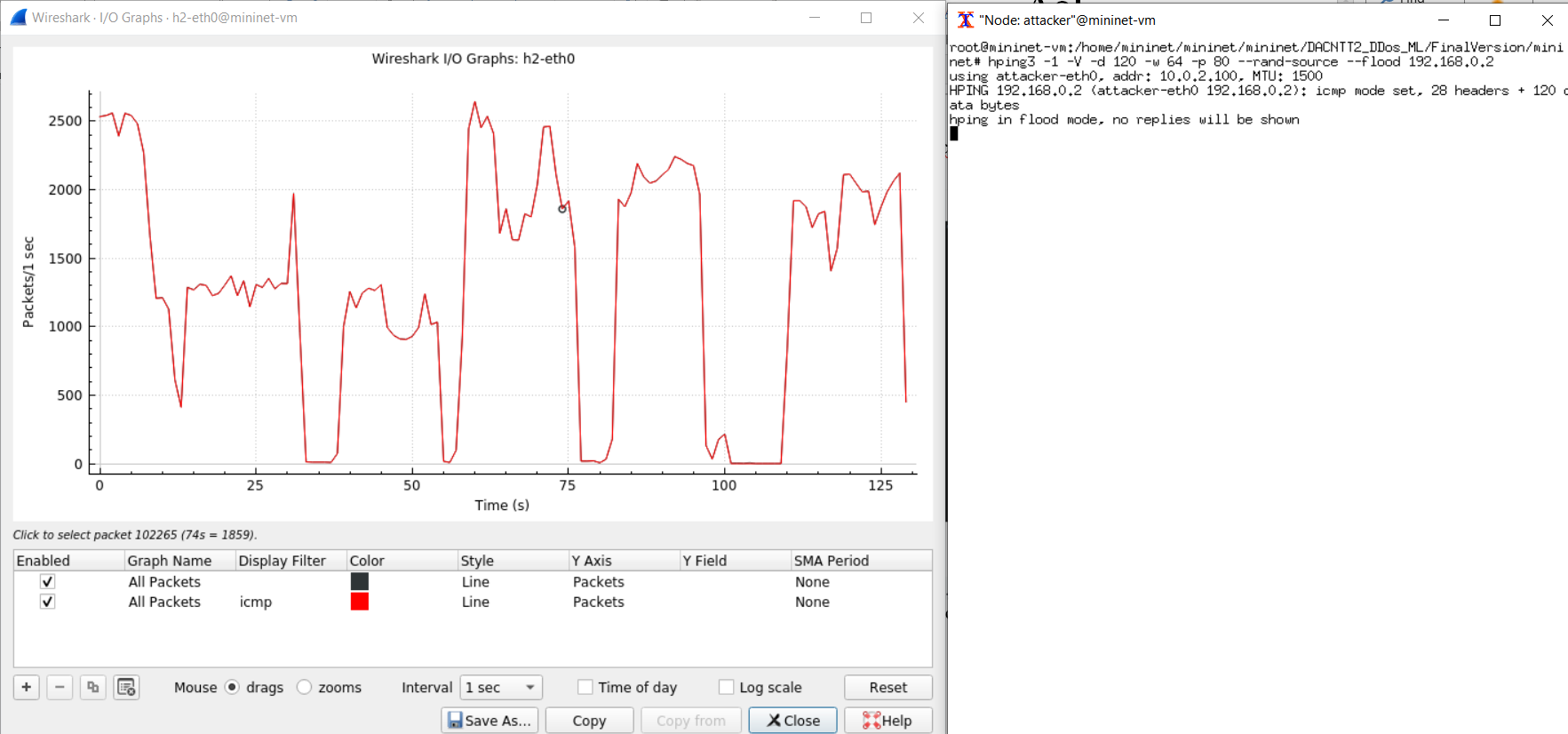
Như kết quả dự đoán hiện thị trên hình dưới, Model KNN xây dựng trong ryu-controller đã dự đoán chính xác Web Server h1 đang bị tấn công



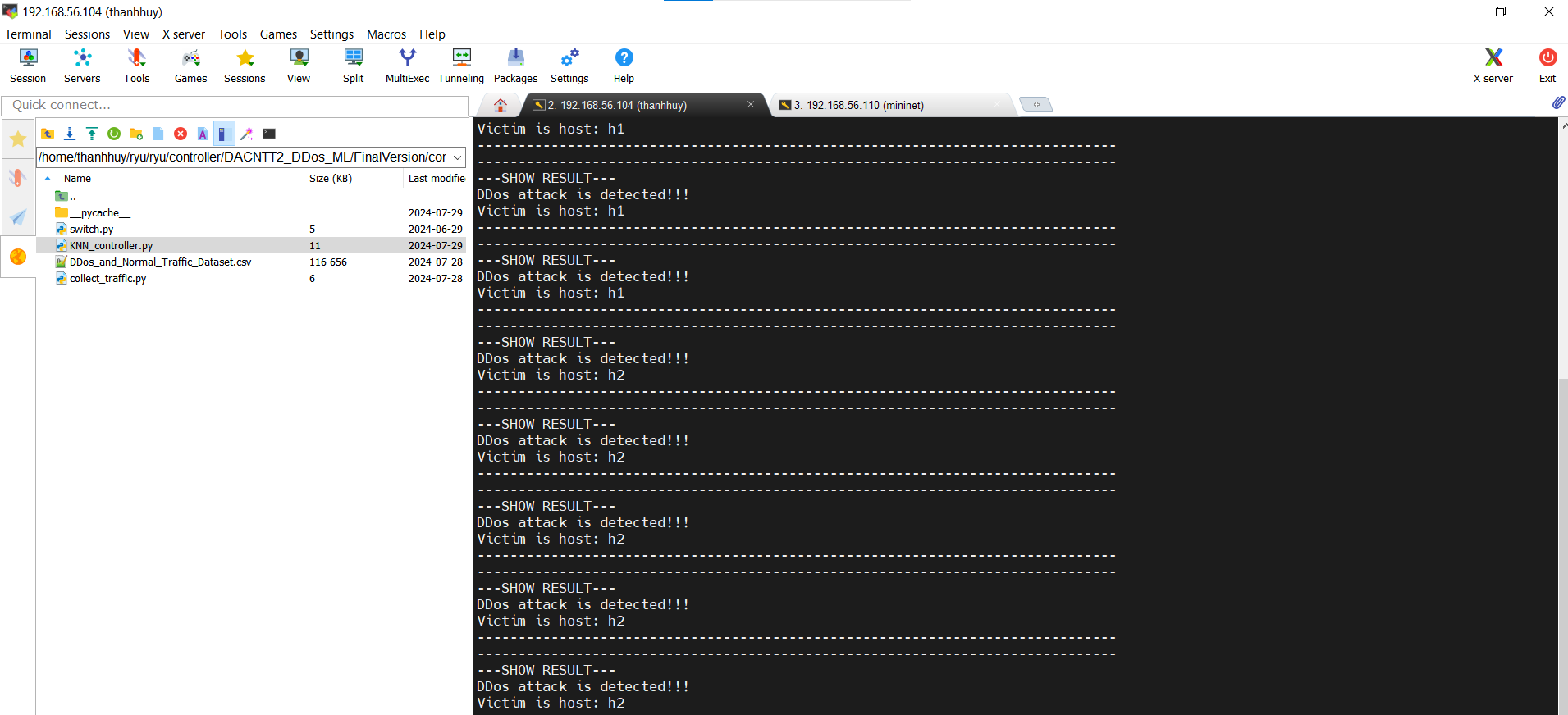
* + 1. Thực hiện tấn cống ICMP Flood tới h2

hping3 -1 -V -d 120 -w 64 -p 80 --rand-source –flood 192.168.0.1

Biểu đồ bên dưới hiển thị các request ICMP đã được tới host h2 trong khi thời gian lệnh ICMP Flood được thực thi

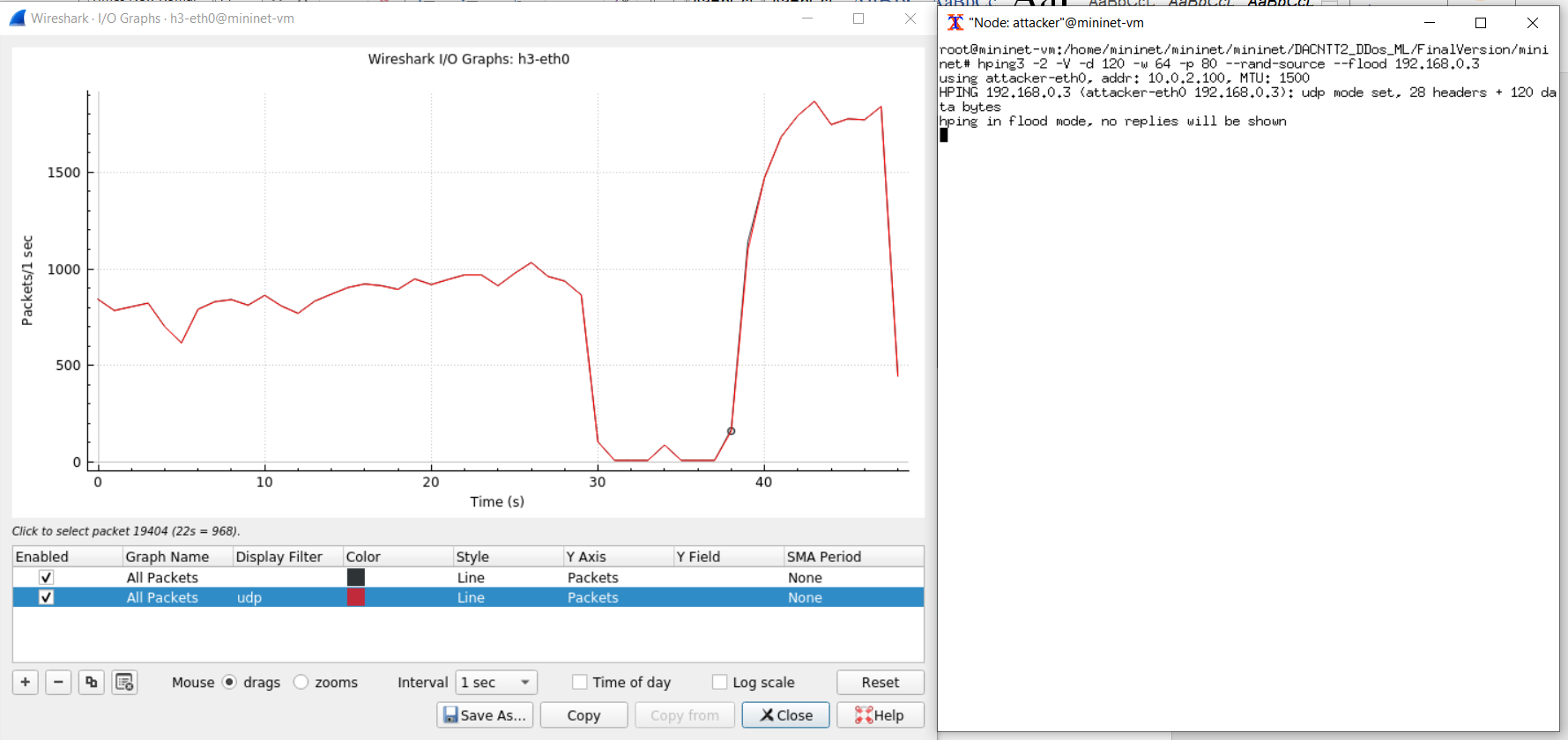


Như kết quả dự đoán hiển thị trên hình dưới, Model KNN xây dựng trong ryu-controller đã dự đoán chính xác host h2 đang bị tấn công

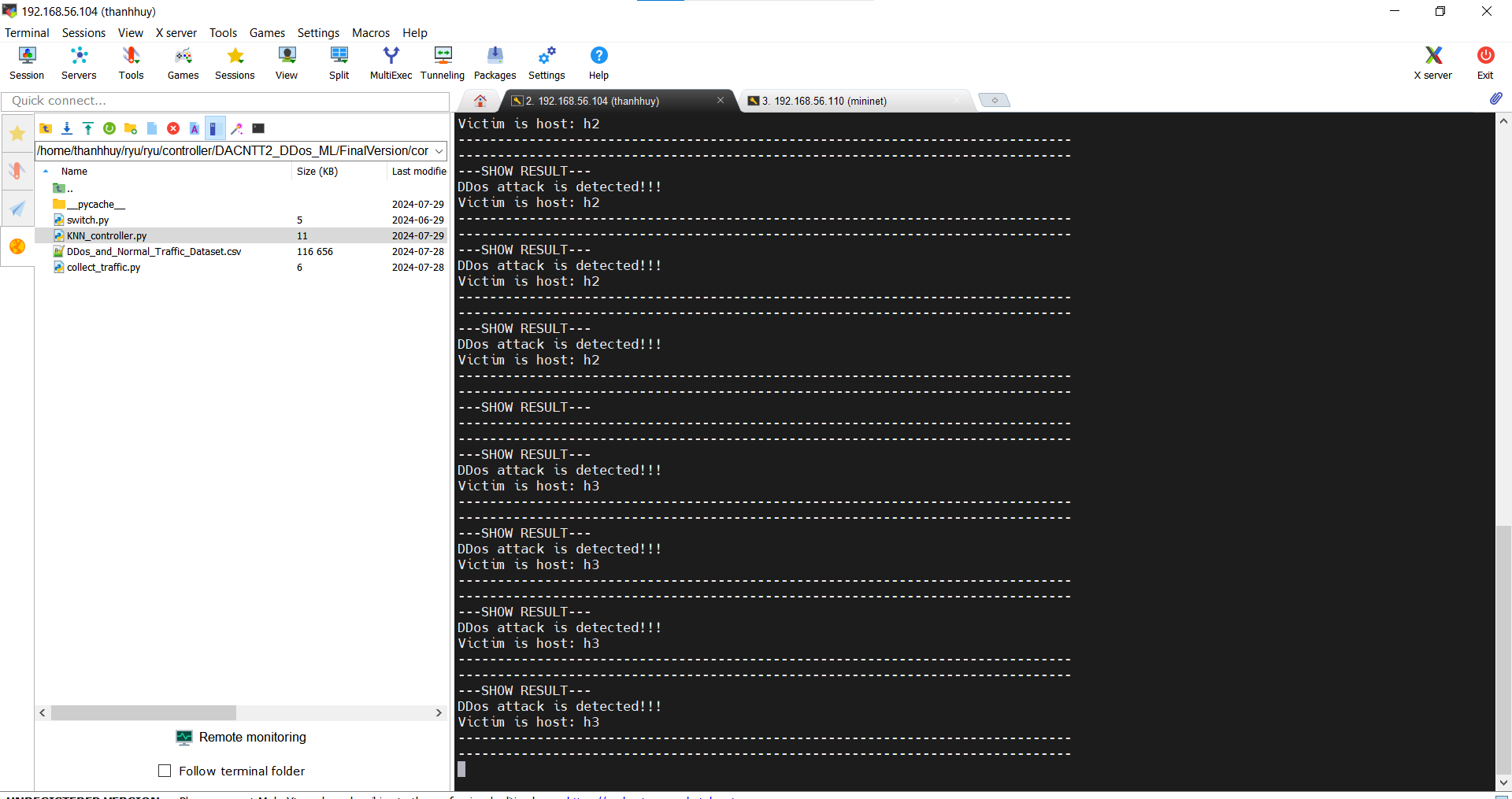


* + 1. Thực hiện tấn công UDP Flood tới h3

hping3 -2 -V -d 120 -w 64 -p 80 --rand-source –flood 192.168.0.3

Biểu đồ bên dưới hiển thị các traffic UDP đã được tới host h2 trong khi thời gian lệnh UDP Flood được thực thi 

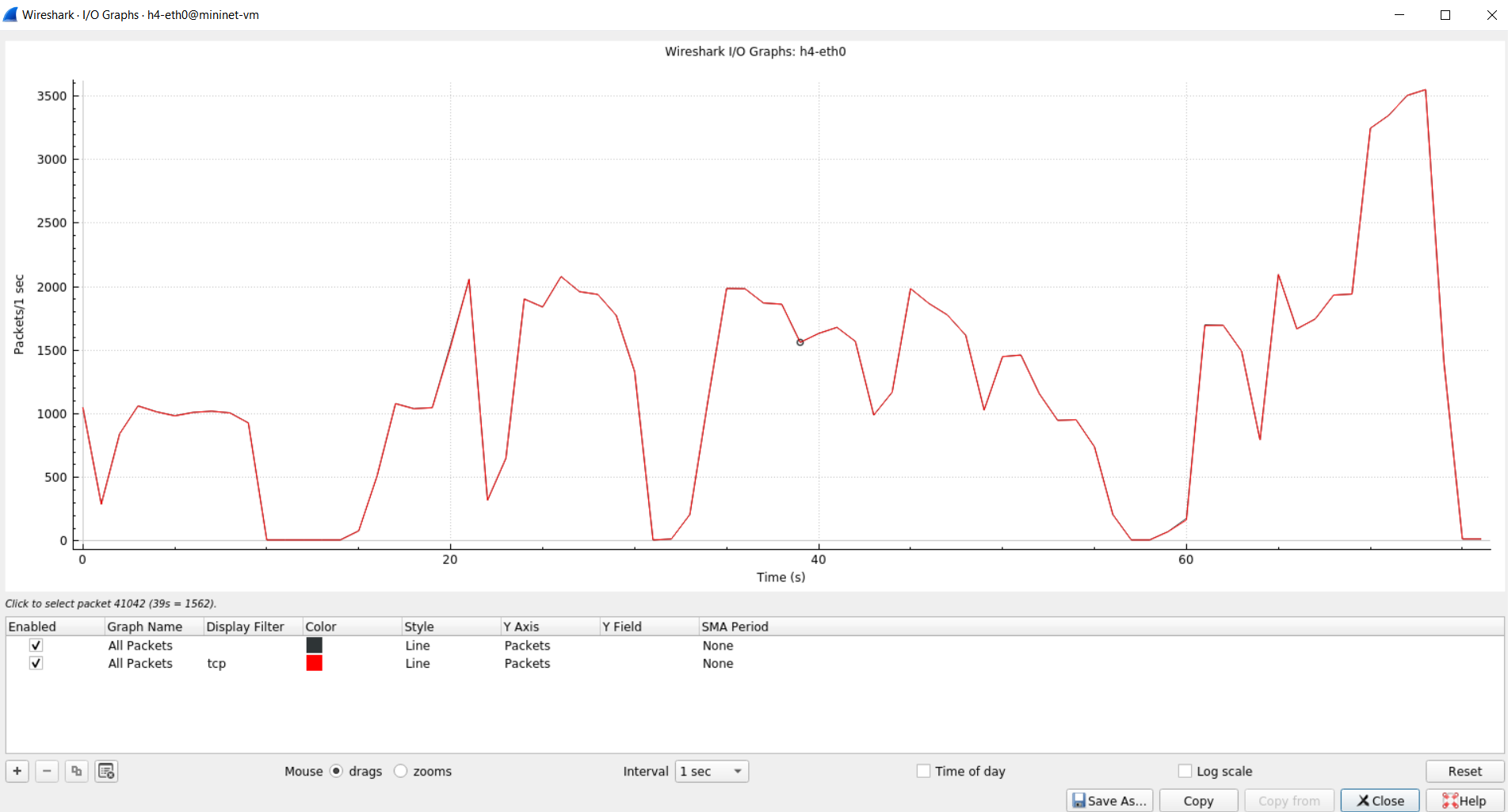
Như kết quả dự đoán hiển thị trên hình dưới, Model KNN xây dựng trong ryu-controller đã dự đoán chính xác host h3 đang bị tấn công



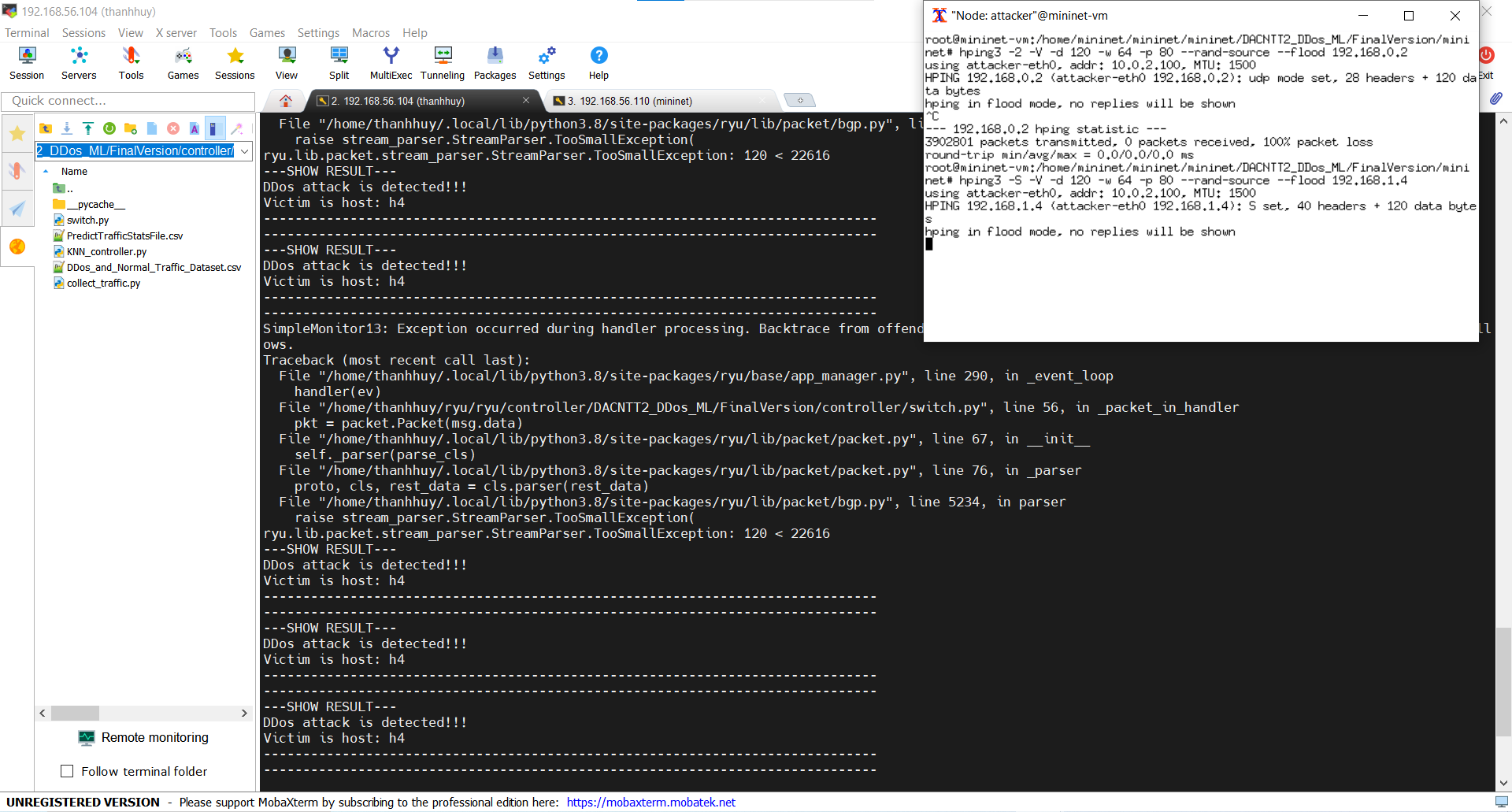
* + 1. Thực hiện tấn công TCP-SYN Flood tới h4

hping3 -S -V -d 120 -w 64 -p 80 --rand-source –flood 192.168.1.4

Biểu đồ bên dưới hiển thị các traffic TCP-SYN đã được tới host h4 trong khi thời gian lệnh TCP-SYN Flood được thực thi



Như kết quả dự đoán hiển thị trên hình dưới, Model KNN xây dựng trong ryu-controller đã dự đoán chính xác host h4 đang bị tấn công

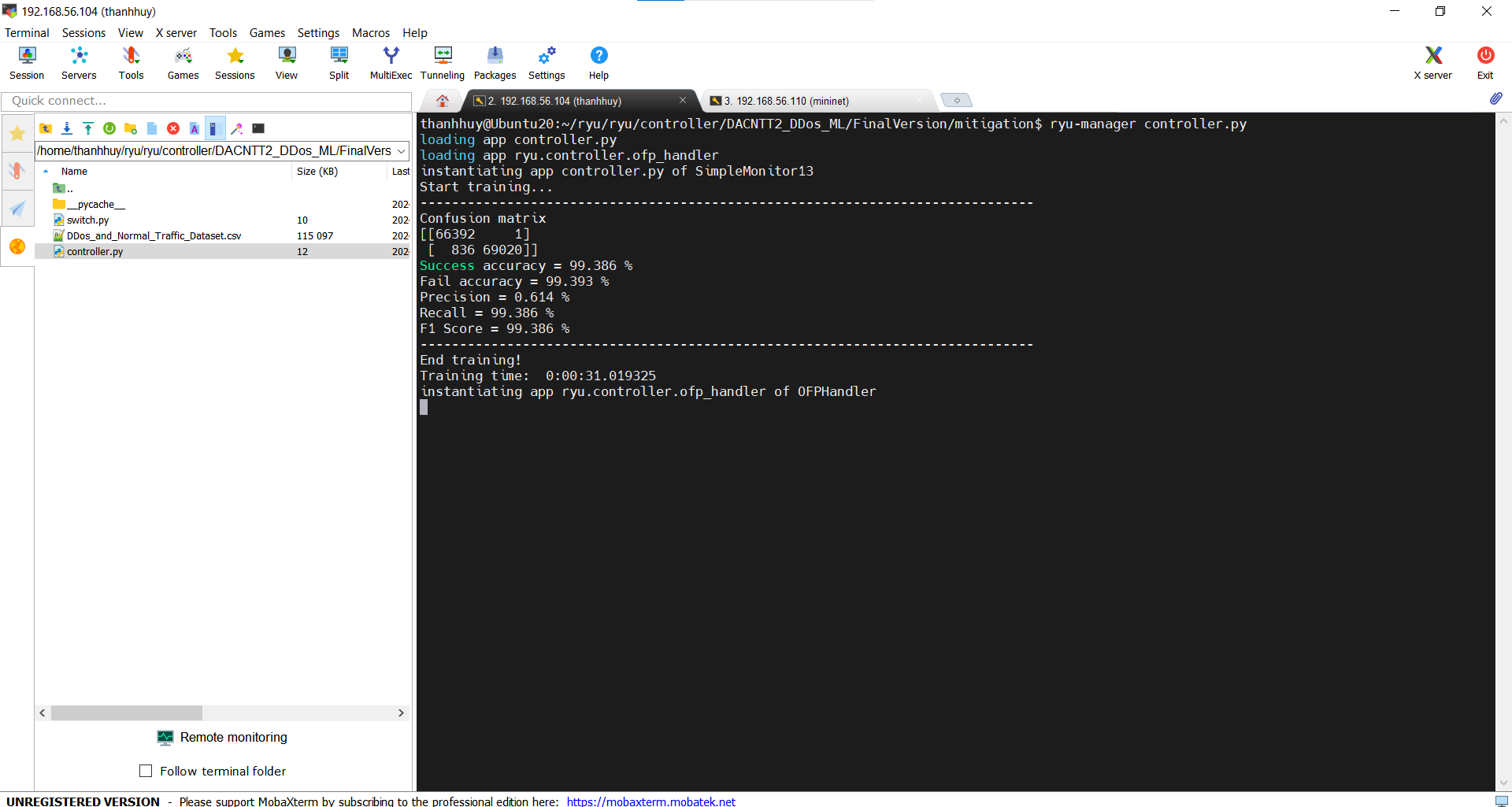


1. **Mitigation**
   1. Trên RYU

Cd tới path migitation

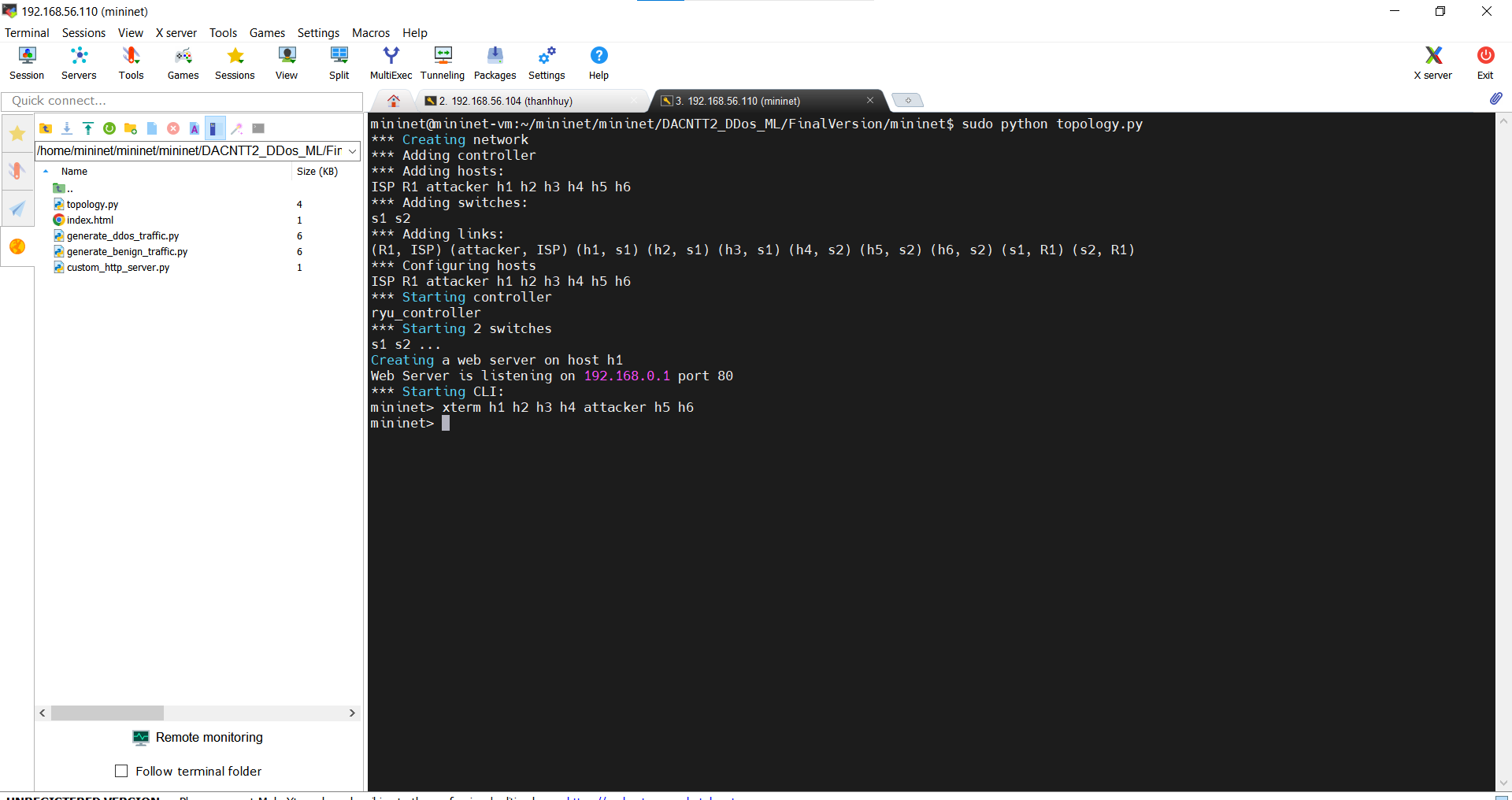
ryu-manager controller.py

Hình bên dưới hiển thị đã chạy thành công controller đã tích hợp KNN model và mitigation model block port khi phát hiện host bị tấn công



* 1. Trên Mininet VM

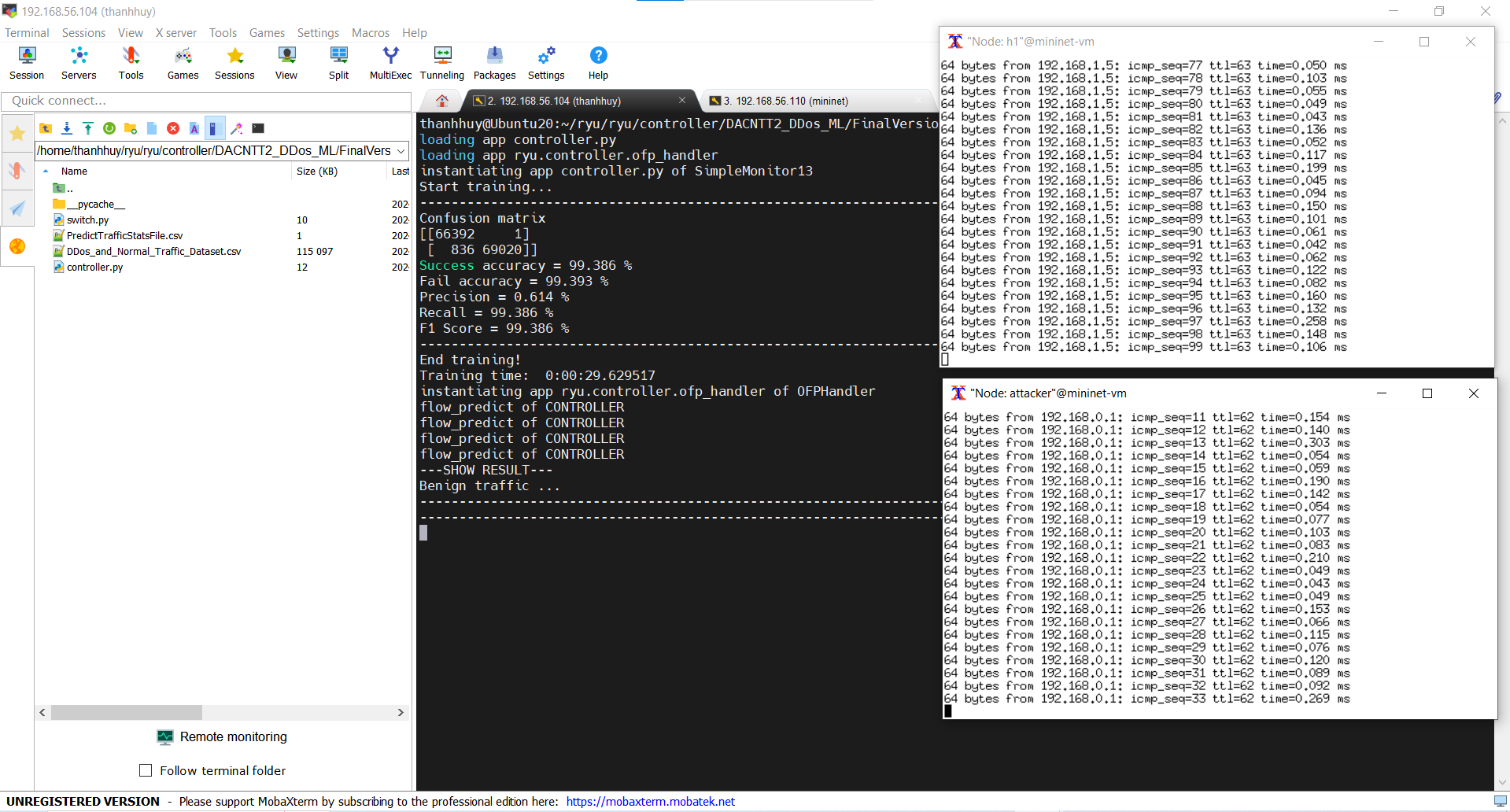
sudo python topology.py



Chạy lại các lệnh tấn công ở phần 4 và kiểm tra sau khi phát hiện bị tấn công DDos thì port của host đó bị block.

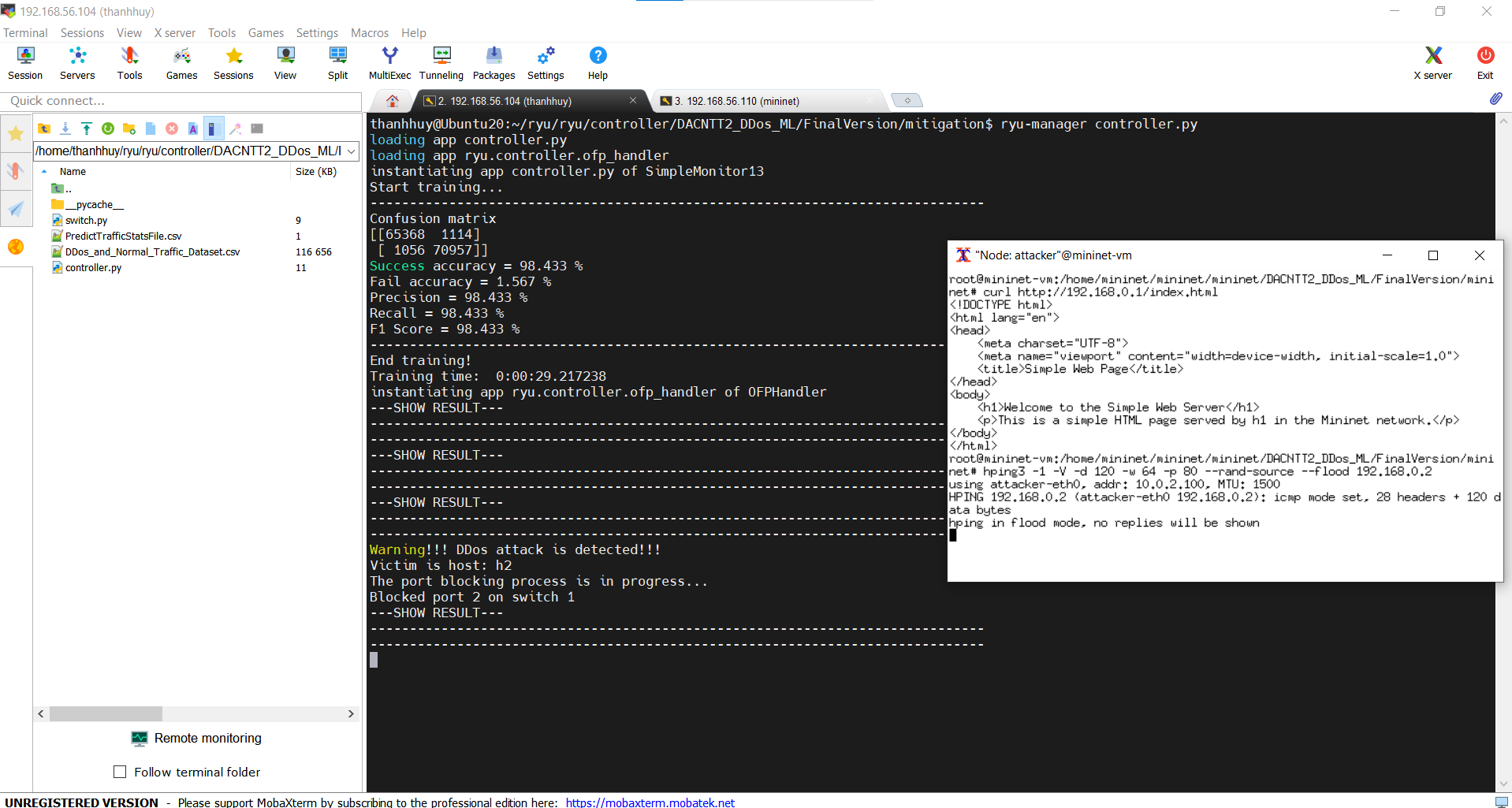
* 1. Test các traffic bình thường trong topology mạng

Hình bên dưới thể hiện giải thuật dự đoán được các traffic bình thường



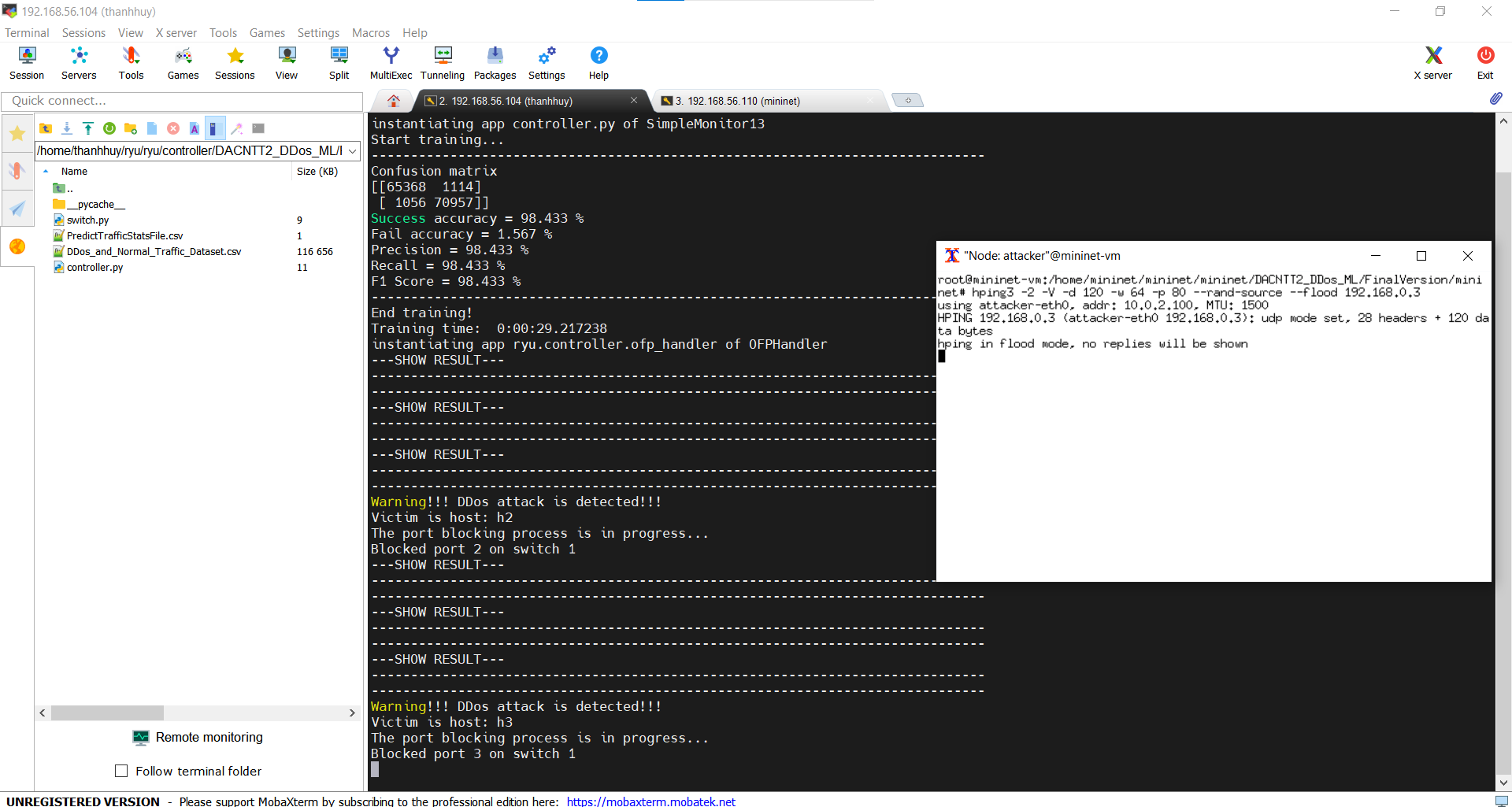
* 1. Thực nghiệm tấn công và ngăn chặn block port khi phát hiện DDos
     1. HTTP Flood (Chưa làm được)
     2. ICMP Flood tới host h2

Hình dưới cho thấy hệ thống đã phát hiện được host h2 bị tấn công và tự động block port đi vào của h1 trên switch 1



* + 1. Thực hiện tấn công UDP Flood tới host h3

Hình dưới cho thấy hệ thống đã phát hiện được host h3 bị tấn công và tự động block port đi vào của h3 trên switch 1



* + 1. Thực hiện tấn công TCP-SYN Flood tới h4

Hình dưới cho thấy hệ thống đã phát hiện được host h4 bị tấn công và tự động block port đi vào của h4 trên switch 2

