**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

Ảnh có chứa biểu tượng, Đồ họa, hình mẫu, Phông chữ

Mô tả được tạo tự động🙥**🕮**🙧

**BÁO CÁO HỆ HỖ TRỢ QUYẾT ĐỊNH**

**TÌM HIỂU BÀI BÁO  
“Decision Support System for**

**Predicting Survivability of Hepatitis Patients”**

**Lớp:** IS254.O21

**GVHD:** ThS.Nguyễn Hồ Duy Trí

**Nhóm sinh viên thực hiện:**

Nguyễn Thanh Sơn 20521847

Nguyễn Thị Thu Thủy 20520797

Nguyễn Cẩm Tú 20520837

Phạm Thanh Nhựt 20521728

**Thành phố Hồ Chí Minh, tháng 5 năm 2024**

# NHẬN XÉT CỦA GIẢNG VIÊN

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

# MỤC LỤC

[NHẬN XÉT CỦA GIẢNG VIÊN 2](#_Toc165578242)

[MỤC LỤC 3](#_Toc165578243)

[PHỤ LỤC HÌNH ẢNH 4](#_Toc165578244)

[PHỤ LỤC BẢNG 5](#_Toc165578245)

[NỘI DUNG 6](#_Toc165578246)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 6](#_Toc165578247)

[1.1. Lý do chọn đề tài 6](#_Toc165578248)

[1.2. Giới thiệu bài báo 7](#_Toc165578249)

[1.3. Bộ dữ liệu 7](#_Toc165578250)

[CHƯƠNG 2: HỆ THỐNG HỖ TRỢ RA QUYẾT ĐỊNH – DSS 9](#_Toc165578251)

[CHƯƠNG 3: CÁC KỸ THUẬT, MÔ HÌNH, THUẬT TOÁN SO SÁNH 18](#_Toc165578252)

[3.1. Tiền xử lý dữ liệu 18](#_Toc165578253)

[3.2. Các thuật toán so sánh 18](#_Toc165578254)

[3.3. Thông số đánh giá 25](#_Toc165578255)

[CHƯƠNG 4: THỰC NGHIỆM & KẾT QUẢ 27](#_Toc165578256)

[4.1. Thực nghiệm 1 27](#_Toc165578257)

[4.2. Thực nghiệm 2 27](#_Toc165578258)

[4.3. Thực nghiệm 3 28](#_Toc165578259)

[4.4. Thực nghiệm 4 29](#_Toc165578260)

[CHƯƠNG 5: KẾT LUẬN 30](#_Toc165578261)

[5.1. Phân tích kết quả 30](#_Toc165578262)

[5.2. Hướng phát triền 31](#_Toc165578263)

[THAM KHẢO 32](#_Toc165578264)

[BẢNG PHÂN CÔNG CÔNG VIỆC 33](#_Toc165578265)

# PHỤ LỤC HÌNH ẢNH

[Hình 1‑1. Minh hoạ bệnh gan 6](#_Toc165578266)

[Hình 2‑1. Bản xem trước của hệ thống dự đoán bệnh viêm gan được đề xuất 9](#_Toc165578267)

[Hình 2‑2. BiLSTM được sử dụng cho hệ thống dự đoán viêm gan. 13](#_Toc165578268)

[Hình 2‑3. Nhập thông tin bệnh nhân 15](#_Toc165578269)

[Hình 2‑4. Tải lên tập dữ liệu 16](#_Toc165578270)

[Hình 2‑5. Tiền xử lý dữ liệu 16](#_Toc165578271)

[Hình 2‑6. Huấn luyện và kiểm tra mô hình 17](#_Toc165578272)

[Hình 2‑7. Dự đoán khả năng sống sót của bệnh nhân 17](#_Toc165578273)

[Hình 4‑1. Minh hoạ Linear SVM và Non-linear SVM 19](#_Toc165578274)

[Hình 4‑2. Minh họa thuật toán KNN 20](#_Toc165578275)

[Hình 4‑3. Cấu trúc của LSTM 21](#_Toc165578276)

[Hình 4‑4. Cấu trúc cơ bản của CNN 23](#_Toc165578277)

[Hình 4‑5. Hàm ReLU 24](#_Toc165578278)

[Hình 4‑6. Sơ đồ mô hình RNN 24](#_Toc165578279)

[Hình 5‑1. Kết quả 10 thí nghiệm BiLSTM 27](#_Toc165578280)

[Hình 5‑2. Kết quả so sánh giữa dữ liệu cân bằng và mất cân bằng 28](#_Toc165578281)

[Hình 5‑3. So sánh các nghiên cứu khác và mô hình BiLSTM 29](#_Toc165578282)

# PHỤ LỤC BẢNG

[Bảng 1. Bảng mô tả các thuộc tính 8](#_Toc165498925)

[Bảng 2. Phân tích dữ liệu về bệnh viêm gan 10](#_Toc165498926)

[Bảng 3. Trình phân loại dựa trên ML so với mô hình được đề xuất (BiLSTM) 28](#_Toc165498927)

[Bảng 4. BiLSTM so với các mô hình học sâu (DL) khác. 28](#_Toc165498928)

[Bảng 5. Chuyên gia về con người so với tiên lượng hệ thống được đề xuất 30](#_Toc165498929)

# NỘI DUNG

## TỔNG QUAN ĐỀ TÀI

### Lý do chọn đề tài

Viêm gan virus là một vấn đề sức khỏe chung của cộng đồng lớn trên toàn cầu. Nó chủ yếu ảnh hưởng đến các quốc gia kém phát triển nhất trên thế giới, đặc biệt là các khu vực có tính dịch tễ cao mà lại có nguồn lực hạn chế và chỉ số phát triển con người thấp. Viêm gan mãn tính có thể dẫn đến xơ gan, suy gan, ung thư và cuối cùng là tử vong. Chẩn đoán và điều trị sớm nhiễm viêm gan có thể giúp giảm gánh nặng bệnh tật và lây truyền cho những người có nguy cơ nhiễm bệnh hoặc tái nhiễm. Sàng lọc là điều quan trọng để đạt được các mục tiêu của WHO vào năm 2030. Do đó, các hệ thống tự động để dự đoán chính xác bệnh viêm gan là cần thiết.

Ảnh có chứa Tác phẩm nghệ thuật của trẻ con, tác phẩm nghệ thuật, hình vẽ, trái tim

Mô tả được tạo tự động

Hình 1‑1. Minh hoạ bệnh gan

Các nghiên cứu trước đây sử dụng máy học truyền thống và mạng nơ-ron nhân tạo vẫn chưa đạt được kết quả tối ưu trong việc dự đoán viêm gan từ dữ liệu không cân bằng. Do đó, cần có một phương pháp mới, tiên tiến hơn.

Bài nghiên cứu đã tiến hành thực hiện nhiều mô hình lẫn máy học và học sâu truyền thống như là Support Vector Machine, K-Nearest Neighbors, Naive Bayes, LSTM, CNN, RNN và BiLSTM nhưng trong đó mô hình BiLSTM (bước nhớ dài/ngắn hạn hai chiều) dựa trên học sâu được đề xuất có tiềm năng cải thiện đáng kể độ chính xác trong phân loại và dự đoán khả năng sống sót của bệnh nhân viêm gan là cao nhất.

Một hệ thống hỗ trợ quyết định chính xác dựa trên BiLSTM có thể trợ giúp đáng kể cho các nhà lâm sàng trong việc đưa ra chẩn đoán và điều trị phù hợp, góp phần giảm tỷ lệ tử vong do viêm gan.

### Giới thiệu bài báo

Nguồn bài báo: [Frontiers | Decision Support System for Predicting Survivability of Hepatitis Patients (frontiersin.org)](https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2022.862497/full?fbclid=IwZXh0bgNhZW0CMTAAAR2gk2aewVfyZxMSGTlNfLOGeHJI9pTEhN0a8AxttdnG433iXMTxOPMSPFg_aem_AYam5SWaMch3dTqliHawapzRduV88NcvALvl1DQWO_EOvocOgsMtzSSUKcfbR2-xpbiyiI-BqTPODbUlgjQyWG8y)

Bài báo *“Decision Support System for Predicting Survivability of Hepatitis Patients”* của nhóm tác giả Fahad R. Albogamy, Junaid Asghar, Fazli Subhan, Muhammad Zubair Asghar, Mabrook S. Al-Rakhami, Aurangzeb Khan, Haidawati Mohamad Nasir, Mohd Khairil Rahmat, Muhammad Mansoor Alam, Adidah Lajis, Mazliham Mohd Su'ud đã trình bày hệ thống hỗ trợ quyết định dự đoán khả năng sống sót của bệnh nhân viêm gan.

Nghiên cứu này đề xuất một hệ thống hỗ trợ quyết định (DSS) dựa trên học sâu sử dụng mô hình BiLSTM (bước nhớ dài/ngắn hạn hai chiều) để hỗ trợ phân loại nhị phân chính xác về chẩn đoán viêm gan (tỷ lệ sống sót hoặc tử vong) ở bệnh nhân mắc viêm gan nặng. Dữ liệu cân bằng được sử dụng để huấn luyện mô hình, và kết quả thử nghiệm của nghiên cứu cho thấy một hiệu suất rất khả quan với độ chính xác 95,24%, độ dự đoán 95,45%, độ phủ 95,45% và điểm F1 95,45%. Bên cạnh nghiên cứu còn so sánh BiLSTM với các mô hình máy học và học sâu truyền thống khác, tuy nhiên BiLSTM vẫn là mô hình có độ chính xác cao nhất và phù hợp nhất với bộ dữ liệu được đưa vào.

Kết luận của nghiên cứu là việc sử dụng mô hình BiLSTM trong phát hiện viêm gan mang lại hiệu suất phân loại tốt hơn đáng kể so với các phương pháp hiện tại.

### Bộ dữ liệu

Nguồn dữ liệu: [Hepatitis Data (kaggle.com)](https://www.kaggle.com/datasets/codebreaker619/hepatitis-data?fbclid=IwZXh0bgNhZW0CMTAAAR1XXYoiL9M2VIbh_W2FeMlDA7ETBd0yh4q-sVOv-57o61MdNIgLb6y9iRM_aem_AYZO7KM72ReykZg61Vm4nfgKJ99cC5Te9-TMlgNLE8mq4KIlBjS9-uuYwBABrUycrVyJa45P6TdLZ9c5iaeCgttj)

Bộ dữ liệu gồm 155 dòng và 20 cột

Bảng 1 Bảng mô tả các thuộc tính

|  |  |
| --- | --- |
| Attribute | Description |
| Age | Độ tuổi của bệnh nhân |
| Sex | Giới tính của bệnh nhân  *(có thể là "Male" hoặc "Female")* |
| Fatigue | Mức độ mệt mỏi của bệnh nhân  *(có thể là “TRUE” hoặc “FALSE”)* |
| Malaise | Tình trạng tổng thể của bệnh nhân  *(cảm thấy không khỏe, không thoải mái)*  *(có thể là “TRUE” hoặc “FALSE”)* |
| Anorexia | Tình trạng không muốn ăn của bệnh nhân  *(có thể là “TRUE” hoặc “FALSE”)* |
| Liver Big | Kích thước gan của bệnh nhân có lớn hay không? *(có thể là “TRUE” hoặc “FALSE”)* |
| Liver Firm | Đặc tính của gan  *(có thể là “TRUE” hoặc “FALSE”)* |
| Spleen Palpable | Có thể kiểm tra thấy của lách hay không?  *(có thể là “TRUE” hoặc “FALSE”)* |
| Spiders | Mạng tĩnh mạch trên da của bệnh nhân  *(có thể là “TRUE” hoặc “FALSE”)* |
| Ascites | Sự tích nước trong bụng của bệnh nhân  *(có thể là “TRUE” hoặc “FALSE”)* |
| Varices | Sự xuất hiện của các tĩnh mạch giãn nở trên dạ dày hoặc ruột của bệnh nhân  *(có thể là “TRUE” hoặc “FALSE”)* |
| Bilirubin | Mức độ bilirubin trong máu của bệnh nhân |
| Alk Phosphate | Mức độ enzyme alk phoshatase trong máu của bệnh nhân |
| Sgot | Mức độ enzyme SGOT (Aspartate Aminotransferase) trong máu của bệnh nhân |
| Albumin | Mức độ albumin trong máu của bệnh nhân |
| Protime | Thời gian chảy của huyết tương trong bệnh nhân |
| Histology | Kết quả của xét nghiệm mô học  *(có thể là “TRUE” hoặc “FALSE”)* |
| Steroid | Sử dụng steroid  *(có thể là “TRUE” hoặc “FALSE”)* |
| Antivirals | Sử dụng thuốc chống vi rút  *(có thể là “TRUE” hoặc “FALSE”)* |
| Class | Kết quả cuối cùng của bệnh nhân ("Die" nếu bệnh nhân qua đời, "Live" nếu bệnh nhân sống sót) |

## HỆ THỐNG HỖ TRỢ RA QUYẾT ĐỊNH – DSS

Phương pháp đề xuất cho vấn đề ra quyết định phức tạp này nhấn mạnh vào việc sử dụng các kỹ thuật học sâu để tích hợp dữ liệu và kinh nghiệm hiện có vào hệ thống hỗ trợ ra quyết định (DSS). Hệ thống DSS đã được xây dựng để tổng hợp dữ liệu, kinh nghiệm và mô hình, cung cấp cho các chuyên gia y tế lâm sàng về bệnh viêm gan các thông tin cần thiết để đưa ra quyết định chẩn đoán. Để tạo sự thu hút đối với người sử dụng, đã tiến hành tương tác với các chuyên gia y tế ở mọi giai đoạn của quá trình thiết kế. Công trình của Turban et al (Hình 2-1) đã được tham khảo trong việc xây dựng DSS do tính phức tạp của vấn đề ra quyết định và bản chất đặc biệt của DSS. Cấu trúc của DSS bao gồm bốn thành phần chính: quản lý dữ liệu, quản lý mô hình, quản lý dựa trên kiến thức và giao diện người dùng.

Ảnh có chứa văn bản, ảnh chụp màn hình, thiết kế

Mô tả được tạo tự động

Hình 2‑1. Bản xem trước của hệ thống dự đoán bệnh viêm gan được đề xuất

**PHẦN 1: QUẢN LÝ DỮ LIỆU**

Một hệ thống hỗ trợ quyết định (DSS) sử dụng cơ sở dữ liệu và/hoặc các bộ dữ liệu để cung cấp dữ liệu liên quan cho hệ thống hỗ trợ quyết định. Ngoài các nguồn cục bộ, công cộng và tùy chỉnh, dữ liệu DSS cũng có thể được lấy từ các nguồn tổ chức và chính thức. Bộ dữ liệu được sử dụng trong nghiên cứu này là bộ dữ liệu về viêm gan từ kho dữ liệu của Đại học California, Irvine. Nếu nhìn vào bộ dữ liệu có 155 trường hợp với 20 thuộc tính. Bộ dữ liệu chứa 19 thuộc tính và một lớp duy nhất (kết quả), có thể chia thành năm nhóm như sau:

Bảng 2. Phân tích dữ liệu về bệnh viêm gan

|  |  |  |  |
| --- | --- | --- | --- |
| Category | Attribute | Type | Description |
| Patient Profile | Age | Continuous  Min=17, mean=41.2,  meadian=39,  max=78,Sd=12.57 | Patient age |
| Sex | Categorical | Male=139, female=16 |
| Sign and Symptom | Fatigue | Categorical | No=100, yes=54, missing=1 |
| Malaise | Categorical | No=61, yes=93, missing=1 |
| Anorexia | Categorical | No=32, yes=122, missing=1 |
| Pshysical Examination | Liver Big | Categorical | No=25, yes=120, missing=10 |
| Liver Firm | Categorical | No=60, yes=84, missing=11 |
| Spleen Palpable | Categorical | No=30, yes=120,missing=5 |
| Spiders | Categorical | No=,51 yes=99, missing=5 |
| Ascites | Categorical | No=20, yes=130,missing=5 |
| Varices | Categorical | No=,18 yes=132, missing=5 |
| Lab Test | Bilirubin | Continuous  Min=0.3, mean=1.43,  meadian=1.0,  max=8.0, Sd=1.21 | 0.39, 0.80, 1.20, 2.00, 3.0, 4.0 |
| Alk Phosphate | Continuous  Min=26, mean=105.32,  meadian=85,  max=295, Sd=51.51 | 33, 80, 120, 160, 200, 250 |
| Sgot | Continuous  Min=14, mean=85.89,  meadian=58  max=648, Sd=89.65 | 13,100, 200, 300, 400, 500, |
| Albumin | Continuous  Min=2, mean=3.82,  meadian=4  max=6.4, Sd=0.65 | 2.1, 3.0, 3.8, 4.5, 5.0, 6.0 |
| Protime | Continuous  Min=0, mean=61.85,  meadian=61  max=100, Sd=22.85 | 10, 20, 30, 40, 50, 60, 70, 80, 90 |
| Histology | Categorical | No=85, Yes=70 |
| Medication Management | Steroid | Categorical | No=76, Yes=78, missing=1 |
| Antivirals | Categorical | No=24, Yes=131 |
| Target Class | Die, Live | outcome | Die=32, Live=123 |

Sau khi phân loại, tiến hành chia các tập huấn luyện/kiểm tra thông qua việc sử dụng công cụ phân vùng huấn luyện (80%), xác thực (10%), kiểm tra (10%).

*Tập Huấn Luyện:* Khoảng 80% dữ liệu từ tập huấn luyện được sử dụng trong toàn bộ quá trình huấn luyện. Cả các mô tả kết quả (biến phản ứng) và các tham số đầu vào được bao gồm trong giai đoạn huấn luyện (biến dự đoán).

*Tập Xác Thực:* Overfitting và underfitting có thể được xử lý bằng cách sử dụng dữ liệu xác thực trong hệ thống. Mô hình được đánh giá bằng dữ liệu từ một phần tử con xác thực 10%. Khi làm việc với Keras, bạn có thể thay đổi thông số thủ công hoặc tự động. Việc xác thực tự động trong nghiên cứu này sẽ cho phép đánh giá một cách khách quan hơn về chiến lược đề xuất.

*Tập Kiểm Tra:* Tập kiểm tra cung cấp các trường hợp độc đáo để đánh giá hiệu suất của thuật toán. Quá trình này được sử dụng sau quá trình huấn luyện và kiểm tra đáng kể. Đánh giá mô hình có thể được thực hiện thông qua việc sử dụng một tập kiểm tra. 10% dữ liệu kiểm tra được sử dụng, không có mối quan hệ nào với các trường hợp huấn luyện. Nó được sử dụng sau khi mô hình đã được huấn luyện đủ. Sau đó, nó được xác thực đối với dữ liệu thực tế.

**PHẦN 2: QUẢN LÝ MÔ HÌNH**

Được tích hợp vào DSS là một cơ sở mô hình gồm các thuật toán thống kê và khác cho phép cung cấp phân tích phức tạp. Một Hệ thống Quản lý Mô hình (MMS) chuyển đổi dữ liệu từ DMS thành thông tin bằng cách áp dụng các mô hình vào nó. Để phát triển một mô hình dự đoán có khả năng, việc tiền xử lý dữ liệu sức khỏe có sẵn là cần thiết. Việc thay thế giá trị null và các bộ dữ liệu không cân bằng đều được hệ thống quản lý dữ liệu xử lý.

*Quản lý Các Bộ Dữ Liệu Không Cân Bằng:*

Có một sự mất cân bằng đáng kể trong bộ dữ liệu cơ bản, nơi xử lý hai nhóm không cân bằng. Các bệnh nhân mất tích chiếm 21%, trong khi 79% số người còn lại là số người sống sót. Có thể khó để có được dự đoán chính xác từ các thuật toán dự đoán khi dữ liệu không cân bằng. Khi một mô hình học từ dữ liệu bị lệch và phân loại không đồng đều, đầu ra thường ưu tiên lớp chính trong khi bỏ qua các danh mục nhỏ hơn trong quá trình phân loại. Điều này được coi là vấn đề mất cân bằng lớp. Oversampling, một kỹ thuật lấy mẫu xử lý dữ liệu, được sử dụng để làm cho tất cả các trường hợp của một lớp trở nên bằng nhau. Các lớp nhỏ có thể được mở rộng bằng cách lấy mẫu quá mức. Việc tăng tỷ lệ mất cân bằng là điều mà việc lấy mẫu ngẫu nhiên đạt được. Kết quả phân loại đã được cải thiện nhiều khi thêm sự nhân bản nhóm nhỏ.

*Cách xử lý dữ liệu thiếu sót:*

Dữ liệu đáng tin cậy nên được nhập vào mô hình để điền vào các thuộc tính trống. Việc điền vào các khoảng trống có thể được thực hiện bằng cách tính trung bình giá trị của các mục trong các cấp độ trước, chọn một phần tử ngẫu nhiên, hoặc quay trở lại và sử dụng số của cấp độ trước đó. Sau khi quyết định theo con đường thứ ba, tác giả tiếp tục cập nhật tất cả dữ liệu bị thiếu. Trong tập dữ liệu, chỉ có một vài giá trị bị thiếu. Thuộc tính PROTIME có 67 giá trị bị thiếu (43%), chiếm dưới 10% của tổng số tập dữ liệu.

**PHẦN 3: QUẢN TRỊ DỰA TRÊN KIẾN THỨC**

Kết hợp với các mô-đun DSS khác, thành phần này có thể cung cấp thông tin mới nhất để giúp khắc phục vấn đề. Khi thông tin đã được phát hiện, thu thập và tổ chức theo một cách nào đó, việc chuyển đổi thông tin thành tri thức là cần thiết. Phân tích và phân loại dữ liệu là một phần cần thiết của quá trình nghiên cứu. Tóm lại, mô hình được đề xuất bao gồm ba phần chính: cấu trúc dữ liệu lớp nhúng, lưu trữ thông tin ngữ cảnh tiến và lùi, và phân loại dựa trên lớp softmax. Mô-đun thứ hai có thể gói gọn các đặc điểm bằng cách sử dụng biểu diễn số. Sử dụng Bi-LSTM để mã hóa ngữ cảnh dữ liệu trong một chuỗi khi đến lúc mô-đun cuối cùng, kích hoạt softmax được sử dụng để phân loại dữ liệu. Đây là một phân chia nhỏ của mỗi thành phần.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động

Hình 2‑2. BiLSTM được sử dụng cho hệ thống dự đoán viêm gan.

*Lớp Embedding*

Bằng cách sử dụng nhúng, có thể chuyển đổi các danh mục thành một mảng số liên tục có kích thước nhất định (vector). Trong mạng nơ-ron, các yếu tố đại diện cho biến rời rạc dưới dạng nhúng vector chiều thấp, sau đó được đại diện dưới dạng một hàm của dữ liệu đầu vào. Có một số ưu điểm khi sử dụng nhúng nơ-ron trong việc giảm dữ liệu phân loại và xác định các danh mục trong không gian kết quả so với các phương pháp giảm dữ liệu và mô tả khác. Keras giúp việc sử dụng nhúng trở nên dễ dàng hơn. Vector nhúng với các biểu diễn cấp độ thuộc tính được sử dụng để biểu diễn dữ liệu trong bộ dữ liệu viêm gan, với một chiều nhúng bằng hoặc lớn hơn kích thước của dữ liệu đầu vào. Vector nhúng dữ liệu được xây dựng bằng một lớp nhúng Keras với 32 lớp. Trong lớp nhúng, một ma trận nhúng hai chiều (ma trận đặc trưng) được xây dựng trong đó "đại diện cho độ dài của dữ liệu đầu vào" và "chỉ ra chiều của một từ nhúng" được sử dụng làm tham số đầu vào. Khi ma trận nhúng đã được tạo ra, nó được chuyển đến tầng tiếp theo trong cấu trúc.

*Lớp Bi-Directional LSTM:*

Khi dự đoán bệnh viêm gan, kỹ thuật đề xuất sử dụng mạng nơ-ron sâu, đặc biệt là một mạng trí tuệ nhân tạo hai chiều dài và ngắn (BiLSTM), có thể phân biệt giữa D1 (Sức sống của bệnh nhân viêm gan = có) và D2 (Sức sống của bệnh nhân viêm gan = không). Các phụ thuộc dài hạn được phát hiện thông qua việc sử dụng lớp Bi-LSTM. Do đó, nó giúp giữ lại hai ngữ cảnh trước và sau của dữ liệu đã được mã hóa. Thay vì lưu trữ thông tin từ ngữ cảnh trước, một lớp LSTM một chiều đơn giản chỉ giữ lại thông tin đã được lưu trữ. Do đó, Bi-LSTM có thể phân tích dữ liệu bệnh nhân đã được mã hóa với sâu sắc hơn nhiều. Bi-LSTM sử dụng cả hai LSTM thuận và ngược để học ngữ cảnh dữ liệu trong quá khứ và tương lai. Điều này được thực hiện thông qua việc sử dụng các công thức toán học sau đây:

Công thức cho lớp Forward LSTM:

Ảnh có chứa văn bản, Phông chữ, thuật in máy, chữ viết tay

Mô tả được tạo tự động

Công thức cho lớp Backward LSTM:

Ảnh có chứa văn bản, Phông chữ, chữ viết tay, thuật in máy

Mô tả được tạo tự động

*Dự đoán Dựa trên Thuật toán SoftMax*

Sau đó, SoftMax được sử dụng để xác định xác suất dự đoán các nhãn mục tiêu (tức là, bệnh viêm gan). Sử dụng công thức, có thể tính toán giá trị net của đầu vào phương trình.

Ảnh có chứa Phông chữ, thuật in máy, Đồ họa, số

Mô tả được tạo tự động

Vector đầu vào được ký hiệu là “l,” trong khi vector trọng số được ký hiệu là “w.” Độ lệch được biểu diễn bởi “b.” Bằng cách đưa nó vào phương trình, người ta có thể tính toán SoftMax.

Ảnh có chứa Phông chữ, văn bản, ảnh chụp màn hình, Đồ họa

Mô tả được tạo tự động

**PHẦN 4: GIAO DIỆN NGƯỜI DÙNG**

Trong phần mềm, có ba phần chính: thu thập dữ liệu, phân loại và phát hiện viêm gan.

**(1) Nhập thông tin bệnh nhân**: Lịch sử bệnh án của bệnh nhân là bắt buộc cho phần này. Mỗi bệnh nhân trong cơ sở dữ liệu có một mã xác định mới được tạo ra từ thông tin cải thiện và mới nhất. Hệ thống dự đoán viêm gan sử dụng lịch sử bệnh án của bệnh nhân để tạo ra các ước đoán có kiến thức về tình trạng của họ. Ngay khi thông tin cần thiết được nhập vào phần mềm, một định danh cụ thể cho bệnh nhân được tạo ra. Sử dụng định danh này, tình trạng cụ thể của mỗi bệnh nhân có thể được chẩn đoán và theo dõi. Một bản ghi của trường hợp bệnh nhân được hiển thị (Hình 2-3) trong trên giao diện đã được sử dụng để thu thập và đánh giá dữ liệu. Các khẳng định trước đó được áp dụng vào dữ liệu khi nó được xử lý trên máy chủ hệ thống. Chính vì vậy mà chúng ta có một thiết kế có thể được điều chỉnh theo nhu cầu của mỗi bệnh nhân duy nhất.

A screenshot of a computer

Description automatically generated

Hình 2‑3. Nhập thông tin bệnh nhân

**(2) Tải lên tập dữ liệu**: Có thể huấn luyện bộ phân loại và phát triển mô hình bằng cách sử dụng tập dữ liệu viêm gan, như được thể hiện trong .

A screenshot of a computer

Description automatically generated

Hình 2‑4. Tải lên tập dữ liệu

**(3) Tiền xử lý**: Tiền xử lý dữ liệu tạo ra một bộ phân loại và mô hình có thể dự đoán kết quả của viêm gan. Ở phía sau, dữ liệu được tiền xử lý

A screenshot of a computer

Description automatically generated

Hình 2‑5. Tiền xử lý dữ liệu

**(4) Huấn luyện và Kiểm tra mô hình**: Màn hình dưới đây cho thấy điều gì xảy ra khi một bệnh nhân nhấp vào tùy chọn "Huấn luyện mô hình".Nó hoạt động như một công cụ hỗ trợ huấn luyện bằng cách hiển thị dữ liệu được tải. Việc huấn luyện mô hình được kích hoạt bằng cách nhấp vào tùy chọn "train-test model"

A screenshot of a computer

Description automatically generated

Hình 2‑6. Huấn luyện và kiểm tra mô hình

**(5)** **Dự đoán khả năng sống sót của bệnh nhân viêm gan**: Để dự đoán viêm gan, bạn chỉ cần nhập thông tin bệnh nhân cần thiết và nhấp vào tùy chọn "Predict hepatitis" trên trang web. Nút "Update Training Set" kích hoạt một tập dữ liệu huấn luyện được cập nhật. Sau khi nhập chi tiết bệnh tật của bệnh nhân và nhấp vào nút "predict hepatitis", các kết quả sẽ được hiển thị dưới dạng "D1: Hepatitis\_patient\_survivability = yes,", "D2: Hepatitis\_patient\_survivability = no," và một mức độ tự tin dự đoán cho mỗi lựa chọn.

A screenshot of a computer

Description automatically generated

Hình 2‑7. Dự đoán khả năng sống sót của bệnh nhân

## CÁC KỸ THUẬT, MÔ HÌNH, THUẬT TOÁN SO SÁNH

### Tiền xử lý dữ liệu

***Bước 1:*** Xử lý dữ liệu thiếu sót

***Bước 2:*** Xử lý dữ liệu mất cân bằng – Oversampling

Phần Oversampling này sử dụng phương pháp kết hợp giữa SMOTE (Synthetic Minority Over-sampling Technique) và Tomek links. SMOTE tạo ra các mẫu nhân tạo cho lớp thiểu số (trong trường hợp này là lớp "die"), trong khi Tomek links được sử dụng để loại bỏ các mẫu gần lớp thiểu số nhưng thuộc về lớp đa số (trong trường hợp này là lớp "live"), giúp tăng cường sự cân bằng giữa các lớp.

* + - * Đầu tiên, số lượng mẫu trong mỗi nhóm được đếm để biết được tỉ lệ mẫu giữa các nhóm "live" và "die".
      * Từ số lượng mẫu trong mỗi nhóm, tính toán phần trăm số mẫu của mỗi nhóm.
      * Tiếp theo, sử dụng SMOTETomek để thực hiện Oversampling. SMOTETomek sẽ tạo ra các mẫu nhân tạo cho lớp thiểu số và loại bỏ các mẫu gần lớp thiểu số nhưng thuộc về lớp đa số.
      * Kết quả cuối cùng là một dataframe mới (df\_new) chứa các mẫu đã được Oversampling, với số lượng mẫu trong mỗi nhóm đã được cân bằng.

### Các thuật toán so sánh

#### Thuật toán Machine Learning

##### SVM: Support Vector Machine

Thuật toán SVM (Support Vector Machine) là một phương pháp học máy được sử dụng cho cả hai mục đích phân loại và hồi quy.

* **SVM Phân loại (SVM Classification):**

Trong bài toán phân loại, mục tiêu của SVM là tìm ra một ranh giới phân chia tốt nhất giữa các nhóm dữ liệu. SVM cố gắng tìm ra một siêu mặt phẳng (đối với dữ liệu hai chiều) hoặc siêu siêu mặt phẳng (đối với dữ liệu nhiều chiều) để phân chia các điểm dữ liệu thành các nhóm riêng biệt.

* **SVM Hồi quy (SVM Regression):**

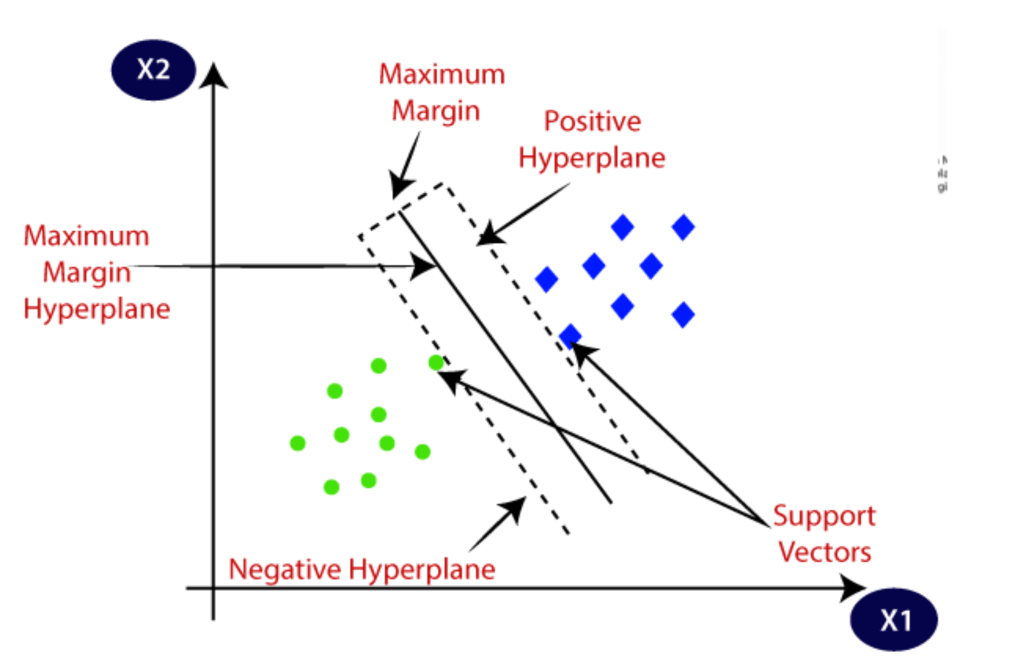
Trong bài toán hồi quy, SVM cố gắng xây dựng một hàm dự đoán để dự đoán giá trị liên tục dựa trên dữ liệu đầu vào. Mục tiêu của SVM hồi quy là tìm ra một siêu mặt phẳng (hoặc siêu mặt phẳng trong không gian nhiều chiều) sao cho sai số dự đoán của các điểm dữ liệu so với giá trị thực tế là nhỏ nhất.

Ngoài ra, SVM cũng có thể được phân loại dựa trên loại kernel được sử dụng:

* **SVM Tuyến tính (****Linear SVM):**

Trong SVM tuyến tính, ranh giới phân loại là một siêu mặt phẳng tuyến tính, tức là một đường thẳng trong không gian hai chiều hoặc một siêu mặt phẳng trong không gian nhiều chiều.

Hyper-plane nó chỉ hiểu đơn giản là 1 đường thẳng có thể phân chia các lớp ra thành hai phần riêng biệt.

 Ảnh có chứa biểu đồ, thiết kế, nghệ thuật gấp giấy origami

Mô tả được tạo tự động

Hình 4‑1. Minh hoạ Linear SVM và Non-linear SVM

* **SVM Phi tuyến tính (Non-linear SVM):**

Trong SVM phi tuyến tính, để xử lý dữ liệu không phân chia tuyến tính, chúng ta sử dụng các hạng mục (kernel) để ánh xạ dữ liệu vào một không gian cao hơn, nơi một ranh giới tuyến tính có thể được áp dụng. Các kernel phổ biến bao gồm hạng mục đa thức (polynomial kernel), hạng mục Radial Basis Function (RBF kernel) và hạng mục sigmoid.

##### KNN: K-Nearest Neighbors

K-Nearest Neighbors (KNN) là một thuật toán học máy có giám sát, đơn giản và dễ triển khai. Nó thường được dùng trong các bài toán phân loại và hồi quy.

Ý tưởng chính của thuật toán KNN là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.

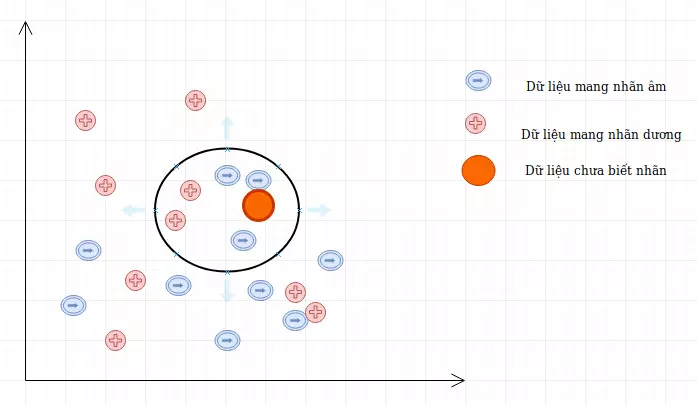
*Các bước chính của thuật toán KNN:*

Bước 1: Xác định giá trị K (số lượng láng giềng gần nhất).

Bước 2: Tính khoảng cách từ điểm dữ liệu mới tới tất cả các điểm trong tập dữ liệu huấn luyện.

Bước 3: Chọn K điểm dữ liệu trong tập huấn luyện có khoảng cách nhỏ nhất tới điểm dữ liệu mới.

Bước 4: Dự đoán nhãn của điểm dữ liệu mới bằng cách sử dụng phương pháp bầu chọn đa số từ K điểm dữ liệu gần nhất.



Hình 4‑2. Minh họa thuật toán KNN

##### NB: Naive Bayes

Thuật toán Naïve Bayes là một thuật toán phân lớp xác suất sử dụng định lý Bayes. Định lý Bayes cho biết rằng xác suất của một sự kiện xảy ra phụ thuộc vào xác suất của sự kiện đó xảy ra trong một số điều kiện nhất định [6].

Định lý Bayes được định nghĩa như sau:

A black and white math equation

Description automatically generated with medium confidence

Trong đó,  tần suất của A xảy ra dựa trên điều kiện là B đã xảy ra, viết tắt là P(A|B) , khi mà chúng ta biết tần suất B xảy ra với điều kiện A đã xảy ra P(B|A), xác suất A xảy ra P(A), B xảy ra P(B).

Thuật toán Naïve Bayes được sử dụng trong việc phân lớp các vấn đề về phân loại như phân loại tài liệu, phân loại thư rác và chẩn đoán bệnh một cách hiệu quả.

#### Thuật toán Deep Learning

##### LSTM: Long Short-Term Memory

Long Short-Term Memory (LSTM) là một loại RNN đặc biệt có thêm tính năng ghi nhớ chuỗi dữ liệu. Việc ghi nhớ xu hướng trước đó của dữ liệu có thể thực hiện được thông qua một số cổng cùng với dòng bộ nhớ được tích hợp trong LSTM điển hình. [7]

* Một lớp LSTM được tạo thành từ một tập hợp các khối bộ nhớ được kết nối thường xuyên.
* Mỗi khối chứa một hoặc nhiều ô nhớ được kết nối thường xuyên thông qua ba đơn vị nhân - cổng đầu vào, đầu ra và cổng quên.
* Cung cấp các hoạt động tương tự liên tục của các hoạt động ghi, đọc và đặt lại của ô nhớ.
* Giảm thiểu nhược điểm của việc biến mất độ dốc bằng cách cho phép thông tin truyền trực tiếp hơn qua trạng thái ô nhớ.

*Cấu trúc của LSTM:*

*A diagram of a flowchart

Description automatically generated*

Hình 4‑3. Cấu trúc của LSTM

* LSTM được cấu tạo bởi các cổng, các cổng trả về các giá trị từ “0” đến “1”. Với “0” có nghĩa là cổng sẽ chặn tất cả mọi giá trị, “1” có nghĩa là cho phép mọi giá trị đi qua cổng.
* LSTM gồm các cổng: Input Gate, Forget Gate, Output Gate.

Phương trình các cổng của LSTM:

**Input gate:**

**Forget gate:**

**Output gate:**

Trong đó:

: phương trình sigmoid

: trọng số cho các nơ-ron cổng (x) tương ứng

: đầu ra của khối lstm trước đó (tại dấu thời gian t − 1)

: đầu vào ở dấu thời gian hiện tại

: độ lệch cho các cổng tương ứng (x)

Ý nghĩa:

* Input Gate cho chúng ta biết thông tin mới nào chúng ta sẽ lưu trữ ở ô trạng thái
* Forget Gate thông báo loại bỏ thông tin khỏi ô trạng thái.
* Output Gate được sử dụng để cung cấp kích hoạt cho đầu ra cuối cùng của khối LSTM tại timestamp “t”.

Các phương trình cho ô trạng thái, ô trạng thái ứng cử viên và đầu ra cuối cùng:

**Temporary cell state:**

**Current cell state:**

**Final Output:**

Trong đó:

: ô trạng thái tại timestamp(t)

: ứng viên cho ô trạng thái tại timestamp(t)

Ý nghĩa:

* Ở bất kỳ timestamp nào, ô trạng thái biết rằng nó cần quên những gì ở trạng thái trước đó và những gì nó cần xem xét từ timestamp hiện tại
* Cuối cùng, lọc ô trạng thái và sau đó nó được chuyển qua hàm kích hoạt để dự đoán phần nào sẽ xuất hiện dưới dạng đầu ra của đơn vị LSTM hiện tại tại timestamp t.

##### CNN: Convolutional Neural Network

Convolutional Neural Network (CNN), còn được gọi là ConvNet, là một loại mạng nơ-ron thích hợp cho việc xử lý và phân tích dữ liệu có cấu trúc không gian như hình ảnh và video. CNN đã đạt được sự nổi tiếng lớn trong lĩnh vực thị giác máy tính và các bài toán liên quan đến xử lý ảnh. CNN hoạt động bằng cách sử dụng các lớp tích chập để trích xuất các đặc điểm từ hình ảnh đầu vào, sau đó sử dụng các lớp kết nối đầy đủ để phân loại các đặc điểm đó thành các lớp.



Hình 4‑4. Cấu trúc cơ bản của CNN

*Cấu trúc cơ bản của CNN:*

Lớp Convolutional (Convolutional Layer): Lớp này thực hiện các phép tích chập trên dữ liệu đầu vào, giúp phát hiện các đặc trưng cục bộ như cạnh, đường cong và mẫu trong hình ảnh.

Lớp Pooling (Pooling Layer): Lớp này thực hiện giảm kích thước không gian của dữ liệu đầu vào bằng cách chọn giá trị đại diện từ các vùng của dữ liệu. Phổ biến nhất là lớp max pooling.

Lớp Fully Connected (Fully Connected Layer): Lớp này được sử dụng để kết nối các đặc trưng được trích xuất từ lớp convolutional và pooling với các đầu ra dự đoán.

Lớp Activation (Activation Layer): Lớp này áp dụng các hàm kích hoạt như ReLU (Rectified Linear Unit) để thêm tính phi tuyến tính vào mô hình, giúp nó học được các biểu diễn phức tạp hơn.

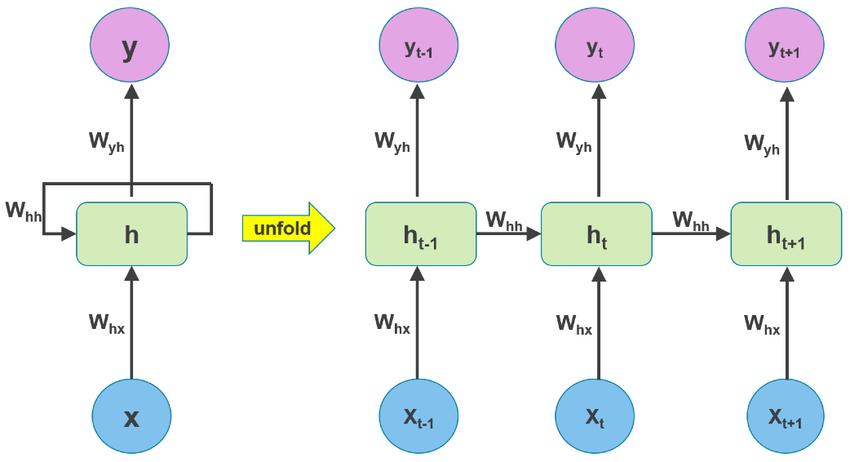
A graph of a function

Description automatically generated

Hình 4‑5. Hàm ReLU

##### RNN: Recurrent Neural Network

Recurrent Neural Network (RNN) là một loại mạng nơ-ron nhân tạo được thiết kế để xử lý và mô hình dữ liệu có cấu trúc chuỗi, trong đó thông tin từ các bước thời gian trước đó có thể ảnh hưởng đến quá trình hiện tại. RNN có khả năng tự nắm bắt và sử dụng thông tin liên quan từ tất cả các phần tử trong chuỗi đầu vào, làm cho chúng trở thành một công cụ mạnh mẽ cho việc dự đoán và phân tích dữ liệu chuỗi như ngôn ngữ tự nhiên, chuỗi thời gian và nhiều ứng dụng khác.



Hình 4‑6. Sơ đồ mô hình RNN

*RNN bao gồm các thành phần chính sau:*

Đơn vị RNN (RNN Cell): Là một module tính toán được lặp lại trong mỗi bước thời gian của chuỗi đầu vào. Các đơn vị này duy trì một trạng thái ẩn và xử lý dữ liệu đầu vào tại từng thời điểm.

Trạng thái ẩn (Hidden State): Là một vector dài ℎ lưu trữ thông tin về chuỗi đầu vào từ tất cả các bước thời gian trước đó. Trạng thái ẩn được cập nhật trong mỗi bước thời gian và chứa thông tin cần thiết để dự đoán đầu ra tại bước thời gian hiện tại.

Đầu vào (Input): Là dữ liệu đầu vào tại mỗi bước thời gian trong chuỗi. Đầu vào có thể là các vector biểu diễn cho các từ trong ngôn ngữ tự nhiên hoặc các đặc trưng của dữ liệu chuỗi khác.

Đầu ra (Output): Là dự đoán hoặc biểu diễn của chuỗi đầu vào tại mỗi bước thời gian. Đầu ra có thể là một vector, một số hoặc một phân phối xác suất.

Trạng thái ẩn tại bước thời gian t:

* là trạng thái ẩn tại bước thời gian 𝑡
* là đầu vào tại bước thời gian 𝑡
* và là ma trận trọng số cho trạng thái ẩn và đầu vào tương ứng
* là vectơ bias
* là hàm kích hoạt

Đầu ra tại bước thời gian t:

* là đầu ra tại bước thời gian 𝑡
* là ma trận trọng số đầu ra
* là trạng thái ẩn tại bước thời gian 𝑡
* là vectơ bias
* là hàm kích hoạt

### Thông số đánh giá

TP (True Positive): Số lượng dự đoán đúng với nhãn dương tính.

TN (True Negative): Số lượng dự đoán đúng với nhãn âm tính.

FP (False Positive): Số lượng dự đoán sai với nhãn dương tính (false alarm).

FN (False Negative): Số lượng dự đoán sai với nhãn âm tính (missed detection).

#### Accuracy

Accuracy đo lường tỷ lệ các dự đoán chính xác so với tổng số mẫu trong tập dữ liệu.

Công thức:

#### Precision

Precision (Độ chính xác): Đo lường tỷ lệ của các dự đoán thuộc lớp dương tính thực sự so với tổng số dự đoán thuộc lớp dương tính.

#### Recall

Recall (Độ nhạy): Đo lường tỷ lệ của các dự đoán thuộc lớp dương tính so với tổng số mẫu thuộc lớp dương tính trong tập dữ liệu.

#### F- score

F-score: Cung cấp một chỉ số duy nhất cân nhắc cả hai yếu tố của Precision và Recall. F-score nằm trong khoảng từ 0 đến 1, với 1 là tốt nhất. Giá trị beta quyết định mức độ ưu tiên của Recall so với Precision trong F-score.

Ảnh có chứa Phông chữ, văn bản, màu trắng, hàng

Mô tả được tạo tự động

*F\_β = (1 + β^2) \* (Precision \* Recall) / (β^2 \* Precision + Recall)*

Khi β = 1, F-score trở thành trung bình nhân của Precision và Recall.

Khi β > 1, F-score nhấn mạnh nhiều hơn vào Recall.

Khi β < 1, F-score nhấn mạnh nhiều hơn vào Precision.

Việc lựa chọn thông số nào để đánh giá mô hình phụ thuộc vào mục đích và yêu cầu của bài toán cụ thể. Trong nhiều trường hợp, Precision và Recall đóng vai trò quan trọng hơn Accuracy vì chúng cung cấp thông tin chi tiết hơn về hiệu suất của mô hình trên từng lớp.

## THỰC NGHIỆM & KẾT QUẢ

### Thực nghiệm 1

Dựa trên dữ liệu bệnh của bệnh nhân, tác giả áp dụng mô hình học sâu BiLSTM để tạo ra dự đoán về viêm gan. Thực nghiệm 1 đã được trả lời bằng cách điều chỉnh các tham số của một trong những mô hình phổ biến nhất cho việc dự đoán khả năng sống sót của bệnh nhân viêm gan, thuật toán BiLSTM. Nhiều epoch và kỹ thuật lọc khác nhau cũng đã được sử dụng. Một số kích thước lô và epoch khác nhau được bao gồm trong ba lớp ẩn của thuật toán. Kết quả sử dụng hàm kích hoạt SoftMax đã tạo ra 62 vector từ vựng, và kích thước nhúng là 128 (8, 16 và 32). Một số mô hình BiLSTM được chọn lọc về độ chính xác, độ thu hồi và điểm F được hiển thị trong Hình 5-1. Bộ lọc này có độ chính xác 93% khi được sử dụng với các tham số sau: kích thước bộ lọc là 280, kích thước đơn vị là 2, "điểm F1" là 94%, độ thu hồi là 94%, và độ chính xác là 94%. Bộ lọc số 8 có độ chính xác là 95,08%. Kết quả của 10 thí nghiệm BiLSTM với các giá trị tham số biến thiên được trình bày trong Bảng 2. Tác giả đã so sánh độ chính xác của mỗi mô hình với nhau. Với tám kích thước bộ lọc, mười sáu bộ lọc và mười đơn vị LSTM, nó có độ chính xác cao nhất so với tất cả các mô hình được thử nghiệm.

A graph of different colored bars

Description automatically generated with medium confidence

Hình 5‑1. Kết quả 10 thí nghiệm BiLSTM

### Thực nghiệm 2

Phân loại dữ liệu không cân bằng, phân chia huấn luyện ngẫu nhiên và tập kiểm tra đã tạo ra kết quả lớp không đều. Điều này không cân bằng vì có 123 bệnh nhân sống sót và 32 bệnh nhân tử vong. Để làm cân bằng tập dữ liệu, tác giả đã sử dụng phương pháp làm mất cân bằng ngẫu nhiên. Như được thể hiện trong Hình 5-2, khi sử dụng cân bằng dữ liệu thay vì không sử dụng cân bằng dữ liệu, hiệu suất tăng đáng kể. Bằng cách phân tích các giá trị kinh nghiệm, có thể thấy rằng mô hình được đề xuất có thể được sử dụng để dự đoán khả năng sống sót của bệnh nhân viêm gan trong các tình huống thực tế.

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Mô tả được tạo tự động

Hình 5‑2. Kết quả so sánh giữa dữ liệu cân bằng và mất cân bằng

### Thực nghiệm 3

Thí nghiệm này được thực hiện để so sánh mô hình BiLSTM trong việc dự đoán sự sống sót của bệnh nhân viêm gan với học máy truyền thống và học sâu. Các thuật toán học máy truyền thống và mạng nơ-ron sâu đã được so sánh để xem kết quả của BiLSTM trong việc dự đoán bệnh tiểu đường như thế nào.

Bảng 3. Trình phân loại dựa trên học máy (ML) so với mô hình được đề xuất (BiLSTM)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ML model** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F-score (%)** |
| SVM | 91.3 | 100 | 91.3 | 95.4 |
| KNN | 73.9 | 100 | 73.9 | 85.0 |
| NB | 85.1 | 82.1 | 92.0 | 86.7 |
| Proposed  (BiLSTM) | 95.2 | 95.5 | 95.5 | 95.5 |

Bảng 4. BiLSTM so với các mô hình học sâu (DL) khác.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DL model** | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F-score (%)** |
| LSTM | 91.4 | 92.0 | 92.0 | 92.0 |
| CNN | 91.3 | 100 | 84.6 | 91.6 |
| RNN | 82.6 | 80.0 | 92.3 | 85.7 |
| Proposed  (BiLSTM) | 95.2 | 95.5 | 95.5 | 95.5 |

### Thực nghiệm 4

Thí nghiệm này so sánh kết quả của bài báo với các công trình của các bài báo dự đoán viêm gan khác:

*Công trình của Kashif et al.:* Kashif et al. đã tạo ra một phương pháp dự đoán bệnh viêm gan dựa trên học máy. Học máy được sử dụng để dự đoán viêm gan tốt hơn bằng cách sử dụng rừng ngẫu nhiên (RF), máy vector hỗ trợ (SVM) và các bộ phân loại khác.

*Công trình của Akbar et al.:* Dự đoán viêm gan sử dụng học máy đã được đề xuất. Tập dữ liệu về viêm gan đã được phân tích bằng cách sử dụng một loạt các phương pháp học máy. Hiệu suất của một mô hình học sâu có thể được cải thiện bằng cách sử dụng một phương pháp cân bằng dữ liệu hiệu quả hơn.

*Công trình đề xuất (mô hình của tác giả):* Phương pháp dự đoán viêm gan dựa trên học sâu sử dụng một mạng nơ-ron sâu. Hình 5-3 cho thấy các biến dự đoán (Hình 5-3) được chọn có ảnh hưởng đáng kể đến tham số dự đoán (mục tiêu). Sự kết hợp giữa cân bằng dữ liệu và mô hình học sâu BiLSTM là yếu tố quan trọng nhất trong thành công của tác giả trong việc dự đoán các bệnh viêm gan. Bằng cách sử dụng lớp BiLSTM, thông tin ngữ cảnh có thể được giữ lại.

A graph with blue bars

Description automatically generated

Hình 5‑3. So sánh các nghiên cứu khác và mô hình BiLSTM

## KẾT LUẬN

### Phân tích kết quả

Thí nghiệm thực hiện so sánh giữa dự đoán của các chuyên gia và dự báo được cung cấp bởi chiến lược đề xuất, và hiệu suất của phương pháp đề xuất được đánh giá. Các quy trình cho 11 bệnh nhân đầu tiên được mô tả trong Bảng 5.

Bảng 5. Chuyên gia về con người so với tiên lượng hệ thống được đề xuất

|  |  |  |
| --- | --- | --- |
| **Suspected Hepatitis Patient** | **Diagnosis by**  **hepatitis expert** | **Prediction by BiLSTM (proposed)** |
| 1 | Hepatitis\_patient\_ survivability = yes | Hepatitis\_patient\_ survivability = yes |
| 2 | Hepatitis\_patient\_ survivability = yes | Hepatitis\_patient\_ survivability = yes |
| 3 | Hepatitis\_patient\_ survivability = no | Hepatitis\_patient\_ survivability = no |
| 4 | Hepatitis\_patient\_ survivability = yes | Hepatitis\_patient\_ survivability = no |
| 5 | Hepatitis\_patient\_ survivability = yes | Hepatitis\_patient\_ survivability = yes |
| 6 | Hepatitis\_patient\_ survivability = yes | Hepatitis\_patient\_ survivability = yes |
| 7 | Hepatitis\_patient\_ survivability = yes | Hepatitis\_patient\_ survivability = yes |
| 8 | Hepatitis\_patient\_ survivability = no | Hepatitis\_patient\_ survivability = no |
| 9 | Hepatitis\_patient\_ survivability = yes | Hepatitis\_patient\_ survivability = yes |
| 10 | Hepatitis\_patient\_ survivability = no | Hepatitis\_patient\_ survivability = no |
| 11 | Hepatitis\_patient\_ survivability = yes | Hepatitis\_patient\_ survivability = yes |

### Hướng phát triền

Với sự mở rộng đáng kể trong nội dung chăm sóc sức khỏe, việc thu thập và phân tích dữ liệu chăm sóc sức khỏe để phát hiện bệnh viêm gan ở bệnh nhân trở nên quan trọng hơn bao giờ hết. Một mô hình DSS thành công dựa trên học sâu đã được tạo ra và được thực hiện để đạt được điều này. Một mạng nơ-ron sâu (BiLSTM) được sử dụng để dự đoán viêm gan dựa trên việc thu thập dữ liệu, tiền xử lý và xác thực. Các thí nghiệm bổ sung đã được tiến hành bằng cách sử dụng cả tập dữ liệu cân bằng và không cân bằng. Một mô hình BiLSTM cũng đã được sử dụng để dự đoán khả năng mắc viêm gan trong tương lai. Kết quả rất khích lệ khi so sánh với các nỗ lực trước đó. Sự phụ thuộc của chiến lược đề xuất vào việc nhúng thay vì trên một mô hình được huấn luyện trước là một hạn chế rõ ràng. Trong tương lai, việc sử dụng các thuật toán được huấn luyện trước như Word2vec hoặc Fasttext với các tập dữ liệu viêm gan từ các lĩnh vực khác nhau (ví dụ, dữ liệu bệnh nhân từ các lĩnh vực khác nhau) có thể được xem xét.

# THAM KHẢO

[1]F. R. Albogamy et al., “Decision Support System for Predicting Survivability of Hepatitis Patients,” Front. Public Health, vol. 10, Apr. 2022, doi: 10.3389/fpubh.2022.862497. Available: <https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2022.862497/full>. [Accessed: Apr. 18, 2024]

[2]“HepatitisData.”Available: <https://www.kaggle.com/datasets/codebreaker619/hepatitis-data>. [Accessed: Apr. 18, 2024]

[3] “[SVM] Thuật toán hỗ trợ máy vec-tơ Support vector machines,” Hỗ trợ nghiên cứu định lượng. Accessed: Apr. 29, 2024. [Online]. Available: <https://luanvanhay.org/lam-thue/svm-thuat-toan-ho-tro-may-vec-to-support-vector-machines/>

[4] Xiang Fan, Phuong Truong, “An Introduction to Convolutional Neural Network (CNN)”, Feb. 2022, [Online]. Available: <https://medium.com/sfu-cspmp/an-introduction-to-convolutional-neural-network-cnn-207cdb53db97> . [Accessed: Apr. 30, 2024]

[5] “KNN (K-Nearest Neighbors) #1.” Accessed: Apr. 29, 2024. [Online]. Available: <https://viblo.asia/p/knn-k-nearest-neighbors-1-djeZ14ejKWz>

[6] Vikramkumar, V. B, and Trilochan, “Bayes and Naive Bayes Classifier.” arXiv, Apr. 03, 2014. doi: [10.48550/arXiv.1404.0933](https://doi.org/10.48550/arXiv.1404.0933).

[7] “What is LSTM? Introduction to Long Short-Term Memory | by Rebeen Hamad | Medium.” Accessed: May 02, 2024. [Online]. Available: <https://medium.com/@rebeen.jaff/what-is-lstm-introduction-to-long-short-term-memory-66bd3855b9ce>

# BẢNG PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nội dung công việc | | Thanh Sơn | Thu Thủy | Cẩm Tú | Thanh Nhựt |
| Lựa chọn đề tài bài báo | | X |  |  | X |
| Tìm dataset liên quan | |  | X |  |  |
| Word | Chương 1 |  | X | X |  |
| Chương 2 |  |  | X |  |
| Chương 3 |  |  | X |  |
| Chương 4 | X | X |  | X |
| Chương 5 | X | X | X | X |
| Chương 6 |  |  | X |  |
| Code | Tiền xử lý dữ liệu  + BiLSTM |  |  | X |  |
| SVM + KNN |  | X |  |  |
| NB + LSTM |  |  |  | X |
| CNN + RNN | X |  |  |  |
| Tổng hợp bài code | X |  |  |  |
| Giao diện người dùng |  |  |  | X |
| Slide | | X | X | X | X |
| Đánh giá | | 99% | 99% | 99% | 99% |