PHÁT HIỆN TẤN CÔNG XSS SỬ DỤNG HỌC MÁY KẾT HỢP

Vũ Xuân Hạnh[*], *Trần Tiến Dũng*[†]

Ngày tòa soạn nhận được bài báo: 05/12/2022
Ngày nhận kết quả phản biện đánh giá: 05/06/2023
Ngày bài báo được duyệt đăng: 29/06/2023

DOI: 10.59266/houjs.2023.272

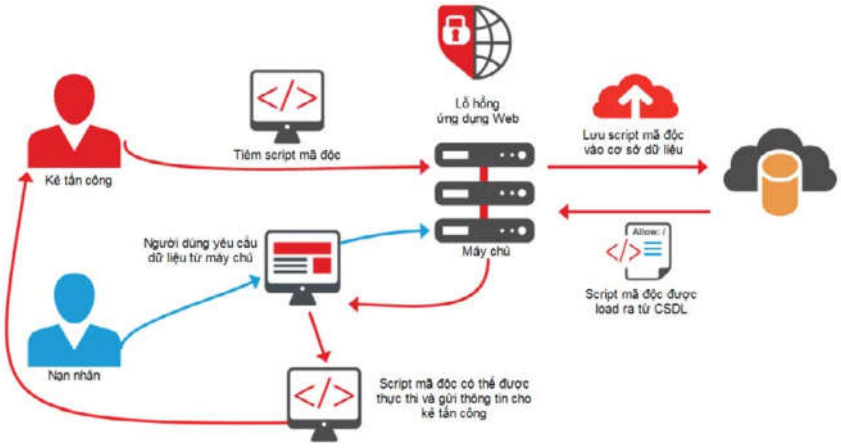
Tóm tắt: *Cross-Site Scripting là một dạng tấn công phổ biến trong các ứng dụng web. Các giải pháp hiện có như dựa trên bộ lọc, phân tích động và phân tích tĩnh không hiệu quả trong việc phát hiện các cuộc tấn công XSS không xác định. Một số nghiên cứu phát hiện các cuộc tấn công XSS sử dụng học máy đã công bố có khả năng phát hiện các cuộc tấn công XSS không xác định tuy nhiên tồn tại một số vấn đề như: bộ phân loại cơ sở đơn, tập dữ liệu nhỏ và hiệu suất mô hình chưa cao. Phương pháp học kết hợp được sử dụng trong nghiên cứu này bao gồm AdaBoost; Bagging với SVM, Extra-Trees; Stacking với Extra-Tree, Naïve Bayes và Randomforest cùng với 3 tệp dữ liệu riêng biệt, 3 nhóm đặc trưng cơ bản. Trong nghiên cứu này, mô hình đạt hiệu suất 99.32% với thuật toán Random Forest (một thuật toán thuộc nhóm Bagging).*

Từ khóa: *Tấn công XSS, Cross-site scripting, Phát hiện tấn công XSS, An ninh mạng, Học kết hợp.*

I. Đặt vấn đề

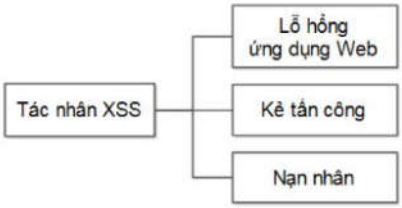
Một kiểu tấn công lớp ứng dụng đặc biệt được gọi là tấn công Cross-Site Scripting (XSS) đã trở nên nguy hiểm trong vài thập kỷ qua. Theo truyền thống, các cuộc tấn công này được sử dụng để đánh cắp thông tin cá nhân, dẫn đến khả năng mạo danh nạn nhân. Tuy nhiên, gần đây với sự phát triển của công nghệ, các cuộc tấn công này đang được sử dụng với các kỹ thuật tấn công trên mạng xã hội để tạo và khởi động các cuộc tấn công khác. Trong các cuộc tấn công XSS, kẻ tấn công có thể lấy cookie, dữ liệu nhạy cảm của nạn nhân, triển khai keyloggers ghi lại tại trình duyệt và làm hỏng uy tín của một trang web đáng tin cậy. Vấn đề phổ biến trong các kỹ thuật phòng ngừa XSS hiện có là không thể phát hiện được các cuộc tấn công XSS mới hoặc chưa biết [1]. Hình 1 cho thấy các bước liên quan đến việc khởi chạy thành công một cuộc tấn công XSS. Kẻ tấn công không nhắm mục tiêu trực tiếp vào nạn nhân, mà sử dụng các lỗ hổng trong ứng dụng Web dễ bị tổn thương làm công cụ để gửi mã độc đến trình duyệt của nạn nhân.

[*] Khoa Công nghệ Thông tin, Trường Đại học Mở Hà Nội
[†] Phòng Tổ chức-Hành chính, Trường Đại học Mở Hà Nội

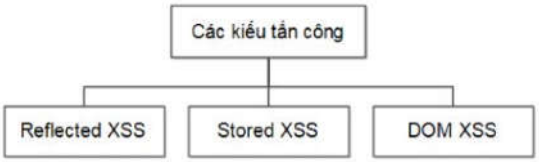


Hình 1: Các bước trong một cuộc tấn công XSS

Có ba tác nhân trong một cuộc tấn công XSS như trong hình 2. Tùy thuộc vào cách mã độc được đưa vào ứng dụng Web, các cuộc tấn công XSS được phân loại thành ba biến thể như trong hình 3 [1] [2].



Hình 2: Các tác nhân XSS



Hình 3: Các kiểu tấn công XSS

Trong bài báo này, chúng tôi sẽ đưa ra một đề xuất để xác định tấn công XSS dựa trên các đặc trưng nhanh và hiệu quả. Trong phần còn lại của bài báo được cấu trúc như sau: mục II, chúng tôi thảo luận về một số nghiên cứu liên quan đến phát hiện XSS, mục III trình bày về mô hình đề xuất, chi tiết về các đặc trưng trong XSS và các chỉ số đánh giá. Kết quả thí nghiệm của chúng tôi được phân tích trong mục IV. Kết luận được trình bày trong mục V.

II. Cơ sở lý thuyết

Đã có nhiều công trình nghiên cứu đề xuất các kỹ thuật khác nhau để phát hiện các cuộc tấn công XSS Script. Một cách tiếp cận chuẩn cho nhà phát triển ứng dụng web là sử dụng "tiệt trùng" và "thoát" để ngăn nội dung không đáng tin cậy được hiểu là mã [3]. Ngoài ra cách ly mức phân tích cú pháp có thể hạn chế dữ liệu đầu vào của người dùng trong suốt thời gian tồn tại của ứng dụng web [4]. Một kỹ thuật khác để chống lại lỗ hổng XSS là sử dụng các tiền tố không gian tên ngẫu nhiên với các phần tử ngôn ngữ đánh dấu nguyên thủy để khiến kẻ tấn công khó sử dụng các phần tử này [5]. Sự kết hợp của các kỹ thuật tĩnh và động sử dụng phân tích mã độc để ngăn dữ liệu nhạy cảm được gửi cho bên thứ ba bằng cách giám sát luồng dữ liệu trong trình duyệt [6].

Các kỹ thuật học máy đã được áp dụng để phát hiện các cuộc tấn công XSS và hấp dẫn vì chúng có thể thích ứng với những thay đổi và biến thể của các tập lệnh độc hại. Likarish và cộng sự. [7] đã đánh giá các bộ phân loại Naive Bayes, ADTree, SVM và RIPPER trong việc phát hiện sự che giấu của các tập lệnh (*dưới dạng proxy cho phần mềm độc hại*), bằng cách sử dụng các tính năng theo dõi số lần các biểu tượng

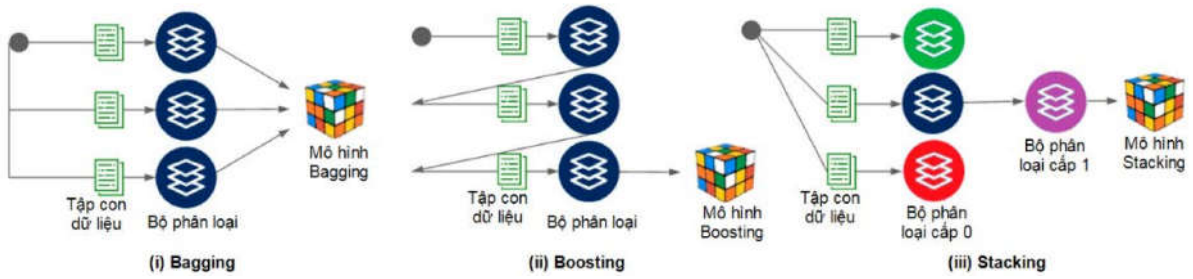
xuất hiện trong các tập lệnh lành tính và độc hại. Các bộ phân loại được đánh giá bằng xác thực chéo 10 lần cho độ chính xác là khoảng 92.00%. Fawaz và cộng sự. [8] điều tra bằng cách sử dụng SVM, k-NN và Rừng ngẫu nhiên để phát hiện và hạn chế các cuộc tấn công này, dù đã biết hay chưa biết. Việc sử dụng một bộ tính năng thú vị kết hợp cú pháp ngôn ngữ và các tính năng hành vi dẫn đến các bộ phân loại mang lại độ chính xác và độ chính xác cao trên các tập dữ liệu lớn trong thế giới thực mà không chỉ giới hạn sự chú ý đến việc làm xáo trộn. 59 đặc trưng phân loại thành hai nhóm, (1) cấu trúc, và (2) hành vi được xem xét và cho độ chính xác 97.22%. Một mô hình dựa trên tri giác đa lớp của Mokbal và cộng sự [9] đề xuất đạt được độ chính xác bảo mật là 99.32% trong việc phát hiện các cuộc tấn công. Tập dữ liệu của họ chứa tổng cộng 138,569 mẫu và trong số đó có 38,569 mẫu tấn công. Họ đã trích xuất nội dung biểu mẫu dựa trên URL, dựa trên HTML và dựa trên Script và sử dụng các tính năng này trong việc đào tạo các mô hình được đề xuất. Một số tính năng như độ dài URL và ký tự đặc biệt trong URL, thẻ HTML, sự kiện Script. Wang, Cai và Wei [10] đã đề xuất một framework dựa trên deep learning để phát hiện Script độc hại. Họ đã trích xuất các đặc trưng từ mã Script bằng cách sử dụng bộ mã hóa tự động khử nhiễu xếp chồng (SdA). Các đặc trưng này được sử dụng để đào tạo SVM hoặc mô hình hồi quy logistic. Phân loại mã độc thực hiện bằng hồi quy logistic. Tập dữ liệu được dán nhãn của họ chứa 14,783 mã Script độc hại và 12,320 mẫu lành tính. Mô hình của họ đạt độ chính xác 94.9%.

Nhằm tăng cao hiệu quả, chúng tôi đã xem xét các đặc trưng được trích chọn từ script, tìm kiếm hiệu suất rồi giảm chiều dữ liệu để giảm thời gian huấn luyện, trích chọn dữ liệu và tăng hiệu suất phát hiện.

III. Phương pháp nghiên cứu

3.1. Học máy kết hợp

Các phương pháp học máy kết hợp sử dụng các thuật toán khác nhau để đạt được tỷ lệ dự đoán tốt hơn. Thông thường, học kết hợp bao gồm những thuật toán học máy cơ bản. Hạn chế trong các phương pháp học kết hợp là đòi hỏi nhiều tính toán hơn so với một mô hình duy nhất. Có 3 phương pháp học máy kết hợp như sau: (i) Bagging: trong bagging (*tổng hợp bootstrap*), các thuật toán học yếu áp dụng trên một tập dữ liệu mẫu nhỏ và lấy trung bình tất cả các dự đoán của người học. Bagging sẽ làm giảm phương sai; (ii) Boosting: đây là một phương pháp lặp đi lặp lại, trong việc tăng trọng lượng mẫu được điều chỉnh dựa trên phân loại trước đó. Tăng cường sẽ giảm lỗi thiên vị; (iii) Stacking: trong trường hợp này, đầu ra của một mô hình được cung cấp làm đầu vào cho một mô hình khác. Xếp chồng sẽ làm giảm phương sai hoặc sai lệch dựa trên các mô hình được sử dụng [2].



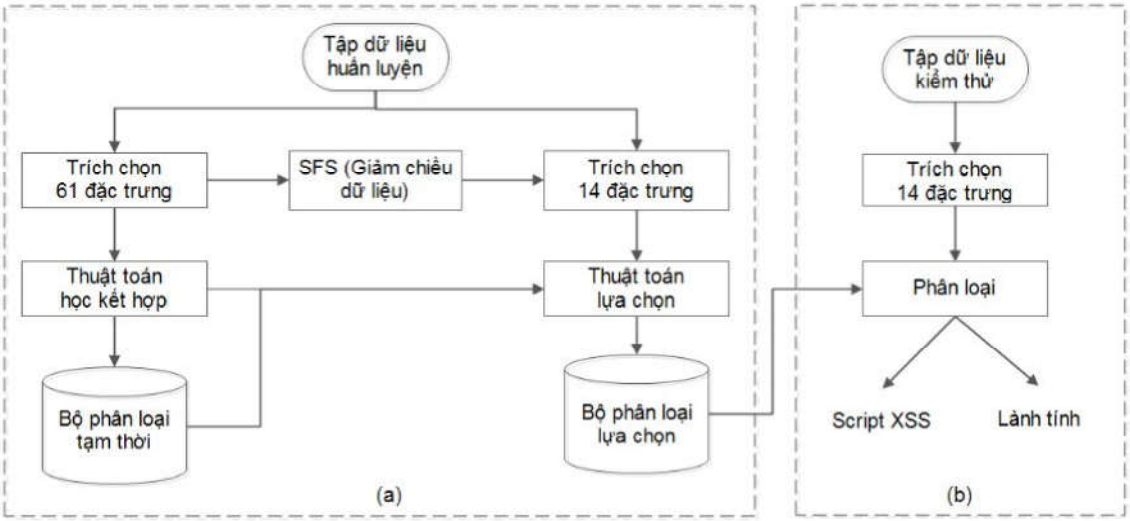
Hình 4: Phân loại học kết hợp

Phương pháp học kết hợp được sử dụng trong nghiên cứu này bao gồm: AdaBoost; Bagging với SVM, Extra-Trees; Stacking với Extra-Tree, Naïve Bayes và Random Forest (40 cây).

3.2. Giảm chiều dữ liệu

Giảm chiều dữ liệu là sự biến đổi dữ liệu từ không gian d-dim thành không gian k-dim chiều (k<d), giữ lại một số đặc trưng có ý nghĩa và quan trọng của dữ liệu gốc, loại bỏ các đặc trưng ít quan trọng thậm chí làm nhiễu. Trong nghiên cứu này, sử dụng thuật toán SFS (Sequential Forward Selection) để giảm chiều dữ liệu nhằm tăng hiệu suất của mô hình, tiết kiệm thời gian huấn luyện, kiểm thử cũng như giảm thiểu sử dụng nguồn tài nguyên. Qua thực nghiệm cho thấy SFS tốt hơn SBS (Sequential Backward Selection) đối với bài toán đang giải quyết.

Lý do để thực hiện việc giảm chiều dữ liệu từ tập các đặc trưng được trích chọn bởi lẽ: việc lựa chọn các đặc trưng chưa chắc là tốt nhất vì chưa xác định thành phần nào quan trọng hơn. Trong trường hợp khác, lượng thông tin các đặc trưng cung cấp có tỷ trọng tương đương, khi đó lược bớt đặc trưng nào cũng sẽ dẫn đến mất một lượng thông tin đáng kể. Do đó, để đảm bảo rằng các đặc trưng sử dụng là quan trọng và đủ để phân loại, chúng tôi giảm chiều để đi đến một tập các đặc trưng quan trọng.



Hình 5: Mô hình phát hiện đề xuất

3.3. Mô hình phát hiện

Mô hình phát hiện Script XSS dựa trên học kết hợp đề xuất được chia thành 2 giai đoạn được minh họa tại hình 5 như sau:

(a) Giai đoạn huấn luyện: Tệp dữ liệu huấn luyện bao gồm Script XSS ***và lành tính.*** Ở giai đoạn 1, 61 đặc trưng được trích chọn sử dụng 4 thuật toán tại mục 3.1 để lựa chọn ra thuật toán có hiệu suất cao nhất. Sau đó, sử dụng thuật toán giảm chiều dữ liệu ***để*** giảm số đặc trưng cần trích chọn xuống. Sau đó, sử dụng thuật toán đã lựa chọn với hiệu suất tốt nhất để đưa ra bộ phân loại.

(b) Giai đoạn phát hiện: các Script XSS trong tệp kiểm thử ***được*** trích chọn các đặc trưng, sử dụng bộ phân loại đã được huấn luyện để xác định Script XSS

3.4. Trích chọn đặc trưng

3.4.1. Giới thiệu

Hiện nay, việc phát triển các trang web sử dụng ngôn ngữ JavaScript làm cho chúng năng động và dễ tương tác hơn. Đó là phía máy khách cho phép mã nguồn được thực thi trong trình duyệt web thay vì trên máy chủ. Điều này cho phép các chức năng chạy sau khi tải trang web mà không cần giao tiếp với máy chủ, chẳng hạn như tạo cảnh báo lỗi trước khi gửi thông tin đến máy chủ. Các tập lệnh có thể được chèn vào trong HTML hoặc có thể được tham chiếu trong một tệp .js riêng biệt. JavaScript là một lựa chọn tốt cho những kẻ tấn công thực hiện các cuộc tấn công của chúng và phát tán chúng trên Internet, bởi vì phần lớn các trang web sử dụng JavaScript và nó được hỗ trợ bởi tất cả các trình duyệt web. Do đó, nó là mục tiêu của nhiều cuộc tấn công XSS, SQL injection và tải xuống thụ động.

Nghiên cứu này tập trung vào các đặc trưng được trích chọn từ Script, chỉ cần xem xét script *mà không cần quan tâm đến các đặc trưng mạng, các danh sách đã có* từ trước...

3.4.2. Lựa chọn đặc trưng XSS

Có thể trích chọn rất nhiều các đặc trưng phù hợp của Script để phân loại. Các đặc trưng trong nghiên cứu này được phân loại thành 3 nhóm, (1) cấu trúc, (2) hành vi và (3) thống kê. Tổng cộng, 61 đặc trưng được xem xét.

Bảng 1: Các đặc trưng được trích chọn

Nhóm đặc trưng	Đặc trưng	Số lượng
a) Đặc trưng cấu trúc		
Dấu câu nối	[] | &# | == | // | ">< | '><	6
Dấu câu	& % / \ + ' ? ! ; # = [ ] $ ( ) * , - < > @ _ : . { } ~ space | &lt ^ `	33
b) Đặc trưng hành vi		
Attributes	hasSRC | hasHref | hasCookie	3
Events	hasOnload | hasOnError	2
External File	hasJS	1
Functions	hasAlert | hasEval	2
Methods	hasStringfromCharCode | hasSearch	2
Objects	hasDocument | hasWindow | hasIframe | hasLocation	4
Protocol	hasHTTP	1
Reserve	hasVal	1
Tags	hasSCRIPT | hasDIV | hasIMG	3
c) Đặc trưng thống kê		
Thống kê	LettersRatio | NumbuersRatio | SymbolsRatio	3
Tổng cộng		61

- Đặc trưng cấu trúc: là tập hợp đầy đủ các ký tự không phải chữ và số có thể xuất hiện trong Script. Những điều này có thể xảy ra trong bất kỳ tập lệnh nào, nhưng nếu kẻ tấn công đang sử dụng các kỹ thuật để lẩn tránh sự phòng vệ trên các ứng dụng Web thì điều này có thể thay đổi phạm vi ký tự được sử dụng trong tập lệnh. Để đưa ra một ví dụ đơn giản, tập lệnh độc hại có thể thêm dấu cách hoặc ký hiệu không cần thiết giữa các lệnh hoặc thẻ, chẳng hạn như < \ sc ri pt >. Một kịch bản lành tính sẽ không làm điều này. Như một ví dụ khác, hãy xem xét quyền truy cập cookie được tách thành hai phần và việc sử dụng dấu + để kết hợp lại toàn bộ lệnh một lần

nữa, document +′ .′ + cookie. Cũng bao gồm trong các đặc trưng cấu trúc là sự kết hợp của các ký tự có thể được sử dụng để xây dựng các tập lệnh độc hại. Bảng 1(a) có 33 ký tự không phải chữ, số và 6 tổ hợp khác trong số này được xem xét. Các đặc trưng có thể được đo bằng nhiều cách khác nhau, tuy nhiên trong nghiên cứu này các đặc trưng này nhận các giá trị 0/1 cho sự có hay không xuất hiện trong tập lệnh.

- Đặc trưng hành vi: là một tập hợp các lệnh và chức năng có thể được sử dụng trong Script. Kẻ tấn công có thể sử dụng chúng một cách đáng ngờ và khác với các Script lành tính. Nghĩa là, các Script lành tính không cần che giấu ý định mã, ngược lại, Script XSS sẽ sử dụng một loạt lệnh để tạo tập lệnh độc hại. Ví dụ: thường xuyên sử dụng hàm eval, sử dụng các hàm khử nhiễu trong tập lệnh hoặc bao gồm tập lệnh độc hại trong thẻ hình ảnh. Có 19 đặc trưng trong phân loại này được trích chọn gồm: thuộc tính, thẻ, hàm, giao thức,..thể hiện tại bảng 1 (b). Cũng như đặc trưng cấu trúc, sử dụng các giá trị 0/1 cho biết đặc trưng này không hoặc không xuất hiện trong tập lệnh.

- Đặc trưng thống kê: nhận thấy sự phân bố không đồng đều của các ký tự là chữ, số và đặc biệt. 3 đặc trưng thống kê trong bảng 1 (c) được trích chọn để phân loại Script lành tính và XSS.

3.4.3. Phương pháp đánh giá

Một số độ đo dùng để đánh giá mô hình đề xuất bao gồm: TPR, FPR, FNR, PPV, ACC, F1 và được tính toán theo các công thức dưới đây:

- Độ chính xác (PPV-Positive Predictive Value) được tính theo công thức:

$$PPV = \frac{TP}{TP + FP} \quad (1)$$

- Tỷ lệ dương tính đúng (TPR), hay độ nhạy, được tính theo công thức:

$$TPR = \frac{TP}{TP + FN} \quad (2)$$

- Tỷ lệ dương tính giả (FPR) hay còn gọi "nhầm lẫn", được tính theo công thức:

$$FPR = \frac{FP}{FP + TN} \quad (3)$$

- Tỷ lệ âm tính giả (FNR) hay còn gọi "bỏ sót", được tính theo công thức:

$$FNR = \frac{FN}{FN + TP} \quad (4)$$

- *Độ đo F1 được tính theo công thức:*

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (5)$$

- *Độ chính xác toàn cục, hay độ chính xác chung ACC, được tính theo công thức:*

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

trong đó, TP là số lượng Script XSS được phân loại đúng, TN là số lượng Script lành tính được phân loại đúng, FP là số lượng Script lành tính bị phân loại sai thành Script XSS và FN là số lượng các Script XSS bị phân loại sai thành Script lành tính.

- Tỷ lệ kiểm thử (Detect Rate – DR), được tính theo công thức:

$$DR = \frac{NTestCorrect}{NTest} \quad (7)$$

trong đó, NtestCorrect là số mẫu kiểm thử chính xác, NTest là tổng số mẫu kiểm thử.

IV. Kết quả và thảo luận

4.1. Tập dữ liệu huấn luyện và kiểm thử

Tập dữ liệu huấn luyện được tập hợp từ 2 nguồn [11] [12] gồm 34,381 Script lành tính và 22,555 Script XSS. Các Script XSS được gán nhãn 1 và Script lành tính được gán nhãn 0. Tập dữ liệu kiểm thử lấy từ [13] với 6,581 Script XSS không gán nhãn.

Bảng 2: Dữ liệu huấn luyện và kiểm thử

Tập dữ liệu huấn luyện và kiểm thử	**Script**	
	Lành tính	***XSS***
Tập huấn luyện	34,381	22,555
Tập kiểm thử		6,581

4.2. Lựa chọn thuật toán

Với tập dữ liệu huấn luyện, sử dụng một số thuật toán học máy kiểm tra chéo 10 lần để xác định hiệu suất của mô hình. Dựa vào kết quả tại bảng 3, với ACC và F1 lần lượt bằng 99.66% và 99.57% kèm theo tỷ lệ âm tính giả và dương tính giả là 0.55% và 0.20% thuật toán Rừng ngẫu nhiên cho hiệu quả tốt nhất. Mặt khác, thử nghiệm thuật toán với lần lượt 35, 40, 45 cây được ACC lần lượt là: 99.47%, 99.66%, 99.52%. Do đó, trong nghiên cứu này lựa chọn thuật toán Random Forest với số cây là 40 để huấn luyện mô hình và kiểm thử.

Bảng 3: Hiệu suất của một số kỹ thuật học máy

Thuật toán	**ACC**	**F1**
Random Forest (40 trees)	99.66%	99.57%
Stacking (J48, Naïve Bayes)	99.30%	99.12%
AdaBoostM1 (J48)	99.62%	99.51%
Bagging (J48)	99.45%	99.30%

Sử dụng thuật toán Random Forest xây dựng mô hình cùng với tập dữ liệu huấn luyện tại bảng 2. Thống kê trong bảng 4 cho thấy, mô hình đề xuất của chúng tôi cùng 61 đặc trưng Script XSS cho ACC tốt nhất là 99.66%, tỷ lệ âm tính giả là 0.55%, tỷ lệ dương tính giả là 0.2%.

Bảng 4: So sánh các công bố trước đó

Đề xuất	**Sử dụng**	ACC
Likarish và cộng sự [7]	SVM, ADTree, Naïve Bayes,…	92.00%
Fawaz và cộng sự. [8]	SVM, KNN, Random Forest	97.22%
Mokbal và cộng sự. [9]	Perception đa lớp	99.32%
Wang, Cai và Wei [10]	SVM, hồi quy logisyic	94.90%
Của chúng tôi	RF (40)	99.66%

Bảng 5: Đặc trưng được chọn khi giảm chiều bằng thuật toán SFS

Nhóm đặc trưng	**Đặc trưng**	**Số lượng**
a) Đặc trưng cấu trúc		
Dấu câu	% ? ; ('	5
b) Đặc trưng hành vi		
Tags	hasSCRIPT	1
c) Đặc trưng thống kê		
Thống kê	LettersRatio | NumbuersRatio | SymbolsRatio	3
Tổng cộng		9

Bảng 5 liệt kê phân loại các đặc trưng của tệp huấn luyện sau khi giảm chiều dữ liệu với thuật toán SFS, số đặc trưng còn lại là 9 đặc trưng.

Bảng 6: So sánh mô hình 61, 9, 30 đặc trưng

Đặc trưng	Thời gian huấn luyện	ACC
61 (bảng 1)	11.21 giây	99.66%
9 (bảng 5)	4.8 giây	98.80%

Kết quả huấn luyện tệp dữ liệu tại bảng 2 với máy tính i7-9750H 2.60G, GeForce GTX 1050 3GB, RAM 32GB tại bảng 6 cho thấy: (i) thời gian huấn luyện giảm đáng kể, cụ thể đối với mô hình sử dụng 9 đặc trưng sử dụng 4.8 giây giảm 57.18% so với thời gian huấn luyện mô hình 61 đặc trưng; ACC toàn cục của mô hình 9 đặc trưng chỉ giảm 0.84% so với mô hình sử dụng 61 đặc trưng.

4.3. Kết quả và đánh giá

Sử dụng mô hình đề xuất với thuật toán RF sử dụng 40 cây kiểm thử trên tệp dữ liệu 6,581 mẫu XSS Script cho DR lần lượt là 97.34% ***và*** 96.34% được thể hiện tại Bảng 8.

Bảng 8: Hiệu suất kiểm thử

Mô hình	Phát hiện	DR
61 đặc trưng	6406	97.34%
9 đặc trưng	6365	96.72%

Từ kết quả kiểm thử mô hình, so sánh với một số nghiên cứu trước được thể hiện tại Bảng 4 cho thấy mô hình của chúng tôi có hiệu suất cao hơn. Mặt khác, trong 9 đặc trưng sau khi đã giảm chiều dữ liệu vẫn tồn tại 3 đặc trưng thông mà trong nghiên cứu của Fawaz và cộng sự [8] không đề xuất, như vậy có thể kết luận rằng trong các đặc trưng về cấu trúc và hành vi có rất nhiều các đặc trưng ít quan trọng và 3 đặc trưng về thống kê là những đặc trưng quan trọng. Ngoài ra, việc huấn luyện dựa trên 2 tập dữ liệu và kiểm thử trên 1 tập dữ liệu hoàn toàn tách biệt tăng độ tin cậy đối với hiệu suất của mô hình đề xuất.

V. Kết luận

Bảo mật và an toàn thông tin là nhu cầu rất cấp thiết hiện nay, với mục đích hạn chế các cuộc tấn công trên mạng nói chung và các cuộc tấn công XSS Script nói riêng. Chúng tôi đã nghiên cứu chi tiết các đặc trưng của XSS Script, đưa ra 3 nhóm đặc trưng, sau đó giảm chiều dữ liệu, lựa chọn 9 đặc trưng quan trọng để xây dựng mô hình. Việc giảm chiều dữ liệu trong đề xuất của chúng tôi làm tăng hiệu quả của việc sử dụng tài nguyên, giảm thiểu thời gian trích chọn đặc trưng, huấn luyện và kiểm thử. Random Forest là thuật toán thuộc nhánh Bagging họ học kết hợp, thuật toán này được sử dụng khá nhiều trong các nghiên cứu gần đây của các nhà khoa học, tuy nhiên trong nghiên cứu này chúng tôi kiểm thử với các nhánh học kết hợp khác để đề xuất thuật toán tốt nhất để xây dựng mô hình phù hợp với dữ liệu và bài toán phân loại nhị phân.

Với đề xuất này, ứng dụng được triển khai sẽ hỗ trợ dev trong vấn đề tăng cường bảo mật cho các ứng dụng web khi bỏ sót hoặc chưa phát hiện ra các lỗ hổng. Trong tương lai, chúng tôi tiếp tục nghiên cứu các bộ đặc trưng khác nhau và sử dụng các tập dữ liệu lớn hơn để giúp phát hiện tấn công XSS Script chính xác và hiệu quả hơn.

Tài liệu tham khảo:

[1]. Sarmah, U., Bhattacharyya, D. K., & Kalita, J. K., "A survey of detection methods for XSS attacks," *Journal of Network and Computer Applications,* pp. 113-143, 2018.

[2]. PMD Nagarjun1, Shaik Shakeel Ahamad2, "Ensemble Methods to Detect XSS Attacks," *International Journal of Advanced Computer Science and Applications,* vol. 11, pp. 695-700, 2020.

[3]. Weinberger, J., Saxena, P., Akhawe, D., Finifter, M., Shin, R., Song, D, "A systematic analysis of XSS sanitization in web application frameworks," *Lecture Notes in Computer Science,* vol. 6879, pp. 150-171, 2011.

[4]. Nadji, Y., Saxena, P., Song, D,

"Document structure integrity: a robust basis for cross-site scripting defense," *Network and Distributed System Security Symposium,* 2009.

[5]. Van Gundy, M., Chen, H., "Noncespaces: using randomization to defeat cross-site scripting attacks," *Comput. Secur.,* vol. 31, no. 4, pp. 612-628, 2012.

[6]. Vogt, P., Nentwich, F., Jovanovic, N., Kirda, E., Kruegel, C., Vigna, G, "Cross site scripting prevention with dynamic data tainting and static analysis," *Network and Distributed System Security Symposium, p. 12. Internet Society ,* 2007.

[7]. Likarish, P., Jung, E., Jo, I., "Obfuscated malicious JavaScript detection using classification techniques," *Malicious and Unwanted Software,* pp. 47-54, 2009.

[8]. Fawaz .M, Jacob .H, "Detecting Cross-Site Scripting Attacks Using Machine Learning," *The International Conference on Advanced Machine Learning Technologies and Applications,* 2018.

[9]. F. M. M. Mokbal, W. Dan, A. Imran, L. Jiuchuan, F. Akhtar, and W. Xiaoxi, "MLPXSS: An Integrated XSS-Based Attack Detection Scheme in Web Applications Using Multilayer Perceptron Technique," *IEEE,* vol. 7, p. 100567–100580, 2019.

[10]. Y. Wang, W. Cai, and P. Wei, "A deep learning approach for detecting malicious JavaScript code," *Secur. Commun. network,* vol. 9, no. 11, pp. 1520-1535, 2016.

[11]. Kaggle, "Cross site scripting - xss dataset for deep learning," [Online]. Available: https://www.kaggle.com/datasets/syedsaqlainhussain/cross-site-scripting-xss-dataset-for-deep-learning. [Accessed 09 2022].

[12]. Github, "Cross-Site-Scripting-XSS-Dataset," [Online]. Available: https://github.com/fmereani/Cross-Site-Scripting-XSS. [Accessed 09 2022].

[13]. Github, "XSS Payload List," [Online]. Available: https://github.com/payloadbox/xss-payload-list/blob/master/Intruder/xss-payload-list.txt. [Accessed 09 2022].

DETECT XSS ATTACK USING ENSEMBLE LEARNING

Vu Xuan Hanh[‡], Tran Tien Dung[§]
Email: hanhvx@hou.edu.vn

Abstract: *Cross-site scripting is a common type of attack in web applications. Existing solutions such as filter-based, dynamic, and static analysis are ineffective in detecting unknown XSS attacks. Some published studies on using machine learning to detect XSS attacks can detect unknown XSS attacks, but they create some issues, such as single base classifiers, small datasets, and low model performance. The ensemble learning method used in this study includes AdaBoost; Bagging with SVM, Extra-Trees; Stacking with Extra-Tree and Naïve Bayes, and Randomforest with three separate data files and three basic feature groups. In this study, the model achieved a performance of 99.32% with the Random Forest algorithm.*

Keywords: *XSS attack, Cross-site scripting, Detection of XSS attack, Network security, Ensemble learning.*

[‡] Factulty of Information technology, Hanoi Open University

[§] Department of Personnel and Administration