

## PHÂN TÍCH VÀ DỰ ĐOÁN KHẢ NĂNG TỬ VONG DỰA TRÊN TẬP DỮ LIỆU MIMIC-III

Đoàn Phạm Thảo Như<sup>1</sup>, Nguyễn Linh Chi<sup>2</sup>

<sup>1</sup>Trường ĐH Kinh tế-Tài chính TP HCM, Việt Nam, [nhudpt20@uef.edu.vn](mailto:nhudpt20@uef.edu.vn)

<sup>2</sup>Trường ĐH Kinh tế-Tài chính TP HCM, Việt Nam, [chinl220@uef.edu.vn](mailto:chinl220@uef.edu.vn)

**Tóm tắt:** Tập dữ liệu MIMIC-III là một cơ sở dữ liệu y tế lớn chứa thông tin chi tiết về bệnh nhân tại các bệnh viện. Trong nghiên cứu này, chúng tôi tập trung vào vấn đề dự đoán khả năng tử vong của bệnh nhân bằng cách sử dụng dữ liệu từ MIMIC-III. Cụ thể, chúng tôi so sánh hiệu suất của các mô hình học máy phổ biến bao gồm Logistic Regression, Random Forest, Convolutional Neural Networks (CNN) và Vision Transformer (ViT). Kết quả thực nghiệm cho thấy tất cả các mô hình đều đạt độ chính xác dự đoán rất cao, trên 89%. Trong đó, mô hình ViT nổi trội với thời gian huấn luyện ngắn nhưng vẫn đảm bảo độ chính xác tốt. Nhìn chung, đây là một nhiệm vụ rất quan trọng, vì trong quá trình bệnh nhân nằm viện, việc dự đoán khả năng tử vong có thể góp phần sớm thay đổi phác đồ điều trị và tăng khả năng sống sót cho bệnh nhân. Nghiên cứu của chúng tôi cung cấp cơ sở để áp dụng các mô hình học máy tiên tiến trong nhiệm vụ dự đoán quan trọng này.

**Từ khóa:** MIMIC-III, MIMIC-Extract, Dự đoán tử vong, Học máy, Logistic Regression, Random Forest, CNN, ViT, TS-TCC.

### 1. Giới thiệu chung

#### 1.1. MIMIC-Extract

MIMIC-III (Medical Information Mart for Intensive Care III) là một cơ sở dữ liệu công khai lớn, chứa thông tin chi tiết về chăm sóc sức khỏe của hơn 40.000 bệnh nhân được chăm sóc tại các đơn vị chăm sóc đặc biệt tại Bệnh viện Beth Israel Deaconess Medical Center từ 2001 đến 2012. Dữ liệu này bao gồm thông tin về tình trạng lâm sàng, quá trình điều trị, kết quả xét nghiệm, ghi chú y tế, và nhiều thông tin khác. MIMIC-Extract (Wang và c.s., 2020) là một mã nguồn mở được sử dụng để trích xuất, tiền xử lý và biểu diễn các dữ liệu từ bộ dữ liệu MIMIC-III, bao gồm các thông tin về dân số học tại thời điểm nhập viện, những tín hiệu sống và các kết quả xét nghiệm thay đổi theo thời gian, kết quả sau thời gian nằm viện (sống hoặc tử vong). Phương pháp tiếp cận của MIMIC-Extract là tạo ra một nhóm dữ liệu chuỗi thời gian toàn diện phù hợp với những nhiệm vụ dự đoán khác nhau, đồng thời đưa ra lựa chọn về nhóm và biến một cách linh hoạt.

#### 1.2. Dự đoán khả năng tử vong ở người bệnh

Dự đoán tử vong cho bệnh nhân trong thời gian nằm viện rất quan trọng để đánh giá mức độ nặng của căn bệnh và xác định các liệu pháp can thiệp phù hợp, cũng như nghiên cứu về sáng kiến chăm sóc sức khỏe mới, đặc biệt trong các trường hợp cấp cứu và chăm sóc đặc biệt. Trong nghiên cứu này, nhóm tác giả tập trung vào việc phát triển và đánh giá các mô hình học máy và học sâu

để dự đoán tỷ lệ tử vong của bệnh nhân dựa trên tập dữ liệu MIMIC-III, một nguồn dữ liệu y tế lớn và phức tạp.

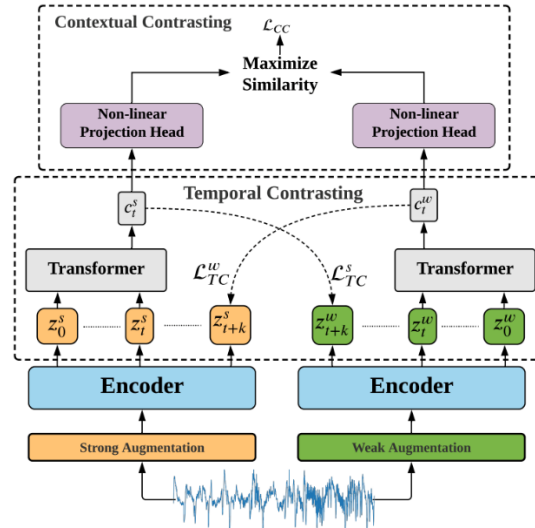
Nhận thấy sự tương đồng giữa dữ liệu về chuỗi thời gian của các bệnh nhân trong thời gian nhập viện với dữ liệu hình ảnh và chuỗi thời gian thông thường, nhóm tác giả đề xuất áp dụng các mô hình đã được tiến hành trên các loại dữ liệu tương ứng cho nhiệm vụ dự đoán này. Những mô hình này bao gồm Logistic Regression, Random Forest và Convolutional Neural Networks (CNN), Vision Transformer (Dosovitskiy và c.s., 2020) và một mô hình mới Time-Series Representation Learning via Temporal and Contextual Contrasting (Eldele và c.s., 2021).

Kết quả cho thấy hầu hết các mô hình đều cho độ chính xác cao với sự khác biệt rõ ràng về thời gian huấn luyện. Trong đó đáng nhắc đến là mô hình ViT với thời gian huấn luyện ngắn hơn nhiều so với các mô hình khác trong khi độ chính xác lại gần như tương đương. Ngoài ra, nhóm tác giả cũng nhận thấy mô hình TS-TCC không thích hợp với dữ liệu đầu vào của bài nghiên cứu. Tóm lại, áp dụng các mô hình xử lý ảnh cho dạng dữ liệu này mang lại các kết quả hết sức khả quan và đầy hứa hẹn.

## 2. Cơ sở lý thuyết

### 2.1. Mô hình Time-Series Representation Learning via Temporal and Contextual Contrasting (TS-TCC)

Mô hình này được giới thiệu trong bài báo cùng tên vào năm 2021 (Eldele và c.s., 2021). TS-TCC sử dụng dữ liệu không gán nhãn để học những biểu diễn hữu ích từ chuỗi thời gian.



Hình 1: Mô hình TS-TCC

#### Tăng cường dữ liệu chuỗi thời gian

Trong khi tăng cường yếu nhân đầu vào với một số ngẫu nhiên để thay đổi tỷ lệ, thì tăng cường mạnh thực hiện chia chuỗi thời gian thành những phần nhỏ hơn và xáo trộn chúng, sau đó thêm nhiều ngẫu nhiên để thu được kết quả mong muốn.

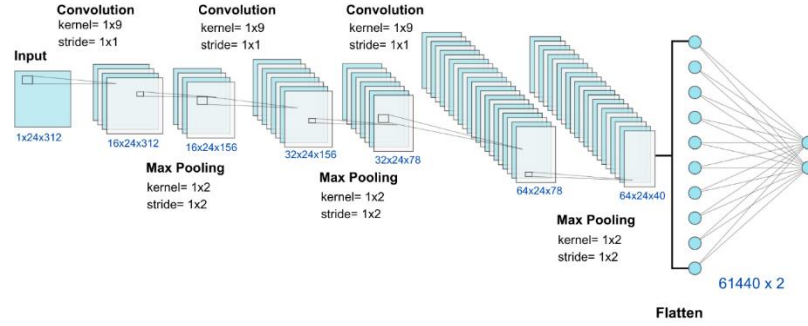
Mỗi bước thời gian (timestep)  $x$  được đưa vào Encoder để ánh xạ những đặc trưng của  $x$  sang biểu diễn chiều cao hơn  $z = f_{enc}(x)$ . Kết quả của bước này là một tập các biểu diễn cho tất cả các

bước thời gian như sau:  $z = [z_1, z_2, \dots, z_T]$ . Bởi vì có hai loại tăng cường khác nhau được tạo ra, nên nhóm tác giả đặt  $z^S$  đại diện cho tăng cường mạnh và  $z^W$  là tăng cường yếu.

### Tinh chỉnh Encoder

Encoder trong mô hình này là mạng CNN 1D đơn giản, để phù hợp với dữ liệu đầu vào là thông tin về chuỗi thời gian 24 giờ đầu tiên một bệnh nhân nhập viện, nhóm tác giả đã thay đổi mạng CNN này thành CNN 2D.

Trong đó, mạng CNN này sử dụng các lớp tích chập liên tiếp để trích xuất đặc trưng từ dữ liệu đầu vào, với mỗi lớp tích chập được theo sau bởi chuẩn hóa batch và hàm kích hoạt ReLU để đảm bảo tính phi tuyến và ổn định trong quá trình huấn luyện. Các lớp pooling được áp dụng để giảm kích thước không gian của các đặc trưng và tăng tính trừu tượng, trong khi lớp dropout được thêm vào sau mỗi khối tích chập nhằm giảm thiểu nguy cơ overfitting bằng cách ngẫu nhiên bỏ một số nút trong mạng.



Hình 2: Mô hình Convolutional Neural Network

Kích thước kernel được thiết lập là (1,9), với stride là (1,1) hoặc (1,2) cho phép mô hình phát hiện các đặc trưng từ dữ liệu đầu vào sao cho giữ nguyên các đặc trưng theo chiều dọc (chuỗi 24 giờ đầu tiên) và chỉ thay đổi về chiều ngang. Kết thúc chuỗi các lớp tích chập là một lớp fully connected, chịu trách nhiệm biến đổi đầu ra từ không gian đặc trưng sang lớp dự đoán cuối cùng. Mô hình được thiết kế để phân loại hai lớp (sống hoặc chết) và được cấu hình với các tham số giống với các tham số đã đặt cho mô hình TS-TCC.

### Mô-đun tương phản theo thời gian (the temporal contrasting module)

Lựa chọn ngẫu nhiên một bước thời gian  $t$ , sau đó một vector ngữ cảnh  $c_t$  được tạo ra cho tập hợp  $z_{\leq t}$  thông qua một mô hình tự hồi quy (autoregressive model)  $c_t = f_{ar}(z_{\leq t})$ . Vector ngữ cảnh  $c_t$  được sử dụng để dự đoán những bước thời gian trong tương lai từ  $z_{t+1}$  đến  $z_{t+K}$  với  $K$  được xác định trước. Để dự đoán bước thời gian tiếp theo, mô hình Log-bilinear được đề xuất để bảo toàn thông tin chung giữa  $c_t$  và  $z_{t+k}$  với  $(1 < k \leq K)$  với công thức như sau:

$$f_k(x_{t+k}, c_t) = \exp\left((W_k(c_t))^T z_{t+k}\right)$$

Với  $W_k$  ánh xạ  $c_t$  về cùng chiều với  $z$ .

Cụ thể hơn, tăng cường mạnh tạo ra  $c_t^s$  và kết quả của tăng cường yếu là  $c_t^w$ . Eldele và đồng sự đã đề xuất nhiệm vụ dự đoán chéo giữa hai góc nhìn thông qua sử dụng ngữ cảnh của tăng cường mạnh  $c_t^s$  để dự đoán  $K$  bước thời gian tiếp theo bắt đầu từ  $t$  cho tăng cường yếu  $z_{t+k}^w$  và ngược lại. Hàm mất mát tương phản (contrastive loss) tối đa hóa tích vô hướng giữa dự đoán và thực tế của cùng một mẫu, trong khi tối thiểu hóa tích vô hướng với các mẫu khác  $\mathcal{N}_{t,k}$ . Điều này có nghĩa là mô hình học cách đưa các biểu diễn giữa dự đoán và thực tế của cùng một mẫu lại gần nhau trong không gian. Như vậy, hai hàm mất mát  $\mathcal{L}_{TC}^s$  và  $\mathcal{L}_{TC}^w$  được tính như sau:

$$\mathcal{L}_{TC}^s = -\frac{1}{K} \sum_{k=1}^K \log \frac{\exp\left((W_k(c_t^s))^T z_{t+k}^w\right)}{\sum_{n \in \mathcal{N}_{t,k}} \exp\left((W_k(c_t^s))^T z_n^w\right)}$$

$$\mathcal{L}_{TC}^w = -\frac{1}{K} \sum_{k=1}^K \log \frac{\exp\left((W_k(c_t^w))^T z_{t+k}^s\right)}{\sum_{n \in \mathcal{N}_{t,k}} \exp\left((W_k(c_t^w))^T z_n^s\right)}$$

Mô hình tự hồi quy được đề xuất với mô-đun tương phản theo thời gian là Transformer, được xây dựng dựa trên chú ý đa đầu (multi-head attention) nối tiếp bởi một khối Perceptron nhiều lớp (Multilayer Perceptron). Khối Perceptron nhiều lớp này gồm hai lớp kết nối đầy đủ (fully-connected layers) với hàm kích hoạt là hàm phi tuyến ReLU và lớp bỏ học (dropout) ở giữa.

Hoạt động của Transformer bắt đầu bằng việc đưa những đặc trưng của các bước thời gian  $z_{\leq t}$  qua một lớp chiếu tuyến tính (Linear Projection) để ánh xạ các đặc trưng ban đầu sang một chiều ẩn phù hợp với mô hình. Đầu ra của lớp này tiếp tục được đưa qua Transformer để tạo ra vector ngữ cảnh.

#### **Mô-đun tương phản theo ngữ cảnh (the contextual contrasting module)**

Mô-đun tương phản theo ngữ cảnh được thêm vào để học thêm các đặc trưng riêng biệt. Vector ngữ cảnh  $c_t$  được gửi đến một khối chiếu phi tuyến (Non-linear Projection) để ánh xạ vector này vào không gian áp dụng tương phản theo ngữ cảnh. Với hai hình thức tăng cường dữ liệu khác nhau, nếu đầu vào có  $N$  mẫu khác nhau thì sẽ tạo ra  $2N$  ngữ cảnh phân biệt. Cụ thể hơn, một ngữ cảnh được tạo ra từ tăng cường mạnh  $c_t^i$  được xem là mẫu tích cực (positive sample) với ngữ cảnh tạo ra từ tăng cường yếu  $c_t^{i+}$  nếu chúng cùng được tạo ra từ một mẫu ban đầu, và cặp  $(c_t^i, c_t^{i+})$  được xem như một cặp tích cực (positive pair). Cùng với đó,  $(2N - 2)$  ngữ cảnh còn lại được tạo ra từ các đầu vào khác được định nghĩa là các cặp tiêu cực (negative pair). Vì vậy, hàm mất mát tương phản theo ngữ cảnh tối đa hóa sự tương đồng giữa các cặp tích cực cũng như tối thiểu hóa sự giống nhau giữa các cặp tiêu cực. Do vậy, kết quả cuối cùng càng học được sự riêng biệt giữa các mẫu.

Như vậy, hàm mất mát tổng cho mô hình học tự giám sát là sự kết hợp giữa hai mô-đun tương phản theo thời gian và tương phản theo ngữ cảnh:

$$\mathcal{L}_{unsup} = \lambda_1 \cdot (\mathcal{L}_{TC}^s + \mathcal{L}_{TC}^w) + \lambda_2 \cdot \mathcal{L}_{CC}$$

Trong đó,  $\lambda_1$  và  $\lambda_2$  là những hằng số cố định biểu thị mức độ ảnh hưởng của hai hàm mất mát này đối với hàm tổng.

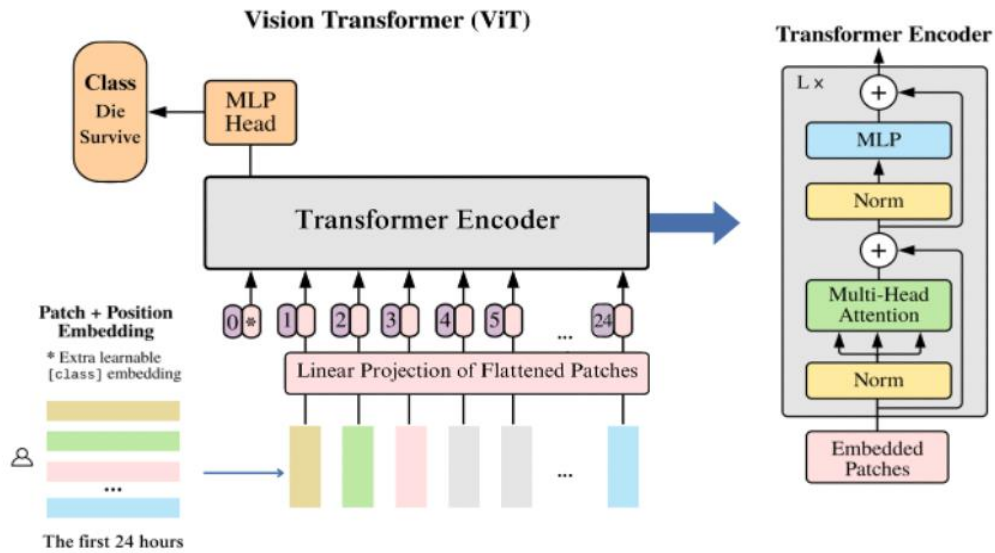
#### **Tinh chỉnh mô hình**

Sau khi học những biểu diễn của chuỗi thời gian nhiều hơn, mô hình sẽ được gọi lại với chế độ “fine-tune” để đánh giá trên những dữ liệu có gán nhãn. Trong chế độ này mô hình không bao gồm hai mô-đun tương phản nêu trên mà dữ liệu chỉ được đưa qua Encoder (CNN) để dự đoán lớp (classes).

## 2.2. Vision Transformer (ViT)

Mô hình ViT được giới thiệu trong bài báo “An image is worth 16x16 words: Transformers for image recognition at scale” bởi Alexey Dosovitskiy và đồng sự ứng dụng mô hình Transformer để phân loại ảnh.

### Mô hình ViT trong nhiệm vụ dự đoán khả năng tử vong



Hình 3: Tổng quan mô hình: Chuỗi thời gian được chia thành các phần nhỏ với kích thước cố định, nhúng tuyến tính từng phần, thêm các nhúng liên quan đến vị trí và đưa chuỗi vector kết quả vào Transformer Encoder tiêu chuẩn. Để thực hiện phân loại, nhóm tác giả

Nhóm tác giả định dạng lại dữ liệu của bệnh nhân  $x \in \mathbb{R}^{24 \times W \times C}$  thành một chuỗi các phần nhỏ (patch) được làm phẳng  $x_p \in \mathbb{R}^{24 \times (1 \cdot W \cdot C)}$ , trong đó  $(24, W)$  là kích thước của chuỗi ban đầu (mỗi giờ ghi lại các số liệu về sự sống của bệnh nhân một lần, ghi liên tiếp 24 giờ),  $C$  là số lượng kênh đầu vào,  $(1, W, C)$  là kích thước của mỗi patch và trong trường hợp này tổng số patch là 24, cũng đóng vai trò là độ dài chuỗi đầu vào của khối Encoder. Sau đó, các patch được làm phẳng và ánh xạ tới chiều  $D$  bằng phép chiếu tuyến tính có thể đào tạo (Eq. 1). Đầu ra của phép chiếu này được xem là nhúng các phần nhỏ (patch embeddings).

Giống với token [class] của mô hình BERT, nhóm tác giả thêm một phần nhúng (embedding) có thể học vào chuỗi các phần nhỏ ( $z_0^0 = x_{class}$ ), trạng thái của nó ở đầu ra của Transformer Encoder  $z_L^0$  đóng vai trò là nhãn biểu thị sự sống của bệnh nhân  $y$  (Eq. 4). Trong cả quá trình tiền

huấn luyện và tinh chỉnh, một đầu phân loại (MLP head) được gắn vào  $z_L^0$ . Các nhúng liên quan đến vị trí được thêm vào để giữ lại thông tin về vị trí.

Khối Transformer Encoder (Vaswani và đồng sự, 2017) bao gồm những lớp xen kẽ của các đầu tự chú ý (multiheaded self-attention – MSA) và khác khối MLP (Eq. 2,3). Chuẩn hóa theo lớp (Layernorm – LN) được áp dụng trước mỗi khối, và kết nối phần dư sau mỗi khối (Wang và đồng sự, 2019; Baevski và Auli, 2019).

Khối MLP gồm hai lớp với hàm phi tuyến GELU.

$$z_0 = [x_{class}; x_p^1 \mathbf{E}; x_p^2 \mathbf{E}; \dots; x_p^N \mathbf{E}] + \mathbf{E}_{pos}, \quad \mathbf{E} \in \mathbb{R}^{(W.C) \times D}, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \quad (1)$$

$$z'_\ell = \text{MSA}(\text{LN}(z_{\ell-1})) + z_{\ell-1}, \quad \ell = 1 \dots L \quad (2)$$

$$z_\ell = \text{MLP}(\text{LN}(z'_\ell)) + z'_\ell, \quad \ell = 1 \dots L \quad (3)$$

$$y = \text{LN}(z_L^0) \quad (4)$$

### 2.3. Tìm kiếm ngẫu nhiên (Random Search)

Tìm kiếm ngẫu nhiên (Random Search) là một phương pháp tối ưu hóa siêu tham số được sử dụng trong học máy. Phương pháp này thay thế cho việc liệt kê toàn bộ các tổ hợp siêu tham số bằng cách chọn chúng một cách ngẫu nhiên. Quá trình này bắt đầu bằng việc xác định không gian siêu tham số và tiếp tục với việc lấy mẫu ngẫu nhiên các tổ hợp siêu tham số, từ đó huấn luyện và đánh giá hiệu suất của mô hình dựa trên từng tổ hợp siêu tham số đó. Cuối cùng, tổ hợp nào cho kết quả tốt nhất sẽ được chọn lựa.

## 3. Bộ dữ liệu và tiền xử lý dữ liệu

### 3.1. Bộ dữ liệu

#### Tổng quan về bộ dữ liệu

MIMIC-III là một bộ dữ liệu liên quan đến sức khỏe của hơn 40,000 bệnh nhân đã nằm trong các đơn vị chăm sóc cấp cứu Trung tâm Y tế Beth Israel Deaconess trong giai đoạn 2001 – 2012, các dữ liệu đã được ẩn danh hóa. Cơ sở dữ liệu này bao gồm thông tin như thông tin nhân khẩu học, dấu hiệu sống được đo lường từ giường bệnh (khoảng 1 điểm dữ liệu/giờ), kết quả xét nghiệm máu, các thủ tục, thuốc, ghi chú của người chăm sóc, báo cáo hình ảnh và tỷ lệ tử vong (bao gồm cả sau khi ra viện).

#### Mô tả bộ dữ liệu

MIMIC là cơ sở dữ liệu quan hệ chứa các bảng dữ liệu liên quan đến bệnh nhân nằm trong phòng chăm sóc đặc biệt tại Trung tâm Y tế Beth Israel Deaconess bao gồm 26 bảng. Mỗi bảng là một cấu trúc lưu trữ dữ liệu tương tự như bảng tính: mỗi cột chứa thông tin nhất quán (ví dụ: mã định danh bệnh nhân) và mỗi hàng chứa phần thể hiện thông tin đó (ví dụ: một hàng chứa số nguyên 340 trong cột mã định dạng bệnh nhân, ngụ ý rằng mã định danh bệnh nhân của hàng là 340).

### 3.2. Tiền xử lý dữ liệu

#### Pipeline MIMIC-Extract

Pipeline “MIMIC-Extract” bao gồm các bước sau:

- Lựa chọn nhóm bệnh nhân từ cơ sở dữ liệu MIMIC-III theo các tiêu chí cụ thể để đảm bảo tính đại diện và tính ứng dụng của dữ liệu trong quá trình phân tích và ứng dụng trong thực tế lâm sàng.
- Trích xuất các biến số dân số tĩnh, thông tin về việc nằm viện tại chăm sóc đặc biệt ICU, các chỉ số và xét nghiệm, cũng như chuỗi thời gian can thiệp theo giờ cho các bệnh nhân.
- Nhóm các đặc trưng nguyên thủy thành các nhóm tóm tắt lâm sàng dựa trên kiến thức lâm sàng để giảm chiều dữ liệu và tăng tính ổn định của các đặc trưng trong quá trình phân tích và ứng dụng.
- Xử lý dữ liệu bao gồm chuyển đổi đơn vị, xử lý các giá trị ngoại lệ và tổng hợp các đặc trưng tương đồng để giảm thiểu sự mất mát và cải thiện tính ổn định của dữ liệu.
- Tạo ra các bảng dữ liệu đầu ra, bao gồm bảng thông tin bệnh nhân (patients), bảng các chỉ số và xét nghiệm (vitals\_labs), bảng các chỉ số và xét nghiệm được tổng hợp theo giờ (vitals\_labs\_mean) và bảng các can thiệp được thực hiện theo giờ (interventions).

### **Tiền xử lý để áp dụng vào mô hình dự đoán**

Để áp dụng được dữ liệu MIMIC-Extract vào mô hình dự đoán, nhóm tác giả đã thực hiện tiền xử lý dữ liệu như sau:

- Điều chỉnh tên cột: Nếu có nhiều hơn hai cấp độ của tên cột, họ thực hiện loại bỏ một số cấp độ để giảm độ phức tạp của tên cột.
- Xử lý giá trị trung bình và số lượng: Họ điền giá trị trung bình cho các giá trị NaN bằng cách sử dụng phương pháp forward fill (ffill), và sau đó, nếu cần, điền giá trị NaN bằng giá trị trung bình của từng nhóm theo các cột xác định bởi ID\_COLS. Đồng thời, họ chuyển đổi cột số lượng thành một cột binary.
- Tính thời gian kể từ khi có dữ liệu: Tạo một cột mới ('time\_since\_measured') để biểu thị thời gian kể từ khi có dữ liệu. Điều này bao gồm tính số giờ kể từ khi có dữ liệu đến thời điểm hiện tại, cũng như việc điền giá trị NaN trong cột mới này.
- Kết quả cuối cùng: Sắp xếp cột của DataFrame kết quả và trả về nó để sử dụng trong các phân tích hoặc mô hình hóa sau này.

## **4. Thực nghiệm và kết quả**

### **4.1. Thực nghiệm**

#### **4.1.1. Mô hình Hồi quy Logistic (Logistic Regression)**

Sử dụng phân phối từ điển (DictDist) để định nghĩa không gian siêu tham số. Các siêu tham số của mô hình này gồm có C là độ mạnh của Regularization, Penalty là loại Regularization, Solver biểu thị thuật toán tối ưu hóa được sử dụng, Max\_iter là số lần lặp tối đa.

- C được lấy mẫu trong khoảng từ  $1e-3$  đến  $1e3$  với 10000 điểm.
- Penalty: l1 và l2
- Solver: liblinear và lbfgs
- max\_iter: 100 và 500.
- Nếu siêu tham số solver là lbfgs, thì penalty là l2.

#### **4.1.2. Mô hình rừng cây (Random Forest)**

Tương tự, các siêu tham số của Random Forest được xác định thông qua phương pháp tìm kiếm ngẫu nhiên, bao gồm n\_estimators là số lượng cây quyết định trong rừng được lấy mẫu ngẫu nhiên

từ một phân phối đều giữa 50 và 500, `max_depth` từ 2 đến 10 để kiểm soát độ sâu của mỗi cây, `min_samples_split` từ 2 đến 75 để xác định số lượng mẫu tối thiểu cần thiết cho việc chia mẫu cho một nút, `min_samples_leaf` từ 1 đến 50 để xác định số lượng mẫu tối thiểu được yêu cầu để tạo thành một lá.

### 4.1.3. Mô hình Vision Transformer (ViT)

- Kích thước ẩn của vector: 128
- Số lớp Encoder: 6
- 8 đầu (head) trong mô hình tự chú ý đa đầu
- Kích thước của MLP: 2048
- Hàm mất mát: Cross Entropy Loss
- Thuật toán tối ưu: Adam
- Learning rate:  $3 \times 10^{-5}$
- Epochs: 20
- Kích thước batch: 64

### 4.1.4. Mô hình Time-Series Representation Learning via Temporal and Contextual Contrasting (TS-TCC)

- Kích thước kernel: 9
- Số bước (stride): 1
- Kênh ở đầu ra cuối cùng: 64
- Dropout: 0.35
- Epochs: 15
- $\beta_1 = 0,9$  và  $\beta_2 = 0,99$  (Kingma và Ba, 2015) và tỉ lệ học là  $3e-4$ .
- Kích thước chiều ẩn: 100
- Số bước thời gian: 10

## 4.2. Kết quả

### Logistic Regression và Random Forest

Sau khi thực hiện tìm kiếm ngẫu nhiên 15 lần thu được các siêu tham số tốt nhất cho mô hình Random Forest là **`n_estimators: 121, max_depth: 8, min_samples_split: 49, min_samples_leaf: 42`** với độ chính xác dự đoán của mô hình trên tập kiểm tra là **0,9033** và mất mát là **0,2489**. Tương tự, tham số phù hợp nhất cho mô hình Logistic Regression là **`C: 0,0323, penalty: l1, solver: liblinear, max_iter: 100`** với kết quả là **0,9058** và mất mát là **0,2536**, chỉ hơi cao hơn một chút so với mô hình Random Forest. Cả hai mô hình đều cho thấy hiệu suất cao, nhưng Logistic Regression nhỉnh hơn một chút về độ chính xác dự đoán và có mất mát tương đương.

### TS-TCC

Khi sử dụng các thông số hiện tại của mô hình để chạy trên tập kiểm tra mà không thực hiện fine-tune thì thu được mất mát là **96,784** và độ chính xác dự đoán là **0,877**. Điều này chứng tỏ có sự không cân bằng lớn giữa hai lớp trong tập dữ liệu huấn luyện.

### Fine-tune

Dựa vào kết quả trên, có thể thấy mô hình này đã thể hiện một quá trình cải thiện liên tục và đáng kể trong suốt quá trình huấn luyện với mất mát khi kiểm thử kết thúc ở **0.3112** và độ chính xác dự đoán là **0.9073**. Sự tăng cho thấy mô hình tổng quát hóa tốt và có khả năng duy trì hiệu suất mạnh mẽ. Mất mát giảm mạnh ở các epoch đầu chứng tỏ mô hình TS-TCC không giúp học dữ liệu hiệu quả, mô hình phải điều chỉnh lại trọng số để học biểu diễn của dữ liệu đầu vào.

### CNN

Có thể thấy một sự tương đồng với kết quả của quá trình fine-tune nhưng cũng có một số điểm khác biệt nhỏ. Train Accuracy duy trì ở mức ổn định, dao động quanh **0.891**, điều này cho thấy mô



hình đã tìm ra được một dạng biểu diễn tốt cho dữ liệu. Điều này càng được khẳng định khi kết quả trên tập kiểm tra ghi nhận mất mát là **0,3313** và độ chính xác là **0,8975**.

### **ViT**

Mô hình này có hiệu suất khá ổn định với độ chính xác dự đoán ổn định ở mức khoảng **89%** cùng với mất mát (loss) giảm nhẹ và liên tục qua từng epoch. Điều này cho thấy mô hình đã học được một cách hiệu quả từ dữ liệu, không chỉ trong quá trình huấn luyện mà còn trên cả tập kiểm tra với độ chính xác dự đoán là **0,8911** và mất mát **0,3438** khá ổn. Mặc dù có sự ổn định này, nhưng không có sự cải thiện đáng kể về độ chính xác dự đoán qua các epochs, điều này có thể chỉ ra rằng mô hình có thể đã đạt tới giới hạn của khả năng học từ tập dữ liệu hiện tại.

<b>Mô hình</b>	<b>Thời gian huấn luyện trung bình</b>	<b>Độ chính xác</b>
Logistic Regression	60 phút	0,9058
Random Forest		0,9033
TS-TCC	13 phút 22 giây/epoch	0,877
Fine-tune	3 phút 19 giây/epoch	0,8975
CNN	3 phút 17 giây/epoch	0,8975
ViT	1 phút	0,8911

*Bảng 1: Bảng tổng hợp kết quả huấn luyện của các mô hình*

Nhìn chung, các mô hình cho độ chính xác dự đoán khá cao nhưng có sự khác nhau về thời gian huấn luyện.

## **5. Kết luận**

MIMIC-Extract cung cấp một cách tiếp cận tiêu chuẩn và linh hoạt để xử lý dữ liệu EHR phức tạp, giúp các nhà nghiên cứu học máy trong lĩnh vực y tế tiếp cận dữ liệu một cách dễ dàng hơn. Đồng thời, sử dụng các đặc trưng trích xuất được thông qua Mimic-Extract để dự đoán khả năng tử vong của người bệnh cũng mang lại một độ chính xác rất cao.

Bên cạnh đó, trong nhiệm vụ dự đoán đã đặt ra, mô hình TS-TCC không phù hợp để sử dụng, tốn nhiều thời gian để huấn luyện nhưng độ chính xác thấp. Như vậy, việc học tự giám sát các biểu diễn sâu hơn của chuỗi thời gian không có hiệu quả trong trường hợp này. Và cũng có thể thấy, các mô hình Logistic Regression, Random Forest, CNN và ViT biểu hiện rất tốt khi thực hiện dự đoán khả năng tử vong của người bệnh, độ chính xác của các mô hình đều rất cao, trên 89%. Trong đó đáng chú ý nhất là ViT, thời gian huấn luyện 1 epoch rất ngắn nhưng kết quả lại rất tốt, nhận thấy tiềm năng cải tiến bằng cách thay đổi thông số mô hình phức tạp hơn để quan độ chính xác của dự đoán có cải tiến không vì bài nghiên cứu này chỉ sử dụng cấu hình rất đơn giản. Cả Logistic Regression và Random Forest đều dự đoán với độ chính xác cao, nhưng có một nhược điểm là việc thực hiện tìm kiếm ngẫu nhiên để xác định tham số cho mô hình tốn nhiều thời gian. Tổng quan, có thể kết luận rằng đây là tác vụ dự đoán tương đối dễ dàng, với hiệu suất đạt mức bão hòa chỉ

với một lượng dữ liệu tối thiểu và ngay cả khi sử dụng các mô hình tương đối đơn giản, như mô hình rừng cây.

**Tài liệu tham khảo**

- [1] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (22/10/2023). *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. International Conference on Learning Representations. <https://openreview.net/forum?id=YicbFdNTTy>
- [2] Eldele, E., Ragab, M., Chen, Z., Wu, M., Kwoh, C. K., Li, X., & Guan, C. (2021). Time-Series Representation Learning via Temporal and Contextual Contrasting. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 2352–2359. <https://doi.org/10.24963/ijcai.2021/324>
- [3] Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980). arXiv. <http://arxiv.org/abs/1412.6980>
- [4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, 30. [https://proceedings.neurips.cc/paper\\_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html)
- [5] Wang, S., McDermott, M. B. A., Chauhan, G., Hughes, M. C., Naumann, T., & Ghassemi, M. (2020). MIMIC-Extract: A Data Extraction, Preprocessing, and Representation Pipeline for MIMIC-III. *Proceedings of the ACM Conference on Health, Inference, and Learning*, 222–235. <https://doi.org/10.1145/3368555.3384469>
- [6] MIT Laboratory for Computational Physiology (2023). *MIMIC documentation*. <https://mimic.mit.edu/docs/>