**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KHÓA**

**MÔN: AN TOÀN BẢO MẬT THÔNG TIN**

**Đề tài: ỨNG DỤNG GANS TẠO MẪU THỬ DDos**

Nhóm sinh viên thực hiện:

1. Nguyễn Giang Thanh Phương-MSHV: 230201050

2. Phạm Xuân Vĩnh Hà – MSHV:

GVHD: Nguyễn Tấn Cầm

Tp. Hồ Chí Minh, 07/2024

**THÔNG TIN CHUNG**

|  |  |  |
| --- | --- | --- |
| Họ và tên | Nguyễn Giang Thanh Phương | Phạm Xuân Vĩnh Hà |
| MSHV | 230201050 |  |
| Email | Phuongngt.18@grad.uit.edu.vn | Hapxv.18@grad.uit.edu.vn |
| Ngành học | Công Nghệ Thông Tin | Công Nghệ Thông Tin |
| Tự đánh giá điểm | …/10 | …/10 |
| Hình cá nhân |  | A person in a suit and tie |

# **MỤC LỤC**

[**MỤC LỤC** 3](#_Toc174987322)

[**Tóm Tắt** 5](#_Toc174987323)

[Chương 1: Tổng Quan 6](#_Toc174987324)

[1.1 Giới Thiệu về DDos 6](#_Toc174987325)

[1.1.1 Tổng quan về DDos 6](#_Toc174987326)

[1.1.2 Các Loại Tấn Công DDoS 6](#_Toc174987327)

[1.1.3 Mục Tiêu của Tấn Công DDoS 6](#_Toc174987328)

[1.1.4 Phương Pháp Phòng Chống DDoS 7](#_Toc174987329)

[1.1.5 Ví Dụ về Tấn Công DDoS 7](#_Toc174987330)

[1.2 Giới Thiệu về ICMP\_FLOOD 7](#_Toc174987331)

[1.2.1 Tổng quan về ICMP\_FLOOD 7](#_Toc174987332)

[1.3 Tổng Kết Chương 9](#_Toc174987333)

[Chương 2: Tồng Quan về GAN 10](#_Toc174987334)

[2.1 Giới Thiệu về GANs 10](#_Toc174987335)

[2.2 Kiến trúc của GANS 10](#_Toc174987336)

[2.2.1 Kiến trúc của Generator 10](#_Toc174987337)

[2.2.2 Kiến trúc của Discriminator 14](#_Toc174987338)

[2.2.3 Hàm Loss 17](#_Toc174987339)

[2.3 Đánh giá mô hình 17](#_Toc174987340)

[2.3.1 Gán nhãn 18](#_Toc174987341)

[2.3.2 Dự đoán 18](#_Toc174987342)

[2.3.3 Kết hợp dự đoán và nhãn 18](#_Toc174987343)

[2.3.4 Chuyển đổi dự đoán thành nhãn nhị phân: 18](#_Toc174987344)

[5.5. Tính toán các chỉ số đánh giá 19](#_Toc174987345)

[5.6 Tính các chỉ số đánh giá 19](#_Toc174987346)

[2.4 Tổng Kết Chương 20](#_Toc174987347)

[Chương 3: Thực nghiệm ứng dung GANS tạo mẫu thử ICMP\_FLOOD (DDos) 21](#_Toc174987348)

[3.1 Thu Thập Dataset 21](#_Toc174987349)

[3.1.1 Nguồn Dataset 21](#_Toc174987350)

[3.1.2 Quá trình thu thập dữ liệu 22](#_Toc174987351)

[3.1.3 Các thách thức và cách giải quyết 22](#_Toc174987352)

[3.2 Tiền xử lý Dataset 22](#_Toc174987353)

[3.3 Phân loại và lọc dữ liệu 23](#_Toc174987354)

[3.4 Chuyển đổi định dạng dữ liệu 23](#_Toc174987355)

[3.5 Xây Dựng Mô hình GANS 25](#_Toc174987356)

[3.5.1 Xây Dựng Generator 25](#_Toc174987357)

[3.5.2 Xây Dựng Discriminator 25](#_Toc174987358)

[3.5.3 Cấu Hình Các Tham số 25](#_Toc174987359)

[3.5.4 Khởi tạo Generator và Discriminator 25](#_Toc174987360)

[3.5.5 Xây Dựng Hàm LOSS 26](#_Toc174987361)

[3.5.6 Tạo hàm train GANS 26](#_Toc174987362)

[3.6 Huấn luyện GANS Tạo mẫu thử DDos 26](#_Toc174987363)

[3.6.1 Huấn luyện mô hình GANS 26](#_Toc174987364)

[3.7 Lưu mẫu thử DDos vao csv và file pcap mới 27](#_Toc174987365)

[3.8 Đánh Gía Mẫu Thử 28](#_Toc174987366)

[3.8.1 Đánh Gía dựa trên dataset từ CIC-2023 28](#_Toc174987367)

[3.8.2 Đánh Giá dựa trên dataset đã tích hợp mẫu thử 31](#_Toc174987368)

[3.8.3 Nhận Xét 32](#_Toc174987369)

[Chương 4: Kết luận và hướng phát triển 33](#_Toc174987370)

[4.1 Kết Kuận 33](#_Toc174987371)

[4.1.1 Hiệu quả của GANs trong việc tạo mẫu thử ICMP flood: 33](#_Toc174987372)

[4.1.2 Nhược điểm và nguyên nhân tiềm ẩn 33](#_Toc174987373)

[4.2 Hướng Phát Triển Trong Tương Lai 33](#_Toc174987374)

[4.2.1. Nâng cao chất lượng data 33](#_Toc174987375)

[4.2.2. Cải Thiện Kiến Trúc GANs 33](#_Toc174987376)

[4.2.3. Đánh giá và kiểm thử mô hình 34](#_Toc174987377)

[4.2.4. Phát triển thêm tính năng 34](#_Toc174987378)

[Tài Liệu Tham Khảo 35](#_Toc174987379)

# **Tóm Tắt**

Các cuộc tấn công DDoS làm cạn kiệt băng thông mạng và tài nguyên máy của hệ thống. Trong đó, ICMP Flood là một trong những cách tấn công từ chối dịch vụ phổ biến, gây ra thiệt hại lớn cho các hệ thống mạng và dịch vụ trực tuyến. Việc phát hiện gặp nhiều hạn chế như độ chính xác thấp, khả năng thích ứng kém với các loại tấn công mới gọi là tấn công với lưu lượng tấn công đối nghịch. Vấn đề này đặt ra nhiều câu hỏi cần có câu trả lời như phương pháp nào có thể phát hiện và phòng chống cuộc tấn công mới, phương pháp mới sẽ sử dụng dữ liệu được thu thập từ đâu, xây dựng tập huấn luyện ra sao? Nhiều đề tài đã nghiên cứu nhằm giải quyết nhưng vẫn còn nhiều hạn chế cần khắc phục. Để tìm hướng giải quyết, thông qua đề tài "Tạo Mẫu Thử ICMP Flood Bằng GANs" tập trung vào việc xây dựng một mô hình GANs để tạo ra các packet ICMP Echo Request nhưng tính chất tương đồng như gói tin từ các cuộc tấn công thật. Qua đó, đề tài không chỉ cung cấp một bộ dữ liệu mẫu thử ICMP Flood để sử dụng trong việc kiểm tra và cải thiện hệ thống phòng thủ chống lại tấn công DDoS mà còn là một công cụ hữu ích cho việc kiểm tra và cải thiện các giải pháp an toàn bảo mật mạng khác đồng thời còn góp phần trong nghiên cứu và ứng dụng Deep learning vào lĩnh vực an toàn

# Chương 1: Tổng Quan

## Giới Thiệu về DDos

### 1.1.1 Tổng quan về DDos

DDoS (Distributed Denial of Service) là một loại tấn công từ chối dịch vụ phân tán, trong đó kẻ tấn công sử dụng nhiều thiết bị bị nhiễm để gửi một lượng lớn các yêu cầu hoặc dữ liệu đến một hệ thống mục tiêu, nhằm làm quá tải hệ thống đó và làm gián đoạn dịch vụ. Dưới đây là tổng quan về DDoS, bao gồm các loại tấn công, cơ chế hoạt động, mục tiêu và phương pháp phòng chống.

### 1.1.2 Các Loại Tấn Công DDoS

* Tấn Công Băng Thông (Volumetric Attack): Kẻ tấn công gửi một lượng lớn dữ liệu hoặc yêu cầu đến mục tiêu để làm ngập băng thông của mạng. Ví dụ: UDP Flood, ICMP Flood.
* Tấn Công Giao Thức (Protocol Attack): Tấn công này nhắm vào các điểm yếu trong các giao thức mạng để làm gián đoạn dịch vụ. Ví dụ: SYN Flood, Ping of Death.
* Tấn Công Ứng Dụng (Application Layer Attack): Tấn công này nhắm vào các ứng dụng web và dịch vụ trực tuyến bằng cách gửi các yêu cầu hợp pháp với tần suất cao. Ví dụ: HTTP Flood, Slowloris.

### 1.1.3 Mục Tiêu của Tấn Công DDoS

* Trang Web và Dịch Vụ Trực Tuyến: Các trang web thương mại điện tử, dịch vụ tài chính, dịch vụ truyền thông xã hội và các dịch vụ trực tuyến khác thường là mục tiêu phổ biến.
* Cơ Quan Chính Phủ và Tổ Chức Quốc Tế: Các cơ quan chính phủ và tổ chức quốc tế có thể bị tấn công vì lý do chính trị hoặc nhằm làm gián đoạn dịch vụ công cộng.
* Hạ Tầng Mạng: Các nhà cung cấp dịch vụ internet (ISP), trung tâm dữ liệu, và các hạ tầng mạng khác cũng có thể là mục tiêu.

### 1.1.4 Phương Pháp Phòng Chống DDoS

* Sử Dụng Các Giải Pháp Chống DDoS: Các giải pháp chống DDoS từ các nhà cung cấp dịch vụ bảo mật mạng có thể giúp phát hiện và giảm thiểu các cuộc tấn công DDoS.
* Tăng Cường Hạ Tầng: Đầu tư vào băng thông lớn hơn, cân bằng tải, và các công nghệ mạng khác để làm giảm tác động của tấn công DDoS.
* Cấu Hình Bảo Mật Hợp Lý: Cấu hình tường lửa, bộ lọc gói tin, và các thiết bị mạng khác để chặn các yêu cầu không hợp lệ và giảm tải cho hệ thống.
* Giám Sát và Phát Hiện Sớm: Sử dụng các hệ thống giám sát mạng để phát hiện sớm các dấu hiệu của tấn công DDoS và phản ứng kịp thời.

### 1.1.5 Ví Dụ về Tấn Công DDoS

* Tấn Công Dyn (2016): Một cuộc tấn công DDoS lớn nhắm vào nhà cung cấp dịch vụ DNS Dyn vào năm 2016 đã làm gián đoạn nhiều dịch vụ trực tuyến lớn như Twitter, Netflix, và Reddit.
* Tấn Công Mirai Botnet: Mirai là một botnet gồm các thiết bị IoT bị nhiễm phần mềm độc hại, đã được sử dụng để thực hiện nhiều cuộc tấn công DDoS lớn.

## Giới Thiệu về ICMP\_FLOOD

### 1.2.1 Tổng quan về ICMP\_FLOOD

ICMP Flood là một dạng tấn công DDoS sử dụng các gói tin ICMP để làm nghẽn băng thông và tài nguyên của máy chủ hoặc mạng. Tấn công này được thực hiện bằng cách gửi một lượng lớn các gói tin ICMP (thường là gói tin Echo Request) tới mục tiêu, làm cho mục tiêu không thể xử lý được tất cả các yêu cầu và dẫn đến gián đoạn dịch vụ.

1.2.2 Tác động của ICMP\_FLOOD

Tấn công ICMP Flood có thể gây gián đoạn dịch vụ mạng, làm giảm hiệu suất hệ thống, và gây ảnh hưởng nghiêm trọng đến hoạt động của tổ chức. Khi máy chủ hoặc mạng bị tấn công bởi ICMP Flood, nó có thể trở nên chậm chạp hoặc không thể truy cập được, gây ra sự bất tiện lớn và thiệt hại tài chính.

1.2.3 các đặc tính chính của ICMP Flood

* Lưu lượng lớn các gói tin ICMP Echo Request
* Tần suất cao: Các gói tin ICMP được gửi đi với tần suất rất cao, thường lên đến hàng ngàn hoặc hàng triệu gói tin mỗi giây.
* Kích thước gói tin: Các gói tin ICMP Echo Request có thể có kích thước lớn để tăng tải trọng trên hệ thống mục tiêu.
* Sự hiện diện của các gói tin Echo Request mà không có Echo Reply
* Echo Request mà không có Echo Reply: Trong tấn công ICMP Flood, các gói tin ICMP Echo Request được gửi đi nhưng không có sự trả lời (Echo Reply) từ phía hệ thống mục tiêu. Điều này có thể do hệ thống quá tải hoặc do kẻ tấn công cố tình không quan tâm đến phản hồi.
* Sử dụng địa chỉ IP giả mạo
* Địa chỉ IP giả mạo (IP Spoofing): Kẻ tấn công thường sử dụng địa chỉ IP giả mạo để gửi các gói tin ICMP Echo Request, nhằm che giấu danh tính và làm tăng độ khó trong việc phát hiện và ngăn chặn tấn công.
* TTL thấp
* TTL thấp: Thời gian sống (TTL) của các gói tin ICMP thường được đặt ở mức thấp trong các cuộc tấn công ICMP Flood. Điều này có thể nhằm mục đích làm giảm khả năng phát hiện và theo dõi nguồn gốc của các gói tin tấn công.
* Tải CPU và băng thông mạng cao
* Tải CPU cao: Hệ thống mục tiêu phải xử lý một lượng lớn các gói tin ICMP, dẫn đến tải CPU cao và giảm hiệu suất hệ thống.
* Băng thông mạng cao: Tấn công ICMP Flood tiêu tốn một lượng lớn băng thông mạng, có thể dẫn đến tắc nghẽn mạng và làm gián đoạn các dịch vụ khác.
* Các đặc điểm thống kê
* Phân phối kích thước gói tin: Kích thước của các gói tin ICMP trong tấn công ICMP Flood thường có một phân phối đặc trưng, có thể khác biệt so với lưu lượng ICMP hợp pháp.
* Thời gian giữa các gói tin (Inter-packet time): Khoảng thời gian giữa các gói tin ICMP Echo Request trong tấn công thường rất ngắn và có thể có một mô hình cụ thể.
* Dễ dàng phát hiện qua các công cụ giám sát mạng
* Công cụ giám sát mạng: Các công cụ giám sát mạng như Wireshark, Tcpdump, và các hệ thống phát hiện xâm nhập (IDS) có thể dễ dàng phát hiện các dấu hiệu của tấn công ICMP Flood dựa trên lưu lượng và đặc tính của các gói tin ICMP.
* Tấn công nhắm mục tiêu rộng
* Mục tiêu rộng: ICMP Flood có thể nhắm mục tiêu vào các thiết bị mạng, máy chủ web, máy chủ ứng dụng, và bất kỳ hệ thống nào khác chấp nhận các gói tin ICMP.

1.2.4 Ví dụ thực tế

Một ví dụ nổi tiếng về tấn công ICMP Flood là cuộc tấn công vào dịch vụ Yahoo! vào năm 2000, khi hàng loạt các gói tin ICMP được gửi đến các máy chủ của Yahoo!, khiến dịch vụ này bị gián đoạn trong nhiều giờ. Các tổ chức khác như Amazon và CNN cũng đã từng trở thành mục tiêu của các cuộc tấn công DDoS bao gồm ICMP Flood.

## Tổng Kết Chương

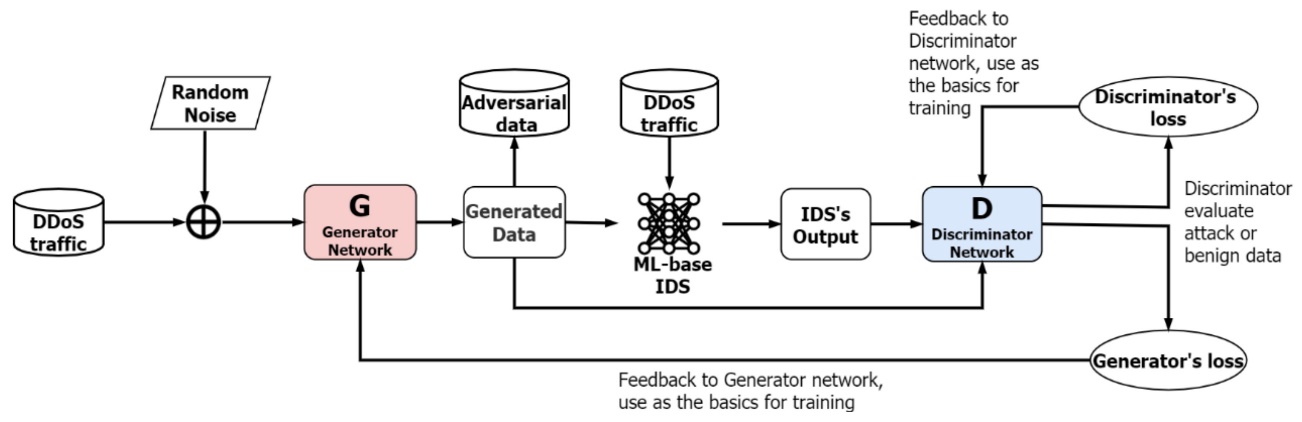
DDoS là một loại tấn công từ chối dịch vụ phân tán, trong đó kẻ tấn công sử dụng nhiều thiết bị để gửi một lượng lớn yêu cầu hoặc dữ liệu đến một hệ thống mục tiêu nhằm làm quá tải và làm gián đoạn dịch vụ. Có nhiều loại tấn công DDoS khác nhau, mỗi loại nhắm vào các điểm yếu khác nhau của hệ thống mạng và dịch vụ. Việc phòng chống DDoS đòi hỏi sự kết hợp của nhiều biện pháp bảo mật, từ việc sử dụng các giải pháp chống DDoS đến tăng cường hạ tầng và cấu hình bảo mật hợp lý.

# Chương 2: Tồng Quan về GAN

## 2.1 Giới Thiệu về GANs

GANs (Generative Adversarial Networks) là một loại mạng nơ-ron học sâu được giới thiệu bởi Ian Goodfellow vào năm 2014. GANs bao gồm hai mô hình, Generator và Discriminator, hoạt động theo cách đối kháng để tạo ra dữ liệu giả lập giống thật.

Trong mô hình GANS, có 02 thành phần chính là Generator tạo ra hàng nhái và Discriminator cố gắng phân biệt đâu là dữ liệu thật.

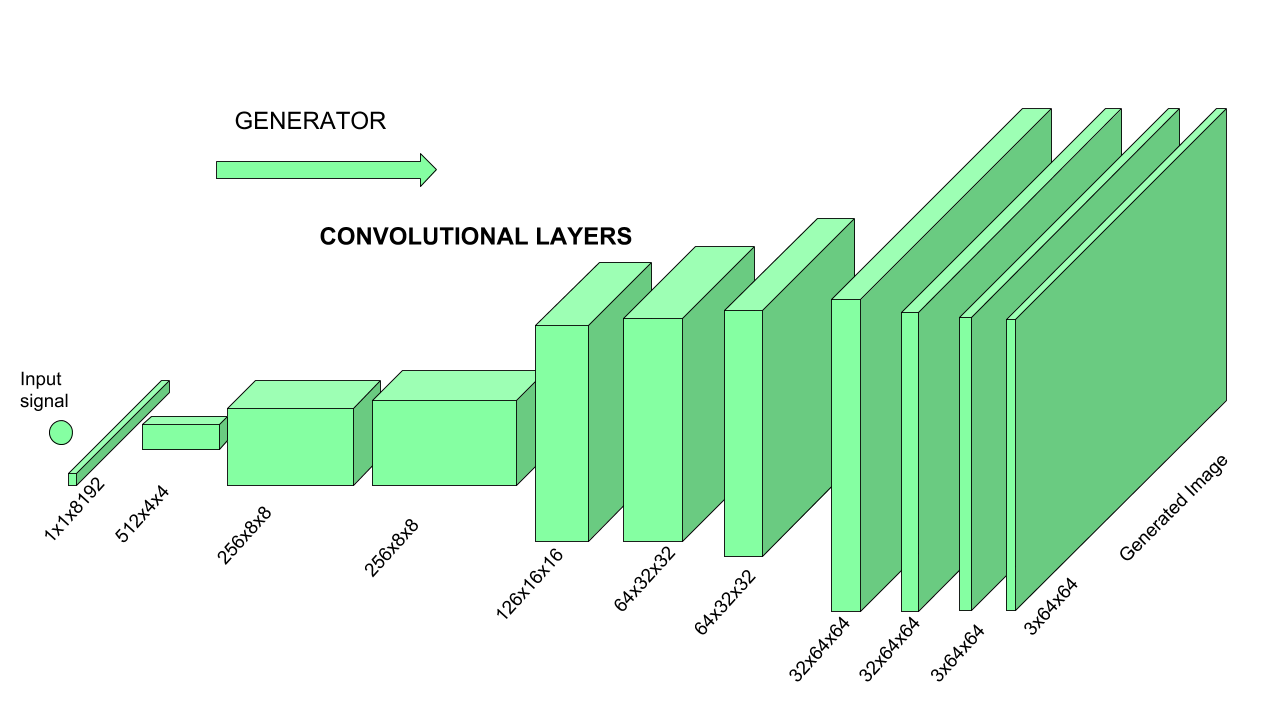


Hình 1: Mô hình Gans trong phòng chống DDoS

## 2.2 Kiến trúc của GANS

### 2.2.1 Kiến trúc của Generator

Generator là một mạng nơ-ron có nhiệm vụ tạo ra các mẫu dữ liệu giả từ một nguồn nhiễu ngẫu nhiên. Mô hình Generator thường bao gồm các lớp Dense (Fully Connected) và các lớp kích hoạt phi tuyến như LeakyReLU để đảm bảo tính đa dạng và phức tạp của dữ liệu giả.



Hình 2: Kiến Trúc Generator

#### 2.2.1.1. Các Lớp của Generator

* Input Layer: Đầu vào của Generator là một vector nhiễu ngẫu nhiên (noise vector), thường có kích thước 100.
* Hidden Layers: Các lớp ẩn sử dụng các lớp Dense kết hợp với các lớp Batch Normalization và các hàm kích hoạt LeakyReLU. Batch Normalization giúp ổn định quá trình huấn luyện và LeakyReLU giúp khắc phục vấn đề chết ReLU (dead ReLU problem).
* Output Layer: Lớp đầu ra của Generator là một vector có cùng kích thước với dữ liệu thật, giúp tạo ra các mẫu dữ liệu giả giống thật.

#### 2.2.1.2 Các Thành Phần Chính của Generator

Layer 1: Dense (128 units) + LeakyReLU(α=0.2) + BatchNorm

**Dense Layer (Fully Connected Layer)**:

**Công thức toán học:**

Trong đó:

* là vector đầu ra của lớp Dense thứ hai.
* là ma trận trọng số của lớp Dense thứ hai.
* là vector đầu ra của lớp Batch Normalization trước đó.
* là vector bias của lớp Dense thứ nhất.

**LeakyReLU Activation:**

**Công thức toán học:**

=

Trong đó:

* ​ là vector đầu ra của hàm kích hoạt LeakyReLU.
* ​ là giá trị đầu vào.
* α là hệ số nhân cho giá trị âm của ​ khi ​ nhỏ hơn 0.

**Batch Normalization:**

**+** Công thức chuẩn hóa đầu ra

Trong đó:

* là vector đầu ra đã được chuẩn hóa.
* là đầu vào của lớp trước khi chuẩn hóa.
* là giá trị trung bình của batch đầu vào.
* là phương sai của batch đầu vào.
* là một giá trị rất nhỏ để tránh chia cho 0.

**+** Công thức điều chỉnh đầu ra:

β

Trong đó:

* là vector chuẩn hóa.
* và là là các tham số huấn luyện được sử dụng để mở rộng và dịch chuyển dữ liệu đã chuẩn hóa.
* là vector đầu ra đã được chuẩn hóa.

**Layer 2: Dense (256 units) + LeakyReLU(α=0.2) + BatchNorm**

**Dense Layer (Fully Connected Layer)**:

**Công thức toán học**:

Trong đó:

* là vector đầu ra của lớp Dense thứ hai.
* là ma trận trọng số của lớp Dense thứ hai.
* là vector đầu ra của lớp Batch Normalization trước đó.
* là vector bias của lớp Dense thứ hai.

**LeakyReLU Activation**:

**Công thức toán học**:

=

Trong đó:

* ​ là vector đầu ra của hàm kích hoạt LeakyReLU.
* ​ là giá trị đầu vào.
* α là hệ số nhân cho giá trị âm của ​ khi ​ nhỏ hơn 0.

**Batch Normalization**

**Công thức toán học:**

**+** Công thức chuẩn hóa đầu ra

Trong đó:

* là vector đầu ra đã được chuẩn hóa.
* là đầu vào của lớp trước khi chuẩn hóa.
* là giá trị trung bình của batch đầu vào.
* là phương sai của batch đầu vào.
* là một giá trị rất nhỏ để tránh chia cho 0.

**+** Công thức điều chỉnh đầu ra:

β

Trong đó:

* là vector chuẩn hóa.
* và là là các tham số huấn luyện được sử dụng để mở rộng và dịch chuyển dữ liệu đã chuẩn hóa.
* là vector đầu ra đã được chuẩn hóa.

**Layer 3: Dense(output\_dim) + Tanh**

**Dense Layer (Fully Connected Layer)**:

**Công thức toán học**

Trong đó:

* h3 là vector đầu ra của lớp Dense cuối cùng.
* W3 là ma trận trọng số của lớp Dense cuối cùng.
* là vector đầu ra của lớp Batch Normalization trước đó.
* b3 là vector bias của lớp Dense cuối cùng**.**

**Tanh Activation:**

**Công thức toán học**

Trong đó:

* là vector đầu ra của hàm kích hoạt Tanh
* Hàm TandHàm Tanh chuyển đổi giá trị của **h3** vào khoảng [-1, 1], giúp đảm bảo các giá trị đầu ra không bị bùng nổ và giữ trong một phạm vi hợp lý.

#### 2.2.1.3 Các Kiểu Dữ Liệu Hỗ Trợ

Generator thường làm việc với các vector nhiễu ngẫu nhiên được lấy mẫu từ phân phối Gaussian hoặc Uniform. Đầu ra của Generator là các vector có cùng kích thước và định dạng với dữ liệu thật, chẳng hạn như hình ảnh, âm thanh, hoặc dữ liệu văn bản.

#### 2.2.1.4 Phương Thức của Generator

**Khởi Tạo Vector Nhiễu Ngẫu Nhiên**

Generator nhận đầu vào là một vector nhiễu ngẫu nhiên 𝑧, thường được lấy mẫu từ một phân phối Gaussian hoặc Uniform. Vector nhiễu này đại diện cho các đặc trưng tiềm ẩn mà Generator sẽ sử dụng để tạo ra dữ liệu giả.

**Chuyển Đổi Vector Nhiễu Thành Dữ Liệu Giả**

Vector nhiễu 𝑧 được đưa qua các lớp Dense và các hàm kích hoạt trong Generator để chuyển đổi thành dữ liệu giả 𝑥. Mỗi lớp Dense thực hiện phép biến đổi tuyến tính trên vector đầu vào, sau đó áp dụng các hàm kích hoạt phi tuyến để học các đặc trưng phức tạp.

**Tạo Dữ Liệu Giả**

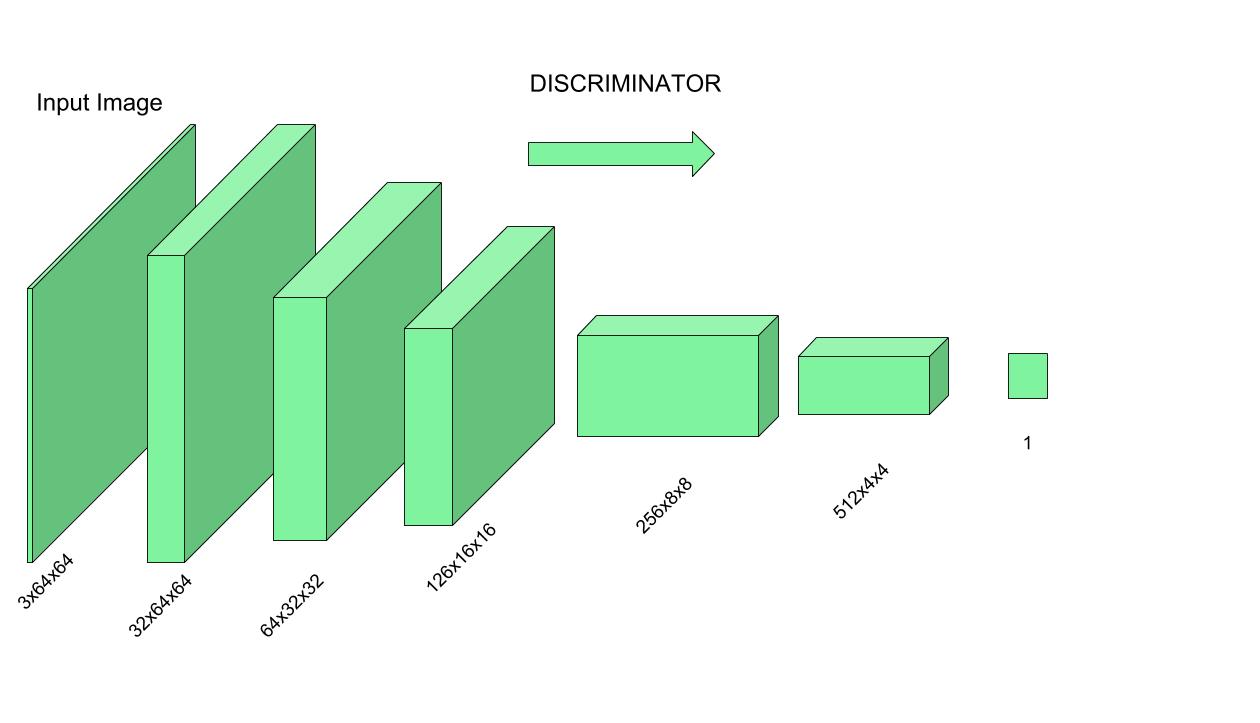
Lớp đầu ra của Generator sử dụng hàm kích hoạt Tanh để tạo ra dữ liệu giả x có cùng kích thước và định dạng với dữ liệu thật. Mục tiêu của Generator là tạo ra dữ liệu giả có độ chân thực cao, mà Discriminator không thể phân biệt được với dữ liệu thật.

**Cập Nhật Trọng Số**

Trọng số của Generator được cập nhật thông qua quá trình huấn luyện đối kháng với Discriminator. Generator cố gắng tối ưu hóa hàm loss của mình để tạo ra dữ liệu giả có thể lừa được Discriminator.

### 2.2.2 Kiến trúc của Discriminator

Discriminator là mạng nơ-ron sâu có nhiệm vụ phân biệt giữa dữ liệu thật và dữ liệu giả. Cấu trúc của Discriminator thường bao gồm nhiều lớp Dense và các hàm kích hoạt phi tuyến để đảm bảo tính chính xác trong việc phân biệt dữ liệu.



Hình 3: Kiến Trúc DISCRIMINATOR

#### 2.2.2.1. Các Lớp của Discriminator

* Input Layer: Đầu vào của Discriminator là một vector dữ liệu, có thể là dữ liệu thật hoặc dữ liệu giả từ Generator.
* Hidden Layers: Các lớp ẩn sử dụng các lớp Dense kết hợp với các hàm kích hoạt LeakyReLU để tăng cường khả năng phân biệt của Discriminator.
* Output Layer: Lớp đầu ra của Discriminator là một giá trị duy nhất, thường được kích hoạt bởi hàm Sigmoid, biểu thị xác suất rằng đầu vào là dữ liệu thật.

#### 2.2.2.2 Các Thành Phần Chính của Discriminator

**Dense Layers (Fully Connected Layers)**

Các lớp Dense trong Discriminator kết nối tất cả các neuron từ lớp trước với tất cả các neuron của lớp sau. Điều này giúp mô hình học được các đặc trưng phức tạp từ dữ liệu đầu vào.

**Công thức toán học:**

Trong đó:

* là đầu ra của lớp Dense hiện tại.
* là ma trận trọng số của lớp Dense hiện tại.
* là đầu ra của lớp trước đó.
* là bias của lớp Dense hiện tại.

**LeakyReLU Activation**

Hàm kích hoạt LeakyReLU được sử dụng để đảm bảo tính phi tuyến và tránh vấn đề vanishing gradient.

**Công thức toán học:**

=

Trong đó

* là một hệ số nhỏ (thường là 0.2).

**Sigmoid Activation**

Hàm kích hoạt Sigmoid được sử dụng ở lớp đầu ra để đảm bảo giá trị đầu ra nằm trong khoảng [0, 1].

**Công thức toán học:**

Trong đó:

* Sigmoid(x) biến đổi giá trị đầu vào thành output nằm trong khoảng từ 0 đến 1.
* Khi x lớn, tức là x→ +∞ thì hàm Sigmoid(x) tiến gần đến 1.
* Khi x nhỏ, tức là x→ −∞ thì hàm Sigmoid(x) tiến gần đến 0.
* Khi x=0, hàm Sigmoid(x) có giá trị bằng 0.5.

#### 2.2.2.3 Các Kiểu Dữ Liệu Hỗ Trợ

Discriminator làm việc với các vector dữ liệu, có thể là dữ liệu thật từ tập huấn luyện hoặc dữ liệu giả từ Generator. Đầu ra của Discriminator là một giá trị xác suất, đại diện cho khả năng phân loại của mô hình.

#### 2.2.2.4. Phương Thức Hoạt Động của Discriminator

Trong quá trình huấn luyện, Discriminator được huấn luyện để phân biệt giữa dữ liệu thật và dữ liệu giả. Cụ thể:

Nhận dữ liệu đầu vào: Discriminator nhận một batch dữ liệu thật từ tập huấn luyện và một batch dữ liệu giả từ Generator.

Tính toán giá trị xác suất: Discriminator tính toán xác suất rằng mỗi mẫu dữ liệu đầu vào là thật.

Cập nhật trọng số: Trọng số của Discriminator được cập nhật dựa trên hàm loss, phản ánh sự khác biệt giữa dự đoán của Discriminator và nhãn thực tế của dữ liệu đầu vào.

### 2.2.3 Hàm Loss

Hàm loss đóng vai trò quan trọng trong quá trình huấn luyện của cả Generator và Discriminator. Các hàm loss phổ biến bao gồm Binary Cross-Entropy Loss và các biến thể của nó.

#### 2.2.3.1 Hàm Loss của Discriminator (D\_loss)

Discriminator cố gắng phân biệt giữa dữ liệu thật và dữ liệu giả. Hàm loss của Discriminator được tính dựa trên Binary Cross-Entropy Loss, phản ánh sự khác biệt giữa dự đoán của Discriminator và nhãn thực tế của dữ liệu đầu vào.

**Công Thức Toán học:**

Trong đó:

#### 2.2.3.2 Hàm Loss của Generator (G\_loss)

Generator cố gắng tạo ra dữ liệu giả mà Discriminator không thể phân biệt được với dữ liệu thật. Hàm loss của Generator cũng được tính dựa trên Binary Cross-Entropy Loss, phản ánh khả năng của Generator trong việc lừa Discriminator.

**Công Thức Toán học:**

Trong đó:

## 2.3 Đánh giá mô hình

Đánh giá mô hình GAN bao gồm nhiều bước để đảm bảo các mẫu thử sinh ra là thực tế và mô hình GANs hoạt động tốt. Trong phần này, chúng tôi mô tả quá trình đánh giá, bao gồm các khía cạnh lý thuyết, công thức và ý nghĩa của chúng mà không đi sâu vào các triển khai mã cụ thể.

### 2.3.1 Gán nhãn

* Dữ liệu thật: Gán nhãn là 1, chỉ ra các gói tin ICMP flood.
* Dữ liệu sinh ra (mẫu thử): Gán nhãn là 0, chỉ ra lưu lượng bình thường hoặc không phải flood.

### 2.3.2 Dự đoán

Dự đoán (Predictions) là bước quan trọng trong việc đánh giá mô hình GAN. Trong bước này, mô hình Discriminator sẽ được sử dụng để dự đoán xem một mẫu là thật hay giả. Trong đó, Mô hình Discriminator sẽ dự đoán xác suất mỗi mẫu dữ liệu thuộc về lớp thật. Xác suất này là giá trị liên tục trong khoảng từ 0 đến 1.

* Các mẫu dữ liệu thật được đưa vào Discriminator và dự đoán xác suất là thật (gọi là real\_predictions).
* Các mẫu dữ liệu giả được đưa vào Discriminator và dự đoán xác suất là thật (gọi là fake\_predictions).

### 2.3.3 Kết hợp dự đoán và nhãn

Kết hợp các dự đoán và nhãn thực tế để tạo ra một tập dữ liệu toàn diện để đánh giá.

### 2.3.4 Chuyển đổi dự đoán thành nhãn nhị phân:

Ngưỡng được sử dụng để quyết định ranh giới giữa các dự đoán dương tính và âm tính.

**Công Thức Toán học:**

predicted\_labels=

* Thông thường, ngưỡng được đặt là 0.5. Nghĩa là nếu xác suất dự đoán lớn hơn 0.5, mẫu được phân loại là thật (nhãn 1), ngược lại nếu nhỏ hơn hoặc bằng 0.5, mẫu được phân loại là giả (nhãn 0).
* Ngưỡng thấp hơn (ví dụ, 0.3) có thể dẫn đến nhiều phân loại dương tính hơn, tăng độ nhạy nhưng có thể giảm độ chính xác do nhiều false positive hơn.
* Ngưỡng cao hơn (ví dụ, 0.7) có thể tăng độ chính xác nhưng giảm độ nhạy bằng cách phân loại ít trường hợp hơn là dương tính, dẫn đến nhiều false negative hơn.
* Ngưỡng 0.5 thường được sử dụng như một điểm cân bằng giữa độ chính xác và độ nhạy.

### 5.5. Tính toán các chỉ số đánh giá

Sử dụng các dự đoán và nhãn thực tế kết hợp, tính toán các chỉ số đánh giá chính:

* **True Positive (TP)**: Số lượng gói tin ICMP flood được dự đoán đúng.

**Công Thức Toán học:**

* **True Negative (TN)**: Số lượng gói tin bình thường được dự đoán đúng.

**Công Thức Toán học:**

* **False Positive (FP)**: Số lượng gói tin bình thường bị dự đoán sai là ICMP flood.

**Công Thức Toán học:**

* **False Negative (FN)**: Số lượng gói tin ICMP flood bị dự đoán sai là bình thường.

**Công Thức Toán học:**

### 5.6 Tính các chỉ số đánh giá

Từ các số đếm cơ bản này, chúng tôi suy ra các chỉ số đánh giá:

* **Độ chính xác (Accuracy)**: Độ chính xác đo lường tỷ lệ dự đoán đúng trên tổng số dự đoán**.**
* **Độ chính xác (Precision)**: Tỷ lệ dự đoán dương đúng trên tổng số dự đoán dương. Nó đo lường độ chính xác của các dự đoán dương.
* **Độ nhạy (Recall):** Tỷ lệ dự đoán dương đúng trên tổng số trường hợp dương thực tế. Nó đo lường khả năng tìm kiếm tất cả các trường hợp dương.
* **F1-Score:** Trung bình điều hòa của Precision và Recall. Nó cân bằng giữa hai chỉ số này, đặc biệt hữu ích khi dữ liệu không cân bằng.

## 2.4 Tổng Kết Chương

Generative Adversarial Networks (GANs) đã mở ra nhiều hướng nghiên cứu và ứng dụng mới trong lĩnh vực trí tuệ nhân tạo và học sâu. Với khả năng tạo ra dữ liệu giả có độ chân thực cao, GANs đã chứng minh được tiềm năng to lớn trong nhiều lĩnh vực khác nhau, từ tạo hình ảnh, chuyển đổi phong cách, nâng cao chất lượng hình ảnh, đến tạo văn bản và âm thanh. Quá trình huấn luyện đối kháng giữa Generator và Discriminator không chỉ giúp cải thiện chất lượng của dữ liệu giả mà còn cung cấp một phương pháp tiếp cận mạnh mẽ để giải quyết nhiều bài toán phức tạp trong học sâu.

# Chương 3: Thực nghiệm ứng dung GANS tạo mẫu thử ICMP\_FLOOD (DDos)

Để ứng dụng tạo mẫu thử ICMP FLOOD nhằm đánh giá mô hình GANs, chúng tôi xây dựng kịch bản như sau:

A diagram of data processing

Description automatically generated

BƯỚC 1: Thu thập dữ liệu từ bộ dữ liệu CIC IoT Dataset2023

BƯỚC 2: Sử dụng scapy để xử lý gói tin pcap, lọc gói tin và lưu file csv

BƯỚC 3: Tiền xử lý các cột trong file trong file csv về dạng số học

BƯỚC 4: Chuẩn hóa dữ liệu và chuyển đổi về cấu trúc pytorch tensor

BƯỚC 5: Tạo mô hình GANs xây dựng generator, discriminator

BƯỚC 6: Truyền các tham số, input, output, tốc độ học, batchsize vào mô hình

BƯỚC 7: Huấn luyện mô hình và tạo 100000 mẫu thử icmp\_flood

BƯỚC 8: Chuyển đổi các giá trị các cột về đúng định dạng ban đầu

BƯỚC 9: Cập nhật mẫu thử icmp\_flood vào file icmp\_flood gốc

BƯỚC 10: Đánh giá kết quả acurary, recall, precision, f1 score2 => Testset lần 1

BƯỚC 11: Retrain mô hình với Dataset đã cập nhât và tạo 10000 mẫu thử mới.

BƯỚC 12: Đánh giá Kết quả acurary, recall, precision, f1 score2 => Testset lần 2

BƯỚC 13: Nhận xét và ĐÁNH GIÁ MÔ HÌNH sau hai lần tạo mẫu (Testset lần 1 với Testset lần 2)

## 3.1 Thu Thập Dataset

### 3.1.1 Nguồn Dataset

Link Download: <http://205.174.165.80/IOTDataset/CIC_IOT_Dataset2023/Dataset/DDoS-ICMP_Flood.tar.gz>

Bộ dữ liệu CIC\_IOT\_Dataset2023 là một tập dữ liệu tấn công IoT để thúc đẩy phát triển các ứng dụng phân tích bảo mật trong các hoạt động IoT thực tế. Bộ dữ liệu bao gồm 33 cuộc tấn công được thực hiện trong một cấu trúc IoT bao gồm 105 thiết bị. Các cuộc tấn công này được phân loại thành bảy loại, cụ thể là DDoS, DoS, Recon, Web-based, Brute Force, Spoofing và Mirai.

### 3.1.2 Quá trình thu thập dữ liệu

Dữ liệu thu thập từ các cuộc tấn công icmp\_flood được lưu trữ trong các file PCAP. Các file PCAP này có các đặc trưng cần thiết như địa chỉ IP nguồn, địa chỉ IP đích, giao thức, kích thước gói tin….

Trong đề tài này, sử dụng Scapy để phân tích và xử lý dữ liệu, đảm bảo tính chính xác và đầy đủ của các thông tin thu thập.



### 3.1.3 Các thách thức và cách giải quyết

Các thách thức bao gồm độ lớn của dữ liệu, tính đa dạng của dữ liệu tấn công, và đảm bảo tính toàn vẹn và chính xác của dữ liệu. Các thách thức này được giải quyết bằng cách sử dụng các công cụ phân tích mạng như Wireshark và Scapy, giúp phân tích và xử lý dữ liệu một cách hiệu quả.

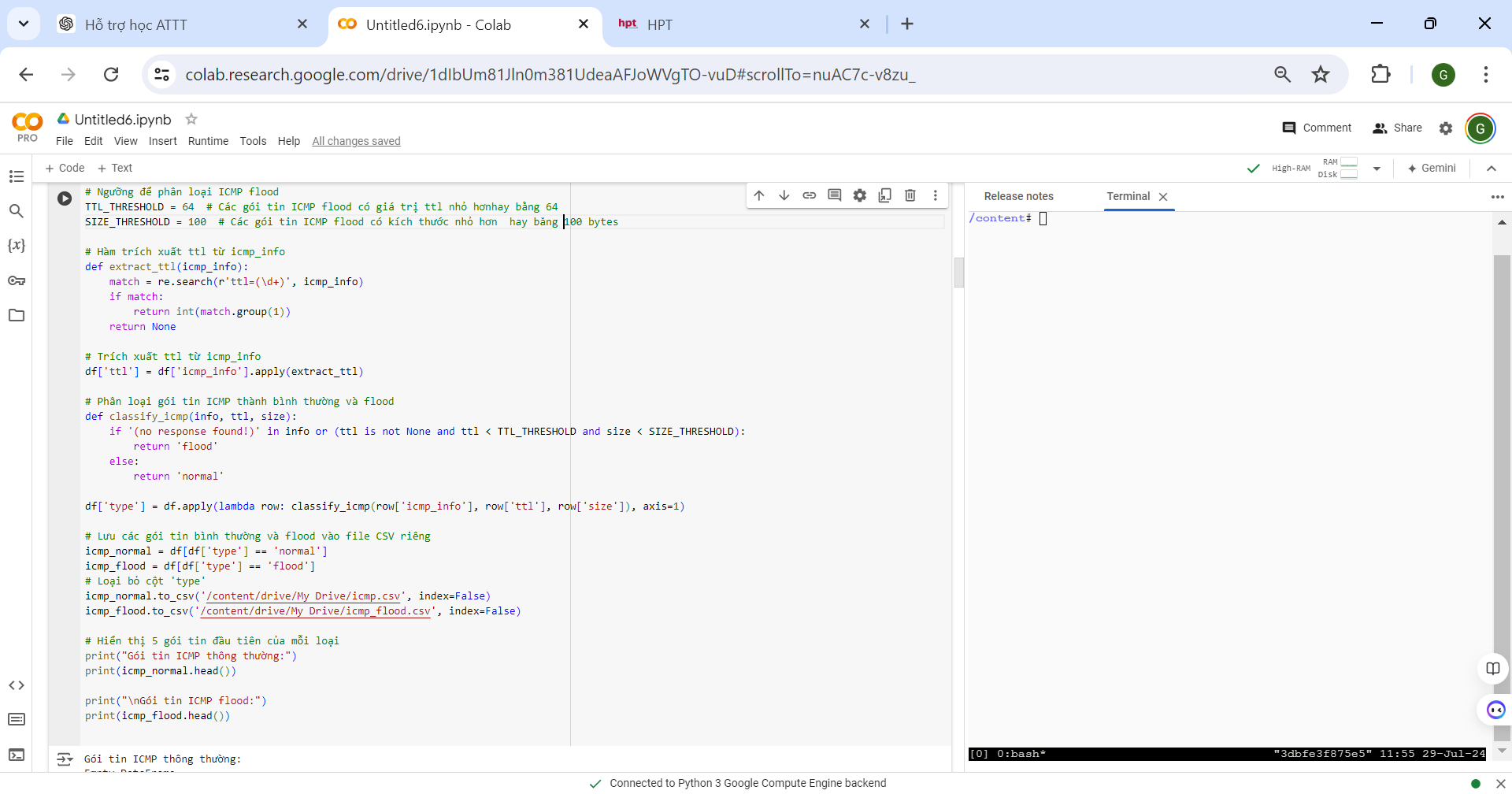
## 3.2 Tiền xử lý Dataset

Loại bỏ các gói tin không liên quan đến icmp\_flood và chỉ giữ lại các trường quan trọng như thời gian, địa chỉ IP nguồn, địa chỉ IP đích, giao thức, kích thước gói tin và thông tin ICMP. Sau đó, lưu vaofile outputfile.csv



### 3.3 Phân loại và lọc dữ liệu

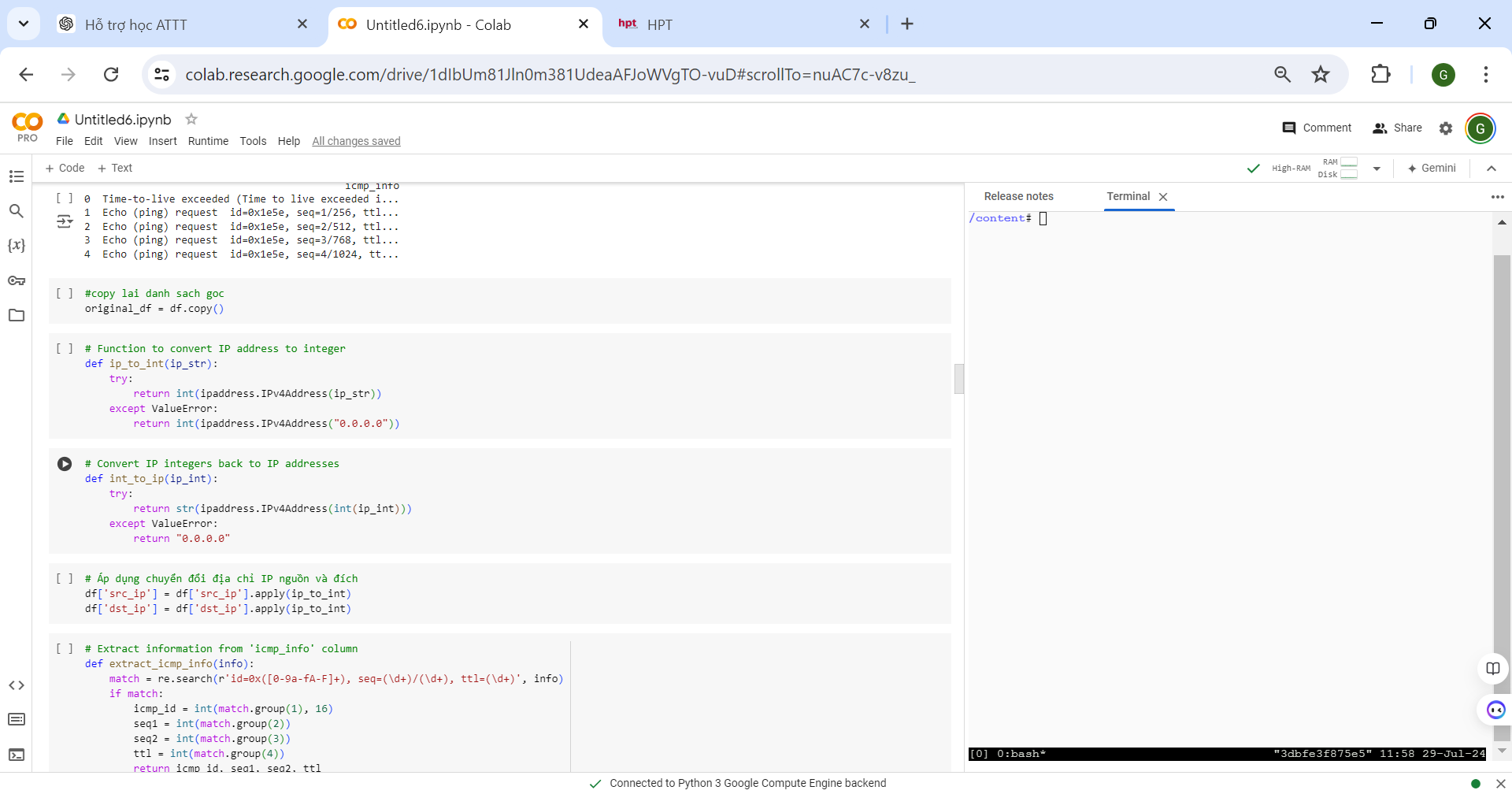
Phân loại gói tin ICMP thành bình thường và flood dựa trên các ngưỡng TTL và kích thước.



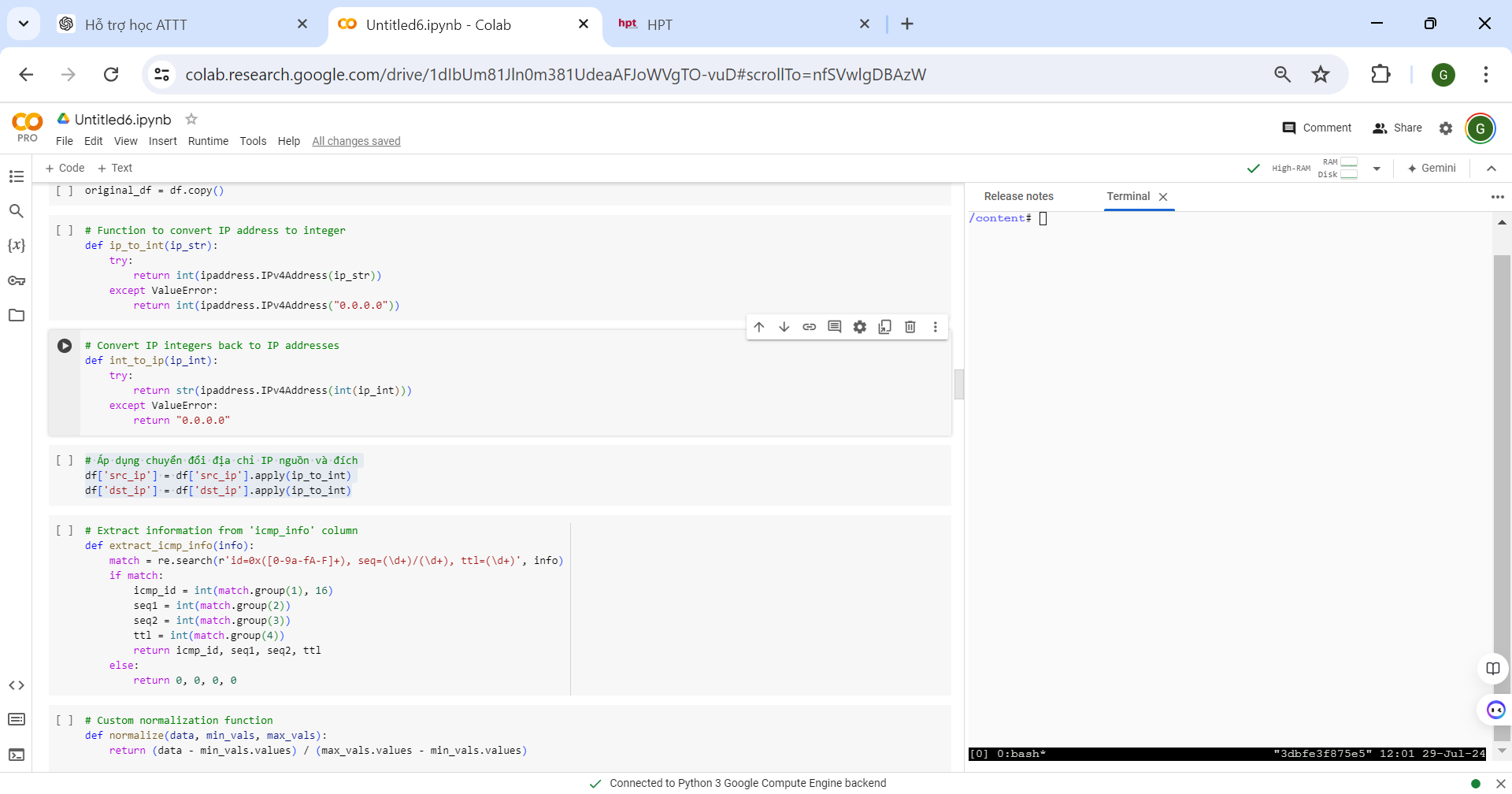
### 3.4 Chuyển đổi định dạng dữ liệu

Chuyển đổi địa chỉ IP từ chuỗi sang số và trích xuất thông tin từ cột ICMP Info, đảm bảo dữ liệu có định dạng phù hợp cho việc huấn luyện mô hình.

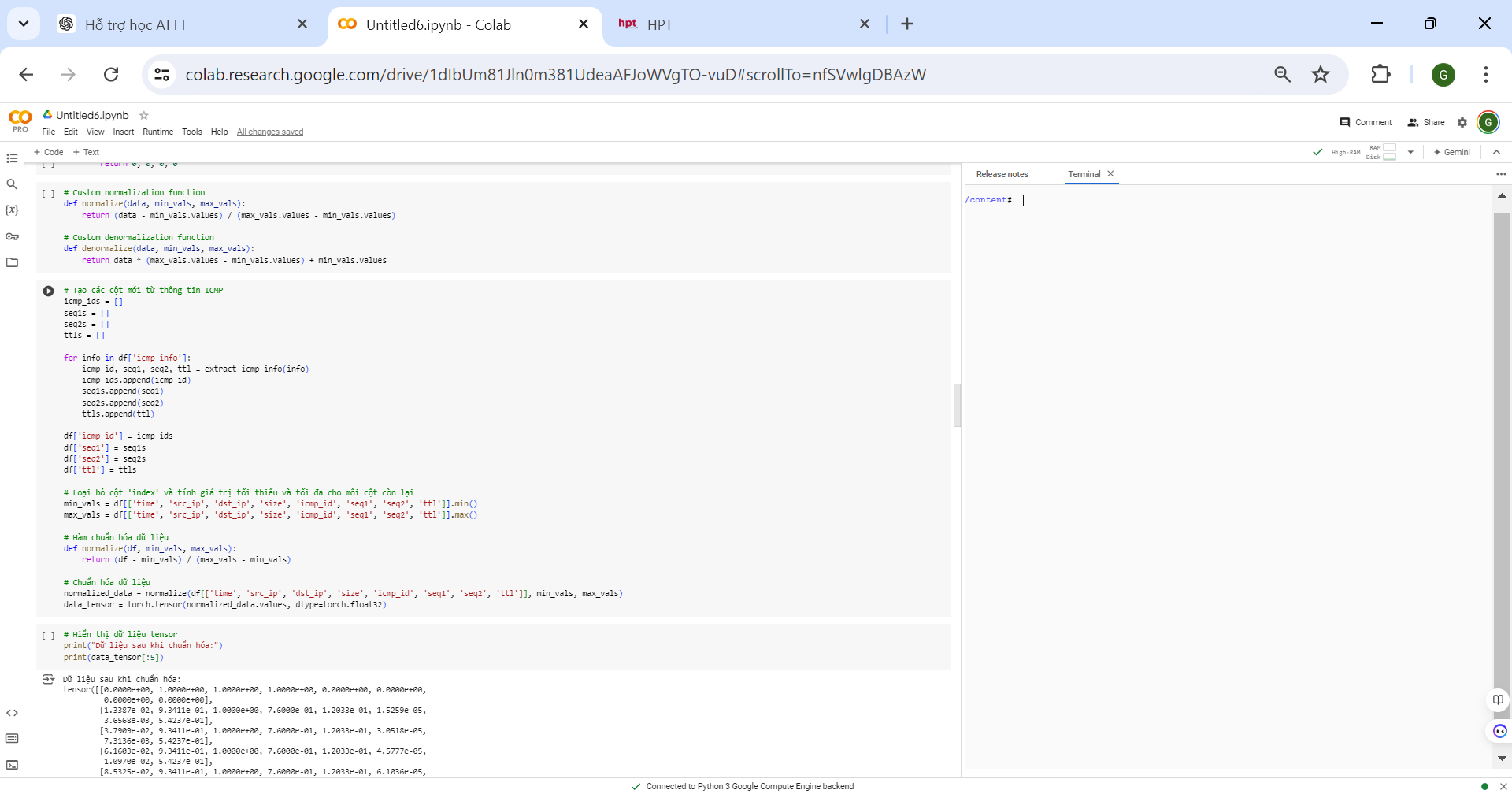
* Chuyển đổi ip từ string sang int



* Áp dụng chuyển đổi ip source\_add và dst



* Chuyển đổi Các cột còn lại

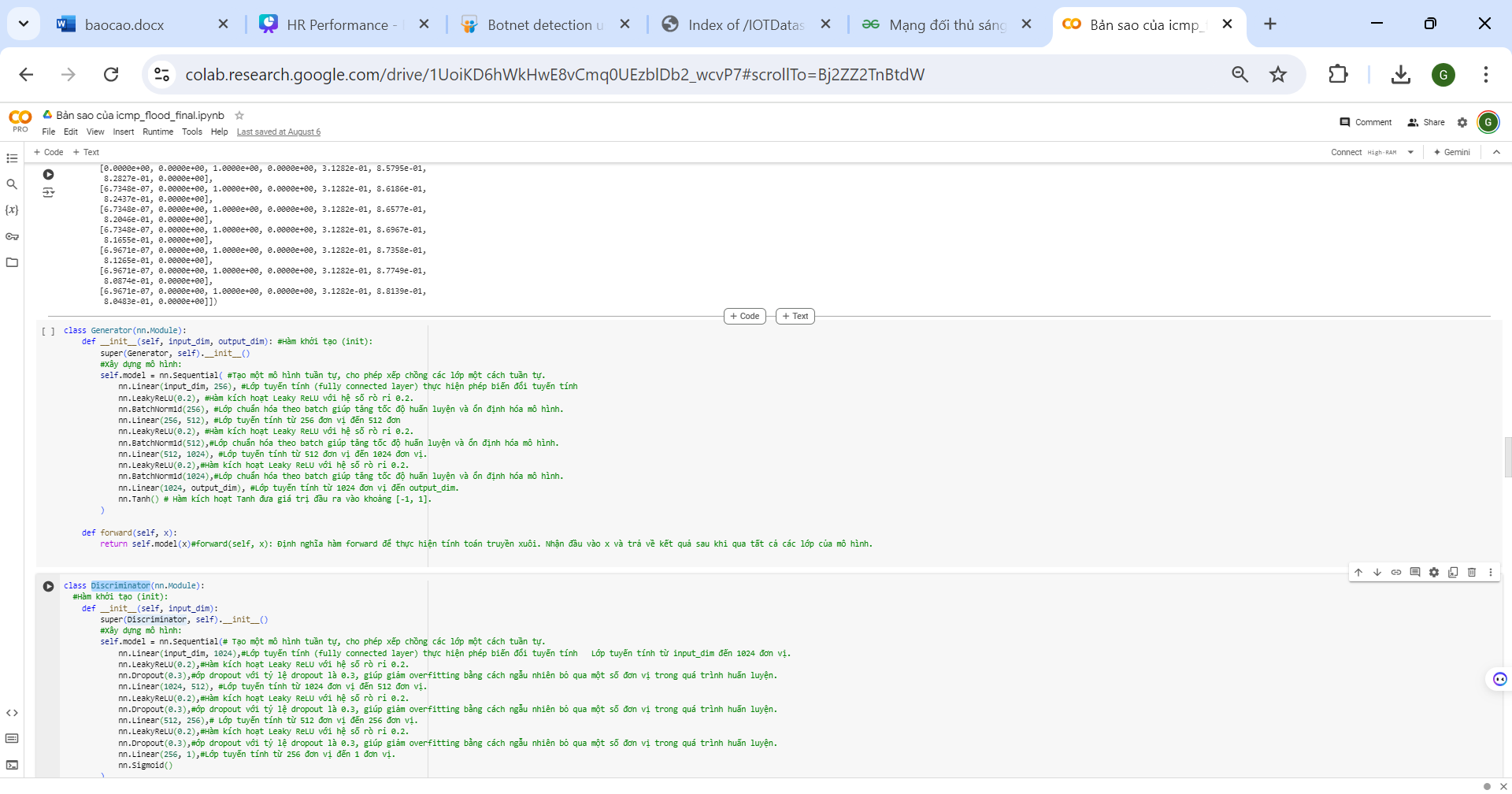


* Dữ liệu sau khi chuyển vào vector

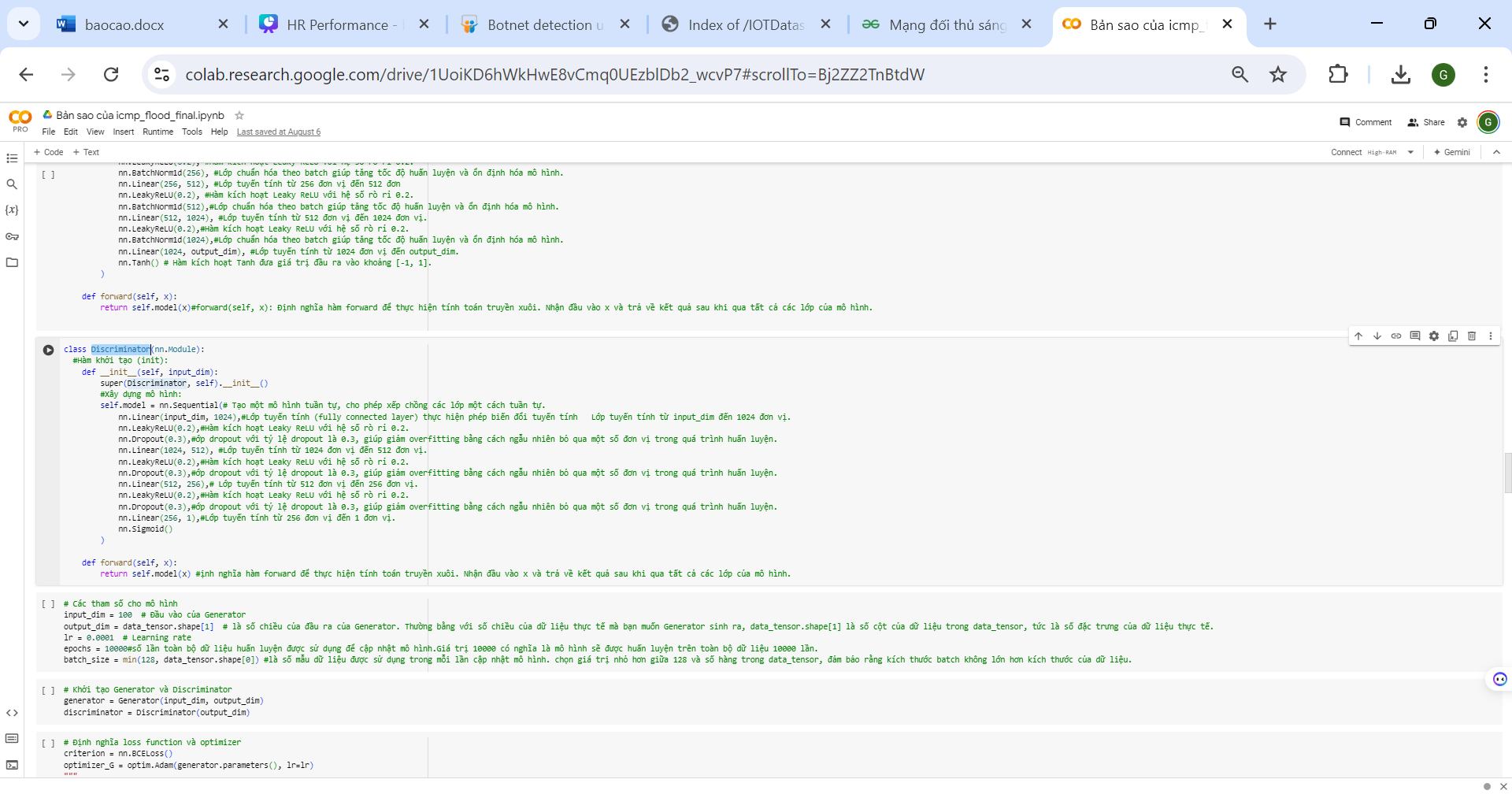


## 3.5 Xây Dựng Mô hình GANS

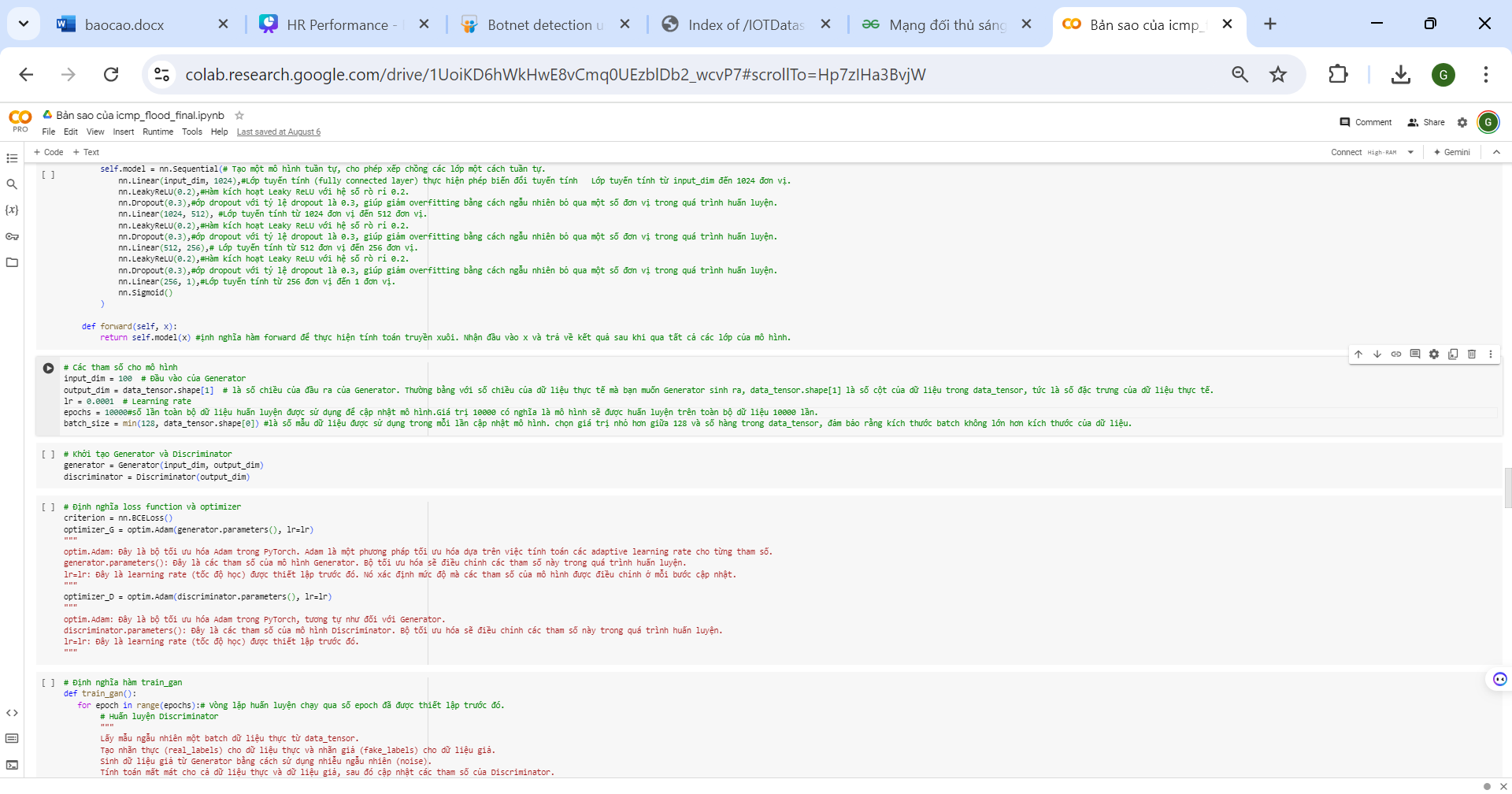
### 3.5.1 Xây Dựng Generator



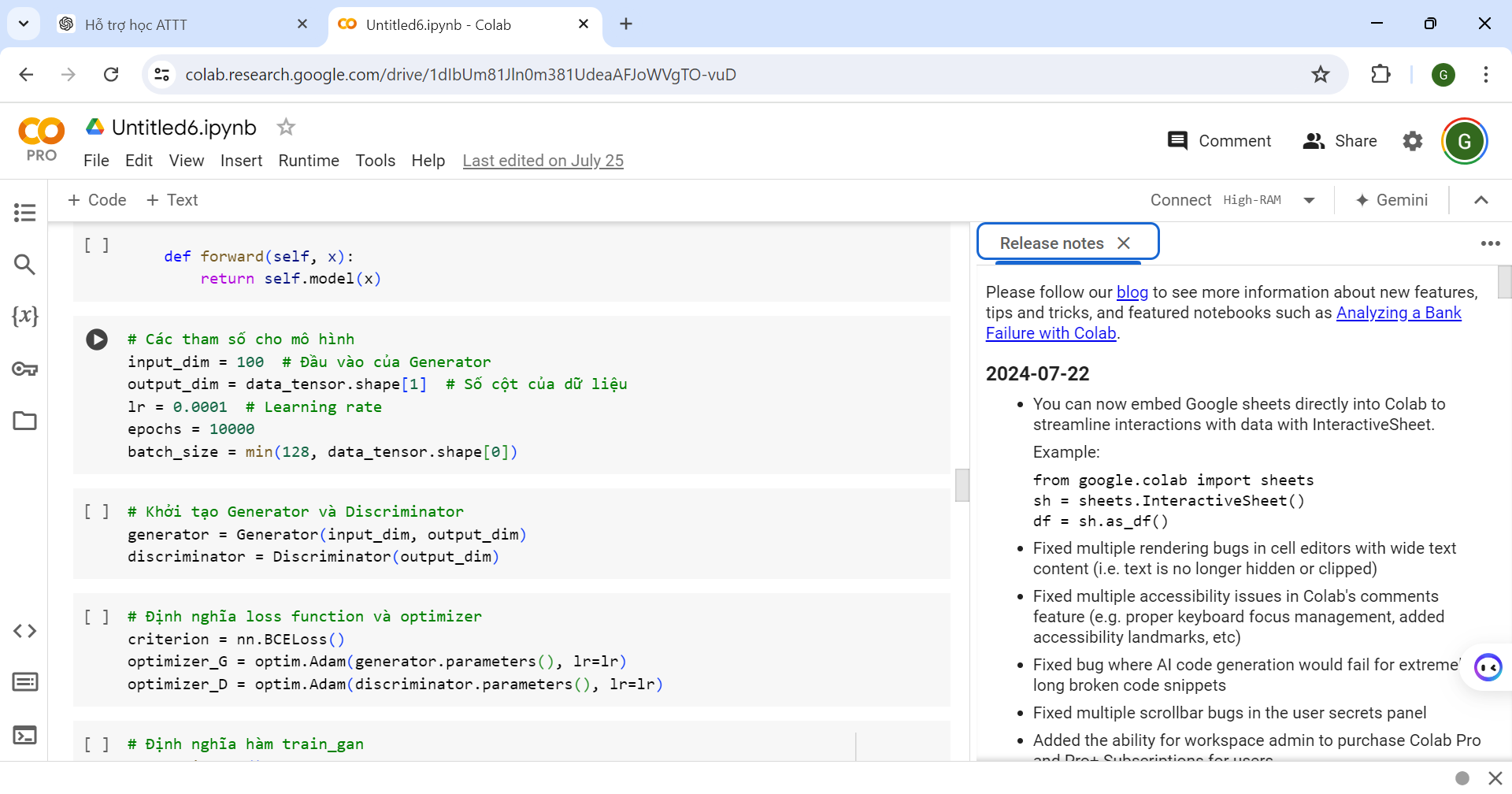
### 3.5.2 Xây Dựng Discriminator



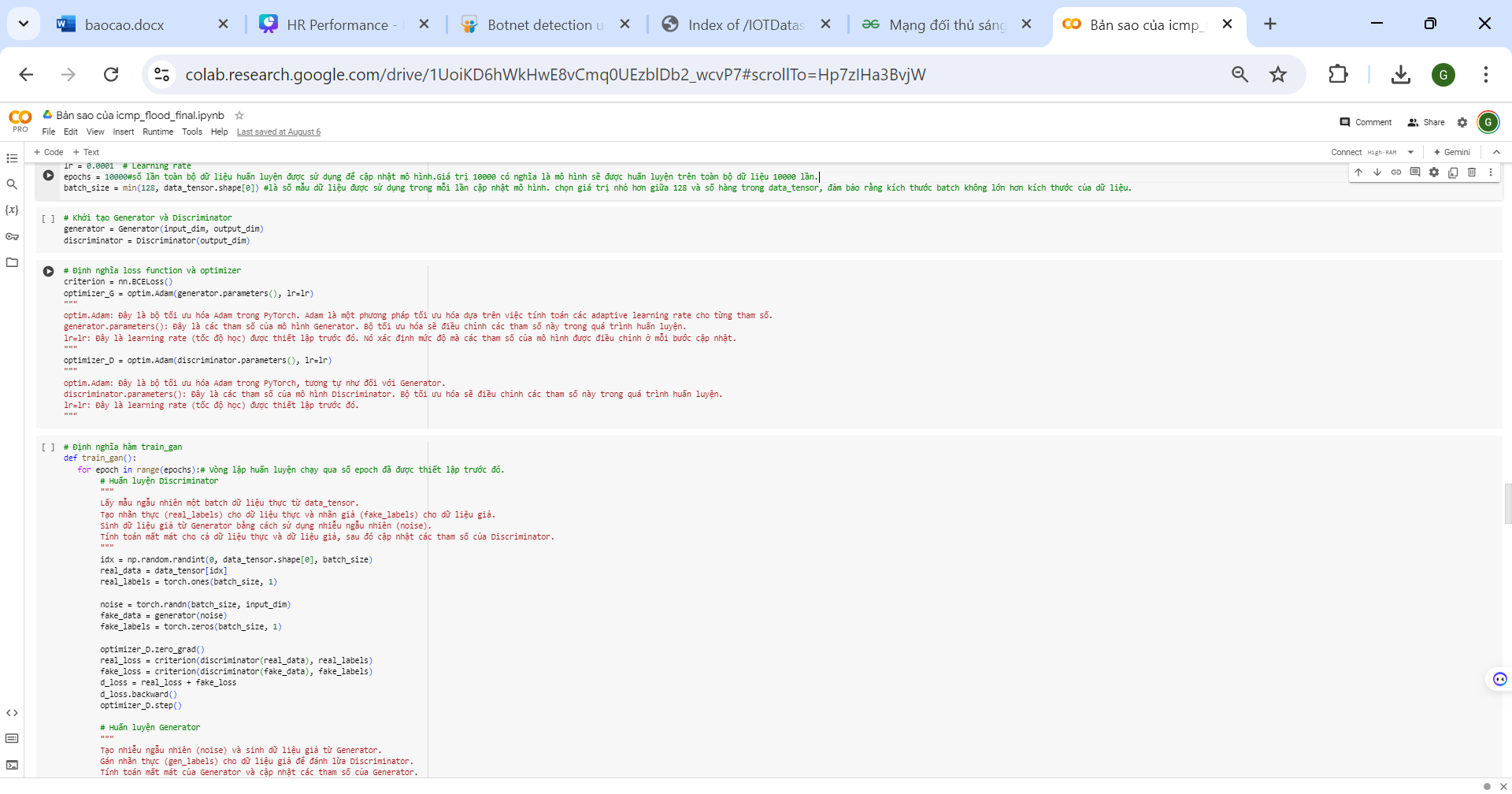
### 3.5.3 Cấu Hình Các Tham số



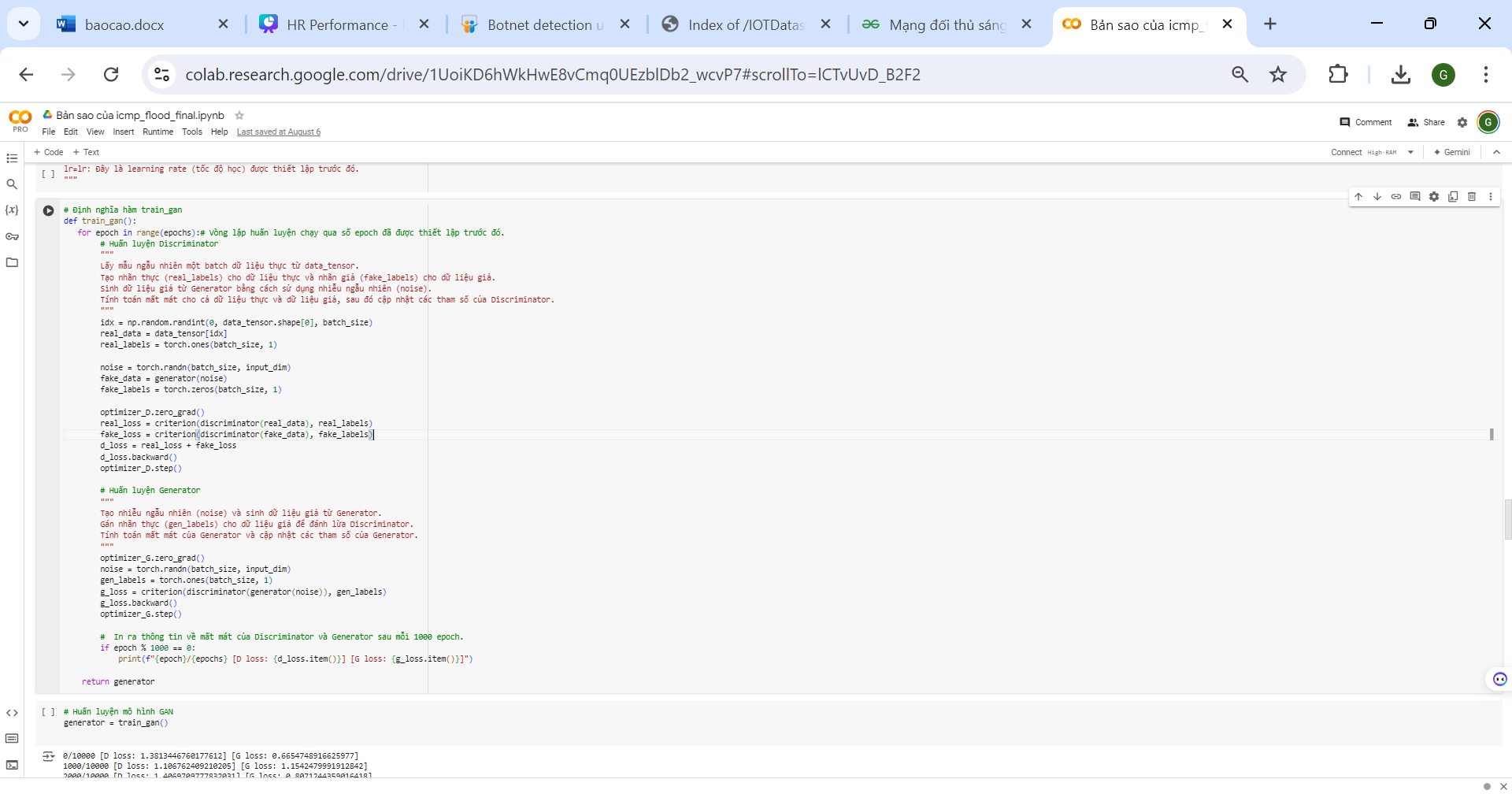
### 3.5.4 Khởi tạo Generator và Discriminator



### 3.5.5 Xây Dựng Hàm LOSS

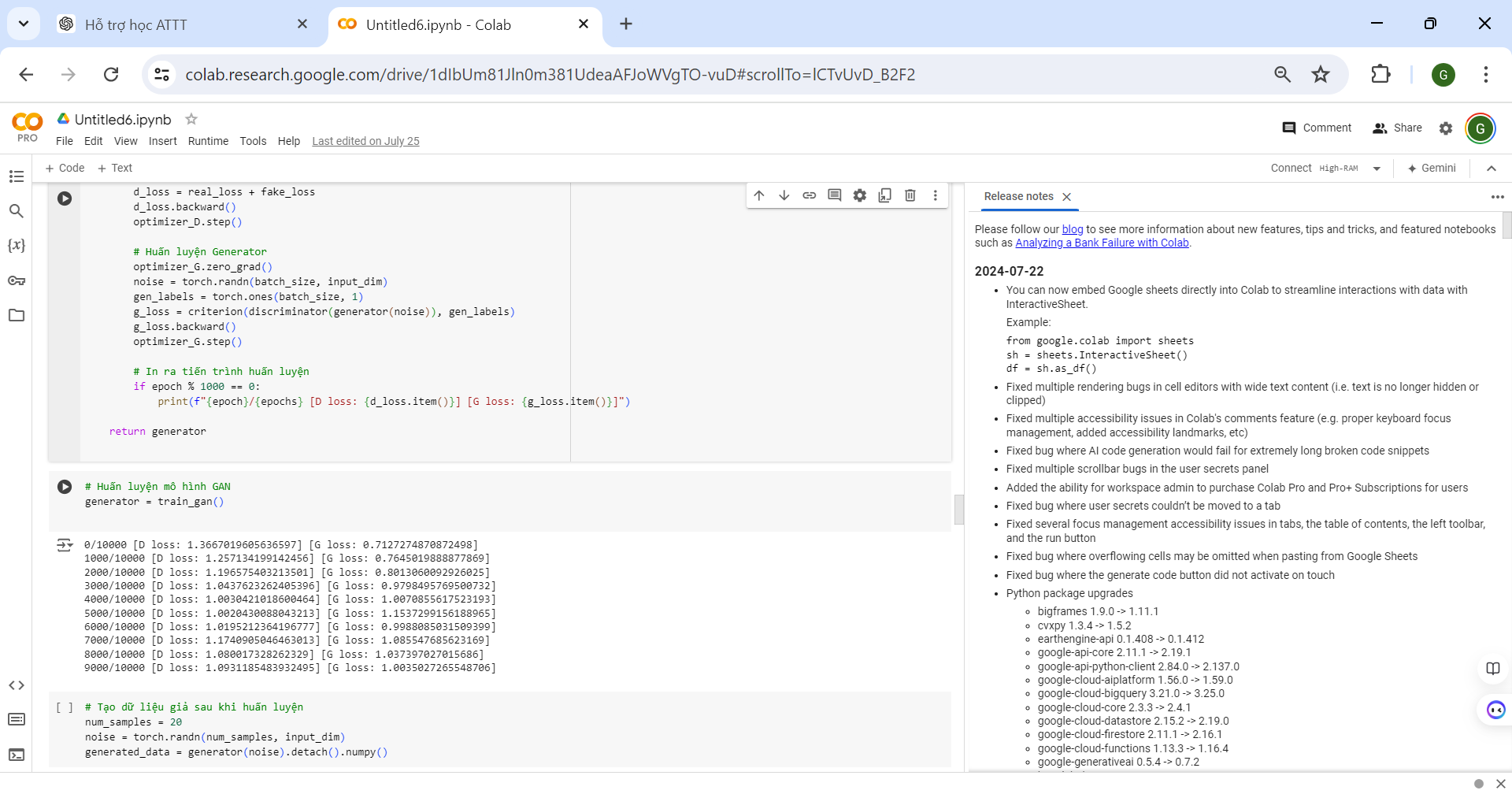


### 3.5.6 Tạo hàm train GANS

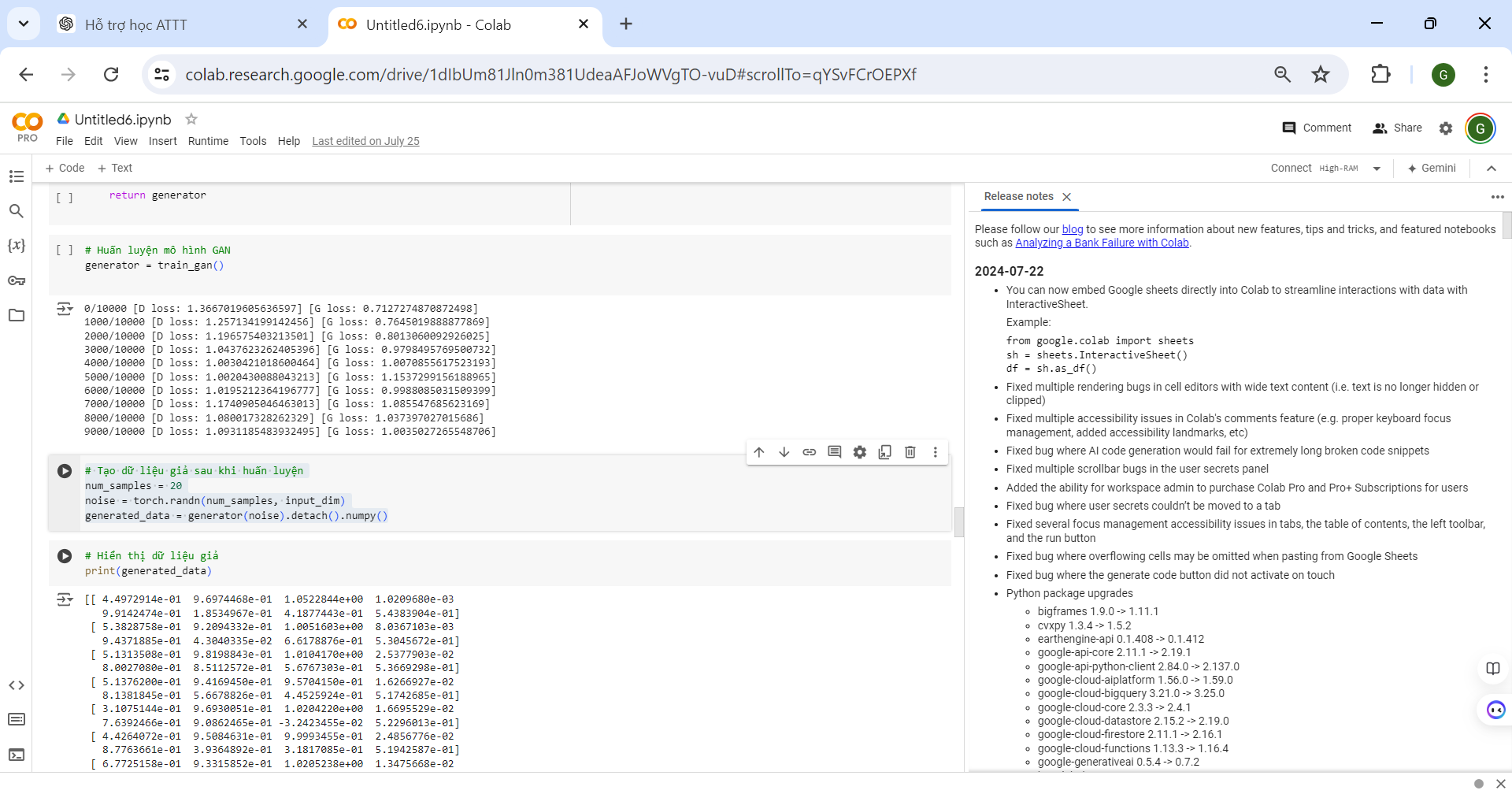


## 3.6 Huấn luyện GANS Tạo mẫu thử DDos

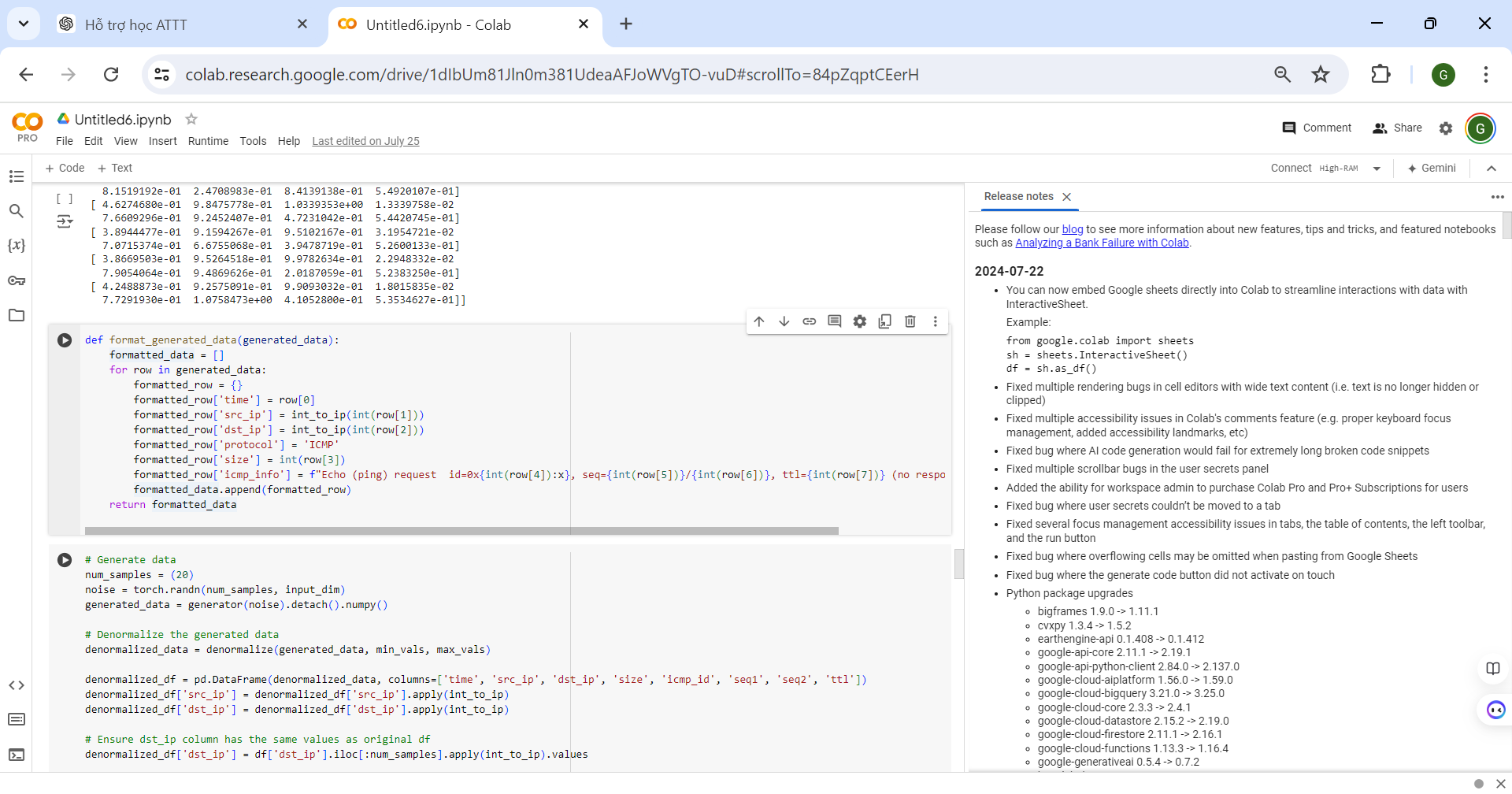
### 3.6.1 Huấn luyện mô hình GANS



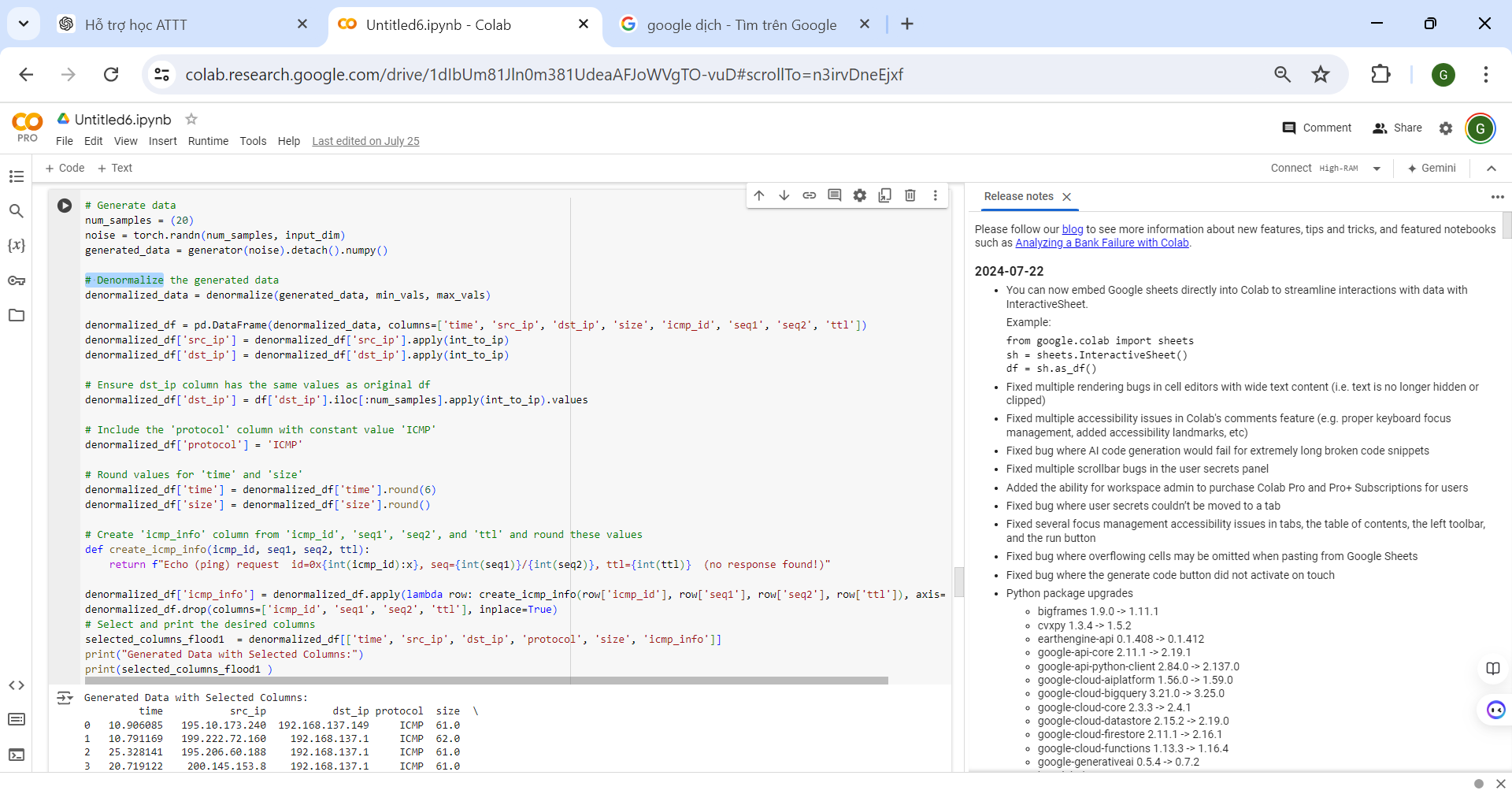
3.6.2 Tạo sớ lượng mẫu thử giả



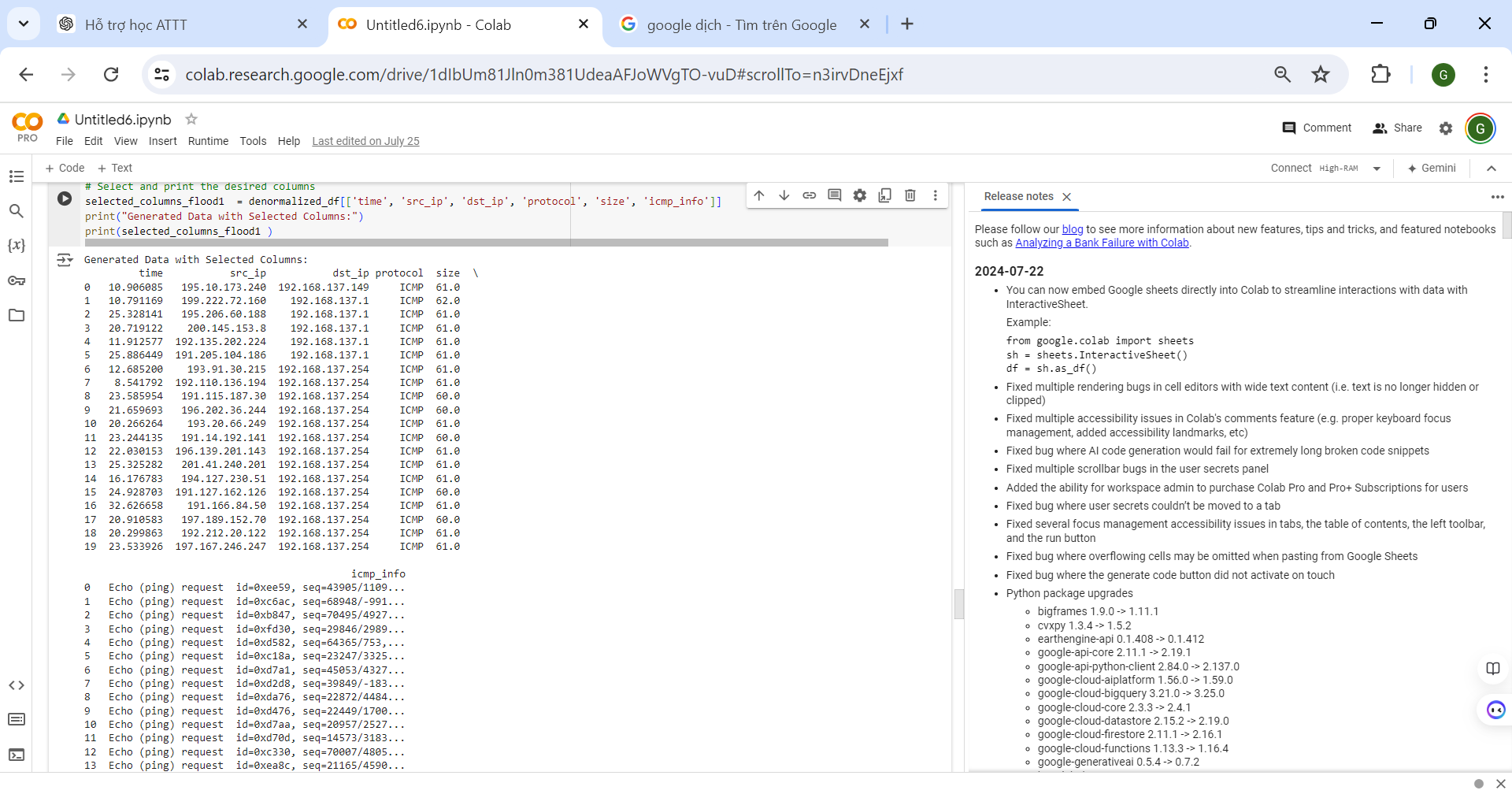
3.6.3 Tạo hàm chuẩn hóa mẫu thử giả



3.6.4 Chuẩn hóa mẫu thử ICMP

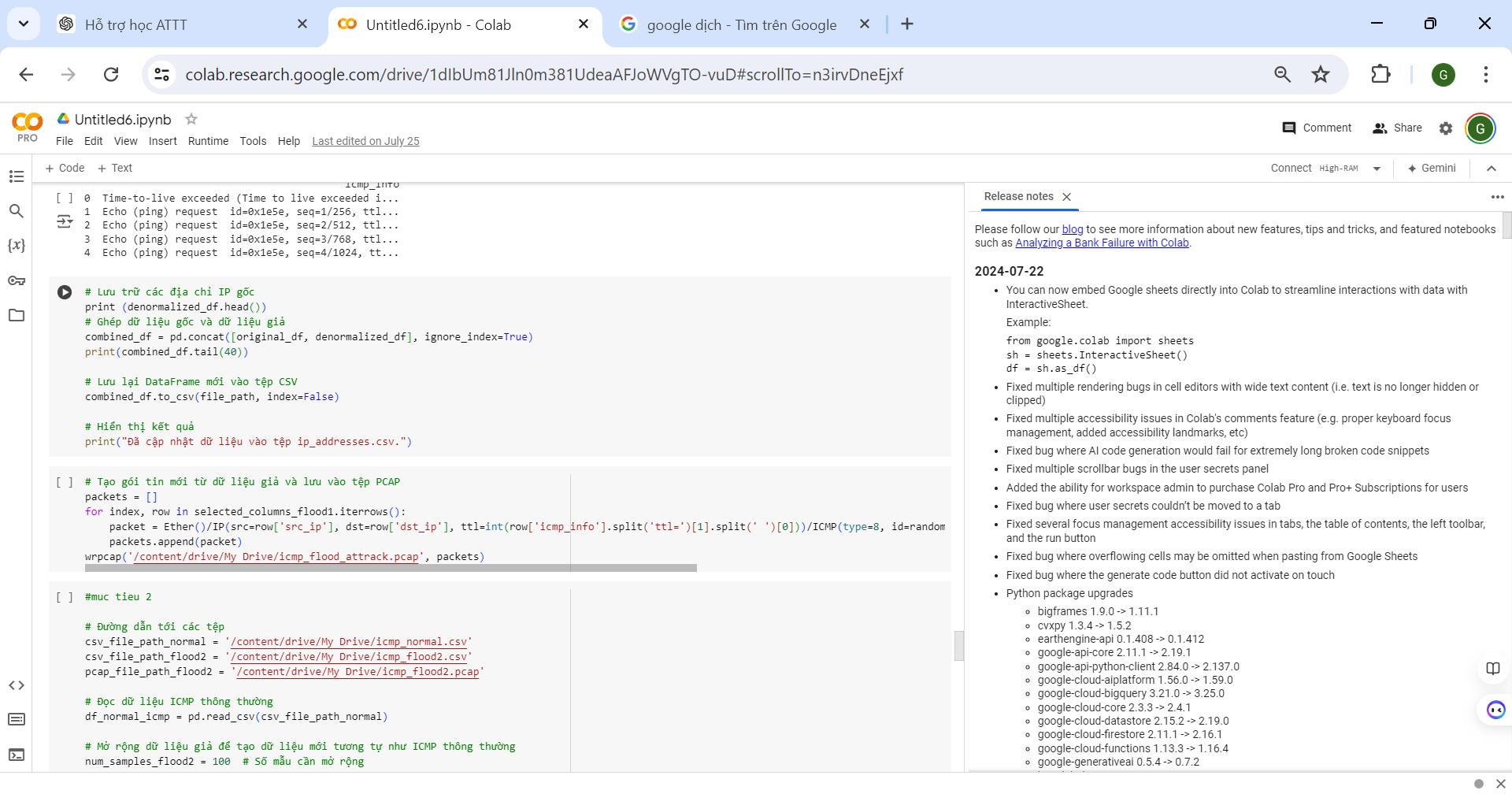


3.6.5 Kết Quả hiển Thị

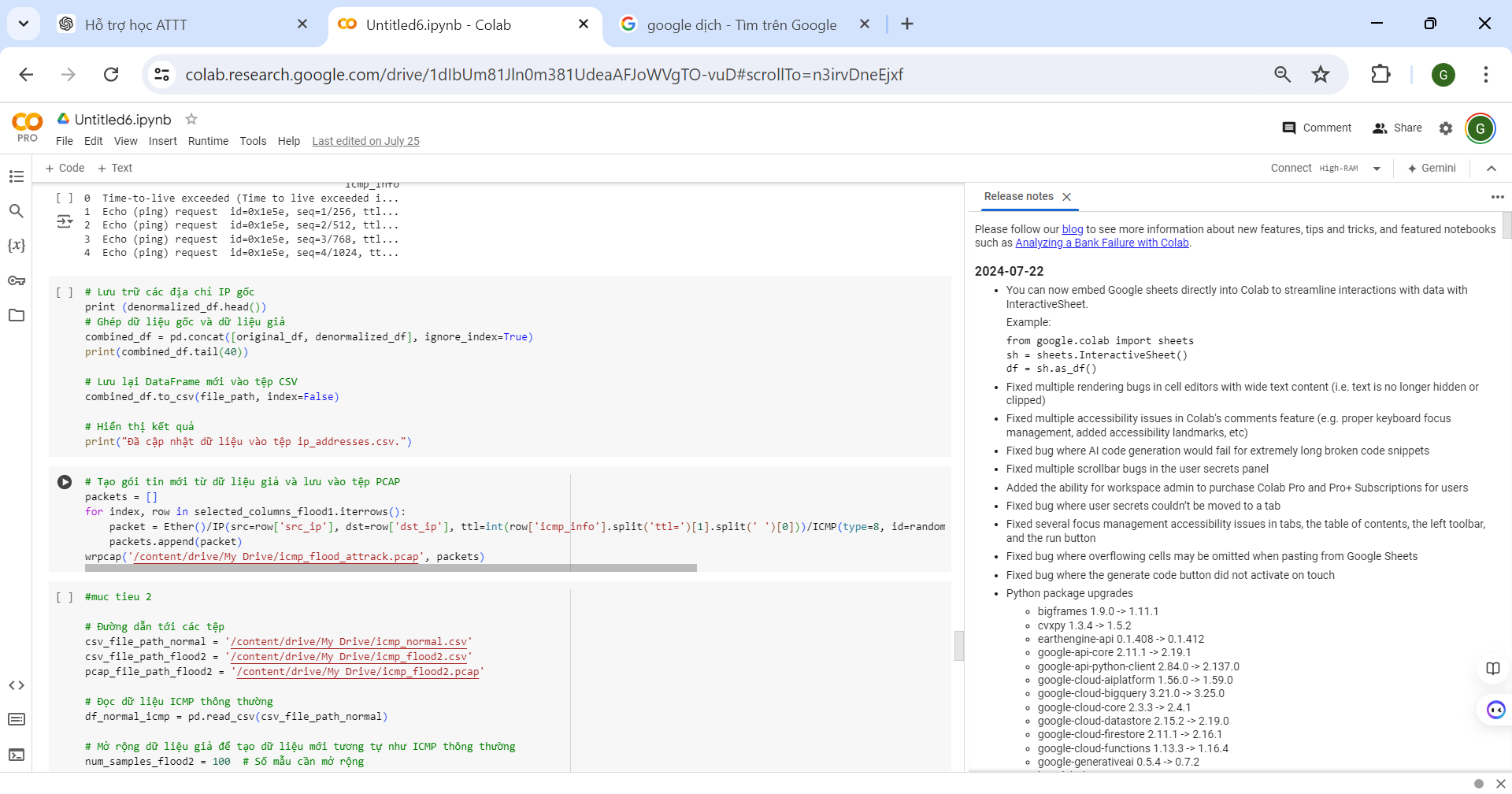


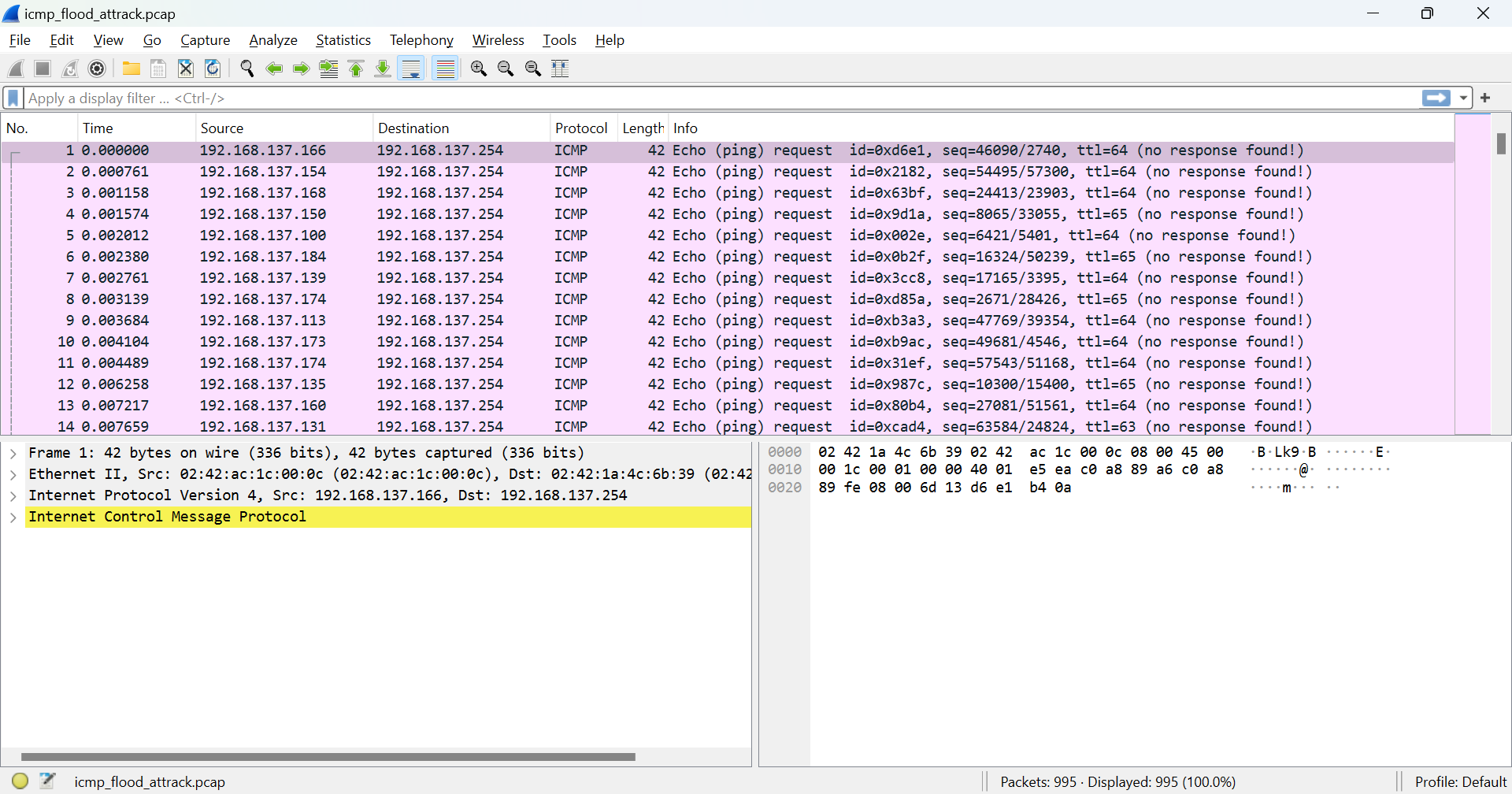
## 3.7 Lưu mẫu thử DDos vao csv và file pcap mới

3.7.1 Lưu mẫu thử vào csv



3.7.2 tạo file pcap mới



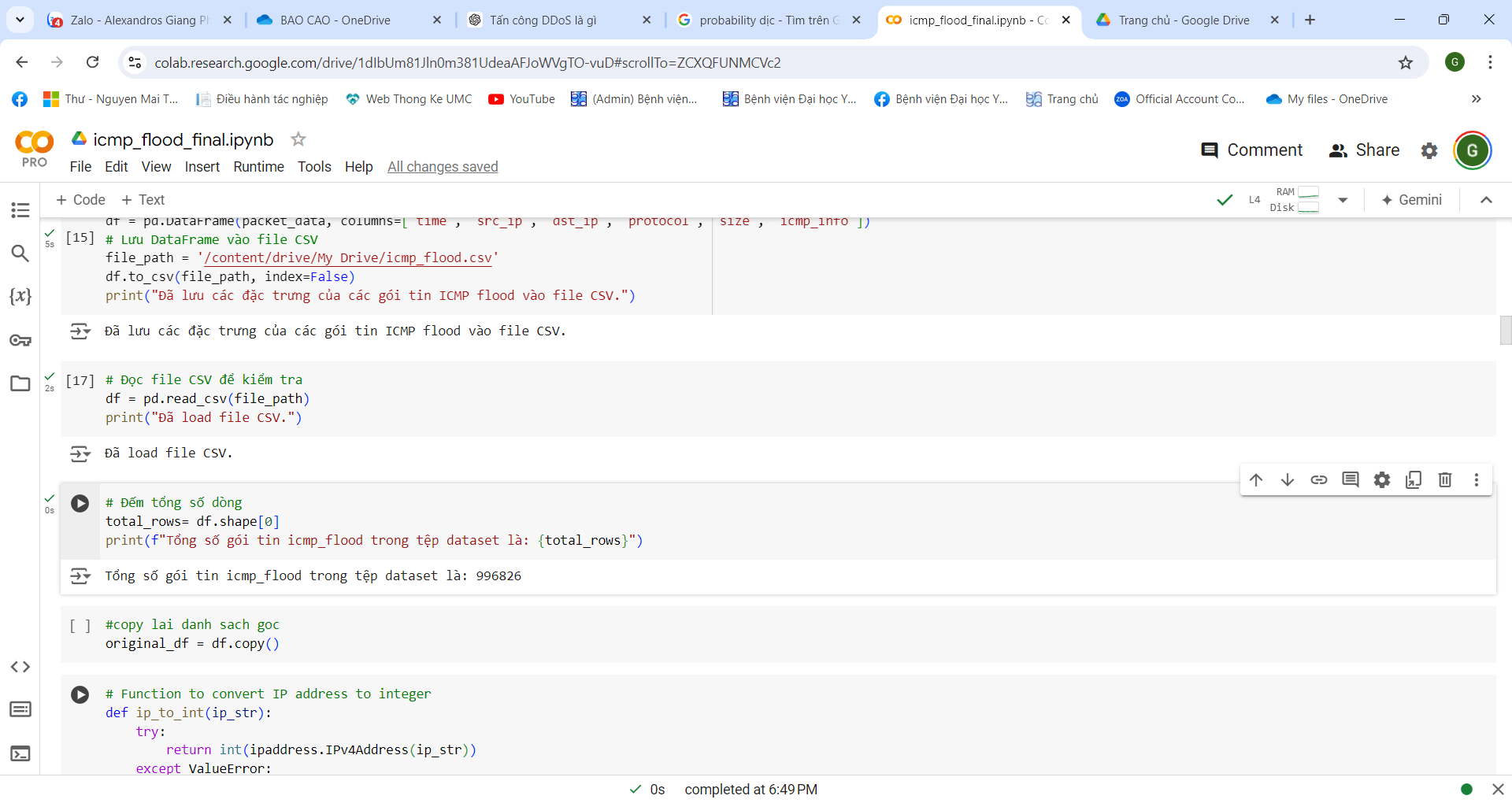


## 3.8 Đánh Gía Mẫu Thử

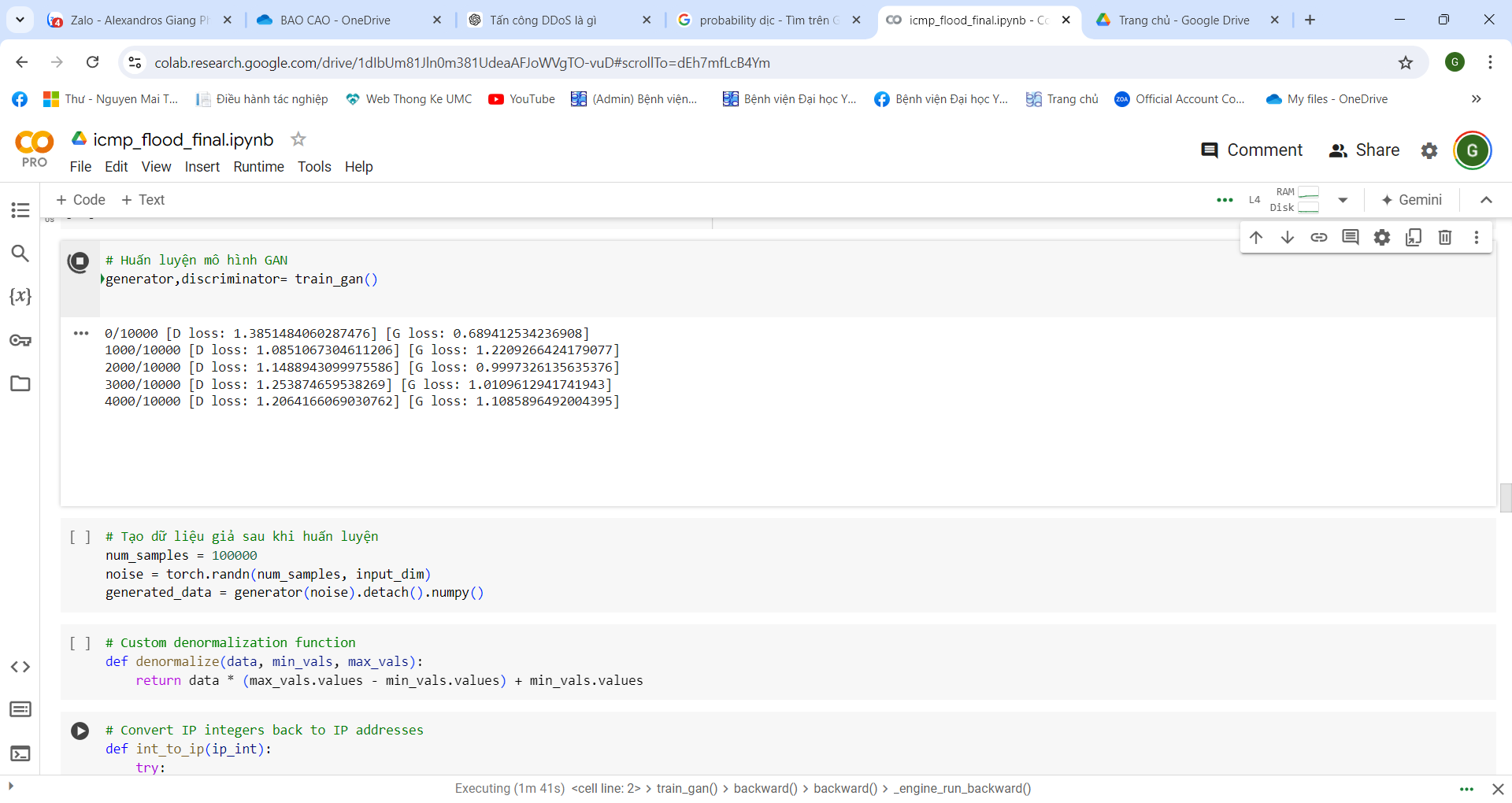
### 3.8.1 Đánh Gía dựa trên dataset từ CIC-2023

Trong đề tài này, sau khi xử lý và trích lọc được các gói tin từ file pcap, chúng ta có được bộ dataset đầu tiên từ CIC và được đăt tên là icmp\_flood.csv.

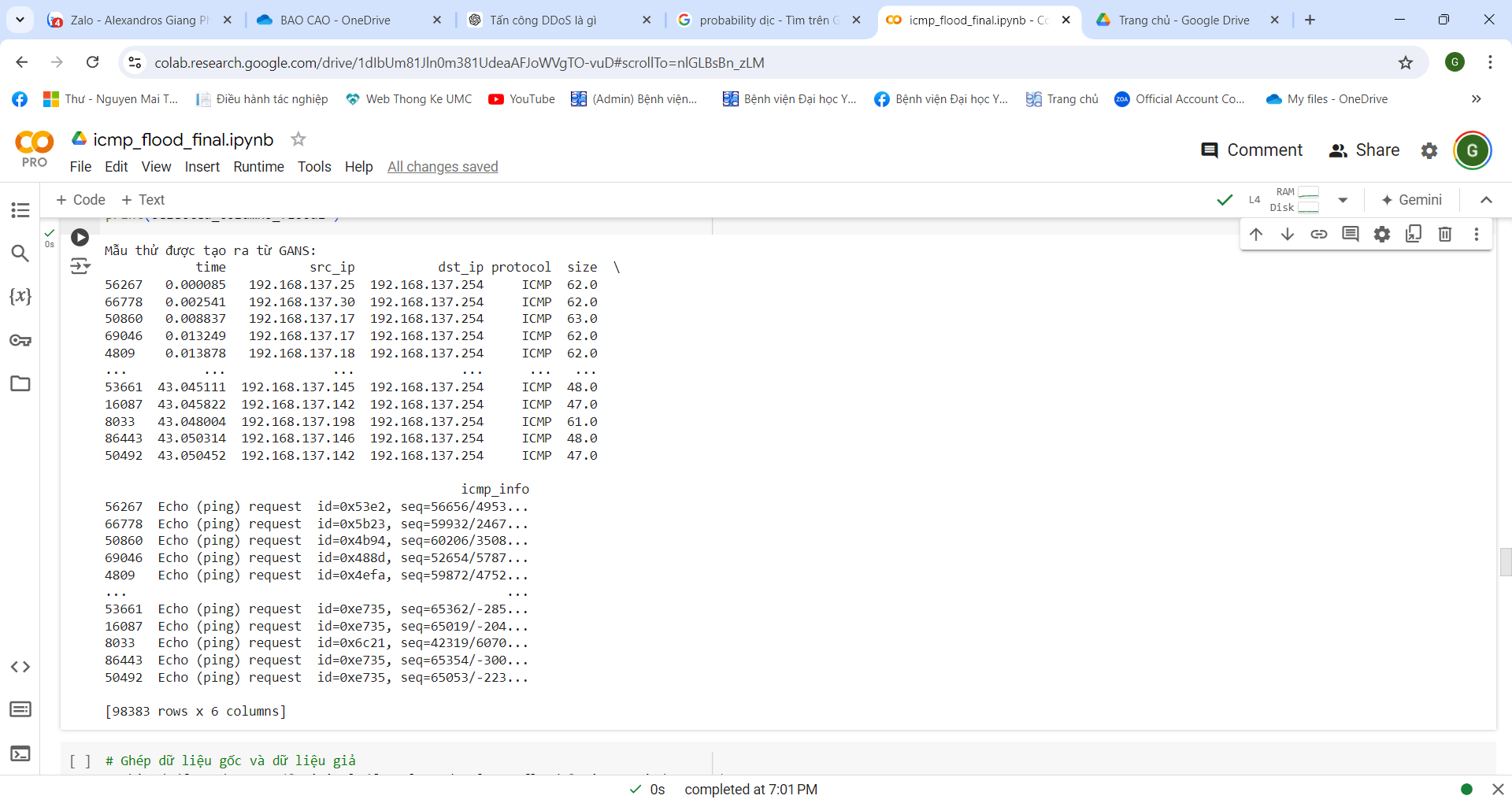
Trong file csv này gồm tổng cộng 996826 gói tin icmp



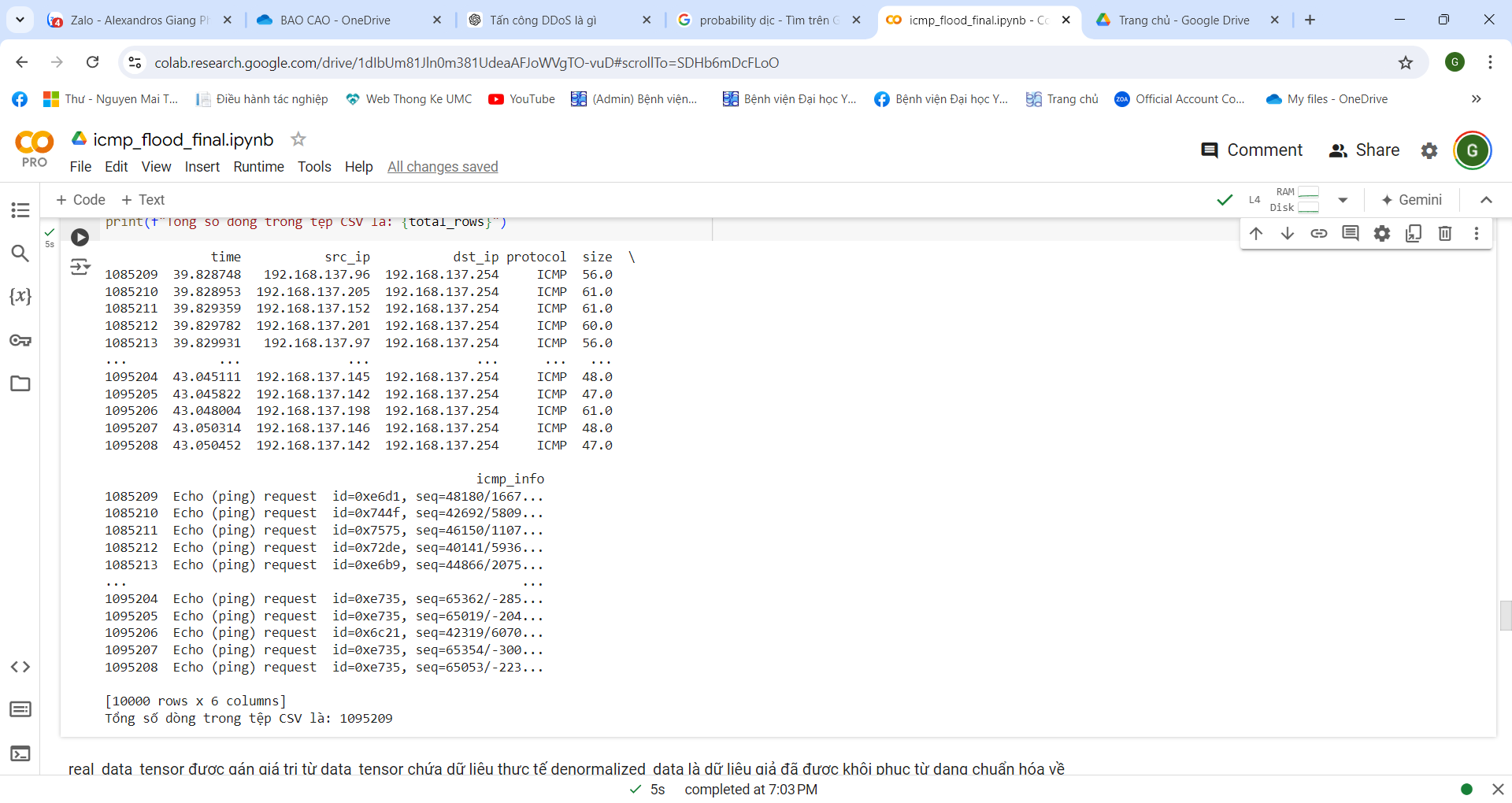
Sau khi build GANs và huấn luyện, chúng tôi tạo 100000 mẫu thử



Sau khi chuẩn hóa dữ liệu về cùng định dạng với dataset và loại bỏ các giá trị âm thì chúng tôi thu được 98383 mẫu thử



Chúng tôi tích hợp những mẫu thử này vào Dataset icmp\_flood đầu tiên. Chúng tôi đã có thêm 1095209 gói tin gồm 996826 gói tin icmp ban đầu và 98383 mẫu thử được tạo ra từ GANS



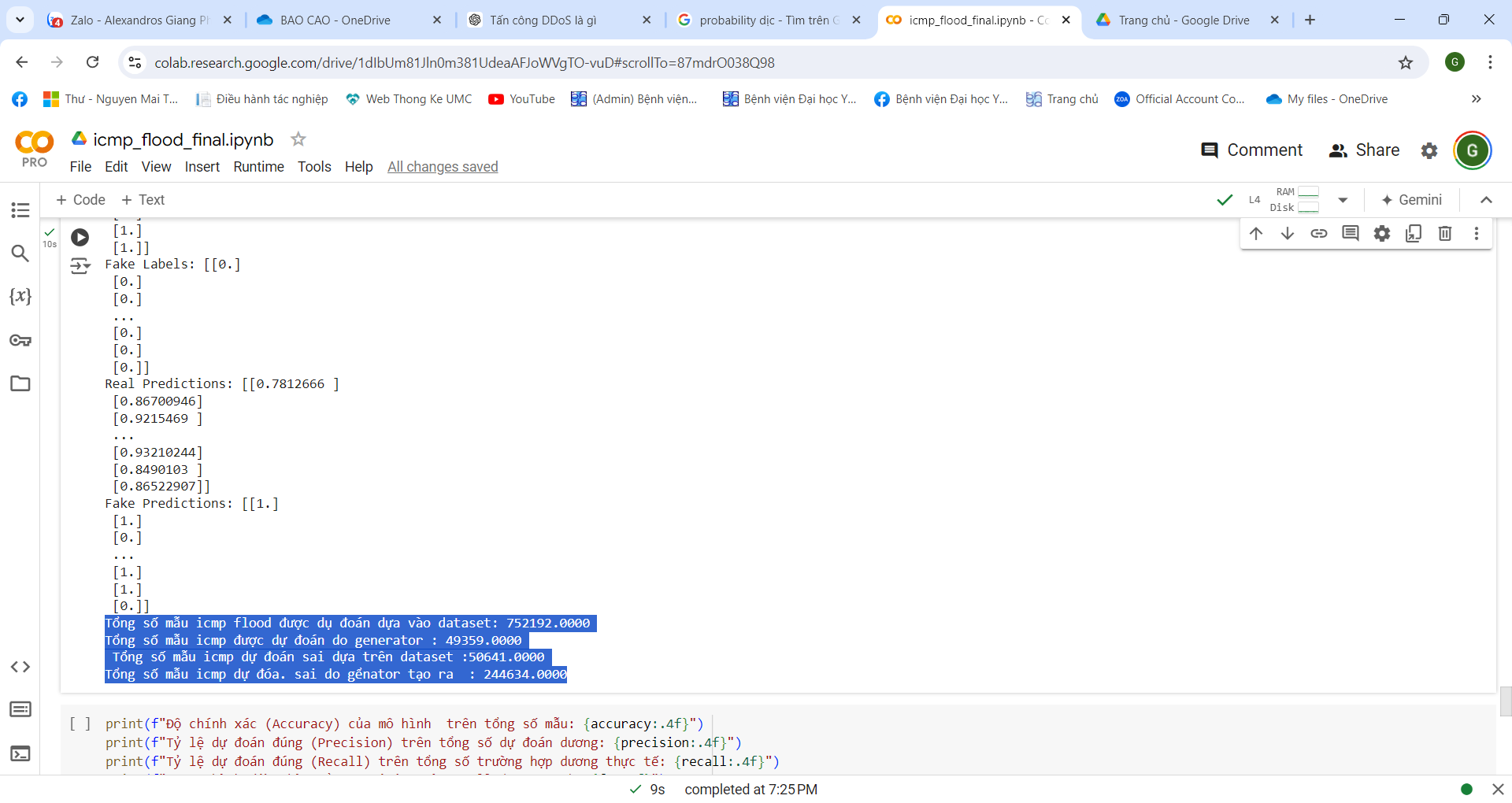
Chúng tôi tiến hành đánh giá sơ bộ mô hình GANs bằng các chỉ số đánh giá. Chúng tôi tiến hành gán nhãn (1) cho dữ liệu từ DATASET và nhãn 0 cho mẫu thử được sinh ra từ GANs

Tiếp theo, chúng tôi dùng discriminator kiểm tra các mẫu thử sinh ra từ GANS là icmp\_flood hay icmp thường.

Chúng tôi tiến hành gộp nhãn thật và nhãn giả vào chung 1 mảng và gộp dự đoán icmp thường và icmp\_flood vào chung 1 mảng.

Chúng tôi tạo ngưỡng dựa trên dự đoán với xác suất là 0.5. Chúng tôi thu được các chỉ số sau:

* Tổng số mẫu icmp flood được dự đoán dựa vào dataset: 752192.0000
* Tổng số mẫu icmp được dự đoán do generator: 49359.0000
* Tổng số mẫu icmp dự đoán sai do genarator tạo ra: 50641.0000
* Tổng số mẫu icmp dự đoán sai dựa trên dataset: 244634.0000



Chúng tôi tiến hành tính toán các chỉ số đánh giá của mô hình chúng tôi có kết quả sau:

* Độ chính xác (Accuracy) của mô hình trên tổng số mẫu: 0.7308
* Tỷ lệ dự đoán đúng (Precision) trên tổng số dự đoán dương: 0.9369
* Tỷ lệ dự đoán đúng (Recall) trên tổng số trường hợp dương thực tế: 0.7546
* Trung bình điều hòa của Precision và Recall (F1 Score): 0.8359

Chúng tôi có nhận xét sau:

* Mô hình GAN hiện tại có độ chính xác khá tốt với các chỉ số đánh giá cho thấy mô hình có thể phát hiện các mẫu ICMP flood một cách hiệu quả.
* Tuy nhiên, để đảm bảo tính chính xác và ổn định, cần xem xét retrain mô hình thường xuyên với dữ liệu mới và đánh giá các chỉ số hiệu suất để đảm bảo mô hình hoạt động tốt

### 3.8.2 Đánh Giá dựa trên dataset đã tích hợp mẫu thử

Sau khi chúng ta tích hợp các mẫu thử đã được tạo ra vào dataset gốc thì dataset của chúng ta đã tăng lên 1095209. Chúng ta tiếp tục huấn luyện dựa trên mô hình GANs để sinh ra mẫu thử.

Sau khi chúng ta huấn luyện mới mô hình GANs sinh ra mẫu thử, chúng ta thu được 98083 mẫu thử icmp\_flood.

Chúng tôi tiến hành đánh giá mô hình GANs sau khi retrain này với 1095209 dữ liệu và 98083. Chúng ta có kết quả như sau:

* Tổng số mẫu icmp flood được dự đoán dựa vào dataset: 743215.0000
* Tổng số mẫu icmp được dự đoán do generator: 22199.0000
* Tổng số mẫu icmp dự đoán sai do generator tạo ra: 77801.0000
* Tổng số mẫu icmp dự đoán sai dựa trên dataset: 351994.0000

Chúng tôi cũng thu được chỉ số đánh giá mô hình retrain như sau:

* Độ chính xác (Accuracy) của mô hình trên tổng số mẫu: 0.6404
* Tỷ lệ dự đoán đúng (Precision) trên tổng số dự đoán dương: 0.9052
* Tỷ lệ dự đoán đúng (Recall) trên tổng số trường hợp dương thực tế: 0.6786
* Trung bình điều hòa của Precision và Recall (F1 Score): 0.7757

### 3.8.3 Nhận Xét

Chúng tôi đưa ra nhận xét đánh giá mô hình như sau:

* **Độ chính xác (Accuracy):**
* Trước khi retrain: 0.7308
* Sau khi retrain: 0.6404

**Nhận xét**: Độ chính xác của mô hình đã giảm sau khi retrain. Điều này cho thấy rằng sau khi retrain, mô hình không còn phân loại chính xác được nhiều mẫu như trước đây.

* **Tỷ lệ dự đoán đúng (Precision)**
* Trước khi retrain: 0.9369
* Sau khi retrain: 0.9052

**Nhận xét**: Tỷ lệ dự đoán đúng giảm nhẹ sau khi retrain. Điều này có nghĩa là mô hình sau retrain vẫn giữ được khả năng dự đoán đúng các mẫu dương, nhưng đã giảm nhẹ so với trước.

* **Tỷ lệ dự đoán đúng (Recall):**
* Trước khi retrain: 0.7546
* Sau khi retrain: 0.6786

**Nhận xét:** Tỷ lệ dự đoán đúng giảm sau khi retrain. Điều này cho thấy rằng mô hình sau khi retrain bỏ lỡ nhiều trường hợp dương hơn so với trước khi retrain.

**F1 Score**:

* Trước khi retrain: 0.8359
* Sau khi retrain: 0.7757

**Nhận xét:** F1 Score giảm sau khi retrain, điều này phản ánh sự suy giảm cả về Precision và Recall. F1 Score giảm cho thấy sự cân bằng giữa Precision và Recall không còn tốt như trước.

# Chương 4: Kết luận và hướng phát triển

## 4.1 Kết Kuận

### 4.1.1 Hiệu quả của GANs trong việc tạo mẫu thử ICMP flood:

* Mô hình GANs đã cho thấy khả năng tạo ra các mẫu thử ICMP flood có độ chính xác cao, giúp cải thiện việc phát hiện và ngăn chặn các cuộc tấn công mạng.
* Đánh giá mô hình qua các chỉ số như Accuracy, Precision, Recall và F1 Score cho thấy mô hình đạt hiệu suất khá tốt trước khi retrain và có một số suy giảm sau khi retrain.

### 4.1.2 Nhược điểm và nguyên nhân tiềm ẩn

* Mô hình có thể bị giảm hiệu suất sau khi retrain, điều này có thể do các nguyên nhân như dữ liệu không đủ đại diện, tham số huấn luyện chưa tối ưu, hoặc overfitting.
* Dữ liệu dùng để retrain có thể không phản ánh đầy đủ và chính xác tất cả các tình huống có thể xảy ra trong thực tế.

## 4.2 Hướng Phát Triển Trong Tương Lai

### 4.2.1. Nâng cao chất lượng data

* Thu thập thêm data: Tăng cường dữ liệu huấn luyện bằng cách thu thập thêm các mẫu dữ liệu đa dạng và đại diện hơn.
* Tiền xử lý data: Xử lý các giá trị bị thiếu, loại bỏ dữ liệu nhiễu và đảm bảo tính nhất quán của dữ liệu.

### 4.2.2. Cải Thiện Kiến Trúc GANs

* Sử dụng các mô hình phức tạp hơn: Thử nghiệm với các biến thể khác của GANs như DCGAN, WGAN để cải thiện khả năng sinh mẫu.
* Kết hợp với các mô hình khác: Kết hợp GANs với các mô hình học sâu khác như LSTM hoặc CNN để tăng cường khả năng học và sinh mẫu.

### 4.2.3. Đánh giá và kiểm thử mô hình

* Đánh giá trên nhiều bộ dữ liệu khác nhau: Kiểm thử mô hình trên các bộ dữ liệu khác nhau để đảm bảo tính tổng quát và khả năng áp dụng rộng rãi.
* Phân tích kết quả: Phân tích kỹ lưỡng kết quả dự đoán để hiểu rõ hơn về hiệu suất của mô hình và các yếu tố ảnh hưởng đến hiệu suất.

### 4.2.4. Phát triển thêm tính năng

* Tự động hóa quá trình huấn luyện và đánh giá: Phát triển các công cụ tự động để huấn luyện và đánh giá mô hình.
* Tích hợp vào hệ thống phát hiện tấn công: Áp dụng mô hình GANs vào hệ thống phát hiện và ngăn chặn tấn công mạng thực tế.

# Tài Liệu Tham Khảo

[1]. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. “Generative adversarial networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Volume 2, 2672–2680”.

[2]. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. “Improved training of Wasserstein GANs. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; 5769–5779”

[3]. Nguyen, T.D.; Le, T.; Vu, H.; Phung, “D. Dual Discriminator Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 4–9 December 2017; pp. 2667–2677”.

[4]. Zhang, X.; Zhao, Y.; Zhang, H. “Dual-discriminator GAN: A GAN way of profile face recognition. In Proceedings of the 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 27–29 June 2020; 162–166”