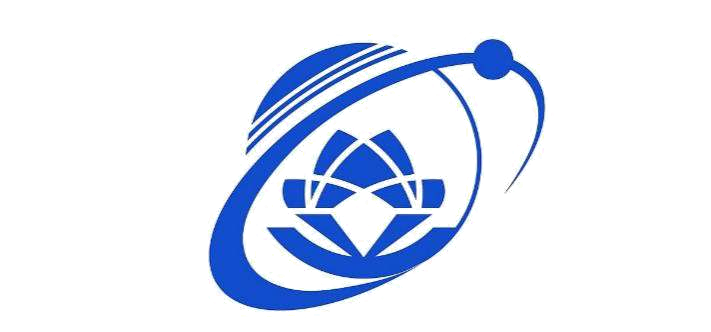
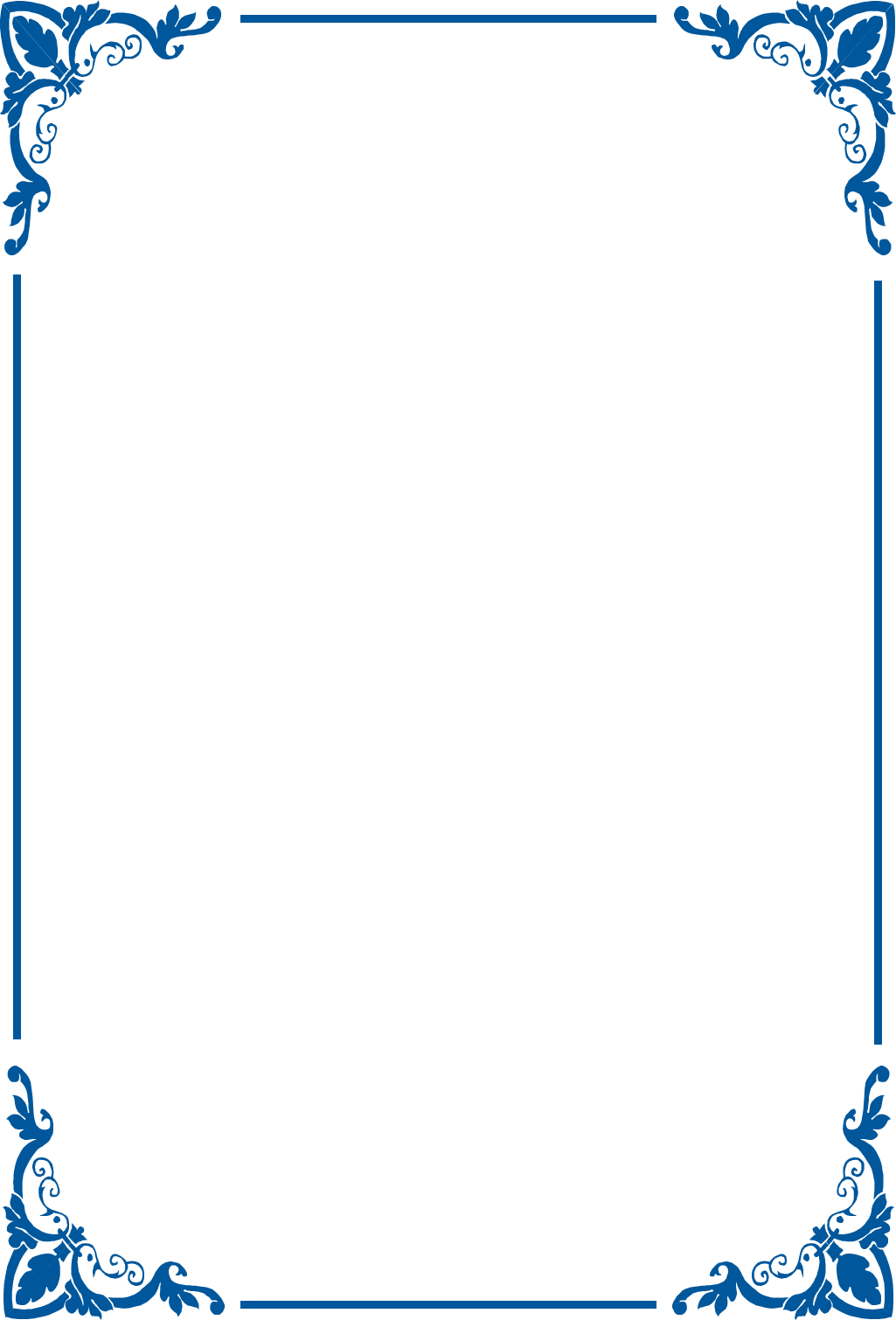
e



# ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

**TRƯỜNG** **ĐẠI** **HỌC** **CÔNG** **NGHỆ** **THÔNG** **TIN** **KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

**BÁO** **CÁO** **ĐỒ** **ÁN**

**HỆ THỐNG TÌM KIẾM, PHÁT HIỆN VÀ NGĂN NGỪA XÂM NHẬP**

**ĐỀ** **TÀI**

**Evasion Generative Adversarial Network**

**for Low Data Regimes**

Giảng viên hướng dẫn: ThS. Đỗ Hoàng Hiển

Sinh viên thực hiện:

Mai Ngọc Phương Trinh 20520823

Nguyễn Thị Hồng Lam 20521518

Nguyễn Thị Thanh Mai 21521112

Nguyễn Lê Thảo Ngọc 21521191

Huỳnh Minh Khuê 21522240

**TP.** **Hồ** **Chí** **Minh,** **tháng 05** **năm 2024**

Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến thầy Đỗ Hoàng Hiển. Với sự tâm huyết, được sự tận tình giảng dạy và hỗ trợ hết lòng của thầy đã cho chúng em nhiều kiến thức bổ ích, không chỉ trong phạm vi sách vở mà còn ở nhiều khía cạnh khác trong thực tế, xã hội. Với tình cảm sâu sắc, chân thành, chúng em xin bày tỏ lòng biết ơn đến thầy đã nhiệt tình, hết mình với sinh viên. Đó là động lực rất lớn để chúng em có thể hoàn thành tốt đồ án lần này.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, chúng em đã cố gắng hết mình nhưng đồ án không thể tránh được những thiếu sót. Chúng em rất hy vọng nhận được sự chỉ bảo và đóng góp ý kiến từ thầy để bổ sung, nâng cao kiến thức của mình, phục vụ và hoàn thiện hơn trong những đồ án sau này và khóa luận tốt nghiệp trong tương lai.

Chúng em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, tháng 05 năm 2024

Nhóm sinh viên thực hiện

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**…………………………………………………………………………………………**

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU 6](#_Toc169384259)

[1.1. Lý do chọn đề tài 6](#_Toc169384260)

[1.2. Mục tiêu nghiên cứu 6](#_Toc169384261)

[1.3. Phương pháp nghiên cứu 6](#_Toc169384262)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 8](#_Toc169384263)

[2.1. Mô hình GAN 8](#_Toc169384264)

[2.2. Mô hình ACGAN 9](#_Toc169384265)

[2.3. Mô hình EVAGAN 10](#_Toc169384266)

[2.4. So sánh mô hình ACGAN và EVAGAN 12](#_Toc169384267)

[2.5. Dataset 14](#_Toc169384268)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 15](#_Toc169384269)

[3.1. Vấn đề: 16](#_Toc169384270)

[3.2. Phương pháp: 17](#_Toc169384271)

[3.3. Kiến trúc tổng quan: 18](#_Toc169384272)

[3.3.1. Bộ sinh dữ liệu (Generator): 19](#_Toc169384273)

[3.3.2. Bộ phân biệt dữ liệu (Discriminator): 20](#_Toc169384274)

[CHƯƠNG 4: HIỆN THỰC HỆ THỐNG 23](#_Toc169384275)

[4.1. Môi trường thực nghiệm 23](#_Toc169384276)

[4.2. Dataset 23](#_Toc169384277)

[4.2.1. CC botnet Dataset 23](#_Toc169384278)

[4.2.2. CV Dataset 24](#_Toc169384279)

[4.3. Tiền xử lý dữ liệu 24](#_Toc169384280)

[CHƯƠNG 5: THỰC NGHIỆM VÀ ĐÁNH GIÁ 25](#_Toc169384281)

[5.1. Thực Nghiệm 25](#_Toc169384282)

[5.1.1. Mục tiêu 25](#_Toc169384283)

[5.1.2. Triển khai 25](#_Toc169384284)

[5.2. Đánh giá 32](#_Toc169384285)

[CHƯƠNG 6: KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN 33](#_Toc169384286)

[6.1. Kết luận 33](#_Toc169384287)

[6.2. Phương hướng phát triển 33](#_Toc169384288)

# CHƯƠNG 1: GIỚI THIỆU

## 1.1. Lý do chọn đề tài

Trong các ứng dụng thực tế, vấn đề của sự khan hiếm dữ liệu là một thách thức lớn khi mẫu dữ liệu bình thường chiếm đa số hơn so với các mẫu dữ liệu bất thường. Sự mất cân bằng này dẫn đến sự thiên vị trong quá trình ra quyết định của các mô hình máy học, làm giảm đi hiệu suất của chúng trong việc phát hiện các mẫu dữ liệu bất thường. Do đó nhóm đã tìm và nghiên cứu về một mô hình GAN mới có tên là Evasion Generative Adversarial Network - EVAGAN.

## 1.2. Mục tiêu nghiên cứu

Mô hình GAN mới dựa trên ACGAN - EVAGAN được thiết kế để giải quyết vấn đề ít dữ liệu. Mô hình này kết hợp khả năng tạo ra các mẫu dữ liệu giả với khả năng phân loại các mẫu dữ liệu bất thường, mà không cần sự hỗ trợ từ các bộ phân loại machine learning. Điều này giúp cải thiện hiệu suất trong việc phát hiện các mẫu dữ liệu bất thường và giảm thiểu ảnh hưởng của sự khan hiếm dữ liệu trong các ứng dụng thực tế như phát hiện xâm nhập trong mạng máy tính và thị giác máy tính.

## 1.3. Phương pháp nghiên cứu

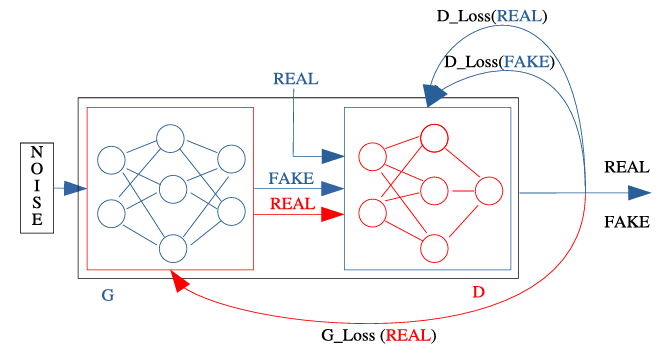
* Một mô hình GAN cổ điển bao gồm generator (G) và discriminator (D). Trong đó, G chịu trách nhiệm tạo ra các mẫu tổng hợp và D sẽ đánh giá các mẫu do G tạo ra.
* Thiết kế EVAGAN được điều chỉnh để đáp ứng các vấn đề dữ liệu thấp, bao gồm 2 mạng neural là EVAGenerator và EVADiscriminator. EVAGenerator sẽ tạo ra các mẫu dữ liệu giả ; EVADiscriminator sẽ phân biệt mẫu dữ liệu thật, giả và đánh giá mức độ tránh bị phân loại của mẫu dữ liệu.
* Sử dụng kĩ thuật oversampling, các mẫu dữ liệu thật, giả được kết hợp với original training dataset tạo thành dataset mở rộng để tăng khả năng phát hiện
* Sử dụng dataset ISCX-2014, CIC-2017, CIC-2018 và MNIST, so sánh mô hình EVAGAN và ACGAN để đánh giá hiệu suất.
* Để kiểm chứng hiệu suất của mô hình EVAGAN, nhóm sẽ so sánh với mô hình ACGAN dựa trên các tiêu chí: Hiệu suất phát hiện; Độ ổn định; Hiệu suất chất lượng; Độ phức tạp thời gian.

# 

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Mô hình GAN

* GAN (Generative Adversaria Networks) là một phương pháp học sâu mạnh mẽ cho việc tạo ra dữ liệu mới từ dữ liệu huấn luyện. Mô hình GAN bao gồm hai mạng neural: Generator (G) và Discriminator (D). Có hai bước liên tiếp trong đó GAN được huấn luyện.
* Ở bước đầu tiên, D được huấn luyện trên dữ liệu thật được gắn nhãn là REAL và dữ liệu do G chưa được huấn luyện tạo ra được gắn nhãn là FAKE.
* Trong bước tiếp theo, sau khi D đã được huấn luyện, nó được kiểm tra trên dữ liệu giả từ G, nhưng lần này cố ý được gắn nhãn là REAL. Lỗi của D trên dữ liệu được gắn nhãn sai này được truyền lại cho G, điều chỉnh các trọng số của nó trong một quá trình huấn luyện batch hoàn chỉnh. Có thể có nhiều lần lặp batch, sau đó một lần duyệt hoàn chỉnh toàn bộ tập dữ liệu được hoàn thành, còn được gọi là một epoch.

*Hình 2.1: Sơ đồ mô hình GANs*

* Các mất mát của D trên dữ liệu thật D\_Loss(REAL) và dữ liệu được tạo ra D\_Loss(FAKE) lần lượt được đưa vào D bằng cách sử dụng lan truyền ngược. Trong bước tiếp theo, trong quá trình lan truyền tiến, với nhãn là REAL cho các mẫu dữ liệu được tạo ra (đến từ G), việc đánh giá được thực hiện bởi D và G\_Loss(REAL) được truyền ngược lại cho G để cập nhật các trọng số của nó. Chúng tôi gọi bước này là huấn luyện mô hình kết hợp. Mô hình kết hợp lấy nhiễu làm đầu vào và đầu ra của D làm phản hồi để cập nhật trọng số của G. Quá trình này tiếp tục lặp lại cho đến khi số epoch đạt đến giá trị đặt trước. Generator và Discriminator sẽ không học thêm nữa khi đạt được điểm cân bằng Nash.

## 2.2. Mô hình ACGAN

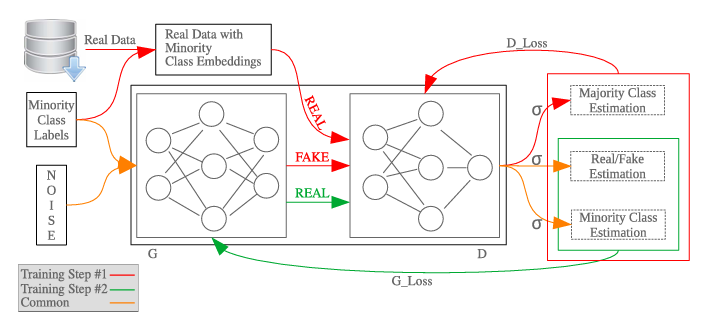
* Auxiliary Classifier GAN (ACGAN) là một biến thể của Generative Adversarial Networks (GAN), Mục tiêu của ACGAN là không chỉ tạo ra dữ liệu giả mà còn phân loại dữ liệu này theo các nhãn cụ thể, do đó, ACGAN có thể sinh ra dữ liệu có điều kiện (conditioned data) với các nhãn nhất định. Do đó, mẫu tổng hợp từ G trong ACGAN là *Xfake*​ = 𝐺(𝑐,𝑧) thay vì *Xfake*​ = 𝐺(𝑧).
* Nói cách khác, ACGAN có thể tạo ra dữ liệu thuộc lớp cụ thể mà chúng ta cung cấp nhãn cho G. Đồng thời, D của ACGAN hoạt động như một bộ phân loại kép để phân biệt giữa dữ liệu thật/giả và các lớp khác nhau của các mẫu đầu vào, dù đến từ nguồn thật hay từ G.
* Hàm mục tiêu của ACGAN bao gồm hai phần:

1. Log-likelihood 𝐿scủa dữ liệu nguồn chính xác.
2. Log-likelihood *Lc* của các nhãn lớp thật.

* D được huấn luyện để tối đa hóa 𝐿c+𝐿s, còn G học để tối đa hóa 𝐿c−𝐿s. Nói cách khác, mục tiêu của D là cải thiện cả hai likelihoods, trong khi mục tiêu của G là hỗ trợ D trong việc cải thiện hiệu suất phân biệt nhãn lớp. G cũng sẽ cố gắng giảm log-likelihood của D trên các mẫu giả. D đầu ra cả phân phối xác suất trên các nguồn và các nhãn lớp, lần lượt là [𝑃(𝑆∣𝑋),𝑃(𝐶∣𝑋)]=𝐷(𝑋) [*P*(*S*∣*X*),*P*(*C*∣*X*)]=*D*(*X*), trong đó S là các nguồn (thật/giả) và C là các nhãn lớp.

## 2.3. Mô hình EVAGAN

* EVAGAN (Evolutionary Generative Adversarial Network) là một biến thể của ACGAN, nhằm phát triển một mô hình phân loại có khả năng tự cải thiện hiệu suất thông qua việc đào tạo trên các mẫu né tránh này. Cấu trúc chính của EVAGAN bao gồm hai mạng neural: generator (G) và discriminator (D). Khác với ACGAN, mô hình của EVAGAN giới hạn ở các nhãn từ một lớp duy nhất được nhúng với nhiễu làm đầu vào cho generator (G)

*Hình 2.2: Sơ đồ EVAGAN*

* Hình 2.2 cho thấy kiến trúc chi tiết của EVAGAN. Có ba loại mũi tên được tô màu khác nhau được hiển thị trong hình.
* Các mũi tên màu đỏ thể hiện bước huấn luyện đầu tiên trong đó chỉ có D được huấn luyện.
* Các mũi tên màu xanh lá cây mô tả bước huấn luyện thứ hai cho G.
* Các mũi tên màu cam cho thấy sự tham gia của các đầu vào chung (minority class và noise) và các đầu ra (real/fake và minority class) cho cả hai bước huấn luyện đã được đề cập trước đó.
* Discriminator D của EVAGAN có ba đầu ra khác nhau để ước lượng majority, minority và real/fake. Các hàm sigmoid đã được sử dụng cho ba đầu ra này, mỗi đầu ra đều sử dụng hàm mất mát binary cross-entropy (BCE). Các hàm mất mát cũng được đề cập trong các phần phụ tương ứng của G và D.
* Hình 3 cho thấy một hộp được viền đỏ ở phía bên phải để minh họa ba ước lượng xác suất khác nhau làm đầu ra từ D. Ba ước lượng này được sử dụng để tính toán mất mát của D trong bước huấn luyện đầu tiên của EVAGAN.
* Một hộp được viền xanh lá cây, bao gồm ước lượng real/fake và ước lượng minority, tính toán G\_Loss để phản hồi lại cho G trong quá trình lan truyền ngược của huấn luyện mô hình kết hợp (bước thứ hai của huấn luyện EVAGAN).

\* Lưu ý rằng đầu ra của D được phân phối qua ba đơn vị sigmoid khác nhau để tách biệt các xác suất của mỗi lớp, tức là majority, nguồn (real/fake), và minority. Các ước lượng majority và minority có thể được kết hợp sử dụng một hàm sigmoid duy nhất. Tuy nhiên, giữa chúng có ba lợi ích.

* Thứ nhất, để tránh mất mát của lớp đa số phản hồi lại cho G.
* Thứ hai, nó đơn giản hóa mô hình mà không tốn thêm chi phí huấn luyện.
* Thứ ba, chúng ta có thể dễ dàng tách biệt các dự đoán cho các mẫu của bộ kiểm tra.
* Đối với mô hình D của EVAGAN, đã tách biệt các ước lượng lớp đa số và thiểu số bằng cách sử dụng hai hàm sigmoid (σ) khác nhau.
* Lợi ích của việc tách biệt các ước lượng lớp đa số và thiểu số là có thể phản hồi chỉ ước lượng lớp thiểu số về cho G.
* Lợi ích khác của cấu trúc này là có thể tính toán riêng biệt các ước lượng của cả hai lớp trên các tập dữ liệu kiểm tra để so sánh với mô hình ACGAN.

## 2.4. So sánh mô hình ACGAN và EVAGAN

**So sánh mô hình ACGAN và EVAGAN:**

|  |  |
| --- | --- |
|  |  |

*Hình 2.3: So sánh EVAGAN và ACGAN*

\* Điểm khác biệt:

- Đầu vào của bộ sinh mô hình EVAGAN chỉ có mẫu ngẫu nhiên, trong khi đầu vào của bộ sinh mô hình ACGAN có cả mẫu ngẫu nhiên và nhãn của tất cả các lớp.

- Đầu vào của bộ phân biệt mô hình EVAGAN có nhận nhãn của lớp thiểu số, trong khi bộ phân biệt mô hình ACGAN thì không.

- Đầu ra của bộ phân biệt mô hình EVAGAN trả về kết quả phân biệt mẫu thật giả và kết quả phân loại lớp thiểu số, đa số, trong khi đầu ra của bộ sinh mô hình ACGAN trả về tất cả các lớp phân loại.

- Do kết quả trả về của lớp phân loại khác nhau, nên ứng dụng của hai mô hình cũng khác nhau, mô hình EVAGAN được sử dụng trong phân loại nhị phân, trong khi mô hình ACGAN sử dụng trong phân loại đa lớp.

- Mô hình EVAGAN phát huy tối đa khả năng của nó trong điều kiện mất cân bằng dữ liệu, trong khi ACGAN không thể, nếu nhận đầu vào là nhãn của tất cả các lớp, có thể sẽ khiến hiệu suất phân loại giảm, gây nhầm lẫn nhiều cho mô hình.

\* Điểm chung của hai mô hình:

- Về cơ bản, cả hai mô hình ACGAN và EVAGAN đều là các biến thể của mạng sinh đối kháng GANs (Generative Adversarial Networks)

- Trong kiến trúc, mô hình ACGAN và EVAGAN cũng có những điểm tương đồng như sau: Kiến trúc chính của cả hai mô hình đều có bộ sinh (Generator) và bộ phân biệt (Discriminator) do thiết kế của EVAGAN được lấy cảm hứng từ kiến trúc mô hình ACGAN, hay nói cách khác, EVAGAN là một biến thể của mô hình ACGAN.

- Hai mô hình này cũng có quá trình huấn luyện đối kháng, trong đó, bộ sinh (G) cố gắng tạo ra các mẫu ngày càng giống thật để đánh lừa bộ phân biệt (D) và bộ phân biệt D sẽ cố gắng phân biệt các mẫu thật/giả. EVAGAN và ACGAN đều dùng loss function để tối ưu hóa quá trình huấn luyện, trong đó bộ phân loại (D) cố gắng tối đa hóa khả năng phân biệt, còn bộ sinh (G) cố gắng tối thiểu hóa khả năng bị bộ phân biệt (D) phân biệt.

- Bộ sinh (G) của cả hai mô hình đều nhận các mẫu ngẫu nhiên làm đầu vào.

- Bộ phân biệt (D) của cả hai mô hình đều trả về kết quả phân biệt mẫu thật/giả.

\*Ưu điểm khi so EVAGAN với ACGAN:

- EVAGAN được thiết kế đặc biệt để giải quyết các vấn đề liên quan đến lớp thiểu số, giúp cải thiện khả năng phân loại các mẫu thuộc lớp thiểu số mà thường bị bỏ qua trong các mô hình truyền thống.

- EVAGAN hoạt động tốt trong các bài toán phân loại nhị phân, đặc biệt khi lớp thiểu số có rất ít mẫu. Điều này giúp tối ưu hóa việc phát hiện các mẫu thuộc lớp thiểu số.

- Do chỉ tập trung vào một lớp thiểu số cụ thể và lớp đa số, EVAGAN có thể giảm bớt phức tạp trong quá trình huấn luyện, dẫn đến thời gian huấn luyện ngắn hơn và yêu cầu tài nguyên tính toán ít hơn.

- Do tập trung vào lớp thiểu số, EVAGAN thường có hiệu suất cao hơn trong việc phát hiện các mẫu thuộc lớp thiểu số so với ACGAN.

\* Nhược điểm:

- EVAGAN chủ yếu được thiết kế cho các bài toán phân loại nhị phân, do đó không phù hợp cho các bài toán phân loại đa lớp.

- Việc mở rộng EVAGAN cho các bài toán đa lớp phức tạp có thể gặp nhiều thách thức và yêu cầu điều chỉnh mô hình đáng kể.

- Do tập trung vào một lớp cụ thể, các mẫu tạo ra bởi EVAGAN có thể thiếu sự đa dạng so với ACGAN, làm giảm tính tổng quát của mô hình.

## 2.5. Dataset

* Các bộ dữ liệu được sử dụng:
* CC botnet: ISCX-2014, CIC-2017 CIC-2018.
* CV MNIST.
* Công cụ trích xuất các đặc trưng: CICFlowMeter-v4.
* Sử dụng tập hợp đặc trưng nonredundant để phân loại và dùng công cụ CICFlowMeter-v4 để trích xuất 80 luồng và đặc trưng Date-time từ các tập tin .pcap.
* Bộ dữ liệu ISCX-2014 được gán nhãn trước khi thực hiện tiền xử lý (do chưa được gán nhãn trước đó). Khi thực hiện tiền xử lý cho bộ dữ liệu này, đã loại bỏ các yếu tố dẫn đến sự xuất hiện các giá trị ngoại lệ cũng như các đặc trưng không cần thiết.
* CIC-IDS2017 và CIC-IDS2018 đã được tiền xử lý sau khi trích xuất các tập hợp con không cân bằng.
* Tỉ lệ phân chia tập train – test trong bài báo: 70% - 30%
* Bộ dữ liệu MNIST:Bộ dữ liệu MNIST là một bộ sưu tập đơn giản các chữ số viết tay từ 0 đến 9 để huấn luyện và kiểm tra các thuật toán ML khác nhau. Mục đích của việc sử dụng bộ dữ liệu này là để đánh giá hiệu suất của EVAGAN so với ACGAN về chất lượng hình ảnh sinh ra trong các tình huống cân bằng và không cân bằng.

# 

# 

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 3.1. Vấn đề:

Bài báo này tập trung giải quyết vấn đề cải thiện hiệu suất của các mô hình phân loại trong việc phát hiện các mẫu né tránh được tạo ra bởi các mạng sinh đối kháng (GAN). Cụ thể, các mẫu né tránh là những mẫu được tạo ra nhằm đánh lừa các hệ thống phân loại, khiến chúng phân loại sai, gây khó khăn trong việc phân biệt các luồng dữ liệu bình thường và bất thường. Bài báo giới thiệu mô hình EVAGAN, một biến thể của ACGAN, nhằm phát triển một mô hình phân loại có khả năng tự cải thiện hiệu suất thông qua việc đào tạo trên các mẫu né tránh này. Mục tiêu chính của EVAGAN là tạo ra một bộ phân loại mạnh mẽ hơn, có khả năng phân biệt tốt hơn giữa các mẫu thật và giả, các lớp thiểu số và đa số, cũng như tăng khả năng thích nghi cho bộ phân loại trong môi trường thực tế, vốn dĩ đã có sự chênh lệch về số lượng giữa các loại dữ liệu rất lớn.

Những vấn đề cụ thể mà EVAGAN giải quyết bao gồm:

* Phát hiện và xử lý mẫu né tránh: Những mẫu né tránh thường được thiết kế để đánh lừa các mô hình phân loại hiện có. Các hệ thống phân loại truyền thống thường gặp khó khăn trong việc phân biệt các mẫu này do chúng được thiết kế rất giống với các mẫu thật, gây nhầm lẫn cho các bộ loại của mô hình trong việc phân biệt mẫu thật giả, dẫn đến hiệu suất kém.
* Cải thiện độ chính xác phân loại: Bằng cách sử dụng các mẫu né tránh để train, mô hình phân loại có thể học được các đặc điểm của các mẫu giả mạo và trở nên chính xác hơn. Với cấu trúc chính gồm bộ sinh và bộ phân loại, mô hình vừa có thể học các đặc điểm của mẫu thật để tạo ra mẫu giả, vừa có thể dùng các mẫu đó làm dữ liệu để train cho cho bộ phân loại, kết quả sau khi thực hiện phân biệt mẫu lại tiếp tục được dùng để cập nhật cho bộ sinh và bộ phân loại, nhằm đảm bảo các mẫu được tạo ra từ bộ sinh giống mẫu thật nhất có thể, đồng thời cũng tăng khả năng nhận biết mẫu thật, mẫu giả, nâng cao hiệu suất cho bộ phân loại.
* Tách biệt các lớp đa số và thiểu số: Bộ phân biệt D của EVAGAN có ba đầu ra khác nhau để ước lượng xác suất của các lớp đa số, thiểu số và mẫu thật/giả. Các đầu ra này được phân biệt bằng các hàm sigmoid khác nhau, giúp mô hình tránh việc mất mát của lớp đa số ảnh hưởng đến quá trình training của bộ sinh. Điều này không chỉ giảm thiểu chi phí training mà còn đơn giản hóa mô hình và giúp dễ dàng tách biệt các dự đoán cho các mẫu kiểm tra.

Mô hình EVAGAN có cấu trúc bộ sinh và bộ phân loại được thiết kế đặc biệt, điều này giúp nâng cao khả năng của mô hình phân loại này trong việc đối phó với các mẫu né tránh, từ đó cải thiện hiệu suất tổng thể của hệ thống phân loại.

## 3.2. Phương pháp:

Mô hình EVAGAN có hai thành phần chính là bộ sinh dữ liệu và bộ phân loại, nhằm phục vụ cho mục tiêu có nhiều số lượng mẫu giả hơn để đa dạng hóa, ngẫu nhiên hóa bộ dữ liệu, tăng tỉ lệ phân loại đúng cho mô hình, cũng như giúp cho bộ phân loại được training tốt hơn.

Bộ sinh dữ liệu sử dụng các hàm mục tiêu để tối thiểu hóa log-likelihood của các mẫu giả được phân loại là giả bởi bộ phân loại, cũng như cải thiện log-likelihood của các mẫu lớp thiểu số từ bộ sinh vào bộ phân loại. Log-likelihood được sử dụng trong hàm mục tiêu của bộ sinh để tối ưu hóa quá trình sinh dữ liệu, đảm bảo rằng các mẫu dữ liệu giả được tạo ra sẽ có độ tương tự cao với các mẫu dữ liệu thật. Để đo được sự tương đồng giữa hai phân phối xác suất p(x) và q(x) cho các sự kiện x trong không gian mẫu X, mô hình EVAGAN sử dụng hàm Cross-Entropy Loss CE(p,q). Kết quả hàm này là mức độ khác biệt giữa hai phân phối xác suất Khi hai phân phối xác suất giống nhau hoàn toàn, tức là p(x) = q(x), kết quả sẽ bằng 0, thể hiện sự giống nhau tuyệt đối giữa hai phân phối xác suất, ngược lại, nếu hai phân phối xác suất khác nhau, kết quả của hàm Cross-Entropy Loss sẽ là một số không âm khác 0. Bộ sinh của mô hình EVAGAN sử hàm mất mát để tính toán mất mát cho bộ sinh, giúp định hình quá trình training cho bộ sinh dữ liệu. Công thức này tổng hợp mất mát của bộ sinh bằng cách kết hợp hai thành phần: Phần thứ nhất đo lường mức độ "tự tin" của bộ sinh rằng các mẫu giả được tạo ra là giả; Phần thứ hai đo lường mức độ "tự tin" của bộ sinh rằng các mẫu thuộc lớp thiểu số. Đây là một phần quan trọng của quá trình huấn luyện mô hình, giúp định hình việc cập nhật trọng số của bộ sinh dữ liệu để tối ưu hóa hiệu suất của nó.

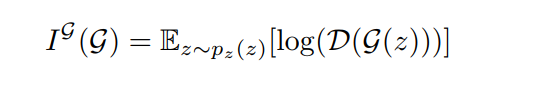
Bộ phân biệt sử dụng hàm Cross-Entropy Loss để đánh giá độ tương thích giữa các mẫu thật thuộc lớp đa số và các dự đoán của bộ phân loại, phân phối của các lớp đa số từ các mẫu thật cần được đánh giá chính xác, do bộ sinh không tạo ra các mẫu thuộc lớp đa số, đây được gọi là giai đoạn Ước lượng phân phối lớp đa số trong quá trình phân loại. Ở giai đoạn Phát hiện và phân biệt các mẫu dữ liệu giả, bộ phân loại cần phát hiện và phân biệt các mẫu dữ liệu giả được tạo ra bởi bộ sinh và đánh giá tính hợp lý của chúng, bằng cách tính toán log-likelihood cho cả các mẫu thật và giả thuộc lớp thiểu số và sử dụng hàm Cross-Entropy Loss để đánh giá sự khác biệt giữa các log-likelihood này. Cuối cùng, ở giai đoạn Ước lượng phân phối lớp thiểu số, mô hình sử dụng hàm cross-entropy loss để đánh giá sự phù hợp giữa các mẫu thật và giả thuộc lớp thiểu số với các dự đoán của bộ phân loại.

## 3.3. Kiến trúc tổng quan:

Mô hình EVAGAN được thiết kế dựa trên cấu trúc của ACGAN với mục tiêu phát triển một mô hình phân loại có khả năng tự cải thiện trên các mẫu né tránh được tạo ra bởi GAN. Cấu trúc chính của EVAGAN bao gồm hai mạng neural: bộ sinh (G) và bộ phân loại (D). Trái ngược với ACGAN, mô hình của EVAGAN được hạn chế chỉ nhận nhãn từ một lớp duy nhất được nhúng với nhiễu để làm đầu vào cho bộ sinh (G). Điều này giúp giảm độ phức tạp của mô hình và tăng tính hiệu quả trong việc tạo ra các mẫu dữ liệu. Ở bộ sinh dữ liệu, (G) nhận dữ liệu gây nhiễu và nhãn lớp đơn làm đầu vào và sử dụng mạng neural để tạo ra các mẫu dữ liệu giả, mô phỏng mẫu thật. Ở bộ phân loại, (D) phân biệt giữa các mẫu thật (từ dataset có sẵn) và giả (dữ liệu được nhận từ bộ sinh (G)), sau đó áp dụng các hàm tính toán để ước lượng xác suất cho lớp đa số, lớp thiểu số và trạng thái thật/giả của mẫu.

### 3.3.1. Bộ sinh dữ liệu (Generator):

Bộ sinh trong mô hình EVAGAN có chức năng tạo ra các mẫu dữ liệu giả mô phỏng mẫu thật dựa trên nhiễu và nhãn lớp đầu vào. Kiến trúc của bộ sinh được thiết kế để tối ưu hóa việc tạo ra các mẫu dữ liệu giả sao cho chúng khó phân biệt được với mẫu thật. Ở đầu vào, bộ sinh nhận vào một vector nhiễu ngẫu nhiên và một nhãn lớp đơn, được nhúng vào lớp đầu vào của mạng. Kiến trúc của bộ sinh dữ liệu gồm 3 phần chính: Tối thiểu hóa log-likelihood của các mẫu giả được phân loại sai là giả bởi bộ phân biệt (discriminator), cải thiện log-likelihood của các mẫu lớp thiểu số được tạo ra từ bộ sinh và được đưa vào bộ phân biệt.  
Hàm mục tiêu để tối thiểu hóa log-likelihood của các mẫu giả được phân loại sai là giả bởi bộ phân biệt:



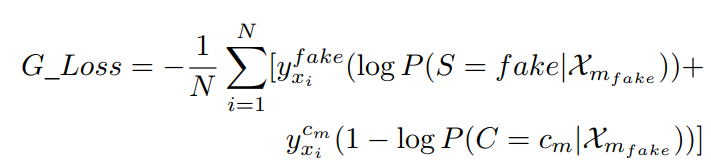
Hàm này được sử dụng để đo lường mức độ mà bộ sinh (G) có thể đánh lừa được bộ phân biệt (D). Điều này được thực hiện bằng cách tính log-likelihood trung bình của các mẫu giả G(z) được tạo ra từ vector nhiễu z lấy mẫu từ phân phối pz(z) sau đó đưa vào bộ phân biệt để phân loại. Mục tiêu của bộ sinh là tối thiểu hóa giá trị này, tức là làm tăng log-likelihood của các mẫu giả được phân loại là thật bởi bộ phân biệt.

Hàm mục tiêu để cải thiện log-likelihood của các mẫu lớp thiểu số được tạo ra từ bộ sinh và được đưa vào bộ phân biệt:



Hàm này đảm bảo rằng các mẫu từ lớp thiểu số được tạo ra bởi G có log-likelihood cao khi đưa vào D. Điều này đảm bảo rằng D sẽ có khả năng phân loại đúng các mẫu từ lớp thiểu số, giúp cải thiện hiệu suất của hệ thống phân loại. Vì vậy hàm mục tiêu này đóng vai trò quan trọng trong quá trình đào tạo của G trong EVAGAN bằng cách đảm bảo rằng các mẫu lớp thiểu số được tạo ra có khả năng phân loại tốt khi đưa vào D, từ đó tăng cường khả năng phân loại của hệ thống tổng thể.

Hàm mất mát:



Này này gồm 2 phần chính, phần đầu tiên (y fake) đo lường mức độ mất mát khi các mẫu giả được phân loại sai như mẫu thật bởi (D). Mục tiêu là giảm thiểu việc (D) phân loại các mẫu giả là thật. Phần hai (y xcmi) đo lường mức độ mất mát khi (D) phân loại các mẫu lớp thiểu số sai hoặc đúng không đủ.

### 3.3.2. Bộ phân biệt dữ liệu (Discriminator):

Mô hình Phân biệt (D) trong EVAGAN đóng vai trò quan trọng trong việc phân biệt giữa các mẫu thật và giả, cũng như ước lượng các lớp liên quan đến các mẫu. Nó sử dụng một kiến trúc phức tạp phân tách ước lượng của các lớp đa số và lớp thiểu số, sử dụng các hàm sigmoid khác nhau cho mỗi lớp.

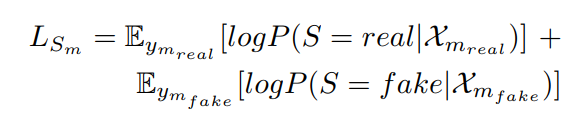
Bằng cách phân tách ước lượng của các lớp đa số và lớp thiểu số, EVAGAN tạo điều kiện cho phản hồi được hướng tới bộ sinh (G) về ước lượng của lớp thiểu số, tăng cường khả năng của mô hình tạo ra các mẫu thực tế từ lớp thiểu số. Ngoài ra, sự phân tách này cho phép đánh giá độc lập của cả hai lớp trên các tập dữ liệu kiểm tra, tạo điều kiện cho so sánh với mô hình ACGAN.

Hàm mục tiêu của D bao gồm ba thành phần được biểu diễn bằng 3 hàm đầu tiên dưới đây.

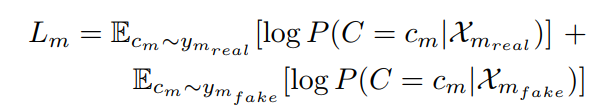
Công thức L\_M, tập trung vào việc ước lượng chính xác phân phối của lớp đa số từ các mẫu thực.



Công thức L\_Sm, đại diện cho việc ước lượng log-tích lũy cho cả các mẫu thực và giả của lớp thiểu số. Công thức này đo lường khả năng của Phân biệt để phân biệt giữa các mẫu thực và giả của lớp thiểu số.



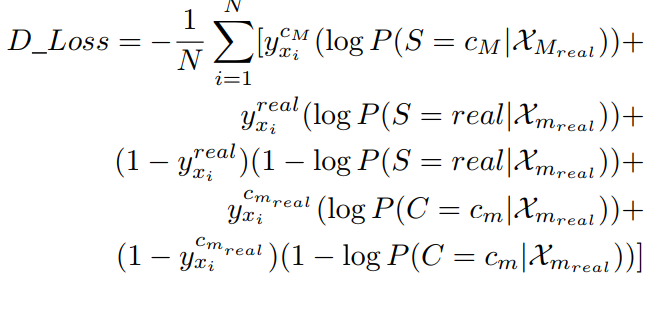
Công thức Lm, tổng hợp các ước lượng log-tích lũy cho cả các mẫu thực và giả của lớp thiểu số, tổng hợp hiệu suất của bộ phân biệt trong việc phân biệt các lớp khác nhau trong lớp thiểu số.



Do đó, hàm mục tiêu tổng thể của D, được biểu diễn bằng công thức L(D)là tổng của ba thành phần log-tích lũy này. Phân biệt mục tiêu tối đa hóa hàm mục tiêu này trong quá trình huấn luyện, tăng cường khả năng phân loại chính xác các mẫu.



Hơn nữa, hàm mất mát của D, được tạo ra từ công thức 5 và 6, được biểu diễn bằng công thức DLoss. Công thức này bao gồm ba hàm mất mát nhị phân cross-entropy riêng biệt liên quan đến lớp đa số, ước lượng nguồn gốc và ước lượng của lớp thiểu số. Đáng chú ý, thuật ngữ mất mát tương ứng với lớp đa số bị giả mạo được bỏ qua vì bộ sinh không tạo ra các mẫu của lớp đa số.



# CHƯƠNG 4: HIỆN THỰC HỆ THỐNG

## 4.1. Môi trường thực nghiệm

Việc training mô hình diễn ra trên google colab và visual studio code (VSCode) với sự hỗ trợ của các thư viện như Keras, TensorFlow, Sklearn và Numpy. Mỗi mô hình (ACGAN và EVAGAN) đều được huấn luyện với 2 loại dataser là CC botnet và CV Dataset để đáp ứng 2 mục tiêu của bài báo.

## 4.2. Dataset

Vì kịch bản triển khai trong môi trường mạng đối kháng nên chúng ta sẽ dùng dataset CC botnet để phân tích định tính nhằm đánh giá hiệu suất của mô hình trong việc tạo ra dữ botnet giả. Ngoài ra CV MNIST cũng được sử dụng nhằm để đánh giá hiệu suất của EVAGAN so với ACGAN về chất lượng hình ảnh được tạo ra trong các trường hợp cân bằng và mất cân bằng mẫu.

# 4.2.1. CC botnet Dataset

Dữ liệu để huấn luyện mô hình sẽ lấy từ cả 3 bộ dữ liệu: ISCX-2014, CIC-2017 và CIC-2018 từ Canadian Institute of Cybersecurity (CIC)

+ ISCX-2014 dataset: là bộ dữ liệu kết hợp từ 3 bộ dữ liệu phổ biến khác là ISOT, ISCX 2012 IDS và CTU 13. Tuy nhiên chúng ta đang cần tìm các mẫu hiếm, có số lượng ít nên chỉ các mẫu Virus botnet được chọn để training.

+ CIC-IDS2017 dataset: là bộ dữ liệu về các cuộc tấn công phổ biến và giống với thực tế. Nó bao gồm các kết quả phân tích lưu lượng truy cập mạng bằng CICFlowMeter với các luồng được gắn nhãn dựa trên dấu thời gian, IP nguồn và IP đích, cổng nguồn và đích, giao thức và tấn công. Từ bộ dữ liệu này chúng ta sẽ chọn botnet Ares làm mẫu huấn luyện.

+ CIC-IDS2018 dataset: bộ dữ liệu này được dùng để tạo 1 bộ dữ liệu con từ 1 bộ dữ liệu không cân bằng với mục đích phân tích. Nó sẽ chỉ chứa botnet Ares và Zeus, còn lại là các mẫu lành tính.

## 4.2.2. CV Dataset

MNIST Dataset: một bộ dữ liệu lớn chứa các chữ số viết tay thường được dùng trong việc huấn luyện các hệ thống xử lý hình ảnh khác nhau. Chúng ta sẽ chỉ dùng bộ dữ liệu này nhằm để so sánh chất lượng hình ảnh mà EVAGAN và ACGAN tạo ra trong trường hợp mất cân bằng dữ liệu hoặc cân bằng.

## 4.3. Tiền xử lý dữ liệu

Vì bộ dữ liệu ISCX-2014 chưa được gắn nhãn để sử dụng trong các thử nghiệm dựa trên ML nên chúng ta sẽ sử dụng thông tin được cung cấp trên trang web CIC dành cho các IP được liên kết với các botnet cụ thể để gắn nhãn cho tập dữ liệu. Sau khi đã dán nhãn, chúng ta sẽ thực hiện tiền xử lý cho bộ dữ liệu này như: làm sạch dữ liệu, loại bỏ giá trị ngoại lệ bằng cách bỏ đi tất cả các giá trị sai lệch cao và thấp.

CIC-IDS2017 và CIC-IDS2018 đều được gán nhãn sẵn nên chúng ta chỉ việc thực hiện làm sạch dữ liệu, loại bỏ giá trị ngoại lệ.

Trong khâu trích xuất đặc trưng, chúng ta sẽ giữ lại phần lớn đặc trưng bởi việc giảm thiểu đặc trưng không giúp nâng cao hiệu suất mô hình lên. Ngược lại việc dùng nhiều đặc trưng sẽ giúp mô hình dễ phát hiện botnet hơn.

# CHƯƠNG 5: THỰC NGHIỆM VÀ ĐÁNH GIÁ

## 5.1. Thực Nghiệm

## 5.1.1. Mục tiêu

Đánh giá hiệu suất của EVAGAN trong việc tạo ra dữ liệu giả mạo có thể làm nhiễu cho các mô hình phát hiện xâm nhập.. Với mục tiêu này nhóm chỉ sử dụng tập dữ liệu CC botnet để huấn luyện các mô hình

## 5.1.2. Triển khai

Mỗi mô hình được huấn luyện trong một số lượng epoch nhất định trên tập huấn luyện và sau đó được đánh giá trên tập kiểm tra. Đo lường các chỉ số định lượng, bao gồm GEN VALIDITY, FAKE BOT EVA, REAL NORMAL EST và REAL BOT EVA.

* GEN VALIDITY (generated samples validity): Chỉ số này đo lường mức độ mà các mẫu dữ liệu được tạo ra bởi Generator trong mô hình GAN có giống thực không. Chỉ số này được đo bằng cách tính tỷ lệ giá trị của các mẫu được dự đoán trong Real/Fake Estimation so với tổng số mẫu được dự đoán. Giá trị của GEN VALIDITY càng gần với 1, tức là càng tốt, cho thấy rằng mô hình Generator đang tạo ra các mẫu dữ liệu giống với dữ liệu thực. Bởi vì nhãn của lớp normal/majority là “1” (tức lớp này sẽ được gán giá trị 1) nên chỉ số GEN VALIDITY càng gần với “1” thì mẫu mà G tạo ra càng giống thực.



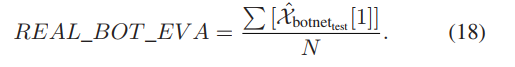
* FAKE BOT EVA (fake/generated botnet samples evasion): Chỉ số này đo lường mức độ mà các mẫu botnet được tạo ra bởi mô hình G có thể trốn tránh được D. Chỉ số này được đo bằng cách tính tỷ lệ của các mẫu được dự đoán trong Minority Class Estimation so với tổng số mẫu được dự đoán bởi D. Giá trị của FAKE BOT EVA càng gần 0, thì khả năng trốn tránh của mẫu do G tạo ra càng thấp. Bởi vì nhãn của lớp botnet/minority là “0” (tức lớp này sẽ được gán giá trị 0)



* REAL NORMAL EST (real normal/majority class estimation): Chỉ số này đo lường mức độ mà mô hình GAN đang ước lượng dữ liệu thực ở lớp bình thường (hay lớp đa số) một cách chính xác. Chỉ số này được đo bằng cách tính tỷ lệ của các mẫu được dự đoán trong Majority Class Estimation so với tổng số mẫu được dự đoán ***(lưu ý các mẫu normal/majority sẽ được lấy từ tập test )***. Giá trị của REAL NORMAL EST càng gần 1, tức là càng tốt, cho thấy rằng mô hình D phân biệt lớp normal/majority có độ chính xác cao.



* REAL BOT EVA (real botnet/minority class evasion Chỉ số này đo lường mức độ mà các mẫu botnet thật được lấy từ Real Data có thể trốn tránh được D. Điều này thường được đo bằng cách tính tỷ lệ của các mẫu được dự đoán trong Minority Class Estimation so với tổng số mẫu được dự đoán ***(lưu ý các mẫu botnet/minority sẽ được lấy từ tập test ).*** Giá trị của REAL\_BOT\_EVA càng gần 0, tức là càng tốt, cho thấy rằng mô hình GAN đang tạo ra các mẫu botnet khó bị phân biệt với dữ liệu thực.



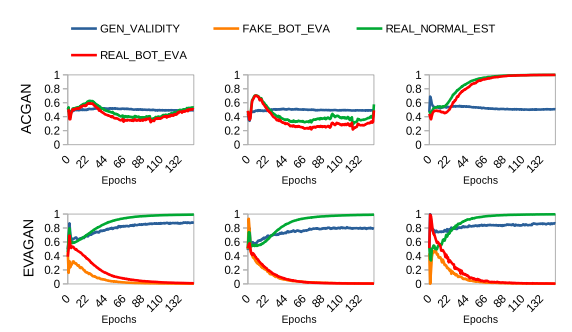
Việc đo lường các chỉ số này nhằm đánh giá hiệu suất của các mô hình GANs, cụ thể là mô hình ACGAN và EVAGAN, khi được áp dụng vào việc tạo ra dữ liệu mới và ước lượng các lớp dữ liệu từ tập dữ liệu gốc. Thông qua các chỉ số này, chúng ta có thể:

* Đánh giá tính hợp lệ của các mẫu dữ liệu được tạo ra bởi mô hình GAN.
* Đánh giá khả năng của mô hình GAN tạo ra các mẫu botnet khó bị phát hiện (tránh né).
* Ước lượng hiệu suất của mô hình GAN trong việc ước lượng các lớp dữ liệu thực và bình thường.
* Đánh giá khả năng của mô hình GAN tránh được các biện pháp phân loại khi đối mặt với các mẫu botnet thực sự từ tập dữ liệu.

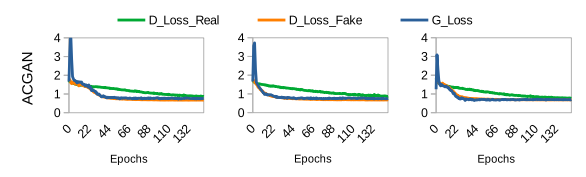
**5.1.3. Đánh giá kết quả thực nghiệm**

**- Mô hình ACGAN với tập dataset CC botnet (ACGAN CC)**

**Kết quả trong bài báo chính**

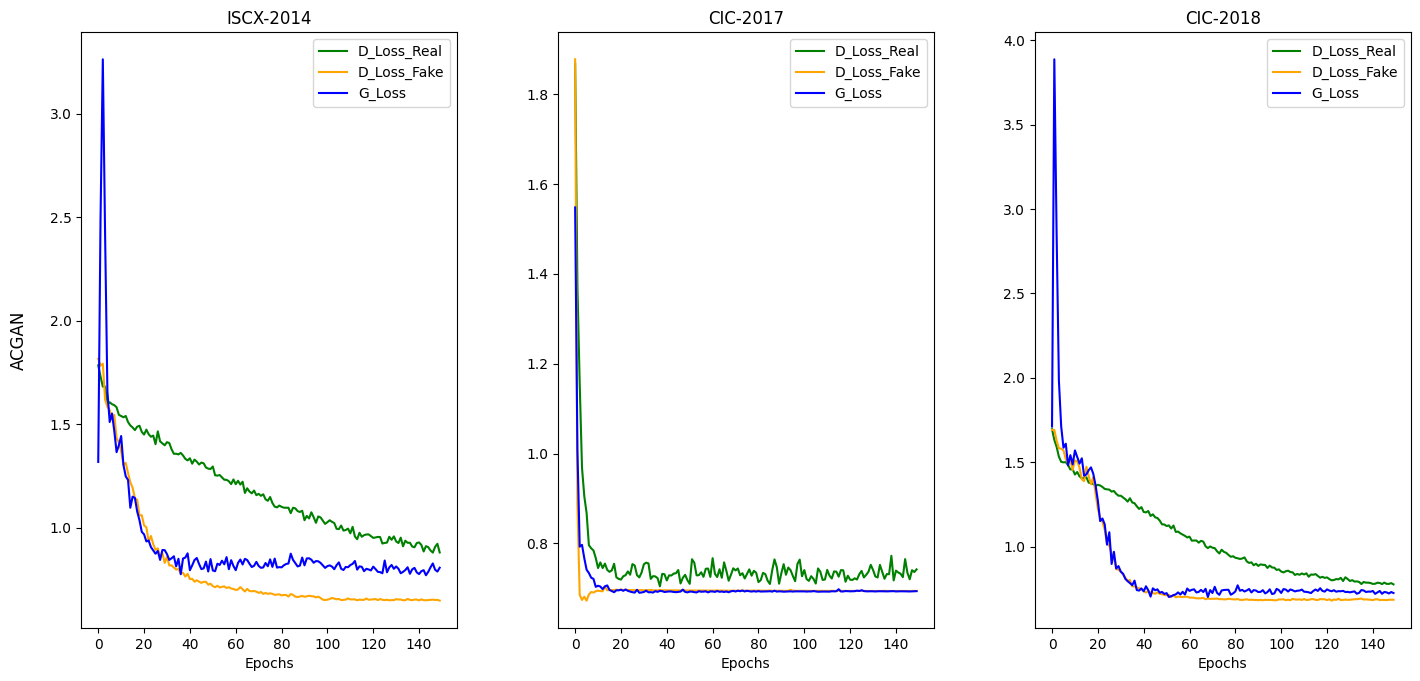
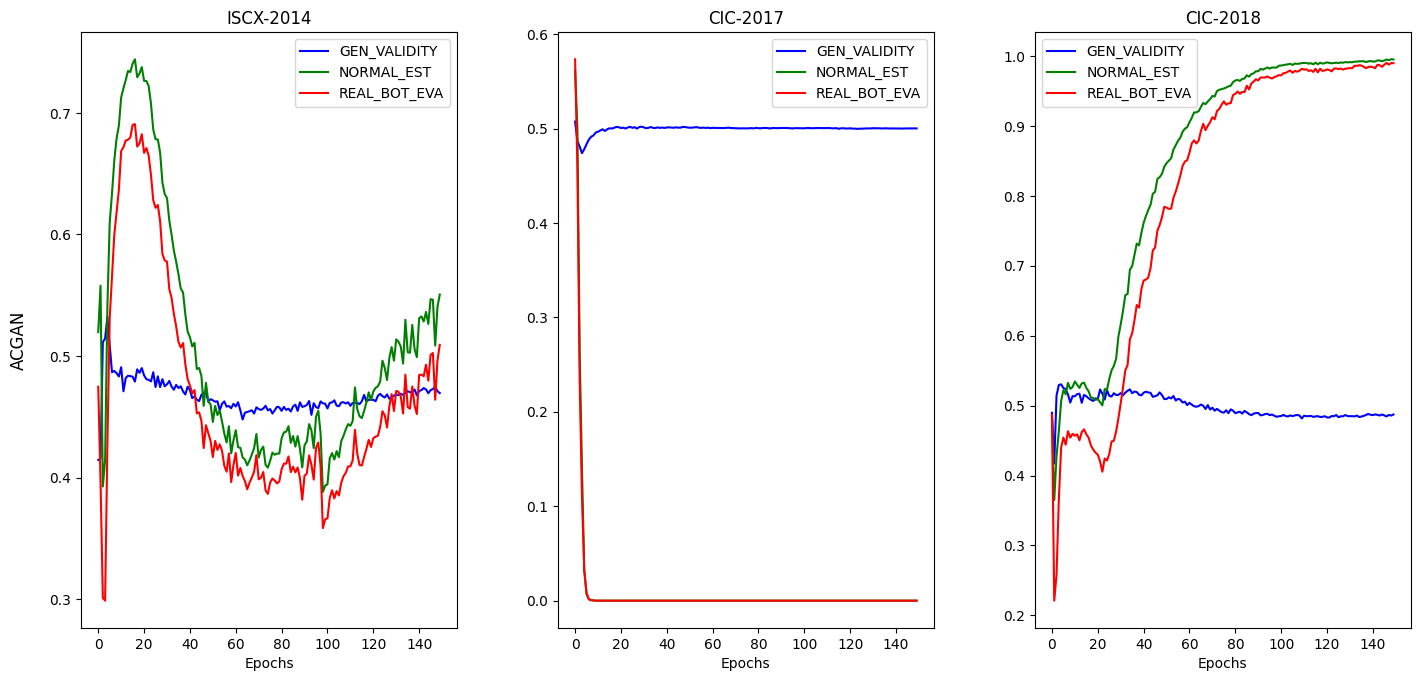


Hình 5.1: Các chỉ số mà báo đạt được để đánh giá hiệu suất làm việc của ACGAN\_CC



Hình 5.2: Các giá trị của hàm Loss mà Báo đạt được ứng với mỗi mô hình ACGAN\_CC

**Kết quả nhóm đạt được**

**** Hình 5.3: Các chỉ số mà Nhóm đạt được để đánh giá hiệu suất làm việc của ACGAN\_CC

Hình 5.4: Các giá trị của hàm Loss mà Báo đạt được ứng với mô hình ACGAN\_CC

**Nhận xét:**

* GEN VALIDITY: Giá trị của chỉ số này duy trì gần 0.5 trên cả ba tập dữ liệu CC, Điều này cho thấy mô hình Discriminator (D) bối rối trong việc quyết định xem các mẫu được tạo từ G là thật hay giả.
* FAKE BOT EVA: Do ACGAN CC tạo ra hai lớp dữ liệu nên FAKE BOT EVA không thể được đánh giá một cách xác định bởi mẫu được tạo ra từ cả hai lớp, dẫn đến giá trị không xác định của chỉ số này.

Do đó trong mô hình ACGAN này, chúng ta sẽ không đo lường chỉ số này

* REAL NORMAL EST và REAL BOT EVA: Các giá trị của chúng khá gần nhau.

Kết quả này cho thấy D của ACGAN CC không thể phân biệt giữa các lớp majority và lớp minority một cách chính xác, đặc biệt là đối với lớp minority do vấn đề về mất cân bằng dữ liệu.

* D Loss: Trong ACGAN CC, do dữ liệu mất cân bằng, mô hình D có thể gặp khó khăn trong việc phân loại chính xác giữa các lớp.

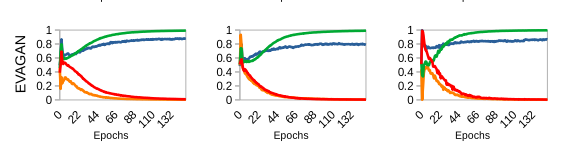
Giá trị D\_\_LOSS\_REAL cao hơn D\_LOSS\_FAKE tại cùng thời điểm.

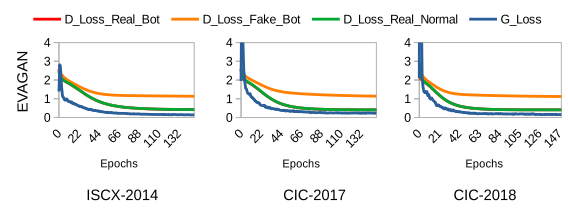
* G Loss: Mô hình G của ACGAN CC gặp khó khăn trong việc tạo ra các mẫu mới một cách hiệu quả, có thể do sự không đồng nhất trong dữ liệu và khó khăn trong việc huấn luyện để tái tạo các đặc điểm của lớp thiểu số.
* Từ đồ thị bên dưới, chúng ta có thể thấy giá trị của tất cả Loss của G và D đều hội tụ lại dưới 1. Điều này cho thấy GAN đang bão hòa theo hướng cân bằng Nash.

**- Mô hình EVAGAN với tập dataset CC botnet (EVAGAN CC)**

**Kết quả trong bài báo chính**

**Hình 5. 7 Các chỉ số mà Nhóm đạt được để đánh giá hiệu suất làm việc của EVAGAN\_CC**



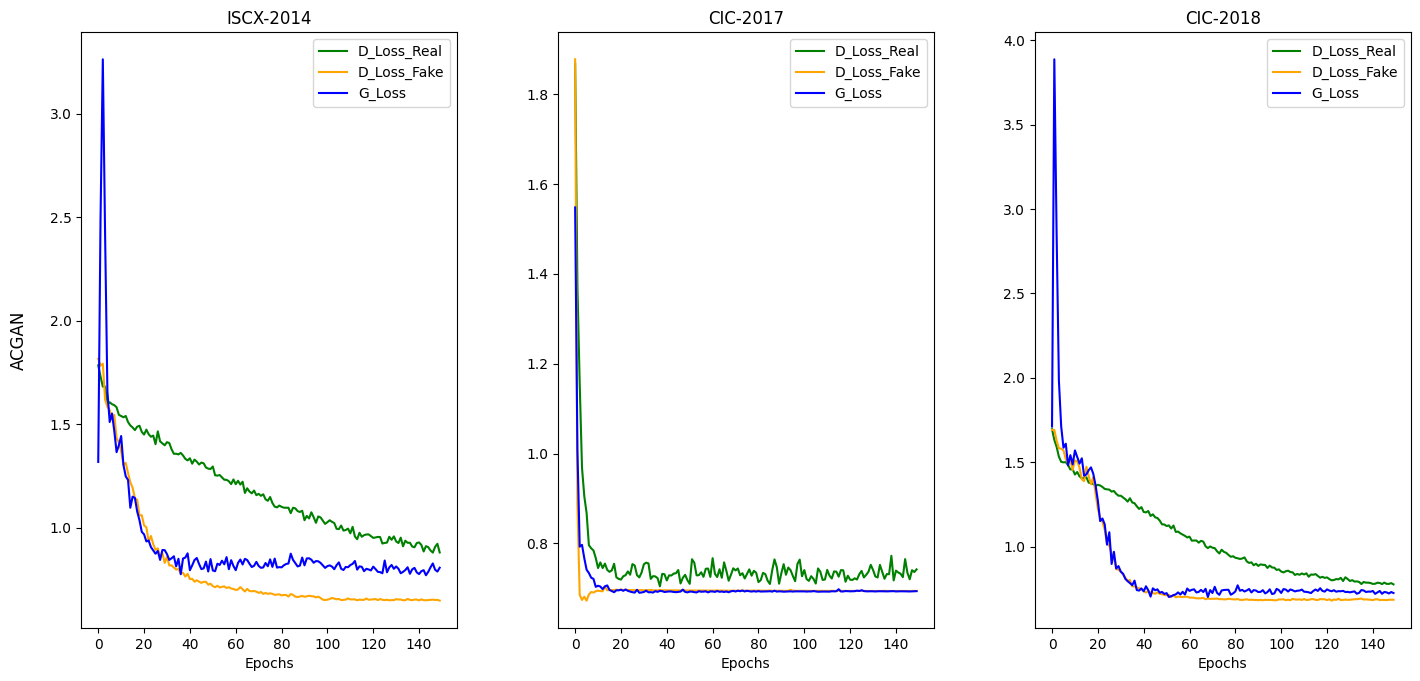
**** Hình 5.5: Các chỉ số mà báo đạt được để đánh giá hiệu suất làm việc của EVAGAN\_CC

Hình 5.6: Các giá trị của hàm Loss mà Báo đạt được ứng với mỗi mô hình EVAGAN\_CC

A group of graphs with different colored lines

Description automatically generated**Kết quả nhóm đạt được**

Hình 5.7: Các chỉ số mà Nhóm đạt được để đánh giá hiệu suất làm việc của EVAGAN\_CC

Hình 5. 8 Các giá trị của hàm Loss mà báo đạt được ứng với mô hình ACGAN\_CC

* GEN VALIDITY: Kết quả cho thấy hiệu suất của mô hình G ngày càng cải thiện qua mỗi epoch, mặc dù D đang tiếp tục học để cải thiện khả năng phân biệt thật giả nhưng vẫn bị đánh lừa bởi G.
* FAKE BOT EVA: Chỉ số này đánh giá khả năng của D phát hiện mẫu giả mạo từ mô hình G, và kết quả cho thấy EVAGAN CC có khả năng phát hiện giả mạo tốt hơn so với các bộ phân loại ML truyền thống.
* REAL NORMAL EST và REAL BOT EVA: D của EVAGAN CC có khả năng phân biệt giữa các lớp đa số và lớp thiểu số tốt hơn so với ACGAN CC, đặc biệt là đối với lớp thiểu số.
* D Loss: Trái ngược với ACGAN CC, mô hình D của EVAGAN CC có khả năng quản lý mất mát hiệu quả hơn trong việc phân biệt giữa các lớp đa số và lớp thiểu số. Điều này cho thấy sự cải thiện trong hiệu suất phát hiện của mô hình. Giá trị D\_\_LOSS\_REAL thấp hơn D\_LOSS\_FAKE tại cùng thời điểm.
* G Loss: Mô hình G của EVAGAN CC có thể tạo ra các mẫu mới một cách hiệu quả hơn, giúp cải thiện sự đa dạng và tính tự nhiên của các mẫu được tạo ra, từ đó giảm thiểu khả năng bị phát hiện.

Các giá trị Loss có xu hướng ổn định hơn theo từng epoch và đạt điểm thấp nhất sớm hơn ACGAN\_CC.

Cả ACGAN CC và EVAGAN CC đều đạt được sự cải thiện trong hiệu suất phát hiện và đối phó với các mẫu giả mạo từ mô hình G so với các phương pháp truyền thống.

Kết quả tự thực nghiệm cho thấy sự tương đồng đáng kể (giống 2/3 tập dataset) với kết quả được báo cáo trong bài báo. Mặc dù khoảng giá trị của các chỉ số đánh giá có thay đổi nhưng hình dạng phân phối thì không đổi.

Với dataset 2017, nhóm không tạo ra kết quả như bài báo được, nguyên nhân có thể là do có lỗi trong quá trình xử lý dữ liệu loại CIC-IDS2017.

Việc giá trị REAL\_NORMAL\_EST và REAL\_BOT\_EVA không đổi sau mỗi epoch là do kết quả dự đoán trong Minority Class Estimation và Majority Class Estimation không có thay đổi gì tức là với dataset này thì G không thể đánh lừa được D tốt hơn nữa.

## 5.2. Đánh giá

* Hiệu suất: EVAGAN cho thấy khả năng cải thiện hiệu suất so với ACGAN trong việc tạo ra dữ liệu giả mạo.
* Tạo mẫu Botnet: EVAGAN có khả năng tạo ra các mẫu botnet khó bị phát hiện hơn, giúp tăng cường khả năng đối phó với các mô hình phát hiện xâm nhập.
* Ước lượng lớp dữ liệu: EVAGAN thể hiện khả năng ước lượng lớp dữ liệu hiệu quả hơn so với ACGAN, giúp cải thiện khả năng phân loại và phát hiện xâm nhập.
* Tính ổn định: EVAGAN có thể quản lý mất mát của cả mô hình D và G hiệu quả hơn, dẫn đến sự cải thiện đáng kể trong hiệu suất phát hiện và đối phó với các mẫu giả mạo từ mô hình Generator.

# 

# CHƯƠNG 6: KẾT LUẬN VÀ PHƯƠNG HƯỚNG PHÁT TRIỂN

## 6.1. Kết luận

* Mô hình GAN mới có tên EVAGAN tạo ra sự trốn tránh đối nghịch trong chế độ dữ liệu thấp. EVAGAN là sự cải tiến của một mô hình nổi tiếng có tên ACGAN nhằm mục đích thiết kế một bộ phân loại nhận biết đối thủ để phát hiện sự bất thường. Nhóm đã sử dụng hai dataset: CC Botnet và CV MNIST.
* Bộ phân biệt đối xử của EVAGAN vượt trội hơn ACGAN về hiệu suất phát hiện, độ ổn định và độ phức tạp về thời gian. Đồng thời, phân tích định tính cho thấy EVAGAN hoạt động tốt hơn ACGAN trong các tình huống không cân bằng. Mô hình EVAGAN được thiết kế cho các bài toán phân loại nhị phân.

## 6.2. Phương hướng phát triển

* Thử nghiệm thêm mô hình EVAGAN với nhiều dataset khác để đánh giá thêm về hiệu suất và tìm hướng phát triển mới
* Nghiên cứu về thiết kế đa lớp được đánh giá là hướng nghiên cứu tiềm năng

**TÀI** **LIỆU** **THAM** **KHẢO**

[1] R. H. Randhawa, N. Aslam, M. Alauthman, H. Rafiq, and M. Khalid, "EVAGAN: Evasion Generative Adversarial Network for Low Data Regimes," *PREPRINT FOR JOURNAL OF IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE*, vol. 00, no. 0, tháng 2021

[2] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” in International conference on machine learning. PMLR, 2017, pp. 2642–2651.

**TÀI LIỆU CỦA NHÓM**

link Github: <https://github.com/rhr407/EVAGAN>

link báo cáo và code: https://drive.google.com/drive/folders/1v9f77f2dW6a-VDtGcXObZCLY1E84p4iG?fbclid=IwZXh0bgNhZW0CMTEAAR2KI0nr0T9HLf1z3eDrC8ZSNnXbeuIyWPCnU-8o50bPCxjnJudA9laBHt0\_aem\_ZmFrZWR1bW15MTZieXRlcw