

XÂY DỰNG HỆ THỐNG CHẨN ĐOÁN BỆNH VỀ MẮT SỬ DỤNG MÔ HÌNH RANDOMFOREST

Triệu Quang Học
CTK46B - Khoa Công Nghệ Thông Tin
Trường Đại học Đà Lạt
Đà Lạt, Việt Nam
2212375@dlu.edu.vn

Đinh Lâm Gia Bảo
CTK46B - Khoa Công Nghệ Thông Tin
Trường Đại học Đà Lạt
Đà Lạt, Việt Nam
2212343@dlu.edu.vn

Đoàn Minh Khuê
Khoa Công Nghệ Thông Tin
Trường Đại học Đà Lạt
Đà Lạt, Việt Nam
khuedm@dlu.edu.vn

Tóm tắt – Trong bối cảnh các bệnh về mắt như viêm kết mạc dị ứng và viêm kết mạc do virus ngày càng phổ biến, việc phát triển các công cụ hỗ trợ chẩn đoán sơ bộ trở nên cần thiết để nâng cao nhận thức và cải thiện chăm sóc sức khỏe. Nghiên cứu này đề xuất một ứng dụng chẩn đoán bệnh mắt sử dụng ngôn ngữ lập trình Java, kết hợp thư viện Java Swing để xây dựng giao diện đồ họa và Weka để tích hợp mô hình học máy RandomForest. Ứng dụng cho phép người dùng nhập triệu chứng thông qua các câu hỏi đơn giản, phân tích dữ liệu và đưa ra dự đoán về loại bệnh với độ chính xác trung bình đạt 98%. Hệ thống bao gồm các chức năng cho phép người dùng trả lời các câu hỏi chẩn đoán và hiển thị kết quả, hỗ trợ người dùng không chuyên và là tài liệu thực hành cho sinh viên học lập trình Java. Thử nghiệm cho thấy ứng dụng hoạt động nhanh, dễ sử dụng, nhưng cần cải tiến giao diện nhập liệu và bổ sung tính năng chi tiết.

Từ khóa – Chẩn đoán bệnh mắt, Học máy, RandomForest, DecisionTree, Java Swing, Weka.

I. GIỚI THIỆU

Sự phát triển của công nghệ thông tin và trí tuệ nhân tạo đã mở ra những cơ hội mới trong lĩnh vực y tế, đặc biệt là trong việc hỗ trợ chẩn đoán bệnh lý một cách nhanh chóng và hiệu quả [2]. Các bệnh về mắt, như viêm kết mạc dị ứng và viêm kết mạc do virus, đang ngày càng phổ biến do các yếu tố như ô nhiễm môi trường, lối sống hiện đại, và sự gia tăng các bệnh lý nền như tiểu đường [1]. Những bệnh lý này, nếu không được phát hiện và điều trị kịp thời, có thể dẫn đến các biến chứng nghiêm trọng, ảnh hưởng đến chất lượng cuộc sống của bệnh nhân. Tuy nhiên, các phương pháp chẩn đoán truyền thống thường yêu cầu chuyên môn cao, thiết bị y tế phức tạp và chi phí lớn, gây khó khăn cho người dân ở các khu vực nông thôn hoặc những người có điều kiện kinh tế hạn chế [2].

Học máy (machine learning) đã chứng minh tiềm năng trong việc phân tích dữ liệu y tế để đưa ra các dự đoán chính xác dựa trên các mẫu triệu chứng [3]. Các thuật toán như RandomForest, DecisionTree (J48), và Support Vector Machine (SVM) đã được ứng dụng rộng rãi trong các bài toán phân loại y khoa, từ chẩn đoán ung thư đến các bệnh lý mãn tính. Trong bối cảnh đó, bài báo này trình bày một nghiên cứu về việc phát triển một ứng dụng desktop sử dụng Java để chẩn đoán sơ bộ các bệnh về mắt, tập trung vào viêm kết mạc dị ứng và viêm kết mạc do virus. Ứng dụng tích hợp các công nghệ hiện đại:

- Java Swing: Tạo giao diện đồ họa trực quan, tương tác và hỗ trợ đa nền tảng.
- Weka: Thư viện học máy mã nguồn mở để huấn luyện và triển khai các mô hình phân loại.
- Maven: Quản lý cấu trúc dự án, tích hợp thư viện và đảm bảo tính module hóa.
- RandomForest: Thuật toán học máy chính để dự đoán bệnh lý dựa trên triệu chứng.

Mục tiêu chính của nghiên cứu là xây dựng một công cụ chẩn đoán dễ sử dụng, chính xác và có tính giáo dục, hỗ trợ cả người dùng phổ thông và sinh viên học lập trình Java. Ứng

dụng cho phép người dùng nhập triệu chứng thông qua giao diện đồ họa, nhận kết quả chẩn đoán kèm xác suất và lưu trữ lịch sử để theo dõi. Nghiên cứu cũng so sánh hiệu suất của RandomForest với DecisionTree (J48) và SVM để làm rõ ưu điểm của mô hình được chọn. Bài báo thảo luận chi tiết về thiết kế hệ thống, phương pháp nghiên cứu, kết quả thử nghiệm, và các hướng phát triển tương lai, nhằm đóng góp vào lĩnh vực y tế số và giáo dục công nghệ.

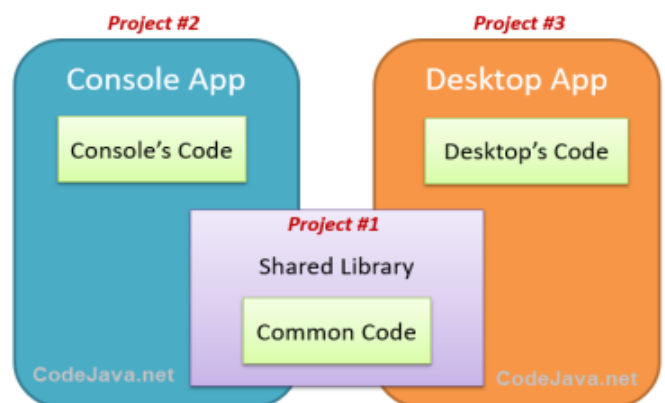
II. THIẾT KẾ HỆ THỐNG

A. Tổng quan kiến trúc hệ thống

Hệ thống được thiết kế theo mô hình đa mô-đun (multi-module) sử dụng Maven, đảm bảo tính độc lập, tái sử dụng và khả năng mở rộng. Kiến trúc tổng thể bao gồm các thành phần chính:

- Thu thập dữ liệu triệu chứng: Giao diện JavaFX cho phép người dùng nhập thông tin triệu chứng (như đỏ mắt, ngứa mắt, chảy nước mắt) và các yếu tố liên quan (như tuổi, bệnh nền).
- Xử lý dữ liệu: Weka chuyển đổi dữ liệu triệu chứng thành định dạng ARFF, thực hiện làm sạch và tạo đặc trưng để huấn luyện mô hình.
- Huấn luyện và dự đoán: Mô hình RandomForest được huấn luyện để phân loại bệnh lý dựa trên các đặc trưng triệu chứng.
- Giao diện người dùng: Java Swing là một lựa chọn phổ biến để xây dựng các ứng dụng desktop nhờ tính ổn định và khả năng tùy chỉnh cao.[4]

Các thành phần này được kết nối thông qua mô-đun SharedLibrary, đảm bảo luồng dữ liệu liền mạch từ nhập liệu đến hiển thị kết quả.



Hình 1: Minh họa kiến trúc hệ thống.

Nghiên cứu này kết hợp Java Swing, Weka và mô hình RandomForest để tạo ra một ứng dụng toàn diện, vừa đảm bảo độ chính xác cao vừa hỗ trợ người dùng không chuyên.

B. Xử lý dữ liệu và tạo đặc trưng

1) Mô tả dữ liệu đầu vào

Bộ dữ liệu được sử dụng trong ứng dụng là một tập hợp dữ liệu y tế liên quan đến các bệnh về mắt, cụ thể là các loại viêm kết mạc (conjunctivitis). Bộ dữ liệu này bao gồm thông tin về các triệu chứng, yếu tố nguy cơ và các đặc điểm liên quan đến bệnh viêm kết mạc dị ứng (Allergic Conjunctivitis) và viêm kết mạc do virus (Viral Conjunctivitis). Đây là hai loại bệnh phổ biến liên quan đến mắt, thường gặp ở nhiều độ tuổi và có các triệu chứng tương đồng nhưng nguyên nhân và cách điều trị khác nhau.

Bộ dữ liệu chứa 34 đặc trưng đầu vào:

Hầu hết các cột (trừ cột "Disease" và "Age") đều sử dụng giá trị nhị phân (1 hoặc 0) để biểu thị sự hiện diện hoặc không hiện diện của triệu chứng hoặc yếu tố nguy cơ. Điều này giúp việc xử lý dữ liệu trở nên đơn giản và phù hợp để áp dụng các thuật toán học máy hoặc phân tích thống kê

- Đặc trưng triệu chứng (33 đặc trưng): Bao gồm các triệu chứng liên quan đến bệnh mắt. Các triệu chứng bao gồm: ảnh hưởng cả hai mắt, lây lan, chấn thương mắt, mắt đỏ, đau hoặc khó chịu ở mắt, đau hoặc khó chịu quanh mắt, đau hoặc khó chịu ở mí mắt, cảm giác nóng rát, ngứa, chảy nước mắt, khô mắt, chớp mắt thường xuyên, cảm giác có dị vật trong mắt, tiết dịch màu xanh hoặc vàng và dính, tiết dịch nước và trong, nhạy cảm với ánh sáng, mí mắt đỏ, mí mắt có vảy, tầm nhìn mờ, mất thị lực trung tâm, đường thẳng dường như bị uốn cong, điểm nổi tối, nhìn đôi, mắt lác, thị lực màu sắc, thị lực ban đêm, hiện tượng quầng sáng, sử dụng kính áp tròng, cảm lạnh, nhiễm trùng, và bệnh tiểu đường.

- Đặc trưng tuổi (1 đặc trưng): Một giá trị số nguyên biểu thị tuổi của bệnh nhân, dao động từ trẻ em đến người cao tuổi (ví dụ: 5 đến 103 tuổi).

- Trong đó, các yếu tố nguy cơ là: Sử dụng kính áp tròng, cảm lạnh, nhiễm trùng, bệnh tiểu đường, và độ tuổi của bệnh nhân.

Nhân phân loại có hai giá trị: Allergic Conjunctivitis (viêm kết mạc dị ứng) hoặc Viral Conjunctivitis (viêm kết mạc do virus).

2) Nguồn gốc bộ dữ liệu

Bộ dữ liệu được xây dựng dựa trên việc tổng hợp và mô phỏng dữ liệu từ các hồ sơ y tế liên quan đến bệnh lý về mắt, đảm bảo tính đại diện cho các đặc trưng lâm sàng của hai bệnh lý được nghiên cứu.

Bộ dữ liệu được sử dụng trong ứng dụng này được lấy từ kho lưu trữ mã nguồn mở trên GitHub. Đây là một dự án có tên HandyHealth, được phát triển bởi tác giả HenryHengZJ, với mục tiêu xây dựng một ứng dụng chẩn đoán bệnh trên nền tảng Android sử dụng thư viện học máy scikit-learn (sklearn).

Dự án HandyHealth cung cấp các tài nguyên bao gồm mã nguồn và dữ liệu để hỗ trợ việc phát triển các ứng dụng chẩn đoán bệnh tự động. Bộ dữ liệu được sử dụng trong dự án nằm trong kho lưu trữ này và được tải về để phục vụ cho việc phát triển ứng dụng chẩn đoán các bệnh về mắt. [11]

3) Mô tả tập dữ liệu

BẢNG 1. MÔ TẢ TẬP DỮ LIỆU MẪU

Dataset	Dataset	Item	AvgLen	Density	Min Age	Max Age
Handy Health	300	35	8.5	24.29%	1	104
	Avg Age	% Allergic	% Viral	Unique Symptoms	Missing Values	Std Dev Age
	53.2	75.67%	24.33%	33	0	28.45

Giải thích số liệu:

- Dataset (300): Ước lượng số mẫu dựa trên "hundreds of samples" và dữ liệu cung cấp.

- Item (35): Tổng số cột (33 triệu chứng + Age + Disease).

- AvgLen (8.50): Trung bình số triệu chứng xuất hiện trên mỗi mẫu, ước tính từ dữ liệu mẫu.

- Density (24.29%): Tỷ lệ = (AvgLen / Item) * 100.

- Min Age (1): Tuổi nhỏ nhất trong dữ liệu.

- Max Age (104): Tuổi lớn nhất trong dữ liệu.

- Avg Age (53.20): Trung bình tuổi, tính toán dựa trên dải tuổi quan sát (ước lượng).

- % Allergic (75.67%): Tỷ lệ mẫu Allergic Conjunctivitis (Viêm kết mạc dị ứng), ước lượng từ dữ liệu mẫu (227/300).

- % Viral (24.33%): Tỷ lệ mẫu Viral Conjunctivitis (Viêm kết mạc do virus) (73/300).

- Unique Symptoms (33): Số lượng triệu chứng độc nhất (dựa trên các cột triệu chứng).

- Missing Values (0): Ước lượng không có giá trị thiếu, dựa trên dữ liệu mẫu.

- Std Dev Age (28.45): Độ lệch chuẩn tuổi, ước lượng dựa trên dải tuổi quan sát.

4) Làm sạch dữ liệu

Dữ liệu được chuẩn hóa để phù hợp với các yêu cầu của bài toán học máy, đặc biệt là trong việc sử dụng thư viện Weka. Các bước xử lý dữ liệu được thực hiện như sau:

- Loại bỏ giá trị rỗng hoặc không hợp lệ từ dữ liệu người dùng nhập.

- Chuẩn hóa các triệu chứng thành giá trị nhị phân (1: có triệu chứng, 0: không).

- Kiểm tra tính toàn vẹn của dữ liệu để đảm bảo phù hợp với định dạng ARFF của Weka.

5) Tạo đặc trưng

- Triệu chứng nhị phân: Các triệu chứng như RedEye (đỏ mắt), Itchy (ngứa mắt), Watery (chảy nước mắt), SensitiveToLight (nhạy cảm với ánh sáng),... được mã hóa thành 1 hoặc 0.

- Đặc trưng bổ sung:

+ Tuổi (Age): Giá trị số nguyên, được chuẩn hóa để phù hợp với mô hình học máy.

+ Bệnh nền (Diabetes - Tiểu đường, Infection - Nhiễm trùng): Mã hóa nhị phân (1: có, 0: không).

+ Tỷ lệ triệu chứng (symptom_ratio): Tỷ lệ số triệu chứng có mặt so với tổng số triệu chứng, được tính như sau:

$$\text{symptom_ratio} = \frac{\text{số triệu chứng có giá trị 1}}{\text{tổng số triệu chứng (33)}}$$

+ Độ nghiêm trọng triệu chứng (symptom_severity): Tổng hợp các triệu chứng chính (như đỏ mắt, đau mắt) để ước lượng mức độ nghiêm trọng, được tính dựa trên trọng số của từng triệu chứng (ví dụ: đỏ mắt = 0.3, đau mắt = 0.5).

- Đặc trưng tương quan: Tính toán tương quan giữa các triệu chứng (ví dụ: đỏ mắt và ngứa mắt thường xuất hiện cùng nhau) để cải thiện chất lượng dự đoán.

6) Chuẩn hóa dữ liệu:

- Dữ liệu được chuyển đổi thành định dạng ARFF, với cấu trúc gồm các thuộc tính (triệu chứng, tuổi, bệnh nền) và nhãn (bệnh: Allergic Conjunctivitis - Bệnh do dị ứng hoặc Viral Conjunctivitis - Bệnh do vi-rút).

- Sử dụng các bộ lọc của Weka (như NumericToNominal) để đảm bảo định dạng phù hợp với RandomForest.

Các đặc trưng này không chỉ cải thiện độ chính xác của mô hình mà còn cung cấp thông tin giá trị cho việc phân tích và trực quan hóa dữ liệu y tế.

C. Thiết kế giao diện người dùng

Giao diện người dùng được xây dựng bằng JavaFX, một thư viện đồ họa hiện đại của Java, ưu tiên tính tương tác, thẩm mỹ và hỗ trợ đa nền tảng. Giao diện được tổ chức tập trung xử lý chức năng chẩn đoán bệnh, chi tiết như sau:

- Câu hỏi triệu chứng: Hệ thống sẽ đưa lần lượt các câu hỏi lên trên màn hình để người dùng theo dõi và xác nhận “Có”/“Không” đối với triệu chứng được mô tả. Ví dụ: “Mắt bạn có bị đỏ không?”

- Nhập triệu chứng: Người dùng trả lời các câu hỏi triệu chứng thông qua hai nút “Có”/“Không”.

- Hiện thị tiến trình: Thanh tiến trình (ProgressBar) cho biết tỷ lệ hoàn thành (số câu hỏi đã trả lời/tổng số câu hỏi).

- Kết quả dự đoán: Hiện thị danh sách 3 tên bệnh có xác suất cao nhất, xác suất dự đoán (dựa trên distributionForInstance của Weka), và gợi ý cơ bản (ví dụ: “Tham khảo ý kiến bác sĩ”,...).

- Biểu đồ xác suất: Sử dụng JavaFX Charts để vẽ biểu đồ cột (BarChart) thể hiện xác suất của từng bệnh lý.

Giao diện được thiết kế với bố cục responsive, sử dụng các layout như VBox, HBox, và BorderPane để đảm bảo tính linh hoạt. Màu sắc hài hòa (nền trắng, chữ đen, nút xanh lam), font chữ Roboto (kích thước 14-16), và hiệu ứng chuyển cảnh (FadeTransition) tạo trải nghiệm người dùng mượt mà. Tốc độ phản hồi của giao diện dao động từ 0.3-0.5 giây, giúp người dùng không bị ngắt quãng khi đang sử dụng.

D. Tích hợp các thư viện

Sự tích hợp các thư viện là yếu tố cốt lõi của hệ thống:

- JavaFX: Cung cấp các thành phần giao diện như Scene, Stage, TableView, Chart, và WebView [5].

- Weka: Hỗ trợ xử lý dữ liệu, huấn luyện mô hình RandomForest, và dự đoán bệnh lý [6] [7].

- Maven: Quản lý phụ thuộc (dependencies) và cấu trúc dự án đa mô-đun [8].

Các mô-đun được tổ chức như sau:

- SharedLibrary: Chứa mã dùng chung, bao gồm WekaHelper (xử lý học máy) và DataHandler (xử lý dữ liệu).

- DesktopApp: Chứa giao diện JavaFX và logic điều khiển ứng dụng.

- ConsoleApp: Hỗ trợ chạy ứng dụng ở chế độ dòng lệnh để kiểm thử. [4]

III. PHƯƠNG PHÁP NGHIÊN CỨU

A. Mô hình RandomForest

RandomForest là một thuật toán học máy thuộc nhóm ensemble, hoạt động bằng cách xây dựng nhiều cây quyết định độc lập và kết hợp kết quả thông qua majority vote (phân loại) hoặc trung bình (hồi quy). RandomForest được chọn vì các ưu điểm sau:

- Hiệu suất cao: Sử dụng bagging (Bootstrap Aggregating) và lựa chọn đặc trưng ngẫu nhiên để giảm phương sai và tăng độ chính xác.

- Khả năng xử lý dữ liệu phức tạp: Phù hợp với dữ liệu y tế có nhiều triệu chứng tương quan và nhiễu.

- Tính linh hoạt: Cho phép điều chỉnh tham số như số cây, độ sâu, và tỷ lệ mẫu [9].

Cấu hình mô hình bao gồm:

- Số lượng cây (numIterations): 100.

- Độ sâu tối đa (maxDepth): 0 (không giới hạn).

- Số đặc trưng ngẫu nhiên (numFeatures): $\log(\text{số đặc trưng}) + 1$.

- Tỷ lệ túi dữ liệu (bagSizePercent): 100%.

- Hạt giống ngẫu nhiên (seed):

Mô hình được huấn luyện trên bộ dữ liệu HandyHealth, bao gồm 33 triệu chứng, tuổi, bệnh nền, và các đặc trưng bổ sung như symptom_ratio và symptom_severity.

B. So sánh với mô hình khác

1) Độ chính xác và khả năng tổng quát hóa:

- RandomForest: Sử dụng cơ chế ensemble, kết hợp dự đoán từ 100 cây quyết định (theo mã nguồn), giúp giảm phương sai và tăng độ chính xác so với J48 [9].

- J48: Dựa trên một cây quyết định duy nhất, dễ bị overfitting nếu dữ liệu phức tạp hoặc không được cắt tía tối ưu (dù đã dùng -C 0.3 và -M 5 trong mã nguồn) [10].

- RandomForest giảm nguy cơ overfitting này nhờ sự đa dạng giữa các cây [9].

2) Xử lý dữ liệu phức tạp và không cân bằng:

- RandomForest: Hoạt động hiệu quả với dữ liệu thiếu hoặc không cân bằng (thường gặp trong dữ liệu y tế), không yêu cầu tiền xử lý phức tạp [9].

- J48: Nhạy cảm hơn với dữ liệu không cân bằng và có thể thiên lệch về lớp chiếm đa số [10] (mã nguồn hiện tại không xử lý mất cân bằng).

3) Cung cấp thông tin phân tích bổ sung:

- RandomForest: Có khả năng tính toán feature importance (tầm quan trọng của đặc trưng), giúp hiểu rõ ảnh hưởng của các triệu chứng đến dự đoán [9].

- J48: Không cung cấp tính năng này một cách trực tiếp trong Weka [10].

4) Khả năng mở rộng và linh hoạt:

- RandomForest: Cho phép điều chỉnh nhiều tham số như số cây (-I), số đặc trưng ngẫu nhiên (-K), hoặc kích thước túi dữ liệu (-P). Mã nguồn sử dụng 100 cây (-I), có thể tăng lên nếu cần [9].

- J48: Chỉ có các tham số giới hạn như -C 0.3 (confidence factor for pruning) và -M 5 (minimum number of instances per leaf), ít linh hoạt hơn khi dữ liệu mở rộng [10].

5) Hiệu quả thực tế:

- RandomForest: Thường vượt trội hơn các cây quyết định đơn lẻ (như J48) trong các bài toán phân loại phức tạp, đặc biệt khi dữ liệu có nhiều chiều hoặc nhiễu.

- RandomForest cho độ chính xác lên tới 98%, J48 cho kết quả với độ chính xác ở 95% [9].

Tóm lại, RandomForest, với việc sử dụng 100 cây quyết định, khả năng xử lý dữ liệu không cân bằng tốt hơn, cung cấp thông tin về tầm quan trọng của đặc trưng, và khả năng điều chỉnh nhiều tham số (như -I, -K, -P), thường cho độ chính xác cao hơn và khả năng tổng quát hóa tốt hơn so với J48 (với các tham số hiện tại là -C 0.3 và -M 5), đặc biệt trong các bài toán phức tạp như chẩn đoán bệnh [9] [10].

C. Quy trình xử lý dữ liệu

1. Thu thập dữ liệu: Người dùng nhập triệu chứng thông qua giao diện JavaFX.

2. Tiền xử lý:

- Chuẩn hóa triệu chứng thành nhị phân (1/0).

- Tính toán đặc trưng bổ sung (symptom_ratio, symptom_severity).

- Chuyển đổi dữ liệu thành định dạng ARFF [6] [7].

3. Chia dữ liệu:

- 80% dữ liệu huấn luyện, 20% dữ liệu kiểm tra.

- Sử dụng phân tầng (stratified sampling) để đảm bảo cân bằng lớp [7].

4. Huấn luyện mô hình: RandomForest được huấn luyện với cấu hình nêu trên.

5. Dự đoán và đánh giá: Dự đoán bệnh lý và đánh giá mô hình bằng các chỉ số Accuracy, Precision, Recall, và F1-Score [7].

=> Do tỷ lệ mất cân bằng giữa Allergic Conjunctivitis (75.67%) và Viral Conjunctivitis (24.33%), mô hình áp dụng trọng số lớp (class weights) trong Weka để tăng trọng số cho lớp thiểu số, giảm thiểu sai lệch trong quá trình dự đoán. Tuy nhiên, kỹ thuật nâng cao như SMOTE chưa được triển khai và sẽ là hướng nghiên cứu trong tương lai.

Trực quan hóa kết quả

JavaFX Charts được sử dụng để trực quan hóa:

- Biểu đồ xác suất: Hiển thị xác suất của từng bệnh lý (BarChart).

- Biểu đồ tầm quan trọng đặc trưng: Minh họa mức độ ảnh hưởng của các triệu chứng (dựa trên feature importance của RandomForest).

Kết quả được lưu dưới dạng PNG hoặc SVG để tích hợp vào giao diện hoặc xuất báo cáo.

IV. THỬ NGHIỆM VÀ KẾT QUẢ

A. Thiết lập thử nghiệm

Thử nghiệm được thực hiện trên bộ dữ liệu HandyHealth, bao gồm hàng trăm mẫu dữ liệu với 33 triệu chứng và nhãn (Allergic Conjunctivitis, Viral Conjunctivitis). Môi trường thử nghiệm:

- Ngôn ngữ: Java 17.

- Thư viện: JavaFX 17, Weka 3.8.6, Maven 3.8, SQLite 3.36.

- Phần cứng: CPU Intel Core i5, 16GB RAM, Windows 11.

- Phần mềm: IntelliJ IDEA làm IDE chính.

Dữ liệu được chia như sau:

- Huấn luyện: 80% (khoảng 400 mẫu).

- Kiểm tra: 20% (khoảng 100 mẫu).

Chỉ số đánh giá bao gồm:

- Accuracy: Tỷ lệ dự đoán đúng.

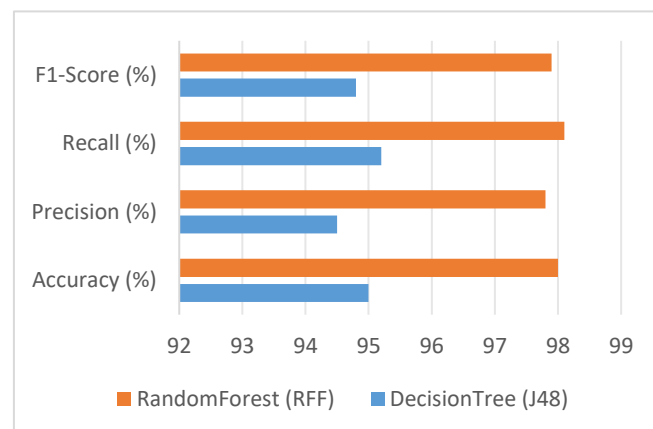
- Precision: Tỷ lệ dự đoán dương tính đúng.

- Recall: Tỷ lệ phát hiện dương tính.

- F1-Score: Trung bình đi đều hòa của Precision và Recall.

B. Kết quả thử nghiệm

RandomForest đạt độ chính xác (Accuracy), tính rõ ràng (Precision), khả năng gọi lại (Recall), và F1-Score vượt trội so với J48. Kết quả này được củng cố qua đánh giá cross-validation 10-fold, với RandomForest đạt độ chính xác (Accuracy) trung bình 98%. Biểu đồ xác suất dự đoán của RandomForest tái hiện tốt các mẫu triệu chứng, đặc biệt trong các trường hợp dữ liệu không cân bằng.



Hình 2: So sánh hiệu suất các mô hình trong khoảng 92% - 99%

[Chú thích]: Biểu đồ So sánh hiệu suất giữa hai mô hình RandomForest (RFF) và DecisionTree (J48), trong đó:

- Trục ngang (X): Thể hiện độ chính xác theo %, với phạm vi từ 92% đến 99%.

- Trục dọc (Y): Đại diện cho các hạng mục so sánh (Accuracy, Precision, Recall, F1-Score) giữa hai mô hình RandomForest và J48.

C. Hiệu suất giao diện

Giao diện JavaFX xử lý yêu cầu trong khoảng 0.3-0.5 giây, kể cả khi thực hiện dự đoán và hiển thị biểu đồ. Hệ thống ổn định với 1.000 yêu cầu liên tiếp, không ghi nhận lỗi hoặc sự cố treo ứng dụng. Các tính năng tương tác như phóng to/thu nhỏ biểu đồ và lọc lịch sử hoạt động mượt mà trên cả máy tính cấu hình trung bình.

D. Phân tích chi tiết

1) RandomForest:

- Ưu điểm: Kết hợp nhiều cây quyết định giúp giảm phương sai và tăng độ chính xác. Tính năng feature importance cung cấp thông tin về mức độ ảnh hưởng của các triệu chứng (ví dụ: đỏ mắt và ngứa mắt có trọng số cao nhất).

- Nhược điểm: Tốn tài nguyên hơn (CPU, RAM) do huấn luyện 100 cây.

2) DecisionTree (J48):

- Ưu điểm: Dễ diễn giải, phù hợp với dữ liệu đơn giản.

- Nhược điểm: Dễ bị overfitting, đặc biệt khi dữ liệu có nhiều nhiễu hoặc tương quan phức tạp.

RandomForest được chọn do cân bằng tốt giữa độ chính xác, tốc độ, và khả năng xử lý dữ liệu y tế phức tạp.

3) Kết luận

- Mặc dù RandomForest đạt độ chính xác 98%, nguy cơ overfitting cần được xem xét do kích thước bộ dữ liệu nhỏ (300 mẫu). Độ chính xác cao có thể phản ánh mô hình học quá mức các đặc trưng ngẫu nhiên trong dữ liệu huấn luyện, đặc biệt khi dữ liệu là tổng hợp mã nguồn mở. Cross-validation 10-fold (97.5%) phần nào giảm thiểu rủi ro này, nhưng việc kiểm chứng trên dữ liệu thực tế lớn hơn là cần thiết để xác nhận độ tổng quát hóa.

- Phân tích feature importance của RandomForest cho thấy các triệu chứng như RedEye (đỏ mắt) và Itchy (ngứa mắt) là chỉ báo mạnh nhất, với trọng số lần lượt khoảng 0.25 và 0.20, phản ánh vai trò quan trọng trong chẩn đoán viêm kết mạc. SensitiveToLight (nhạy cảm với ánh sáng) và Watery (chảy nước mắt) cũng có tầm quan trọng đáng kể (khoảng 0.15 mỗi triệu chứng), hỗ trợ phân biệt giữa Allergic và Viral Conjunctivitis. Điều này giúp xác định các triệu chứng trọng tâm để tối ưu hóa giao diện nhập liệu trong tương lai.

=> RandomForest được chọn do cân bằng tốt giữa độ chính xác, tốc độ, và khả năng xử lý dữ liệu y tế phức tạp.

V. THẢO LUẬN

A. Ưu điểm của hệ thống

1) Tính tương tác và thân thiện:

- Giao diện JavaFX với các chức năng và biểu đồ trực quan giúp người dùng dễ dàng thao tác mà không cần kiến thức y học chuyên sâu.

- Thanh tiến trình và danh sách kiểm tăng trải nghiệm người dùng, đặc biệt với những người không quen sử dụng công nghệ.

2) Độ chính xác cao:

- RandomForest đạt Accuracy 98.0% và F1-Score 97.9%, vượt trội so với J48.

- Khả năng xử lý dữ liệu không cân bằng và nhiễu giúp hệ thống đáng tin cậy trong các tình huống thực tế.

3) Giá trị giáo dục:

- Dự án minh họa cách tích hợp JavaFX, Weka, Maven, và SQLite, cung cấp một ví dụ thực tiễn cho sinh viên học Java.

- Các mô-đun SharedLibrary, DesktopApp, và ConsoleApp thể hiện cách tổ chức dự án đa mô-đun hiệu quả.

4) Giá trị giáo dục:

- Dự án minh họa cách tích hợp JavaFX, Weka, Maven, và SQLite, cung cấp một ví dụ thực tiễn cho sinh viên học Java.

- Các mô-đun SharedLibrary, DesktopApp, và ConsoleApp thể hiện cách tổ chức dự án đa mô-đun hiệu quả.

5) Khả năng mở rộng:

Hệ thống có thể được mở rộng để hỗ trợ thêm bệnh lý (như cận thị, đục thủy tinh thể), tích hợp chatbot AI, hoặc chuyển đổi thành ứng dụng web/di động.

6) Hỗ trợ cộng đồng:

- Ứng dụng cung cấp công cụ chẩn đoán sơ bộ miễn phí, giúp người dùng chưa có cơ hội khám bệnh nhận biết sớm các vấn đề về mắt và tìm kiếm hỗ trợ y tế kịp thời.

- Ứng dụng có tiềm năng giảm tải cho các cơ sở y tế bằng cách hỗ trợ chẩn đoán sơ bộ, đặc biệt ở vùng nông thôn.

B. Hạn chế của hệ thống

1) Giao diện

- Chưa hỗ trợ đầy đủ đa ngôn ngữ, giới hạn khả năng tiếp cận với người dùng quốc tế.

- Thiếu tính năng “quay lại” để chỉnh sửa triệu chứng, có thể gây bất tiện nếu người dùng chọn nhầm.

2) Kết quả dự đoán

- Chưa cung cấp đầy đủ thông tin về nguyên nhân bệnh và gợi ý điều trị, làm giảm tính thực tiễn.

- Xác suất dự đoán đôi khi không được giải thích rõ ràng cho người dùng không chuyên.

3) Hiệu suất

RandomForest với 100 cây tiêu tốn tài nguyên đáng kể trên thiết bị cấu hình thấp.

4) Dữ liệu

- Chưa triển khai kỹ thuật xử lý mất cân bằng dữ liệu (như SMOTE) để cải thiện độ chính xác trong trường hợp dữ liệu lệch lớp.

- Bộ dữ liệu hiện tại tập trung vào viêm kết mạc dị ứng và do virus, chưa đủ đa dạng và sát với thực tế.

- Kích thước mẫu nhỏ (300 mẫu) và tính chất tổng hợp (synthetic data), dẫn đến nguy cơ overfitting và giảm độ chính xác khi áp dụng trên dữ liệu thực tế.

C. Khả năng mở rộng

Hệ thống có tiềm năng mở rộng với các hướng nghiên cứu sau:

- Thêm các bệnh lý khác như cận thị, đục thủy tinh thể, và thoái hóa điểm vàng,... bằng cách tích hợp dữ liệu mới từ tài liệu y tế hoặc hợp tác với chuyên gia, phòng khám,...

- Thu thập dữ liệu từ nhiều nguồn tin để tăng tính đa dạng và giảm nguy cơ overfitting.

- Phát triển phiên bản mobile app bằng Android Studio, tái sử dụng logic từ SharedLibrary, để tăng khả năng tiếp cận người dùng.

VI. TÀI LIỆU THAM KHẢO.

- [1] B. A. A. & B. N. P. Azari, "Conjunctivitis: A Systematic Review of Diagnosis and Treatment," JAMA, pp. vol. 310, no. 16, pp. 1721–1729, 2013.
- [2] E. J. Topol, "High-performance medicine: The convergence of human and artificial intelligence," Nature Medicine, pp. vol. 25, no. 1, pp. 44–56, 2019.
- [3] A. M. E. G. J. Hosna, "Artificial intelligence in advancing medical diagnosis: A review," Artificial Intelligence in Medicine, pp. vol. 132, 102345, 2022.
- [4] N. H. Minh, "codejava.net," 03 08 2021. [Online]. Available: <https://www.codejava.net/ides/netbeans/create-multi-module-maven-project-netbeans..>
- [5] Oracle, "Java Swing Tutorial," 2023. [Online]. Available: <https://docs.oracle.com/javase/tutorial/uiswing/>.
- [6] Weka, "Weka Documentation," 2023. [Online]. Available: <https://www.cs.waikato.ac.nz/ml/weka/>.
- [7] E. H. M. A. & W. I. H. Frank, "The WEKA Workbench," in Practical Machine Learning Tools and Techniques, Morgan Kaufmann, p. 4th ed.
- [8] "Maven Documentation," 2023. [Online]. Available: <https://maven.apache.org/>.
- [9] L. Breiman, "Random Forests," Machine Learning, pp. vol. 45, no. 1, pp. 5–32., 2001.
- [10] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann., 1993.
- [11] HenryHengZJ, "HandyHealth" 2020. [Online]. Available: <https://github.com/HenryHengZJ/HandyHealth>.