DOI: https://doi.org/10.62985/j.huit_ojs.vol26.no1E.373

HIỆU NĂNG HỌC MÁY CHO PHÁT HIỆN TẤN CÔNG WEB

Lê Anh Tuấn*

Trường Đại học Công Thương Thành phố Hồ Chí Minh

*Email: *tuanla@huit.edu.vn*

Ngày nhận bài: 15/4/2026; Ngày nhận bài sửa: 12/5/2026; Ngày chấp nhận đăng: 22/5/2026

TÓM TẮT

Sự gia tăng nhanh chóng của các hệ thống web đã kéo theo nguy cơ gia tăng của nhiều dạng tấn công như SQL injection, XSS và CSRF với mức độ ngày càng tinh vi. Các cơ chế phòng thủ truyền thống bộc lộ hạn chế trong việc nhận diện các biến thể tấn công mới, do thiếu khả năng thích ứng linh hoạt. Bài báo này tập trung xây dựng một phương pháp phát hiện tấn công web dựa trên học máy, sử dụng bộ dữ liệu HTTP CSIC 2010 làm nền tảng thực nghiệm. Dữ liệu được xử lý và trích xuất đặc trưng từ các yêu cầu HTTP nhằm phục vụ huấn luyện các mô hình phân loại. Nhiều thuật toán học máy, bao gồm RF (Random Forest) và SVM (Support Vector Machine), được triển khai để nhận diện hành vi bất thường. Kết quả đánh giá cho thấy RF đạt hiệu năng cao nhất với chỉ số đánh giá: độ chính xác 96,03%, F1-score 96,00% và ROC-AUC 0,995. Các kết quả này cho thấy cách tiếp cận dựa trên học máy có khả năng nâng cao hiệu quả phát hiện các yêu cầu HTTP độc hại trong môi trường web hiện đại.

Từ khóa: Học máy, hệ thống phát hiện, tấn công web.

1. GIỚI THIỆU

Sự phát triển nhanh chóng của các ứng dụng web và dịch vụ trực tuyến đã làm gia tăng đáng kể nguy cơ xuất hiện các hình thức tấn công như SQL injection (SQLi), Cross-site scripting (XSS) và Cross-site request forgery (CSRF) và thao túng tham số. Các mối đe dọa này ngày càng trở nên tinh vi và khó phát hiện, gây ra nhiều rủi ro đối với an toàn thông tin của các hệ thống [1] [2]. Các giải pháp bảo mật truyền thống, bao gồm tường lửa ứng dụng web dựa trên luật và hệ thống phát hiện xâm nhập dựa trên chữ ký, thường gặp hạn chế trong việc thích nghi với các biến thể tấn công mới, đồng thời có thể dẫn đến tỷ lệ cảnh báo sai cao [2].

Trong bối cảnh đó, các phương pháp dựa trên học máy được xem là một hướng tiếp cận hiệu quả nhằm nâng cao khả năng phát hiện các hành vi tấn công trên nền tảng web [3], [4]. Nghiên cứu này đề xuất xây dựng một hệ thống phát hiện tấn công web sử dụng học máy, dựa trên bộ dữ liệu HTTP CSIC 2010 – một tập dữ liệu chuẩn được sử dụng rộng rãi trong lĩnh vực này [5], [6]. Dữ liệu bao gồm các yêu cầu HTTP hợp lệ và bất thường, cho phép tiến hành trích xuất đặc trưng từ URL, nội dung truy vấn và các trường tiêu đề nhằm phục vụ quá trình huấn luyện mô hình phân loại.

Khác với các nghiên cứu chỉ tập trung vào một mô hình phân loại riêng lẻ, nghiên cứu này đề xuất một pipeline phát hiện tấn công web hoàn chỉnh bao gồm tiền xử lý HTTP request, kết hợp đặc trưng thủ công và TF-IDF, đánh giá đa mô hình học máy và triển khai mô phỏng hệ thống phát hiện gần thời gian thực. Trọng tâm của nghiên cứu không nằm ở việc đề xuất thuật toán mới, mà ở việc đánh giá khả năng ứng dụng thực tế của các mô hình học máy trong phát hiện tấn công web.

Các đóng góp chính của nghiên cứu bao gồm:

(i) đề xuất một quy trình phát hiện tấn công web tích hợp giữa đặc trưng thủ công của HTTP request (độ dài URL, ký tự đặc biệt, từ khóa nghi vấn, số lượng tham số) với đặc trưng ngữ nghĩa được biểu diễn bằng TF-IDF, nhằm nâng cao khả năng biểu diễn dữ liệu cho bài toán phát hiện tấn công web;

(ii) thực hiện đánh giá và phân tích thực nghiệm toàn diện nhiều mô hình học máy (RF, RFO, Naive Bayes, GaussianNB, Decision Tree, KNN, LinearSVC và XGBoost) trên cùng bộ dữ liệu CSIC 2010 theo nhiều tiêu chí gồm Accuracy, Precision, Recall, F1-score, ROC-AUC, Log Loss, thời gian huấn luyện và thời gian suy luận, từ đó làm rõ trade-off giữa hiệu năng phát hiện và chi phí tính toán;

(iii) xây dựng môi trường mô phỏng phát hiện tấn công web gần thời gian thực với kiến trúc ba mô-đun gồm ứng dụng web giả lập, mô-đun sinh tấn công và mô-đun phát hiện, qua đó kiểm chứng khả năng triển khai thực tiễn của mô hình học máy trong phát hiện các truy vấn HTTP độc hại.

2. CÁC NGHIÊN CỨU LIÊN QUAN

Trong những năm gần đây, sự mở rộng nhanh chóng của các hệ thống web và dịch vụ trực tuyến không chỉ mang lại nhiều lợi ích mà còn làm gia tăng đáng kể các rủi ro liên quan đến an ninh mạng. Các hình thức tấn công như SQLi, XSS, CSRF và thay đổi tham số đã trở thành những mối đe dọa nghiêm trọng đối với tổ chức, doanh nghiệp cũng như người sử dụng cuối.

Các cơ chế bảo mật truyền thống, bao gồm tường lửa ứng dụng web dựa trên luật và các hệ thống phát hiện xâm nhập dựa trên chữ ký, đang dần bộc lộ những hạn chế nhất định. Những phương pháp này thiếu khả năng thích ứng trước các kỹ thuật tấn công mới, đặc biệt là các biến thể tinh vi của các lỗ hổng đã được nhận diện trước đó.

Mặc dù nhiều nghiên cứu trước đây đã áp dụng học máy và học sâu cho bài toán phát hiện tấn công web, phần lớn các công trình tập trung vào việc tối ưu độ chính xác của từng mô hình riêng lẻ hoặc khai thác các kiến trúc học sâu phức tạp [7-10]. Tuy nhiên, vẫn còn thiếu các nghiên cứu thực hiện đánh giá thực nghiệm toàn diện nhiều mô hình học máy truyền thống trên cùng một quy trình xử lý dữ liệu HTTP, đồng thời phân tích trade-off giữa hiệu năng phát hiện, khả năng suy luận thời gian thực và chi phí tính toán trong môi trường mô phỏng gần thực tế.

Bên cạnh đó, nhiều nghiên cứu chủ yếu tập trung vào kết quả phân loại mà chưa đánh giá đầy đủ khả năng triển khai thực tiễn của mô hình trong hệ thống phát hiện tấn công web hoàn chỉnh. Do đó, nghiên cứu này tập trung xây dựng quy trình phát hiện tấn công web dựa trên học máy, kết hợp giữa trích xuất đặc trưng HTTP, đánh giá đa mô hình và triển khai mô phỏng hệ thống nhằm phân tích khả năng ứng dụng thực tế của các mô hình trong phát hiện yêu cầu HTTP độc hại.

Tại Việt Nam, việc triển khai các giải pháp phát hiện tấn công web dựa trên học máy vẫn còn đối mặt với nhiều thách thức liên quan đến dữ liệu và nguồn nhân lực. Một số nghiên cứu gần đây cho thấy số lượng bộ dữ liệu an ninh mạng công khai phục vụ huấn luyện mô hình học máy còn hạn chế, trong khi việc xây dựng hệ thống phát hiện tấn công dựa trên AI đòi hỏi chuyên môn liên ngành giữa an toàn thông tin và trí tuệ nhân tạo [2, 11]. Ngoài ra, nhiều nghiên cứu trong nước hiện vẫn tập trung chủ yếu vào đánh giá mô hình trong môi trường thực nghiệm hoặc ngoại tuyến, trong khi các đánh giá liên quan đến khả năng triển khai gần thời gian thực và thích nghi với các hình thức tấn công mới vẫn còn hạn chế [12].

3. PHƯƠNG PHÁP ĐỀ XUẤT

3.1. Dữ liệu

Bộ dữ liệu CSIC 2010 [5], được xây dựng bởi Viện Khoa học Máy tính Tây Ban Nha (Spanish Research National Council), được sử dụng làm nền tảng cho nghiên cứu này nhằm phục vụ bài toán phát hiện tấn công web. Dữ liệu được thu thập từ một ứng dụng thương mại điện tử mô phỏng và đã được gán nhãn rõ ràng thành hai lớp: hợp lệ (normal) và bất thường (anomalous). Tổng số mẫu trong tập dữ liệu là 61,065, bao gồm 36,000 yêu cầu bình thường và 25,065 yêu cầu bất thường, với 17 thuộc tính đặc trưng.

Trong giai đoạn tiền xử lý, các yêu cầu HTTP được phân tích và chuẩn hóa thông qua việc tách các thành phần chính như URL, tham số truy vấn, trường tiêu đề (headers) và nội dung (body). Đồng thời, dữ liệu được đưa về định dạng thống nhất bằng cách chuyển toàn bộ ký tự về chữ thường, chuẩn hóa mã hóa và xử lý các giá trị thiếu hoặc không hợp lệ.

Sau đó, các đặc trưng dạng văn bản được chuyển đổi sang dạng số thông qua các kỹ thuật mã hóa như mã hóa One-Hot (One-Hot Encoding) hoặc TF-IDF (Term Frequency – Inverse Document Frequency), nhằm tạo đầu vào phù hợp cho các mô hình học máy. Để giảm thiểu ảnh hưởng của sự mất cân bằng giữa các lớp, phương pháp kỹ thuật cân bằng dữ liệu SMOTE (Synthetic Minority Over-sampling Technique) được áp dụng nhằm tăng cường số lượng mẫu thuộc lớp thiểu số, qua đó hạn chế hiện tượng thiên lệch trong quá trình huấn luyện. Cuối cùng, tập dữ liệu được phân chia thành ba phần:

70% cho huấn luyện, 20% cho kiểm tra và 10% cho xác thực, nhằm đảm bảo việc đánh giá mô hình được thực hiện một cách khách quan và ổn định.

3.2. Xây dựng mô hình học máy

3.2.1. Lựa chọn mô hình

Trong nghiên cứu này, nhiều thuật toán học máy được lựa chọn và triển khai nhằm đánh giá khả năng nhận diện tấn công web thông qua bài toán phân loại các yêu cầu HTTP. Trước tiên, mô hình RF được xem là phương pháp cốt lõi nhờ khả năng xử lý hiệu quả dữ liệu đa đặc trưng, tính ổn định cao trong bối cảnh dữ liệu mất cân bằng, đồng thời hạn chế hiện tượng quá khớp thông qua cơ chế tổ hợp nhiều cây quyết định [13, 14]. Bên cạnh đó, một phiên bản RF được tối ưu hóa tham số (RF Optimized) cũng được xây dựng nhằm cải thiện hiệu suất phân loại và nâng cao độ chính xác trên tập dữ liệu kiểm tra.

Ngoài ra, các mô hình xác suất như Naive Bayes và Gaussian Naive Bayes được sử dụng như các phương pháp cơ sở (baseline), nhờ ưu điểm về tốc độ huấn luyện nhanh và khả năng xử lý hiệu quả dữ liệu văn bản sau khi mã hóa đặc trưng [15]. Thuật toán cây quyết định (Decision Tree) cũng được triển khai để đánh giá khả năng phân loại dựa trên các quy tắc phân tách rõ ràng và dễ diễn giải [14]. Đồng thời, phương pháp KNN (K-Nearest Neighbors) được áp dụng nhằm khảo sát hiệu quả của cách tiếp cận dựa trên khoảng cách trong không gian đặc trưng [16].

Đối với dữ liệu có số chiều lớn, đặc biệt sau khi áp dụng các kỹ thuật như TF-IDF, mô hình LinearSVC (Linear Support Vector Classification) được lựa chọn do khả năng phân tách tuyến tính hiệu quả trong không gian đặc trưng cao chiều [17]. Cuối cùng, XGBoost – một thuật toán boosting tiên tiến – được đưa vào thử nghiệm nhờ khả năng tối ưu hóa mạnh mẽ và thường đạt hiệu năng cao trong các bài toán phân loại, bao gồm cả lĩnh vực an ninh mạng [18].

Việc kết hợp và so sánh nhiều mô hình khác nhau cho phép nghiên cứu thực hiện đánh giá toàn diện, từ đó xác định phương pháp tối ưu nhất cho hệ thống phát hiện tấn công web.

Mặc dù được xây dựng từ năm 2010, bộ dữ liệu HTTP CSIC 2010 vẫn được sử dụng rộng rãi trong nhiều nghiên cứu về phát hiện tấn công web nhờ đặc điểm được gán nhãn đầy đủ, chứa đa dạng các mẫu HTTP bất thường và hỗ trợ khả năng đối chiếu thực nghiệm giữa các nghiên cứu. Ngoài ra, bộ dữ liệu này phù hợp cho việc đánh giá các mô hình học máy truyền thống trong môi trường kiểm soát. Tuy nhiên, nghiên cứu cũng thừa nhận rằng CSIC 2010 chưa phản ánh đầy đủ các hình thức tấn công web hiện đại, đây cũng là một hạn chế của nghiên cứu.

3.2.2. Trích xuất đặc trưng

Trong nghiên cứu này, quá trình trích xuất đặc trưng từ các yêu cầu HTTP được thực hiện nhằm chuyển đổi dữ liệu ban đầu thành tập thuộc tính có ý nghĩa phục vụ cho bài toán phân loại. Các đặc trưng được xây dựng dựa trên hai nguồn thông tin chính gồm URL và nội dung yêu cầu (content). Đối với URL, các thuộc tính được khai thác bao gồm độ dài chuỗi, số lượng ký tự đặc biệt, tần suất xuất hiện của các từ khóa liên quan đến tấn công và số lượng tham số. Tương tự, từ phần nội dung yêu cầu, các đặc trưng như độ dài dữ liệu, mật độ ký tự đặc biệt và tần suất các từ khóa đáng ngờ cũng được trích xuất.

Những đặc trưng này đóng vai trò quan trọng trong việc phát hiện các dấu hiệu bất thường thường gặp trong các payload tấn công, điển hình như SQLi và XSS. Bên cạnh đó, để hạn chế hiện tượng quá khớp (overfitting) và nâng cao khả năng tổng quát hóa của mô hình, nghiên cứu tiến hành loại bỏ các đặc trưng dư thừa, đồng thời áp dụng các kỹ thuật giảm chiều dữ liệu, chẳng hạn như PCA (Principal Component Analysis). Cách tiếp cận này góp phần tối ưu hóa hiệu năng mô hình trong quá trình huấn luyện và đánh giá.

3.2.3. Huấn luyện và tối ưu mô hình

Trong giai đoạn huấn luyện, các thuật toán học máy được xây dựng trên tập dữ liệu huấn luyện chiếm 70% tổng số mẫu. Để đảm bảo tính ổn định và giảm sự phụ thuộc vào một cách chia dữ liệu cố định, phương pháp k-fold cross-validation [19] với k = 5 được áp dụng trong quá trình đánh giá nội bộ. Đối với các mô hình trọng tâm như RF và XGBoost, việc tối ưu siêu tham số được thực hiện thông qua kỹ thuật Grid Search [20] nhằm xác định cấu hình phù hợp nhất cho từng mô hình.

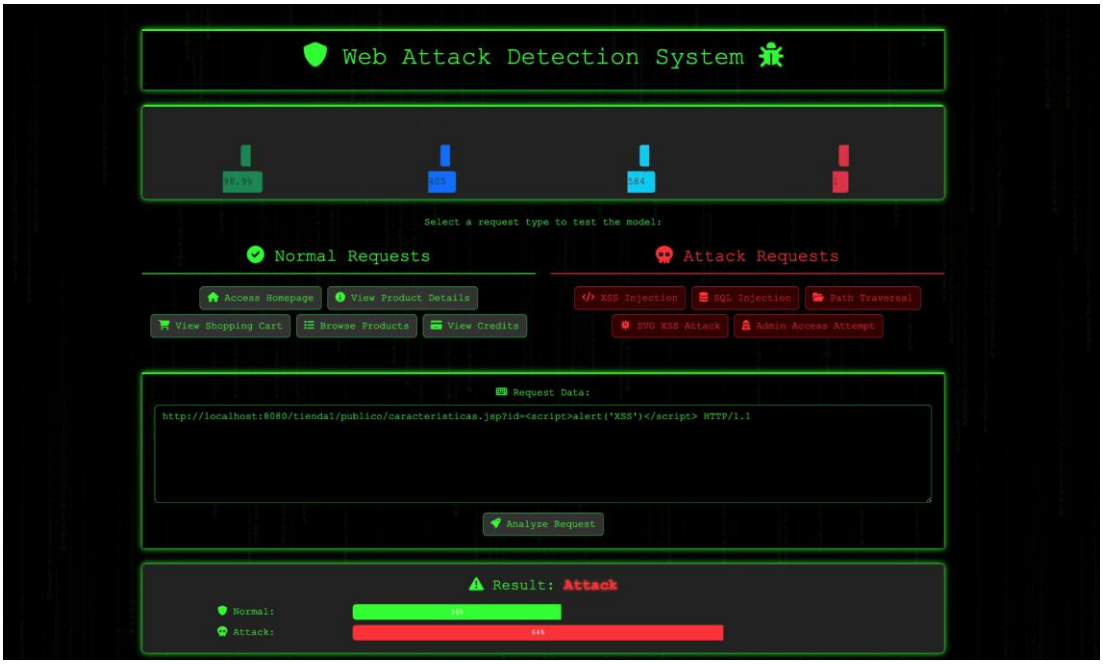
Bên cạnh đó, các biện pháp hạn chế hiện tượng quá khớp được triển khai, bao gồm sử dụng các kỹ thuật điều chuẩn (regularization) như L1 và L2, cũng như dropout trong trường hợp áp dụng các mô

hình học sâu. Ngoài ra, các tham số huấn luyện như tốc độ học (learning rate) và số vòng lặp (epoch) cũng được điều chỉnh nhằm nâng cao hiệu năng của mô hình trên tập dữ liệu kiểm tra.

Cuối cùng, hiệu quả của hệ thống được đánh giá thông qua các chỉ số phổ biến [14] như Precision, Recall, F1-score và ROC-AUC. Trong đó, Recall được đặc biệt chú trọng nhằm giảm thiểu khả năng bỏ sót các yêu cầu HTTP mang tính chất tấn công.

3.2.4. Kiến trúc hệ thống mô phỏng

Mục đích của hệ thống mô phỏng là xây dựng một môi trường web giả định có khả năng phát sinh các hành vi tấn công, qua đó đánh giá năng lực của mô hình trong việc phát hiện các dạng khai thác phổ biến như SQL Injection, XSS và Path Traversal. Ứng dụng được thiết kế với giao diện thân thiện, cho phép hiển thị kết quả suy luận theo thời gian gần thực dưới dạng phân loại "Hợp lệ" hoặc "Bất thường". Đồng thời, hệ thống cung cấp thêm thước đo xác suất nhằm phản ánh mức độ tin cậy của dự đoán, hỗ trợ người dùng trong quá trình phân tích và ra quyết định.



Hình 1. Giao diện hệ thống

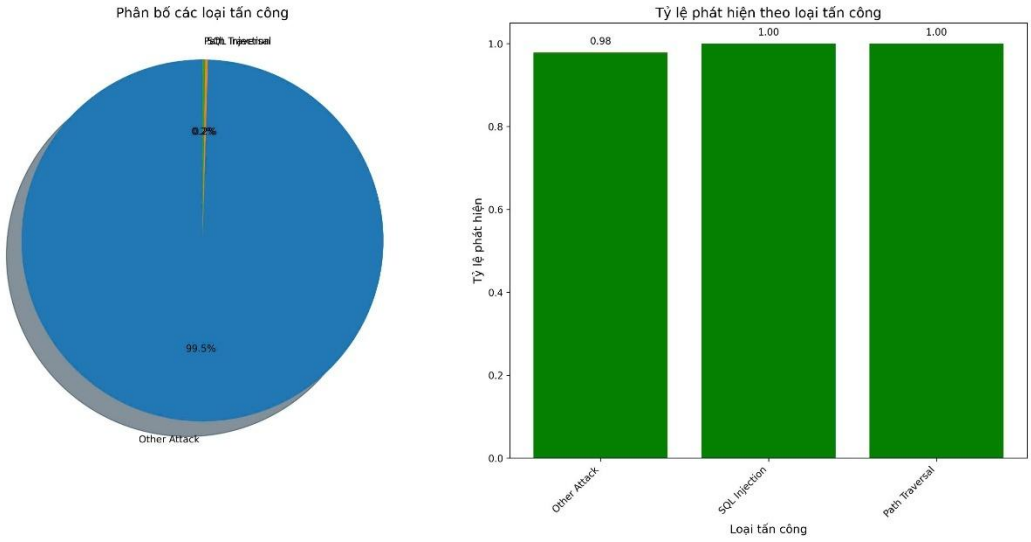
Hệ thống mô phỏng được thiết kế theo kiến trúc gồm ba mô-đun chính có sự tương tác chặt chẽ. Thứ nhất, mô-đun ứng dụng web giả lập đóng vai trò đối tượng kiểm thử, cho phép người dùng gửi các yêu cầu HTTP (GET/POST) tới máy chủ phía sau. Ứng dụng này được phát triển bằng các công nghệ nền tảng như HTML, CSS, JavaScript kết hợp với Flask, nhằm tái hiện các chức năng điển hình như xác thực người dùng hoặc truy vấn dữ liệu.

Thứ hai, mô-đun sinh tấn công đảm nhiệm việc tạo ra các yêu cầu HTTP mang tính chất độc hại để đánh giá hệ thống. Các mẫu payload được xây dựng dựa trên bộ dữ liệu CSIC 2010, bao gồm các chuỗi khai thác tiêu biểu như SQL Injection (ví dụ: ' OR 1=1 --) và XSS (ví dụ: <script>alert('xss')</script>). Quá trình này có thể được thực hiện thông qua các công cụ kiểm thử bảo mật chuyên dụng hoặc các kịch bản tự động hóa.

Thứ ba, mô-đun phát hiện và phòng vệ đóng vai trò trung tâm, sử dụng mô hình học máy XGBoost đã được huấn luyện để phân loại lưu lượng HTTP thành hai nhóm: hợp lệ và nguy hiểm. Khi nhận diện truy vấn có dấu hiệu tấn công, hệ thống sẽ thực hiện cơ chế ngăn chặn đồng thời phát cảnh báo thông qua giao diện người dùng hoặc hệ thống ghi log.

Kết quả thực nghiệm trong môi trường mô phỏng cho thấy giải pháp đề xuất đạt hiệu quả cao trong việc nhận diện các hình thức tấn công phổ biến. Cụ thể, tỷ lệ phát hiện đối với SQL Injection và

Path Traversal đạt mức tối đa, trong khi nhóm tấn công khác cũng đạt độ chính xác xấp xỉ 98%. Mặc dù tồn tại sự mất cân đối dữ liệu với tỷ lệ lớn thuộc về nhóm "Other Attack", mô hình vẫn duy trì được tính ổn định và khả năng phân loại đáng tin cậy.



Hình 2. Kết quả phân bổ từ hệ thống mô phỏng

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ

4.1 Kết quả mô phỏng

Dữ liệu được chia theo tỉ lệ sau: tập train (70%), tập validation (10%) và tập test (20%), kích thước cụ thể:

- Tập train: 50400 mẫu
- Tập validation: 7128 mẫu
- Tập test: 14472 mẫu

RF (Random Forest): Kết quả cho thấy RF có thời gian khoảng 15,15 s, trong khi thời gian dự đoán trên tập dữ liệu kiểm tra chỉ khoảng 0,74 s.

Bảng 1. Đánh giá kết quả Random Forest

Chỉ số	Tập Validation	Tập Test
Accuracy	0.9653	0.9603
Test Error (1 − Accuracy)	0.0347	0.0397
Precision	0.9699	0.9617
Recall	0.9610	0.9584
F1-Score	0.9654	0.9600
ROC-AUC	0.9959	0.9952
Log Loss	0.0921	0.0927

RFO (Random Forest Optimized-Grid Search): Trong nghiên cứu này, mô hình RF được cải tiến thông qua quá trình tối ưu siêu tham số sử dụng phương pháp Grid Search nhằm xác định cấu hình phù hợp nhất. Kết quả tối ưu cho thấy các tham số hiệu quả bao gồm độ sâu tối đa của cây (max_depth) bằng 20, số lượng mẫu tối thiểu để phân tách nút (min_samples_split) là 5 và số lượng cây trong rừng

(n_estimators) đạt 200. Hiệu suất của mô hình được đánh giá bằng phương pháp cross-validation, trong đó chỉ số Recall đạt giá trị cao nhất là 0,9679.

Về mặt tính toán, thời gian cần thiết để huấn luyện mô hình RF sau khi tối ưu vào khoảng 1095,50 s, trong khi thời gian thực hiện dự đoán trên tập dữ liệu kiểm tra tương đối thấp, chỉ khoảng 0,81 s. Kết quả này cho thấy mô hình không chỉ đạt hiệu quả cao trong nhận diện mà còn đảm bảo khả năng suy luận nhanh trong giai đoạn triển khai.

Bảng 2. Đánh giá kết quả RF optimized (grid search)

Chỉ số	Tập Validation	Tập Test
Accuracy	0.9554	0.9524
Test Error (1 − Accuracy)	0.0446	0.0476
Precision	0.9497	0.9416
Recall	0.9624	0.9640
F1-Score	0.9560	0.9527
ROC-AUC	0.9919	0.9909
Log Loss	0.1867	0.1901

Naive Bayes: Thời gian huấn luyện mô hình Naive Bayes là khoảng 0,25 s, trong khi thời gian dự đoán trên tập kiểm tra khoảng 0,12 s.

Bảng 3. Đánh giá kết quả Naive Bayes

Chỉ số	Tập Validation	Tập Test
Accuracy	0.8796	0.8789
Test Error (1 − Accuracy)	0.1204	0.1211
Precision	0.8385	0.8298
Recall	0.9426	0.9515
F1-Score	0.8875	0.8865
ROC-AUC	0.9581	0.9597
Log Loss	0.3056	0.3010

Naive Bayes Optimized by GaussianNB: Thời gian huấn luyện mô hình là khoảng 1,31 s, trong khi thời gian dự đoán trên tập kiểm tra khoảng 1,12 s.

Bảng 4. Đánh giá kết quả Naive Bayes Optimized (GaussianNB)

Chỉ số	Tập Validation	Tập Test
Accuracy	0.9088	0.9089
Test Error (1 − Accuracy)	0.0912	0.0911
Precision	0.9754	0.9705
Recall	0.8402	0.8424
F1-Score	0.9028	0.9019
ROC-AUC	0.9101	0.9102
Log Loss	3.2868	3.2810

Decision Tree: Mô hình dựa trên cơ chế phân tách dữ liệu theo các điều kiện đặc trưng và có ưu điểm dễ triển khai. Thời gian huấn luyện mô hình Decision Tree là khoảng 7,01 s, trong khi thời gian dự đoán trên tập kiểm tra khoảng 0,17 s.

Bảng 5. Đánh giá kết quả Decision Tree

Chỉ số	Tập Validation	Tập Test
Accuracy	0.9592	0.9552
Test Error (1 − Accuracy)	0.0408	0.0448
Precision	0.9630	0.9584
Recall	0.9557	0.9511
F1-Score	0.9593	0.9547
ROC-AUC	0.9612	0.9572
Log Loss	1.3939	1.5384

K-Nearest Neighbor: Mô hình thực hiện dự đoán bằng cách so sánh yêu cầu mới với các mẫu lân cận gần nhất trong tập huấn luyện. Thời gian huấn luyện mô hình KNN là khoảng 0,08 s.

Bảng 6. Đánh giá kết quả K-nearest Neighbor

Chỉ số	Tập Validation	Tập Test
Accuracy	0.9258	0.9243
Test Error (1 − Accuracy)	0.0742	0.0757
Precision	0.9040	0.8973
Recall	0.9541	0.9572
F1-Score	0.9283	0.9263
ROC-AUC	0.9748	0.9743
Log Loss	0.5894	0.6024

LinearSVC: Đây là mô hình dựa trên Support Vector Machine với hàm phân tách tuyến tính, phù hợp với dữ liệu có số chiều lớn sau khi mã hóa đặc trưng. Thời gian huấn luyện mô hình LinearSVC là khoảng 176,30 s, trong khi thời gian dự đoán trên tập kiểm tra khoảng 0,50 s.

Bảng 7. Đánh giá kết quả LinearSVC

Chỉ số	Tập Validation	Tập Test
Accuracy	0.9578	0.9563
Test Error (1 − Accuracy)	0.0422	0.0437
Precision	0.9724	0.9701
Recall	0.9429	0.9412
F1-Score	0.9574	0.9554
ROC-AUC	0.9839	0.9855
Log Loss	0.1433	0.1396

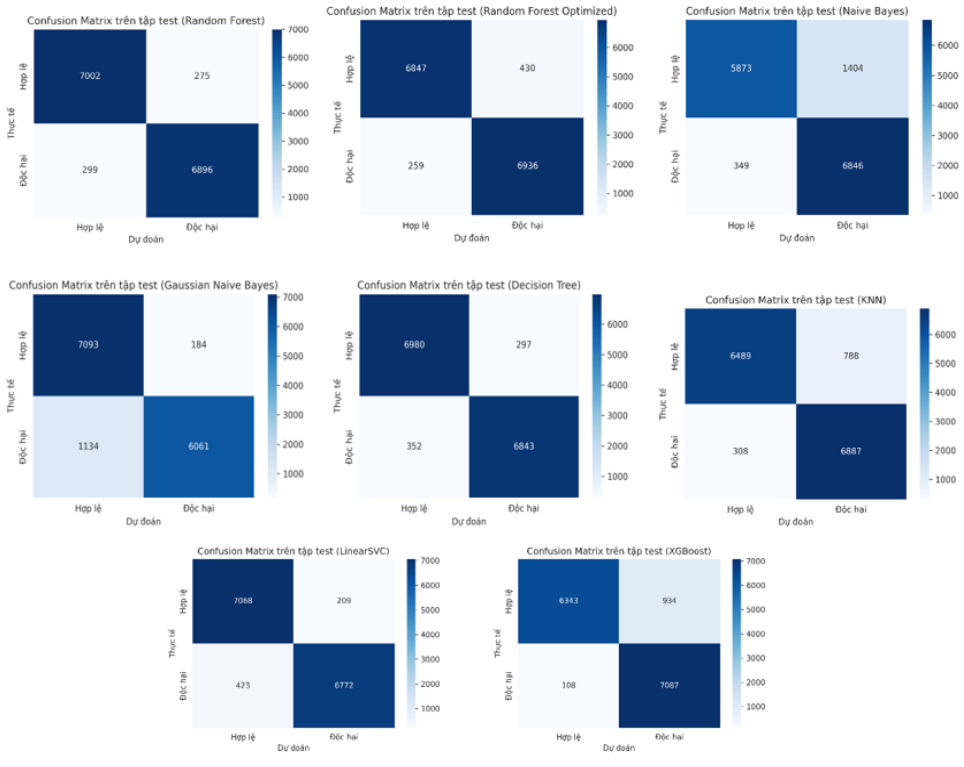
XGBoost: Thuật toán này kết hợp nhiều cây quyết định theo cơ chế tăng cường để cải thiện hiệu suất phân loại. Thời gian huấn luyện mô hình XGBoost là khoảng 13,93 s, trong khi thời gian dự đoán trên tập kiểm tra khoảng 0,20 s.

Bảng 8. Đánh giá kết quả XGboost

Chỉ số	Tập Validation	Tập Test
Accuracy	0.9334	0.9280
Test Error (1 − Accuracy)	0.0666	0.0720
Precision	0.8936	0.8836
Recall	0.9850	0.9850
F1-Score	0.9371	0.9315
ROC-AUC	0.9865	0.9857
Log Loss	0.1956	0.2015

Bảng 9. Bảng so sánh tổng hợp giữa các mô hình

Model	Accuracy	Test Error	Precision	Recall	F1-Score	ROC-AUC	Log Loss	Training Time	Prediction Time
Random Forest	0.960337	0.039663	0.961651	0.958443	0.960045	0.995163	0.092692	15.146718	0.739341
Random Forest Optimized	0.952391	0.047609	0.941624	0.964003	0.952682	0.990859	0.190068	1095.496533	0.807249
Naive Bayes	0.878870	0.121130	0.829818	0.951494	0.886500	0.959658	0.301016	0.245179	0.116610
Gaussian Naive Bayes	0.908928	0.091072	0.970536	0.842391	0.901935	0.910223	3.280981	1.307435	1.116771
Decision Tree	0.955155	0.044845	0.958403	0.951077	0.954726	0.957162	1.538392	7.013248	0.172929
KNN	0.924268	0.075732	0.897329	0.957192	0.926295	0.974287	0.602371	0.078054	95.500406
LinearSVC	0.956329	0.043671	0.970062	0.941209	0.955418	0.985479	0.139628	176.304708	0.504551
XGBoost	0.927999	0.072001	0.883556	0.984990	0.931519	0.985748	0.201526	13.928335	0.200170



Hình 3. Ma trận nhầm lẫn trên tập kiểm tra trên các mô hình học máy

Hình 3 trình bày ma trận nhầm lẫn trên tập kiểm tra đối với các mô hình RF, Optimized Naive Bayes, Gaussian Naive Bayes, Decision Tree, KNN, LinearSVC và XGBoost.

4.2. Thảo luận

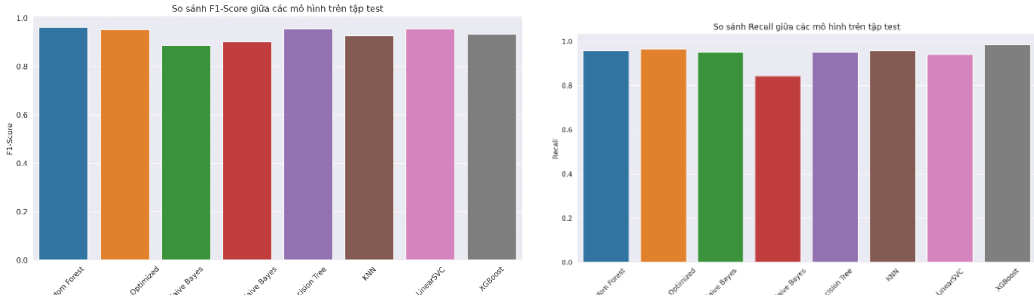
Kết quả thực nghiệm trong Bảng 9 cho thấy các thuật toán học máy được áp dụng đều có khả năng phân loại yêu cầu HTTP thành hai nhóm hợp lệ và bất thường, tuy nhiên hiệu năng giữa các mô hình có sự khác biệt đáng kể. Các chỉ số đánh giá bao gồm Accuracy, Precision, Recall, F1-score, ROC-AUC, Log Loss cùng với thời gian huấn luyện và suy luận đã cung cấp đánh giá toàn diện về ưu điểm và hạn chế của từng phương pháp.

Trong số các mô hình được khảo sát, RF thể hiện hiệu quả tổng thể vượt trội do có sự kết hợp của nhiều cây quyết định độc lập, cơ chế này giúp giảm hiện tượng overfitting và xử lý tốt dữ liệu đa đặc trưng. Giá trị Accuracy đạt 96,03% cho thấy khả năng phân biệt hai lớp rất tốt. Sau khi thực hiện tối ưu siêu tham số bằng Grid Search, phiên bản RFO ghi nhận sự cải thiện về Recall, đạt mức 96,40%, tuy nhiên chi phí huấn luyện tăng lên đáng kể so với mô hình ban đầu.

Các mô hình đơn giản hơn như Decision Tree và LinearSVC cũng đạt kết quả khả quan, với Accuracy lần lượt xấp xỉ 95,52% và 95,63%, đồng thời F1-score đều vượt ngưỡng 95%. Tuy vậy, LinearSVC đòi hỏi thời gian huấn luyện lớn hơn, trong khi Decision Tree có ưu điểm về tốc độ nhưng lại kém hơn về chỉ số ROC-AUC so với RF.

Đối với nhóm mô hình xác suất, Naive Bayes cho kết quả thấp nhất với Accuracy đạt 87,89%, do giả định các đặc trưng là độc lập với nhau việc này khiến cho mô hình không nhận diện được các mẫu tấn công phức tạp, mặc dù Recall ở mức tương đối cao. Điều này cho thấy mô hình có xu hướng phát hiện được nhiều mẫu tấn công nhưng đồng thời làm gia tăng tỷ lệ cảnh báo sai. Phiên bản Gaussian Naive Bayes cải thiện Accuracy lên khoảng 90,89%, tuy nhiên giá trị Log Loss lớn cho thấy độ tin cậy trong dự đoán chưa cao.

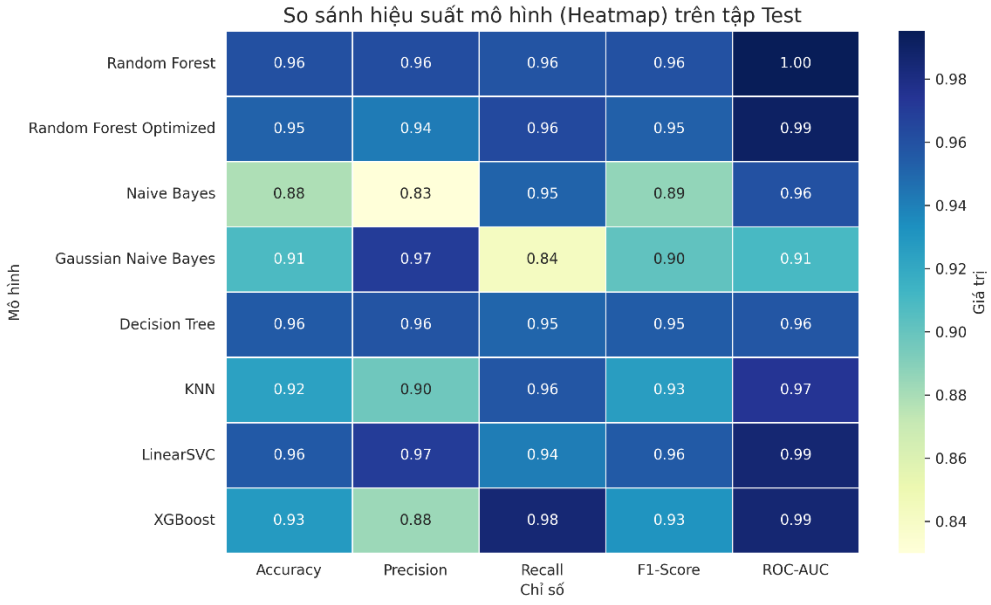
Mô hình K-Nearest Neighbors đạt Accuracy khoảng 92,43%, nhưng thời gian dự đoán rất lớn (khoảng 95,5 s) do nó phải so sánh mẫu mới với toàn bộ tập dữ liệu huấn luyện, điều này gây hạn chế đáng kể khi triển khai trong các hệ thống yêu cầu phản hồi theo thời gian thực. Ngược lại, XGBoost đạt Recall cao nhất, lên đến 98,50%, cùng với thời gian suy luận nhanh (khoảng 0,20 s). Tuy nhiên, Accuracy của mô hình này chỉ đạt khoảng 92,80%, thấp hơn so với các mô hình có hiệu suất tốt nhất.



Hình 4. So sánh f1-score và recall của các mô hình

Dựa trên phân tích từ biểu đồ F1-score, có thể thấy mô hình **RF** đạt giá trị cao nhất, phản ánh khả năng cân bằng tốt giữa Precision và Recall, đồng thời duy trì hiệu năng ổn định trong việc phân biệt giữa các yêu cầu truy cập hợp lệ và độc hại. Bên cạnh đó, phương pháp **LinearSVC** cũng được ghi nhận hoạt động tốt trên dữ liệu có số chiều lớn nhờ khả năng phân tách tuyến tính hiệu quả với F1-score vượt ngưỡng 0,95, cho thấy mức độ hiệu quả đáng kể trong bài toán phân loại này.

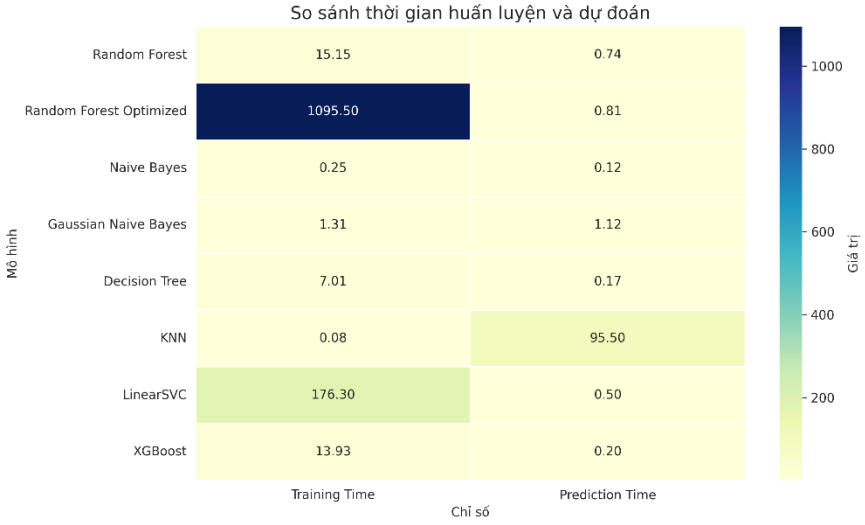
Ngược lại, nhóm mô hình dựa trên giả định xác suất như **Naive Bayes** thể hiện hiệu năng thấp hơn tương đối. Đặc biệt, phiên bản Naive Bayes cơ bản chỉ đạt F1-score xấp xỉ 0,88, điều này cho thấy những hạn chế nhất định khi xử lý các mẫu tấn công có đặc trưng phức tạp và phân bố dữ liệu không tuân theo giả định độc lập giữa các thuộc tính.



Hình 5. Biểu đồ heatmap so sánh hiệu suất các mô hình

Xét theo chỉ số Recall, kết quả trực quan từ biểu đồ cho thấy **XGBoost** đạt giá trị cao nhất (98,50%) do sử dụng cơ chế tăng cường (boosting), cơ chế này tập trung vào việc sửa lỗi cho các cây quyết định trước đó, phản ánh năng lực phát hiện các mẫu tấn công vượt trội, đồng thời làm giảm đáng kể tỷ lệ bỏ sót các yêu cầu độc hại. Các mô hình như: **KNN** và **Naive Bayes** cũng ghi nhận Recall ở mức cao; tuy nhiên, hiệu năng tổng thể của chúng, thể hiện qua Accuracy và F1-score, vẫn thấp hơn so với RF. Trái lại, **Gaussian Naive Bayes** có giá trị Recall suy giảm rõ rệt, cho thấy khả năng nhận diện các mẫu tấn công còn hạn chế và dễ dẫn đến hiện tượng bỏ sót.

Hình 5 minh họa ma trận nhiệt (heatmap) dùng để so sánh hiệu suất của các mô hình học máy trên tập kiểm tra, qua đó cung cấp cái nhìn trực quan về sự khác biệt giữa các phương pháp trong bài toán phát hiện tấn công web. Trong đó, **XGBoost** nổi bật với Recall xấp xỉ 0,98, khẳng định khả năng phát hiện tấn công hiệu quả. Tuy nhiên, do Precision không tương ứng cao, F1-score của mô hình chỉ đạt mức trung bình (khoảng 0,93). Điều này cho thấy xu hướng của mô hình là tăng cường cảnh báo, đồng thời kéo theo số lượng dự đoán dương tính giả cao hơn.



Hình 6. So sánh thời gian huấn luyện và dự đoán của các mô hình học máy

Hình 6 thể hiện bản đồ nhiệt (heatmap) dùng để đối chiếu thời gian huấn luyện và thời gian suy luận của các mô hình học máy trên tập kiểm tra. Hai đại lượng này đóng vai trò then chốt khi đánh giá tính khả thi của việc triển khai trong các hệ thống phát hiện tấn công web, đặc biệt ở những môi trường đòi hỏi độ trễ thấp và phản hồi gần thời gian thực. Kết quả thực nghiệm cho thấy **XGBoost** đạt được sự cân bằng hợp lý giữa chi phí huấn luyện và tốc độ dự đoán, với thời gian đào tạo xấp xỉ 13,93 s và thời gian suy luận rất ngắn, khoảng 0,20 s, qua đó thể hiện ưu thế rõ rệt về hiệu năng vận hành.

Kết quả cho thấy không có mô hình nào tối ưu về mọi mặt, việc lựa chọn mô hình sẽ phụ thuộc vào ngữ cảnh và mục tiêu sử dụng: nếu ưu tiên về độ chính xác tổng thể (Accuracy) và sự cân bằng (F1-score) thì RF là lựa chọn tốt nhất; còn nếu ưu tiên không bỏ sót tấn công (Recall) thì XGBoost lại vượt trội hơn dù có thể gây ra nhiều cảnh báo sai.

5. KẾT LUẬN

Nghiên cứu đã phát triển một hệ thống phát hiện tấn công web dựa trên kỹ thuật học máy, với mục tiêu phân loại các yêu cầu HTTP thành hai nhóm: hợp lệ và độc hại. Quy trình xử lý dữ liệu được thiết kế theo hướng kết hợp giữa các đặc trưng xây dựng thủ công (như độ dài URL, số lượng ký tự đặc biệt, tần suất từ khóa nghi vấn) và đặc trưng trích xuất tự động thông qua phương pháp TF-IDF từ nội dung yêu cầu. Kết quả thực nghiệm cho thấy RF đạt hiệu năng tổng thể tốt nhất, trong khi XGBoost nổi bật về Recall và tốc độ suy luận trên bộ dữ liệu CSIC 2010, với F1-score dao động trong khoảng 0,93–0,97, Recall đạt 0,985 và ROC-AUC xấp xỉ 0,99, phản ánh khả năng nhận diện tấn công hiệu quả và hạn chế đáng kể tỷ lệ bỏ sót. Đồng thời, hệ thống cũng đáp ứng yêu cầu xử lý gần thời gian thực với thời gian suy luận trung bình khoảng 0,20 s. Bên cạnh đó, việc so sánh với nhiều mô hình khác và áp dụng kỹ thuật cân bằng dữ liệu đã củng cố nhận định rằng XGBoost là lựa chọn phù hợp nhờ khả năng xử lý dữ liệu mất cân bằng và hiệu suất tính toán tốt.

Tuy đạt được những kết quả tích cực, nghiên cứu vẫn tồn tại một số hạn chế nhất định. Bộ dữ liệu CSIC 2010 không còn phản ánh đầy đủ các dạng tấn công hiện đại, làm giảm khả năng thích ứng của mô hình trong môi trường thực tế. Ngoài ra, khả năng tổng quát hóa còn bị giới hạn khi đối mặt với dữ liệu ngoài phân bố huấn luyện. Việc sử dụng kỹ thuật tái cân bằng dữ liệu cũng có thể làm sai lệch cấu trúc phân bố ban đầu, từ đó ảnh hưởng đến độ tin cậy của dự đoán. Hơn nữa, hệ thống hiện mới được kiểm chứng trong môi trường mô phỏng và chưa tích hợp với các nền tảng an ninh thực tiễn, đồng thời chưa hỗ trợ cơ chế học liên tục.

Trong các hướng phát triển tiếp theo, nghiên cứu có thể được mở rộng thông qua việc sử dụng các bộ dữ liệu cập nhật hơn và đa dạng hóa mẫu HTTP nhằm cải thiện khả năng khái quát. Các phương pháp cân bằng dữ liệu thay thế, kỹ thuật điều chuẩn nâng cao và cơ chế học trực tuyến có thể được xem xét để nâng cao hiệu quả mô hình. Bên cạnh đó, việc áp dụng các phương pháp xử lý ngôn ngữ tự nhiên tiên tiến như BERT hoặc RoBERTa, kết hợp với các kỹ thuật giải thích mô hình, tối ưu hóa tốc độ xử lý và tích hợp vào các hệ thống bảo mật thực tế sẽ là những hướng nghiên cứu tiềm năng trong tương lai.

TÀI LIỆU THAM KHẢO

[1] A. Salam, F. Ullah, F. Amin, and M. Abrar, “Deep learning techniques for web-based attack detection in industry 5.0: A novel approach,” *Technologies,* vol. 11, no. 4, p. 107, 2023, doi: https://doi.org/10.3390/technologies11040107

[2] P. V. Hau and D. T. T. Hien, “Enhancing Web Application Security: A Deep Learning and NLP-based Approach for Accurate Attack Detection,” doi: https://doi.org/10.54654/isj.v3i20.1008

[3] M. Al Lail, A. Garcia, and S. Olivo, “Machine learning for network intrusion detection—a comparative study,” *Future Internet,* vol. 15, no. 7, p. 243, 2023, doi: https://doi.org/10.3390/fi15070243

[4] J. Wen, S. Li, Z. Lin, Y. Hu, and C. Huang, “Systematic literature review of machine learning based software development effort estimation models,” *Information and software technology,* vol. 54, no. 1, pp. 41-59, 2012, doi: https://doi.org/10.1016/j.infsof.2011.09.002

[5] C. T. Giménez, A. P. Villegas, and G. Á. Marañón, “HTTP data set CSIC 2010,” *Information Security Institute of CSIC (Spanish Research National Council)*, 2010, doi: https://impactcybertrust.org/dataset_view?idDataset=940

[6] J. C. Eunaicy and S. Suguna, "Web attack detection using deep learning models," *Materials Today: Proceedings,* vol. 62, pp. 4806-4813, 2022, doi: https://doi.org/10.1016/j.matpr.2022.03.348

[7] M. K. Baklizi, I. Atoum, M. Alkhazaleh, H. Kanaker, N. Abdullah, O. A. Al-Wesabi, and A. A. Otoom, "Web Attack Intrusion Detection System Using Machine Learning Techniques," *International Journal of Online & Biomedical Engineering,* vol. 20, no. 3, 2024, doi: https://doi.org/10.3991/ijoe.v20i03.45249

[8] Y. Pan, F. Sun, Z. Teng, J. White, D. C. Schmidt, J. Staples, and L. Krause, "Detecting web attacks with end-to-end deep learning," *Journal of Internet Services and Applications,* vol. 10, no. 1, pp. 1-22, 2019, doi: https://doi.org/10.1186/s13174-019-0115-x

[9] M. Alghawazi, D. Alghazzawi, and S. Alarifi, "Detection of sql injection attack using machine learning techniques: a systematic literature review," *Journal of Cybersecurity and Privacy,* vol. 2, no. 4, pp. 764-777, 2022, doi: https://doi.org/10.3390/jcp2040039

[10] L. Tang and Q. H. Mahmoud, "A survey of machine learning-based solutions for phishing website detection," *Machine Learning and Knowledge Extraction,* vol. 3, no. 3, pp. 672-694, 2021, doi: https://doi.org/10.3390/make3030034

[11] V.-H. Pham, H. K. Nghi, and H. Q. Nguyen, "Deception and Continuous Training Approach for Web Attack Detection using Cyber Traps and MLOps," *VNUHCM Journal of Science and Technology Development,* vol. 26, no. 2, pp. 2729-2740, 2023, doi: https://doi.org/10.32508/stdj.v26i2.4044

[12] C. Do Xuan and N. M. Son, "Enhancing web attack detection efficiency based on natural language processing techniques," *Journal of Computer Science and Cybernetics,* vol. 42, no. 1, pp. 73-87, 2026, doi: https://doi.org/10.15625/1813-9663/23407

[13] L. Breiman, "Random forests," *Machine learning,* vol. 45, no. 1, pp. 5-32, 2001, doi: https://doi.org/10.1023/A:1010933404324

[14] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning," ed: Springer series in statistics New-York, 2009, doi: https://doi.org/10.1007/978-0-387-84858-7

[15] K. P. Murphy, *Machine learning: A Probabilistic Perspective*. MIT press, 2012, doi: https://mitpress.mit.edu/9780262018029/machine-learning-a-probabilistic-perspective/

[16] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory,* vol. 13, no. 1, pp. 21-27, 1967, doi: https://doi.org/10.1109/TIT.1967.1053964

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning,* vol. 20, no. 3, pp. 273-297, 1995, doi: https://doi.org/10.1007/BF00994018.

[18] T. Chen and C. Guestrin, "Xgboost: A Scalable Tree Boosting System," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785-794. doi: https://doi.org/10.1145/2939672.2939785

[19] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the IJCAI*, 1995, vol. 14, no. 2: Montreal, Canada, pp. 1137-1145. doi: https://www.ijcai.org/Proceedings/95-2/Papers/016.pdf

[20] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research,* vol. 13, no. 2, 2012, doi: https://www.jmlr.org/papers/v13/bergstra12a.html

ABSTRACT

EVALUATING THE PERFORMANCE OF MACHINE LEARNING MODELS IN WEB ATTACK DETECTION

Le Anh Tuan[1]*

Ho Chi Minh City University of Industry and Trade

*Email: *tuanla@huit.edu.vn*

The rapid proliferation of web-based systems has been accompanied by a growing risk of sophisticated cyberattacks, including SQL injection (SQLi), cross-site scripting (XSS), and cross-site request forgery (CSRF). Traditional defense mechanisms have shown limitations in detecting emerging attack variants due to their lack of adaptability. This study proposes a machine learning–based approach for web attack detection, utilizing the HTTP CSIC 2010 dataset as the experimental foundation. The dataset is preprocessed and feature extraction is performed on HTTP requests to construct meaningful inputs for classification models. Multiple machine learning algorithms, including Random Forest (RF) and Support Vector Machine (SVM), are employed to identify anomalous behaviors. Experimental results demonstrate that the Random Forest model achieves the best performance, with an accuracy of 96.03%, an F1-score of 96.00%, and a ROC-AUC of 0.995. These findings indicate that machine learning–based approaches can significantly enhance the detection of malicious HTTP requests in modern web environments.

Keywords: Machine learning, detection system, web attacks.