PHÁT HIỆN GIAN LẬN THẺ TÍN DỤNG BẰNG MÔ HÌNH AUTOENCODER

Trần Minh Trị, Trần Tiến Triệu, Châu Nguyễn Ngọc Trân, Đỗ Gia Hân, Nguyễn Đàm Lệ Thương, Trần Thành Công *
Trường Đại học Kinh tế - Tài chính TP.HCM

TÓM TẮT

Tình trạng gian lận thẻ tín dụng luôn là vấn đề vô cùng nhức nhối và có xu hướng ngày càng gia tăng trong thời đại mà việc thanh toán trực tuyến phát triển mạnh mẽ như hiện nay. Có nhiều hình thức gian lận thẻ tín dụng, từ việc lấy trộm thông tin thẻ tín dụng của người dùng qua các phương tiện mạng, đến việc hấp dẫn khách hàng đưa thông tin thẻ tín dụng để lừa đảo. Khi thông tin thẻ tín dụng bị lộ ra ngoài, kẻ gian lận có thể sử dụng thông tin này để thực hiện các hoạt động trực tuyến như mua hàng, rút tiền, lừa đảo qua email, xấu dụng điểm tín dụng, và nhiều hình thức khác. Để đối phó với tình trạng gian lận thẻ tín dụng, các tổ chức tài chính và người dùng cần nâng cao nhận thức bảo mật về an toàn thông tin, các mối đe dọa tiềm ẩn khi sử dụng thẻ tín dụng. Trong bài báo này, chúng tôi đưa ra một giải pháp phát hiện gian lận thẻ tín dụng bằng việc áp dụng mô hình Autoencoder và sử dụng mạng nơron nhân tạo (neural network) để phân loại và dự đoán dữ liệu dựa trên tiếp nhận một đầu vào và đưa ra một đầu ra.

Từ khóa: *Deep learning, Autoencoder, mất cân bằng dữ liệu, phát hiện gian lận*

I. TỔNG QUAN

Cùng với sự bùng nổ của công nghệ chuyển đổi số như hiện nay, ngân hàng số đóng vai trò quan trọng trong việc chuyển đổi hành vi tiêu dùng như việc thanh toán trực tuyến, vay trực tuyến hay đầu tư. Thanh toán điện tử, đặc biệt là thanh toán trên các nền tảng di động đang có những bước phát triển mạnh mẽ. Tuy nhiên, điều đó cũng kéo theo sự gia tăng về số lượng những vụ việc liên quan đến an ninh thanh toán, gian lận tài khoản tín dụng, lừa đảo tài chính. Công nghệ 5G và các công nghệ 4.0 giúp tạo ra thế giới siêu kết nối nhưng cũng là nhân tố hình thành nên ngành công nghiệp "đánh cắp tiền" trị giá đến hàng tỷ đô.

Theo dữ liệu được công bố bởi TransUnion, thẻ tín dụng là phương thức thanh toán trực tuyến được sử dụng nhiều nhất trên thế giới. Vì thế các vụ gian lận cũng ngày càng trở ngày càng tinh vi và phổ biến hơn. Báo cáo của Ủy

ban Thương mại Liên bang Hoa Kỳ (FTC) cho thấy lận tín dụng ở thị trường này đạt con số 16,7 triệu nạn nhân trong năm 2017, số vụ khiếu nại gian lận thẻ tín dụng năm 2017 tăng đáng kể và số tiền bị đe dọa sẽ vượt quá khoảng 30 tỷ đô la vào năm 2020.

Hiện nay, trên thị trường cũng có rất nhiều giải pháp, ví dụ như Xác thực 2 yếu tố (2FA). Tuy 2FA được nhiều người biết đến và dễ sử dụng nhưng vẫn còn nhiều hạn chế: SMS 2FA yêu cầu người dùng tiết lộ số điện thoại của họ cho bên thứ ba (nhà cung cấp 2FA) (lo ngại về quyền riêng tư, bảo mật cá nhân và việc trở thành mục tiêu cho quảng cáo) hay TOTP 2FA yêu cầu người dùng phải có thiết bị có khả năng đọc mã QR để xác minh danh tính của họ (Nếu người dùng đặt nhầm thiết bị hoặc mã QR hoặc nếu thiết bị bị đánh cắp, họ sẽ không thể truy cập thông tin của mình nữa).

Trong bài báo này, mô hình được áp dụng có tên gọi là Autoencoder, dùng để phát hiện gian lận trong giao dịch thẻ tín dụng, mô hình được sử dụng trên tập dữ liệu mất cân bằng. Sau quá trình huấn luyện, mô hình có khả năng tái tạo khá hiệu quả với các sự kiện bình thường, nhưng đối với các hiện tượng gian lận, sự kiện gian lận bất thường lại gây ra hiện tượng gian lận. Vì thế, những dữ liệu sau khi chạy qua mô hình huấn luyện Autoencoder có các đặc trưng khác so với đầu vào thì có khả năng là sự kiện gian lận trong giao dịch.

Phần còn lại của bài viết có cấu trúc như sau: **phần 2** giới thiệu về phương pháp học sâu (Deep learning), **phần 3** mô tả mô hình áp dụng - Autoencoder, **phần 4** trình bày các phương pháp đánh giá mô hình và kết quả, **phần 5** là kết luận và hướng phát triển trong tương lai.

II. PHƯƠNG PHÁP HỌC SÂU – DEEP LEARNING

Deep Learning là một phương pháp học máy (machine learning) dựa trên một mạng lưới các neuron nhân tạo, được thiết kế để tự động học và tinh chỉnh các đặc trưng (features) trong dữ liệu đầu vào để thực hiện các nhiệm vụ phân loại, nhận dạng, dự đoán, hoặc xử lý thông tin phức tạp khác. Deep Learning sử dụng các mô hình mạng neuron sâu (deep neural networks) với hàng chục hoặc hàng trăm lớp để tìm ra các kết quả đầu ra phù hợp cho một tác vụ cụ thể. Mỗi lớp trong mạng neuron đóng vai trò như một bộ lọc (filter) giúp học cách biểu diễn các đặc trưng của dữ liệu, từ các đặc trưng đơn giản đến các

đặc trưng phức tạp hơn. Quá trình này giúp tăng độ chính xác của mô hình, giảm độ phức tạp của quá trình lập trình và tăng khả năng tự động hóa trong việc xử lý dữ liệu.

Deep Learning có rất nhiều thuật toán như Convolutional Neural Network (CNN), Deep Belief Network (DBN), Deep Neural Network (DNN), Recurrent Neural Network (RNN), Boltzmann Machine (BM) và một trong số đó là mô hình Autoencoder được sử dụng trong bài. Các thuật toán Deep Learning có khả năng tự động xây dựng và khai thác các mẫu có trong dữ liệu để đưa ra được quyết định tối ưu. Tuy nhiên, việc này cần rất nhiều dữ liệu và tài nguyên tính toán để có được độ chính xác tốt nhất bởi mỗi mô hình mạng nơ-ron nhân tạo có thể bao gồm hàng trăm, thậm chí hàng triệu tham số khác nhau. Chính vì vậy việc tối ưu các tham số này đòi hỏi người xây dựng mô hình phải có kiến thức chuyên sâu và nhiều kinh nghiệm.

III. MÔ HÌNH AUTOENCODER

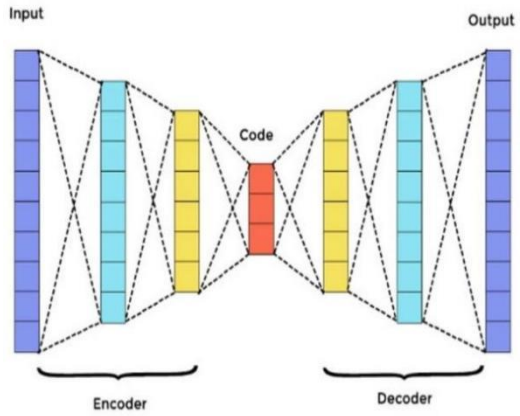
Để thuận lợi trong việc trích xuất các đặc trưng từ dữ liệu và tăng độ chính xác cho hệ thống phát hiện gian lận thẻ tín dụng, trong bài nghiên cứu này sử dụng phương pháp phát hiện gian lận dựa trên mô hình Autoencoder.

Autoencoder hay còn gọi là bộ mã hóa tự động là một loại phương pháp Deep Learning, một dạng của ANN (Artificial neural network) có khả năng học một cách hiệu quả các biểu diễn của dữ liệu đầu vào mà không cần dùng nhãn, điều này có nghĩa là mạng có khả năng học không cần giám sát. Artificial neural network hay còn gọi là mạng nơ-ron thần kinh nhân tạo là một mô hình toán học phức tạp được đưa ra để xử lý thông tin, giải quyết các vấn đề phổ biến trong lĩnh vực học máy, trí tuệ nhân tạo AI, học sâu và tìm kiếm tất cả các mối quan hệ cơ bản trong một tập hợp các dữ liệu. Autoencoder bao gồm 3 phần chính: Encoder, Code và Decoder, cụ thể như sau:

- Encoder (bộ mã hóa): Một module có chức năng nén dữ liệu đầu vào của bộ kiểm tra xác thực thành một biểu diễn được mã hóa. Thông thường nó sẽ nhỏ hơn một vài bậc so với dữ liệu đầu vào.
- Code: Một module chứa các biểu diễn tri thức đã được nén hay còn gọi là output của Encoder, đây là phần quan trọng nhất trong mạng

vì nó mang đặc trưng của dữ liệu đầu vào, có thể sử dụng để lấy đặc trưng của ảnh, tái tạo hình ảnh…

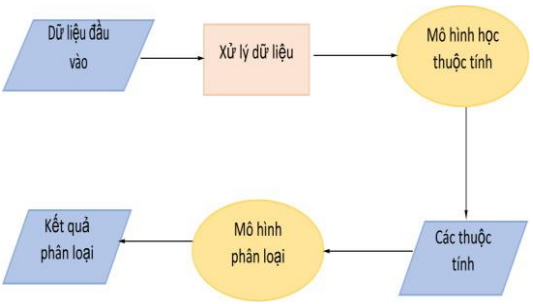
– Decoder (bộ giải mã): Module hỗ trợ mạng giải nén các biểu diễn tri thức và tái tạo lại cấu trúc dữ liệu từ dạng mã hóa của nó. Mô hình học dựa theo việc so sánh đầu ra của Decoder với đầu vào bạn đầu của nó.



Hình 1: Cấu trúc mô hình Autoencoder

Autoencoder xây dựng mô hình học các đặc tính của dữ liệu dựa trên mạng neural nhiều lớp. Thay vì các thuộc tính được thiết kế giả lập, các đặc tính được học từ mạng lưới này sẽ được nhập vào mô hình phân loại để phát hiện các hành vi gian lận.

Từ những dữ liệu đầu vào thông qua việc phân chia và xử lý dữ liệu để có thể sử dụng vào việc huấn luyện, xây dựng mô hình học thuộc tính. Xây dựng mô hình học thuộc tính bằng cách đưa liên tục các bản ghi "normal" vào mô hình để cho mô hình có thể "học" các thuộc tính để nhận dạng được dữ liệu giao dịch bình thường. Việc đưa các bản ghi "normal" vào mô hình autoencoder là để tối ưu các tham số theo dữ liệu normal, khi đưa dữ liệu normal vào nó sẽ tái tạo ra một cái đầu ra gần giống nhất so với dữ liệu đầu vào. Nhưng ngay khi chúng ta đưa dữ liệu bất thường vào, dữ liệu bất thường không được train trên mô hình nên nó sẽ đưa ra kết quả đầu ra không giống với cái đầu vào, dựa trên giá trị loss. Các thuộc tính sau khi học được từ mô hình học thuộc tính sẽ được nhập vào mô hình phân loại từ đó cho ra kết quả phân loại các giao dịch bình thường và giao dịch bất thường trong việc giao dịch thẻ tín dụng.



Hình 2: Quy trình phân loại dữ liệu

A. QUY TRÌNH THỰC NGHIỆM

1. Công việc thực hiện:

- Import dữ liệu đầu vào

- Xử lý dữ liệu
- Phân chia bộ dữ liệu
- Xây dựng và huấn luyện mô hình

2. Về bộ dữ liệu

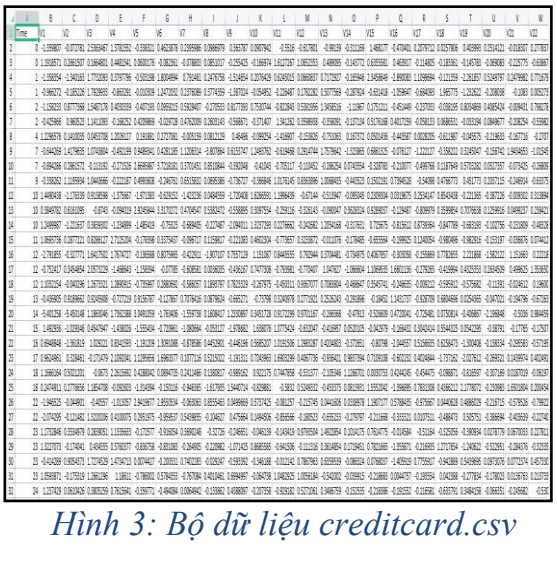
Bộ dữ liệu gồm có 284,807 vectơ kết nối đơn, mỗi vectơ có 31 thuộc tính. Bộ dữ liệu chứa các giao dịch được thực hiện bằng thẻ tín dụng vào tháng 9 năm 2013 bởi các chủ thẻ châu Âu. Tập dữ liệu này trình bày các giao dịch xảy ra trong hai ngày có 492 gian lận trong số 284.807 giao dịch. Dữ liệu được lấy trên thực tế, để số lần giao dịch gian lận ra có tỉ lệ thấp hơn nhiều so với giao dịch thành công bình thường, giao dịch gian lận chiếm chỉ 0,172% tổng số giao dịch. Vậy nên bộ dữ liệu có được rất mất cân xứng.

Giảm chiều dữ liệu trong Machine Learning là quá trình giảm thiểu số lượng đặc trưng biểu diễn dữ liệu. Việc này có thể được thực hiện theo hướng lựa chọn các đặc trưng quan trọng trong các đặc trưng mới từ các đặc trưng đã cho. PCA chính là phương pháp đi tìm một hệ cơ sở mới sao cho thông tin của dữ liệu chủ yếu tập trung ở một vài toạ độ, phần còn lại chỉ mang một lượng nhỏ thông tin. Và để cho đơn giản trong tính toán, PCA sẽ tìm một hệ trực chuẩn để làm cơ sở mới. Trong tập dữ liệu, các biến đầu vào số là kết quả của một phép biến đổi PCA. Do các vấn đề bảo mật, bên cung cấp không thể cung cấp các tính năng gốc và thêm thông tin cơ bản về dữ liệu. Các tính năng V1, V2, V28 là các thành phần chính có được với PCA, các tính năng duy nhất không được chuyển đổi với PCA là 'Thời gian' và 'Số lượng'. Tính năng 'Thời gian' chứa số giây trôi qua giữa mỗi giao dịch và giao dịch đầu tiên trong tập dữ liệu. Tính năng 'Số tiền' là Số tiền giao dịch, tính năng này có thể được sử dụng cho việc học nhạy cảm với chi phí phụ thuộc vào ví dụ. Tính năng 'Lớp' là biến phản hồi và nó nhận giá trị 1 trong trường hợp gian lận và 0 nếu không.

Ý tưởng chính của cách tiếp cận này là nén dữ liệu tạo thành một "biểu diễn tiềm ẩn" và sau đó tái tạo lại dữ liệu. Nếu dữ liệu được tái tạo không giống với dữ liệu ban đầu, thì những dữ liệu đó có hành vi gian lận. Bằng cách sử dụng phân tích thành phần chính PCA, chúng ta có thể so sánh bố cục dữ liệu gốc với phân phối dữ liệu nén PCA.

Hình dạng của tập dữ liệu: (284807, 31)

Không có giá trị Null trong tập dữ liệu



Hình 3: Bộ dữ liệu creditcard.csv

3. Tiến hành thực nghiệm:

3.1 Lấy dữ liệu

Đầu tiên, tiền xử lý bộ dữ creditcard.csv bằng các vectơ đặc trưng chuyển đổi về các giá trị số. Hàm xây dựng vectơ đặc trưng cho mạng Neural như sau:

- Dự đoán / Đầu vào:
  - Mã hóa các giá trị văn bản, phân loại bằng encodetext.
  - Mã hóa các giá trị số bằng encode numeric.
- Đầu ra:
  - Mã hóa các giá trị văn bản, phân loại bằng encode index.
  - Không mã hóa các giá trị số đầu ra.

3.2 Clean Data

Loại bỏ các trường không cần thiết Trong các trường đầu vào của dữ liệu, có 2 trường không có nhiều ý nghĩa cho mô hình đào tạo là trường "time" chỉ thời điểm giao dịch, và trường "số tiền" chứa giá trị phân loại kiểu lựa chọn một vài giá trị tự nhiên xác định nên loại bỏ. Như vậy đầu vào sẽ còn 29 trường dữ liệu liên tục đã được chuẩn hóa.

Để bắt đầu quá trình huấn luyện, nhóm bắt đầu xử lý tệp dữ liệu bằng cách loại bỏ cột Thời gian (sẽ không sử dụng nó) và sử dụng scikit của StandardScaler trên cột Số tiền. Bộ chia tỷ lệ loại bỏ giá trị trung bình và chia tỷ lệ giá trị thành phương sai theo đơn vị.

```
# Drop cột time
df_no_time = df.drop(columns=['Time'])

from sklearn.preprocessing import StandardScaler
df_no_time['Amount'] = StandardScaler().fit_transform(df_no_time['Amount'].values.reshape(-1,1))
```

Hình 4:Lệnh giúp loại bỏ các trường không cần thiết

3.3 Phân chia bộ dữ liệu

Đánh giá hiệu quả của bộ phân lớp là việc quan trọng, bởi vì nó cho phép dự đoán được độ chính xác của các kết quả phân lớp những dữ liệu trong tương lai. Độ chính xác còn giúp so sánh các mô hình phân lớp khác nhau.

Trong các bài toán thực tế, để đánh giá tính hiệu quả, tính chính xác của mô

hình cần phải có 3 bộ dữ liệu: train, valid và test.

- Bộ dữ liệu train: dùng để huấn luyện cho mô hình.
- Bộ dữ liệu valid: dùng để đánh giá tính hiệu quả, độ chính xác của mô hình, từ đó quyết định lựa chọn mô hình và các tham số phù hợp.
- Bộ dữ liệu test: dùng để xác thực tính hiệu quả, độ chính xác cuối cùng. Đây coi như một bộ dữ liệu chưa biết, không dùng bộ dữ liệu test để lựa chọn mô hình và các tham số tối ưu như bộ dữ liệu valid.

Trong phần này, nghiên cứu đã xác định mô hình khảo sát cụ thể là MLP Autoencoder và khảo sát tất cả các tham số chứ không phải nhằm mục đích tối ưu các tham số phù hợp nên không cần sử dụng bộ dữ liệu valid. Chính vì vậy trong phạm vi nghiên cứu này chúng tôi chia bộ dữ liệu thành 2 bộ dữ liệu con train và test với tỉ lệ số lượng mẫu là 80:20. Việc huấn luyện mô hình Autoencoder trong nghiên cứu này sẽ khác một chút so với những đề tài đã được công bố trước đó. Với tập dữ liệu để huấn luyện cho đề tài này là một tập dữ liệu chứa rất nhiều giao dịch không gian lận. Để phát hiện bất kỳ gian lận nào trên các giao dịch mới. Trong phạm vi nghiên cứu này nhóm đã xây dựng một mô hình học không giám sát và huấn luyện mô hình đó của tôi về các giao dịch thông thường. Việc phân đúng lớp trên tập thử nghiệm sẽ cung cấp một cách để đánh giá hiệu suất của mô hình của nghiên cứu này.

Xây dựng mô hình Autoencoder để huấn luyện và kiểm thử. Mô hình Autoencoder được đề xuất xây dựng và huấn luyện dữ liệu là lớp đầu vào, 3 lớp ẩn và lớp đầu ra như dưới đây. Trong đó, lớp ẩn thứ nhất là 128 node, lớp ẩn thứ hai là 64 node, lớp ẩn thứ ba là 32 node. Sử dụng:

- Hàm "relu" cho phân loại lớp đầu ra.
- Hàm tối ưu: Adam (tham số trong thư viện Keras) learning_rate là 0.00001.
- Hàm loss: mse - Epochs: 50 (**epochs** là số lần quá trình huấn luyện học qua tất cả các tập dữ liệu trong tập huấn luyện).

3.4 Xây dựng mô hình Autoencoder

```
from keras.layers import Input, Dense
from keras.models import load_model, Model
from keras.callbacks import ModelCheckpoint

# Xây dựng model
input_dimension = X_train.shape[1]
hidden_size = 128

input_layer = Input(shape=(input_dimension,))
#Encoder
encoder = Dense(hidden_size, activation="relu") (input_layer)
encoder = Dense(hidden_size //2, activation="relu") (encoder)
encoder = Dense(hidden_size //4, activation="relu") (encoder)

# Decoder
decoder = Dense(hidden_size //4, activation="relu") (encoder)
decoder = Dense(hidden_size //2, activation="relu") (decoder)
decoder = Dense(hidden_size, activation="relu") (decoder)

# Output
output_layer = Dense(input_dimension, activation ="relu") (decoder)

auto_encoder_model= Model(inputs= input_layer, outputs= output_layer)
auto_encoder_model.compile(optimizer ="adam", loss="mse", metrics=['accuracy'])
auto_encoder_model.summary()
```

Hình 5: Lập trình mạng Neural Autoencoder

Bài nghiên cứu sử dụng bộ dữ liệu train làm cả đầu vào và đầu ra cho mô hình, sau đó huấn luyện mô hình 50 chu kỳ huấn luyện với kích thước lô là 32. và lưu mô hình hoạt động tốt nhất vào một tệp. ModelCheckpoint do Keras cung cấp thực sự tiện dụng cho những công việc như vậy.

```
# train model
n_epochs = 50
n_batch_size = 32

save_best = ModelCheckpoint(filepath="/content/gdrive/MyDrive/Fraud_data/best.h5",
                            monitor='val_loss', verbose=1, save_best_only=True)
history = auto_encoder_model.fit(X_train, X_train, batch_size=n_batch_size,
                                 epochs=n_epochs, verbose=1, shuffle=True,
                                 callbacks=[save_best], validation_data=(X_test, X_test))
```

Hình 6:. Mô-đun huấn luyện mô hình Autoencoder

IV. PHƯƠNG PHÁP ĐÁNH GIÁ MÔ HÌNH VÀ KẾT QUẢ

A. ĐÁNH GIÁ MÔ HÌNH:

Các tiêu chí được sử dụng để đánh giá hiệu năng của hệ thống phát hiện gian lận: Trong bài toán này, người ta thường định nghĩa lớp dữ liệu quan trọng hơn cần được xác định đúng là lớp Positive (P-dương tính), lớp còn lại được gọi là Negative (N-âm tính). Ta định nghĩa True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN) dựa trên confusion matrix chưa chuẩn hóa như sau:

		Actual Values: Positive (1)	Actual Values: Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Người ta thường quan tâm đến TPR, FNR, FPR, TNR (R - Rate) dựa trên normalized confusion matrix như sau:

predicted→ / real↓	Class_pos	Class_neg
Class_pos	TP	FN
Class_neg	FP	TN

→

predicted→ / real↓	Class_pos	Class_neg
Class_pos	TP/pos	FN/pos
Class_neg	FP/neg	TN/neg

↓

predicted→ / real↓	Class_pos	Class_neg
Class_pos	TPR	FNR
Class_neg	FPR	TNR

←

– False Positive Rate còn được gọi là False Alarm Rate (tỉ lệ báo động nhầm).

– False Negative Rate còn được gọi là Miss Detection Rate (tỉ lệ bỏ sót).

– Accuracy (Độ chính xác) là tỷ lệ số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử:

$$\square = \frac{\square\square + \square\square}{\square\square + \square\square + \square\square + \square\square}$$

– Recall là tỷ lệ số điểm true positive trong số những điểm thực sự là positive (TP + FN):

$$\square\square\square\square\square\square = \square\square\square = \frac{\square\square}{\square\square + \square\square}$$ (TPR: True Positive Rate)

– Precision (P) là thước đo một hệ thống có khả năng phát hiện bình thường hay bất thường:

$$\square = \frac{\square\square}{\square\square + \square\square}$$

– F1 score là harmonic mean của Precision và Recall, sử dụng để đánh giá bộ phân lớp có công thức sau:

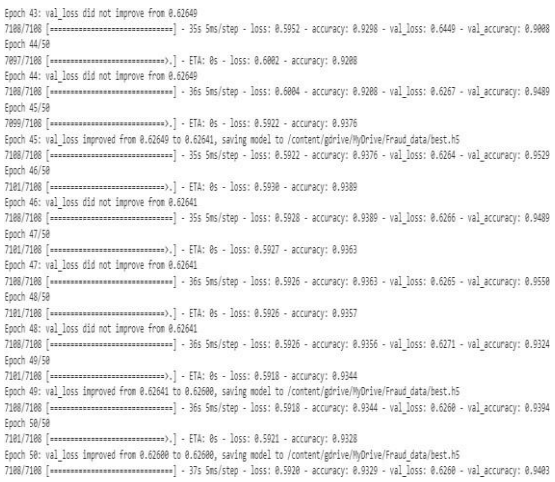
$$\square 1 = \frac{1}{\frac{1}{\square\square\square\square\square\square\square\square\square} + \frac{1}{\square\square\square\square\square\square}}$$

Trong đó:

- True Positives (TP) là số lượng các bất thường được phân loại đúng là bất thường.
- True Negatives (TN) là số lượng các bình thường được phân loại đúng là bình thường. False Positives (FP) là số lượng các bình thường được phân loại sai là bất thường.
- False Negatives (FN) là số lượng các bất thường được phân loại sai thành bình thường.

B. KẾT QUẢ

Lịch sử huấn luyện mô hìn



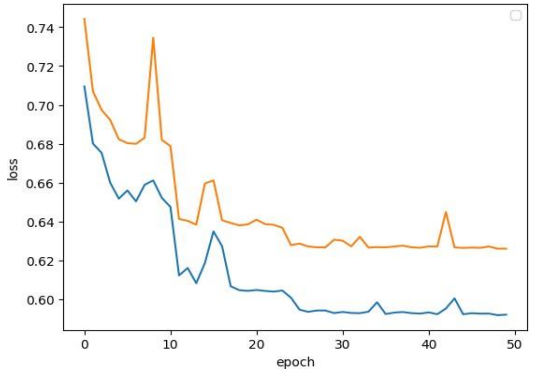
Hình 7: Kết quả xử lý cho 50 epoch

Sau khi chạy lệnh xong, kết quả thực nghiệm như sau:

– Epoch 50/50: đây là 50 lần quá trình huấn luyện dữ liệu học qua

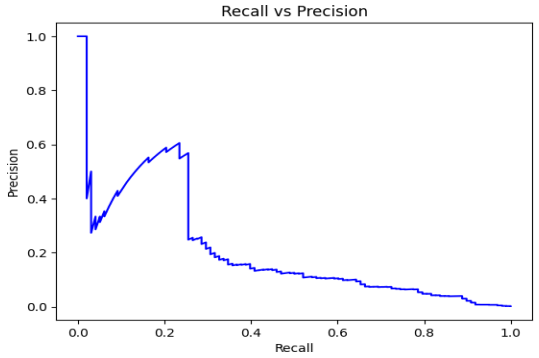
- tất cả dữ liệu có trong tập huấn luyện.
- Loss: 0.592 Đây là độ lỗi trên tập train trong epoch này.
- Accuracy: 0.9329. Đây là độ chính xác trên tập train. Có nghĩa là mạng đã đạt được 93,29% trên tập train.
- Val_loss: 0.626. Đây là độ lỗi trên tập test trong epoch này.
- Val_accuracy: 0.9403

Đây là độ chính xác trên tập test. Có nghĩa mô hình đã đạt được 93,29% trên tập test, thời gian huấn luyện và xử lý cho từng epoch trung bình là 36s.



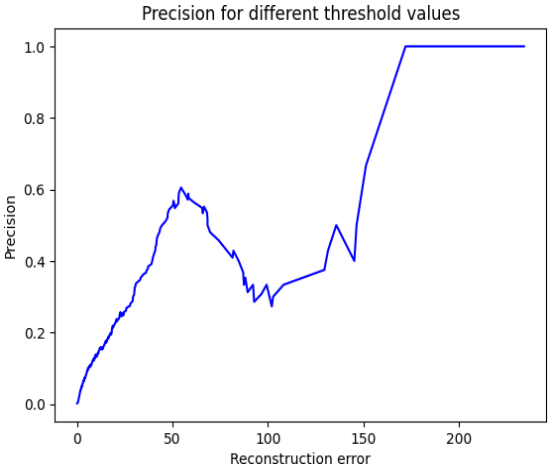
Hình 9: Lịch sử huấn luyện mạng Autoencoder

Trước tiên nhìn vào hình 8 lịch sử huấn luyện mạng Autoencoder với kích thước khối mã bằng 128 dần hội tụ sau 50 chu kỳ huấn luyện của mô hình, sai số trên cả tập train và test đều giảm dần trong đó mất mát ở tập test có xu hướng hội tụ quanh mốc 0.63.



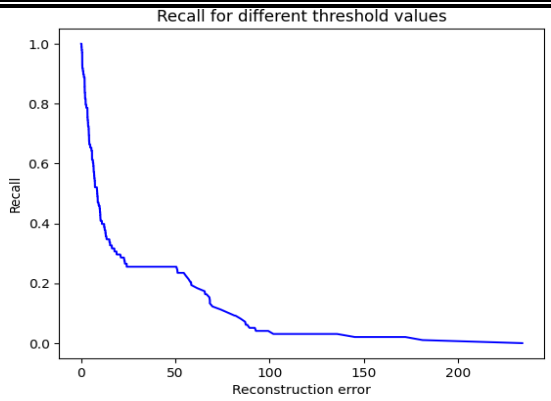
Hình 8: Biểu đồ thể hiện sự tương quan giữa recall và precision

Vùng cao dưới đường cong thể hiện cả khả năng thu hồi cao và độ chính xác cao, trong đó precision mà cao thì liên quan đến tỷ lệ dương tính giả thấp và recall cao thì liên quan đến tỷ lệ âm tính giả thấp. Điểm cao cho cả hai cho thấy trình phân loại đang trả về kết quả chính xác (precision), cũng như trả về phần lớn tất cả các kết quả tích cực (recall).



Hình 10: Biểu đồ thể hiện sự tương quan giữa Precision và Reconstruction error

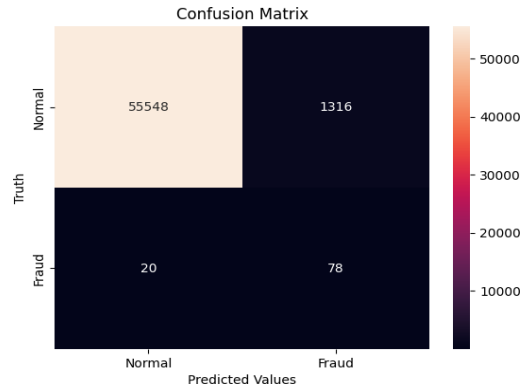
Qua nghiên cứu, chúng tôi phát hiện ra rằng khi lỗi tái tạo tăng (reconstruction error) thì độ chính xác cũng tăng theo.



Hình 11:Biểu đồ thể hiện tương quan giữa Recall và Reconstruction error.

Tuy nhiên, khi so sánh với Recall chúng ta lại thu được một kết quả hoàn toàn ngược lại. Khi lỗi tăng, mức thu hồi giảm.

Để dự đoán liệu một giao dịch mới/chưa thấy là bình thường hay gian lận, chúng tôi sẽ tính toán lỗi tái tạo từ chính dữ liệu giao dịch. Nếu lỗi lớn hơn ngưỡng được xác định trước, chúng tôi sẽ đánh dấu đó là gian lận (vì mô hình của chúng tôi sẽ có lỗi thấp đối với các giao dịch thông thường). Vậy nên ở nhóm sẽ chọn giá trị cho ngưỡng được cho là gian lận là threshold = 2.5. Và sau đó sẽ vẽ confusion matrix để đánh giá kết quả.



Hình 12: Confusion matrix mạng Autoencoder

Mặc dù vẫn còn tồn tại một số nhược điểm, số lượng các giao dịch thông thường được phân loại là gian lận thực sự rất cao. Tuy nhiên, mô hình mà chúng tôi đề xuất đã mang lại hiệu quả thực sự trong việc phát hiện các hình thức gian lận thẻ tín dụng.

Kết quả cài đặt thử nghiệm Deep Learning bằng cách sử dụng mô hình Autoencoder trên tập dữ liệu cho thấy độ chính xác và hiệu quả của mô hình này khá cao. Điều này cũng cho thấy việc ứng dụng Deep Learning vào mô hình phát hiện gian lận thẻ tín dụng trong thời đại số là rất quan trọng và đáng được phát triển hơn nữa.

So sánh với các phương pháp đã có hiện nay

Các phương pháp xử lý dữ liệu và mô hình phát hiện gian lận đều không chiếm ưu thế hoàn toàn. Người sử dụng phải cân nhắc giữa tính hợp lý, độ ổn định và sức mạnh cũng như tính phức tạp khi thực hiện với mỗi mô hình.

1. Công cụ lấy mẫu theo phương pháp Oversampling có ưu điểm dễ thực hiện, tuy tăng về kích thước dữ liệu nhưng các quan sát được lặp lại nên trong một số trường hợp phương pháp thể hiện tính không hiệu quả.

2. Phương pháp SMOTE giải quyết được vấn đề dữ liệu mất cân bằng. Tuy nhiên, phương pháp chỉ hiệu quả với một số lượng gian lận rất thấp, khối lượng tính toán phức tạp và phải thực hiện lặp đi lặp lại nhiều lần. Trong quá trình thực hiện, phương pháp tạo ra các quan sát nhiễu và phải xử lý dữ liệu bị thiếu trước khi thực hiện tăng số quan sát.
3. Với phân tích phân nhóm sử dụng WOE, mô hình Logistic là phương pháp truyền thống được cải tiến, đơn giản, hiệu quả, dễ áp dụng trên nhiều hệ thống, tính ổn định tốt, kiểm soát hoàn toàn được mô hình. Tuy nhiên, mô hình Logistic gặp phải vấn đề dữ liệu mất cân bằng và phải phân tích WOE hợp lý trước khi chạy mô hình.
4. Mạng Bayesian là phương pháp áp dụng được trên tất cả các loại dữ liệu. Phương pháp đem lại kết quả tốt nhưng dễ bị phân loại nhầm bởi các quan sát nhiễu hay lặp lại. Kết quả cho thấy mạng Bayesian ứng dụng hiệu quả nhất với các mô hình có lượng quan sát ít.

Từ kết quả của mô hình mà nhóm sử dụng phân tích. Mô hình cho thấy được tính ứng dụng cao, dễ dàng sử dụng, độ chính xác cao hơn, giải quyết được vấn đề mất cân bằng dữ liệu. Mô hình sử dụng vừa đảm bảo được độ chính xác, vừa đảm bảo được thời gian thực, độ trễ thấp, kết quả phân tích từ đó cũng có tính xác thực hơn.

Mô hình	Dữ liệu sử dụng	FN	TN	FP	TP	Độ chính xác
Logistic (1)	SMOTE	88	284	260	404	66,41%
	Oversampling	86	284	245	406	**67,58%**
Cây quyết định (2)	SMOTE	63	284	118	429	79,76%
	Oversampling	108	284	106	384	75,74%
	Gốc	99	284	31	393	**83,90%**
Mạng Bayesian (3)	SMOTE	74	283	1359	418	32,85%
	Oversampling	79	283	1714	413	27,95%
	Gốc	90	284	294	402	**64,11%**
Stacking (4)	SMOTE	65	284	348	427	63,26%
	Oversampling	79	284	287	413	**65,57%**
	Gốc	77	284	301	415	64,90%

Nguồn: Tác giả tính toán dựa trên bộ dữ liệu xử lý.

Hình 13: Bảng kết quả chạy mô hình

V. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

Trong bài viết này, nhóm đã trình bày và khái niệm được một số thuật toán và khái niệm trong Deep learning, cụ thể hơn là Autoencoder. Từ đó, nhóm đã xây dựng được mô hình Autoencoder, trình bày được một phương pháp để chỉ ra và phát hiện vi phạm liên quan đến việc gian lận thẻ tín dụng. Phương pháp yêu cầu các nhiệm vụ tính toán được thực hiện bởi máy tính, máy tính sau khi qua việc huấn luyện sẽ có thể nhận dạng được những giao dịch hợp lệ với tỉ lệ chính

xác cao.

Chúng tôi đã nghiên cứu các mô hình mạng neural, cụ thể đã ứng dụng mô hình huấn luyện Autoencoder vào bài toán phát hiện gian lận trong giao dịch thẻ tín dụng để cho ra kết quả là phân lớp đúng các giao dịch. Tuy rằng kết quả của mô hình chưa đạt được độ chính xác tuyệt đối, nhưng nó đã giải quyết phần nào việc mất cân bằng dữ liệu của bài toán. Mô hình Autoencoder xây dựng nghiên cứu được cải tiến và hiệu chỉnh đã cho ra những kết quả khả quan, có thể áp dụng được vào hệ thống thực tế. Phương pháp Deep Learning xây dựng mô hình học tính năng dựa trên kỹ thuật Autoencoder học tốt tính năng và cải thiện độ chính xác trong phát hiện gian lận thẻ tín dụng.

Trong tương lai, chúng tôi sẽ tiếp tục nghiên cứu thử nghiệm và cải tiến để giảm thời gian huấn luyện và có hiệu quả, độ chính xác tốt hơn, tăng khả năng thu hồi, giảm khả năng báo động nhằm thu thập tập dữ liệu khác để đánh giá. Xem xét tới việc cập nhật tập dữ liệu mới và thời gian huấn luyện lại mô hình, mức độ thay đổi của các tham số. Có thể nghiên cứu và đánh giá để triển khai thực tế, vừa đảm bảo độ chính xác vừa đảm bảo thời gian thực.

TÀI LIỆU THAM KHẢO

[1]. Autoencoder là gì? Kiến Trúc và Cách tạo Autoencoder. Bizfly. (n.d.). https://bizflycloud.vn/tin-tuc/autoencoder-la-gi-20220526165157229.htm

[2]. Intro to autoencoders : TensorFlow Core. TensorFlow. (n.d.). https://www.tensorflow.org/tutorials/generative/autoencoder

[3]. ULB, M. L. G.-. (2018, March 23). Credit Card Fraud Detection. Kaggle. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[4]. Hưng, N. (2022, November 2). Deep learning là gì? Tìm hiểu về deep learning từA-Z. Vietnix. https://vietnix.vn/deep-learning-la-gi/

[5]. Cao Thị Nhâm. Đánh giá một số thuật toán sử dụng trong phát hiện gian lận thẻ tín dụng. https://tapchinganhang.gov.vn/danh-gia-mot-so-thuat-toan-hoc-may-khong-giam-sat-su-dung-trong-phat-hien-gian-lan-the-tin-dung.htm

[6]. Tổng quan về Neural Network. Itnavi. https://itnavi.com.vn/blog/neural-network-la-gi

[7]. Arpit Devansh, Zhou Yingbo, Ngo Hung, Govindaraju Venu. 2015. Why Regularized Auto-Encoders learn Sparse Representation?

[8]. Zeng Kun, Yu Jun, Wang Ruxin, Li Cuihua, Tao Dacheng. 2017. Coupled Deep Autoencoder for Single Image Super-Resolution. IEEE Transactions on Cybernetics.