PHÁT HIỆN XÂM NHẬP WEBSITE DỰA TRÊN CÂY QUYẾT ĐỊNH VÀ BỘ DỮ LIỆU HUẤN LUYỆN IDS2021-WEB (PHẦN II)

TS. Nguyễn Văn Căn, Đoàn Ngọc Tú, Đỗ Đình Quang

Đại học Kỹ thuật - Hậu cần Công an nhân dân

Phần I của bài báo, nhóm tác giả đã trình bày cách thức xây dựng bộ dữ liệu IDS2021-WEB. Tại phần II này, nhóm tác giả sẽ trình bày cách thức sử dụng thuật toán Cây quyết định (Decision Tree - DT) trong mô hình xây dựng hệ thống phát hiện xâm nhập ứng dụng website dựa trên bộ dữ liệu IDS2021-WEB. Đánh giá hiệu quả của thuật toán DT với một số thuật toán phổ biến khác và đề xuất mô hình tổng thể trong xây dựng hệ thống phát hiện xâm nhập cho ứng dụng website.

THUẬT TOÁN DT

Thuật toán DT với những ưu điểm của mình được đánh giá là một công cụ mạnh, phổ biến và đặc biệt thích hợp cho khai phá dữ liệu (data mining) nói chung và kiểu tấn công dữ liệu nói riêng. Ưu điểm của DT có thể kể đến như xây dựng tương đối nhanh, đơn giản và dễ hiểu.

Thuật toán Cây phân loại và hồi quy (Classification and Regression Tree - CART) là một loại thuật toán của DT, nó hỗ trợ các biến mục tiêu số (hồi quy) và không tính toán các bộ quy tắc. CART thường sử dụng phương pháp Gini để tạo các điểm phân chia. Tương tự như phương pháp tính độ lợi thông tin, Gini index được dùng để đánh giá việc phân chia nút có tốt hay không. Phương pháp Gini được hiểu cụ thể như sau:

- Là phương pháp hướng đến đo lường tần suất một đối tượng dữ liệu ngẫu nhiên trong tập dữ liệu ban đầu được phân loại không chính xác, trên cơ sở đối tượng dữ liệu đã nằm trong một tập con được phân ra từ tập dữ liệu ban đầu, có dán nhãn thể hiện thuộc tính chung bất kỳ của các đối tượng còn lại trong tập con này, giá trị phân loại chính là nhãn của tập con.

- Gini index chính là chỉ số đo lường mức độ đồng nhất hay nhiễu loạn của thông tin, hay sự khác biệt về các giá trị mà mỗi điểm dữ liệu trong một tập con, hoặc một nhánh của DT. Công thức của Gini index có thể dùng cho cả dữ liệu rời rạc và liên tục. Nếu điểm dữ liệu thuộc về một nút và có chung thuộc tính bất kỳ thì nút này thể hiện sự đồng nhất lúc này Gini=0 và ngược lại Gini sẽ lớn.

Công thức tổng quát của Gini index:

$$G(p) = \sum_{i=1}^{n} p_i (1 - p_i) = 1 - \sum_{i=1}^{n} {p_i}^2 \quad (1)$$

Công thức trên để tính giá trị Gini của một nút, khi có nhiều cách phân nhánh, mỗi cách có thể phân ra một số nút nhất định. Trong đó, $p_i = N_i/N$ với N_i là số lượng phần tử lớp thứ i, N là tổng số lượng phần tử ở lớp đó. Công thức sau được sử dụng để tìm ra các phân chia tối ưu nhất:

$$G_{split} = \sum_{i=1}^{n} \frac{N_i}{N} G(i) \quad (2)$$

Trong đó:

- N_i là số điểm dữ liệu có trong nút của nhánh được phân.

- N là số điểm dữ liệu có trong nút được dùng để phân nhánh.

- Hệ số G_{split} càng nhỏ thì cách phân nhánh đó càng tối ưu.

Nghiên cứu sử dụng thuật toán CART làm thuật toán phân loại chính của mô hình thực nghiệm.

CÁC CHỈ SỐ ĐÁNH GIÁ

Confusion matrix

Confusion matrix là một trong những số liệu trực quan và dễ nhất được sử dụng để xác định chính xác

của mô hình. Nó được sử dụng cho bài toán phân loại trong đó đầu ra có thể có hai hoặc nhiều lớp. Ma trận nhầm lẫn là một bảng có hai chiều ("Thực sự" và "Dự đoán") và tập hợp các lớp trong cả hai chiều. Phân loại thực sự là các hàng và dự đoán là cột.

Bảng 1. Confusion matrix

		Dự đoán	
		Dương tính	Âm tính
Thực tế	Dương tính	TP	FN
	Âm tính	FP	TN

- *True Positive (TP):* Là các trường hợp khi lớp thực tế của điểm dữ liệu là 1 (Đúng) và dự đoán cũng là 1 (Đúng).

- *True Negative (TN):* Là các trường hợp khi lớp thực tế của điểm dữ liệu là 0 (Sai) và dự đoán cũng là 0 (Sai).

- *False Positive (FP):* Là trường hợp khi lớp thực tế của điểm dữ liệu là 0 (Sai) và dự đoán là 1 (Đúng).

- *False Negative (FN):* Là các trường hợp khi lớp thực tế của điểm dữ liệu là 1 (Đúng) và dự đoán là 0 (Sai).

Kịch bản lý tưởng mà là mô hình sẽ cho kết quả 0 FP và 0 FN.

Accuracy

Là tỷ lệ phần trăm của các mẫu được phân loại chính xác trên tổng số mẫu được đánh giá. Cách đánh giá này đơn giản tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử. Có hai cách tính độ chính xác là sử dụng thư viện có sẵn hoặc tự xây dựng hàm.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Độ chính xác có thể không phải là thước đo hiệu suất tốt khi số lượng mẫu được đánh giá là không tương xứng. Do đó cần phải tạo các ma trận nhầm lẫn để khắc phục nhược điểm của cách đánh giá dựa trên độ chính xác.

Precision, Recal và F1-score

Precision là tỉ lệ thực sự dương tính trên tổng số các trường hợp được mô hình dán nhãn "Positive".

Recall là tỉ lệ phân loại dương tính đúng trên tổng số các trường hợp dương tính.

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

Vì vậy, về cơ bản nếu muốn tập trung nhiều hơn vào việc giảm thiểu các âm tính giả, thì cần Recall càng gần 100% càng tốt mà không cần quá chính xác. Nếu muốn tập trung vào việc giảm thiểu các dương tính giả, thì cần Precision càng gần 100% càng tốt. Trong bài toán phát hiện tấn công ứng dụng website, ta tuân theo quy tắc "báo nhầm hơn bỏ sót", tức là sẽ quan tâm đến chỉ số Recall nhiều hơn.

$$F1 = 2 \times \frac{1}{\frac{1}{Recall} + \frac{1}{Precision}} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$F_\beta = (1 + \beta^2) \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}$$

Trường hợp tổng quát của F1-score là $F_{\beta\text{-score}}$. F1 chính là một trường hợp đặc biệt của F_β khi $\beta=1$. Khi $\beta>1$, recall được coi trọng hơn precision, khi $\beta<1$, precision được coi trọng hơn. Hai đại lượng β thường được sử dụng là $\beta=2$ và $\beta=0.5$.

ĐÁNH GIÁ KẾT QUẢ

Phân loại nhị phân

Phân loại nhị phân là quá trình tiến hành việc phân lớp dữ liệu vào một trong hai lớp tấn công hoặc không tấn công.

Dựa vào các kết quả huấn luyện ở Hình 1, nhận thấy mô hình DT đạt độ chính xác cao (0.9965). Mô hình Naive Bayes có tốc độ nhanh nhất nhưng không đạt độ chính xác lớn (0.772). Mô hình DT cũng là thuật toán có tốc độ huấn luyện khá nhanh.

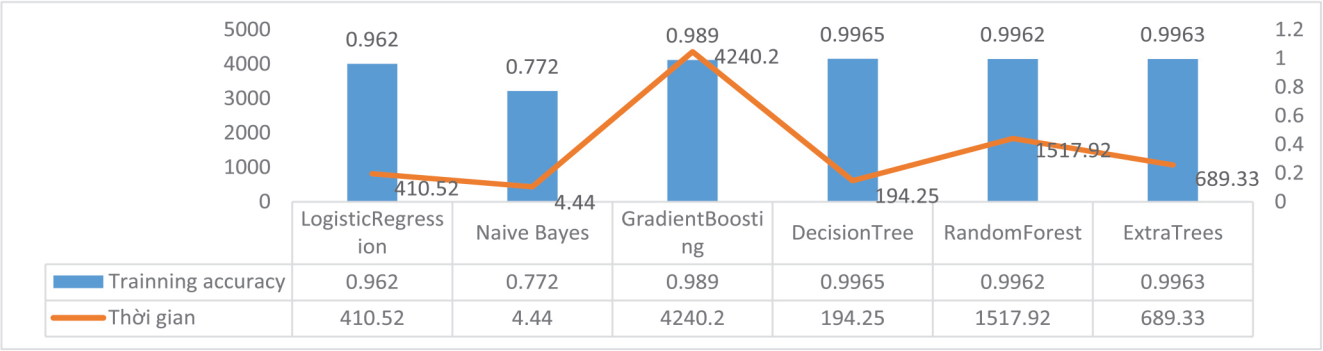
Đồ thị trong Hình 2 thể hiện rất nhiều thông tin. Có thể thấy quá trình kiểm tra phát hiện tấn công này, mô hình NaiveBayes có hiệu suất kém nhất. Các chỉ số Precision, Recall và F1-score, có độ cân bằng cao thể hiện ở các mô hình là mô hình Gradient Boosting, DT, Rừng ngẫu nhiên và Extra Tree. Trong đó, mô hình DT có thời gian kiểm tra nhanh nhất (2.63 giây). Do đó, nếu lựa chọn thuật toán để tiến hành áp dụng học máy trong phát hiện có hay không tấn công vào ứng dụng website thì DT là thuật toán lựa chọn phù hợp.

Phân loại đa lớp

Phân lớp đa lớp là quá trình phân lớp với số lượng lớp lớn hơn hai. Như vậy, tập hợp dữ liệu trong miền xem xét được phân chia thành nhiều lớp chứ không đơn thuần chỉ là hai lớp như trong bài toán phân lớp nhị phân. Về bản chất, phân lớp nhị phân là trường hợp riêng của phân lớp đa lớp. Trong đó, các lớp là các loại hình tấn công.

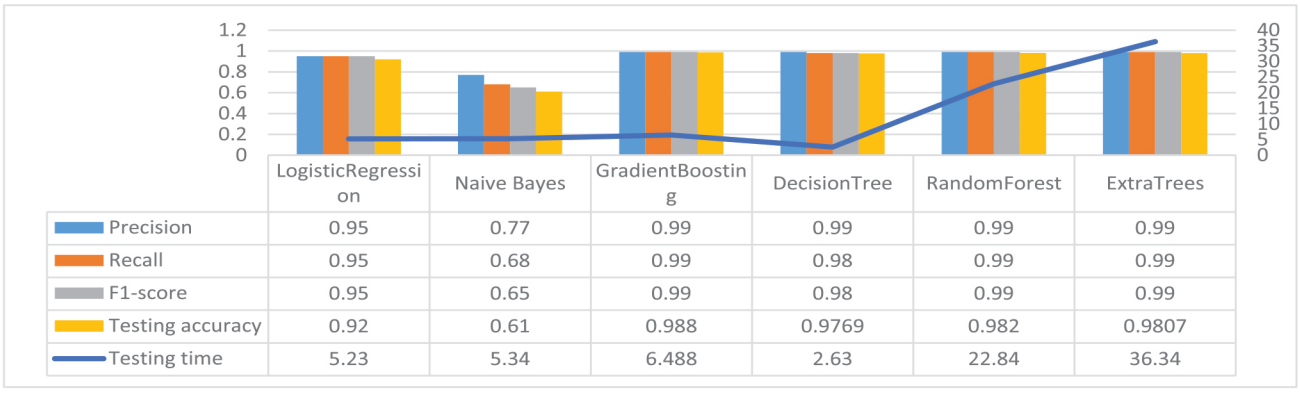
Đối với hiệu suất của quá trình huấn luyện để phát hiện loại tấn công được thể hiện tại Hình 3, có thể thấy mô hình Gradient Boosting có thời gian huấn luyện lớn nhất. Mô hình DT có kết quả tốt nhất (99.65). Trong kiểm tra với tập kiểm tra thể hiện tại Hình 4, ta thấy các chỉ số của thuật toán DT có kết quả tốt nhất và thời gian nhanh nhất.

Đánh giá trên ma trận nhầm lẫn tại Hình 5, đường chéo chính trong các ma trận chính là số lượng các mẫu thử được phân loại đúng. Có 4 loại lưu lượng tấn công không có độ phân loại chính xác cao ở tất cả các mẫu, đó là: BruteForce-Web, BruteForce-XSS, SQL Injection và Infilteration. Đối với các loại tấn công có số lượng mẫu ít (dưới 2000) thì việc phân loại nhầm là hoàn toàn có thể xảy ra. Tuy nhiên đối với tấn công xâm nhập (Infilteration) có thể có các thuộc tính của lưu lượng tấn công xâm nhập rất giống với các lớp khác, đặc biệt là lưu lượng mạng bình thường. Tức là các mô hình không thể phân biệt được hành vi xâm nhập trái phép và sử dụng hợp lệ.



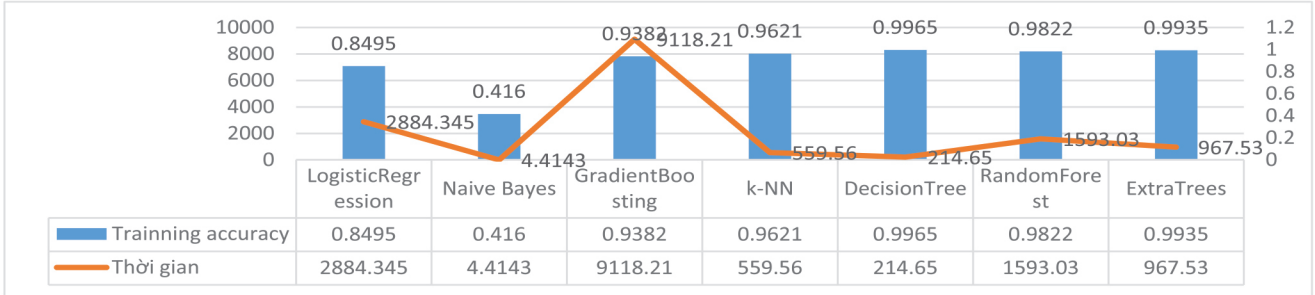
	LogisticRegression	Naive Bayes	GradientBoosting	DecisionTree	RandomForest	ExtraTrees
Trainning accuracy	0.962	0.772	0.989	0.9965	0.9962	0.9963
Thời gian	410.52	4.44	4240.2	194.25	1517.92	689.33

Hình 1. Hiệu suất quá trình huấn luyện cho phát hiện tấn công



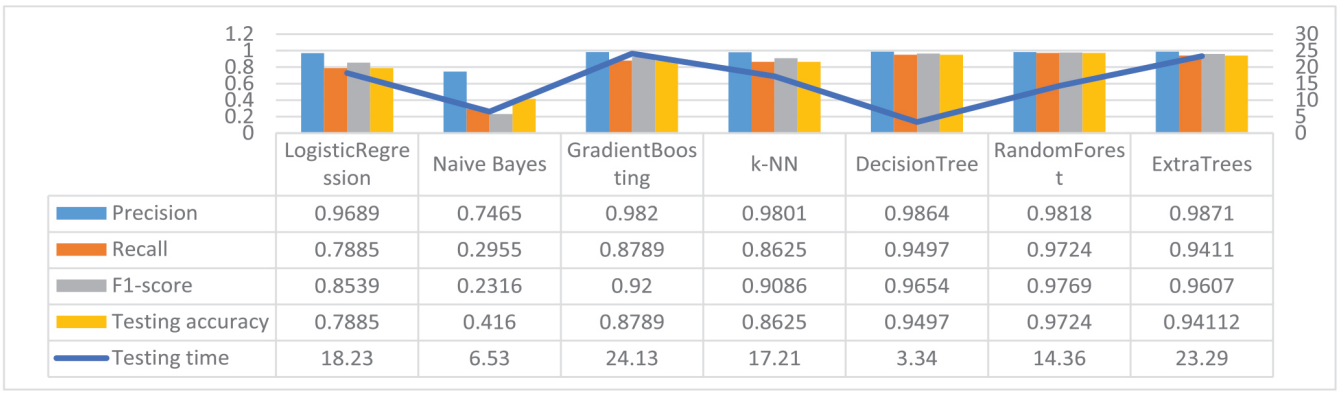
	LogisticRegression	Naive Bayes	GradientBoosting	DecisionTree	RandomForest	ExtraTrees
Precision	0.95	0.77	0.99	0.99	0.99	0.99
Recall	0.95	0.68	0.99	0.98	0.99	0.99
F1-score	0.95	0.65	0.99	0.98	0.99	0.99
Testing accuracy	0.92	0.61	0.988	0.9769	0.982	0.9807
Testing time	5.23	5.34	6.488	2.63	22.84	36.34

Hình 2. Hiệu suất quá trình kiểm tra cho phát hiện tấn công



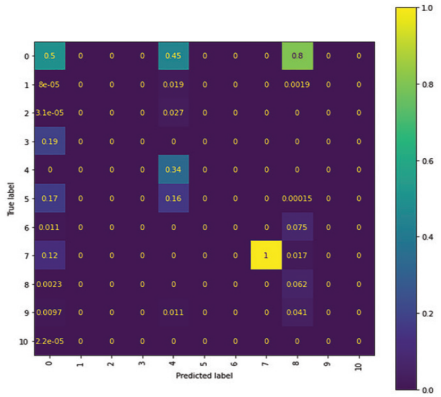
	LogisticRegression	Naive Bayes	GradientBoosting	k-NN	DecisionTree	RandomForest	ExtraTrees
Trainning accuracy	0.8495	0.416	0.9382	0.9621	0.9965	0.9822	0.9935
Thời gian	2884.345	4.4143	9118.21	559.56	214.65	1593.03	967.53

Hình 3. Hiệu suất quá trình huấn luyện cho phân loại tấn công

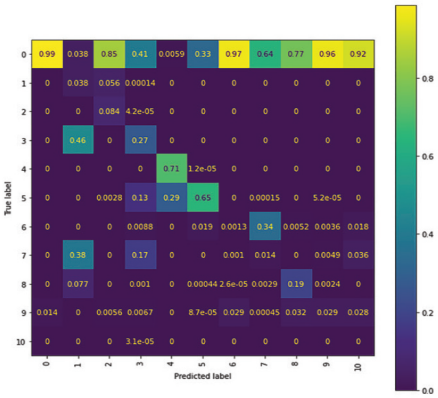


	LogisticRegression	Naive Bayes	GradientBoosting	k-NN	DecisionTree	RandomForest	ExtraTrees
Precision	0.9689	0.7465	0.982	0.9801	0.9864	0.9818	0.9871
Recall	0.7885	0.2955	0.8789	0.8625	0.9497	0.9724	0.9411
F1-score	0.8539	0.2316	0.92	0.9086	0.9654	0.9769	0.9607
Testing accuracy	0.7885	0.416	0.8789	0.8625	0.9497	0.9724	0.94112
Testing time	18.23	6.53	24.13	17.21	3.34	14.36	23.29

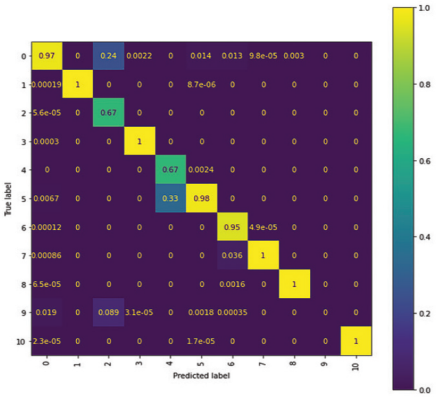
Hình 4. Hiệu suất quá trình kiểm tra cho phát hiện tấn công



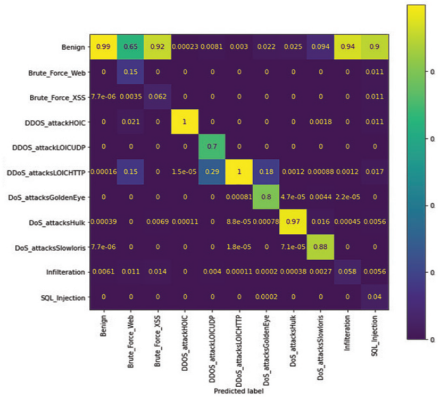
LogisticRegression



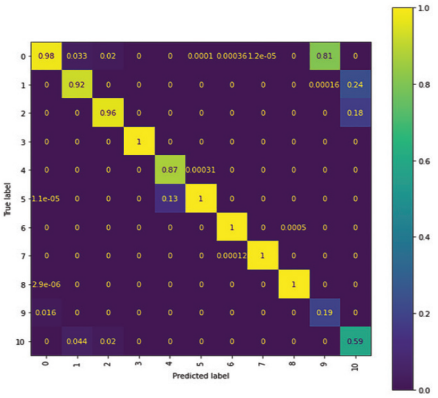
Naive Bayes



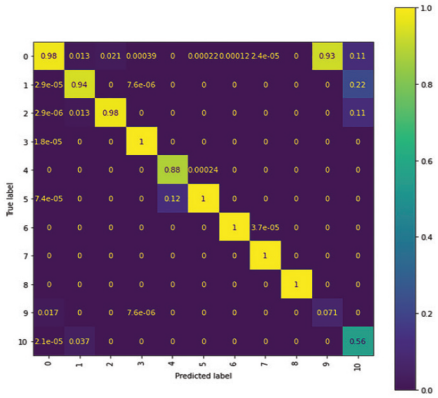
GradientBoosting



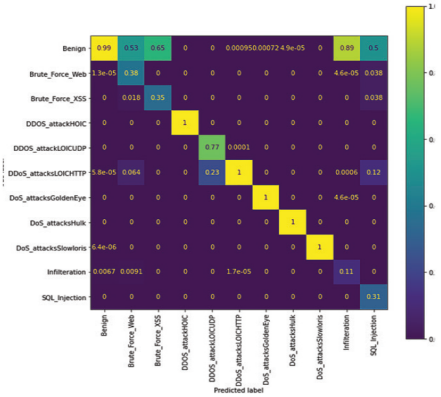
KNeighbors



DecisionTree

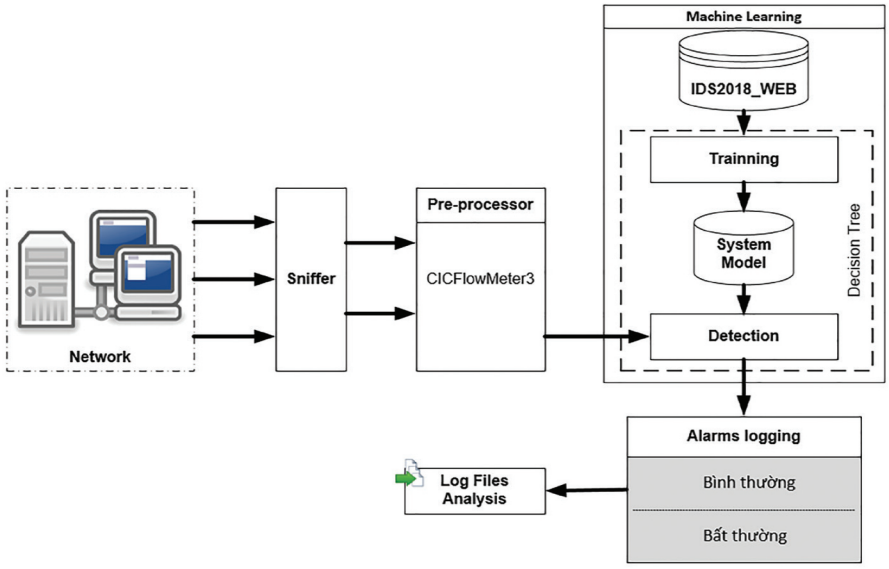


RandomForest



ExtraTrees

Hình 5. Ma trận nhầm lẫn của các thuật toán



Hình 6. Mô hình tổng thể phát hiện tấn công ứng dụng website trên bộ dữ liệu IDS-WEB2018

KẾT QUẢ THỰC NGHIỆM VÀ THẢO LUẬN

Qua quá trình thực nghiệm các mô hình Navie Bayes, Gradient Boosting, k-NN, DT, Cây mở rộng, Rừng ngẫu nhiên trên bộ dữ liệu IDS2021-WEB. Có thể nhận thấy, mô hình DT có hiệu năng tốt nhất qua các chỉ số huấn luyện và kiểm tra trong phân loại nhị phân và đa lớp. Có thể sử dụng các phương pháp lựa chọn các thuộc tính tốt để có thể áp dụng mô hình này cho hệ thống phát hiện dựa trên thời gian thực bởi tính ổn định và thời gian phát hiện nhanh.

Đối với việc phân loại đa lớp về bản chất giống với phân loại nhị phân về mục đích cảnh báo về các cuộc tấn công. Tuy nhiên nó còn có ý nghĩa khác đó là xác định cụ thể loại tấn công, từ đó trợ giúp hệ thống ngăn chặn tấn công đưa ra các giải pháp cụ thể. Một số trường hợp lưu lượng mạng của các lớp có số lượng mẫu nhỏ bị phân loại nhầm sang lớp có số lượng mẫu lớn hơn. Nhưng trường hợp lưu lượng của hình thức tấn công xâm nhập mặc dù có số lượng mẫu tương đối lớn nhưng vẫn bị phân loại nhầm với lưu lượng thông thường, do có thể trong quá trình huấn luyện các thuộc tính của 2 lớp này hầu như không có sự khác biệt, dẫn đến quá trình huấn luyện không được chính xác.

Để xây dựng giải pháp áp dụng phát hiện tấn công ứng dụng website dựa trên học máy, áp dụng mô hình DT làm mô hình chính để tiến hành phân loại trên tập dữ liệu IDS2021-WEB (trích xuất dựa trên CICFlowMeter3) trên cả các phương pháp phát hiện tấn công và phát hiện loại hình tấn công. Các chỉ số đánh giá thể hiện rõ, DT có chỉ số đánh giá ở mức độ rất cao, tỉ lệ bỏ sót gói tin tấn công nhỏ và có thời gian phân loại là nhanh nhất so với các thuật toán học máy phổ biến khác. Mô hình xử lý phát hiện tấn công ứng dụng website có thể được xây dựng như Hình 6.

KẾT LUẬN

Việc cải tiến các hệ thống phát hiện tấn công mạng là một trong những vấn đề bảo mật rất đáng quan tâm, IDS hoạt động bằng cách theo dõi hoạt động của hệ thống thông qua việc kiểm tra các lỗ hổng bảo mật, tính toàn vẹn của các tệp tin và tiến hành phân tích các mẫu dựa trên các cuộc tấn công đã biết, nó cũng tự động theo dõi lưu lượng mạng để tìm kiếm các mối đe dọa mới nhất có thể dẫn đến một cuộc tấn công trong tương lai.

Học máy đang dần được ứng dụng nhiều trong các hệ thống phát hiện xâm nhập. Nghiên cứu đã xây dựng giải pháp áp dụng học máy trong phát hiện tấn công ứng dụng website, dựa trên bộ dữ liệu IDS2021-WEB, mô phỏng hệ thống phát hiện và phân loại tấn công dựa trên học máy.❖

TÀI LIỆU THAM KHẢO

1. "IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB". https://www.unb.ca/cic/datasets/ids-2018.html (truy cập tháng 9 17, 2020).
2. "Correlation coefficient", Wikipedia. tháng 7 10, 2021. Truy cập: tháng 8 05, 2021. [Online]. Available at: https://en.wikipedia.org/w/index.php?title=Correlation_coefficient&oldid=1032873140.
3. "t-distributed stochastic neighbor embedding", Wikipedia. tháng 7 13, 2021. Truy cập: tháng 8 05, 2021. [Online]. Available at: https://en.wikipedia.org/w/index.php?title=T-distributed_stochastic_neighbor_embedding&oldid=1033326648.