Tạp chí Khoa học Trường Đại học Cần Thơ
Phần A: Khoa học Tự nhiên, Công nghệ và Môi trường
website: ctujsvn.ctu.edu.vn

DOI:10.22144/ctu.jvn.2022.035

HỆ THỐNG PHÁT HIỆN XÂM NHẬP HAI TẦNG CHO CÁC MẠNG IOT SỬ DỤNG MÁY HỌC

Thái Minh Tuấn[1*], Phạm Hoàng Hảo[2] và Trần Thanh Nam[3]

[1]*Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Cần Thơ*

[2]*Sinh viên ngành Công nghệ Thông tin - Chương trình chất lượng cao - Khóa 42*

[3]*Học viên thạc sĩ ngành Hệ thống thông tin - Khóa 25*

**Người chịu trách nhiệm về bài viết: Thái Minh Tuấn (email: minhtuan@ctu.edu.vn)*

Thông tin chung:
Ngày nhận bài: 17/09/2021
Ngày nhận bài sửa: 16/11/2021
Ngày duyệt đăng: 22/04/2022

Title:
A two-tier intrusion detection system for IoT networks using machine learning

Từ khóa:
An ninh mạng, hệ thống phát hiện xâm nhập, IoT, máy học

Keywords:
Cyber security, IDS, IoT, machine learning

ABSTRACT

Due to the increasing popularity and lack of security standards, Internet of Things (IoT) devices have become the targets of malicious activities such as intrusions and DoS attacks. With the aim of providing a solution for securing such devices, this paper introduces a two-tier intrusion detection system that applies machine learning models. The first tier of the proposed solution is a lightweight binary model, implemented in a gateway of an IoT network to detect intrusions and attacks in real-time. The second tier is a complicated multi-class classification model, located on a remote cloud server, to classify malicious activities and detect intrusions and attacks which occur on multi-networks. The experimental results display that the proposed solution can detect malicious activities using modified parameters more efficiently than Snort, which is a traditional signature-based IDS.

TÓM TẮT

Do sự phổ biến ngày càng tăng và thiếu các tiêu chuẩn bảo mật, các thiết bị Internet of Things (IoT) đã trở thành mục tiêu của các hoạt động độc hại như xâm nhập mạng và tấn công DoS. Với mục đích cung cấp một giải pháp an ninh cho các thiết bị IoT, một hệ thống phát hiện xâm nhập hai tầng áp dụng các mô hình máy học được giới thiệu trong bài viết này. Tầng thứ nhất của giải pháp là một mô hình phân loại nhị phân gọn nhẹ, được cài đặt trên gateway của các nhánh mạng IoT để phát hiện các hành vi độc hại trong thời gian thực. Tầng thứ hai là một mô hình phân loại đa lớp, được triển khai trên máy chủ đám mây để xác định loại cụ thể các hoạt động độc hại xảy ra trên nhiều nhánh mạng cùng lúc. Kết quả thực nghiệm cho thấy rằng giải pháp được đề xuất hoạt động hiệu quả, có thể phát hiện các hành vi tấn công sử dụng các tham số tùy biến hiệu quả hơn so với công cụ IDS truyền thống Snort.

1. GIỚI THIỆU

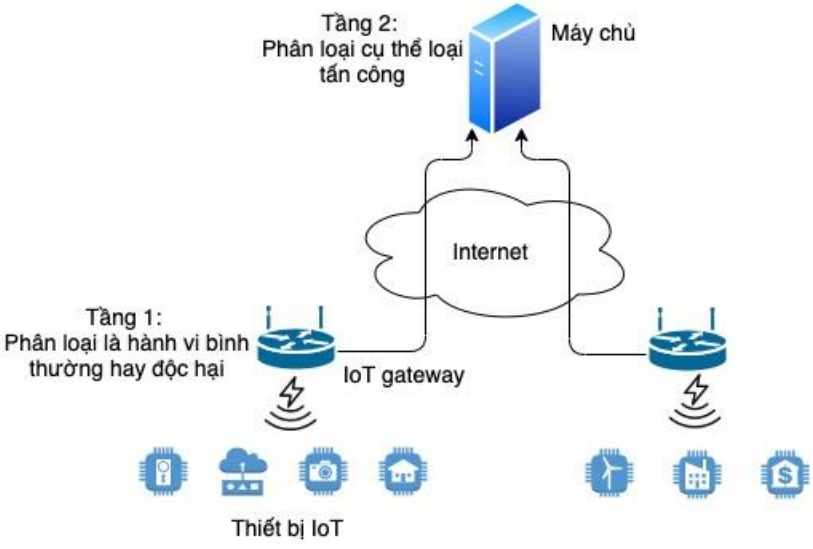
Sự phổ biến ngày càng tăng theo cấp số nhân của các thiết bị Internet of Things (IoT) đã khiến chúng trở thành mục tiêu thường xuyên của các cuộc tấn công và xâm nhập mạng (Vinayakumar et al., 2019). Do các đặc điểm về thiết kế nên các hệ thống IoT thường dễ bị tổn thương ngay cả đối với các cuộc tấn công qui mô nhỏ. Thiệt hại về kinh tế do các cuộc tấn công và xâm nhập mang lại là rất lớn. Vì số lượng, đặc điểm và tính không đồng nhất của các

thiết bị IoT nên việc đảm bảo an ninh cho chúng bằng các phương pháp truyền thống là không phù hợp. Thật vậy, các thiết bị này có thời gian hỗ trợ nâng cấp (nếu có) từ các nhà sản xuất là rất ngắn. Ngoài ra, chúng cũng không khả thi để cài đặt các giải pháp an ninh đầu cuối (endpoint security solution), ví dụ như tường lửa hay chương trình chống mã độc, lên chính các thiết bị này do sự hạn chế cả về phần cứng và phần mềm của chúng. Những phân tích trên cho thấy việc đề xuất và xây dựng một giải pháp phù hợp và hiệu quả để phát hiện các hình thức tấn công và xâm nhập mạng cho các hệ thống IoT là rất cấp thiết.

Các giải pháp phát hiện xâm nhập ở mức mạng (network-based intrusion detection system - NIDS) truyền thống như Snort và Suricata, bằng cách phân tích các đặc điểm của dữ liệu mạng (traffic signature) đã tỏ ra khá hiệu quả trong việc phát hiện các hình thức tấn công và xâm nhập đã biết (Albin et al., 2012). Tuy nhiên, chúng lại tỏ ra kém hiệu quả trong nhận biết các hình thức tấn công mới hoặc được điều chỉnh từ các hình thức cũ (Mishra et al., 2019). Thời gian qua đã có một số nghiên cứu (Mafra et al., 2010; Vinayakumar et al., 2019) áp dụng các kỹ thuật máy học với mục đích nâng cao hiệu quả và khắc phục các nhược điểm của các giải pháp IDS truyền thống. Tuy nhiên, những giải pháp này yêu cầu phải được triển khai trên các thiết bị có yêu cầu về phần cứng và phần mềm cao, nên không phù hợp cho các môi trường IoT.

Nhằm giải quyết hạn chế của những giải pháp hiện tại, nghiên cứu này thiết kế và cài đặt một hệ thống phát hiện xâm nhập hai tầng, phù hợp và hoạt động hiệu quả cho các mạng thiết bị IoT. Như được mô tả trong Hình 1, tầng đầu tiên của hệ thống được cài đặt trên gateway của các nhánh mạng IoT, sẽ bắt và trích xuất các thông tin của dữ liệu mạng thô đến và đi tới nhánh mạng. Sau đó, các thông tin sẽ được chuyển đến một mô hình mạng nơ-ron nhân tạo *4* lớp đơn giản để phát hiện và cảnh báo các hành vi xâm nhập trong thời gian thực. Mô hình này được thiết kế gọn nhẹ và chi phí thấp để có thể triển khai trên các IoT gateway. Tầng thứ hai của hệ thống được triển khai trên các máy chủ đám mây ở xa, sử dụng các mô hình máy học phức tạp hơn như Cây quyết định (Decision tree), Gaussian naïve bayes và Rừng ngẫu nhiên (Random forest) để xác định thông tin cụ thể các loại hành vi tấn công mạng. Ngoài ra, tầng này cũng hoạt động như giải pháp dự phòng cho tầng đầu tiên và hướng tới phát hiện các hành vi độc hại dựa trên thông tin thu thập được trên nhiều mạng IoT cùng lúc.

Hệ thống đề xuất đã được cài đặt và triển khai kiểm thử trên môi trường thực nghiệm cục bộ. Kết quả thực nghiệm ban đầu cho thấy hệ thống đề xuất hoạt động hiệu quả và có tỉ lệ phát hiện các hành vi tấn công và xâm nhập sử dụng các tham số được tùy biến cao hơn so với Snort, một công cụ IDS truyền thống hoạt động dựa trên các nguy cơ đã biết.



Hình 1. Hệ thống phát hiện xâm nhập 2 tầng cho các mạng IoT

2. CÁC NGHIÊN CỨU LIÊN QUAN

Một trong những phương pháp phổ biến để đảm bảo an ninh mạng là sử dụng các hệ thống phát hiện xâm nhập IDS. Trong thực tế, các IDS thường được kết hợp với các công cụ giám sát, tường lửa, chương trình phát hiện mã độc,... để tạo thành một giải pháp

bảo mật hoàn chỉnh cho các hệ thống công nghệ thông tin. Các hệ thống IDS có thể triển khai theo hai mô hình chính bao gồm: Network-based (NIDS) được cài đặt trên một thiết bị mạng và sẽ giám sát toàn bộ một nhánh mạng; và Host-based (HIDS) được cài đặt trực tiếp trên một máy chủ trong vùng DMZ.

Các giải pháp IDS truyền thống như Snort và Suricata hoạt động dựa trên các dấu hiệu đặc biệt về các nguy cơ đã biết (Signature-based IDS), hoặc dựa trên so sánh lưu thông mạng hiện tại với baseline (thông số đo đạt chuẩn của hệ thống có thể chấp nhận được ngay tại thời điểm hiện tại) để tìm ra các dấu hiệu khác thường (Anomaly-based IDS) nhằm phát hiện các hoạt động xâm nhập trái phép vào hệ thống (Hall et al., 2005). Mặc dù khá hiệu quả trong việc phát hiện các hình tấn công đã biết và được triển khai trong thực tiễn; tuy nhiên, những hệ thống đề cập lại tỏ ra kém chính xác và có tỷ lệ báo động giả (false positive) cao trong việc phát hiện các hình thức tấn công mới hoặc được tùy biến từ các hình thức cũ (Mishra et al., 2019).

Đã có một số nghiên cứu áp dụng các tiếp cận máy học nhằm nâng cao độ chính xác và khắc phục các nhược điểm của các giải pháp IDS truyền thống. Vinayakumar et al. (2019) đã áp dụng một mạng nơ-ron sâu (DNN) để phát hiện các cuộc tấn công xâm nhập mạng không lường trước. Nghiên cứu của (Mafra et al., 2010) trình bày mô hình máy học vector hỗ trợ SVM phát hiện các hành vi bất thường, được áp dụng trong một hệ thống phát hiện xâm nhập có tên gọi là Octopus-IIDS. Kết quả thực nghiệm của nghiên cứu cho thấy hệ thống có tỉ lệ phát hiện cao và giảm tỷ lệ false positive. Nhìn chung, các nghiên cứu đề cập (Mafra et al., 2010; Mishra et al., 2019; Vinayakumar et al., 2019) đã nhận định việc áp dụng các kỹ thuật máy học có thể nâng cao hiệu quả của các hệ thống phát hiện xâm nhập mạng. Tuy nhiên, các nghiên cứu này chỉ đưa ra các giải pháp cho các hệ thống mạng máy tính và thiết bị di động truyền thống.

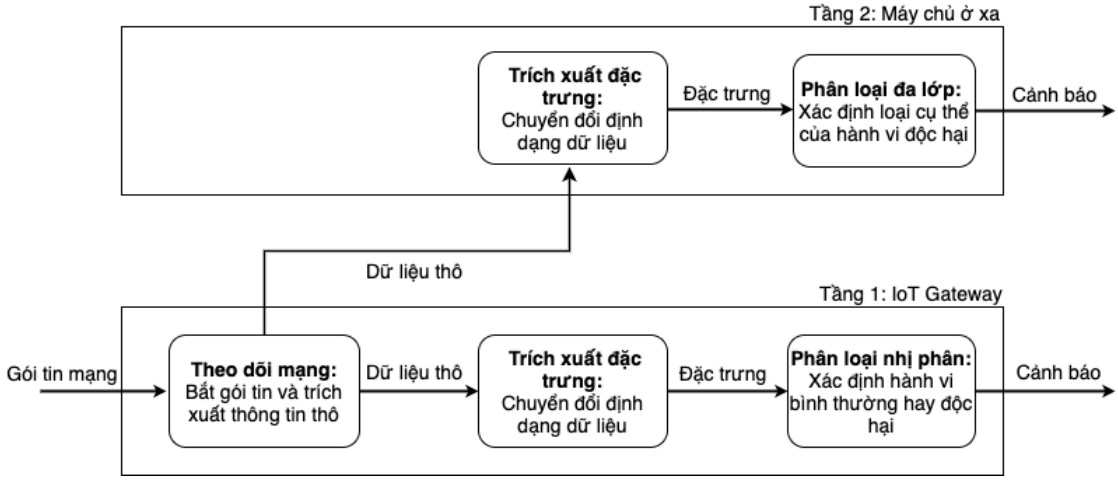
Trong những năm gần đây, sự phổ biến của các thiết bị IoT, đi kèm với những nguy cơ về tấn công và xâm nhập trên những thiết bị này đã có một số nghiên cứu được thực hiện nhằm đưa ra những giải pháp an ninh cho các môi trường IoT. Trong đó, Eskandari et al. (2020) đã giới thiệu một giải pháp IDS có thể triển khai trên IoT gateway và bảo vệ các thiết bị nối kết trực tiếp vào nó. Nhóm tác giả Nguyen et al. (2019) đã giới thiệu một giải pháp có tên là DioT, bao gồm hai tầng là Security gateway và Security service. Trong đó, Security gateway có trách nhiệm theo dõi các thiết bị trong nhánh mạng và phát hiện các hành vi bất thường trong nhánh mạng. Trong khi đó, Security service tích hợp mô hình phát hiện xâm nhập từ các Security gateway để đảm bảo an ninh toàn bộ hệ thống. Tương tự, giải pháp IoT Keeper (Hafeez et al., 2018) cũng bao gồm hai tầng: Keeper gateway sẽ lọc các dữ liệu mạng độc hại trên một nhánh mạng sử dụng các giải thuật máy học bán giám sát và Keeper service có vai trò hỗ trợ dự phòng cho các Keeper gateway. Qua khảo sát các nghiên cứu liên quan, có thể khẳng định phương pháp triển khai IDS trên IoT gateway kết hợp với một tầng dự phòng ở xa đang được xem là một giải pháp an ninh phù hợp cho các môi trường IoT.

3. THIẾT KẾ VÀ CÀI ĐẶT HỆ THỐNG

3.1. Tổng quan thiết kế hệ thống

Hình 2 mô tả thiết kế của hệ thống phát hiện xâm nhập mạng 2 tầng cho các mạng thiết bị IoT được đề xuất trong bài báo này. Tầng đầu tiên của hệ thống là một mô hình máy học đơn giản được cài đặt trên các gateway của các mạng IoT với mục đích phát hiện các hành vi xâm nhập trong thời gian thực. Như đã giới thiệu, tầng này được thiết kế gọn nhẹ và chi phí thấp để có thể triển khai trên các IoT gateway, vốn hạn chế về khả năng phần cứng và phần mềm. Tầng thứ hai là một mô hình máy học phức tạp chạy trên các máy chủ đám mây ở xa, được sử dụng để phân loại các hành vi tấn công và xâm nhập xảy ra trên nhiều mạng IoT cùng lúc. Trong phạm vi nghiên cứu này, đối với thiết bị IoT giao tiếp qua nối kết không dây, hệ thống đề xuất chỉ hỗ trợ phát hiện các hành vi độc hại thực hiện trên các nối kết chế độ cơ sở hạ tầng (Infrastructure mode).

Trong hệ thống đề xuất, mô-đun *Theo dõi mạng* được cài đặt trên các gateway sẽ bắt và trích xuất các thuộc tính của các gói tin mạng thô đến và đi tới nhánh mạng. Sau đó, mô-đun *Trích xuất đặc trưng* ở Tầng 1 sẽ chuyển đổi các dữ liệu thô sang các đặc trưng được định dạng phù hợp và gửi đến mô-đun *Phân loại nhị phân* để phát hiện có hành vi độc hại hay không. Tương tự, mô-đun *Trích xuất đặc trưng* ở Tầng 2 cũng tạo các đặc trưng từ các thông tin được gửi từ mô-đun *Theo dõi mạng* ở Tầng 1. Các đặc trưng này sẽ được gửi đến mô-đun *Phân loại đa lớp* để xác định loại cụ thể hành vi độc hại. Khi các hành vi tấn công và xâm nhập mạng được phát hiện, hệ thống sẽ đưa ra các cảnh báo người dùng về các nguy cơ nếu có.



Hình 2. Quy trình hoạt động của Hệ thống phát hiện xâm nhập 2 tầng

3.2. Huấn luyện các mô hình phát hiện xâm nhập

3.2.1. Tiền xử lý tập dữ liệu huấn luyện

Nghiên cứu này sử dụng bộ dữ liệu UNSW-NB15 (Moustafa et al., 2017) được tạo bởi phòng thí nghiệm của Trung tâm An ninh mạng Úc (ACCS) để huấn luyện cho các mô hình phân loại nhị phân và đa lớp của hệ thống đề xuất. Đây là bộ dữ liệu nổi tiếng được sử dụng trong nhiều nghiên cứu về tấn công và xâm nhập mạng. UNSW-NB15 bao gồm các thuộc tính của dữ liệu mạng bình thường và của các hành vi độc hại. Có 9 kiểu tấn công và xâm nhập được hỗ trợ trong tập dữ liệu bao gồm: *Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode* và *Worms*. Bộ dữ liệu có tổng cộng *2,539,740* dòng dữ liệu, mỗi dòng chứa *49* thuộc tính của một nối kết mạng. Nghiên cứu này lựa chọn *20* thuộc tính phổ biến và phù hợp nhất của tập dữ liệu để huấn luyện cho các mô hình phân loại như mô tả trong Bảng 1. Kết quả nghiên cứu của các tác giả Zoghi et al. (2021) đã chứng minh việc không sử dụng các thuộc tính còn lại trong tập dữ liệu UNSW-NB15 không ảnh hưởng đến độ chính xác của các mô hình phân loại. Trong khi đó, điều này giúp giảm yêu cầu về năng lực xử lý của IoT gateway trong việc trích xuất các thuộc tính của dữ liệu mạng. Lưu ý, *2* thuộc tính Giao thức nối kết (*proto*) và Trạng thái giao thức (*state*) sẽ được chuyển từ kiểu chuỗi sang số trước khi sử dụng huấn luyện cho các mô hình. Trong đó, thuộc tính *state* sẽ được mã hóa theo thứ tự trong danh sách các trạng thái. Trong khi đó, thuộc tính *proto* được chuyển sang giá trị số theo Assigned Numbers Internet Protocol.

Bảng 1. Các thuộc tính của nối kết mạng được sử dụng cho các mô hình phân loại

#	Tên	Giải thích	Kiểu	#	Tên	Giải thích	Kiểu
1.	sport	Số cổng nguồn	integer	11.	dttl	Giá trị time-to-live từ đích đến nguồn	integer
2.	dport	Số cổng đích	integer	12.	sload	Số bits nguồn mỗi giây	integer
3.	dur	Thời gian của nối kết	float	13.	dload	Số bits đích mỗi giây	integer
4.	proto	Giao thức nối kết	nominal	14.	sloss	Số gói tin từ nguồn được truyền lại hoặc loại bỏ	integer
5.	state	Trạng thái giao thức	nominal	15.	dloss	Số gói tin từ đích được truyền lại hoặc loại bỏ	integer
6.	spkts	Số gói tin từ nguồn đến đích	integer	16.	synack	Thời gian giữa SYN và SYN_ACK	float
7.	dpkts	Số gói tin từ đích đến nguồn	integer	17.	ackdat	Thời gian giữa SYN_ACK và ACK	float
8.	sbytes	Số bytes từ nguồn đến đích	integer	18.	smeansz	Trung bình kích thước gói tin từ nguồn	integer
9.	dbytes	Số bytes từ đích đến nguồn	integer	19.	dmeansz	Trung bình kích thước gói tin từ đích	integer
10.	sttl	Giá trị time-to-live từ nguồn đến đích	integer	20.	tcprtt	Tổng thời gian thiết lập nối kết TCP	float

Phương pháp Hold-out (Yada & Shukla, 2016) được sử dụng trong nghiên cứu chia tập dữ liệu thành 80% cho tập huấn luyện (train) và 20% cho tập kiểm thử (test). Số lượng dòng dữ liệu dùng huấn luyện và kiểm thử được mô tả trong Bảng 2.

Bảng 2. Số lượng dữ liệu dùng huấn luyện và kiểm thử các mô hình

Nhãn dữ liệu	Số lượng dữ liệu huấn luyện	Số lượng dữ liệu kiểm tra
Normal	1.774.848	443.604
Analysis	2.138	539
Backdoors	1.850	479
DoS	13.165	3.188
Exploits	35.618	8.907
Fuzzers	19.445	4.801
Generic	172.143	43.337
Reconnaissance	11.221	2.766
Shellcode	1.232	279
Worms	128	46

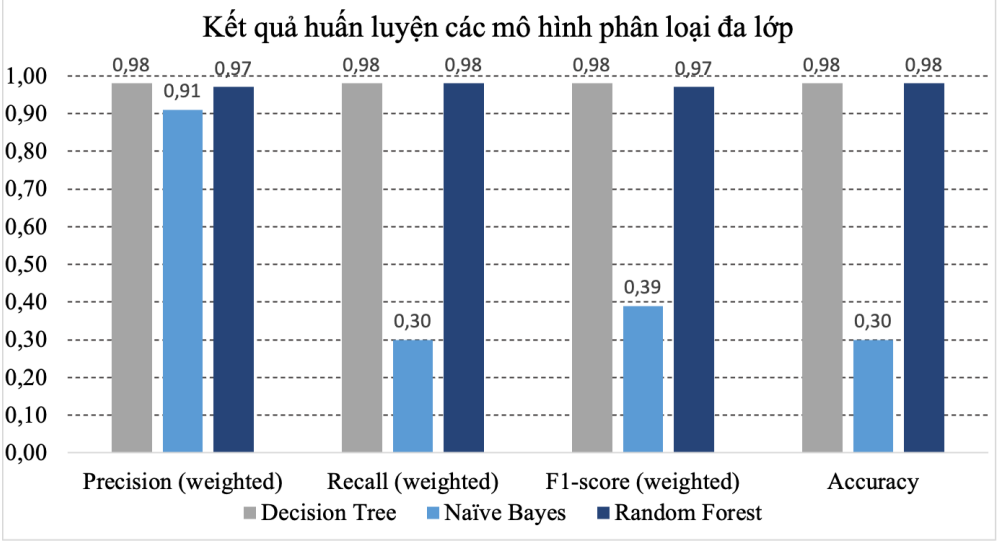
3.2.2. Huấn luyện mô hình Phân loại nhị phân cho Tầng 1

Do yêu cầu Tầng 1 của hệ thống phát hiện các hành vi xâm nhập và tấn công mạng có thể triển khai được trên các IoT gateway, vốn có khả năng phần cứng và phần mềm hạn chế. Vì vậy, mô hình mạng nơ-ron nhân tạo *4* lớp đơn giản được sử dụng để có thể phát hiện và cảnh báo các hành vi xâm nhập trong thời gian thực. Lớp đầu vào của mô hình gồm *20* nút và hai lớp ẩn, mỗi lớp ẩn gồm *32* nút sử dụng hàm kích hoạt ReLU. Lớp đầu ra gồm *1* nút sử dụng hàm kích hoạt Sigmoid để xác định dữ liệu mạng là độc hại. Mô hình này đã được thực nghiệm chứng tỏ có hiệu quả cao trong việc phát hiện các hành vi mạng độc hại (Vinayakumar et al., 2019), trong khi không yêu cầu quá nhiều năng lực xử lý của IoT gateway như các mô hình khác.

Mô hình được huấn luyện sử dụng thư viện máy học Keras (Chollet et al., 2015). Quá trình huấn luyện sử dụng trình tối ưu hóa momentum Adam với tốc độ học (learning_rate) là *0,01*. Hàm lỗi (loss function) là *binary_crossentropy*. Số lần quá trình huấn luyện học qua tất cả các dữ liệu trong tập huấn luyện (epochs) là *20*, với số lượng mẫu huấn luyện sẽ được gửi đến mô hình cùng một lúc (batch_size) là *128*. Kết quả huấn luyện cho thấy độ chính xác (accuary) của mô hình là khoảng *0,91*.

3.2.3. Huấn luyện mô hình Phân loại đa lớp cho Tầng 2

Để dự phòng cho Tầng 1 và hướng tới phát hiện các hành vi tấn công và xâm nhập mạng dựa trên thông tin thu thập trên nhiều mạng IoT cùng lúc, Tầng 2 của giải pháp đề xuất được cài đặt ở máy chủ ở xa (có thể triển khai trên môi trường đám mây). Do không có những hạn chế về phần cứng và phần mềm cùng với chi phí triển khai, tầng này có thể sử dụng các mô hình máy học phức tạp hơn Tầng 1, với mục đích xác định loại cụ thể các hành vi độc hại (*Fuzzers*, Analysis, *Backdoors*, *DoS*, *Exploits*, *Generic*, *Reconnaissance*, *Shellcode* và *Worms*) và nâng cao độ chính xác của hệ thống.



Hình 3. Kết quả huấn luyện các mô hình phân loại đa lớp

Có 3 mô hình máy học được sử dụng cho Phân loại đa lớp, đó là Cây quyết định (Decision tree), Gaussian naïve bayes và Rừng ngẫu nhiên (Random forest). Các mô hình máy học này đã được áp dụng và chứng tỏ hiệu quả trong các nghiên cứu đương đại về phát hiện tấn công và xâm nhập mạng (Mishra et al., 2019). Trong nghiên cứu này, cả 3 mô hình này đều được huấn luyện và kiểm thử trong để quan sát độ chính xác thu được, từ đó có cơ sở để lựa chọn mô hình phù hợp cho hệ thống. Thư viện máy học Scikit-Learn (Pedregosa et al., 2011) được sử dụng để huấn luyện các mô hình với những tham số huấn luyện được hỗ trợ mặc nhiên bởi thư viện. Hình 3 mô tả kết quả huấn luyện của 3 mô hình. Kết quả cho thấy hai mô hình Cây quyết định và Rừng ngẫu nhiên có độ chính xác khá tốt và tương đương nhau (*0,98*). Trong khi đó, Gaussian naïve bayes cho kết quả thấp (*0,30*). Điều này có thể lý giải là do mô hình này không hoạt động tốt trên các tập dữ liệu mất cân bằng như UNSWNB-15.

3.3. Cài đặt hệ thống

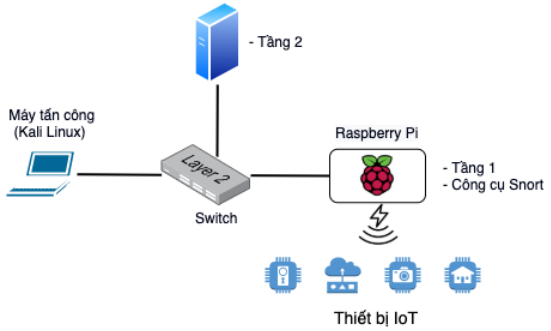
Mô-đun *Theo dõi mạng* và *Trích xuất đặc trưng* của hệ thống đề xuất được cài đặt sử dụng công cụ Argus (openargus, n.d.). Đây là một công cụ giám sát mạng cho phép người dùng theo dõi các gói tin đến và đi tới một nối kết mạng được chỉ định. Trong đó, công cụ *Argus-sensor* được sử dụng bởi mô-đun *Theo dõi mạng* để thu thập, phân loại gói tin mạng và xử lý thông tin thô. Mô-đun *Trích xuất đặc trưng* sử dụng công cụ *Argus-client* để giao tiếp *Argus-sensor* trên mô-đun *Theo dõi mạng* để nhận dữ liệu thô. Trong đó, thành phần *Argus-Ra* của công cụ được sử dụng để rút trích và tạo đặc trưng đầu vào cho các mô hình phân loại.

Mô hình *Phân loại đa lớp* của hệ thống được cài đặt sử dụng thư viện Scikit-Learn dựa trên nền tảng máy học TensorFlow (Abadi, n.d.). Trong khi đó, mô hình *Phân loại nhị phân* được cài đặt sử dụng thư viện Keras dựa trên nền tảng TensorFlow Lite (TensorFlow Lite, n.d.), với mục đích tạo mô hình có yêu cầu thấp, phù hợp cho các IoT gateway.

4. THỰC NGHIỆM

4.1. Môi trường thực nghiệm

Để đánh giá hiệu quả của hệ thống phát hiện xâm nhập 2 tầng được đề xuất trong nghiên cứu này, một môi trường thực nghiệm đơn giản được bao gồm 2 máy tính và 1 thiết bị Raspberry Pi mẫu 3B+ được xây dựng như mô tả trong Hình 4. Trong đó, Tầng 1 của hệ thống được cài đặt trên thiết bị Raspberry, đóng vai trò gateway của một nhánh mạng IoT. Tầng 2 của hệ thống được triển khai trên 1 máy tính sử dụng hệ điều hành Ubuntu, trong khi máy còn lại sử dụng hệ điều hành Kali Linux và đóng vai trò máy tấn công. Thiết bị Raspberry Pi Zero được sử dụng mô phỏng các thiết bị IoT trong thực nghiệm.



Hình 4. Môi trường thực nghiệm

Trong nghiên cứu này, độ chính xác của hệ thống đề xuất được so sánh với công cụ Snort, là một IDS mã nguồn mở đang được sử dụng khá phổ biến. Trong thực tế, công cụ Snort được sử dụng như một chương trình kiểm tra gói tin để giám sát hệ thống trong thời gian thực. Người quản trị mạng có thể sử dụng công cụ để theo dõi tất cả các gói tin đến và tìm ra những gói tin nguy hiểm cho hệ thống. Snort sử dụng các quy tắc (rules) tĩnh được thiết lập trước để phát hiện dấu hiệu bất thường trên mạng. Độ chính xác của Snort phụ thuộc vào các quy tắc được sử dụng, đòi hỏi kinh nghiệm người quản trị mạng. Các thực nghiệm trong nghiên cứu này sử dụng các quy tắc được hỗ trợ bởi cộng đồng người dùng Snort (Community Rules).

Các công cụ Hping3, Nmap, Hydra và Metasploit trên Kali Linux được sử dụng để tạo ra các cuộc tấn công nhằm kiểm tra đánh giá độ chính xác của cả hai hệ thống. Các hình thức tấn công được thực hiện bao gồm Từ chối dịch vụ (DoS), tấn công thăm dò (Reconnaissance/Analysis), tấn công lỗ hổng bảo mật (Exploits), tấn công khai thác (Backdoor). Các cuộc tấn công được thực hiện với 2 hình thức: Sử dụng các tham số được hỗ trợ mặc nhiên bởi các công cụ tấn công và sử dụng các tham số tấn công được tùy biến. Trong đó, các tham số tấn công được điều chỉnh dựa vào phân tích các quy tắc trong Community Rules của công cụ Snort. Mỗi hình thức tấn công được thực hiện *10* lần trong các thực nghiệm.

4.2. Kết quả thực nghiệm

Bảng 3 và 4 so sánh tỷ lệ phát hiện các hành vi tấn công của Tầng 1-Phân loại nhị phân và Tầng 2-Phân loại đa lớp của hệ thống được đề xuất trong bài báo này và công cụ Snort. Như mô tả ở Bảng 3, kết

quả thực nghiệm cho thấy Tầng 1 có tỉ lệ phát hiện trung bình khá cao (*87,5%*) với ít nhất *80%* (*8/10*) ở cả 4 hình thức tấn công sử dụng tham số mặc nhiên. Trong khi đó, Tầng 2 có tỉ lệ phát hiện gần tương đương với công cụ Snort trong 2 hình thức từ chối dịch vụ và lỗ hổng bảo mật, và vượt trội về tỉ lệ phát hiện tấn công khai thác (*8/10* so với *0/10*). Tuy nhiên, nó lại kém hơn về ở hình thức tấn công thăm dò (*2/10* so với *9/10*). Ở kết quả thực nghiệm, các hình thức tấn công sử dụng tham số tuỳ biến (Bảng 4), cả 2 tầng của hệ thống vẫn có tỉ lệ phát hiện gần như tương đương với tấn công sử dụng tham số mặc nhiên. Trong khi đó, công cụ Snort có sự giảm tỉ lệ phát hiện lớn (từ *72,5%* xuống *45,0%*), đặc biệt là ở hai loại tấn công từ chối dịch vụ và lỗ hổng bảo mật (từ *10/10* xuống lần lượt *6/10* và *3/10*). Kết quả thực nghiệm cho thấy giải pháp đề xuất hoạt động hiệu quả, và cho kết quả khả quan hơn công cụ Snort trong việc phát hiện các cuộc tấn công được tùy biến tham số.

Bảng 3. Tỉ lệ phát hiện các hình thức tấn công sử dụng tham số mặc nhiên

Hình thức tấn công	Tầng 1: Phân loại nhị phân	Tầng 2: Phân loại đa lớp	Công cụ Snort
Từ chối dịch vụ (DoS)	9/10	8/10	10/10
Thăm dò (Reconnaissance/Analysis)	8/10	2/10	9/10
Lỗ hổng bảo mật (Exploits)	10/10	10/10	10/10
Khai thác (Backdoor)	8/10	8/10	0/10
Tổng cộng	35/40 (87,5%)	28/40 (70%)	29/40 (72,5%)

Bảng 4. Tỉ lệ phát hiện các hình thức tấn công sử dụng tham số tùy biến

Hình thức tấn công	Tầng 1: Phân loại nhị phân	Tầng 2: Phân loại đa lớp	Công cụ Snort
Từ chối dịch vụ (DoS)	9/10	6/10	6/10
Thăm dò (Reconnaissance/Analysis)	8/10	2/10	9/10
Lỗ hổng bảo mật (Exploits)	9/10	9/10	3/10
Khai thác (Backdoor)	8/10	8/10	0/10
Tổng cộng	34/40 (85,0%)	25/40 (62,5%)	18/40 (45,0%)

5. KẾT LUẬN

Kết quả thực nghiệm của nghiên cứu cho thấy rằng giải pháp hệ thống phát hiện xâm nhập hai tầng đề xuất hoạt động thành công và có thể phát hiện các hình thức tấn công sử dụng các tham số tuỳ biến khả quan hơn so với công cụ truyền thống Snort. Các thực nghiệm cũng chứng minh rằng hệ thống được đề xuất có thể hoạt động hiệu quả, với tác động tối thiểu đến trải nghiệm người dùng. Hơn nữa, hoạt động của hệ thống không yêu cầu phần cứng chuyên dụng hoặc thực hiện các sửa đổi trên các thiết bị IoT.

Hướng phát triển của nghiên cứu này có thể tập trung vào hai vấn đề chính. Thứ nhất là cần thu thập thêm dữ liệu huấn luyện và xử lý vấn đề mất cân bằng của các tập dữ liệu hiện có, để có thể phát triển các mô hình phát hiện xâm nhập có độ chính xác cao và hiệu quả hơn; đặc biệt là cần tập trung vào giải pháp phát hiện hình thức tấn công từ chối dịch vụ sử dụng các mạng máy tính ma (Botnet), vốn phổ biến trên các mạng IoT hiện nay. Thứ hai là triển khai và tiến hành thực nghiệm diện rộng trên các môi trường IoT thực tế và sử dụng các hình thức tấn công phức tạp hơn để có kết quả thực nghiệm đa dạng hơn, làm nền tảng để tiếp tục cải tiến các giải pháp hiện có.

LỜI CẢM TẠ

Nghiên cứu này được tài trợ bởi kinh phí từ đề tài nghiên cứu khoa học công nghệ cấp cơ sở T2020-13, được cấp bởi Trường Đại học Cần Thơ.

TÀI LIỆU THAM KHẢO

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: *A system for large-scale machine learning*. Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, 265–283.

Albin, E., & Rowe, N. C. (2012). *A Realistic Experimental Comparison of the Suricata and Snort Intrusion-Detection Systems*. 2012 26th International Conference on Advanced Information Networking and Applications

Workshops, 122–127. https://doi.org/10.1109/WAINA.2012.29

Chollet, F. (2015). *keras, GitHub*. https://github.com/keras-team/keras

Eskandari, M., Janjua, Z. H., Vecchio, M., & Antonelli, F. (2020). Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices. *IEEE Internet of Things Journal*, *7*(8), 6882–6897. https://doi.org/10.1109/JIOT.2020.2970501

Hafeez, I., Antikainen, M., Ding, A. Y., & Tarkoma, S. (2018). *IoT-KEEPER: Securing IoT Communications in Edge Networks*. ArXiv:1810.08415 [Cs]. http://arxiv.org/abs/1810.08415

Hall, J., Barbeau, M., & Kranakis, E. (2005). Anomaly-based intrusion detection using mobility profiles of public transportation users. WiMob'2005), *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, 2, 17-24. https://doi.org/10.1109/WIMOB.2005.1512845

Mafra, P. M., Moll, V., da Silva Fraga, J., & Altair Olivo Santin. (2010). Octopus-IIDS: An anomaly based intelligent intrusion detection system. *The IEEE Symposium on Computers and Communications*, 405–410. https://doi.org/10.1109/ISCC.2010.5546735

Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2019). A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection. *IEEE Communications Surveys Tutorials*, *21*(1), 686–728. https://doi.org/10.1109/COMST.2018.2847722

Moustafa, N., Creech, G., & Slay, J. (2017). Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models. In I. Palomares Carrascosa, H. K. Kalutarage, & Y. *Huang (Eds.), Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications* (pp. 127–156). Springer International Publishing. https://doi.org/10.1007/978-3-319-59439-2_5

Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., & Sadeghi, A.-R. (2019). *DIoT: A Federated Self-Learning Anomaly Detection System for IoT*. ArXiv:1804.07474 [Cs]. http://arxiv.org/abs/1804.07474

Openargus. (n.d.). Retrieved September 11, 2021, from https://openargus.org/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830.

TensorFlow Lite. (n.d.). TensorFlow. Retrieved September 11, 2021, from https://www.tensorflow.org/lite

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access*, 7, 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

Yadav, S., & Shukla, S. (2016). Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. 2016 *IEEE 6th International Conference on Advanced Computing (IACC)*, 78–83. https://doi.org/10.1109/IACC.2016.25

Zoghi, Z., & Serpen, G. (2021). *UNSW-NB15 Computer Security Dataset: Analysis through Visualization*. ArXiv:2101.05067 [Cs]. http://arxiv.org/abs/2101.05067