

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**NIÊN LUẬN CƠ SỞ
NGÀNH CÔNG NGHỆ THÔNG TIN**

Đề tài

**PHÁT HIỆN BẤT THƯỜNG CỦA NHỊP TIM
BẰNG DỮ LIỆU ĐIỆN TÂM ĐỒ (ECG)**

Sinh viên: ĐẶNG THÀNH TRUNG

Mã số: B1910322

Khóa: K45

Cần Thơ, 05/2022

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
BỘ MÔN CÔNG NGHỆ THÔNG TIN**

**NIÊN LUẬN CƠ SỞ
NGÀNH CÔNG NGHỆ THÔNG TIN**

Đề tài

**PHÁT HIỆN BẤT THƯỜNG CỦA NHỊP TIM
BẰNG DỮ LIỆU ĐIỆN TÂM ĐỒ (ECG)**

**Người hướng dẫn
TS. PHẠM THẾ PHI**

**Sinh viên thực hiện
ĐẶNG THÀNH TRUNG
Mã số: B1910322
Khóa: K45**

Cần Thơ, 05/2022

MỤC LỤC

MỤC LỤC.....	1
DANH MỤC HÌNH	3
DANH MỤC BẢNG.....	4
GIẢI THÍCH TỪ NGỮ.....	5
PHẦN 1. GIỚI THIỆU.....	6
I. ĐẶT VẤN ĐỀ.....	6
II. MỤC TIÊU ĐỀ TÀI	6
III. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU	6
1. Đối tượng nghiên cứu.....	6
2. Phạm vi nghiên cứu.....	6
IV. PHƯƠNG PHÁP NGHIÊN CỨU	7
V. NỘI DUNG NGHIÊN CỨU.....	7
VI. BỐ CỤC QUYỀN NIÊN LUẬN.....	7
1. Phần giới thiệu.....	7
2. Phần nội dung.....	7
3. Phần kết luận	7
PHẦN 2. NỘI DUNG	8
CHƯƠNG 1. MÔ TẢ BÀI TOÁN	8
I. MÔ TẢ RỐI LOẠN NHỊP TIM.....	8
II. ĐẶC TẢ YÊU CẦU.....	8
III. MÔ TẢ TẬP DỮ LIỆU	9
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	10
I. GIỚI THIỆU VỀ HỌC SÂU	10
II. DỮ LIỆU CHUỖI THỜI GIAN.....	11
1. Chuỗi thời gian.....	11
2. Phát hiện bất thường trên dữ liệu chuỗi thời gian.....	11

III. LSTM AUTOENCODER	12
1. Long Short Term Memory	12
2. Autoencoder	15
3. LSTM Autoencoder	16
CHƯƠNG 3. KẾT QUẢ THỰC HIỆN.....	17
I. TIỀN XỬ LÝ DỮ LIỆU	17
II. XÂY DỰNG MÔ HÌNH	18
III. ĐÀO TẠO MÔ HÌNH.....	20
CHƯƠNG 4. ĐÁNH GIÁ KIỂM THỬ.....	21
I. CHỌN NGUỒNG ĐÁNH GIÁ	21
II. ĐÁNH GIÁ MÔ HÌNH	22
PHẦN 3. KẾT LUẬN	24
TÀI LIỆU THAM KHẢO.....	25

DANH MỤC HÌNH

Hình 1: Thống kê số lượng các lớp.....	9
Hình 2: Mối liên hệ giữa trí tuệ nhân tạo, máy học và học sâu	10
Hình 3: Minh họa dữ liệu chuỗi thời gian.....	11
Hình 4: Recurrent Neural Network.....	12
Hình 5: Module của một RNN.....	13
Hình 6: Cấu trúc của Long Short Term Memory.....	13
Hình 7: Trạng thái tế bào	14
Hình 8: Cổng gate	14
Hình 9: Quá trình hoạt động một module LSTM	14
Hình 10: Mô hình trực quan của Autoencoder	15
Hình 11: Hoạt động của LSTM Autoencoder tái tạo ECG.....	16
Hình 12: Chuỗi thời gian trước và sau khi tái tạo.....	16
Hình 13: Tạo dữ liệu lớp bình thường	17
Hình 14: Tạo dữ liệu lớp không bình thường	17
Hình 15: Phân chia tập dữ liệu.....	17
Hình 16: Chuyển đổi chuỗi thời gian.....	18
Hình 17: Class Encoder.....	18
Hình 18: Class Decoder	19
Hình 19: Sử dụng encoder và decoder	19
Hình 20: Quá trình đào tạo mô hình	20
Hình 21: Tổn thất trong quá trình đào tạo.....	20
Hình 22: Losses tập dữ liệu bình thường	21
Hình 23: Losses tập dữ liệu không bình thường	22
Hình 24: Kết quả dự đoán	23

DANH MỤC BẢNG

Bảng 1: Mô tả tập dữ liệu ECG5000.....	9
Bảng 2: Kết quả dự đoán.....	22

GIẢI THÍCH TỪ NGỮ

STT	Từ viết tắt	Tiếng Anh	Tiếng Việt
1	AI	Artificial Intelligence	Trí tuệ nhân tạo
2	RNN	Recurrent Neural Network	Mạng nơ ron hồi quy
3	LSTM	Long Short Term Memory networks	Mạng bộ nhớ dài ngắn
4	ECG	Electrocardiogram	Điện tâm đồ
5	MRI	Magnetic Resonance Imaging	Cộng hưởng từ

PHẦN 1. GIỚI THIỆU

I. ĐẶT VẤN ĐỀ

Rối loạn nhịp tim xuất hiện ngày càng phổ biến và có xu hướng trẻ hóa, nếu không phát hiện và điều trị kịp thời người bệnh có thể gặp nguy hiểm đến tính mạng.

Ngày nay, các bác sĩ chẩn đoán rối loạn nhịp tim bằng nhiều phương pháp như dựa vào dấu hiệu, các triệu chứng lâm sàng (khó thở, đau ngực,...) và các thông số cận lâm sàng như: điện tâm đồ (ECG), siêu âm tim, hình ảnh cộng hưởng từ (MRI),... Trong đó điện tâm đồ là một trong những phương pháp được sử dụng đầu tiên trong chẩn đoán.

Mặc dù vậy, việc chẩn đoán đòi hỏi tốn nhiều thời gian và chuyên môn. Các bác sĩ phải dựa vào đồ thị của dòng điện tim được ghi lại trên giấy để phân tích, từ đó tìm ra những bất thường liên quan đến bệnh lý như rối loạn nhịp tim, nhồi máu cơ tim, suy tim cấp, tràn dịch màng ngoài tim,... Tuy nhiên, ECG có rất nhiều hình dạng khác nhau do đó đòi hỏi các bác sĩ phải có một lượng lớn kiến thức. Tốc độ phân tích cũng là một vấn đề lớn vì một số bệnh lý cấp tính cần phải can thiệp càng sớm càng tốt để đảm bảo tính mạng của bệnh nhân.

Từ các vấn đề nêu trên, ứng dụng trí tuệ nhân tạo (AI) vào phân tích ECG là một giải pháp đúng đắn và cần thiết trong thời đại ngày nay

II. MỤC TIÊU ĐỀ TÀI

Mục tiêu chính của đề tài là **“Phân loại rối loạn nhịp tim bằng dữ liệu điện tâm đồ (ECG)”**, có sử dụng kiến trúc mạng LSTM Autoencoder để phân loại các rối loạn.

Các mục tiêu cụ thể:

- Xây dựng mô hình phân loại rối loạn xung nhịp tim dựa trên dữ liệu điện tâm đồ ECG.
- Nghiên cứu kiến trúc mạng LSTM.
- Nghiên cứu mô hình Autoencoder.
- Nghiên cứu mô hình LSTM Autoencoder.

III. ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU

1. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính của đề tài là mô hình LSTM Autoencoder hỗ trợ phân loại rối loạn nhịp tim và giải thuật Autoencoder trong xử lý dữ liệu chuỗi thời gian.

2. Phạm vi nghiên cứu

Đề tài tập trung nghiên cứu mô hình học sâu để chẩn đoán rối loạn nhịp tim dựa trên dữ liệu điện tâm đồ. Mô hình cần xác định bệnh nhân có rối loạn nhịp tim hay không.

IV. PHƯƠNG PHÁP NGHIÊN CỨU

Phương pháp của đề tài gồm hai nội dung chính:

- Nghiên cứu tài liệu:

- + Nghiên cứu tổng quan về dữ liệu chuỗi thời gian.
- + Nghiên cứu tập dữ liệu ECG, các tài liệu về rối loạn nhịp tim.
- + Nghiên cứu mô hình LSTM, Autoencoder qua các bài báo khoa học.

- Thực nghiệm:

- + Xây dựng mô hình phát hiện rối loạn nhịp tim.
- + Kiểm thử hệ thống.

V. NỘI DUNG NGHIÊN CỨU

- Nghiên cứu mô hình LSTM Autoencoder.
- Nghiên cứu các bài báo khoa học có liên quan đến việc sử dụng mô hình LSTM Autoencoder để phát hiện bất thường với dữ liệu ECG.

VI. BỐ CỤC QUYỂN NIÊN LUẬN

1. Phần giới thiệu

Bao gồm các nội dung:

- Đặt vấn đề
- Mục tiêu đề tài
- Đối tượng và phạm vi nghiên cứu
- Phương pháp nghiên cứu
- Nội dung nghiên cứu
- Bố cục quyển niên luận

2. Phần nội dung

Bao gồm các nội dung:

- Chương 1: Mô tả bài toán
- Chương 2: Cơ sở lý thuyết
- Chương 3: Kết quả thực hiện
- Chương 4: Đánh giá kiểm thử

3. Phần kết luận

Bao gồm các nội dung:

- Kết luận
- Hướng phát triển

PHẦN 2. NỘI DUNG

CHƯƠNG 1. MÔ TẢ BÀI TOÁN

I. MÔ TẢ RỐI LOẠN NHỊP TIM

Rối loạn nhịp tim là sự bất thường trong nhịp đập của tim, gây ra tình trạng tim đập quá nhanh (trên 100 nhịp/phút) hoặc quá chậm (dưới 60 nhịp/phút) hoặc không đều. Tùy vào loại rối loạn nhịp tim khác nhau có các triệu chứng khác nhau: tim đập nhanh, đau ngực, thở gấp, ngất xỉu,... Một số rối loạn nhịp tim phổ biến:

Ngoại tâm thu thất (PVC): đây là trường hợp rất phổ biến, nhịp tim bị lỗi nhịp do tâm thất co lại quá sớm.

Ngoại tâm thu nhĩ (PAC hoặc APC): đây là những nhịp đập thêm trong các buồng trên của tim.

Nhịp nhanh trên thất (SVT) hoặc nhịp nhanh kịch phát trên thất (PSVT): trong đó "kịch phát" có nghĩa là theo từng giai đoạn, đây là nhịp tim nhanh 150 - 250 nhịp mỗi phút, do xung điện gây ra trong tâm nhĩ.

Rung nhĩ: đây là nhịp tim nhanh, không đều, nơi các cơ hoặc sợi trong tim bạn co bóp.

Rung thất: đây là mức khẩn cấp trong y tế do các buồng tim ở phần thấp không thể co lại để bơm máu.

Block tim: là một vấn đề khi tốc độ của các xung điện được gửi qua tim, và trong một số trường hợp, một khối các xung điện hoàn toàn, gây ra nhịp tim không đều.

Nguyên nhân gây ra rối loạn nhịp tim có thể do căng thẳng, nhiễm trùng, sốt, dùng thuốc,... Ngoài ra, rối loạn còn có thể xảy ra do các vấn đề khác như: huyết áp cao, rối loạn tuyến giáp, bệnh phổi,...

II. ĐẶC TẢ YÊU CẦU

Trong những năm gần đây, sự phát triển mạnh mẽ trong việc ứng dụng công nghệ vào phát hiện nhịp tim bất thường đã góp phần cải thiện kết quả chẩn đoán bệnh tim. Đặc biệt là chẩn đoán dựa vào dữ liệu điện tâm đồ. Điều này đặt ra vấn đề cần ứng dụng các mô hình học sâu để phát hiện chính xác và nhanh chóng những bất thường của xung nhịp tim, giúp bệnh nhân giảm chi phí và thời gian điều trị.

Từ vấn đề như trên, đề tài tập trung vào thực hiện:

- Xây dựng mô hình LSTM Autoencoder để phát hiện bất thường trên dữ liệu điện tâm đồ.
- Thử nghiệm với các ngưỡng khác nhau để đánh giá hiệu suất mô hình.

III. MÔ TẢ TẬP DỮ LIỆU

Đề tài đã sử dụng tập dữ liệu ECG5000 để huấn luyện và kiểm tra cho mô hình phân loại rối loạn nhịp tim. Bộ dữ liệu gốc của ECG5000 là một ECG (điện tâm đồ) dài 20 giờ được tải xuống từ Physionet, tên là BIDMC Congestive Heart Failure Database. Bộ dữ liệu được xuất bản trong "Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23)". Bảng 1 mô tả tóm tắt tập dữ liệu.

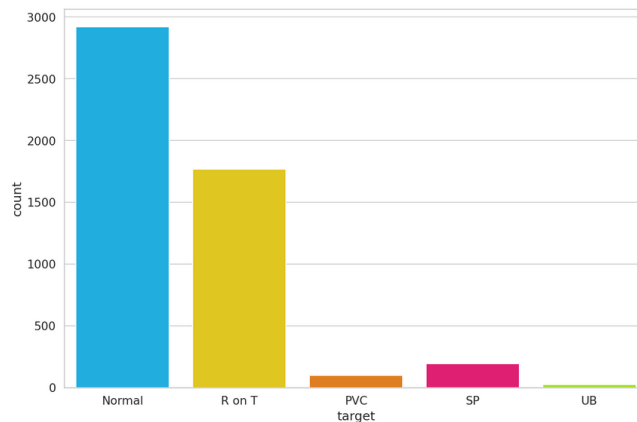
Bảng 1: Mô tả tập dữ liệu ECG5000

Train Size	Test Size	Length	Number of Classes	Type
500	4500	140	5	ECG

Dữ liệu được xử lý theo hai bước: (1) trích xuất từng nhịp tim, (2) làm cho mỗi nhịp tim có độ dài bằng nhau bằng cách sử dụng phép nội suy. Ban đầu tập dữ liệu được sử dụng trong bài báo "A general framework for never-ending learning from time series streams", DAMI 29(6). Sau đó, 5000 nhịp tim được chọn ngẫu nhiên.

Bộ dữ liệu dùng để xây dựng mô hình gồm 5 lớp:

- Bình thường (Normal)
- Co thắt tâm thất sớm R-on-T (R-on-T PVC)
- Co thắt tâm thất sớm (PVC)
- Nhịp đập ngoài tử cung hoặc sớm trên thất (SP hoặc EB)
- Chưa phân loại (UB)

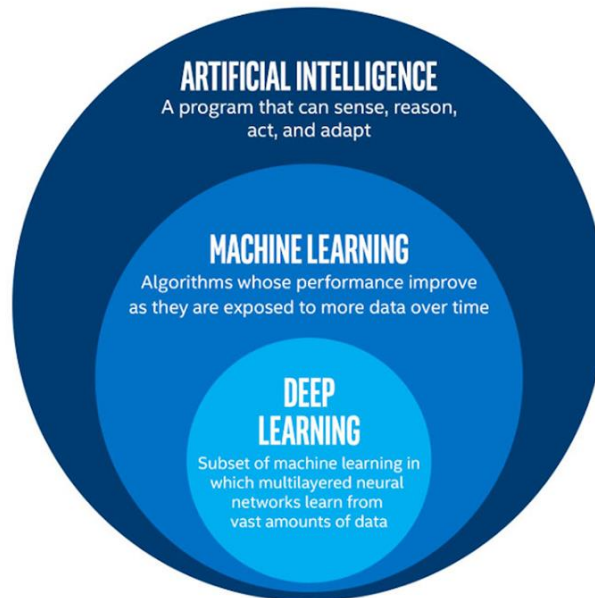


Hình 1: Thống kê số lượng các lớp

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

I. GIỚI THIỆU VỀ HỌC SÂU

Học sâu là một lĩnh vực con của trí tuệ nhân tạo (AI) mô phỏng hoạt động của bộ não con người trong việc xử lý dữ liệu và tạo ra các mẫu để sử dụng cho việc ra quyết định. Học sâu là tập con của máy học, có khả năng học tập mà không cần giám sát từ dữ liệu không có cấu trúc hoặc không được gán nhãn, còn được gọi là mạng thần kinh sâu. Mối tương quan giữa học sâu, máy học và trí tuệ nhân tạo được thể hiện trong Hình 2.



Hình 2: Mối liên hệ giữa trí tuệ nhân tạo, máy học và học sâu

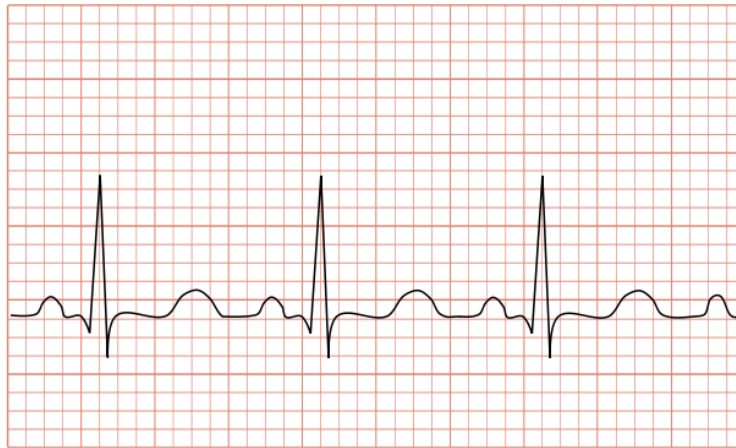
Với sự phát triển mạnh mẽ của trí tuệ nhân tạo trong kỷ nguyên kỹ thuật số, học sâu đã và đang được áp dụng mạnh mẽ vào lĩnh vực y học. Những đổi mới về học sâu đã đưa các công nghệ y tế đến những thành tựu y học đầy hứa hẹn, chẳng hạn như mô phỏng một ca lâm sàng trong phân tích, phát triển thuốc kháng sinh, chẩn đoán hình ảnh,... Hơn nữa, những tiến bộ trong chẩn đoán hình ảnh tạo nhiều động lực cho sự phát triển của hình ảnh y học. Nhờ các mô hình trí tuệ nhân tạo, các bác sĩ có thể tăng số lượng thăm khám bệnh nhân khi độ chính xác được cải thiện, người bệnh tiết kiệm chi phí điều trị vì được xây dựng phác đồ điều trị phù hợp với tình trạng bệnh.

II. DỮ LIỆU CHUỖI THỜI GIAN

1. Chuỗi thời gian

Dữ liệu chuỗi thời gian là dữ liệu được đo đạc một cách tuần tự theo thời gian. Dữ liệu chuỗi thời gian được ứng dụng trong rất nhiều lĩnh vực khác nhau như: thống kê, chứng khoán, dịch vụ tài chính, dự báo thời tiết, y học,... Ví dụ về chuỗi thời gian là điện tâm đồ, lượng mưa hàng năm ở Việt Nam, thời tiết trong tuần,... Hình 3 minh họa một ví dụ về chuỗi thời gian biểu diễn kết quả điện tâm đồ.

Một số bài toán điển hình trong khai phá dữ liệu chuỗi thời gian bao gồm: Gom cụm (Clustering), Phân lớp (Classification), Phát hiện Motif (Motif detection), phát hiện chuỗi bất thường (Anomaly detection),...



Hình 3: Minh họa dữ liệu chuỗi thời gian

2. Phát hiện bất thường trên dữ liệu chuỗi thời gian

Các lĩnh vực nghiên cứu như y học, kinh tế tài chính,... thường cần độ chính xác rất cao. Trong khi đó, những bất thường trên dữ liệu chuỗi thời gian ảnh hưởng rất lớn đến kết quả khai phá dữ liệu. Vì vậy bài toán xác định chuỗi bất thường trên dữ liệu chuỗi thời gian đóng vai trò rất quan trọng và thường được xem như bước tiền xử lý cho việc khai phá dữ liệu chuỗi thời gian.

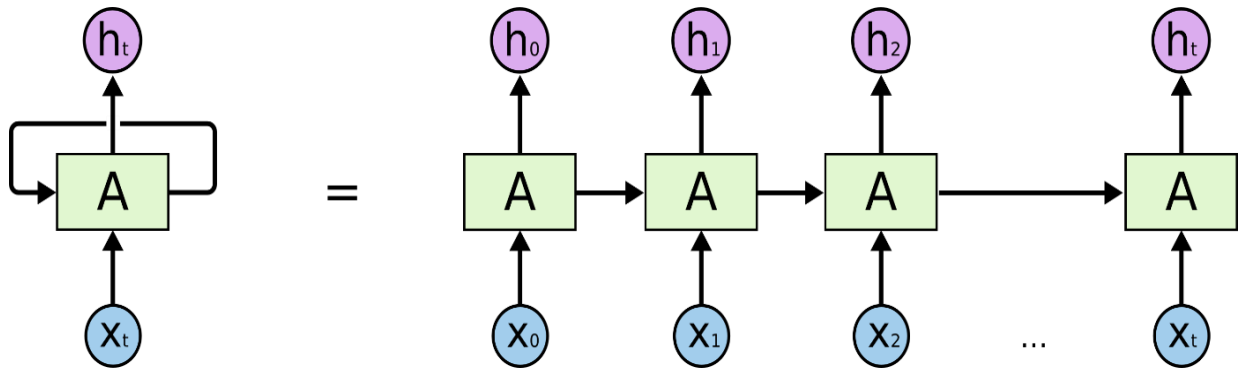
Phát hiện bất thường của nhịp tim bằng dữ liệu điện tâm đồ là một ứng dụng quan trọng của bài toán phát hiện chuỗi bất thường trên dữ liệu chuỗi thời gian. Điện tâm đồ (ECG) là phương pháp theo dõi hoạt động, tốc độ cũng như nhịp điệu của tim, khi hoạt động tim co bóp sẽ phát ra các biến thiên của dòng điện, lúc này điện tâm đồ là một đường cong có chức năng ghi lại các biến thiên đó. Một bất thường trong dữ liệu này có thể là một mẫu không phù hợp về mặt chu kỳ hoặc biên độ, điều này có thể chỉ ra rằng có vấn đề sức khỏe.

III. LSTM AUTOENCODER

1. Long Short Term Memory

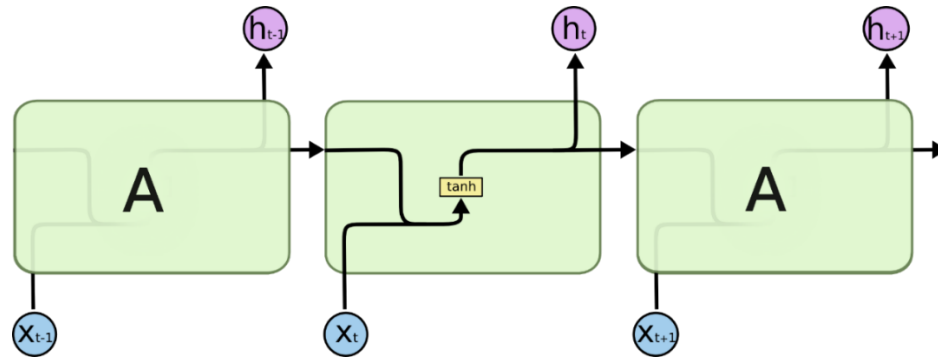
Học sâu là tập hợp các thuật toán để mô hình hóa dữ liệu trừu tượng bằng cách sử dụng một mạng lưới nơ ron thần kinh lớp với cấu trúc phức tạp, tương tự như thiết kế của não bộ con người. Học sâu lấy ý tưởng dựa trên tân vỏ não, các nút phân được được sắp xếp trong một loạt các đường dẫn truyền dữ liệu ở giữa, chúng có thể kết với nhau giống như các nút lớp phân tầng. Các mô hình học sâu ngày càng được áp dụng rộng rãi vào các vấn đề thực tế như nhận diện, phân loại,... với độ chính xác ngày càng cao, chúng được đào tạo bằng các kiến trúc mạng nơ ron nhiều lớp. Tuy nhiên, các mô hình mạng nơ-ron truyền thống này có một khuyết điểm lớn là thiếu bộ nhớ để lưu lại những gì diễn ra trước đó, do đó khó khăn trong một số vấn đề cần dựa vào thông tin trước đó.

Mạng nơ-ron hồi quy (Recurrent Neural Network) minh họa như Hình 4 ra đời để giải quyết vấn đề nêu trên. Mạng này chứa vòng lặp giúp lưu lại các thông tin trước đó. Hình 4 mô tả diễn giải kiến trúc của mạng nơ-ron hồi quy A với đầu vào x_t và đầu ra h_t , về cơ bản thì kiến trúc RNN cũng tương tự với các mạng nơ ron truyền thống. RNN gồm một vòng lặp cho phép các thông tin có thể được truyền từ bước này qua bước khác của mạng nơ-ron, thông tin vừa là đầu ra của mạng này đồng thời là đầu vào của mạng khác.



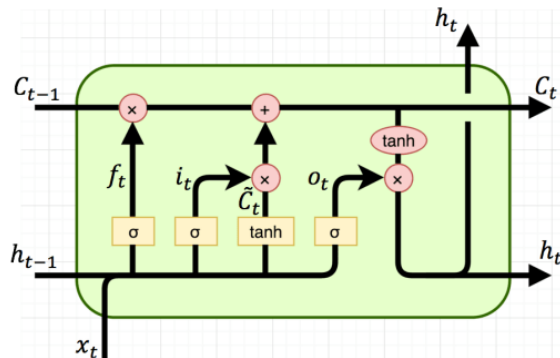
Hình 4: Recurrent Neural Network

Mọi mạng hồi quy (RNN) đều có dạng một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-đun này có cấu trúc rất đơn giản, thường là một hàm tanh. Hình 5 minh họa cấu trúc một RNN chuẩn. Ý tưởng chính của RNN chính là sử dụng lại những thông tin đã có trước đó để dự đoán hiện tại, tương tự việc con người dùng những gì học được trong quá khứ nhằm quyết định tương lai. Tuy nhiên, RNN có một vấn đề là với những thông tin có khoảng cách xa thì chúng bắt đầu quên (phụ thuộc xa) và không thể tiếp tục học. Về mặt lý thuyết, RNN vẫn có khả năng xử lý thông tin ở khoảng cách xa bằng cách chỉnh sửa các tham số một cách khéo léo. Trong thực tế, thì RNN khó có thể học được các số đó.



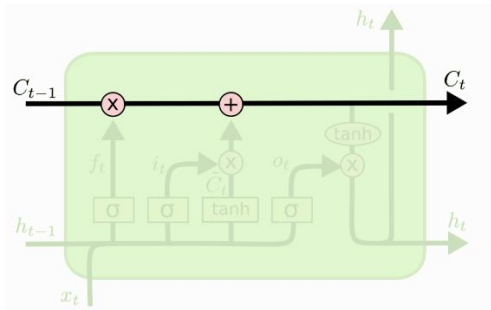
Hình 5: Module của một RNN

Long Short Term Memory networks (LSTM – mạng bộ nhớ dài ngắn) là một dạng đặc biệt của RNN, ra đời nhằm giải quyết vấn đề phụ thuộc xa nêu ra bên trên. LSTM có khả năng trích xuất tác động ngắn hạn và cả dài hạn của sự kiện trong quá khứ, tránh được vấn đề phụ thuộc xa của RNN. LSTM cũng có kiến trúc dạng chuỗi, nhưng các mô-đun trong nó có cấu trúc mạng khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron thì chúng có tới 4 tầng tương tác với nhau một cách rất khác biệt minh họa như Hình 6. Nhớ thông tin trong thời gian dài là đặc trưng của LSTM, chúng hoạt động hiệu quả trên nhiều bài toán khác nhau và ngày càng phổ biến.

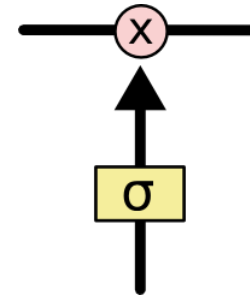


Hình 6: Cấu trúc của Long Short Term Memory

Điểm khác biệt cốt lõi của LSTM so với RNN chính là có thêm trạng thái tế bào (cell state) như hình Hình 7. Đây chính là chìa khóa giúp cho LSTM có khả năng lưu lại các thông tin trong quá khứ, được xem như bộ nhớ dài hạn mạng. Các tế bào truyền xuyên suốt các mô đun và chỉ tương tác để xem xét thông tin có cần bỏ đi hay không dựa vào các cổng (gate) như Hình 8. Cổng làm nhiệm vụ kiểm tra thông tin đi qua có cần quên đi hay không, chúng gồm một tầng mạng sigmoid và một phép nhân. Đầu ra sẽ nằm trong đoạn đoạn $[0,1]$ mô tả bao nhiêu thông tin được đi qua. Kết quả là 0 thì nghĩa là tất cả thông tin đều sẽ bị quên, ngược lại bằng 1 thì có nghĩa là ghi nhớ tất cả thông tin.

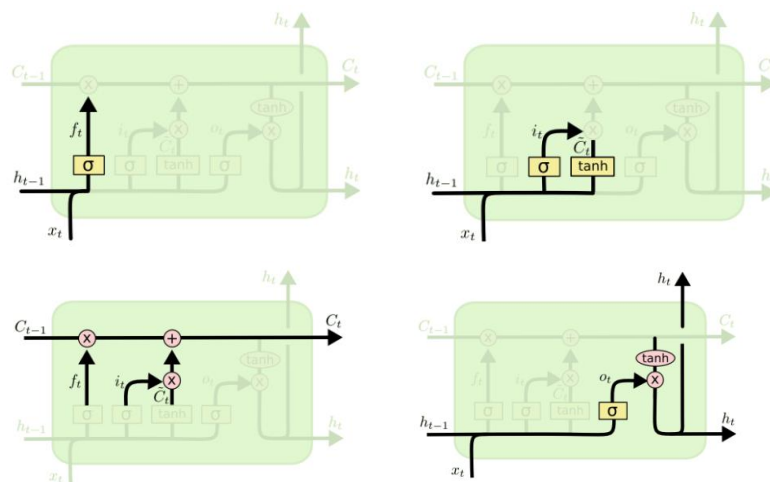


Hình 7: Trạng thái tế bào



Hình 8: Cổng gate

Hoạt động của LSTM bắt đầu bằng việc xem xét thông tin nào cần được quên, quyết định này được đưa ra bởi tầng sigmoid – tầng cổng quên. Tiếp theo là xem xét thông tin mới nào sẽ được lưu vào trạng thái tế bào, việc này kết hợp giữa tầng sigmoid (được gọi là tầng cổng vào) và tầng tanh. Cuối cùng là quyết định kết quả đầu ra, dựa vào trạng thái tế bào, hàm tanh và cổng sigmoid để đưa kết quả đầu ra.

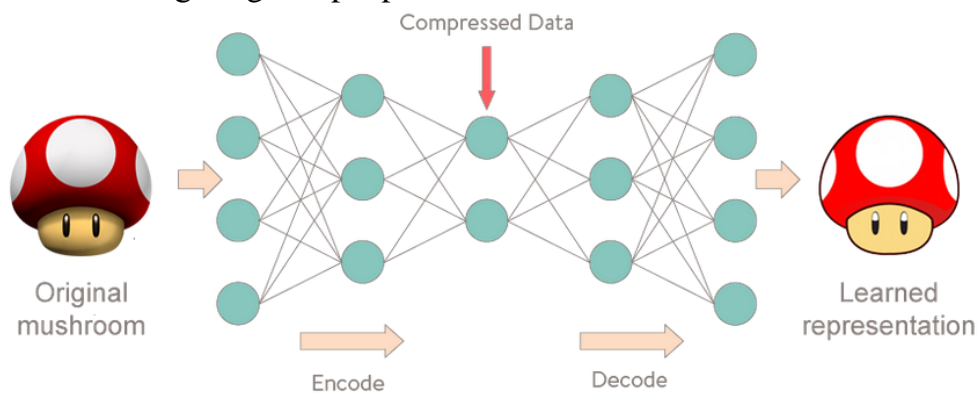


Hình 9: Quá trình hoạt động một module LSTM

2. Autoencoder

Autoencoder là một mạng thần kinh đặc biệt được huấn luyện để sao chép đầu vào của nó thành đầu ra của nó. Ý tưởng chính của autoencoder được sử dụng để giảm kích thước dữ liệu hoặc học các đặc trưng dữ liệu. Một autoencoder có ba phần chính: một bộ mã hóa, mã và một bộ giải mã. Kiến trúc này được trình bày một cách trực quan như Hình 10. Autoencoder được thiết kế để không thể sao chép một cách hoàn hảo, thông thường chúng chỉ sao chép gần đúng với dữ liệu đào tạo. Mô hình này buộc phải tập trung vào đặc trưng ưu tiên của đầu vào để sao chép nên chúng có thể học được các thuộc tính hữu ích của dữ liệu.

Đầu vào của mạng này là dữ liệu không có nhãn, thuộc dạng học không giám sát. Tùy thuộc vào bài toán cụ thể mà đầu ra giá trị có thể khác nhau. Autoencoder được ứng dụng trong một số lĩnh vực như giảm chiều dữ liệu, khử nhiễu, phân loại,... Trong nghiên cứu này tập trung vào phân loại là chính. Ý tưởng chính của việc phân loại là xây dựng một trình tự mã tự động để tái tạo lại phiên bản của một lớp cụ thể dùng để đào tạo, khi có một lớp mới cần phân loại sẽ cung cấp chúng cho bộ mã tự động tái tạo lại. Trong quá trình tái tạo đầu ra, các lỗi xây dựng sẽ được ghi nhận lại và dùng để phát hiện bất thường khi lỗi vượt ngưỡng cho phép.



Hình 10: Mô hình trực quan của Autoencoder

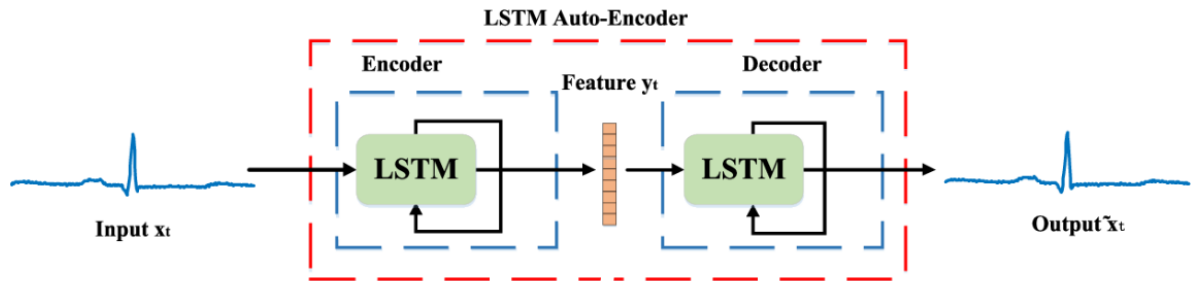
Kiến trúc của Autoencoder gồm 3 phần chính:

- Encoder: Module có nhiệm vụ là nén dữ liệu đầu vào thành một biểu diễn được mã hóa, có kích thước nhỏ hơn so với dữ liệu đầu vào.
- Bottleneck: Module chứa các tri thức được nén, là đầu ra của encoder. Đây là phần quan trọng nhất của mạng bởi nó mang đặc trưng của dữ liệu đầu vào, dùng để tái tạo lại trong quá trình decode.
- Decoder: Module có nhiệm vụ giải nén và tái cấu trúc từ bottleneck, mô hình học được so sánh với dữ liệu ban đầu.

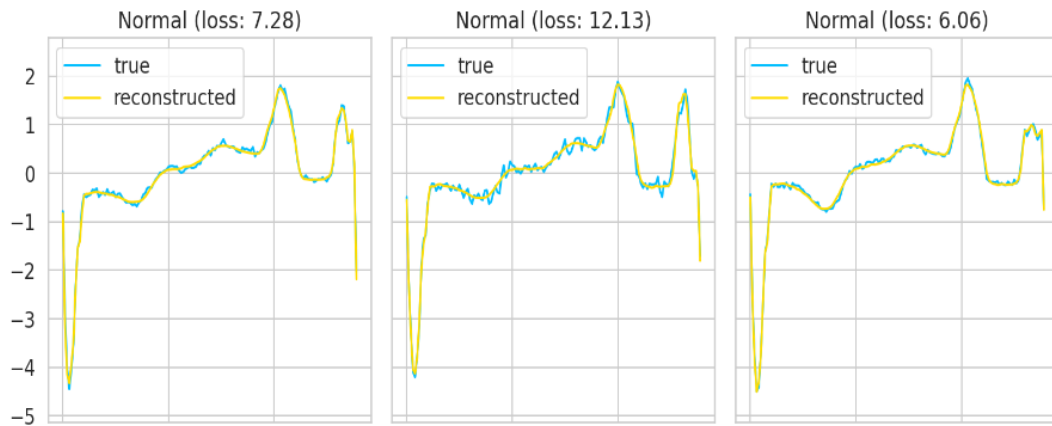
3. LSTM Autoencoder

LSTM Autoencoder là một triển khai của mô hình autoencoder (được trình bày như trên) được dùng để đọc chuỗi đầu vào, mã hóa, giải mã và tái tạo lại chuỗi.

Hình 11 trình bày hoạt động của LSTM Autoencoder, mô hình LSTM (được trình bày như trên) được dùng để đọc tuần tự chuỗi đầu vào. Sau khi đọc toàn bộ chuỗi, trạng thái ẩn hoặc đầu ra của mô hình biểu diễn chuỗi đầu vào dưới dạng vecto có độ dài cố định. Vecto này sau đó được cung cấp như đầu vào cho bộ giải mã và tái tạo lại chuỗi. Hiệu suất của mô hình được đánh giá trên khả năng tái tạo lại trình tự của chuỗi đầu vào. Các vecto kết quả được ứng dụng trong nhiều lĩnh vực khác nhau, đặc biệt là biểu diễn nén như chuỗi đầu vào của mô hình học có giám sát.



Hình 11: Hoạt động của LSTM Autoencoder tái tạo ECG



Hình 12: Chuỗi thời gian trước và sau khi tái tạo

Hình 12 mô tả kết quả chuỗi thời gian trước và sau khi tái tạo dùng mô hình LSTM Autoencoder.

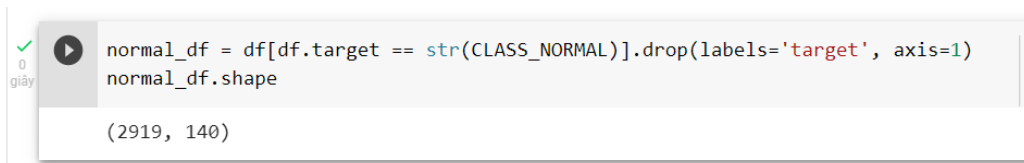
CHƯƠNG 3. KẾT QUẢ THỰC HIỆN

I. TIỀN XỬ LÝ DỮ LIỆU

Mô hình LSTM Autoencoder được viết bằng ngôn ngữ Python, phiên bản 3.7.13. Mạng LSTM được xây dựng bằng thư viện PyTorch. Quá trình xây dựng mô hình sử dụng bộ dữ liệu ECG5000, trong đó nhịp tim bình thường làm dữ liệu huấn luyện và ghi lại tổn thất trong quá trình tái tạo.

Tiền xử lý dữ liệu gồm các bước như sau:

- Lấy tất cả dữ liệu của lớp bình thường gán vào tập `normal_df` và xóa cột nhãn, minh họa như Hình 13.

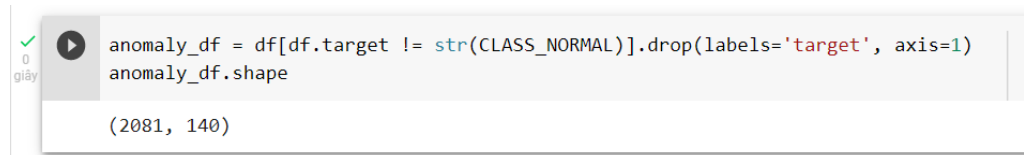


```
normal_df = df[df.target == str(CLASS_NORMAL)].drop(labels='target', axis=1)
normal_df.shape
```

(2919, 140)

Hình 13: Tạo dữ liệu lớp bình thường

- Lấy tất cả dữ liệu của các lớp không bình thường gán vào tập `anomaly_df` và xóa cột nhãn, minh họa như Hình 14.

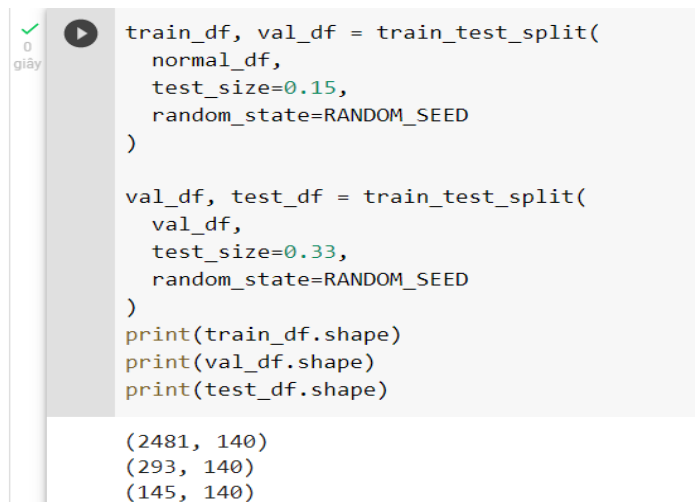


```
anomaly_df = df[df.target != str(CLASS_NORMAL)].drop(labels='target', axis=1)
anomaly_df.shape
```

(2081, 140)

Hình 14: Tạo dữ liệu lớp không bình thường

- Phân chia dữ liệu lớp bình thường thành tập huấn luyện (`train_df`), tập xác thực (`val_df`) và tập kiểm tra (`test_df`), minh họa như Hình 15.



```
train_df, val_df = train_test_split(
    normal_df,
    test_size=0.15,
    random_state=RANDOM_SEED
)

val_df, test_df = train_test_split(
    val_df,
    test_size=0.33,
    random_state=RANDOM_SEED
)
print(train_df.shape)
print(val_df.shape)
print(test_df.shape)
```

(2481, 140)
(293, 140)
(145, 140)

Hình 15: Phân chia tập dữ liệu

- Chuyển đổi chuỗi thời gian thành dạng vecto, minh họa như Hình 16.

```
[18] def create_dataset(df):
    sequences = df.astype(np.float32).to_numpy().tolist()
    dataset = [torch.tensor(s).unsqueeze(1).float() for s in sequences]
    n_seq, seq_len, n_features = torch.stack(dataset).shape
    return dataset, seq_len, n_features

train_dataset, seq_len, n_features = create_dataset(train_df)
val_dataset, _, _ = create_dataset(val_df)
test_normal_dataset, _, _ = create_dataset(test_df)
test_anomaly_dataset, _, _ = create_dataset(anomaly_df)
```

Hình 16: Chuyển đổi chuỗi thời gian

II. XÂY DỰNG MÔ HÌNH

Mô hình được xây dựng gồm các bước:

- Class Encoder dùng nén dữ liệu đầu vào, minh họa như Hình 17.

```
class Encoder(nn.Module):
    def __init__(self, seq_len, n_features, embedding_dim=64):
        super(Encoder, self).__init__()
        self.seq_len, self.n_features = seq_len, n_features
        self.embedding_dim, self.hidden_dim = embedding_dim, 2 * embedding_dim

        self.rnn1 = nn.LSTM(
            input_size=n_features,
            hidden_size=self.hidden_dim,
            num_layers=1,
            batch_first=True
        )

        self.rnn2 = nn.LSTM(
            input_size=self.hidden_dim,
            hidden_size=embedding_dim,
            num_layers=1,
            batch_first=True
        )

    def forward(self, x):
        x = x.reshape((1, self.seq_len, self.n_features))
        x, (_, _) = self.rnn1(x)
        x, (hidden_n, _) = self.rnn2(x)

        return hidden_n.reshape((self.n_features, self.embedding_dim))
```

Hình 17: Class Encoder

- Class Decoder dùng để giải mã, minh họa như Hình 18.

```

class Decoder(nn.Module):
    def __init__(self, seq_len, input_dim=64, n_features=1):
        super(Decoder, self).__init__()
        self.seq_len, self.input_dim = seq_len, input_dim
        self.hidden_dim, self.n_features = 2 * input_dim, n_features

        self.rnn1 = nn.LSTM(
            input_size=input_dim,
            hidden_size=input_dim,
            num_layers=1,
            batch_first=True
        )

        self.rnn2 = nn.LSTM(
            input_size=input_dim,
            hidden_size=self.hidden_dim,
            num_layers=1,
            batch_first=True
        )
        self.output_layer = nn.Linear(self.hidden_dim, n_features)

    def forward(self, x):
        x = x.repeat(self.seq_len, self.n_features)
        x = x.reshape((self.n_features, self.seq_len, self.input_dim))
        x, (hidden_n, cell_n) = self.rnn1(x)
        x, (hidden_n, cell_n) = self.rnn2(x)
        x = x.reshape((self.seq_len, self.hidden_dim))

        return self.output_layer(x)

```

Hình 18: Class Decoder

- Xây dựng hàm sử dụng encoder và decoder

```

class RecurrentAutoencoder(nn.Module):
    def __init__(self, seq_len, n_features, embedding_dim=64):
        super(RecurrentAutoencoder, self).__init__()

        self.encoder = Encoder(seq_len, n_features, embedding_dim).to(device)
        self.decoder = Decoder(seq_len, embedding_dim, n_features).to(device)

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)

        return x

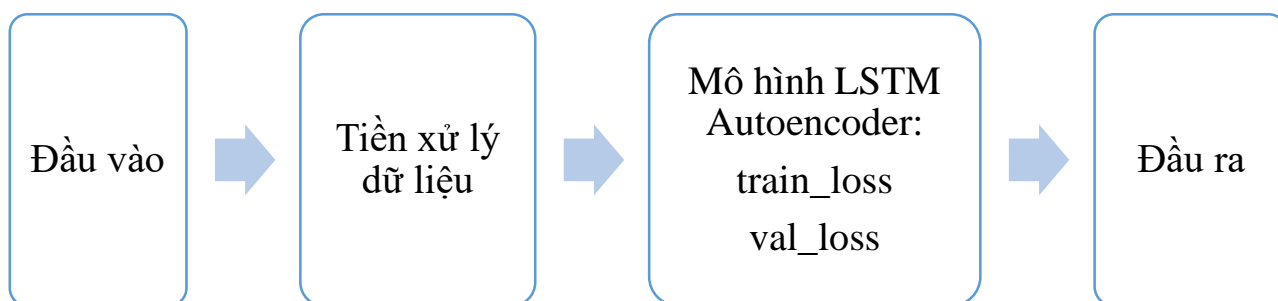
```

Hình 19: Sử dụng encoder và decoder

III. ĐÀO TẠO MÔ HÌNH

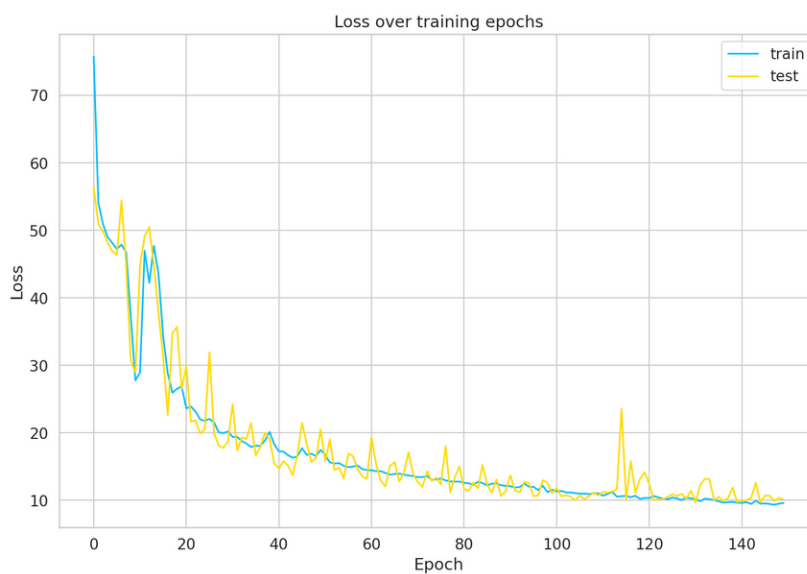
Các bước đào tạo mô hình được trình bày như Hình 20, gồm các bước:

- Tiền xử lý dữ liệu.
- LSTM dùng để đọc chuỗi đầu vào.
- Mã hóa bằng encoder.
- Giải mã bằng decoder.
- Đầu ra.



Hình 20: Quá trình đào tạo mô hình

Tại mỗi epoch, quá trình đào tạo cung cấp cho mô hình tất cả dữ liệu dùng để đào tạo và đánh giá hiệu suất trên bộ xác thực. Kết quả sau 150 epochs, những tổn thất sau trong quá trình đào tạo và xác nhận được ghi lại như . Mô hình có kết quả hội tụ khá tốt.



Hình 21: Tổn thất trong quá trình đào tạo.

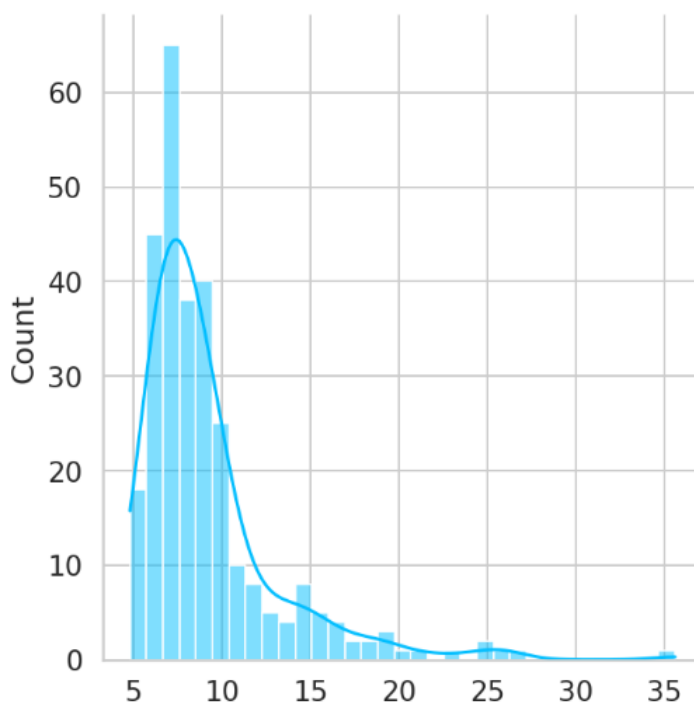
CHƯƠNG 4. ĐÁNH GIÁ KIỂM THỬ

I. CHỌN NGƯỠNG ĐÁNH GIÁ

Với mô hình đã huấn luyện, cần xem xét lỗi xây dựng trên tập huấn luyện để chọn ngưỡng phát bất thường. Việc sử dụng ngưỡng, biến vấn đề đặt ra trong bài toán thành nhiệm vụ phân loại nhị phân:

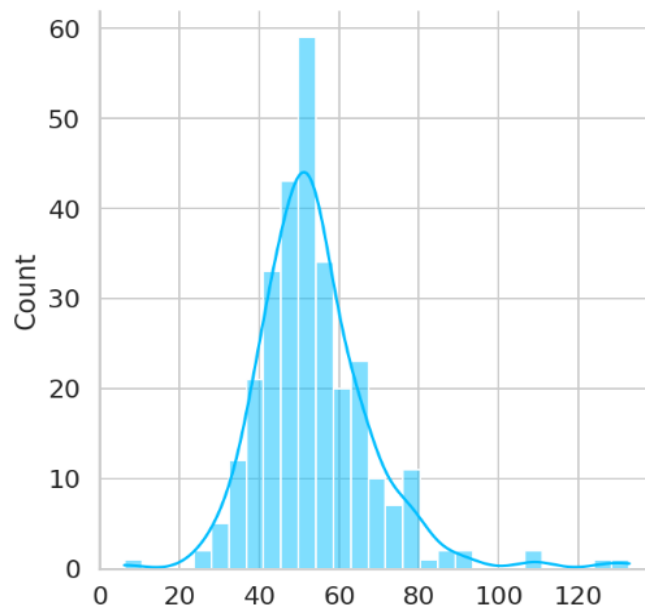
- Nếu tổn thất tái tạo dưới ngưỡng, kết quả phân loại rồi loạn sẽ là bình thường.
- Ngược lại, nếu tổn thất cao hơn ngưỡng sẽ là bất thường.

Để kiểm tra mô hình hoạt động tốt như thế nào với nhịp tim bình thường, mô hình sử dụng nhịp tim bình thường từ bộ dữ liệu huấn luyện và bộ dữ liệu kiểm tra để đánh giá. Kết quả như Hình 22, lỗi trong quá trình đào tạo vào khoảng từ 5 đến 26 do đó có thể chọn ngưỡng tổn thất 26 để nói rằng kết quả phân loại là bình thường.



Hình 22: Losses tập dữ liệu bình thường

Tương tự, kiểm tra mô hình với tập dữ liệu bất thường thu được kết quả như Hình 23. Kết quả lỗi trong quá trình tái tạo nằm trong đoạn 26 trở lên do ta sẽ thử với các ngưỡng từ 26 để nói rằng kết quả phân loại là không bình thường.



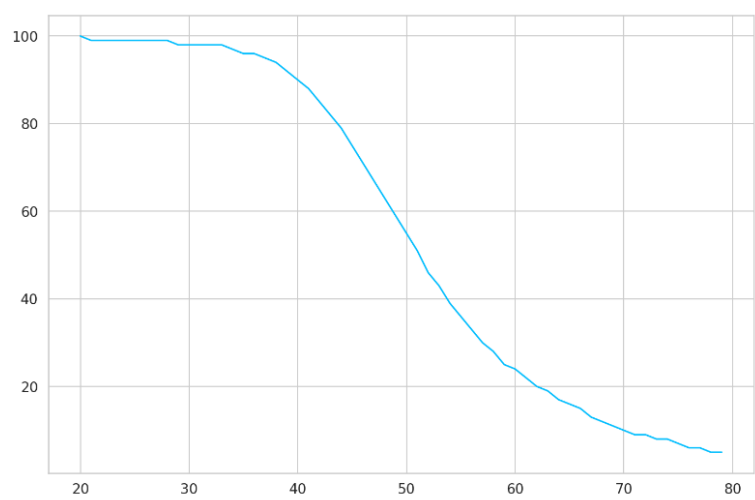
Hình 23: Losses tập dữ liệu không bình thường

II. ĐÁNH GIÁ MÔ HÌNH

Kết quả dự đoán đúng lớp không bình thường khi thay đổi ngưỡng được trình bày như Bảng 2 và biểu diễn như Hình 24.

Bảng 2: Kết quả dự đoán

Threshold	Correct	Percent	Threshold	Correct	Percent
20	1783	99.55	38	1675	93.52
22	1782	99.5	40	1620	90.45
24	1778	99.27	42	1527	85.26
26	1774	99.05	44	1414	78.95
28	1765	98.55	46	1280	71.47
30	1760	98.27	48	1129	63.04
32	1752	97.82	50	979	54.66
34	1738	97.04	52	832	46.45
36	1711	95.53	54	701	39.14



Hình 24: Kết quả dự đoán

PHẦN 3. KẾT LUẬN

I. KẾT QUẢ ĐẠT ĐƯỢC

Một số kết quả chính đạt được:

- Xây dựng được mô hình LSTM Autoencoder dựa trên lý thuyết đã trình bày.
- Phân loại được rối loạn nhịp tim dựa trên dữ liệu điện tâm đồ.
- Đã nghiên cứu được các lý thuyết về chuỗi thời gian, mạng LSTM, mạng LSTM Autoencoder.

II. HẠN CHẾ

Đề tài chỉ mới dừng lại ở mức phân loại nhị phân, chưa phân loại được thành các lớp cụ thể. Ngoài ra, tập dữ liệu huấn luyện chưa đủ lớn để đánh giá tính chính xác khi áp dụng và thực tế.

Kiến thức về chẩn đoán nhịp đoán nhịp tim liên quan đến kiến thức chuyên sâu của ngành y do đó khó để tiếp cận và hiểu rõ.

III. HƯỚNG PHÁT TRIỂN

Tiếp tục nghiên cứu nâng cao chất lượng chẩn đoán, phân lớp cụ thể hơn của từng rối loạn. Bổ sung thêm một số thông tin để có thể áp dụng vào thực tiễn.

Nghiên cứu thêm một số chuyên môn tim mạch để áp dụng vào việc xây dựng mô hình một cách chính xác.

TÀI LIỆU THAM KHẢO

- [1] Venelin Valkov. Time Series Anomaly Detection Tutorial with PyTorch in Python | LSTM Autoencoder for ECG Data.
- [2] Sajad Mousavi, Fatemeh Afghah. Inter- and intra- patient ECG heartbeat classification for arrhythmia detection: a sequence to sequence deep learning approach.
- [3] Huỳnh Cẩm. Phát hiện chuỗi bất thường trên dữ liệu chuỗi thời gian.
- [4] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, Gautam Shroff. LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection.
- [5] dominhhai.github.io. [RNN] LSTM là gì?
- [6] www.mountelizabeth.com.sg. Tim của bạn đã từng lỗi 1 nhịp? Tất cả về chứng rối loạn nhịp tim.