# **Chương I. TỔNG QUAN VỀ ĐỀ TÀI**

## **1.1. GIỚI THIỆU NỘI DUNG NGHIÊN CỨU**

Thế kỷ 21 với sự bùng nổ và phát triển mạnh mẽ của công nghệ thông tin, đã đặt Internet vào vị trí quan trọng trong đời sống, trở thành một trong những lĩnh vực phát triển mạnh mẽ nhất. Bằng chứng là trong thời gian gần đây, số lượng người sử dụng Internet không ngừng tăng lên. Theo thống kê từ tờ báo Thanh Niên [1] cho thấy 66% dân số trái đất (tương đương 5,35 tỉ người) đang dùng mạng Internet, trong khi đó tại Việt Nam có khoảng 73,7% dân số sử dụng mạng xã hội. Với phần trăm lượng người sử dụng Internet như trên, việc đảm bảo sự riêng tư của thông tin cá nhân và an toàn trực tuyến trở thành một ưu tiên hàng đầu. Một trong số những mối đe dọa phổ biến mà người dùng Internet phải đối mặt là phishing website, mục đích của những trang web này là thu thập thông tin cá nhân và tài khoản của những người truy cập đến nhằm chiếm đoạt để thực hiện các mục đích xấu.

Tấn công phishing website từ lâu luôn là một mối đe dọa đáng lo ngại trong cộng đồng an ninh mạng nói chung và đối với các tổ chức cũng như người dùng Internet nói riêng [2]. Trong thời đại mà Internet đã có tầm quan trọng đối với các giao dịch và thanh toán trực tuyến, việc đảm bảo sự bảo mật thông tin cá nhân càng trở nên cực kỳ quan trọng. Điều này đặc biệt đúng khi người dùng thường tin tưởng rằng việc cung cấp thông tin như tài khoản và mật khẩu của họ lên Internet là an toàn. Tận dụng điều này, những kẻ tấn công đã tạo ra các trang web giả mạo để lừa lấy thông tin riêng tư của người dùng.

Một số cuộc tấn công bằng hình thức này đã được Nivedita James Palatty [3] thống kê trong năm 2024 như sau:

- 55% các trang web lừa đảo đã sử dụng tên thương hiệu của các tổ chức nổi tiếng để lừa dối và chiếm đoạt thông tin của người dùng.

- 92% tổ chức tại Úc đã hứng chịu một cuộc tấn công phishing thành công, tăng một cách đáng kể so với năm 2021 (là 53%).

- Các thương hiệu bị mạo danh nhiều nhất trong các cuộc tấn công phishing là: Google (13%), Amazon (13%), Facebook (10%), Netflix (2%).

- Các trang web lừa đảo vẫn có thể sử dụng giao thức HTTPS để thể hiện rằng đó là một trang web bảo mật và tạo sự tin cậy cho người dùng. Các trang web này được thống kê chiếm khoảng 32% trong tổng số các trang web lừa đảo (năm 2020).

- 40% các trang web lừa đảo sử dụng tên miền .com; 1,8% sử dụng .org và 3% sử dụng .net.

Có thể nói phishing website là hình thức tấn công nhắm vào các yếu điểm tâm lý của người dùng Internet thay vì lợi dụng từ các lỗ hổng bảo mật trong trang web [4]. Điều này chỉ ra rằng những kẻ tấn công sẽ tận dụng sự sơ hở trong cách tư duy nhận biết của người dùng với một trang web để thực hiện các cuộc tấn công bằng phishing website. Để giải quyết mối nguy hiểm từ phishing website, nhiều nhà nghiên cứu đã nghiên cứu và sử dụng nhiều phương pháp để có thể ngăn chặn người dùng trở thành nạn nhân từ các cuộc tấn công này. Một trong số các phương pháp thử nghiệm đó là việc phân tích URL của một trang web phishing sau đó so sánh với dữ liệu phân tích từ URL của một trang web an toàn, cuối cùng dựa trên sự so sánh đưa ra kết quả liệu rằng đó có phải là một phishing website hay không. Một phương pháp khác là sử dụng thuật toán heuristics. Thuật toán này sử dụng các mẫu suy luận hoặc các điểm đặc trưng của các cuộc tấn công phishing đã biết trước để phát hiện ra phishing website. Cụ thể, thuật toán này so sánh các đặc điểm của URL hoặc các yếu tố khác của trang web với các mẫu của các cuộc tấn công đã được ghi nhận trước đó [5].

Ngoài các phương pháp đã được đề cập, một số nhà nghiên cứu đã sử dụng kỹ thuật Machine Learning (học máy) để phát hiện và ngăn chặn phishing website. Machine Learning [6] là một nhánh của trí tuệ nhân tạo (AI) và là một lĩnh vực của khoa học máy tính. Với sự tiến bộ không ngừng của công nghệ, các ứng dụng từ Machine Learning ngày càng thân thiện và tiếp cận gần hơn đến đời sống hiện nay. Không thể phủ nhận rằng Machine Learning là một phương pháp tiên tiến và là một công cụ mạnh mẽ để áp dụng trong việc đối phó với các cuộc tấn công từ phishing website. Nhiều nghiên cứu đã chứng minh sức mạnh của Machine Learning đối với vấn đề này, cụ thể:

- Nghiên cứu của Sanglerdsinlapachai và Rungsawang [7] bổ sung các đặc trưng mới cho đặc trưng heuristics của CANTINA và sử dụng sáu kỹ thuật học máy để cải thiện hiệu suất chặn, tăng độ chính xác phát hiện lên 15% và 20% đối với f-measure và tỷ lệ lỗi tương ứng.

- Mohammad và nhóm tác giả [8] đã thiết kế công cụ chống phishing, có thể dự đoán tấn công phishing sử dụng mạng neural networks (NN). Sử dụng 17 đặc trưng và một tập dữ liệu gồm 600 URL chính thức và 800 URL phishing, đạt độ chính xác 92.48%.

- Pradeepthi và Kannan [9] đã thực hiện khảo sát với 4,500 URL, sử dụng nhiều thuật toán như Decision Trees (DT), Random Forest (RF), và C4.5. Kết luận cho thấy rằng các bộ phân loại dựa trên cây là phù hợp nhất cho việc phân loại URL phishing.

- Mamun và nhóm tác giả [10] đã tiến hành phân loại các loại tấn công khác nhau, sử dụng nhiều thuật toán như K-Nearest Neighbors (KNN) và Random Forest (RF), với độ chính xác đáng kể.

- Sirageldin và nhóm tác giả [11] trình bày cơ chế phát hiện trang web phishing dựa trên phân tích từ vựng URL và nội dung trang. Sử dụng các thuật toán như DT, ANN, NB, SVM, và KNN với độ chính xác lần lượt là 95.12%, 96.01%, 88.47%, 93.57%, và 92.90%.

- Hieu Nguyen và Thai Nguyen [12], 2 tác giả người Việt đã đánh giá và so sánh 5 thuật toán phân loại để phát hiện các trang web phishing bằng cách trích xuất các đặc tính URL và thông tin nội dung trang web. Các thuật toán bao gồm J48 DT, RF, SVM, NB và NN với độ chính xác phân loại lần lượt là 98,5%, 98,8%, 86,1%, 96,9% và 98,4%.

Dựa vào kết quả từ các công trình nghiên cứu trên có thể thấy giải thuật Random Forest (RF – Rừng ngẫu nhiên) và Decision Tree (DT – Cây quyết định) có độ chính xác tương đối cao trong việc phát hiện phishing website. Weedon và nhóm tác giả [13] đã đánh giá hiệu suất của Random Forest bằng cách so sánh nó với ba thuật toán khác là J48, Naive Bayes (NB) và Logistic Regression, sử dụng cùng bộ dữ liệu huấn luyện. Kết quả cho thấy được rằng Random Forest đạt độ chính xác cao nhất với 86,9% và tỷ lệ sai dương thấp hơn so các thuật toán so sánh. Tuy nhiên, ở một nghiên cứu khác nhóm tác giả Musa Hajara, A.Y. Gital, Fatima Umar Zambuk, A. Umar [14] sử dụng thuật toán XGBoost để cải thiện hiệu suất của một mô hình dự đoán phishing website và so sánh kết quả với phương pháp Probabilistic Neural Networks (PNN – Mạng nơ-ron xác suất) và Random Forest. Kết quả nghiên cho thấy hiệu suất dự đoán của mô hình sử dụng thuật toán XGBoost đã được tối ưu hóa lên 97,29% vượt qua cả PNN và RF.

Chính vì vậy, trong phạm vi chuyên đề này tôi sẽ tiến hành nghiên cứu và sử dụng ba thuật toán Random Forest, Decision Tree và XGBoost để huấn luyện mô hình dự đoán phishing website, kết hợp với việc phân tích và kiểm thử hiệu suất của từng mô hình trên các tập dữ liệu cụ thể.

## **1.2. CÁC NỘI DUNG THỰC HIỆN**

Chuyên đề này sẽ tìm hiểu, nghiên cứu về ba thuật toán Random Forest, Decision Tree và XGBoost. Đồng thời sẽ sử dụng công cụ Google Colab để xây dựng, huấn luyện mô hình dự đoán phishing website dựa trên ba thuật toán này. Theo đó, cấu trúc và nội dung các chương của bài báo cáo chuyên đề này như sau:

* **Chương 1.** **Tổng quan về đề tài**: Trong chương này sẽ cung cấp một cái nhìn tổng quan về đề tài nghiên cứu cũng như nội dung của các chương. Chương sẽ tập trung khai thác và tìm hiểu các bài báo cũng như các nghiên cứu liên quan đến phishing website và việc ứng dụng Machine Learning để phát hiện loại tấn công này của các tài liệu.
* **Chương 2.** **Cơ sở lý thuyết**: Phần cơ sở lý thuyết sẽ đề cập đến các vấn đề trong lĩnh vực an ninh mạng; những khái niệm liên quan đến phishing website, machine learning; các thuật toán được áp dụng để xây dựng mô hình dự đoán phishing website.
* **Chương 3.** **Huấn luyện và kiểm thử mô hình phát hiện phishing website**: Trong chương này sẽ đề cập đến các tập dữ liệu được sử dụng trong nghiên cứu, quá trình huấn luyện và kiểm thử mô hình trên các tập dữ liệu.
* **Chương 4.** **Kết luận và hướng phát triển**: Phần này sẽ trình bày các nhận định chính từ kết quả nghiên cứu, kết quả hoàn thành của đề tài so với mục tiêu ban đầu. Những hạn chế và khó khăn trong quá trình nghiên cứu, đồng thời đưa ra những gợi ý về hướng phát triển trong tương lai.

# **Chương II. CƠ SỞ LÝ THUYẾT**

## **2.1. CÁC MỐI ĐE DỌA TRONG AN NINH MẠNG VÀ PHƯƠNG THỨC TẤN CÔNG PHISHING WEBSITE**

### **2.1.1. Tổng quan về an ninh mạng và các mối đe dọa**

An ninh mạng (network security) tại khoản 1 điều 2 của Luật An ninh mạng 2018 được định nghĩa “An ninh mạng là sự bảo đảm hoạt động trên không gian mạng không gây phương hại đến an ninh quốc gia, trật tự, an toàn xã hội, quyền và lợi ích hợp pháp của cơ quan, tổ chức, cá nhân”.

Ngoài ra, theo Harrington [15] định nghĩa an ninh mạng là quá trình bảo vệ dữ liệu được lưu trữ trên hoặc truyền qua mạng khỏi sự tiết lộ hoặc sửa đổi không ủy quyền, bao gồm cả các sự cố vô tình như lỗi phần mềm gây tổn thất dữ liệu. Đồng thời, nó cũng bao gồm việc ngăn chặn truy cập không ủy quyền đối với dữ liệu, bảo vệ sự riêng tư và bảo đảm tính toàn vẹn của hệ thống mạng. Một hệ thống mạng [16] được cho là an toàn khi đáp ứng đầy đủ ba yếu tố của một tam giác bảo mật đều CIA: C – Confidentiality (Bí mật); I – Intergrity (Toàn vẹn); A – Availability (Sẵn có)



Hình 2.. Tam giác bảo mật CIA

Khi công nghệ ngày càng tiến bộ [17], sự phụ thuộc của con người vào máy tính, thiết bị và mạng lưới cũng tăng cao. Mọi khía cạnh của cuộc sống, từ công việc, giải trí đến các sinh hoạt hằng ngày mạng lưới đều đóng vai trò quan trọng. Trên thực tế, một thế giới hiện đại sẽ không hoạt động nếu như không có nó. Vì vậy không thể phủ nhận vai trò quan trọng của an ninh mạng trong việc bảo vệ người dùng trong thế giới số ngày nay.

Vai trò của an ninh mạng [15] là bảo vệ dữ liệu của người dùng, đồng thời cũng đảm bảo tính bí mật, tính toàn vẹn và tính sẵn có của dữ liệu:

- Bảo vệ dữ liệu: Đảm bảo rằng người không được ủy quyền sẽ không thể biết được dữ liệu cần được bảo mật.

- Bảo vệ tính toàn vẹn của dữ liệu: Đảm bảo rằng dữ liệu sẽ không bị sửa đổi hoặc phá hủy một cách trái phép.

- Đảm bảo tính sẵn có: Đảm bảo rằng dữ liệu luôn có sẵn khi cần thiết.

- Phòng ngừa nguy hiểm từ bên ngoài: Bảo vệ mạng khỏi các cuộc tấn công từ hacker hoặc cracker.

- Ngăn chặn đe dọa từ bên trong: Những sự cố vô tình từ bên trong sẽ không gây ra nguy cơ cho an ninh mạng

- Đảm bảo tuân thủ pháp lý: Các hoạt động an ninh mạng luôn tuân thủ các quy định pháp lý liên quan đến bảo vệ thông tin cá nhân và dữ liệu.

Ngoài việc bảo vệ dữ liệu của người dùng, an ninh mạng [18] còn là một yếu tố cần thiết để đảm bảo tính toàn vẹn và sự uy tín của các hệ thống doanh nghiệp, giúp các doanh nghiệp đảm bảo được sự an toàn của thông tin đồng thời xây dựng và duy trì một môi trường kinh doanh an toàn ổn định và đáng tin cậy.

Ngày nay [16], việc bảo vệ dữ liệu và hệ thống ngày càng trở nên quan trọng, đặc biệt khi môi trường an ninh mạng phải đối mặt với nhiều mối đe dọa bên trong và bên ngoài. Phổ biến nhất trong những mối đe dọa là các cuộc tấn công từ bên ngoài bao gồm các kiểu tấn công như phishing, DDoS (tấn công từ chối dịch vụ), ransomware, malware và các cuộc tấn công khác. Đồng thời, các nguy cơ từ bên trong cũng nguy hiểm không kém. Sự lạm dụng thông tin xác thực và tấn công nội bộ có thể gây ra tổn thất lớn cho các tổ chức khi nhân viên hoặc người dùng bị lừa hoặc thao túng để tiết lộ thông tin xác thực của họ cho kẻ tấn công.

### **2.1.2. Phương thức tấn công phishing website**

Phishing [19] là một thuật ngữ dùng trong an ninh mạng, nó mô tả một hình thức tấn công lừa đảo thông tin. Thuật ngữ này được sử dụng từ giữa những năm 1990 để mô tả những kẻ tấn công (fishers) tạo ra các trang web phishing hoặc giả email, giống hệt với trang web và email của các tổ chức, hoặc cá nhân đáng tin cậy, để khiến người dùng dễ bị đánh lừa và cung cấp thông tin cá nhân của họ.

Phương thức [16] tấn công phishing website phổ biến là kẻ tấn công sẽ gửi một email giả mạo và bên trong có chứa các liên kết. Khi người nhận mở email, và nhấp vào các liên kết đó họ sẽ bị chuyển hướng đến các trang web phishing, và được yêu cầu cung cấp thông tin cá nhân nhạy cảm hoặc tải xuống phần mềm độc hại. Thông thường, trên URL của các trang web này thường có một số đặc điểm về số lượng dấu chấm, số lượng tên miền phụ, địa chỉ IP trong tên miền và các đặc điểm khác để người dùng có thể nhận dạng đó là một phishing website. Ngoài phương thức trên, các kẻ lừa đảo luôn tìm cách tận dụng các lỗ hổng trong bảo mật để thực hiện các cuộc tấn công phishing. Một số phương thức phổ biến khác mà chúng sử dụng như:

-Algorithm-Based Phishing: Đây là loại tấn công đầu tiên được phát hiện bởi America Online (AOL). Kẻ lừa đảo sử dụng thuật toán để khớp số thẻ tín dụng của các tài khoản AOL.

- Hosts File Poisoning: Hacker thay đổi hoặc làm giả tệp hosts trên hệ điều hành Windows. Khi người dùng cố gắng truy cập vào một trang web cụ thể, họ sẽ bị chuyển hướng đến trang web của hacker hoặc nhận thông báo lỗi "The Page Not Found". Nếu chuyển hướng thành công đến trang web giả mạo, thông tin của người dùng sẽ bị ghi lại và đánh cắp.

- Whaling: Loại tấn công này nhắm vào các nhà quản lý cấp cao của tổ chức, thường thông qua việc gửi email với nội dung giống như phàn nàn từ phía khách hàng để lừa đảo.

## **2.2. MACHINE LEARNING VÀ THUẬT TOÁN PHÁT HIỆN PHISHING WEBSITE**

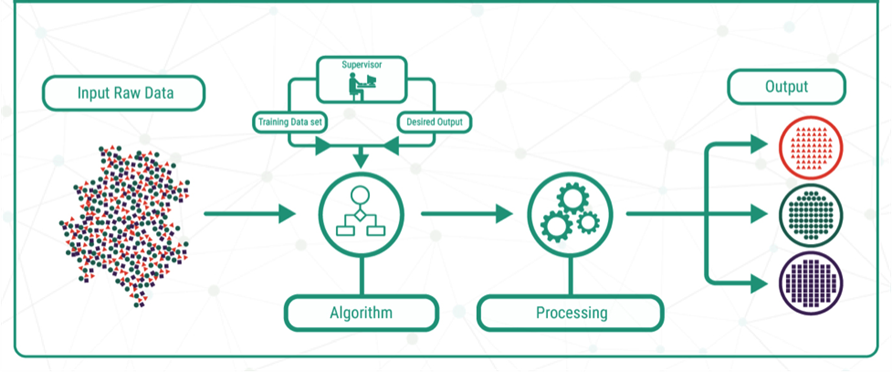
### **2.2.1. Tổng quan về machine learning**

Machine Learning [6] hay còn gọi là học máy là một nhánh của trí tuệ nhân tạo (AI), nó sử dụng các thuật toán được đào tạo trên các bộ dữ liệu để tạo ra các mô hình tự học có khả năng dự đoán kết quả và phân loại thông tin mà không cần sự can thiệp trực tiếp của con người. Trong các bài toán thực tế cho mục đích kinh doanh, học máy thường được sử dụng chẳng hạn như gợi ý sản phẩm cho người tiêu dùng dựa trên lịch sử mua sắm, dự đoán biến động thị trường chứng khoán, và một số ứng dụng khác.

Thông thường [20] hai thuật ngữ “Machine Learning” và “Artificial Intelligence” (AI) thường được sử dụng thay thế cho nhau, nhưng chúng có ý nghĩa khác nhau. Trong khi trí tuệ nhân tạo đề cập đến nỗ lực tổng thể để phát triển máy tính có khả năng tư duy giống con người, thì Machine Learning cụ thể ám chỉ việc dùng thuật toán được đào tạo trên các bộ dữ liệu để thực hiện mục tiêu này.

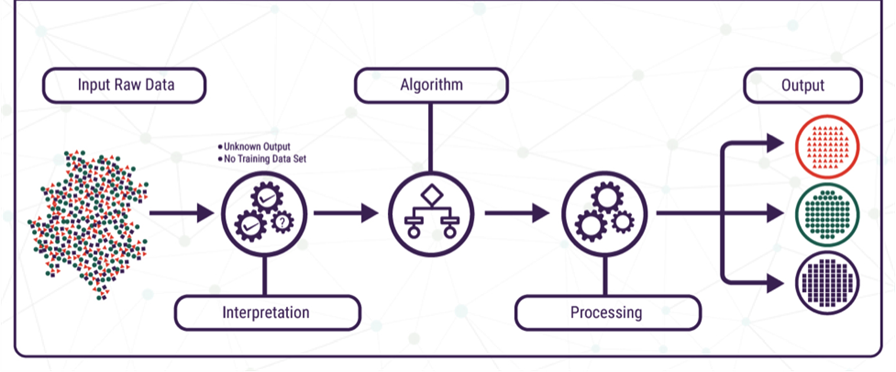
Các thuật toán học máy [21] thường được phân thành bốn nhóm chính dựa trên phương pháp học:

- Học có giám sát (Supervised learning): Trong học có giám sát, dữ liệu đầu vào đã được gắn nhãn và sắp xếp trước, và mục tiêu là dự đoán hoặc phân loại dữ liệu mới dựa trên mẫu đã biết trước. Ví dụ, trong việc phân loại email, thuật toán có thể được dạy để nhận biết các tin nhắn rác dựa trên các đặc điểm đã được xác định trước.



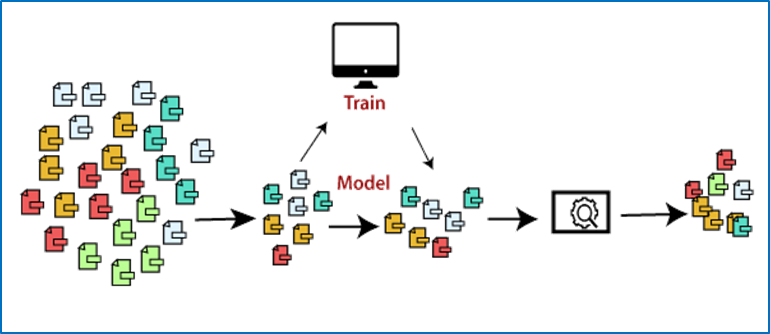
Hình 2.2 Mô hình học có giám sát

- Học không giám sát (Unsupervised learning): Trái ngược với học có giám sát, trong học không giám sát, không có dữ liệu được gắn nhãn trước và mục tiêu là khám phá cấu trúc ẩn trong dữ liệu. Ví dụ, một ứng dụng của học không giám sát có thể là việc phân nhóm khách hàng dựa trên hành vi mua hàng của họ mà không cần biết trước các nhóm mong đợi.



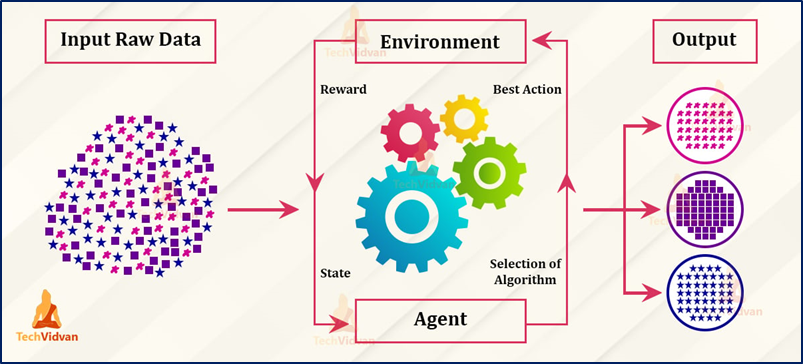
Hình 2.3 Mô hình học không giám sát

- Học bán giám sát (Semi-supervised learning): Trong học bán giám sát, chỉ một phần dữ liệu đầu vào được gắn nhãn. Thuật toán cố gắng học từ dữ liệu được gắn nhãn và áp dụng kiến thức đó để phân loại hoặc dự đoán các dữ liệu không gắn nhãn.



Hình 2.4 Mô hình học bán giám sát

- Học tăng cường (Reinforcement learning): Học tăng cường là một kỹ thuật học máy trong đó mô hình học bằng cách tương tác với môi trường. Hệ thống thực hiện các hành động và nhận phản hồi từ môi trường dựa trên các hành động đó để tối ưu hóa một mục tiêu cụ thể. Ví dụ, trong việc huấn luyện một robot để điều khiển nó di chuyển qua một mê cung, mô hình sẽ nhận phản hồi dựa trên các hành động của nó để tìm đường đi tối ưu.



Hình 2.5 Mô hình học tăng cường

### **2.2.2. Vai trò của machine learning trong an ninh mạng**

Hiện nay, học máy [6] đã chứng minh được giá trị của mình về phân tích dữ liệu, từ lĩnh vực tài chính đến chăm sóc sức khỏe, đến ngành công nghiệp robot và kiểm soát chất lượng. Điều này cho thấy rằng nếu được sử dụng đúng cách, học máy có thể hỗ trợ các chuyên gia an ninh mạng xây dựng các hệ thống phòng thủ vững chắc chống lại các rủi ro và lỗ hổng bảo mật ngày càng phức tạp.

Học máy [22] đã đem lại các phương pháp an ninh mạng mới, vượt ngoài phương pháp truyền thống dựa trên quy tắc và hệ thống phát hiện dựa trên chữ ký, tạo ra một trạng thái bảo mật mạnh mẽ hơn cho các tổ chức. Một số cách mà trí tuệ nhân tạo và học máy được áp dụng vào an ninh mạng bao gồm:

- Phát hiện bất thường: Học máy có thể phân tích lượng lớn dữ liệu để xác định các hành vi không bình thường và phát hiện các mối đe dọa tiềm ẩn.

- Phát hiện phần mềm độc hại (malware): Học máy giúp xác định các chủng phần mềm độc hại mới và đang biến đổi bằng cách phân tích đặc điểm của tệp và hành vi mã nguồn.

- Phát hiện lừa đảo (phishing): Các mô hình học máy có thể phân tích nội dung email, URL và hành vi người dùng để nhận diện và ngăn chặn các nỗ lực lừa đảo.

- Phân tích hành vi: Học máy có thể liên tục theo dõi và phân tích hành vi người dùng để xác định hoạt động đáng ngờ hoặc không được ủy quyền.

- Săn mối đe dọa: Các chuyên gia an ninh mạng có thể sử dụng học máy để nhận diện các mối đe dọa zero-day và xử lý với các mối đe dọa kiên trì cao cấp.

- Giảm số lượng báo lỗi giả định: Học máy giúp giảm số lượng báo lỗi giả định và giải phóng thời gian cho đội an ninh để điều tra và đối phó với các mối đe dọa thực sự.

- Khả năng mở rộng: Học máy có khả năng thích nghi với sự thay đổi của cảnh quan mạng và mở rộng theo tỷ lệ khi lượng dữ liệu tăng lên.

- Bảo mật đám mây: Sử dụng các phương pháp bảo mật học máy trong môi trường đám mây giúp nhận diện và xử lý các mối đe dọa số mà các phương pháp truyền thống dựa trên ranh giới thường không đủ.

### **2.2.3. Thuật toán machine learning trong phát hiện phishing website**

Trong bài nghiên cứu này, các thuật toán học máy được sử dụng là các giải thuật học có giám sát bao gồm Random Forest, Decision Tree và XGBoost:

- Random Forest: Random Forest [19] là một thuật toán mạnh mẽ trong học máy, sử dụng phương pháp đóng gói để kết hợp nhiều cây quyết định và sử dụng kết quả của chúng để tạo ra dự đoán tốt hơn. Random Forest được áp dụng để phân loại trang web giữa hợp pháp và lừa đảo. Nhờ sự kết hợp của nhiều cây quyết định, Random Forest tạo ra độ chính xác cao và khả năng xử lý dữ liệu bị mất.

- Decision Tree: Decision Tree [23] được coi là một trong những thuật toán phân loại nhị phân phổ biến nhất trong lĩnh vực học máy. Nó hoạt động nhanh chóng bằng cách xây dựng một cây quyết định từ dữ liệu huấn luyện và dự đoán dữ liệu mới. Thuật toán này sử dụng các thuộc tính có sẵn để phân loại và liên tục tách dữ liệu thành các nhóm cho đến khi tìm được kết quả cuối cùng. Decision Tree tạo ra một mô hình có khả năng dự đoán giá trị hoặc lớp mục tiêu.

- XGBoost: XGBoost (eXtreme Gradient Boosting) [24] là một giải thuật dựa trên gradient boosting, sử dụng Decision Tree làm base model. XGBoost đã đạt giải Kaggle’s Higgs Machine Learning Challenge năm 2014. Khác với việc huấn luyện một cây quyết định lớn, XGBoost sử dụng nhiều cây nhỏ hơn. Mặc dù mỗi cây nhỏ có độ chính xác riêng không cao, nhưng kết hợp kết quả của chúng giúp tạo ra một mô hình có độ chính xác cao và phương sai thấp. XGBoost được đánh giá là tránh được vấn đề quá khớp và có hiệu suất tốt nhất so với các giải thuật boosting khác như AdaBoost, GradientbTree Boosting.

## **2.3. MÔI TRƯỜNG THỰC HIỆN NGHIÊN CỨU**

Hai công cụ chính mà tôi dùng trong nghiên cứu này là Google Colab và ngôn ngữ lập trình Python:

- Google Colab: Tài khoản mà tôi sử dụng trên Google Colab được đăng ký miễn phí với bộ nhớ 27,07GB/107,72GB và RAM 0,87GB/12,67GB. Google Colab [25] là viết tắt của Google Colaboratory, là một dịch vụ môi trường Jupyter Notebook trực tuyến. Người dùng có thể tạo, chia sẻ và chỉnh sửa các notebook một cách thuận tiện mà không cần cài đặt phần mềm. Kết hợp với Google Drive, Google Colab cho phép trực tiếp truy cập và xử lý dữ liệu trong notebook. Mục đích chính của nó là hỗ trợ người dùng trong phát triển và chia sẻ các dự án về khoa học dữ liệu, học máy và nghiên cứu khoa học thông qua môi trường lập trình Python dễ sử dụng. Google Colab do Google cung cấp [26], là một nền tảng điện toán đám mây miễn phí dành cho việc thực hiện các tác vụ về khoa học dữ liệu, học máy và trí tuệ nhân tạo. Một số điểm nổi bật của Google Colab:

* Miễn phí: Google Colab là miễn phí, điều này là một lợi ích lớn cho sinh viên và người mới bắt đầu trong lĩnh vực khoa học dữ liệu.
* Dễ sử dụng: Giao diện của Google Colab thân thiện và trực quan, giúp người dùng dễ dàng tạo và thực thi mã Python mà không cần cài đặt phần mềm.
* Truy cập vào tài nguyên mạnh mẽ: Google Colab cung cấp CPU và GPU mạnh mẽ, giúp xử lý các tác vụ phức tạp trong khoa học dữ liệu.
* Lưu trữ đám mây: Dữ liệu và mã người dùng được lưu trữ an toàn trên đám mây của Google, giúp dễ dàng truy cập và chia sẻ từ mọi nơi.
* Tích hợp với các công cụ khác: Google Colab tích hợp tốt với các công cụ khoa học dữ liệu phổ biến khác như TensorFlow, PyTorch và Pandas, giúp người dùng tiện lợi trong việc sử dụng các công cụ này trong dự án của mình.

- Ngôn ngữ lập trình Python [27] là một lựa chọn phổ biến trong lĩnh vực học máy nhờ vào sự đơn giản và dễ sử dụng của nó. Python được biết đến với cú pháp ngắn gọn, dễ đọc và logic, hỗ trợ tối ưu hóa thời gian cho các nhà phát triển. Đặc biệt, trong lĩnh vực này, Python cung cấp các thư viện và framework như SciPy, NumPy và Pandas để phân tích dữ liệu và liên kết với các ứng dụng web dễ dàng:

* SciPy (Scientific Python): SciPy cung cấp các công cụ để thực hiện các tính toán khoa học phức tạp như tối ưu hóa, lập trình tuyến tính, tính toán đại số tuyến tính và xử lý tín hiệu và hình ảnh.
* NumPy (Numerical Python): NumPy cho phép tạo, xử lý và thực hiện tính toán trên các mảng nhiều chiều. Nó cung cấp các công cụ để thực hiện các phép toán số học cơ bản và phức tạp trên dữ liệu.
* Pandas: Pandas là một thư viện mạnh mẽ để làm việc với dữ liệu có cấu trúc, thường được áp dụng trong phân tích và xử lý dữ liệu. Nó cũng cung cấp các công cụ để thực hiện các thao tác như lọc, sắp xếp và nhóm dữ liệu.

# **Chương III. HUẤN LUYỆN VÀ KIỂM THỬ MÔ HÌNH PHÁT HIỆN PHISHING WEBSITE**

Trong chương này tôi sẽ thực hiện các công việc sau: Tiền xử lý dữ liệu đầu vào (trực quan hóa và làm sạch dữ liệu). Huấn luyện các mô hình Random Forest, Decision Tree và XGBoost để phát hiện phishing website trên tập dữ liệu đầy đủ gồm 48 thuộc tính và 10.000 mẫu. Sau đó, để giảm thời gian huấn luyện và kích thước của các mô hình, trên tập dữ liệu huấn luyện tôi đã tiến hành trích chọn ra 30 đặc trưng dựa trên hai phương pháp là so sánh hệ số tương quan giữa các thuộc tính và danh sách các đặc trưng quan trọng có giá trị cao nhất của thuật toán XGBoost để huấn luyện mô hình. Sau khi đã huấn luyện và xuất các mô hình, tôi sẽ tiến hành kiểm thử các mô hình trên tập dữ liệu tự thu mà tôi đã xây dựng gồm 100 mẫu. Các kết quả đạt được sẽ được tôi so sánh và đánh giá.

## **3.1. DỮ LIỆU**

Ở phần này tôi sẽ giới thiệu về hai tập dữ liệu mà tôi đã dùng trong bài nghiên cứu của mình: tập dữ liệu có sẵn dùng để huấn luyện mô hình; tập dữ liệu tự thu thập và xây dựng để kiểm thử mô hình đã được huấn luyện. Hai tập dữ liệu này sẽ được tôi làm sạch và trực quan hóa trước khi đưa vào huấn luyện mô hình.

### **3.1.1. Tập dữ liệu có sẵn**

Tập dữ liệu có sẵn mà tôi dùng cho việc huấn luyện có tên là “*Phishing\_Legitimate\_full.csv”* [28] được lấy từ nguồn cung cấp công khai tại trang Kaggle, tập dữ liệu gồm: 10.000 mẫu trong đó 5.000 mẫu là trang web phishing và 5.000 mẫu là trang web non-phishing; 50 cột là 48 thuộc tính (trừ cột id và label) đại diện cho 48 đặc trưng của một trang web trong việc phân biệt các trang web phishing và non-phishing (Bảng 3.1). Các mẫu đã được phân tích và tính giá trị cụ thể cho mỗi đặc trưng, dựa vào đó để xác định được một trang web có phải là phishing hay không. Ở cuối mỗi hàng của tập dữ liệu có một cột kết quả (cột label) để xác định bản chất thực sự của trang web đó, là 0 nếu đó là một trang web non-phishing và là 1 nếu đó là một trang web phishing. Đặc điểm [29] của một số dấu hiệu được dùng để phát hiện phishing trên một trang web như sau:

- Số kí tự dấu chấm trong URL (NumDots): Thông thường một trang web được xem là an toàn khi trong URL của trang web đó không nhiều hơn 2 dấu chấm (trừ www.). Vì vậy các URL có nhiều hơn 2 dấu chấm sẽ được phân loại là phishing website.

- Số lượng tên miền phụ trong URL (Subdomain Level): Những kẻ tấn công thường thêm nhiều tên miền phụ hơn và có thể thêm cả tên miền của trang web gốc làm tên miền phụ để đánh lừa người dùng. Vì vậy ở phần tên miền của các trang web phishing thường có 3 dấu chấm trở lên để phân cách giữa các tên miền phụ.

- Độ dài của URL (URL Length): Độ dài URL trung bình của một trang web an toàn thường là 75. Chính vì vậy các trang web được xếp là phishing nếu có độ dài URL lớn hơn 75.

- Ký tự “@” trong URL (At Symbol): Đối với một trang web an toàn thì trong URL thường sẽ không xuất hiện ký tự “@”. Đây là ký tự mà những kẻ tấn công thường thêm vào cuối tên miền trang web để khiến nó trông giống như một tên miền trang web hợp pháp đối với người dùng.

- Ký tự “-” trong tên miền URL (Num Dash): Trong các tên miền an toàn, dấu “-” hiếm khi được sử dụng. Vì vậy, để đánh lừa người dùng những kẻ tấn công thường sử dụng thêm tiền tố hoặc hậu tố cách nhau bằng dấu “-” trong tên miền.

- Địa chỉ IP trong tên miền (IP Address): Nếu trong URL của một trang web có chứa địa chỉ IP thì sẽ được xem là phishing website.

- Favicon từ tài nguyên ngoại vi (Ext Favicon): Favicon là một biểu tượng đóng vai trò nhận dạng cho một trang web. Những kẻ tấn công thường sao chép favicon từ trang web chính thức để hiển thị trên trang web giả mạo của mình. Do đó, nếu liên kết của favicon không trùng với tên miền của trang thì trang web đó được xem là phishing.

Ngoài ra trong tập dữ liệu này, tôi sẽ tiến hành chia tỷ lệ training và testing cho việc huấn luyện. Theo như nhóm tác giả [30] đã đánh giá tỷ lệ 70:30 là tỷ lệ tốt nhất cho các mô hình học máy. Tuy nhiên, trong bài nghiên cứu của tác giả Rishikesh Mahajan và Irfan Siddavatam [23] đã tiến hành chia tập dữ liệu với tỷ lệ training và testing lần lượt là 50:50, 70:30 và 90:10. Kết quả cuối cùng cho thấy được rằng độ chính xác của mô hình càng cao khi tỷ lệ cho phần training càng lớn. Vì vậy để có sự đánh giá khách quan nhất cho cho mô hình huấn luyện của mình, tôi sẽ chia tập dữ liệu theo tỷ lệ 80:20, trong đó 80% dữ liệu training và 20% dữ liệu testing.

Bảng 3.1 48 thuộc tính của bộ dữ liệu

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Ý nghĩa | Thuộc tính | Ý nghĩa |
| NumDots | Số kí tự dấu chấm trong URL | NumSensitiveWords | Số lượng từ nhạy cảm trong URL |
| Subdomain Level | Số lượng phần con (subdomain) của URL | Embedded Brand Name | Thương hiệu có xuất hiện như một phần của URL hay không |
| Path Level | Số lượng mức độ trong đường dẫn URL | PctExtHyperlinks | Phần trăm siêu liên kết ngoại vi trong trang web |
| URL Length | Độ dài của URL | PctExtResourceUrls | Phần trăm URL tài nguyên ngoại vi trong trang web |
| Num Dash | Số lượng kí tự gạch ngang trong URL | Ext Favicon | Có sử dụng favicon từ tài nguyên ngoại vi hay không |
| NumDashInHostname | Số lượng dấu gạch ngang trong phần tên miền của URL | InsecureForms | Có chứa mẫu không an toàn (http) hay không |
| At Symbol | Có tồn tại ký tự “@” trong URL hay không | RelativeFormAction | Hành động mẫu là tương đối (đối với giao thức http) hay không |
| Tilde Symbol | Có tồn tại ký tự “~” trong URL hay không | ExtFormAction | Hành động mẫu chứa tài nguyên ngoại vi hay không |
| Num Underscore | Số lượng ký tự gạch dưới trong URL | Abnormal Form Action | Hành động mẫu không bắt đầu bằng “http” hoặc “https” |
| Num Percent | Số lượng ký tự “%” trong URL | PctNullSelfRedirect Hyperlinks | Phần trăm siêu liên kết tự đổi hướng rỗng. |
| NumQueryComponents | Số lượng thành phần truy vấn trong URL | Frequent Domain Name Mismatch | Tần suất không khớp tên miền |
| NumAmpersand | Số lượng ký tự “&” trong URL | FakeLinkInStatusBar | Siêu liên kết giả mạo trong thanh trạng thái |
| Num Hash | Số lượng ký tự “#” trong URL | RightClickDisabled | Không cho phép bấm chuột phải trên trang web |
| NumNumericChars | Số lượng ký tự số trong URL | PopUpWindow | Trang web sử dụng cửa sổ popup |
| No HTTPS | URL có sử dụng giao thức HTTPS hay không | SubmitInfoToEmail | Trang web gửi thông tin đăng nhập qua email |
| Random String | Có chứa chuỗi ngẫu nhiên trong URL hay không | IframeOrFrame | Trang web chứa thẻ iframe hoặc frame |
| IP Address | URL có chứa địa chỉ IP hay không | MissingTitle | Trang web thiếu tiêu đề |
| DomainInSubdomains | Tên miền có xuất hiện trong các phần con của nó hay không | ImagesOnlyInForm | Chỉ có hình ảnh trong mẫu |
| Domain In Paths | Tên miền có xuất hiện trong đường dẫn URL hay không | SubdomainLevelRT | Số lượng phần con (subdomain) dạng thời gian thực (real-time). |
| HttpsInHostname | Phần tên miền có chứa "https" hay không | UrlLengthRT | Độ dài của URL dạng thời gian thực |
| HostnameLength | Độ dài của phần tên miền. | Pct Ext Resource Urls RT | Phần trăm URL tài nguyên ngoại vi dạng thời gian thực |
| Path Length | Độ dài của đường dẫn URL | Abnormal Ext Form ActionRT | Hành động mẫu không bắt đầu bằng “http” hoặc “https” dạng thời gian thực. |
| Query Length | Độ dài của truy vấn trong URL | Ext Meta Script Link RT | Tài nguyên ngoại vi dạng thời gian thực |
| DoubleSlashInPath | Có chứa hai dấu gạch chéo liền nhau trong đường dẫn hay không | Pct Ext Null Self Redirect Hyperlinks RT | Phần trăm siêu liên kết tự đổi hướng rỗng dạng thời gian thực |

### **3.1.2. Tập dữ liệu tự thu**

Tập dữ liệu tự thu của tôi có tên là *“NTU\_phishing.csv”* được dùng để kiểm thử mô hình gồm 100 mẫu được tôi thu thập từ 100 trang web khác nhau, trong đó gồm 50 trang web non-phishing (Bảng 3.2) được lấy ngẫu nhiên và 50 trang web phishing (Bảng 3.3) được lấy từ trang PhishTank. Trên tập dữ liệu tự thu cũng gồm 48 thuộc tính. Các số liệu của các đặc trưng được tôi phân tích thủ công dựa trên cơ sở từ các đặc trưng của tập dữ liệu huấn luyện.

Bảng 3.2 50 trang web non-phishing

|  |  |
| --- | --- |
| **STT** | **URL của các trang web non-phishing được lấy ngẫu nhiên** |
|  | https://elearning.ntu.edu.vn/ |
|  | https://soha.vn/bao-moi.html |
|  | https://translate.google.com/?hl=vi |
|  | https://pethouse.com.vn/cho-kangal-giong-cho-co-luc-can-manh-nhat-the-gioi/ |
|  | https://phoxanhonline.com/ |
|  | https://thoitiet.edu.vn/khanh-hoa/ninh-hoa |
|  | https://nettruyenx.com/ |
|  | https://vuighe3.com/dragon-ball-z-battle-of-gods/full |
|  | https://www.netflix.com/vn/ |
|  | https://dictionaryblog.cambridge.org/2024/04/03/talking-about-plants/ |
|  | https://www.dictionary.com/e/the-new-meaning-of-preppy/ |
|  | https://www.thesaurus.com/e/grammar/tis-the-season/ |
|  | https://www.saga.vn/chuyen-muc/ung-dung-cntt |
|  | https://www.imf.org/en/publications/weo |
|  | https://nhac.vn/album/tuyet-pham-10-bai-hat-nhac-tre-hay-nhat-moi-thoi-dai-plKao?st=soKD3p |
|  | https://dtruyen.com/bay-buoc-toi-mua-he/ |
|  | https://www.lottecinemavn.com/LCHS/Contents/Movie/Movie-List.aspx |
|  | https://play.google.com/store/apps |
|  | https://www.facebook.com/signup |
|  | https://github.com/ |
|  | https://truyenngan.com.vn/ |
|  | https://thuthuat.taimienphi.vn/website-rut-gon-link-3573n.aspx |
|  | https://aztest.vn/index.php/tin-tuc/thong-bao/ |
|  | https://nhandan.vn/gio-ve-ngang-can-bep-post793914.html |
|  | https://truyencuoihay.vn/truyen-trang-quynh/dau-to-bang-cai-bo |
|  | https://congdongshop.com/truyen-trang-quynh-full-tron-bo-gia-re/ |
|  | https://blogtruyenmoi.com/c781998/tddv-tap-117 |
|  | https://vtv.vn/truyen-hinh-tu-truyen/vtv3.htm |
|  | https://htv3tv.vn/ |
|  | https://vieon.vn/truyen-hinh-tu-truyen/htv3/ |
|  | https://hplus.com.vn/xem-kenh-htv2-hd-2669.html |
|  | https://kenh14.vn/musik.chn |
|  | https://vietteltelecom.vn/ |
|  | https://vinamilk.com.vn/vi/nhan-hieu |
|  | https://scratch.mit.edu/ |
|  | https://iq.com/album/sat-thu-nhan-tao-phan-2-mau-vat-con-lai-2022-1frz1a7qjt4?lang=vi\_vn |
|  | https://zigavn.com/ |
|  | https://playok.com/vi/cotuong/ |
|  | https://waka.vn/ebook/dac-nhan-tam-dale-carnegie-bYoKVW.html |
|  | https://kaggle.com/ |
|  | https://thetealmango.com/technology/6g-network/ |
|  | https://thinkgeo.com/blog/dijkstras-routing-algorithm |
|  | https://linkedin.com/pulse/ospf-open-short-path-first-routing-protocol-using-dijkstra-kohad |
|  | https://tuyencongchuc.vn/thong-bao/bo-noi-vu-tuyen-dung-cong-chuc-nam-2024/ |
|  | https://kplus.vn/ |
|  | https://cgv.vn/default/aquaman-2.html |
|  | https://taimienphi.vn/download-arduino-21997 |
|  | https://vi.wikipedia.org/wiki/Chanel |
|  | https://gucci.com/ |
|  | https://moi.com.vn/ |

Bảng 3.3 50 trang web phishing

|  |  |
| --- | --- |
| **STT** | **URL của các trang web phishing được lấy trên trang PhishTank** |
|  | https://brindesserasaexperian.com.br/wpse-login.p?redirect\_to=https%3A... |
|  | https://briolopacks.com/mnm1146/swps/GLOBE/3701/?dom=track.pliolaktechs.com&m1=&m2=&m3=&m4=Cum%20Phuong%20Mai&m5=&vr=logo# |
|  | https://secure.na3.documents.adobe.com/public/esign?tsid=CBFCIBAACBSCTBABDUAAABACAABAAUlT3DK7KDsqpOl-YmB2PhoTTB\_8kMY9OBh6PdxlEsnwYCpvfWc9SRzcXhc8c5GkfCnxVNQOLKzqykQqFuWbBF2qNlaLEmJnk3WVDDnnjsyVdPN0x0LdCb5O-Y36-ohFq&; |
|  | https://lonos.se.de.stossberger.de/kundenservice-m |
|  | https://my-site-101236-102129.weeblysite.com/ |
|  | https://future-energy-orlen.com/ |
|  | https://jeffwealth.com/index/login/login/token/dc737dfb0b882535dff63a0... |
|  | https://agodahotels.live/index/index/home.html#https://santander.co.u... |
|  | https://bitflyer.life/#/pages/base/index/?shiny... |
|  | https://f91cdb-2.myshopify.com/pages/inicio |
|  | https://top.ofertatv4k-magazineluiza.com/?product=2EFoRsgPYmftBdpN81N5... |
|  | https://segurosvehiculos.co/?gad\_source=1&%3Bgclid=EAIaIQobChMI15aa... |
|  | https://segurosysoatmotoscarros.com/ |
|  | https://donativos.marymount.edu.mx/wp-content/themes/ocawil/?=... |
|  | https://supercaffeineoficial.shop/ |
|  | https://innowacje-w-osge.com/ |
|  | https://indenizacao-digital.com/login-serasa/ |
|  | https://aseguratuvehiculosura.com/sura/soat/ |
|  | https://mvia.app/?gad\_source=1&gclid=CjwKCAjw\_LOwBhBFEiwAmSEQAYpPj... |
|  | https://court-order.winstonhausfeld.com/ |
|  | https://form-us1-461222.public.500apps.org/forms/2... |
|  | https://ruekrew.com/girl/courtney-thompson |
|  | https://disq.us/?url=https%3A%7S%2Fwww.gepard.ru%9Flogin%2Daccount%2F&key=JgDq7zS0Gftd6srC39cdig |
|  | https://paxful-club.com/#/ |
|  | https://pancake-projclaim.vercel.app/ |
|  | https://bank-raiffeisen.com/de/receive/bank/raiffeisen/232939398... |
|  | https://lien-de-securite.org/MB/auth/info.php |
|  | https://free.mymapdaypress.com/index.jhtml?partner=^CQ6^xdm870&amp;s2=1411767797878521346&amp;s1=714041 |
|  | https://espace-oney.fr.service.ajour.specialtralgaming.com/login/95401f0bbc7489a8899e831bf52488c6/ |
|  | https://bw-webupdate.app/ |
|  | https://1.jornalnot2.com/ |
|  | https://mlcr0sof7-off1c017.typeform.com/to/ubjpx5/ |
|  | https://digitale-rakbank.com/app/loginos.php |
|  | https://gamestop-memes.web.app/ |
|  | https://cake-magpiexyz.pages.dev/ |
|  | https://viadeo.journaldunet.com/ |
|  | https://by-pay.net/ |
|  | https://eudoraoficiall.store/ |
|  | https://playpix-fortune.com/ |
|  | https://playpix-plataforma.com/ |
|  | https://playpix-baixar.com/ |
|  | https://ofertas-exclusivas-cuponsde-abril.com/542871936... |
|  | https://service-onlline.org/YNG/auth/info.php |
|  | http://64.23.202.212/oboticario/ |
|  | https://fragali.sistema2.com.br/?amp%3Butm\_source=gotex&%3Butm\_medi... |
|  | https://login-htx.life/ |
|  | https://70off.outletmicado.com/products/7929528811710?variant=49348307... |
|  | https://ambient-finance.com/ |
|  | http://fragranciaweb.pro/ |
|  | https://xpertwebinfotech.com/script/index.html |

### **3.1.3. Trực quan hóa và làm sạch dữ liệu trước khi huấn luyện**

Trước khi bắt đầu việc đưa dữ liệu vào huấn luyện mô hình, quá trình trực quan hóa và làm sạch dữ liệu đóng vai trò quan trọng không thể phủ nhận. Bằng cách xử lý và biểu diễn dữ liệu một cách trực quan sẽ giúp hiểu rõ hơn về đặc điểm, phân phối cũng như cấu trúc của tập dữ liệu.

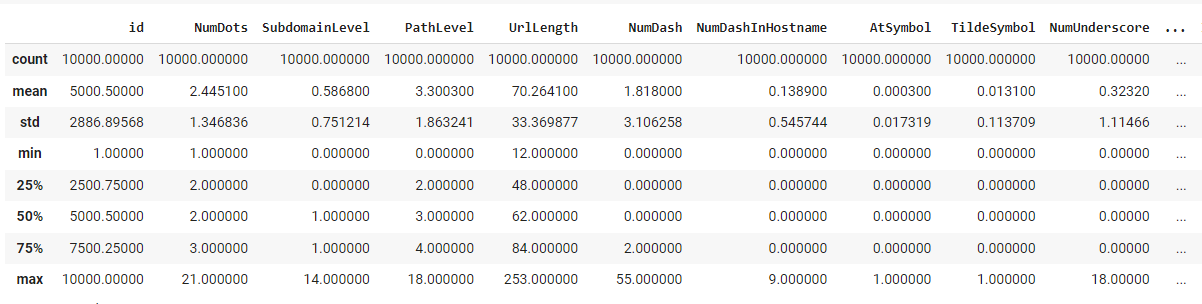
Trong tập dữ liệu của mình, tôi đã tiến hành kiểm tra và hiển thị loại dữ liệu của từng thuộc tính. Điều này hữu ích để đảm bảo các thuộc tính đang được xử lý đúng cách. Kết quả trong Hình 3.1 cho thấy trong 48 thuộc tính chỉ có 3 thuộc tính sử dụng kiểu dữ liệu *“float64”* là *“PctExtHyperlinks”*, *“PctExtResourceUrls”* và *“PctNullSelfRedirectHyperlinks”* ngoài ra tất cả các thuộc tính còn lại đều dùng kiểu dữ liệu là *“int64”*.

|  |  |
| --- | --- |
|  |  |

Hình 3.1 Kiểu dữ liệu của 48 thuộc tính

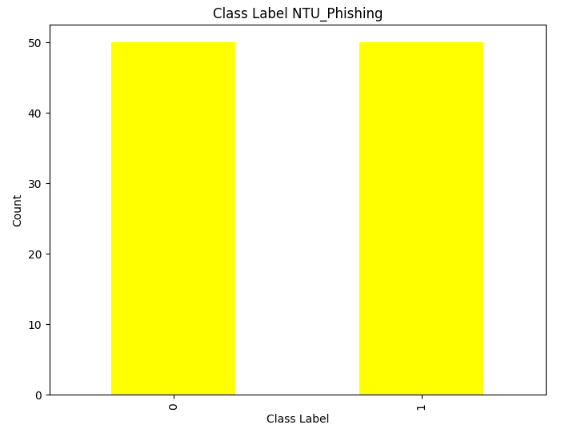
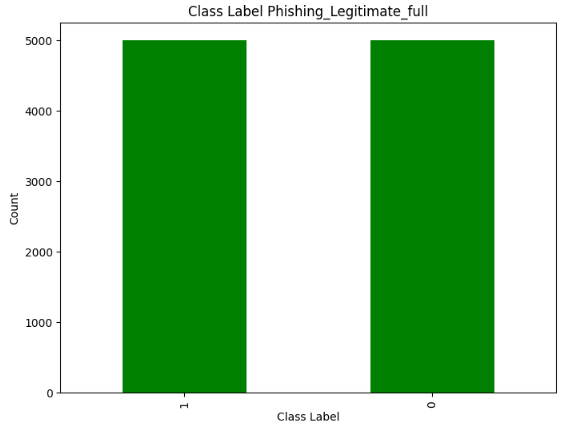
Ngoài ra tôi đã tạo một bảng tóm tắt thống kê từng thuộc tính của tập dữ liệu. Thống kê này bao gồm các thông tin như số lượng mẫu (count), giá trị trung bình (mean), độ lệch chuẩn (standard deviation), giá trị nhỏ nhất (min), giá trị lớn nhất (max), và các phần centile (25%, 50%, 75%). Việc này có thể tạo một cái nhìn tổng quan về phân phối của dữ liệu, chẳng hạn như đối với thuộc tính NumDots (Hình 3.2):

* *“Count = 10000”*: Cho biết có tổng cộng 10000 mẫu cho thuộc tính NumDots.
* *“Mean = 2.445100”*: Cho biết giá trị trung bình của tất cả các giá trị của các mẫu trong thuộc tính NumDots.
* *“Std = 1.346836”*: Đây là độ lệch chuẩn, một thước đo mức độ phân tán của các giá trị xung quanh giá trị trung bình. Độ lệch chuẩn này cho thấy các giá trị có mức độ phân tán khá lớn xung quanh giá trị trung bình của thuộc tính NumDots.
* *“Min = 1.000000”*: Thông tin này cho thấy trong thuộc tính NumDots, mẫu có giá trị nhỏ nhất là bằng 1.
* *“25% = 2.000000”*: Đây là giá trị tại phân vị thứ nhất (25% của dữ liệu). Điểu này cho thấy 25% các giá trị đầu tiên của thuộc tính NumDots nhỏ hơn hoặc bằng 2.
* *“50% = 2.000000”*: Đây là giá trị trung vị của dữ liệu. Điểu này cho thấy 50% các giá trị của thuộc tính NumDots nhỏ hơn hoặc bằng 2.
* *“75% = 3.000000”*: Đây là giá trị tại phân vị thứ ba (75% của dữ liệu). Điểu này cho thấy 75% các giá trị của thuộc tính NumDots nhỏ hơn hoặc bằng 3.
* *“Max = 21.000000”*: Thông tin này cho thấy trong thuộc tính NumDots, mẫu có giá trị nhỏ nhất là bằng 21.



Hình 3.2 Thống kê một số thuộc tính của tập dữ liệu

Một tập dữ liệu bị mất cân bằng có thể ảnh hưởng tiêu cực đến mô hình học máy bằng cách khiến mô hình thiên vị về phía lớp chiếm ưu thế, điều này làm giảm khả năng nhận diện và phân loại chính xác các mẫu thuộc lớp thiểu số, gây ra các quyết định sai lệch trong thực tế. Vì vậy để không làm ảnh hưởng đến khả năng nhận diện cũng như phân loại của các mô hình, các tập dữ liệu mà tôi dùng trong nghiên cứu này đều là tập dữ liệu cân bằng (Hình 3.4) với mỗi lớp có tỷ lệ 50:50.



Hình 3.3 Biểu đồ cột thể hiện nhãn của các tập dữ liệu

Làm sạch dữ liệu là quá trình quan trọng trong quy trình chuẩn bị dữ liệu trước khi đưa vào huấn luyện mô hình, nó sẽ loại bỏ các thuộc tính gây lỗi trong tập dữ liệu để đảm bảo cho kết quả của việc huấn luyện mô hình được dựa trên dữ liệu tin cậy và chính xác. Trong nghiên cứu của nhóm tác giả [31], một số lỗi thường gặp trong tập dữ liệu như: dữ liệu thiếu (missing data), dữ liệu nhiễu (noise), dữ liệu không đồng nhất (inconsistent data), dữ liệu dư thừa (redundant data). Đối với tập dữ liệu của tôi, trước khi đưa vào huấn luyện cũng như kiểm thử mô hình, tôi cần loại bỏ hai thuộc tính để tránh gây nhiễu trong lúc thực hiện đó là thuộc tính *“ID”* và *“Class\_Label”*. Đồng thời tôi cũng tạo một biến để lưu trữ các giá trị của thuộc tính *“Class\_Label”* có trong tập dữ liệu, mục đích là để biến này sẽ đóng vai trò là biến mục tiêu cho việc dự đoán của các mô hình huấn luyện. Tôi cũng tiến hành chuẩn hóa dữ liệu để đảm bảo rằng các thuộc tính có cùng phạm vi hoặc phân phối tương tự nhau, cụ thể trong bài của tôi các thuộc tính sẽ có giá trị trung bình gần bằng 0 và độ lệch chuẩn gần bằng 1, điều này giúp cho mô hình học máy hội tụ nhanh hơn và làm cho các trọng số của mô hình dễ đọc và dễ hiểu hơn.

## **3.2. HUẤN LUYỆN MÔ HÌNH**

Trong phần này tôi sẽ tiến hành huấn luyện các mô hình trên tập dữ liệu *“Phishing\_Legitimate\_full.csv”* ở các trường hợp: đầy đủ tất cả các thuộc tính; 30 thuộc tính quan trọng dựa trên hệ số tương quan; 30 thuộc tính quan trọng dựa trên thuật toán XGBoost. Thư viện mà tôi dùng cho việc huấn luyện các mô hình là scikit-learn, thư viện scikit-learn [32] là một trong những thư viện phổ biến và mạnh mẽ nhất hỗ trợ cho các mô hình học máy dùng ngôn ngữ Python, nó hỗ trợ nhiều công cụ và thuật toán hiệu quả cho học máy và mô hình thống kê bao gồm giảm chiều dữ liệu (dimensionality reduction), phân cụm (clustering), hồi quy (regression), phân loại (classification). Sau khi hoàn thành huấn luyện các mô hình ở các trường hợp, tôi sẽ tiến hành tổng hợp, đánh giá và so sánh kết quả thông qua các thông số của từng mô hình. Trong bài nghiên cứu này, các thông số [33] mà tôi đã sử dụng để đánh giá mô hình là:

- Accuracy (độ chính xác): Là tỷ lệ dự đoán đúng trên tổng số trường hợp của dữ liệu. Được tính theo công thức (1):

*Accuracy =* **(1)**

- Recall (tỉ lệ nhớ): Đo lường tỷ lệ của dự đoán dương tính thật so với các trường hợp được dự đoán cùng một nhãn. Được tính theo công thức (2):

*Recall* = **(2)**

- Precision: Đo lường tỷ lệ của dự đoán dương tính thật so với các dự đoán dương tính. Được tính theo công thức (3):

*Precision* = **(3)**

- F1 Score: Là sự kết hợp giữa Precision và Recall, cung cấp một đánh giá tổng thể về hiệu suất của mô hình. Được tính theo công thức (4):

*F1 Score* = **(4)**

### **3.2.1. Huấn luyện mô hình với các thuộc tính đầy đủ**

Trong phần này tôi sẽ tiến hành huấn luyện các mô hình trên tập dữ liệu Phishing\_Legitimate\_full với đầy đủ các thuộc tính (Hình 3.4).

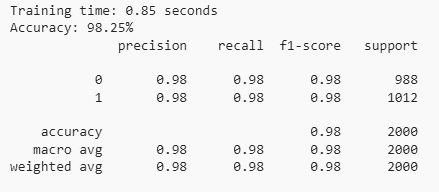
|  |  |
| --- | --- |
|  |  |

Hình 3.4 Tập dữ liệu với đầy đủ các thuộc tính

#### **3.2.1.1. Huấn luyện mô hình Random Forest**

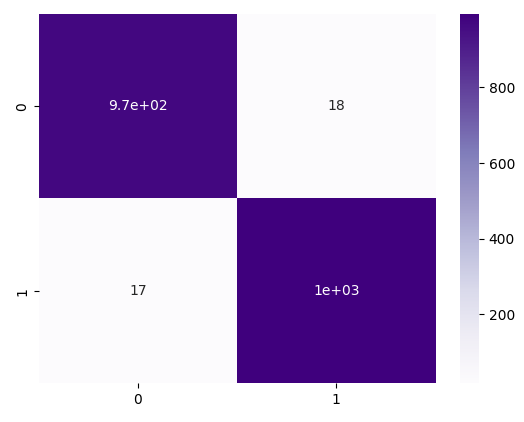
Tôi sẽ huấn luyện mô hình Random Forest với tham số “*n\_estimators = 100”* và tham số *“random\_state = 42”*. Hai tham số này góp phần quan trọng đến độ chính xác của mô hình khi huấn luyện. Đối với tham số *“n\_estimators”*, nó xác định số lượng cây quyết định mà mô hình Random Forest sử dụng, tham số này càng cao thì thường giúp cải thiện độ chính xác của mô hình, bởi vì với nhiều cây hơn, mô hình có nhiều cơ hội hơn để khắc phục các lỗi của từng cây đơn lẻ, tuy nhiên sau một mức nào đó việc tăng thêm số lượng cây có thể không còn mang lại cải thiện đáng kể về độ chính xác mà thay vào đó làm cho thời gian huấn luyện của mô hình lâu hơn và có thể dẫn đến overfitting. Trong trường hợp này mô hình Random Forest của tôi sẽ tạo ra và sử dụng 100 cây quyết định để huấn luyện và đưa ra dự đoán. Tham số *“random\_state”* được tôi thiết lập để đảm bảo tính nhất quán của kết quả giữa các lần chạy của mô hình.

Quá trình sau đó tôi đã tiến hành huấn luyện mô hình trên tập dữ liệu Phishing\_Legitimate\_full đã được chia tỷ lệ trước đó, quá trình này được tôi ghi lại khoảng thời gian mà mô hình đã mất để được huấn luyện. Sau khi kết thúc quá trình huấn luyện tôi sẽ tiến hành cho mô hình dự đoán trên tập testing để thu kết quả. Kết quả tôi thu được cho việc dự đoán của mô hình Radom Forset: thời gian huấn luyện là 0,85 giây và độ chính xác là 98,25%. Ngoài ra các thông số cụ thể khác như precision, recall, f1-score, support đã được tôi ghi lại ở Hình 3.5, các thông số này phản ánh kết quả dự đoán của mô hình ở các nhãn lớp 0 và 1.



Hình 3.5 Kết quả huấn luyện mô hình Random Forest

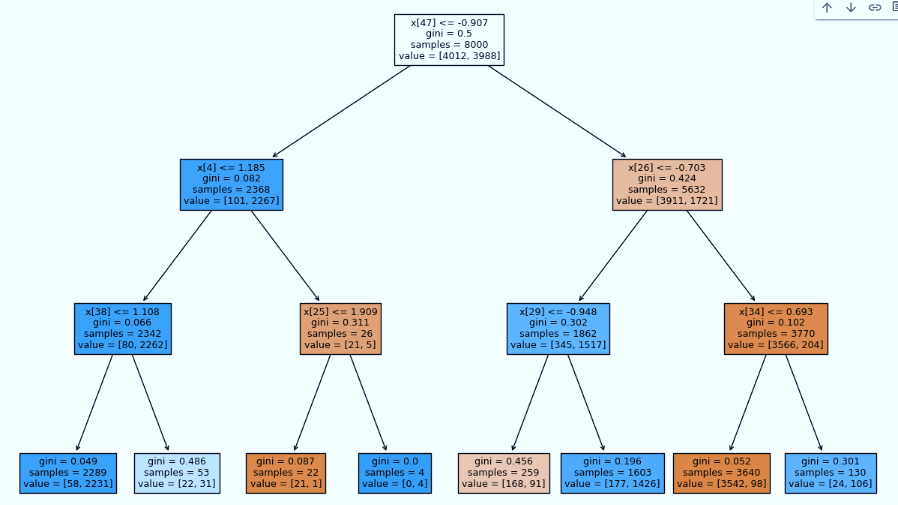
Bên cạnh đó dựa vào biểu đồ ma trận nhầm lẫn (Hình 3.6) có thể thấy tổng số mẫu mà mô hình đã dự đoán sai là 35. Cụ thể, kết hợp với giá trị *“support”* của mỗi nhãn lớp trong kết quả của Hình 3.5 có thể thấy, trong 1012 mẫu được dự đoán thuộc nhãn lớp 1 thì mô hình dự đoán sai 17 mẫu và trong 988 mẫu được dự đoán thuộc nhãn lớp 0 thì mô hình dự đoán sai 18 mẫu.



Hình 3.6 Biểu đồ ma trận nhầm lẫn của mô hình huấn luyện Random Forest

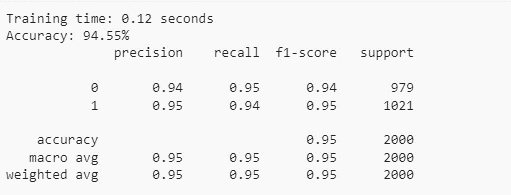
#### **3.2.1.2. Huấn luyện mô hình Decision Tree**

Với mô hình Decision Tree tôi cũng tiến hành huấn luyện mô hình với tham số giới hạn độ sâu *“max\_depth = 3”*. Với tham số này, mỗi nút trong cây quyết định chỉ được phép phân chia tối đa là 3 lần. Điều này giúp hạn chế hiện tượng quá mức phức tạp của mô hình (overfitting) bằng cách giảm số lượng quyết định được thực hiện trong cây. Trong mô hình Decision Tree, tôi đã tiến hành vẽ một cây quyết định (Hình 3.7), trong đó mỗi nút biểu diễn cho một quyết định dựa trên một đặc trưng và giá trị của nút, cùng với các dấu chấm và màu sắc để thể hiện nhãn dữ liệu.



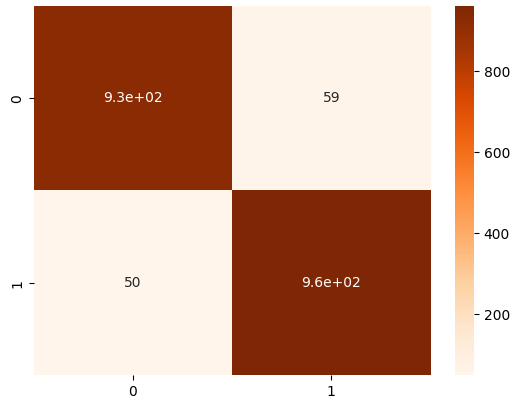
Hình 3.7 Cây quyết định của mô hình Decision Tree

Kết quả tôi thu được cho việc dự đoán của mô hình Decision Tree trên tập testing sau khi mô hình đã được huấn luyện trên tập training là: thời gian huấn luyện là 0.12 giây và độ chính xác là 94,55%. Ngoài ra các thông số cụ thể khác như precision, recall, f1-score, support đã được tôi ghi lại ở Hình 3.8, các thông số này phản ánh kết quả dự đoán của mô hình ở các nhãn lớp 0 và 1.



Hình 3.8 Kết quả huấn luyện mô hình Decision Tree

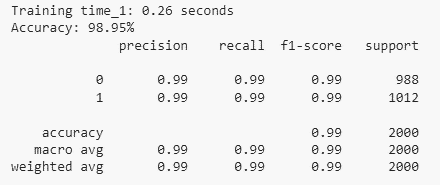
Bên cạnh đó dựa vào biểu đồ ma trận nhầm lẫn (Hình 3.9) có thể thấy tổng số mẫu mà mô hình đã dự đoán sai là 109, nhiều hơn hơn so với mô hình Random Forest. Cụ thể, kết hợp với giá trị *“support”* của mỗi nhãn lớp trong kết quả của Hình 3.8 có thể thấy, trong 1021 mẫu được dự đoán thuộc nhãn lớp 1 thì mô hình dự đoán sai 50 mẫu và trong 979 mẫu được dự đoán thuộc nhãn lớp 0 thì mô hình dự đoán sai 59 mẫu.



Hình 3.9 Biểu đồ ma trận nhầm lẫn của mô hình huấn luyện Decision Tree

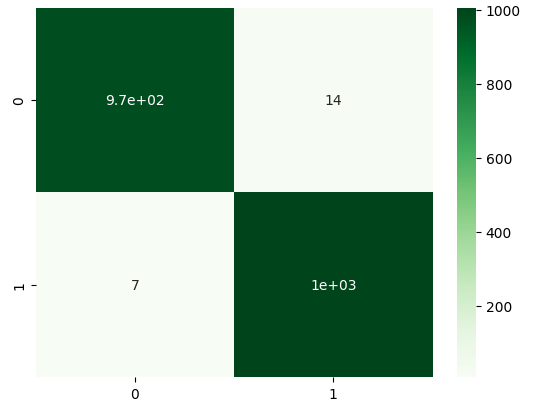
#### **3.2.1.3. Huấn luyện mô hình XGBoost**

Tương tự như hai mô hình trên tôi cũng tiến hành import thư viện, tạo và huấn luyện mô hình XGBoost trên tập dữ liệu Phishing\_Legitimate\_full. Sau khi đã huấn luyện mô hình trên tập training, tôi cũng sẽ cho mô hình dự đoán trên tập testing. Kết quả tôi thu được cho việc dự đoán của mô hình XGBoost là: thời gian huấn luyện là 0.26 giây và độ chính xác là 98,95%. Ngoài ra các thông số cụ thể khác như precision, recall, f1-score, support đã được tôi ghi lại ở Hình 3.10, các thông số này phản ánh kết quả dự đoán của mô hình ở các nhãn lớp 0 và 1.



Hình 3.10 Kết quả huấn luyện mô hình XGBoost

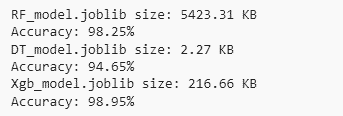
Tương tự dựa vào biểu đồ ma trận nhầm lẫn (Hình 3.11) có thể thấy tổng số mẫu mà mô hình đã dự đoán sai là 21, thấp nhất trong cả ba mô hình huấn luyện. Cụ thể, kết hợp với giá trị *“support”* của mỗi nhãn lớp trong kết quả của Hình 3.10 có thể thấy, trong 1012 mẫu được dự đoán thuộc nhãn lớp 1 thì mô hình dự đoán sai 7 mẫu và trong 988 mẫu được dự đoán thuộc nhãn lớp 0 thì mô hình dự đoán sai 14 mẫu.



Hình 3.11 Biểu đồ ma trận nhầm lẫn của mô hình huấn luyện XGBoost

#### **3.2.1.4. Xuất các mô hình và tổng hợp kết quả huấn luyện**

Sau khi đã hoàn tất việc huấn luyện tất cả mô hình, tôi sẽ tiến hành xuất các mô hình huấn luyện để có thể sử dụng chúng cho việc kiểm thử trên tập dữ liệu tự thu của mình. Tôi sử dụng hàm *“dump”* từ module *“joblib”* để tiến hành xuất và lưu trữ các mô hình huấn luyện. Các mô hình Random Forest, Decision Tree và XGBoost được xuất tương ứng với các tên file trong trường hợp này là *“RF\_model.joblib”*; *“DT\_model.joblib”*; *“Xgb\_model.joblib”*, kích thước của các mô hình sau khi được xuất lần lượt là 5423KB; 2,27KB và 217KB.



Hình 3.12 Tên và kích thước của các mô hình sau khi được xuất

Ngoài ra, dựa vào kết quả dự đoán của các mô hình huấn luyện, tôi đã tiến hành thống kê lại tất cả số liệu vào Bảng 3.4 để có thể đánh giá và so sánh các thông số dự đoán giữa các mô hình.

Bảng 3.4 So sánh các thông số của các mô hình được huấn luyện trên tập dữ liệu đầy đủ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Thông số** | **Accuracy**  **(%)** | **Precision** | | **Recall** | | **F1 – score** | |
| 0 | 1 | 0 | 1 | 0 | 1 |
| **Random Forest**  **Mô hình** | 98,25% | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 |
| **Decision Tree** | 94,55% | 0.94 | 0.95 | 0.95 | 0.94 | 0.94 | 0.95 |
| **XGBoost** | 98,95% | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 | 0.99 |

* Cả ba mô hình đều có độ chính xác cao, với XGBoost đạt được độ chính xác cao nhất (98.95%), theo sau là Random Forest (98.25%) và Decision Tree (94.55%).
* Trong ba mô hình có thể thấy mô hình XGBoost là mô hình có giá trị tỷ lệ của các thông số là cao nhất. Thông số precision ở cả lớp 0 và 1 đều là 0.99, điều này cho thấy các dự đoán dương tính giả của mô hình ở hai lớp này rất thấp. Tương tự thông số recall ở cả hai lớp 0 và 1 cũng đều là 0.99, cho thấy các dự đoán âm tính giả của mô hình ở hai lớp này cũng rất thấp.
* F1 – score là một phép đo tổng hợp của precision và recall. Cả ba mô hình đều có F1 – score cao, trong đó XGBoost tỷ lệ cao nhất (0.99), tiếp theo là Random Forest (0.98) và Decision Tree (0.94 đối với nhãn 0 và 0.95 đối với nhãn 1).

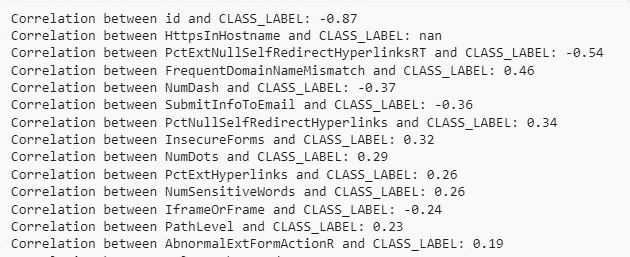
Có thể thấy, cả ba mô hình được huấn luyện trong trường hợp này đều cho độ chính xác cao và phân loại tốt giữa các mẫu phishing và non-phishing. Trong đó, XGBoost là mô hình có độ chính xác cao nhất và khả năng phân loại giữa các mẫu là tốt nhất.

### **3.2.2. Huấn luyện mô hình với 30 thuộc tính dựa trên hệ số tương quan**

#### **3.2.2.1. Hệ số tương quan và tập dữ liệu gồm các thuộc tính được trích chọn**

Như mục đích đã được đề cập là nhằm tối ưu kích thước và thời gian huấn luyện của các mô hình, trong tập dữ liệu tôi sẽ tiến hành chọn ra 30 thuộc tính có giá trị hệ số tương quan cao nhất để huấn luyện mô hình. Hệ số tương quan là một thống kê mô tả mức độ tương quan tuyến tính giữa hai biến ngẫu nhiên. Hệ số này nằm trong khoảng từ -1 đến 1, với các giá trị càng gần 1 hoặc -1 thì thể hiện mối tương quan mạnh mẽ hơn.

Trong tập dữ liệu, để chọn ra được 30 thuộc tính dựa trên hệ số tương quan tôi đã tiến hành tính toán giá trị tương quan giữa thuộc tính *“Class\_Label”* với từng thuộc tính khác. Hệ số tương quan này cho biết mối quan hệ tuyến tính giữa các đặc trưng và nhãn. Điều này có thể giúp hiểu được đặc tính quan trọng nào ảnh hưởng nhiều đến nhãn trong dữ liệu.



Hình 3.13 Một số giá trị tương quan giữa các đặc trưng với nhãn label

Sau khi có danh sách giá trị hệ số tương quan của các đặc trưng tôi đã chọn ra 30 thuộc tính có các giá trị tương quan cao nhất trong các đặc trưng, đó là: *PctExtNullSelfRedirectHyperlinksRT, FrequentDomainNameMismatch, NumDash, SubmitInfoToEmail, PctNullSelfRedirectHyperlinks, InsecureForms, NumDots, PctExtHyperlinks, NumSensitiveWords, IframeOrFrame, PathLevel, AbnormalExtFormActionR, UrlLengthRT, HostnameLength, NumDashInHostname, NumQueryComponents, AbnormalFormAction, EmbeddedBrandName, IpAddress, DomainInPaths, MissingTitle, ExtMetaScriptLinkRT, ExtFormAction, DomainInSubdomains, NumUnderscore, TildeSymbol, RandomString, RelativeFormAction, NumAmpersand, QueryLength*. Các giá trị tương quan của các thuộc tính này đã được tôi thống kê (Bảng 3.5). Sau khi đã chọn ra được 30 thuộc tính có giá trị tương quan cao nhất, tôi sẽ tiến hành tạo một Dataframe mới loại bỏ các cột chứa các thuộc tính đặc trưng không nằm trong danh sách 30 thuộc tính được chọn để bắt đầu huấn luyện mô hình với Dataframe này.

Bảng 3.5 30 thuộc tính của hệ số tương quan

|  |  |
| --- | --- |
|  |  |

#### **3.2.2.2. Kết quả huấn luyện các mô hình khi dùng tập dữ liệu được trích chọn**

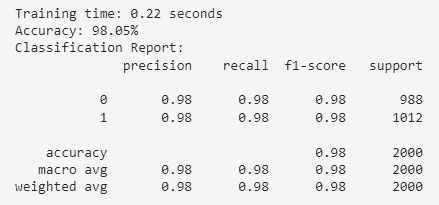
Như trên trường hợp tập dữ liệu đầy đủ các thuộc tính, các mô hình Random Forest, Decision Tree và XGBoost trong trường hợp này tôi cũng tiến hành import thư viện, tạo mô hình, huấn luyện và đánh giá mô hình trên tập dữ liệu chỉ gồm 30 thuộc tính đặc trưng dựa trên hệ số tương quan. Kết quả dự đoán của các mô hình huấn luyện được thể hiện trong Bảng 3.6

Bảng 3.6 Kết quả dự đoán của các mô hình được huấn luyện dựa trên 30 thuộc tính đặc trưng của hệ số tương quan

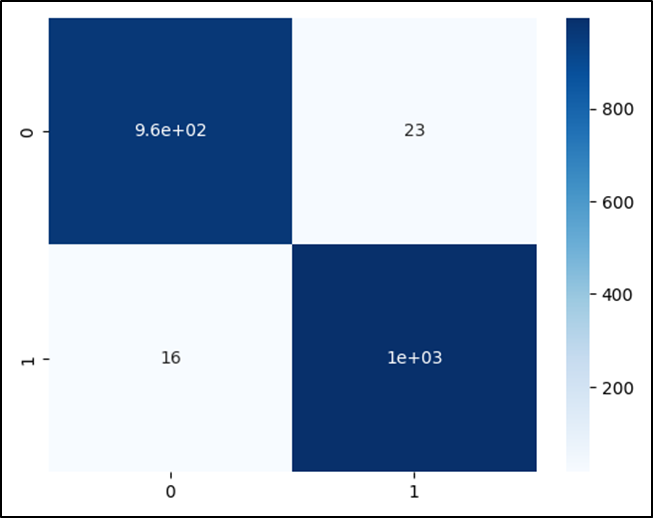
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Thông số** | **Accuracy**  **(%)** | **Precision** | | **Recall** | | **F1 – score** | |
| 0 | 1 | 0 | 1 | 0 | 1 |
| **Random Forest**  **Mô hình** | 97,70% | 0.97 | 0.98 | 0.98 | 0.97 | 0.98 | 0.98 |
| **Decision Tree** | 94,55% | 0.94 | 0.95 | 0.95 | 0.94 | 0.94 | 0.95 |
| **XGBoost** | 98,05% | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 |

* Độ chính xác của Random Forest (97,70%) và XGBoost (98,05%) đều cao hơn so với Decision Tree (94,55%), cho thấy hiệu suất dự đoán của hai mô hình này tốt hơn trên tập dữ liệu với 30 thuộc tính quan trọng dựa trên hệ số tương quan.
* Trong ba mô hình có thể thấy XGBoost là mô hình có giá trị tỷ lệ của các thông số ở các lớp là cao nhất. Thông số precision ở cả lớp 0 và 1 đều là 0.98, điều này cho thấy các dự đoán dương tính giả của mô hình ở hai lớp này rất thấp. Tương tự thông số recall ở cả hai lớp 0 và 1 cũng đều là 0.98, cho thấy các dự đoán âm tính giả của mô hình ở hai lớp này cũng rất thấp.
* Tỷ lệ của F1 – score cũng cho thấy hiệu suất của mô hình XGBoost là tốt nhất.

Dựa vào bảng so sánh trên, có thể nhận thấy rằng cả ba mô hình đều đạt được hiệu suất dự đoán tốt trên tập dữ liệu chỉ gồm 30 thuộc tính dựa trên hệ số tương quan. Trong đó có thể thấy mô hình XGBoost vẫn đạt được hiệu suất tốt nhất so với hai mô hình còn lại. Kết quả dự đoán của mô hình XGBoost (Hình 3.14) đã được tôi trực quan bằng một biểu đồ ma trận nhầm lẫn (Hình 3.15). Dựa vào biểu đồ có thể thấy tổng số mẫu mà mô hình đã dự đoán sai là 39 mẫu. Cụ thể, kết hợp với giá trị *“support”* của mỗi nhãn lớp trong kết quả của Hình 3.14 có thể thấy, trong 1012 mẫu được dự đoán thuộc nhãn lớp 1 thì mô hình dự đoán sai 16 mẫu và trong 988 mẫu được dự đoán thuộc nhãn lớp 0 thì mô hình dự đoán sai 23 mẫu.



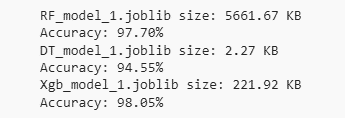
Hình 3.14 Kết quả huấn luyện mô hình XGBoost với 30 thuộc tính của hệ số tương quan



Hình 3.15 Biểu đồ ma trận nhầm lẫn mô hình XGBoost với 30 thuộc tính của hệ số tương quan

#### **3.2.2.3. Xuất các mô hình và so sánh kết quả huấn luyện**

Sau khi đã hoàn tất việc huấn luyện tất cả mô hình với 30 thuộc tính quan trọng dựa trên hệ số tương quan, tôi cũng sẽ tiến hành xuất các mô hình huấn luyện để có thể sử dụng chúng cho việc kiểm thử trên tập dữ liệu tự thu của mình. Trong đó, *“RF\_model\_1.joblib”* là mô hình Random Forest được huấn luyện với 30 thuộc tính quan trọng dựa trên hệ số tương quan; *“DT\_model\_1.joblib”* là mô hình Decision Tree được huấn luyện với 30 thuộc tính quan trọng dựa trên hệ số tương quan; *“Xgb\_model\_1.joblib”* là mô hình XGBoost được huấn luyện với 30 thuộc tính quan trọng dựa trên hệ số tương quan. Kích thước của các mô hình sau khi được xuất lần lượt là 5662KB; 2,27KB và 223KB. Có thể thấy, nếu so sánh với kích thước của các mô hình khi được huấn luyện trên tập dữ liệu đầy đủ thuộc tính thì trong trường hợp này kích thước của các mô hình có sự lớn hơn.



Hình 3.16 Tên và kích thước của các mô hình được huấn luyện với 30 thuộc tính của hệ số tương quan sau khi được xuất

Ngoài ra, dựa vào độ chính xác và thời gian huấn luyện của các mô hình khi được huấn luyện trên tập dữ liệu chỉ gồm 30 thuộc tính dựa trên hệ số tương quan, tôi đã tiến hành so sánh với độ chính xác và thời gian huấn luyện của các mô hình khi được huấn luyện trên tập dữ liệu đầy đủ thuộc tính (Bảng 3.7). Kết quả cho thấy chỉ có mô hình Decision Tree được huấn luyện trong trường hợp này có độ chính xác bằng với trường hợp huấn luyện trên tập dữ liệu đầy đủ thuộc tính, trong khi mô hình Random Forest và XGBoost có độ chính xác giảm đi. Ngoài ra thời gian huấn luyện của các mô hình trong trường hợp này cũng đã được rút ngắn.

Bảng 3.7 So sánh hiệu suất của các mô hình khi được huấn luyện trên tập dữ liệu đầy đủ thuộc tính và trên tập dữ liệu 30 thuộc tính của hệ số tương quan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mô hình**  **Trường hợp** | **Random Forest** | | **Decision Tree** | | **XGBoost** | |
| *Độ chính xác (%)* | *Thời gian huấn luyện (s)* | *Độ chính xác (%)* | *Thời gian huấn luyện (s)* | *Độ chính xác (%)* | *Thời gian huấn luyện (s)* |
| ***Tập dữ liệu***  ***đầy đủ thuộc tính*** | 98,25 | 0,85 | 94,55 | 0,12 | 98,95 | 0,26 |
| ***Tập dữ liệu với 30 đặc trưng của hệ số tương quan*** | 97,70 | 0,75 | 94,55 | 0,05 | 98,05 | 0,22 |

### **3.2.3. Huấn luyện mô hình với 30 thuộc tính quan trọng của XGBoost**

#### **3.2.3.1. Thuộc tính quan trọng và tập dữ liệu gồm các thuộc tính được trích chọn**

Tương tự với mục đích là nhằm tối ưu kích thước và thời gian huấn luyện của các mô hình, tôi cũng sẽ tiến hành tìm ra 30 thuộc tính quan trọng dựa trên thuật toán XGBoost để huấn luyện mô hình. Thuật toán XGBoost sẽ chọn ra 30 đặc trưng quan trọng dựa vào cách tính mức độ quan trọng của mỗi đặc trưng. Theo đó, thuật toán sẽ dựa trên sự đóng góp của mỗi đặc trưng vào việc cải thiện hiệu suất của mô hình trong quá trình huấn luyện. Để có được 30 thuộc tính quan trọng dựa trên thuật toán XGBoost, tôi đã tiến hành lấy các giá trị mức độ quan trọng của các đặc trưng từ mô hình XGBoost đã được huấn luyện (cụ thể là mô hình XGBoost đã được huấn luyện trên tập dữ liệu đầy đủ thuộc tính), sau đó tôi tiến hành in ra 30 thuộc tính đặc trưng quan trọng của XGBoost.

Theo đó, 30 đặc trưng quan trọng có giá trị cao nhất của XGBoost là: *PctExtNullSelfRedirectHyperlinksRT, PctExtHyperlinks, InsecureForms, NoHttps, FrequentDomainNameMismatch, SubmitInfoToEmail, NumDash, PctNullSelfRedirectHyperlinks, ExtFavicon, NumDashInHostname, PathLevel, IframeOrFrame, NumQueryComponents, ExtFormAction, NumDots, QueryLength, ExtMetaScriptLinkRT, NumSensitiveWords, NumAmpersand, NumUnderscore, PctExtResourceUrls, NumNumericChars, SubdomainLevel, NumPercent, MissingTitle, AbnormalExtFormActionR, AbnormalFormAction, ImagesOnlyInForm, RelativeFormAction, DomainInPaths*. Các giá trị mức độ quan trọng của các đặc trưng này đã được tôi thống kê trong Bảng 3.8. Sau khi đã chọn ra được 30 thuộc tính quan trọng của XGBoost, tôi sẽ tiến hành tạo một Dataframe mới loại bỏ các cột chứa các thuộc tính đặc trưng không nằm trong danh sách 30 thuộc tính được chọn để bắt đầu huấn luyện mô hình với Dataframe này.

Bảng 3.8 30 đặc trưng quan trọng của XGBoost

|  |  |
| --- | --- |
|  |  |

#### **3.2.3.2. Kết quả huấn luyện các mô hình khi dùng tập dữ liệu được trích chọn**

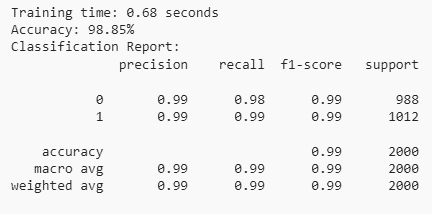
Trong phần này, các mô hình Random Forest, Decision Tree và XGBoost cũng được tôi tiến hành import thư viện, tạo mô hình, huấn luyện và đánh giá mô hình trên tập dữ liệu chỉ gồm 30 thuộc tính quan trọng của thuật toán XGBoost. Bảng 3.9 trình bày kết quả dự đoán của các mô hình huấn luyện.

Bảng 3.9 Kết quả dự đoán của các mô hình được huấn luyện dựa trên 30 thuộc tính quan trọng của XGBoost

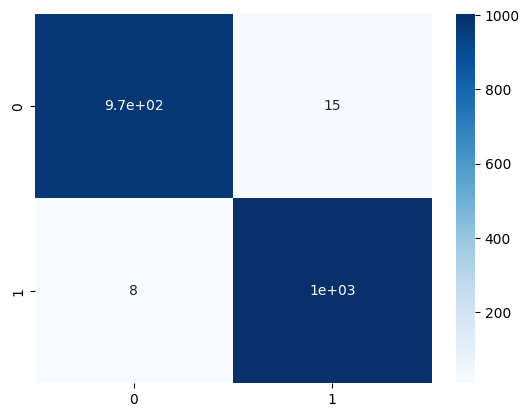
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Thông số** | **Accuracy**  **(%)** | **Precision** | | **Recall** | | **F1 – score** | |
| 0 | 1 | 0 | 1 | 0 | 1 |
| **Random Forest**  **Mô hình** | 98,25% | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 |
| **Decision Tree** | 94,65% | 0.94 | 0.95 | 0.95 | 0.94 | 0.95 | 0.95 |
| **XGBoost** | 98,85% | 0.99 | 0.99 | 0.98 | 0.99 | 0.99 | 0.99 |

* Mô hình XGBoost có độ chính xác cao nhất với 98.85%, theo sau là mô hình Random Forest với 98.25% và Decision Tree với 94.65%.
* Trong ba mô hình có thể thấy XGBoost vẫn là mô hình có giá trị tỷ lệ của các thông số ở các lớp là cao nhất. Thông số precision ở cả lớp 0 và 1 đều là 0.99, điều này cho thấy các dự đoán dương tính giả của mô hình ở hai lớp này rất thấp. Tương tự thông số recall ở lớp 0 là 0.98 và lớp 1 là 0.99 cho thấy các dự đoán âm tính giả của mô hình ở hai lớp này cũng rất thấp.
* Tỷ lệ F1 – score của XGBoost đều cao nhất trong cả ba mô hình cho cả hai lớp 0 và 1.

Dựa vào bảng so sánh trên kết hợp với Bảng 3.6 có thể thấy các mô hình huấn luyện trong trường hợp này đã đạt được độ chính xác cao hơn so với trường hợp các mô hình được huấn luyện với 30 thuộc tính được chọn dựa trên hệ số tương quan, trong đó mô hình XGBoost một lần nữa có độ chính xác nhất trong cả ba mô hình. Kết quả dự đoán của mô hình XGBoost (Hình 3.17) đã được tôi trực quan bằng một biểu đồ ma trận nhầm lẫn (Hình 3.18). Dựa vào biểu đồ có thể thấy tổng số mẫu mà mô hình đã dự đoán sai là 23 mẫu. Cụ thể, kết hợp với giá trị *“support”* của mỗi nhãn lớp trong kết quả của Hình 3.17 có thể thấy, trong 1012 mẫu được dự đoán thuộc nhãn lớp 1 thì mô hình dự đoán sai 8 mẫu và trong 988 mẫu được dự đoán thuộc nhãn lớp 0 thì mô hình dự đoán sai 15 mẫu.



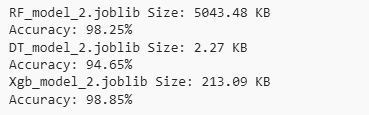
Hình 3.17 Kết quả huấn luyện mô hình XGBoost với 30 thuộc tính quan trọng của XGBoost



Hình 3.18 Biểu đồ ma trận nhầm lẫn mô hình XGBoost với 30 thuộc tính quan trọng của XGBoost

#### **3.2.3.3. Xuất các mô hình và so sánh kết quả huấn luyện**

Sau khi đã hoàn tất việc huấn luyện tất cả mô hình với 30 thuộc tính quan trọng của thuật toán XGBoost, tôi cũng sẽ tiến hành xuất các mô hình huấn luyện để có thể sử dụng chúng cho việc kiểm thử trên tập dữ liệu tự thu của mình. Trong đó, *“RF\_model\_2.joblib”* là mô hình Random Forest được huấn luyện với 30 thuộc tính quan trọng của XGBoost; *“DT\_model\_2.joblib”* là mô hình Decision Tree được huấn luyện với 30 thuộc tính quan trọng của XGBoost; *“Xgb\_model\_2.joblib”* là mô hình XGBoost được huấn luyện với 30 thuộc tính quan trọng của XGBoost. Kích thước của các mô hình sau khi được xuất lần lượt là 5044KB; 2,27KB và 213KB. Có thể thấy, nếu so sánh với kích thước của các mô hình khi được huấn luyện trên tập dữ liệu đầy đủ thuộc tính thì trong trường hợp này kích thước của các mô hình đã nhỏ hơn.



Hình 3.19 Tên và kích thước của các mô hình được huấn luyện với 30 thuộc tính quan trọng của XGBoost sau khi được xuất

Ngoài ra, dựa vào độ chính xác và thời gian huấn luyện của các mô hình khi được huấn luyện trên tập dữ liệu chỉ gồm 30 thuộc tính quan trọng của XGBoost, tôi đã tiến hành so sánh với độ chính xác và thời gian huấn luyện của các mô hình khi được huấn luyện trên tập dữ liệu đầy đủ thuộc tính (Bảng 3.10). Kết quả cho thấy chỉ có mô hình Random Forest được huấn luyện trong trường hợp này bằng với trường hợp huấn luyện trên tập dữ liệu đầy đủ thuộc tính, trong khi đó mô hình XGBoost có độ chính xác thấp hơn và mô hình Decision Tree có độ chính cao hơn. Bên cạnh đó có thể thấy thời gian huấn luyện của các mô hình trong trường hợp này lâu hơn ngoại trừ mô hình Decision Tree có thời gian huấn luyện được rút ngắn.

Bảng 3.10 So sánh hiệu suất của các mô hình huấn luyện trên tập dữ liệu 30 đặc trưng quan trọng của XGBoost với tập dữ liệu đầy đủ thuộc tính

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Mô hình**  **Trường hợp** | **Random Forest** | | **Decision Tree** | | **XGBoost** | |
| *Độ chính xác (%)* | *Thời gian huấn luyện (s)* | *Độ chính xác (%)* | *Thời gian huấn luyện (s)* | *Độ chính xác (%)* | *Thời gian huấn luyện (s)* |
| ***Tập dữ liệu***  ***đầy đủ*** | 98,25 | 0,85 | 94,55 | 0,12 | 98,95 | 0,26 |
| ***Tập dữ liệu với 30 đặc trưng của XGBoost*** | 98,25 | 1,16 | 94,65 | 0,02 | 98,85 | 0,68 |

## **3.3. KIỂM THỬ MÔ HÌNH VỚI TẬP DỮ LIỆU TỰ THU**

Trong phần này tôi sẽ tiến hành sử dụng các mô hình đã được huấn luyện ở phần trước để kiểm thử trên tập dữ liệu tự thu của mình có tên là *“NTU\_phishing.csv”*, sau đó tôi sẽ tổng hợp lại tất cả các kết quả để đánh giá và nhận xét.

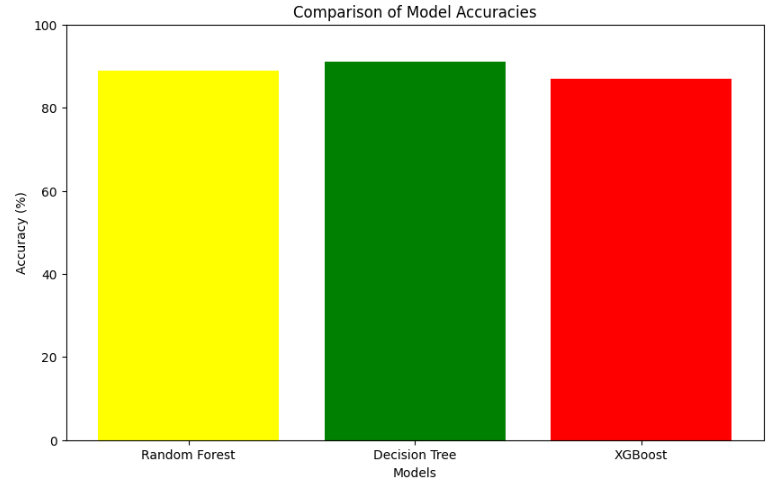
### **3.3.1. Mô hình huấn luyện đầy đủ**

Để kiểm thử trên tập dữ liệu tự thu của mình trong trường hợp này, tôi sẽ sử dụng ba mô hình là Random Forest, Decision Tree và XGBoost đã được huấn luyện trên tập dữ liệu đầy đủ thuộc tính. Ba mô hình này đã được tôi tiến hành xuất mô hình và lưu trữ với các tên file lần lượt là *“RF\_model.joblib”*; *“DT\_model.joblib”* và *“Xgb\_model.joblib”*. Sau khi tiến hành quá trình kiểm thử mô hình trên tập dữ liệu tự thu của mình, kết quả đã được tôi ghi nhận lại như trong Bảng 3.11.

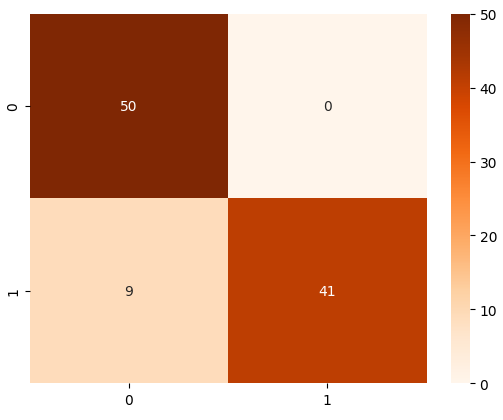
Bảng 3.11 Kết quả kiểm thử các mô hình huấn luyện trên tập dữ liệu tự thu

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mô hình**  **Lớp** | **RF\_model.joblib** | | | | **DT\_model.joblib** | | | | **Xgb\_model.joblib** | | | |
| Accuracy  (%) | Precision | Recall | F1 – score | Accuracy  (%) | Precision | Recall | F1 – score | Accuracy  (%) | Precision | Recall | F1 – score |
| 0 | 89% | 0.82 | 1.00 | 0.90 | 91% | 0.85 | 1.00 | 0.92 | 87% | 0.80 | 0.98 | 0.88 |
| 1 | 1.00 | 0.78 | 0.88 | 1.00 | 0.82 | 0.90 | 0.97 | 0.76 | 0.85 |

Từ bảng trên có thể thấy trên tập dữ liệu tự thu của tôi, mô hình đạt được độ chính xác cao nhất trong các mô hình huấn luyện được dùng trong trường hợp này là mô hình *“DT\_model.joblib”* với độ chính là 91%. Đồng thời dựa vào tỷ lệ của các thông số có thể thấy được rằng cả ba mô hình đều có khả năng dự đoán đúng cao đối với các mẫu là non-phishing khi tỷ lệ recall của nhãn lớp 0 và tỷ lệ precision của nhãn lớp 1 đều đạt được tỷ lệ cao nhất trong các tỷ lệ của các mô hình. Ngoài ra để trực quan hơn về kết quả kiểm thử của các mô hình, tôi đã vẽ một biểu đồ cột so sánh độ chính xác của từng mô hình (Hình 3.20) và một biểu đồ ma trận nhầm lẫn của mô hình dự đoán tốt nhất trong trường hợp này là *“DT\_model.joblib”* (Hình 3.21).



Hình 3.20 Biểu đồ so sánh độ chính xác của các mô hình sau khi kiểm thử trên tập dữ liệu tự thu



Hình 3.21 Biểu đồ ma trận nhầm lẫn mô hình DT\_model.joblib trên tập dữ liệu tự thu

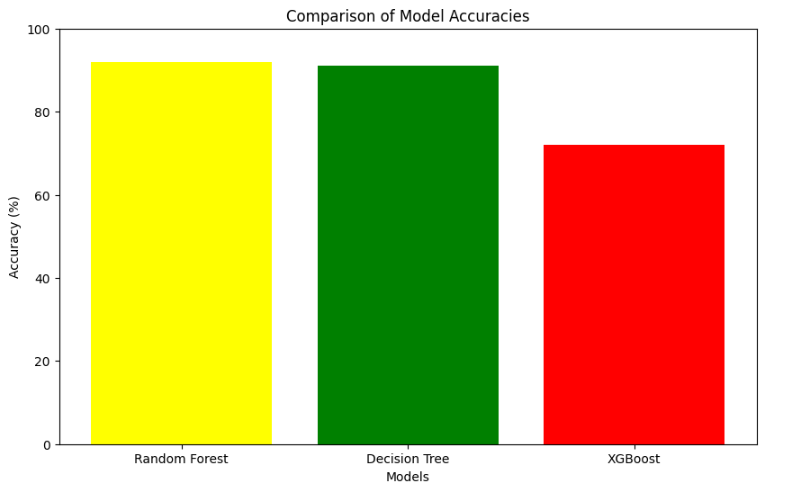
### **3.3.2. Mô hình huấn luyện sử dụng hệ số tương quan**

Ở mô hình huấn luyện sử dụng hệ số tương quan tôi sẽ sử dụng các mô hình đã được huấn luyện trên tập dữ liệu chỉ gồm 30 thuộc tính dựa trên hệ số tương quan là Random Forest, Decision Tree và XGBoost. Ba mô hình này đã được tôi tiến hành xuất mô hình và lưu trữ với các tên lần lượt là *“RF\_model\_1”*, *“DT\_model\_1”* và *“Xgb\_model\_1”*. Sau khi tiến hành quá trình kiểm thử mô hình trên tập dữ liệu tự thu của mình, kết quả đã được tôi ghi nhận lại như trong Bảng 3.12.

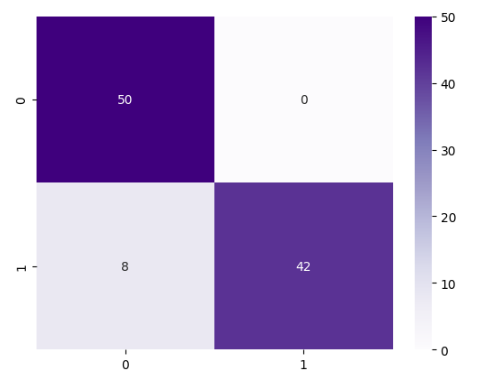
Bảng 3.12 Kết quả kiểm thử trên tập dữ liệu tự thu của các mô hình huấn luyện với 30 đặc trưng của hệ số tương quan

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mô hình**  **Lớp** | **RF\_model\_1.joblib** | | | | **DT\_model\_1.joblib** | | | | **Xgb\_model\_1.joblib** | | | |
| Accuracy  (%) | Precision | Recall | F1 – score | Accuracy  (%) | Precision | Recall | F1 – score | Accuracy  (%) | Precision | Recall | F1 – score |
| 0 | 92% | 0.86 | 1.00 | 0.93 | 91% | 0.85 | 1.00 | 0.92 | 72% | 0.64 | 1.00 | 0.78 |
| 1 | 1.00 | 0.84 | 0.91 | 1.00 | 0.82 | 0.90 | 1.00 | 0.44 | 0.61 |

Từ bảng trên có thể thấy trên tập dữ liệu tự thu của tôi, trong trường hợp mô hình đạt được độ chính xác cao nhất trong các mô hình huấn luyện được dùng là *“RF\_model\_1.joblib”* với độ chính là 92%. Đồng thời dựa vào tỷ lệ của các thông số có thể thấy được rằng cả ba mô hình đều có khả năng dự đoán đúng cao đối với các mẫu là non-phishing khi tỷ lệ recall của nhãn lớp 0 và tỷ lệ precision của nhãn lớp 1 đều đạt được tỷ lệ 100% trong các tỷ lệ của các mô hình. Ngoài ra để trực quan hơn về kết quả kiểm thử của các mô hình, tôi đã vẽ một biểu đồ cột so sánh độ chính xác của từng mô hình (Hình 3.22) và một biểu đồ ma trận nhầm lẫn của mô hình dự đoán tốt nhất trong trường hợp này là *“RF\_model\_1.joblib”* (Hình 3.23).



Hình 3.22 Biểu đồ so sánh độ chính xác của các mô hình sau khi kiểm thử trên tập dữ liệu tự thu



Hình 3.23 Biểu đồ ma trận nhầm lẫn mô hình RF\_model\_1.joblib trên tập dữ liệu tự thu

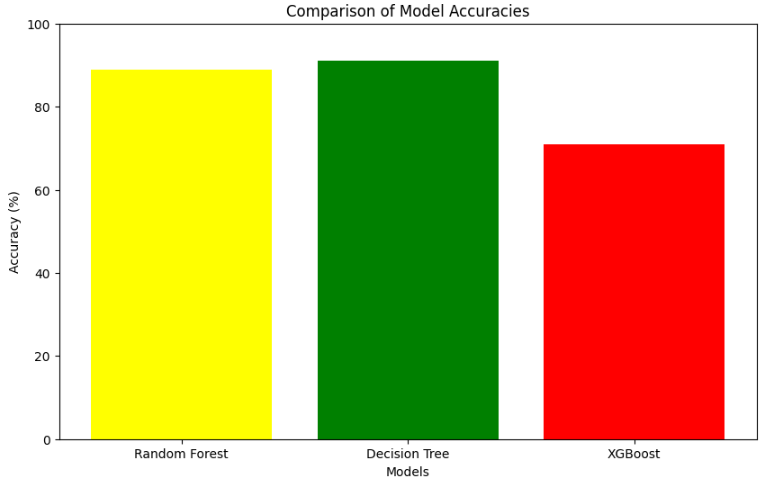
### **3.3.3. Mô hình huấn luyện sử dụng các đặc trưng quan trọng của XGBoost**

Ở mô hình huấn luyện sử dụng các đặc trưng quan trọng của XGBoost tôi sẽ sử dụng các mô hình đã được huấn luyện trên tập dữ liệu chỉ gồm 30 thuộc tính quan trọng của XGBoost là Random Forest, Decision Tree và XGBoost. Ba mô hình này đã được tôi tiến hành xuất mô hình và lưu trữ với các tên lần lượt là *“RF\_model\_2”*, *“DT\_model\_2”* và *“Xgb\_model\_2”*. Sau khi tiến hành quá trình kiểm thử mô hình trên tập dữ liệu tự thu của mình, kết quả đã được tôi ghi nhận lại như trong Bảng 3.13.

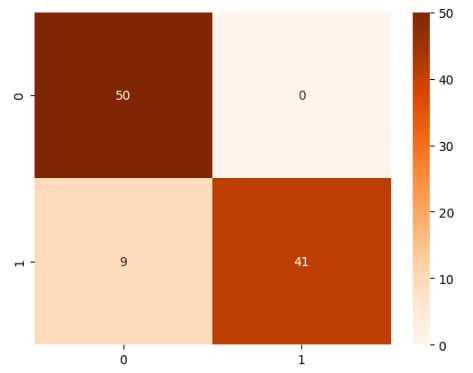
Bảng 3.13 Kết quả kiểm thử trên tập dữ liệu tự thu của các mô hình huấn luyện với 30 đặc trưng của XGBoost

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mô hình**  **Lớp** | **RF\_model\_2.joblib** | | | | **DT\_model\_2.joblib** | | | | **Xgb\_model\_2.joblib** | | | |
| Accuracy  (%) | Precision | Recall | F1 – score | Accuracy  (%) | Precision | Recall | F1 – score | Accuracy  (%) | Precision | Recall | F1 – score |
| 0 | 89% | 0.82 | 1.00 | 0.90 | 91% | 0.85 | 1.00 | 0.92 | 71% | 0.63 | 1.00 | 0.78 |
| 1 | 1.00 | 0.78 | 0.88 | 1.00 | 0.82 | 0.90 | 1.00 | 0.42 | 0.59 |

Từ bảng trên có thể thấy trên tập dữ liệu tự thu của tôi, mô hình đạt được độ chính xác cao nhất trong các mô hình huấn luyện được dùng trong trường hợp này là mô hình *“DT\_model\_2.joblib”* với độ chính là 91%. Đồng thời dựa vào tỷ lệ của các thông số có thể thấy được rằng cả ba mô hình đều có khả năng dự đoán đúng cao đối với các mẫu là non-phishing khi tỷ lệ recall của nhãn lớp 0 và tỷ lệ precision của nhãn lớp 1 đều đạt được tỷ lệ 100% trong các tỷ lệ của các mô hình. Ngoài ra để trực quan hơn về kết quả kiểm thử của các mô hình, tôi đã vẽ một biểu đồ cột so sánh độ chính xác của từng mô hình (Hình 3.24) và một biểu đồ ma trận nhầm lẫn của mô hình dự đoán tốt nhất trong trường hợp này là *“DT\_model\_2.joblib”* (Hình 3.25).



Hình 3.24 Biểu đồ so sánh độ chính xác của các mô hình sau khi kiểm thử trên tập dữ liệu tự thu



Hình 3.25 Biểu đồ ma trận nhầm lẫn mô hình DT\_model\_2.joblib trên tập dữ liệu tự thu