VNUHCM - UNIVERSITY OF SCIENCE
Faculty of Mathematics & Computer Science

Báo cáo Seminar
Phát hiện gian lận tài chính trong thời gian thực

Sinh viên thực hiện:
21110320 Lê Công Khánh
21110310 Nguyễn Ngọc Huynh

Giảng viên hướng dẫn:
TS. Võ Đức Cẩm Hải

Ngày 12 tháng 1 năm 2025

Mục lục

1 Lí do chọn đề tài

1.1 Thực trạng hiện tại trong gian lận tài chính

- Gian lận tài chính là một hành vi cố ý, trái pháp luật, trái với các quy định hoặc chính sách nhằm mục đích có được lợi ích tài chính trái phép. Gian lận tài chính được chia thành bốn loại sau: gian lận ngân hàng, gian lận chứng khoán và gian lận hàng hóa, gian lận bảo hiểm và gian lận tài chính khác. Các loại gian lận tài chính trên có thể được phân loại cụ thể hơn, như: gian lận thẻ tín dụng, gian lận rửa tiền, gian lận bảo hiểm xe ô tô, gian lận bảo hiểm y tế, gian lận tiếp thị, gian lận doanh nghiệp...

- Hiện nay, gian lận tài chính ngày càng phổ biến và có thể gây ra những hậu quả kinh tế nghiêm trọng đối với các tổ chức, cá nhân và chính phủ. Do đó, việc phát hiện và ngăn chặn kịp thời gian lận tài chính đóng vai trò ngày càng quan trọng. Mục tiêu của phát hiện gian lận tài chính là tối đa hóa những dự đoán chính xác và duy trì những dự đoán không chính xác ở mức chấp nhận được. Nghĩa là khả năng không phát hiện được gian lận phải ở mức thấp nhất và tối thiểu tỷ lệ dự đoán các trường hợp không gian lận được dự đoán là gian lận. Từ đó, giúp các cơ quan, tổ chức sớm phát triển những chính sách và chiến lược phù hợp giảm ảnh hưởng của gian lận tài chính.

- Hiện nay, thanh toán bằng thẻ tín dụng đã trở thành một phương thức tiêu dùng quan trọng ở cuộc sống hiện đại. Tuy nhiên, với sự phát triển nhanh chóng của ngành thẻ tín dụng, tình trạng gian lận giao dịch đã nổi lên như một vấn đề nghiêm trọng. Gian lận thẻ tín dụng không chỉ dẫn đến những tổn thất tài chính mà còn gây tổn hại đến uy tín của các tổ chức tài chính, khiến người dân mất niềm tin vào việc thanh toán bằng thẻ tín dụng. Vì vậy, việc phát hiện gian lận thẻ tín dụng đã trở thành một nhiệm vụ quan trọng của các tổ chức tài chính.

- Đặc điểm của dữ liệu gian lận tài chính:
  - Dữ liệu rất không cân bằng, với số lượng giao dịch gian lận chiếm tỷ lệ nhỏ, thông thường chỉ dưới 1% tổng số giao dịch.
  - Dữ liệu giao dịch tài chính thường có sự biến động lớn về các đặc trưng, bao gồm số tiền giao dịch, địa điểm, và thời gian.

1.2 Vấn đề mất cân bằng trong dữ liệu bảng

1.2.1 Định nghĩa và nhiệm vụ

- Dữ liệu không cân bằng xảy ra khi một lớp (giao dịch gian lận) có số lượng mẫu ít hơn rất nhiều so với lớp còn lại (giao dịch hợp lệ).

- **Nhiệm vụ**: Phát hiện chính xác các giao dịch gian lận mà không làm tăng đáng kể tỷ lệ báo động sai (false positives).

1.2.2 Các phương pháp tiếp cận xử lý mất cân bằng dữ liệu hiện có

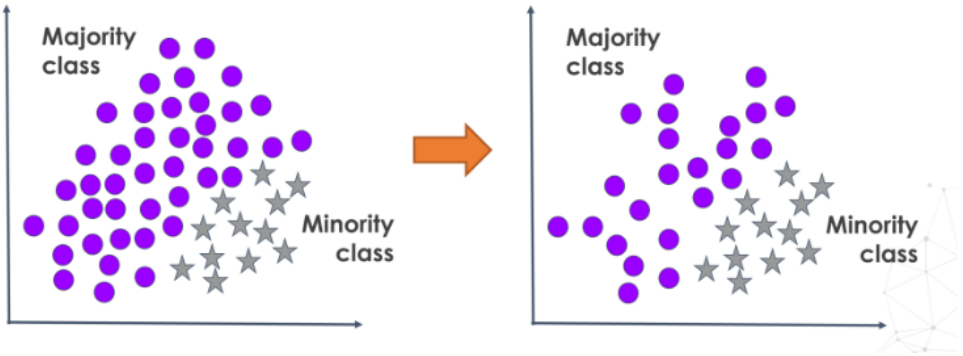
- **Dưới đây là phần trình bày chi tiết và bản chất về các phương pháp xử lý dữ liệu mất cân bằng:**

1. Nhóm phương pháp ở mức dữ liệu (Data-level Approaches)
Đây là nhóm *can thiệp trực tiếp vào dữ liệu* trước khi huấn luyện mô hình.

1.1. Undersampling
Undersampling là một kỹ thuật nhằm giảm số lượng các quan sát thuộc lớp đa số (majority class) để cân bằng với số lượng quan sát của lớp thiểu số (minority class). Kỹ thuật này thường được áp dụng trong các bài toán học máy mất cân bằng dữ liệu nghiêm trọng, nơi mà số lượng mẫu giữa các lớp có sự chênh lệch lớn. Ưu điểm chính của undersampling là làm cân bằng dữ liệu một cách nhanh chóng mà không cần giả lập thêm mẫu mới. Tuy nhiên, kỹ thuật này có thể làm mất thông tin quan trọng từ lớp đa số, dẫn đến rủi ro underfitting.

Một số phương pháp phổ biến bao gồm:

- **Random Undersampling**
  Bản chất: Phương pháp này chọn ngẫu nhiên một lượng mẫu từ lớp chiếm đa số sao cho kích thước của lớp này giảm xuống tương xứng với lớp thiểu số.
  Giả sử tập dữ liệu có N_{majority} mẫu thuộc lớp đa số và N_{minority} mẫu thuộc lớp thiểu số với $N_{\text{majority}} > N_{\text{minority}}$. Random Undersampling chọn ngẫu nhiên N_{minority} mẫu từ lớp đa số và giữ nguyên toàn bộ mẫu từ lớp thiểu số, tạo thành tập dữ liệu cân bằng.

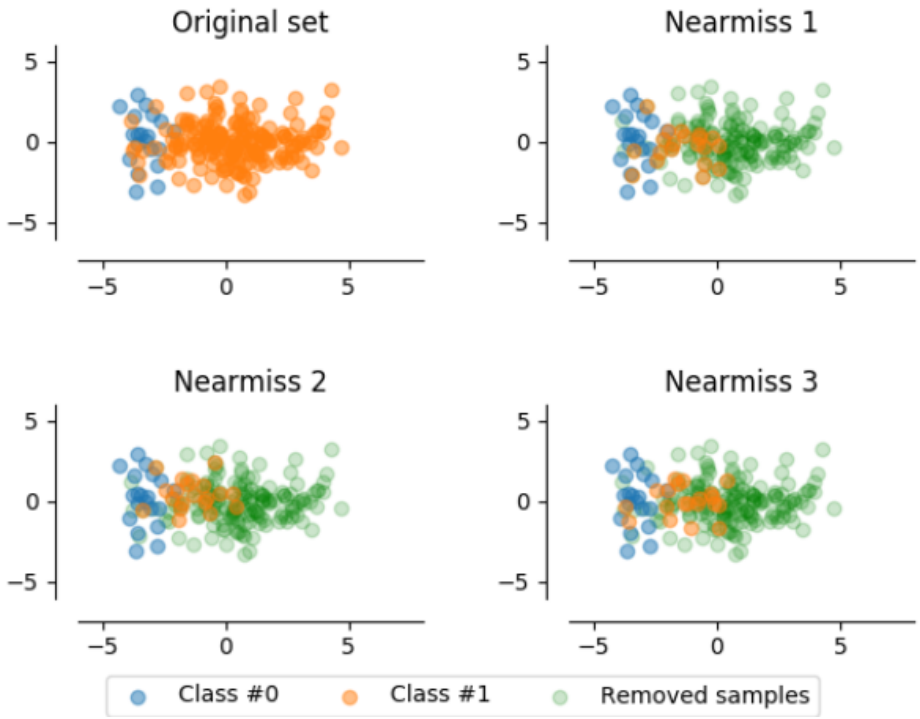


Hình 1: Ví dụ minh họa về Random Undersampling

Ưu điểm: Phương pháp đơn giản, nhanh chóng và dễ triển khai. Giảm nhanh độ chênh lệch giữa hai lớp.
Nhược điểm: Có nguy cơ loại bỏ các mẫu quan trọng từ lớp đa số, dẫn đến mất thông tin. Có thể gây underfitting (mô hình học kém do dữ liệu quá ít).

- **NearMiss**
  Bản chất: NearMiss chọn các mẫu thuộc lớp đa số dựa trên khoảng cách của chúng tới các điểm trong lớp thiểu số. Phương pháp này có nhiều biến thể, trong đó mỗi biến thể tập trung vào việc giảm độ mất cân bằng thông qua các tiêu chí khác nhau.
  - Giả sử khoảng cách giữa hai điểm dữ liệu x_i và x_j được tính theo công thức Euclidean:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2}$$



Hình 2: Ví dụ minh họa về Near Miss

Với d là số chiều dữ liệu. *Ưu điểm*: Tăng cường sự gần gũi giữa dữ liệu của hai lớp, giúp mô hình dễ học ranh giới quyết định hơn.
Nhược điểm: Tốn nhiều thời gian tính toán hơn so với Random Undersampling, do phải tính toán khoảng cách giữa nhiều cặp dữ liệu. Có thể vẫn bỏ qua các mẫu quan trọng của lớp đa số.

- Các biến thể:

  – **NearMiss-1**: Chọn các mẫu lớp đa số có khoảng cách gần nhất đến k mẫu gần nhất của lớp thiểu số.

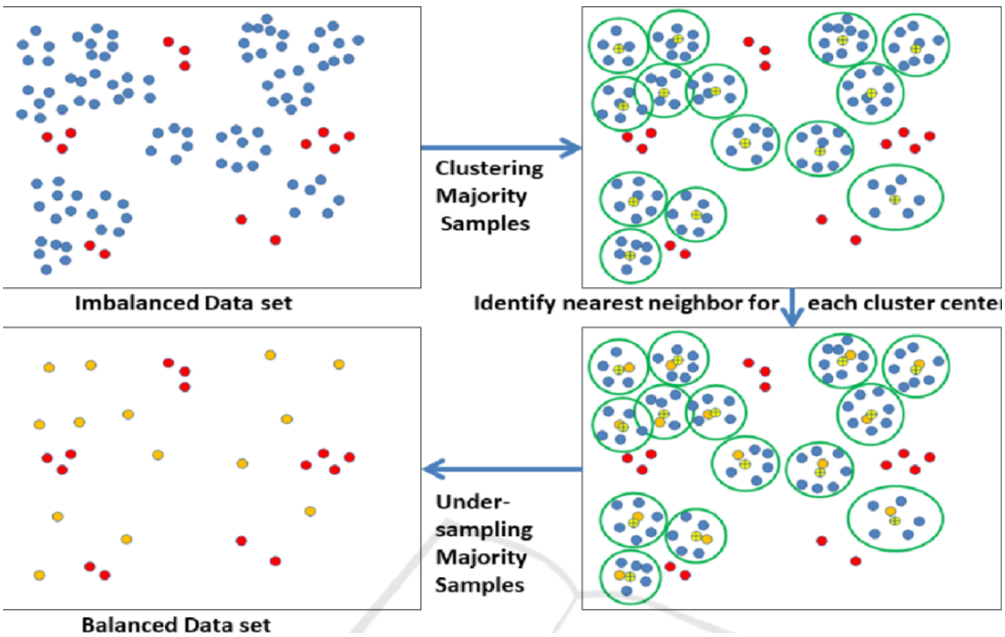
  – **NearMiss-2**: Chọn các mẫu lớp đa số có khoảng cách nhỏ nhất đến k mẫu xa nhất của lớp thiểu số.

  – **NearMiss-3**: Giữ lại các mẫu lớp đa số mà khoảng cách trung bình đến các điểm lớp thiểu số nhỏ hơn một ngưỡng nhất định.

- **Cluster-based Undersampling**
Bản chất: Gom cụm (clustering) các mẫu thuộc lớp đa số và chọn một hoặc một số đại diện từ mỗi cụm để giảm kích thước lớp này.
Giả sử lớp đa số N_{majority} được chia thành k cụm $C_1, C_2, \ldots, C_k$ thông qua thuật toán phân cụm như K-means. Từ mỗi cụm C_i, chọn một mẫu đại diện x_i^*:

$$x_i^* = \arg\min_{x \in C_i} d(x, \mu_i)$$

Trong đó, μ_i là tâm cụm C_i, và $d(x, \mu_i)$ là khoảng cách giữa x và tâm cụm. *Ưu điểm*: Phương pháp này giúp giữ được tính đa dạng của lớp đa số vì các mẫu đại diện được chọn từ nhiều cụm khác nhau.
Nhược điểm: Phụ thuộc vào chất lượng của thuật toán gom cụm. Nếu phân cụm không



Hình 3: Ví dụ minh họa về Cluster-based Undersampling

tốt, các mẫu đại diện có thể không phản ánh đúng bản chất dữ liệu. Tốn thời gian tính toán hơn các phương pháp khác.

1.2. Oversampling

Over sampling là các phương pháp giúp giải quyết hiện tượng mất cân bằng mẫu bằng cách gia tăng kích thước mẫu thuộc nhóm thiểu số bằng các kĩ thuật khác nhau. Có 2 phương pháp chính để thực hiện over sampling đó là:

- Lựa chọn mẫu có tái lập.
- Mô phỏng mẫu mới dựa trên tổng hợp của các

mẫu cũ.

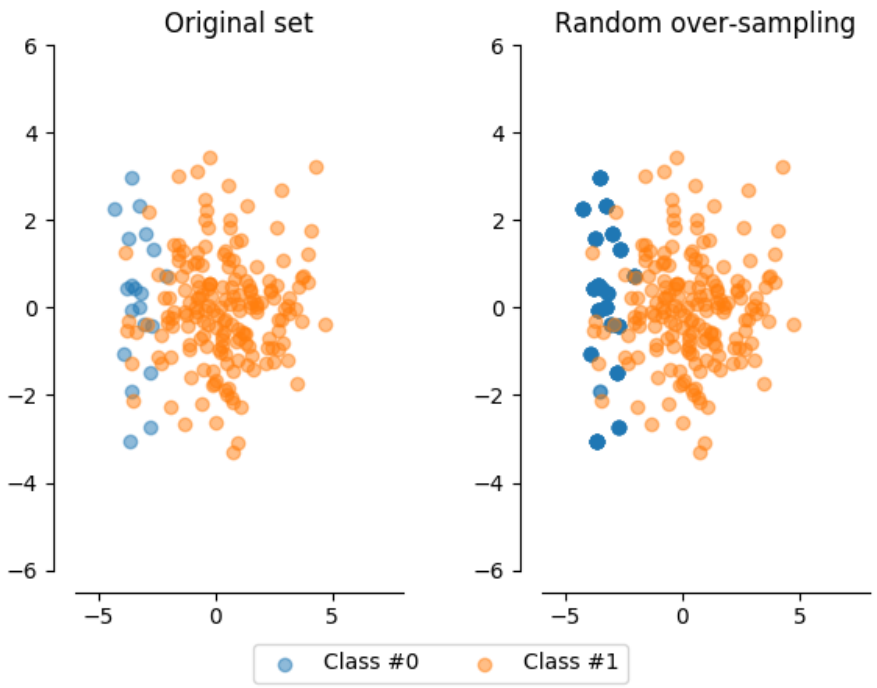
- **Random Oversampling**
  Bản chất: Phương pháp này thực hiện bằng cách sao chép (lặp lại) ngẫu nhiên các mẫu thuộc lớp thiểu số để tăng số lượng dữ liệu. Các mẫu được sao chép chính xác như mẫu ban đầu, không thay đổi đặc điểm hoặc giá trị gốc.

  Cách thực hiện:

  – Bước 1: Xác định số lượng mẫu cần bổ sung để cân bằng với lớp đa số.
  – Bước 2: Chọn ngẫu nhiên các mẫu từ lớp thiểu số.
  – Bước 3: Sao chép và thêm các mẫu được chọn vào tập dữ liệu huấn luyện.

  Ưu điểm:

  – Đơn giản, dễ thực hiện, không đòi hỏi tính toán phức tạp.
  – Giữ nguyên thông tin gốc của lớp thiểu số, đảm bảo phân phối ban đầu không bị thay đổi.



Hình 4: Ví dụ minh họa về Random Oversampling

Nhược điểm:

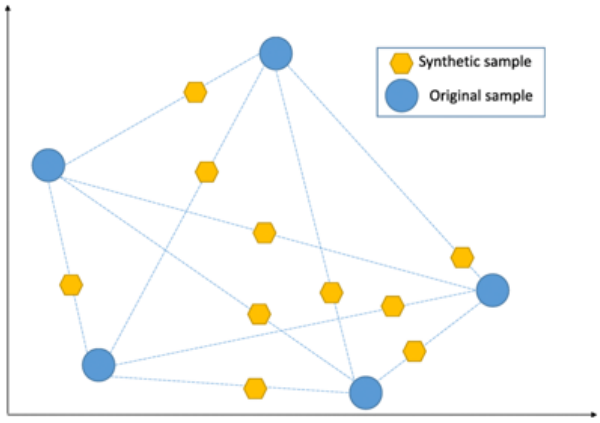
- Dễ gây hiện tượng **overfitting**, đặc biệt khi lớp thiểu số có rất ít mẫu và các mẫu được lặp lại nhiều lần.
- Không bổ sung thêm thông tin mới cho lớp thiểu số, khiến mô hình khó học được các đặc điểm mới.

- **SMOTE (Synthetic Minority Oversampling Technique)**
  Bản chất: SMOTE tạo các mẫu tổng hợp mới bằng cách nội suy tuyến tính giữa các điểm dữ liệu lớp thiểu số gần nhau trong không gian đặc trưng. Phương pháp này không sao chép lại mẫu gốc mà mở rộng không gian lớp thiểu số bằng cách tạo ra các mẫu mới nằm giữa các điểm dữ liệu hiện có.
  Cách thực hiện:
  - Bước 1: Với mỗi mẫu trong lớp thiểu số, xác định k-hàng xóm gần nhất (thường chọn $k = 5$).
  - Bước 2: Chọn ngẫu nhiên một hàng xóm từ k-hàng xóm gần nhất.
  - Bước 3: Tạo mẫu mới bằng công thức nội suy:

$$x_{\text{new}} = x_{\text{original}} + \lambda \cdot (x_{\text{neighbor}} - x_{\text{original}})$$

    với λ là giá trị ngẫu nhiên trong khoảng $[0, 1]$.

  Ưu điểm:
  - Tăng tính đa dạng của dữ liệu lớp thiểu số, giảm nguy cơ **overfitting** so với Random Oversampling.
  - Phân phối các mẫu mới rộng hơn trong không gian đặc trưng, giúp mô hình học tốt hơn tại ranh giới phân lớp.



Hình 5: Ví dụ minh họa về SMOTE

Nhược điểm:

- Có thể tạo ra các mẫu tổng hợp chồng lấn tại ranh giới giữa lớp thiểu số và lớp đa số, gây khó khăn trong phân loại.
- Giả định rằng các điểm gần nhau thuộc cùng một phân phối, điều này không phải lúc nào cũng chính xác trong dữ liệu thực tế.

- **ADASYN (Adaptive Synthetic Sampling)**
  Bản chất: ADASYN là một biến thể nâng cao của SMOTE, tập trung tạo nhiều mẫu tổng hợp hơn ở các vùng dữ liệu thiểu số khó phân loại, nơi các điểm dữ liệu thưa thớt hoặc gần ranh giới phân lớp.
  Cách thực hiện:
  - Bước 1: Tính mức độ khó d_i của từng mẫu thiểu số x_i dựa trên tỉ lệ hàng xóm k-lớp đa số gần mẫu x_i.
  - Bước 2: Gán trọng số w_i cho từng mẫu, với mẫu có d_i cao hơn được ưu tiên tạo thêm dữ liệu.
  - Bước 3: Tạo mẫu mới bằng cách nội suy tương tự SMOTE, nhưng ưu tiên các mẫu có trọng số cao.

  Ưu điểm:
  - Tập trung cải thiện dữ liệu ở vùng khó phân loại, giúp mô hình học tốt hơn tại các ranh giới phân lớp.
  - Thích nghi với phân bố dữ liệu, thay vì tạo mẫu đồng đều như SMOTE.

  Nhược điểm:
  - Có thể sinh ra các mẫu không hợp lý tại ranh giới giữa các lớp.
  - Phức tạp hơn SMOTE, đòi hỏi tính toán nhiều hơn và thời gian xử lý lâu hơn.

- **Kết hợp Oversampling và Undersampling (SMOTEENN, SMOTETomek)**
  Bản chất: Các phương pháp này kết hợp SMOTE để tạo thêm mẫu thiểu số và các kỹ thuật undersampling (ENN, Tomek Link) để loại bỏ nhiễu và tinh chỉnh ranh giới giữa các lớp.
  Cách thực hiện:

- Bước 1: Sử dụng SMOTE để tạo mẫu tổng hợp cho lớp thiểu số.
- Bước 2: Áp dụng ENN hoặc Tomek Link để loại bỏ các mẫu dữ liệu nhiễu hoặc gần ranh giới không rõ ràng.

Ưu điểm:

- Cải thiện độ rõ ràng tại ranh giới phân lớp, tăng độ chính xác của mô hình.
- Loại bỏ nhiễu, giúp dữ liệu trở nên sạch và phù hợp hơn cho quá trình huấn luyện.

Nhược điểm:

- Phức tạp hơn các phương pháp đơn lẻ, đòi hỏi nhiều bước xử lý và tính toán.
- Có nguy cơ loại bỏ nhầm các mẫu dữ liệu quan trọng nếu thiết lập không chính xác.

- **Sử dụng Generative Model**
  - Với sự phát triển không ngừng của lĩnh vực Deep Learning, các mô hình sinh dữ liệu (Generative Models) như GANs (Generative Adversarial Networks) và VAEs (Variational AutoEncoders) đã mở ra một kỷ nguyên mới trong việc giải quyết vấn đề dữ liệu mất cân bằng, một thách thức lớn trong các bài toán học máy hiện đại. Những phương pháp này cho phép tạo ra các mẫu tổng hợp có chất lượng cao, phản ánh chính xác các đặc tính phức tạp của dữ liệu gốc mà không làm mất đi thông tin quan trọng. Điều này không chỉ cải thiện hiệu suất của các mô hình học máy mà còn giúp chúng trở nên ổn định hơn khi làm việc với các tập dữ liệu bị mất cân bằng nghiêm trọng.

  - Không giống như các phương pháp truyền thống, thường dựa vào việc nhân bản hoặc nội suy tuyến tính để tăng số lượng mẫu lớp thiểu số (ví dụ như Random Oversampling hay SMOTE), các Generative Models hoạt động dựa trên nguyên lý mô phỏng phân phối xác suất của dữ liệu. Điều này cho phép chúng tái tạo lại dữ liệu với độ chính xác cao, đồng thời nắm bắt được các đặc điểm không tuyến tính và phức tạp của tập dữ liệu ban đầu. Đặc biệt, khả năng sinh ra các mẫu tổng hợp này không chỉ dừng lại ở việc tạo ra các điểm dữ liệu "giống như thực" mà còn giúp đa dạng hóa các mẫu, làm giàu thêm thông tin cho lớp thiểu số mà không gây ra hiện tượng overfitting.

  - Trong đó, GANs nổi bật nhờ cơ chế hoạt động dựa trên sự cạnh tranh giữa hai mạng nơ-ron: Generator và Discriminator. Generator cố gắng tạo ra các mẫu tổng hợp sao cho không thể phân biệt được với dữ liệu thật, trong khi Discriminator liên tục học để phân biệt giữa dữ liệu thật và dữ liệu giả. Quá trình này không chỉ giúp Generator cải thiện khả năng mô phỏng dữ liệu thực mà còn đảm bảo rằng các mẫu sinh ra không bị sai lệch so với phân phối ban đầu. Điều này rất quan trọng trong các tập dữ liệu có đặc điểm phức tạp, chẳng hạn như các quan hệ phi tuyến tính giữa các biến hoặc sự tồn tại của các nhóm dữ liệu ngầm (latent clusters).

  - Bên cạnh GANs, VAEs cũng là một phương pháp mạnh mẽ trong việc sinh dữ liệu. Dựa trên nguyên lý xác suất Bayes, VAEs không chỉ học được cấu trúc của dữ liệu mà còn cung cấp một cách tiếp cận có hệ thống để sinh ra các mẫu mới.

Tuy không cạnh tranh trực tiếp như GANs, VAEs lại có ưu thế trong việc kiểm soát và đảm bảo tính liên tục của không gian dữ liệu, giúp tạo ra các mẫu tổng hợp mượt mà và đa dạng hơn.

– Những ưu điểm vượt trội này của các Generative Models khiến chúng trở thành công cụ không thể thiếu trong các bài toán xử lý dữ liệu mất cân bằng, đặc biệt là trong các lĩnh vực yêu cầu tính chính xác cao như y tế, tài chính, và an ninh mạng. Chẳng hạn, trong các bài toán phát hiện gian lận (fraud detection), nơi mà dữ liệu gian lận thường rất ít, GANs và VAEs có thể sinh ra các mẫu dữ liệu gian lận giả lập có chất lượng cao, giúp cải thiện đáng kể hiệu suất của các hệ thống nhận diện.

– Trong phần tiếp theo, chúng ta sẽ đi sâu hơn vào vai trò và cơ chế hoạt động cụ thể của GANs, một trong những mô hình sinh mạnh mẽ nhất hiện nay. Đồng thời, các ứng dụng thực tiễn của GANs trong việc xử lý dữ liệu mất cân bằng và mô phỏng các đặc điểm phức tạp của dữ liệu sẽ được phân tích chi tiết, minh họa rõ ràng thông qua các phương pháp cụ thể.

2. Nhóm phương pháp ở mức thuật toán (Algorithm-level Approaches)
Thay vì thay đổi dữ liệu, ta *giữ nguyên dataset* nhưng điều chỉnh mô hình.

- **Class Weighting**
  Bản chất: Gán trọng số cao hơn cho lớp thiểu số trong hàm mất mát (loss function).
  Ưu điểm: Dễ áp dụng (nhiều thư viện hỗ trợ), không thay đổi dữ liệu.
  Nhược điểm: Cần chọn trọng số hợp lý, dễ phải thử nghiệm nhiều.

- **Cost-Sensitive Learning**
  Bản chất: Định nghĩa chi phí sai sót (cost) riêng cho mỗi lớp, nhất là lớp thiểu số.
  Ưu điểm: Phản ánh rủi ro, tập trung vào bài toán thực tế.
  Nhược điểm: Khó xác định chính xác chi phí cho từng loại lỗi, một số mô hình không hỗ trợ sẵn.

- **Ensemble Methods (EasyEnsemble, BalanceCascade, SMOTEBoost...)**
  Bản chất: Kết hợp ý tưởng sampling (undersampling/SMOTE) với sức mạnh của multiple learners (boosting/bagging).
  Ưu điểm: Tận dụng ưu thế tổng hợp nhiều mô hình, nâng cao hiệu suất.
  Nhược điểm: Thời gian huấn luyện lâu, phức tạp hơn việc dùng mô hình đơn.

3. Nhóm phương pháp ở mức đánh giá (Evaluation-level Approaches)
Các *chỉ số đánh giá* rất quan trọng khi dữ liệu bị mất cân bằng.

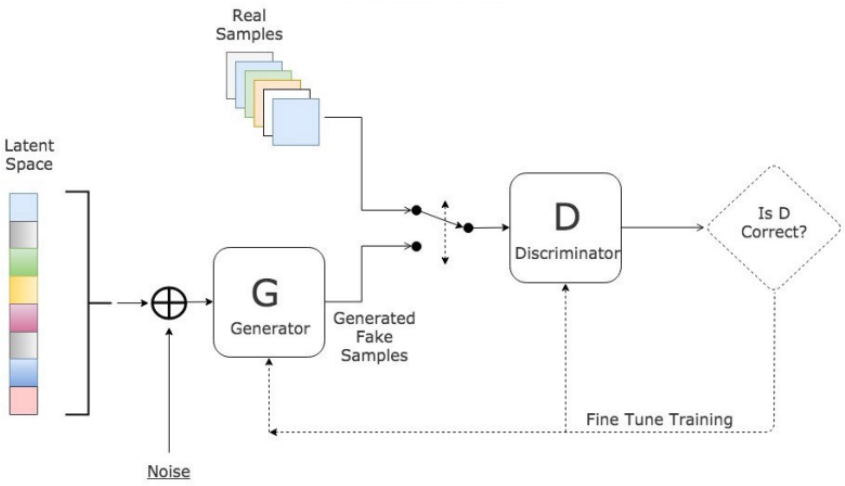
- **Chọn chỉ số phù hợp**
  (Precision, Recall, F1-score, AUC-PR, Balanced Accuracy...)
  Lý do: Accuracy không phù hợp khi dữ liệu mất cân bằng nặng, vì mô hình đoán tất cả là lớp đa số vẫn có độ chính xác rất cao.

- **Tập kiểm thử phản ánh thực tế**
  Giữ nguyên phân phối thật trong test set, để đánh giá đúng hiệu năng trên môi trường thực tế.

2 Vai trò của GANs trong tổng hợp dữ liệu mất cân bằng

2.1 Mô hình GAN truyền thống

- **Định nghĩa**: Generative Adversarial Networks (GAN) là một loại mô hình học sâu được đề xuất bởi Ian Goodfellow và các cộng sự vào năm 2014. GAN bao gồm hai mạng neural network: mạng sinh (Generator) và mạng phân biệt (Discriminator). Mạng sinh cố gắng sinh ra dữ liệu giả mà mạng phân biệt không thể phân biệt được với dữ liệu thật. Mạng phân biệt cố gắng phân biệt giữa dữ liệu thật và dữ liệu giả sinh bởi mạng sinh.
  - **Generator**: Sinh dữ liệu giả dựa trên nhiễu ngẫu nhiên.
  - **Discriminator**: Phân biệt giữa dữ liệu thật và dữ liệu giả.



Hình 6: Cấu trúc của mô hình GAN

- **Khó khăn**: GAN truyền thống không phù hợp để sinh dữ liệu dạng bảng do:
  - Tính phức tạp của dữ liệu bảng (bao gồm cả dữ liệu rời rạc và liên tục).
  - Vấn đề *mode collapse* (sập mode), khiến mô hình chỉ sinh ra một số mẫu dữ liệu hạn chế.

2.2 Những nghiên cứu ứng dụng GAN trong tổng hợp dữ liệu mất cân bằng

- GAN đã được sử dụng rộng rãi trong các bài toán như phát hiện gian lận, nhận diện bất thường, và chẩn đoán bệnh hiếm gặp.
- Các nghiên cứu cho thấy GAN có khả năng cải thiện đáng kể hiệu suất của mô hình học máy bằng cách sinh dữ liệu tổng hợp chất lượng cao.
- Chẳng hạn, một số hướng tiếp cận nổi bật như:

- **medGAN** sử dụng kết hợp auto-encoder và GAN để sinh dữ liệu y tế không có tính chuỗi thời gian (non-time-series), hỗ trợ cả dữ liệu nhị phân và dữ liệu liên tục.
- **ehrGAN** được thiết kế chuyên biệt để sinh bản ghi y tế (electronic health records - EHR) theo hướng mở rộng (augmented EHR), giúp tăng cường tính đa dạng của dữ liệu y tế.
- **tableGAN** sử dụng mô hình mạng nơ-ron tích chập (CNN) để sinh dữ liệu dạng bảng, tập trung tối ưu hóa chất lượng cột nhãn, từ đó có thể dùng dữ liệu sinh để huấn luyện các mô hình phân loại.
- **PATE-GAN** là một mô hình sinh dữ liệu đảm bảo tính riêng tư khác biệt (differential privacy), đáp ứng yêu cầu bảo mật và riêng tư trong nhiều ứng dụng nhạy cảm.

2.3 Những khó khăn và vấn sinh dữ liệu bảng

- Nhiệm vụ sinh dữ liệu tổng hợp yêu cầu huấn luyện một bộ tổng hợp dữ liệu G học từ một bảng $\mathbf{T}$ và sau đó sử dụng G để sinh một bảng tổng hợp $\mathbf{T}_{\text{syn}}$. Một bảng $\mathbf{T}$ chứa:
  - N_c cột liên tục $\{C_1, \ldots, C_{N_c}\}$,
  - N_d cột rời rạc $\{D_1, \ldots, D_{N_d}\}$,

  Trong đó mỗi cột được coi là một biến ngẫu nhiên. Các biến ngẫu nhiên này tuân theo một phân phối chung chưa biết $P\left(C_{1:N_c}, D_{1:N_d}\right)$.

  Một hàng $\mathbf{r}_j = \{c_{1,j}, \ldots, c_{N_c,j}, d_{1,j}, \ldots, d_{N_d,j}\}, j \in \{1, \ldots, n\}$, là một quan sát từ phân phối chung.
- $\mathbf{T}$ được chia thành hai tập:
  - Tập huấn luyện: $\mathbf{T}_{\text{train}}$,
  - Tập kiểm tra: $\mathbf{T}_{\text{test}}$.

  Sau khi huấn luyện G trên $\mathbf{T}_{\text{train}}$, $\mathbf{T}_{\text{syn}}$ được xây dựng bằng cách lấy mẫu độc lập các hàng sử dụng G.

Đánh giá hiệu quả của bộ sinh theo 2 yếu tố:

1. **Likelihood fitness:** Các cột trong $\mathbf{T}_{\text{syn}}$ có tuân theo cùng phân phối chung như $\mathbf{T}_{\text{train}}$ không?
2. **Machine learning efficacy:** Khi huấn luyện một bộ phân loại hoặc bộ hồi quy để dự đoán một cột sử dụng các cột khác làm đặc trưng, liệu bộ phân loại hoặc bộ hồi quy học được từ $\mathbf{T}_{\text{syn}}$ có đạt được hiệu suất tương tự trên $\mathbf{T}_{\text{test}}$ như một mô hình học được trên $\mathbf{T}_{\text{train}}$ không?

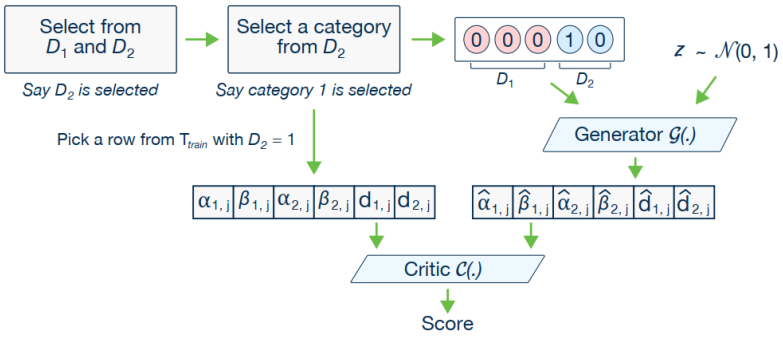
Một số tính chất của dữ liệu dạng bảng gây khó khăn khi thiết kế mô hình GAN.

- **Mixed data types:** Dữ liệu dạng bảng trong thực tế thường bao gồm cả cột rời rạc và cột liên tục. Việc tạo đồng thời cả hai loại cột này đòi hỏi GAN phải áp dụng cả hàm softmax và tanh trên đầu ra, gây khó khăn cho quá trình huấn luyện.
- **Non-Gaussian distributions:** Các giá trị liên tục trong dữ liệu dạng bảng thường không tuân theo phân phối Gaussian. Việc sử dụng chuẩn hóa min-max, thường được dùng trong xử lý ảnh, có thể dẫn đến vấn đề gradient biến mất.
- **Multimodal distributions:** Nhiều cột liên tục trong dữ liệu dạng bảng có nhiều mode (đỉnh). GAN truyền thống gặp khó khăn trong việc mô hình hóa tất cả các mode, dẫn đến việc tạo ra dữ liệu không phản ánh đầy đủ phân phối thực tế.
- **Learning from sparse one-hot-encoded vectors:** Khi tạo dữ liệu, GAN được huấn luyện để tạo phân phối xác suất trên tất cả các hạng mục bằng cách sử dụng softmax. Tuy nhiên, dữ liệu thực tế thường được biểu diễn bằng vector one-hot thưa thớt. Bộ phân biệt (discriminator) có thể dễ dàng phân biệt dữ liệu thật và giả chỉ bằng cách kiểm tra độ thưa thớt của phân phối thay vì xem xét tính thực tế tổng thể của một hàng.
- **Highly imbalanced categorical columns:** Nhiều cột rời rạc trong dữ liệu dạng bảng có sự mất cân bằng cao, trong đó một hạng mục chính xuất hiện trong phần lớn các hàng. Điều này tạo ra hiện tượng sập mode (mode collapse), khiến GAN bỏ qua các hạng mục nhỏ hơn và tạo ra dữ liệu kém đa dạng. Việc thiếu một hạng mục nhỏ chỉ gây ra những thay đổi nhỏ đối với phân phối dữ liệu, khiến bộ phân biệt khó phát hiện. Dữ liệu mất cân bằng cũng dẫn đến việc thiếu cơ hội huấn luyện cho các lớp thiểu số.

2.4 Đề xuất CTGAN

2.4.1 Mô hình CTGAN

- **Tổng quan**: CTGAN (Conditional Table GAN) là một biến thể của GAN được thiết kế đặc biệt để sinh dữ liệu bảng. CTGAN sử dụng mạng sinh(Generator G) và mạng phân biệt (Critic C) để sinh dữ liệu bảng mới dựa trên dữ liệu đầu vào.



Hình 7: Cấu trúc của mô hình CTGAN

2.4.2 Những ưu điểm của CTGAN

- Khả năng xử lý tốt dữ liệu bảng với các đặc trưng không cân bằng.
- Sinh dữ liệu tổng hợp chính xác, giúp cải thiện hiệu suất mô hình phân loại.

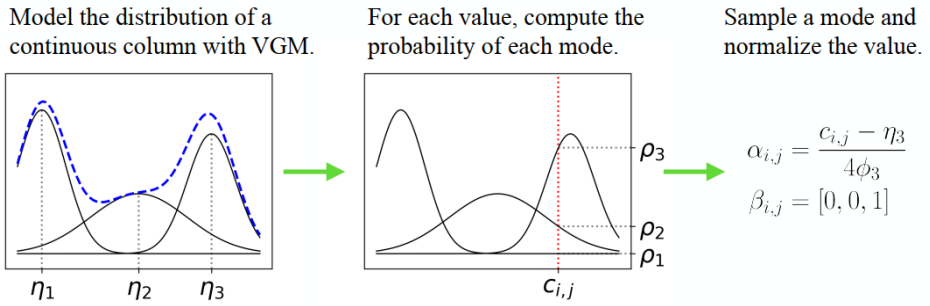
3 Cơ chế nổi bật của CTGAN trong sinh dữ liệu dạng bảng

CTGAN (Conditional Tabular GAN) được thiết kế để giải quyết hiệu quả các thách thức trong việc sinh dữ liệu bảng, đặc biệt là xử lý dữ liệu liên tục phức tạp và dữ liệu rời rạc mất cân bằng. Hai cơ chế nổi bật của CTGAN là **Mode-Specific Normalization** và **Conditional Generator with Training-by-Sampling**. Dưới đây là mô tả chi tiết từng bước thực hiện của hai cơ chế này.

3.1 Mode-Specific Normalization (Chuẩn hóa theo đỉnh)

Thách thức:

Dữ liệu liên tục như số tiền giao dịch (*Transaction Amount*) trong các tập dữ liệu phát hiện gian lận thường có nhiều đỉnh (mode) (multimodal distributions) và không tuân theo phân phối Gaussian. Ví dụ, số tiền giao dịch có thể tập trung ở các cụm giá trị như dưới \$10, \$100–\$500, và trên \$1000. Các phương pháp chuẩn hóa truyền thống như Min-Max Scaling không thể biểu diễn tốt sự phức tạp này.



Hình 8: Ví vụ về Mode-Specific Normalization

Các bước thực hiện:

Mode-Specific Normalization sử dụng *Variational Gaussian Mixture Model (VGM)* để phân tích và chuẩn hóa dữ liệu liên tục thành dạng phù hợp hơn cho các mạng thần kinh. Các bước thực hiện chi tiết như sau:

1. **Xác định số chế độ:**
   - Sử dụng VGM để ước lượng số lượng chế độ (m) trong cột liên tục.

- VGM học phân phối Gaussian hỗn hợp, trong đó mỗi chế độ k được mô tả bằng trung bình (η_k), độ lệch chuẩn (ϕ_k), và trọng số (μ_k).

2. **Tính xác suất chế độ:**

   - Với mỗi giá trị c_i trong cột, tính xác suất thuộc về từng mode k:

   $$\rho_k = \mu_k \cdot N(c_i|\eta_k, \phi_k),$$

   trong đó $N(c_i|\eta_k, \phi_k)$ là mật độ xác suất của phân phối Gaussian.

3. **Xác định chế độ phù hợp:**

   - Chọn chế độ k^* với xác suất lớn nhất: $k^* = \arg\max_k \rho_k$.

4. **Chuẩn hóa giá trị:**

   - Biểu diễn giá trị c_i bằng:
     - Một *vector one-hot* chỉ định mode: $[0, 1, 0]$ nếu thuộc chế độ k^*.
     - Một giá trị chuẩn hóa trong mode k^*:

     $$\alpha_i = \frac{c_i - \eta_{k^*}}{\phi_{k^*}},$$

     trong đó η_{k^*} và ϕ_{k^*} lần lượt là trung bình và độ lệch chuẩn của modek^*.

Ví dụ minh họa:

Giả sử cột "số tiền giao dịch" có giá trị \$200:

- VGM phát hiện ba mode: [\$0–\$10], [\$100–\$500], và [\$1000+].
- Giá trị \$200 thuộc chế độ thứ hai ([\$100–\$500]) với xác suất cao nhất.
- Giá trị này được biểu diễn bằng:
  - Vector mode: $[0, 1, 0]$.
  - Giá trị chuẩn hóa: $(200 - 300)/100 = -1$ (giả sử trung bình chế độ là \$300, độ lệch chuẩn là \$100).

3.2 Conditional Generator và Training-by-Sampling

Thách thức:

Dữ liệu rời rạc như "loại giao dịch" (*Transaction Type*) trong phát hiện gian lận thường rất mất cân bằng. Ví dụ, các giao dịch hợp pháp chiếm hơn 99%, trong khi giao dịch gian lận chỉ chiếm dưới 1%. Nếu không được xử lý, các mô hình sẽ bỏ qua các giao dịch gian lận quan trọng.

Các bước thực hiện:

CTGAN sử dụng Conditional Generator kết hợp với Training-by-Sampling để giải quyết vấn đề mất cân bằng dữ liệu. Chi tiết các bước như sau:

1. **Tạo vector điều kiện:**
   - Một vector điều kiện (*condition vector*) được tạo để biểu diễn giá trị cần sinh trong cột rời rạc.
   - Ví dụ: Với cột "loại giao dịch", giá trị "gian lận" được biểu diễn bằng vector $[0, 1]$, trong khi "hợp pháp" là $[1, 0]$.
2. **Chọn điều kiện trong huấn luyện:**
   - Trong mỗi bước huấn luyện, chọn một giá trị điều kiện dựa trên **logarit tần suất** của giá trị đó.
   - Điều này giúp tăng xác suất chọn các giá trị thiểu số, ví dụ "gian lận", trong quá trình huấn luyện.
3. **Sinh dữ liệu điều kiện:**
   - Generator nhận input gồm một vector ngẫu nhiên z và vector điều kiện.
   - Generator học cách sinh dữ liệu phù hợp với vector điều kiện đã chọn.
4. **Đánh giá bởi Critic:**
   - Critic kiểm tra tính xác thực của dữ liệu sinh, đảm bảo rằng:
     - Dữ liệu tuân theo phân phối thực.
     - Vector điều kiện được duy trì chính xác (ví dụ: giao dịch sinh ra thật sự là "gian lận").

Ví dụ minh họa:

Giả sử trong tập dữ liệu, giao dịch "gian lận" chiếm 1%:

- Giá trị "gian lận" được chọn làm điều kiện trong một bước huấn luyện, với xác suất cao hơn logarit tần suất.
- Generator được cung cấp vector điều kiện $[0, 1]$ (gian lận) và sinh dữ liệu giao dịch.
- Critic đánh giá dữ liệu sinh và phản hồi để đảm bảo rằng mẫu sinh ra phản ánh đúng giao dịch gian lận.

3.3 Kết luận

Hai cơ chế **Mode-Specific Normalization** và **Conditional Generator cùng Training-by-Sampling** là các thành phần cốt lõi giúp CTGAN giải quyết hiệu quả các vấn đề phức tạp trong phát hiện gian lận thẻ tín dụng. Chúng không chỉ giúp xử lý dữ liệu liên tục và rời rạc, mà còn đảm bảo rằng dữ liệu tổng hợp sinh ra có chất lượng cao, phù hợp với phân phối thực tế, và đại diện đầy đủ cho các giao dịch gian lận – yếu tố quan trọng trong bài toán mất cân bằng dữ liệu.

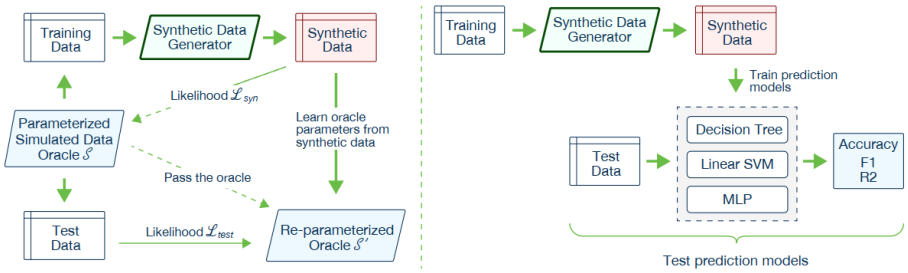
3.4 Metric đánh giá dữ liệu sau khi sinh

Likelihood Fitness: Đánh giá liệu phân phối của dữ liệu tổng hợp có tương đồng với phân phối dữ liệu gốc không, thông qua:

- L_{syn}: Tính xác suất của dữ liệu tổng hợp trên mô hình oracle ban đầu.
- L_{test}: Đánh giá dữ liệu kiểm tra trên một mô hình oracle mới được huấn luyện từ dữ liệu tổng hợp.

Machine Learning Efficacy: Đo lường hiệu quả dữ liệu tổng hợp khi sử dụng làm dữ liệu huấn luyện cho các mô hình học máy. Chỉ số đánh giá bao gồm:

- Precision, Recall, F1-Score, GMean cho mô hình **phân loại**.



Hình 9: Đánh giá dữ liệu mô phỏng(phải) dữ liệu thật (trái)

- **Precision**

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}} \tag{1}$$

Precision đo lường tỷ lệ các giao dịch được dự đoán là gian lận thực sự là gian lận. Nó trả lời câu hỏi: "Trong số các giao dịch được dự đoán là gian lận, bao nhiêu là đúng?" Precision cao cho thấy mô hình ít dự đoán nhầm giao dịch hợp lệ thành gian lận.

- **Recall**

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{2}$$

Recall đo lường khả năng của mô hình trong việc phát hiện các giao dịch gian lận. Nó trả lời câu hỏi: "Trong số các giao dịch gian lận thực sự, bao nhiêu được phát hiện?" Recall cao cho thấy mô hình ít bỏ sót giao dịch gian lận.

- **F1-Score**

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

F1-Score là trung bình điều hòa của Precision và Recall, giúp cân bằng giữa hai chỉ số này. F1-Score rất hữu ích khi dữ liệu bị mất cân bằng và cần một chỉ số tổng quát để đánh giá hiệu suất mô hình.

- **Geometric Mean (GMean)**

$$\text{GMean} = \sqrt{\text{True Positive Rate (TPR)} \times \text{True Negative Rate (TNR)}} \tag{4}$$

Trong đó:

$$\text{True Positive Rate (TPR)} = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

$$\text{True Negative Rate (TNR)} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}} \tag{6}$$

GMean đo lường khả năng duy trì cân bằng giữa khả năng phát hiện giao dịch gian lận (TPR) và việc phân loại đúng các giao dịch hợp lệ (TNR). Đây là một chỉ số lý tưởng khi dữ liệu bị mất cân bằng nghiêm trọng.

- **R-squared (R^2)** cho bài toán hồi quy.

- **Total Variation Distance (TVD)**: Đánh giá sự khác biệt giữa phân phối dữ liệu gốc và dữ liệu tổng hợp.
- **Jensen-Shannon Divergence (JSD)**: Đo mức độ tương đồng giữa hai phân phối.
- Sử dụng các framework như *SDMetrics* để đánh giá hiệu quả dữ liệu sinh ra.

4 Áp dụng

4.1 Dữ liệu được áp dụng

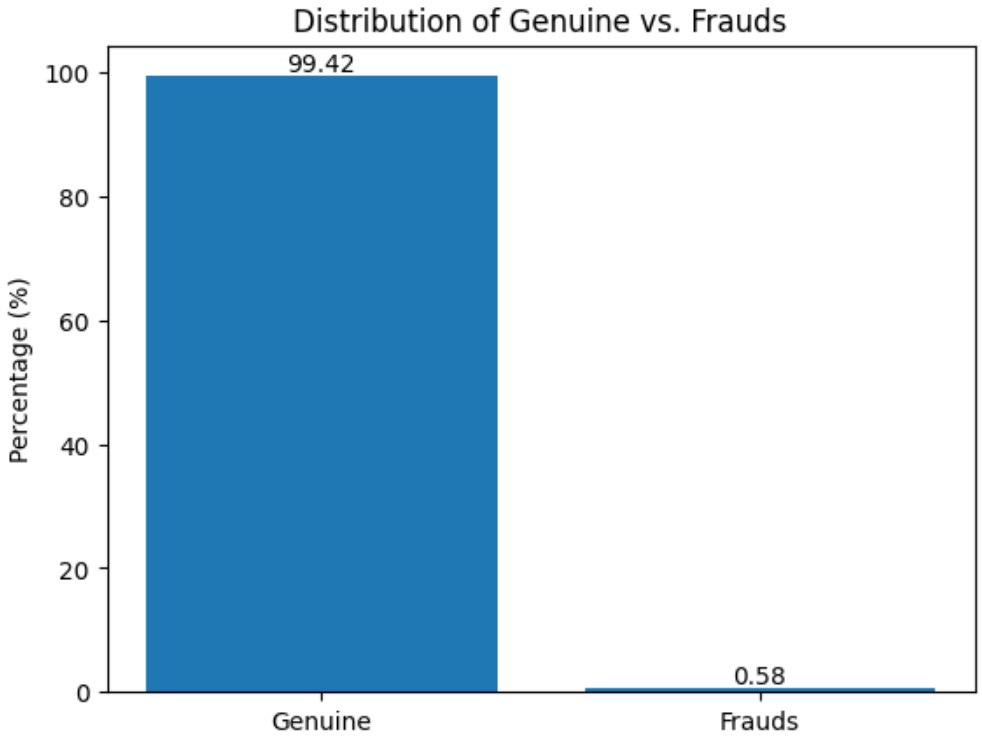
Trong bài báo cáo này chúng em sử dụng 2 bộ dữ liệu để đánh giá kết của của CT GAN trong vao trò sinh thêm các lớp thiểu số để cải thiện hiệu suất mô hình phân loại:

1. Fraud Detection Dataset1
2. Credit Card Fraud Dataset2

4.2 Bộ dữ liệu 1

4.2.1 Tổng quan về dữ liệu

- Mô tả:Đây là một bộ dữ liệu giao dịch thẻ tín dụng mô phỏng, bao gồm các giao dịch hợp pháp và gian lận trong khoảng thời gian từ ngày 1 tháng 1 năm 2019 đến ngày 31 tháng 12 năm 2020. Nó bao gồm thẻ tín dụng của 1.000 khách hàng thực hiện giao dịch với một nhóm gồm 800 khách hàng.
- Dữ liệu giao dịch thẻ tín dụng với các đặc trưng như số tiền giao dịch, loại giao dịch, thời gian, địa điểm.
- Tỷ lệ giao dịch gian lận chiếm khoảng 0.58% trong toàn bộ dữ liệu.



Hình 10: Tỉ lệ giao dịch lợp lệ và gian lận ở bộ dữ liệu 1

4.2.2 Case base: Áp dụng Dữ liệu mất cần bằng cùng mô hình phân loại

Sử dụng trực tiếp dữ liệu mất cân bằng nghiêm trọng cho cho các mô hình phân loại, hiệu suất của mô hình và tính sai lệch sẽ được đề cập ở phần tiếp theo.

4.2.3 Case 1: Áp dụng Nhóm oversampling truyền thống để cân bằng dữ liệu cùng mô hình phân loại

Bao gồm hai mô hình Smote, Random OverSampling

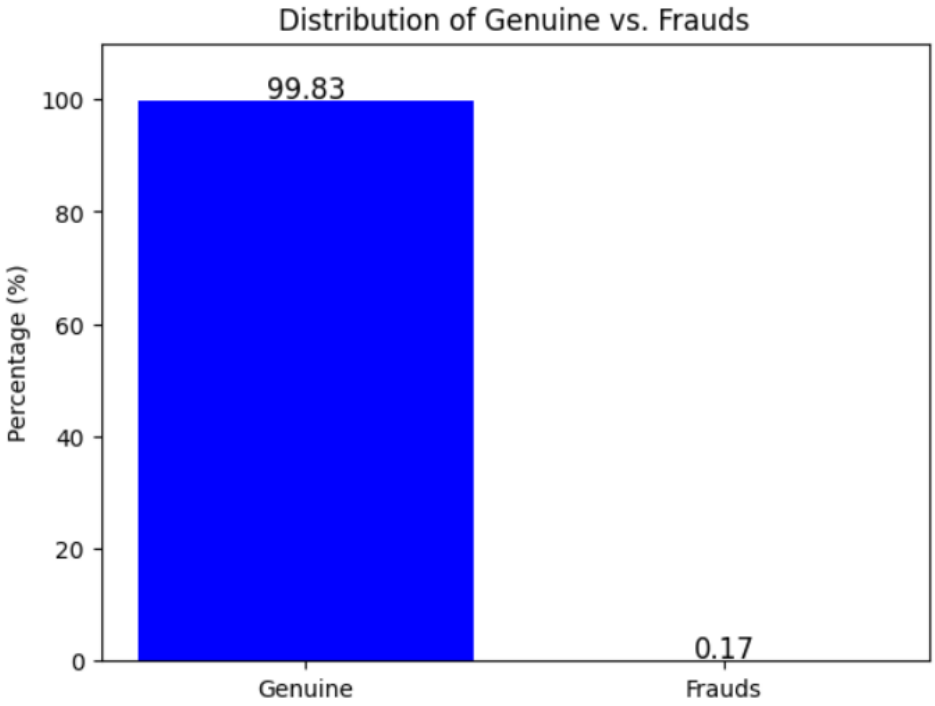
4.2.4 Case 2: Áp dụng CTGAN dể cân bằng dữ liệu cùng mô hình phân loại

- **Mục tiêu:** Sinh thêm dữ liệu cho lớp gian lận (fraud) để cân bằng tỷ lệ giữa lớp fraud và lớp non-fraud.
- **Quy trình:**
  1. Sử dụng tập train để huấn luyện CTGAN, học phân phối của dữ liệu thực.
  2. Sinh thêm dữ liệu fraud tổng hợp bằng CTGAN.
  3. Kết hợp dữ liệu tổng hợp với dữ liệu gốc để tạo ra tập huấn luyện cân bằng theo theo các tỉ lệ 50:50, 40:60,
  4. Huấn luyện các mô hình phân loại như Random Forest, Logistic Regression, và XGBoost trên tập dữ liệu cân bằng này.

4.3 Bộ dữ liệu 2

4.3.1 Tổng quan về dữ liệu

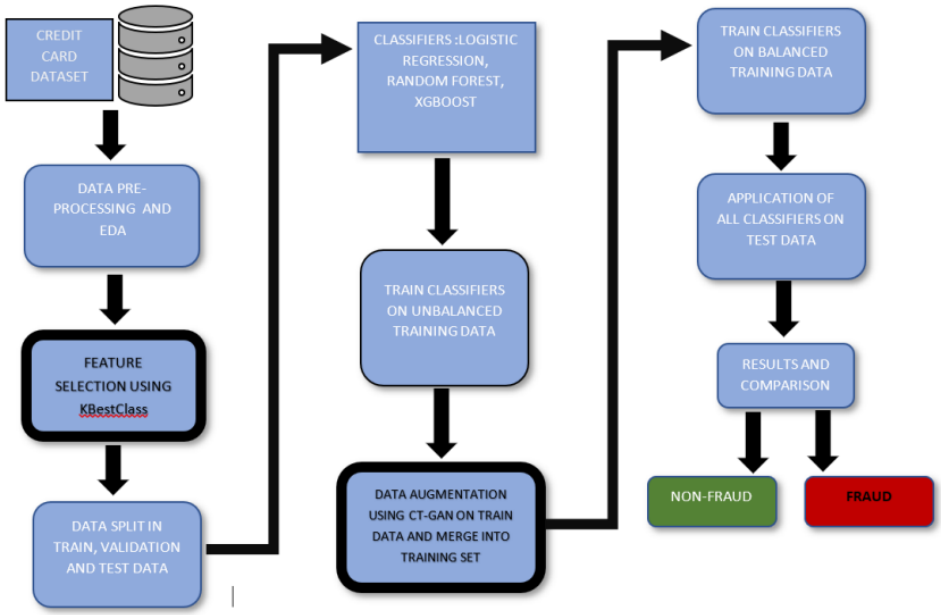
- Dữ liệu này bao gồm các giao dịch được thực hiện bằng thẻ tín dụng vào tháng 9 năm 2013 bởi những chủ thẻ châu Âu.Bộ dữ liệu này trình bày các giao dịch đã xảy ra trong hai ngày, trong đó có 492 giao dịch gian lận trong tổng số 284.807 giao dịch. Bộ dữ liệu này có sự mất cân bằng cao, lớp dương tính (gian lận) chiếm 0,172%. Bộ dữ liệu này được PCA để bảo mật thông tin.



Hình 11: Tỉ lệ giao dịch lợp lệ và gian lận ở bộ dữ liệu 1

4.3.2 Quy trình áp dụng

Lọc các đặc trưng bằng phương pháp chọn đặc trưng KBestClass để chỉ giữ lại các cột liên quan cho phân tích. Sau đó, chia dữ liệu thành tập huấn luyện, tập xác thực và tập kiểm tra, tiếp theo là huấn luyện trên tập dữ liệu huấn luyện không cân bằng. Ở một giai đoạn sau, chúng tôi sẽ thực hiện tăng cường dữ liệu bằng CT-GAN để tạo ra các mẫu cho lớp thiểu số. Dữ liệu tăng cường này với dữ liệu huấn luyện để cân bằng nó. Sau đó, chúng tôi tiếp tục huấn luyện ba mô hình phân loại trên các tập dữ liệu huấn luyện đã được cân bằng này. Hiệu suất của các mô hình phân loại được huấn luyện trên cả tập dữ liệu huấn luyện không cân bằng và cân bằng sẽ được kiểm tra và đánh giá ở giai đoạn cuối cùng bằng cách sử dụng tập dữ liệu kiểm tra. Ở giai đoạn cuối cùng, chúng em đánh giá và so sánh kết quả cho cả hai kịch bản.



Hình 12: Sơ đồ quy trình cho bộ dữ liệu 2

5 Kết quả thu được

5.1 Dữ liệu 1

Dữ liệu không cân bằng gây ra những thách thức lớn cho các mô hình, đặc biệt Logistic Regression khi hiệu suất trên tập kiểm tra rất thấp. Random Forest và XGBoost hoạt động tốt hơn trên dữ liệu không cân bằng, nhưng vẫn gặp hạn chế trong việc xử lý các giao dịch gian lận, với Recall và F1-Score không đạt mức tối ưu. Điều này cho thấy cần thiết phải sử dụng các phương pháp cân bằng dữ liệu để cải thiện hiệu suất.

Trong các phương pháp cân bằng, CTGAN cho thấy hiệu suất vượt trội nhất khi cải thiện đáng kể Recall và F1-Score, đặc biệt khi áp dụng với Random Forest và XGBoost. Tuy nhiên, hiện tượng overfitting rõ rệt xuất hiện ở CTGAN, làm giảm tính tổng quát hóa của mô hình, điều này đòi hỏi cần có biện pháp kiểm soát để đảm bảo hiệu quả thực tế. SMOTE ở trường hợp này là chọn khả thi với hiệu suất ổn định và ít nguy cơ overfitting hơn CTGAN, nhưng cũng không mang lại hiệu suất cao.

Với Random Forest và XGBoost là hai mô hình nổi bật nhất trong việc xử lý dữ liệu cân bằng. Điều này nhấn mạnh tầm quan trọng của việc lựa chọn phương pháp cân bằng phù hợp để tối ưu hóa hiệu quả trong các bài toán dữ liệu mất cân bằng.

Bảng 1: Hiệu suất với dữ liệu mất cân bằng

Model	F1-Score (Test)	Recall (Test)	AUC	Geometric Mean (Test)
Logistic Regression - Imbalanced	0.00%	0.00%	99.94%	0.00%
Random Forest - Imbalanced	84.97%	77.25%	99.99%	87.88 %
XGBoost - Imbalanced	83.63 %	76.60%	99.99%	87.51 %

Bảng 2: Hiệu suất với dữ liệu cân bằng bằng SMOTE

Model	F1-Score (Test)	Recall (Test)	AUC	Geometric Mean (Test)
Logistic Regression - SMOTE	6.00%	77.06%	92.86%	83.62%
Random Forest - SMOTE	55.57%	86.99%	100.00%	93.04%
XGBoost - SMOTE	6.99%	96.36%	99.03%	93.17 %

Bảng 3: Hiệu suất với dữ liệu cân bằng bằng CTGAN

Model	F1-Score (Test)	Recall (Test)	AUC	Geometric Mean (Test)
Logistic Regression - CTGAN	6.17%	74.69%	95.03%	82.58 %
Random Forest - CTGAN	100.00%	90.3%5	96.97%	93.60%
XGBoost - CTGAN	98.72%	95.20%	99.46%	83.12 %

5.2 Dữ liệu 2

5.2.1 Mô hình phân loại khi dữ liệu mất cân bằng

Bảng 4: Hiệu suất của các bộ phân loại trên dữ liệu không cân bằng

Classifier	Recall	F1-Score	AUC Score	Geometric Mean
Random Forest	75%	84%	87%	85%
XGBoost	72%	81%	86%	82%
Logistic Regression	57%	69%	78%	70%

- Trong thí nghiệm đầu tiên, đã huấn luyện một mô hình cơ bản cho từng bộ phân loại bằng cách sử dụng dữ liệu huấn luyện không cân bằng, sau khi thực hiện chọn đặc trưng và chia dữ liệu. Bảng 4 tóm tắt các chỉ số hiệu suất cho từng bộ phân loại. Các giá trị Recall thấp trong Bảng 4 cho thấy, mặc dù đạt độ chính xác cao, mô hình vẫn không dự đoán được các giao dịch gian lận. Điều này chỉ ra rằng mô hình bị overfitting nghiêm trọng do mất cân bằng lớp trong dữ liệu. Điều này cũng được xác minh bằng các giá trị AUC thấp.

  Trong nghiên cứu này, trong số 3 bộ phân loại, Random Forest vượt trội hơn hai bộ còn lại ở tất cả các chỉ số với Recall đạt 75%, trong khi Logistic Regression hoạt động kém nhất với giá trị Recall thấp nhất là 57%. Để kiểm chứng giả thuyết rằng "việc cân bằng dữ liệu bằng CT-GAN có thể cải thiện hiệu suất của các bộ phân loại hay

không", chúng em đã tiếp tục áp dụng các bộ phân loại này trên dữ liệu đã được cân bằng trong case 2.

5.2.2 Mô hình phân loại với dữ liệu đã cân bằng

- Trong nghiên cứu trường hợp thứ hai của chúng em, các bộ phân loại đã được kiểm tra trên dữ liệu cân bằng sử dụng CTGAN, và kết quả được tóm tắt trong Bảng 5. Rõ ràng từ các giá trị quan sát được trong bảng rằng sau khi huấn luyện các mô hình trên dữ liệu cân bằng, hiệu suất của chúng đã cải thiện đáng kể. Trong số tất cả các bộ phân loại, Random Forest đứng đầu với giá trị recall đạt 100% sau khi được huấn luyện trên dữ liệu cân bằng. Giá trị recall cao này cho thấy Random Forest không bỏ sót bất kỳ giao dịch gian lận nào trong toàn bộ dữ liệu. Ngoài ra, so với Logistic Regression (đạt recall 81%), Random Forest cũng hoạt động rất tốt. Nhìn chung, sự tăng đáng kể trong hiệu suất của các mô hình khẳng định tính hiệu quả của phương pháp được đề xuất. Chúng em sẽ xem xét chi tiết tác động của CTGAN lên tất cả các bộ phân loại trong phần tiếp theo.

Bảng 5: Hiệu suất của các bộ phân loại trên dữ liệu đã được cân bằng với CTGAN

Classifier	**Recall**	**F1-Score**	**AUC Score**	**Geometric Mean**
Random Forest + CTGAN	100%	100%	100%	100%
XGBoost + CTGAN	84%	84%	92%	84%
Logistic Regression + CTGAN	81%	81%	91%	81%

- Với trọng số cao của các giao dịch gian lận trong bài toán phân loại của chúng em,nhận thấy rằng không thể để mô hình dự đoán sai các giao dịch gian lận thành không gian lận. Để đảm bảo điều này, chúng em đã xem **Recall** là một chỉ số đánh giá quan trọng trong nghiên cứu . **Recall** thể hiện tỷ lệ phần trăm các giao dịch gian lận được mô hình phát hiện chính xác, do đó, giá trị **Recall** cao đồng nghĩa với hiệu suất cao của mô hình. Ngoài ra, **F1-score** thể hiện hiệu suất tổng thể của các mô hình vì nó cung cấp sự cân bằng giữa các chỉ số **Precision** và **Recall**. Giá trị **F1-score** cao hơn cho thấy mô hình dự đoán cả giao dịch gian lận và không gian lận với số lỗi tối thiểu.

- Chúng em nhận thấy rằng mô hình Random Forest vượt trội hơn tất cả các bộ phân loại khác trong mọi trường hợp và chứng minh là một bộ phân loại tốt hơn ngay cả khi áp dụng trên dữ liệu mất cân bằng. Bằng cách triển khai CT-GAN để tạo mẫu dữ liệu tổng hợp cho lớp thiểu số, chúng tôi đã cân bằng dữ liệu và huấn luyện các mô hình trên dữ liệu đã cân bằng. Qua nghiên cứu kết quả, chúng tôi có thể khẳng định rằng kỹ thuật tăng cường dữ liệu được đề xuất đã cải thiện hiệu suất mô hình trên mọi khía cạnh. Để đảm bảo kết quả đáng tin cậy từ các mô hình, chúng tôi chỉ sử dụng dữ liệu được tăng cường để huấn luyện và kiểm tra các mô hình đã huấn luyện bằng các mẫu thực từ dữ liệu gốc.

6 Tổng kết

Như đã thấy từ các nghiên cứu trước đây, vấn đề phát hiện gian lận thẻ tín dụng đã thu hút sự chú ý trong vài năm gần đây, và nhiều nghiên cứu đã được thực hiện nhằm đạt được

hiệu suất dự đoán tối ưu bằng cách sử dụng học máy. Như đã thảo luận trước đó, sự mất cân bằng lớp là một trở ngại lớn trong lĩnh vực phát hiện gian lận thẻ tín dụng.

Trong nghiên cứu này, Ở bộ dữ liệu 1 xảy ra tình trạng overfiting khi cố gắng tạo ra các lớp gian lận với số lượng mất cân bằng lớn lên tới 1 triệu dòng, Prccision quá thấp dẫn tới, F1-score không mang lại hiệu quả tốt, do đó việc cân nhắc sử dụng GMean và recall là tối ưu nhất. Ở bộ dữ liệu 2 chúng em đã sử dụng sự kết hợp lai giữa một phương pháp mô hình hóa dữ liệu - CT-GAN và kỹ thuật chọn đặc trưng - SelectKBest để giải quyết thách thức mất cân bằng lớp khi sử dụng các mô hình học máy. Chúng tôi đã phát triển 3 mô hình dựa trên phương pháp này. Về giá trị Recall và F1-score, Random Forest kết hợp với SelectKBest và CT-GAN đã vượt trội hơn tất cả các bộ phân loại khác (LR, XGB), đạt được 100% giá trị Recall và F1-score.

Ngoài ra, sau khi được huấn luyện trên dữ liệu tăng cường bằng CT-GAN, tất cả các bộ phân loại đều cho thấy sự cải thiện đáng kể về hiệu suất dự đoán tổng thể, đồng thời giảm các lỗi. Dữ liệu được sử dụng trong nghiên cứu này bao gồm toàn bộ các đặc trưng số. Trong tương lai, phương pháp này có thể được kiểm tra trên dữ liệu dạng catalogical để xác minh độ tin cậy của nó. Phương pháp đề xuất này cũng có thể được áp dụng để giải quyết các vấn đề phân loại khác trong các lĩnh vực khác.

7 Bàn luận

7.1 Những hạn chế và thách thức

- **Khả năng tổng quát hóa chưa cao**
  Mặc dù CTGAN đã chứng minh hiệu quả vượt trội trong việc sinh dữ liệu tổng hợp trên một số bộ dữ liệu tabular, mô hình vẫn tồn tại hạn chế khi xử lý các tập dữ liệu có cấu trúc phức tạp hơn. Đặc biệt, với các tập dữ liệu chứa nhiều biến danh mục đa cấp hoặc các mối quan hệ phi tuyến giữa các biến, khả năng tái hiện chính xác phân phối dữ liệu thực tế của CTGAN vẫn chưa được kiểm chứng đầy đủ. Đây là một thách thức quan trọng cần được giải quyết để mở rộng tính ứng dụng của mô hình.

- **Hiện tượng overfitting trong dữ liệu tổng hợp**
  Một vấn đề lớn chúng em nhận thấy trong quá trình thử nghiệm là CTGAN có xu hướng tái tạo dữ liệu tổng hợp quá giống với dữ liệu huấn luyện. Hiện tượng này dẫn đến overfitting, làm giảm khả năng tổng quát hóa của dữ liệu khi áp dụng vào các bài toán thực tế. Ngoài ra, việc sinh dữ liệu quá gần với dữ liệu gốc cũng đặt ra rủi ro tiềm ẩn về tính riêng tư, đặc biệt trong các ứng dụng nhạy cảm như tài chính hoặc y tế.

- **Chưa đa dạng hóa chỉ số đánh giá**
  Hiện tại, các tiêu chí đánh giá hiệu quả chủ yếu tập trung vào các chỉ số như F1-Score và Recall. Mặc dù đây là những chỉ số quan trọng trong các bài toán học máy, nhưng chúng chưa đủ để đánh giá toàn diện chất lượng dữ liệu tổng hợp. Các chỉ số như Jensen-Shannon Divergence (JSD) hoặc Total Variation Distance (TVD), vốn đo lường mức độ tương đồng giữa phân phối dữ liệu tổng hợp và dữ liệu gốc, vẫn chưa được khai thác một cách sâu sắc.

- **Phạm vi ứng dụng còn hạn chế**
  CTGAN hiện chủ yếu được thử nghiệm trong các bài toán phát hiện gian lận tài chính.

Tuy nhiên, trong các lĩnh vực khác như bảo hiểm, chăm sóc sức khỏe hoặc giao dịch ngân hàng – nơi dữ liệu dạng bảng thường có đặc điểm phức tạp và mang tính nhạy cảm cao – mô hình chưa được áp dụng rộng rãi. Điều này đặt ra nhu cầu thử nghiệm trên nhiều lĩnh vực hơn để khẳng định tính đa dụng của CTGAN.

- **Thách thức trong xử lý dữ liệu danh mục phức tạp**
  Với các tập dữ liệu danh mục có nhiều nhãn hoặc mức độ mất cân bằng nghiêm trọng, CTGAN gặp khó khăn trong việc tái hiện đầy đủ các nhãn thiểu số. Điều này không chỉ ảnh hưởng đến độ chính xác của dữ liệu tổng hợp mà còn có thể dẫn đến hiện tượng sinh ra các nhãn giả không phù hợp, làm suy giảm chất lượng của dữ liệu.

- **Thiếu đánh giá về tính bảo mật và riêng tư**
  Một khía cạnh quan trọng khác mà chúng em nhận thấy là việc đảm bảo tính bảo mật và riêng tư của dữ liệu tổng hợp. Điều này đặc biệt quan trọng khi ứng dụng trong các lĩnh vực yêu cầu tính nhạy cảm cao. Nghiên cứu hiện tại chưa có đánh giá sâu sắc về mức độ bảo mật hoặc khả năng tuân thủ các quy định pháp lý như GDPR và HIPAA khi sử dụng CTGAN để sinh dữ liệu.

- **Yêu cầu tài nguyên tính toán lớn**
  CTGAN đòi hỏi tài nguyên tính toán đáng kể, đặc biệt khi xử lý các tập dữ liệu lớn hoặc các phân phối phức tạp với nhiều đỉnh (modes). Điều này làm hạn chế khả năng ứng dụng của mô hình trong các hệ thống thực tế, nơi hiệu quả tính toán và chi phí phần cứng là yếu tố quan trọng. Việc tối ưu hóa kiến trúc và chiến lược huấn luyện để giảm thiểu yêu cầu tài nguyên vẫn là một thách thức cần giải quyết.

7.2 Định hướng trong tương lai

- Trong tương lai, chúng em sẽ tập trung vào việc nâng cao khả năng tổng hợp dữ liệu của CTGAN, đặc biệt đối với các tập dữ liệu có cấu trúc phức tạp và chứa nhiều mối quan hệ phi tuyến. Các phương pháp tối ưu hóa như cải tiến kiến trúc mạng, giảm thiểu thời gian huấn luyện, và tích hợp với các cơ chế bảo mật như Differential Privacy sẽ được nghiên cứu nhằm tăng tính ứng dụng thực tiễn và đảm bảo dữ liệu tổng hợp không làm lộ thông tin nhạy cảm. Đồng thời, chúng em cũng hướng đến việc mở rộng ứng dụng CTGAN trong các lĩnh vực mới như bảo hiểm, chăm sóc sức khỏe, và phân tích hành vi khách hàng, từ đó khẳng định vai trò của mô hình trong việc hỗ trợ giải quyết các bài toán dữ liệu phức tạp.

- Bên cạnh đó, các tiêu chí đánh giá chất lượng dữ liệu sẽ được mở rộng để không chỉ tập trung vào hiệu quả của mô hình học máy mà còn phản ánh độ đa dạng và sự tương đồng phân phối giữa dữ liệu tổng hợp và dữ liệu gốc. Chúng em cũng kỳ vọng ứng dụng CTGAN và những biến thể của GANs vào các hệ thống thời gian thực, tích hợp với các công cụ phân tích dữ liệu hiện đại để hỗ trợ việc ra quyết định trong các tổ chức. Và hơn hết là đảm bảo tính công bằng, sự an toàn của mỗi cái nhân, tổ chức và doang nghiệp trong thị trường tài chính có nhiều biến động

8 Tài liệu tham khảo

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2672-2680. Link.

- Xu, L., Skoularidou, M., Cuesta-Infante, A., & Veeramachaneni, K. (2019). Modeling tabular data using Conditional GAN. *Advances in Neural Information Processing Systems*, 32, 7335-7345. Link.

- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321-357. Link.

- Douzas, G., Bacao, F., & Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*, 465, 1-20. Link.

- Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*. Link.

- Fiore, U., De Santis, A., Perla, F., Zanetti, P., & Palmieri, F. (2019). Using generative adversarial networks for improving classification effectiveness in credit card fraud detection. *Information Sciences*, 479, 448-455. Link.

- Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications*, 39(3), 3446-3453. Link.

- Yazıcı, H., Sharma, G., Alelyani, S., & Tan, P. N. (2015). A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 49(1), 1-37. Link.

- Chalapathy, R., & Chawla, S. (2019). Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*. Link.

- Ngai, E. W., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision Support Systems*, 50(3), 559-569. Link.