DOI: 10.15625/vap.2022.0205

TRÍCH CHỌN ĐẶC TRƯNG CHO MẠNG NƠRON TÍCH CHẬP TRONG BÀI TOÁN NHẬN DIỆN TẤN CÔNG MẠNG

Nguyễn Năng Hùng Vân[1], Đỗ Phúc Hảo[2], Phạm Minh Tuấn[3]
[1,3]Trường Đại học Bách khoa, Đại học Đà Nẵng
[2]Trường Đại học Kiến trúc Đà Nẵng
nguyenvan@dut.udn.vn, haodp@dau.edu.vn, pmtuan@dut.udn.vn

TÓM TẮT: *Ngày nay, với sự phát triển không ngừng của thiết bị thông minh và mạng máy tính đã dẫn đến những cuộc tấn công mạng ngày càng trở nên phổ biến và tinh vi hơn. Các tin tặc đã sử dụng nhiều kỹ thuật tấn công mạng khác nhau để truy cập trái phép vào hệ thống máy tính và thiết bị có kết nối mạng (IoT) để đánh cắp thông tin hoặc mã hóa thông tin quan trọng và đòi tiền chuộc. Do đó, vấn đề an ninh mạng đã trở nên cấp thiết và tác động rất lớn tới hiệu quả hoạt động của mạng máy tính và thiết bị IoT. Một trong những biện pháp phổ biến nhất hiện nay để bảo đảm an toàn cho các hệ thống mạng là hệ thống phát hiện xâm nhập trái phép (Intrustion Detector System). Tuy nhiên, các biện pháp này tỏ ra không hiệu quả, độ tin cậy không cao và không có khả năng tự cập nhật để phát hiện các xâm nhập mới hơn một cách linh hoạt. Một hướng tiếp cận mới ngày càng thể hiện tính ưu việt và khắc phục được các hạn chế trên là ứng dụng kỹ thuật học máy (machine learning) để phát hiện tấn công mạng. Bài báo này, đề xuất một phương pháp mới là trích chọn đặc trưng để phân loại tấn công mạng máy tính dựa vào đặc trưng của gói tin bằng cách sử dụng mạng học sâu. Ngoài ra, bài báo chỉ ra những đặc trưng quan trọng trong bộ dữ liệu Bot-IoT có thể thiết lập dưới dạng ma trận để mạng nơron tích chập phân loại và nâng cao độ chính xác phát hiện cuộc tấn công mạng. Trước tiên, nghiên cứu đề xuất phương pháp tiền xử lý dữ liệu để tối ưu hóa dữ liệu. Bước tiếp theo, sử dụng phương pháp trích chọn đặc trưng để tạo các véc-tơ và ma trận đặc trưng. Cuối cùng, nghiên cứu sử dụng CNN để phân loại dựa vào các ma trận đặc trưng. Mô hình đề xuất được thực nghiệm trên bộ dữ liệu Bot-IoT và kết quả tốt nhất có độ chính xác là 99 %.*

Từ khoá: *Internet of Things, feature selection, Botnet dataset, Attack traffic, Convolutional Neural Network.*

I. GIỚI THIỆU

Ngày nay, sự bùng nổ công nghệ cùng với các thiết bị thông minh và internet vạn vật (IoT) đã làm gia tăng các cuộc tấn công mạng có chủ đích để phát tán mã độc, ăn cắp dữ liệu của các hệ thống thông tin quan trọng. Tính đến nữa đầu năm 2021 thế giới chứng kiến khoảng 1,5 tỷ cuộc tấn công mạng vào các thiết bị thông minh và IoT, những kẻ tấn công đã tìm cách ăn cắp dữ liệu, khai thác tiền điện tử hoặc xây dựng các botnet [1]. Theo báo cáo của Symantec thì cứ 2 phút sẽ có một thiết bị IoT bị tấn công [2]. Còn theo báo cáo của Kaspersky trong năm 2018 đã thu thập 121.588 mẫu phần mềm độc hại đã tấn công các thiết bị IoT [3] điều này cho thấy các cuộc tấn công mạng trong năm 2018 nhiều hơn khoảng bốn lần so với năm 2017 [4]. Tại Việt Nam, theo báo cáo của Trung tâm Giám sát an toàn không gian mạng quốc gia tính đến tháng 7/2022 đã ghi nhận 652.221 địa chỉ IP của Việt Nam nằm trong mạng botnet [5].

Nhìn chung các cuộc tấn công chủ yếu nhắm vào các lỗ hổng bảo mật của hệ thống chẳng hạn như mã hóa dữ liệu và bảo mật mật khẩu. Kết quả là, các cuộc tấn công mạng đã nổi lên như một trở ngại quan trọng đối với việc triển khai các dịch vụ IoT. Một số hình thức tấn công phổ biến vào thiết bị thông minh và IoT như tấn công Sybil [6], Man-In-The-Middle tấn công [7], tấn công định tuyến [8], từ chối dịch vụ (DoS) và các cuộc tấn công từ chối dịch vụ phân tán (DDoS) [9], Elevation của các cuộc tấn công đặc quyền (EoP) [10], các cuộc tấn công bằng phần mềm độc hại [11]. Mỗi hình thức tấn công sẽ có các cách triển khai và tác động đến hệ thống máy tính và IoT khác nhau. Hiện nay, có nhiều giải pháp đã được đưa ra để hạn chế tác động của các cuộc tấn công này, đặc biệt là các giải pháp dựa trên máy học để phát hiện các hiện tượng bất thường của các gói tin [12, 13] hoặc trí tuệ nhân tạo (AI) [14, 15].

Tuy nhiên, để các giải pháp này phát huy hiệu quả, thì đòi hỏi một quá trình phân loại mã độc và lưu lương dữ liệu (*traffic data*) trước đó phải chính xác. Trong nghiên cứu này, chúng tôi đã đề xuất một phương pháp mới để cải thiện quá trình phân loại tấn công mạng từ lưu lượng dữ liệu (*traffic data*) để nâng cao tỷ lệ xác định chính xác các cuộc tấn công mạng. Trước tiên, đề xuất phương pháp tiền xử lý dữ liệu và trích chọn đặc trưng từ lưu lượng dữ liệu để tạo các véc-tơ đặc trưng. Bước tiếp theo, trong mỗi véc-tơ có 64 đặc trưng nên chúng tôi đề xuất chuyển các đặc trưng này về dạng ma trận kích thước 2×32, 4×16 và 8×8 (*image-base*) tương ứng. Cuối cùng, sử dụng mạng nơron tích chập (*Convolutional Neural Network* - CNN) để huấn luyện và nhận dạng các loại tân công. Mô hình đề xuất được thực nghiệm trên bộ dữ liệu Bot-IoT [16] để đánh giá kết quả mô hình đề xuất.

Bố cục của bài báo được tổ chức thành các phần chính như sau: Phần thứ nhất, chúng tôi giới thiệu về tính cấp thiết của nội dung nghiên cứu. Phần thứ hai, trình bày về những nghiên cứu liên quan như phương pháp trích chọn đặc trưng và phân loại dữ liệu sử dụng CNN. Phần thứ ba, trình bày về mô hình đề xuất bao gồm trích chọn đặc trưng và phân loại tấn công mạng. Phần thứ tư, là thực nghiệm và đánh giá kết quả của mô hình đề xuất. Cuối cùng là kết luận và hướng nghiên cứu trong tương lai.

II. NGHIÊN CỨU LIÊN QUAN

Cuộc cách mạng lần thứ tư đã tạo ra những bước đột phá trong nhiều lĩnh vực như trí tuệ nhân tạo, thị giác máy tính, IoT, xe tự hành, robot, in 3D và công nghệ sinh học. Tuy nhiên, việc phát triển đột phá trong lĩnh vực IoT đã làm gia

tăng các cuộc tấn công mạng và đe doạ các thiết bị IoT. Một số hãng công nghệ đã sử dụng các thiết bị phần cứng như bức tường lửa để ngăn chặn các cuộc tấn công xâm nhập vào hệ thống mạng nội bộ của mình. Tuy nhiên, các hệ thống này thường có bộ nhớ nhỏ, tốc độ tính toán chậm và cập nhật các mã độc mới khá khó trăn. Vì vậy, việc sử dụng các phương pháp học máy để phát hiện các cuộc tấn công mạng ngày càng được áp dụng phổ biến hơn.

Al-Garadi và nhóm nghiên cứu [17] đã áp dụng thuật toán học sâu để nhận dạng và bảo mật hệ thống. Xie và cộng sự [18] đã đề xuất xây dựng thành phố thông minh bằng giải pháp phát hiện xâm nhập dựa vào mô hình LSTM-NN và phân loại đa lớp bằng thuật toán Perceptron. Một số nghiên cứu khác đã sử dụng các thuật toán phân loại dữ liệu [19] dựa vào máy hỗ trợ véc-tơ, K-nearest Neighbor (KNN), và thuật toán Naive Bayes (NB).

Hơn nữa, dữ liệu của Botnet thường rất lớn và có cấu trúc phức tạp nên chúng ta cần có một số phương pháp để loại bỏ những thành phần không quan trọng và giảm xuống [20] số chiều của dữ liệu. Từ đó, các mô hình học máy sẽ phân lớp nhanh và chính xác hơn.

A. Phương pháp trích chọn đặc trưng

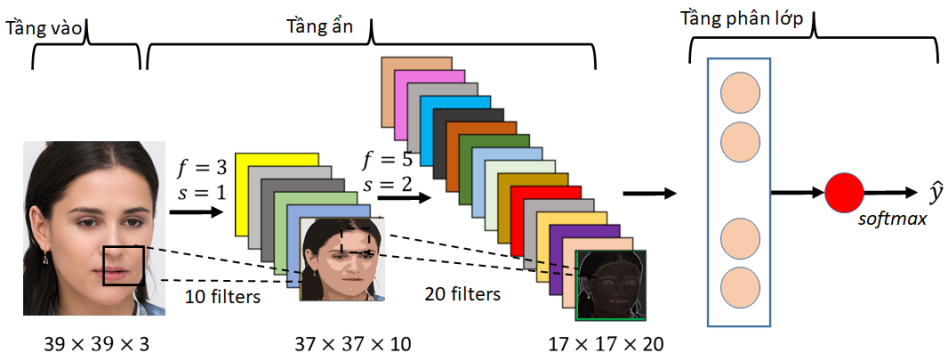
Phương pháp lựa chọn đặc trưng (*feature selection*) hay trích chọn đặc trưng (*feature extraction*) là quá trình biến đổi dữ liệu từ không gian có số chiều lớn sang không gian có số chiều nhỏ hơn để giảm chi phí tính toán và không gian lưu trữ. Hiện nay, có hai phương pháp là phân tích thành phần chính (PCA) và phân tích biệt thức tuyến tính (LDA) được áp dụng rất phổ biến để trích chọn đặc trưng. Một cách tiếp cận khác đơn giản hơn để giảm số chiều dữ liệu là chỉ lựa chọn những thông tin hay đặc trưng cần thiết từ dữ liệu đầu vào để giải quyết bài toán đặt ra ban đầu. Đối với tập dữ liệu Bot-IoT thì phương pháp trích chọn đặc trưng được đề xuất để gom các thuộc tính giống nhau thành một lớp để dễ phân loại các cuộc tấn công mạng hơn. **Bảng 1** là một số đặc trưng của dữ liệu BOT-IoT.

Bảng 1. Dữ liệu BOT-IoT

STT	Feature	Description
1	pkSeqID	Row Identifier
2	Stime	Record start time
3	Flgs	Flow state flags seen in transactions
4	flgs_number	Numerical representation of feature flags
5	Proto	Textual representation of transaction protocols present in network flow
6	proto_number	Numerical representation of feature proto
7	Saddr	Source IP address
8	Sport	Source port number
9	Daddr	Destination IP address
10	Dport	Destination port number

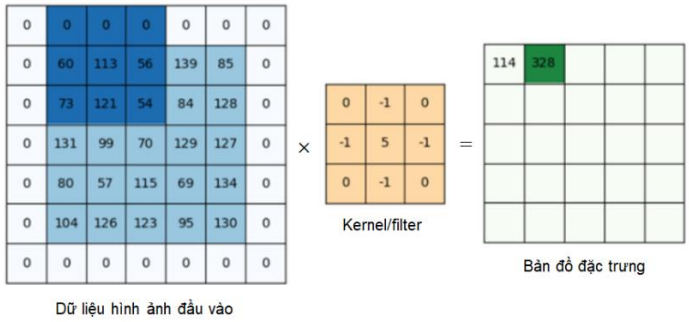
B. Mạng nơron tích chập

Học sâu là một trong những thuật toán trong học máy được sử dụng để xử lý và phân tích dữ liệu phức tạp và dữ liệu lớn. Mô hình học sâu được áp dụng khá phổ biến để phát hiện các cuộc tấn công mạng botnet từ môi trường IoT. Học sâu có 2 mô hình chính đó là mạng nơron tích chập thường được áp dụng trong xử lý ảnh [21] và mạng nơron hồi quy (*Recurrent Neural Network - RNN*) [22] để xử lý các bài toán nhận dạng chuỗi (*sequence/ time-series*) và nhận dạng hành động. Tuy nhiên, mô hình CNN cũng có thể giúp thiết kế các hệ thống hiệu quả cho các mục đích bảo mật [17, 23]. Các nơron trong mạng CNN không liên kết hoàn toàn với toàn bộ nơron kế tiếp mà chỉ liên kết với một vùng nhỏ. Cuối cùng, tầng phân lớp (đầu ra) được tối giản thành một véc-tơ với các giá trị xác suất dự đoán.



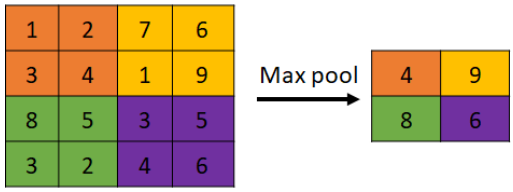
Hình 1. Kiến trúc chung của mạng CNN

(1) ***Tầng đầu vào:*** Dữ liệu đầu vào (*image-base*) được trích chọn đặc trưng để tạo dữ liệu dưới dạng ma trận.



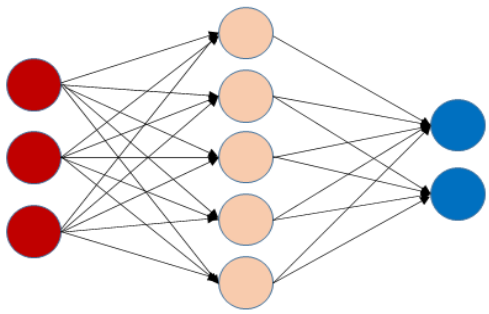
Hình 2. Phép tích chập trong mạng nơron tích chập

(2) ***Tầng ẩn:*** Tại tầng ẩn sẽ thực hiện các phép tích chập bằng cách trượt ma trận bộ lọc (*kernel/filter*) trên ma trận đầu vào. Tại mỗi vị trí tương ứng của ma trận đầu vào và ma trận bộ lọc, thực hiện phép nhân ma trận và tính tổng các giá trị để đưa vào bản đồ đặc trưng, phép tích chập minh họa tại **Hình 2**. Sau mỗi tầng tích chập, chúng ta cho kết quả đi qua một tầng hợp nhất (*max pool layer*) bằng cách chọn giá trị lớn nhất trong mỗi vùng **Hình 3** để giảm số chiều và thiết lập một ma trận mới.



Hình 3. Tầng hợp nhất (*max pool*) với bộ lọc và bước nhảy (*stride*) bằng 2 trong mạng nơron tích chập.

(3) ***Tầng phân lớp:*** Là tầng cuối cùng của mạng nơron tích chập được biểu diễn bằng lớp kết nối đầy đủ. Quá trình phân lớp sẽ tiến hành từ các đặc trưng đã được trích chọn trong tầng ẩn và kết quả thu được cuối cùng là một véc-tơ với các giá trị xác suất dự đoán, minh hoạ như **Hình 4.**



Hình 4. Tầng phân lớp trong CNN

C. Hàm hoạt động Softmax

Hàm **Softmax** là hàm tính toán xác suất xảy ra của một sự kiện trong học máy hay còn gọi là hàm xác suất xuất hiện của một lớp trong tổng số tất cả các lớp đầu vào. Hàm **Softmax** được áp dụng rất phổ biến trong mô hình phân loại hồi quy logistic và các mô hình mạng nơron nhận tạo.

Công thức hàm **Softmax**:

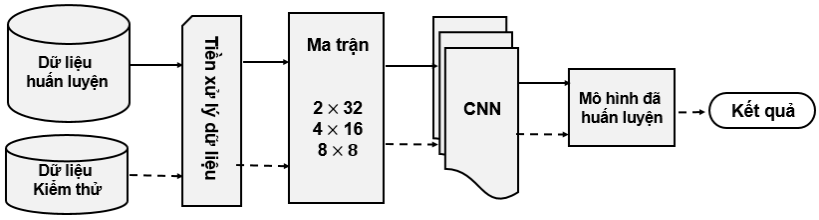
$$a_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}, \forall j = 1, 2, \dots, C \quad (1)$$

trong đó: $X = \{\boldsymbol{x}_i\}$ là tập dữ liệu đầu vào, a_i thể hiện xác suất để đầu vào đó rơi vào lớp j, giá trị $z_i = \boldsymbol{w}_i^T \boldsymbol{x}$, $\boldsymbol{w}$ là tham số mô hình (ma trận trọng số) và C là số lớp.

III. PHƯƠNG PHÁP ĐỀ XUẤT

Phương pháp đề xuất nhằm mục đích phân loại các cuộc tấn công mạng từ bộ dữ liệu Bot-IoT [16]. Trước tiên, dữ liệu Bot-IoT được tiến hành tiền xử lý để phân thành 05 nhóm và các thuộc tính trong mỗi nhóm sẽ có một số đặc trưng là giống nhau, như **Bảng 2.** Tiếp đến, dữ liệu sau khi tiền xử lý là các véc-tơ đặc trưng có kích thước là 64 chiều, vì vậy để sử dụng mạng nơron tích chập huấn luyện thì chúng tôi lần lượt chuyển các véc-tơ đặc trưng thành các ma trận có kích

thước là 2×32, 4×16 và 8×8. Cuối cùng, sử dụng mạng nơron tích chập để tiến hành phân loại các lớp dữ liệu, phương pháp đề xuất được minh hoạ trong **Hình 5**.

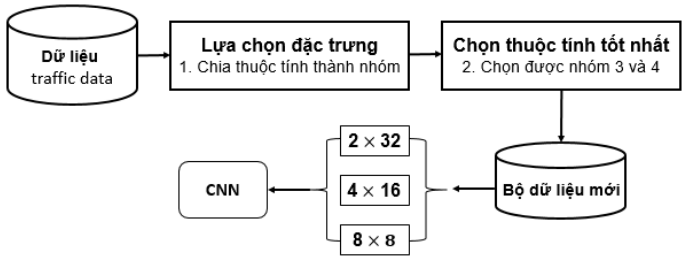


Hình 5. Mô hình đề xuất phân loại các cuộc tấn công bằng botnet

A. Tiền xử lý dữ liệu

Trong bài báo này, chúng tôi tiền xử lý dữ liệu bằng cách phân chia lưu lượng dữ liệu (*traffic data*) thành các nhóm có thuộc tính tương đồng nhau và quá trình này gọi là phương pháp trích chọn đặc trưng. Các luồng cơ bản trong quá trình trích chọn đặc trưng và xử lý dữ liệu cụ thể như sau:

- ***Bước 1***: *Lựa chọn thuộc tính*. Chúng tôi lựa chọn các thuộc tính có "chung" một số đặc điểm vào chung các nhóm giống nhau. Trong quá trình lựa chọn chúng tôi đã phân chia dữ liệu thành 05 nhóm, trong đó nhóm thứ 5 là chứa các nhãn dùng để phân nhóm.
- ***Bước 2***: *Loại bỏ các nhóm không liên quan*. Từ những thuộc tính trong các nhóm, chúng tôi nhận thấy nhóm thứ nhất có các thuộc tính không liên quan đến mục đích của bài toán nên loại bỏ nhóm thứ nhất. Như vậy, lúc này dữ liệu đầu vào để phân nhóm chỉ còn được 03 nhóm chứa tất cả các thuộc tính tốt nhất.
- ***Bước 3***: *Gộp các nhóm dữ liệu*. Sử dụng cây quyết định để tạo tổ hợp các nhóm và quá trình này đã chọn được nhóm 3 và nhóm 4 có những thuộc tính tốt nhất để làm đầu vào cho quá trình phân nhóm của bài toán.
- ***Bước 4***: *Tạo ma trận dữ liệu* (*image-base*): Quá trình gộp dữ liệu từ bước 3 chúng ta có được dữ liệu dưới dạng véc-tơ đặc trưng kích thước là 64 chiều. Từ đây, chúng tôi chuyển các véc-tơ đặc trưng thành các ma trận có kích thước là 2×32, 4×16 và 8×8 cho mô hình CNN phân loại.



Hình 6. Phương pháp trích chọn đặc trưng và tiền xử lý dữ liệu

B. Phương pháp nhận diện tấn công mạng dựa vào CNN

Như vậy, sau khi tiền xử lý dữ liệu và trích chọn đặc trưng thì tập dữ liệu mới X sẽ là:

$$X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\} \quad (2)$$

trong đó: $\boldsymbol{x}_i$ là véc-tơ đặc trưng có kích thước là 64 và $\boldsymbol{x}_i \in \{0 \to 255\}$.

Để sử dụng CNN phân loại các cuộc tấn công thì chúng ta chuyển véc-tơ $\boldsymbol{x}_i$ thành các ma trận $\boldsymbol{A}_{M,N}$. trong đó, $(M = 2, N = 32)$, $(M = 4, N = 16)$ hoặc $(M = 8, N = 8)$.

1. CNN với tầng đầu vào ma trận $\mathbf{A}_{2,32}$

Tầng đầu vào trong trường hợp này là mà trận $\boldsymbol{A}_{2,32}$ nên chúng ta chỉ sử dụng bộ trượt (*convolutions - kernel/filter*) có kích thước 2×2 và trượt trên ma trận đầu vào $\boldsymbol{A}_{2,32}$. Trường hợp này, chỉ sử dụng quy nhất một bộ trượt có kích thước 2×2 và sau khi trượt hết trên tầng đầu vào thì chúng ta sẽ có được một véc-tơ $\boldsymbol{x}'_i$. Từ tầng ẩn này, chúng ta có thể chuyển $\boldsymbol{x}'_i$ sang tầng kết nối đầy đủ (*full connected*) là $\boldsymbol{x}'^T$cho mạng nơron phân loại. Để xác định dữ liệu được phân vào nhóm nào thì chúng ta sử dụng hàm **Softmax** để tính xác suất đầu ra cho từng trường hợp.

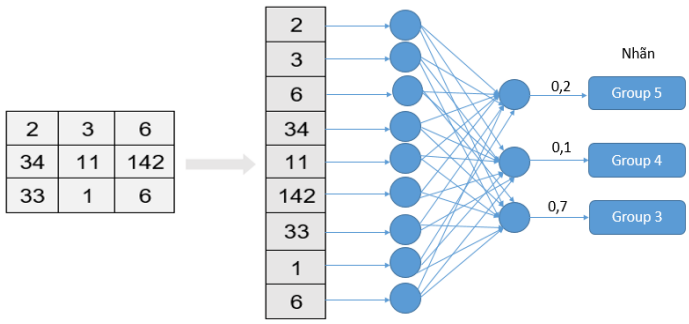
2. CNN với dữ liệu đề xuất ma trận $\mathbf{A}_{4,16}$

Trong trường hợp tầng đầu vào là ma trận $\boldsymbol{A}_{4,16}$ thì chúng ta có thể sử dụng bộ trượt lần đầu có kích thước 3×3 để tạo ra tầng ẩn thứ hai có kích thước $\boldsymbol{A}'_{2,13}$. Từ tầng ẩn này, có thể sử dụng bộ trượt thứ hai có kích thước 2×2 để tạo ra tầng ẩn tiếp theo có dạng véc-tơ $\boldsymbol{x}'_i$ và thực hiện các bước như phần trên.

3. CNN với dữ liệu đề xuất ma trận $\mathbf{A}_{8,8}$

Đối với ma trận đầu vào $\boldsymbol{A}_{8,8}$ thì chúng ta có thể sử dụng nhiều bộ trượt có kích thước khác nhau để tạo véc-tơ $\boldsymbol{x}'_i$ và tầng kết nối đầy đủ (*full connected*) là $\boldsymbol{x}'^T$. Trong trường hợp tầng ẩn tiếp theo là một ma trận thì chúng ta có thể

chuyển ma trận này về tầng kết nối đầy đủ (*full connected*) là $\boldsymbol{x}'^{T}$ bằng cách chuyển các hàng của mang trận thành các cột của véc-tơ, minh hoạ như hình sau:



Hình 7. Chuyển dữ liệu từ tầng ẩn (bên trái) sang tầng kết nối đầy đủ (bên phải)

IV. KẾT QUẢ THỰC NGHIỆM

Thực nghiệm được tiến hành trên máy tính với hệ điều hành Windows 10, CPU: i7, Memory: 8GB, môi trường phát triển: Python 3.6, Matplotlib: phiên bản 3.2, NumPy: 1.18, Pandas: 1.0,.... Các bước thực nhiệm được tiến hành dựa trên mô hình đã đề xuất để điều chỉnh các tham số trong mô hình và đánh giá kết quả.

A. Tiền xử lý dữ liệu

Dữ liệu BOT-IoT từ **Bảng 1** được tiến hành tiến xử lý, kết quả được chia thành 5 nhóm với các thuộc tính tương đồng nhau, các nhóm có tên: *Network Flow Identifiers, Attack Mode, Sliding Window, Network Flow Characteristics* và *Log Info*, như **Bảng 2**.

Bảng 2. Dữ liệu BOT-IoT được chia thành 5 nhóm dựa vào các đặc trưng

№	Group_1 (Network Flow Identifiers)	Group_2 (Related to Attack mode)	Group_3 (Related to sliding window)	Group_4 (Network Flow Characteristics)	Group_5 (Log info)
1	Proto	Attack	TnBPSrcIP	Flgs	pkSeqID
2	proto_number	Category	TnBPDstIP	flgs_number	Stime
3	Saddr	Subcategory	TnP_PSrcIP	Pkts	Ltime
4	Sport		TnP_PDstIP	Bytes	Seq
5	Daddr		TnP_PerProto	State	Dur
6	Dport		TnP_Per_Dport	state_number	

Dữ liệu từ sau khi được phân chia nhóm từ **Bảng 2** sẽ được tiến hành mô hình hoá bằng cách sử dụng hàm z (*z-function*) [24] để tạo bộ dữ liệu mới X từ công thức (2), với $\boldsymbol{x}_i$ có kích thước là 64 và các giá trị của $\boldsymbol{x}_i$ từ 0 đến 255, như **Bảng 3**.

Bảng 3. Dữ liệu của véc-tơ đặc trưng sau khi tiền xử lý

№	1	2	3	4	5	6	7	...	59	61	60	61	62	63	64
$\boldsymbol{x}_1$	7	9	34	211	9	55	110	...	77	6	13	87	44	5	99
$\boldsymbol{x}_2$	4	56	75	33	23	85	22	...	2	34	17	8	223	161	11
$\boldsymbol{x}_3$	76	55	96	17	58	5	65	...	11	2	17	91	27	52	43
$\boldsymbol{x}_4$	22	45	235	83	54	71	83	...	57	68	9	8	215	97	37
...	...	...	...	...	..	...	...	...	...	...	...	...	...	...	...

B. Phương pháp đánh giá kết quả

Mô hình phân loại được ứng dụng rộng rãi trong các ứng dụng như nhận diện khuôn mặt, phân loại video youtube, phân loại văn bản, phân loại giọng nói… Để kiểm tra hệ thống phân loại chính xác đến mức độ nào thì các khái niệm về độ đo như accuracy, recall, precision, và F1-score được xác định, trong đó:

$$
\begin{aligned}
&\blacksquare\ \text{Accuracy} = \frac{TP+TN}{FP+FN+TP+TN} \times 100\% \\
&\blacksquare\ \text{Precision} = \frac{TP}{TP+FP} \times 100\% \\
&\blacksquare\ \text{Recall} = \frac{TP}{TP+FN} \times 100\% \\
&\blacksquare\ \text{F1} - \text{score} = 2 * \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \times 100\%
\end{aligned}
\qquad (3)
$$

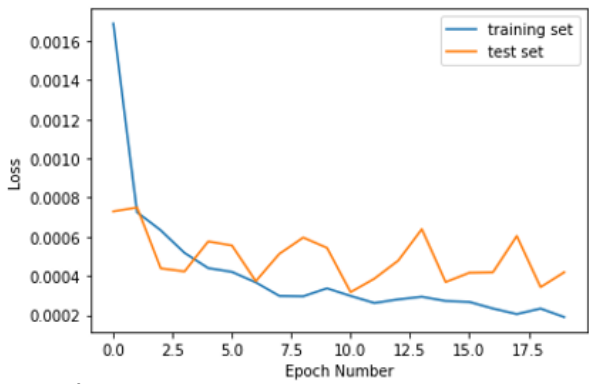
trong đó: TP (*True Positive*) là đối tượng ở lớp Positive (có), mô hình phân đối tượng vào lớp Positive (dự đoán đúng); TN (*True Negative*) là đối tượng ở lớp Negative (không) được phân vào lớp Negative (dự đoán đúng); FP (*False Positive*) là đối tượng ở lớp Negative được phân vào lớp Positive (dự đoán sai); FN (*False Negative*) là đối tượng ở lớp Positive được phân vào lớp Negative (dự đoán sai).

C. Kết quả thực nghiệm

Thực nghiệm được thực hiện dựa trên mô hình đề xuất là tiền xử lý dữ liệu và chuyển dữ liệu từ véc-tơ đặc trưng có kích thước 64 chiều (giá trị) thành ma trận có kích thước là 2×32, 4×16 và 8×8 cho mô hình CNN phân loại. Mô hình thực nghiệm được thực hiện với số epochs = 20 và hàm hoạt động là hàm **softmax**, Kết quả thực nghiệm như sau:

– **Hình 8** là kết quả sử dụng ma trận có kích thước 2×32, với precision = 85 %, recall = 95 % và F1-score = 91 %. Đây là kết quả khá cao có thể áp dụng được trong thực tế, tuy nhiên cũng cần thêm một số thực nghiệm khác trên mô hình đề xuất như ma trận 4×16 và ma trận 8×8.

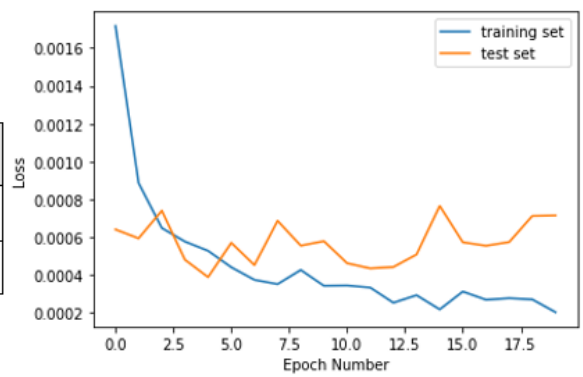
	Precission (%)	Recall (%)	F1-score (%)
Test	88	95	91
Training	100	100	100



Hình 8. Bảng kết quả và biểu đồ hàm mất mát (*Loss Function*) của ma trận 2×32.

– Trong **Hình 9** là kết quả của thực nghiệm dựa trên ma trận đặc trưng có kích thước 4×16, từ kết quả cho thấy với ma trận kích thước 4×16 có kết quả các đơn vị đo đều tốt hơn ma trận 2×32. Tuy nhiên, phần Recall kết quả 89 % thấp hơn so với kết quả của ma trận 2×32 là 95 %.

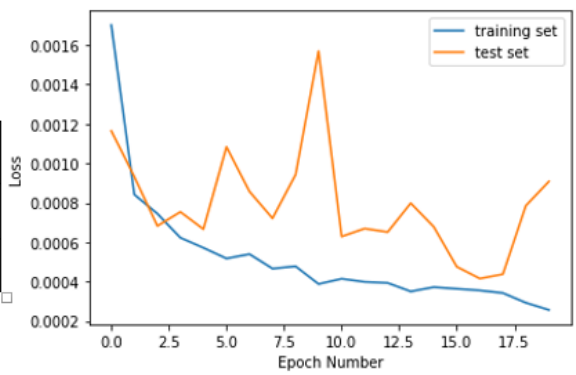
	Precission (%)	Recall (%)	F1-score (%)
Test	96	89	92
Training	100	100	100



Hình 9. Kết quả đề xuất chuyển véc-tơ đặc trưng thành ma trận 4×16

– Kết quả thực nghiệm cuối cùng với đề xuất ma trận đặc trưng có kích thước 8×8, **Hình 10** cho thấy kết quả precission = 96 % là cao hơn hai trường hợp trên, riêng Recall = 66 % và F1-score = 80 % thấp hơn hai trường hợp trên.

	Precission (%)	Recall (%)	F1-score (%)
Test	99	66	80
Training	100	100	100



Hình 10. Kết quả đề xuất chuyển véc-tơ đặc trưng thành ma trận 8×8

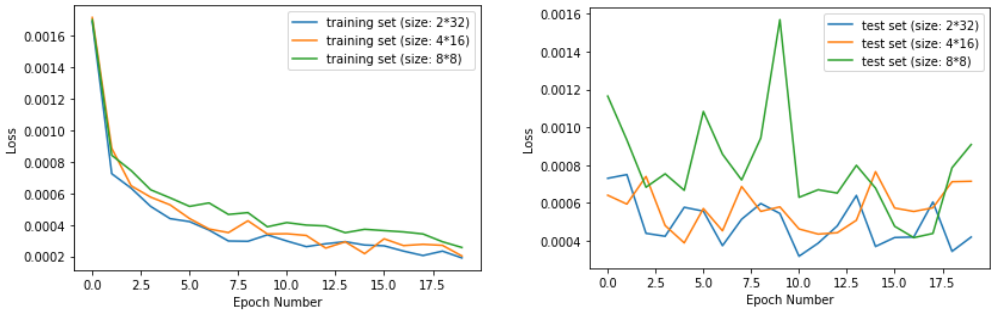
D. Đánh giá kết quả

Bảng 3 là bảng so sánh kết quả của quá trình thực nghiệm, nhìn chung phương pháp đề xuất có kết quả rất cao và có thể ứng dụng mô hình này vào thực tế để phát hiện các cuộc tấn công mạng. Kết quả ở **Bảng 3** cho thấy kết quả này tốt hơn so với một số nghiên cứu trước đây [23] chỉ sử dụng mô hình CNN-LSTM để phát hiện các cuộc tấn công mạng.

Bảng 3. Bảng so sánh các kết quả thực nghiệm

		Accuracy (%)	Precission (%)	Recall (%)	F1-score (%)
Test	2 × 32	100	88	95	91
	4 × 16		96	89	92
	8 × 8		99	66	80
Training			100	100	100

Kết quả từ **Bảng 3** cho thấy với việc chia véc-tơ đặc trưng thành ma trận 8 × 8 thì kết quả precission = 99 % là tốt nhất và accuracy = 100 % độ chính xác tuyệt đối cho các trường hợp. Tuy nhiên, kết quả của recall và F1-score lại thấp hơn so với hai trường hợp khác, từ công thức (3) có thể giả thích recall thấp là do đối tượng (dữ liệu) ở lớp Positive được phân vào lớp Negative tăng dẫn đến FN cao lên.



(a) *(b)*

Hình 11. Biểu đồ hàm mất mát (*Loss function*) trong thực nghiệm

Từ **Hình 11(b)** cho thấy hàm mất mát trong trường hợp ma trận đặc trưng 8 × 8 là không ổn định và có xu hướng cao hơn các trường hợp khác. Nguyên nhân là mô hình CNN không liên kết hoàn toàn với toàn bộ nơron kế tiếp mà chỉ liên kết với một vùng nhỏ với tầng trước, do đó càng sử dụng ít bộ trượt (convolution) thì đặc trưng bị biến đổi ít hơn. Trường hợp, ma trận 2 × 32 chúng ta chỉ cần sử dụng bộ trượt (convolution) một lần, ma trận 4 × 16 có thể sử dụng 2 lần và ma trận 8 × 8 thì nhiều hơn dẫn đến biểu đồ hàm mất mát sẽ phức tạp hơn.

V. KẾT LUẬN

Trong bài báo chúng tôi sử dụng phương pháp trích chọn đặc trưng từ tập dữ liệu đầu vào để từ đó chọn ra một số các đặc trưng đủ tốt để phân nhóm. Từ đó, dữ liệu được biểu diễn lại dưới dạng ma trận (ảnh xám) để áp dụng CNN phân lớp theo từng dạng kích thước ma trận. Các dạng kích thước của ma trận đầu vào có ảnh hưởng đến kết quả phân lớp của bài toán. Quá trình thực nghiệm, cho thấy khi kích thước ma trận hình dạng ở dạng vuông 8 × 8 thì kết quả phân lớp đạt tỉ lệ cao hơn so với những trường hợp khác.

Trong tương lai, chúng tôi sẽ áp dụng phương pháp đề xuất này vào nhiều bộ dữ liệu khác nhau để kiểm chứng thêm kết quả của mô hình đề xuất, từ đó áp dụng mô hình đề xuất vào trong thực tế để nhận diện các cuộc tấn công mạng. Đề xuất trích chọn đặc trưng cho mạng nơron tích chập trong bài toán nhận diện tấn công mạng có tính thực tiễn rất cao, giá thành rẻ và có thể áp dụng được trên rất nhiều hệ thống và thiết bị khác nhau.

Lời cảm ơn: Bài báo này được tài trợ bởi Trường Đại học Bách khoa, Đại học Đà Nẵng với đề tài có mã số: T2022-02-12.

TÀI LIỆU THAM KHẢO

[1] T. Seals, "Iot attacks skyrocket, doubling in 6 months," Online: https://threatpost.com/iot-attacks-doubling/ 169224/, 2021.

[2] Cisco, "Cisco Visual networking Index (VNI)," Global Mobile data traffic Forecast update, 2017–2022, San Jose, CA, USA, 2019.

[3] Broadcom, "Symantec Internet Security Threat Report 2019," Tech. Rep., 2020," https://docs.broadcom.com/doc/ istr-24-2019-en., 2020.

[4] M. Kuzin, Y. Shmelev, and V. Kuskov, "New Trends in the World of IoT Threats-Securelist Kaspersky Lab," https://securelist.com/new-trends-in-the-world-of-iot-threats/87991/., 2018.

[5] Ncsc, "Tình hình an toàn thông tin tháng 7/2022," Trung tâm Giám sát an toàn không gian mạng quốc gia (NCSC), 2022.

[6] K. Zhang, X. Liang, R. Lu, and X. Shen, "Sybil attacks and their defenses in the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 5, p. 372–383, 2014.

[7] R. E. Navas, H. Le Bouder, N. Cuppens, F. Cuppens, and G. Z. Papadopoulos, "Do not trust your neighbors! a small iot platform illustrating a man-in-the-middle attack," in *in International conference on ad-hoc networks and wireless*, 2018.

[8] S. Choudhary and N. Kesswani, "Detection and prevention of routing attacks in internet of things," in *17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018.

[9] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *Journal of Network and Computer Applications,* vol. 128, p. 33–55, 2019.

[10] S. Plaga, N. Wiedermann, S. D. Anton, S. Tatschner, H. Schotten, and T. Newe, "Securing future decentralised industrial iot infrastructures: Challenges and free open source solutions," *Future Generation Computer Systems,* p. 596–608, 2019.

[11] A. Costin and J. Zaddach, "Iot malware: Comprehensive survey, analysis framework and case studies," BlackHat USA, 2018.

[12] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K.-K. R. Choo, "A deep recurrent neural network based approach for internet of things malware threat hunting," *Future Generation Computer Systems,* vol. 85, p. 88–96, 2018.

[13] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of eight data mining algorithms for smarter internet of things (IoT)," *rocedia Computer Science,* vol. 98, p. 437–442, 2016.

[14] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, and N. O. P.IoT. Tippenhauer, "A machine learning approach for IoT device identification based on network traffic analysis," in *Proceedings of the Symposium on Applied Computing*, Morocco, April 2017.

[15] H. Alkahtani, T. Aldhyani, and M. Al-Yaari, "Adaptive anomaly detection framework model objects in cyberspace," *Applied Bionics and Biomechanics,* vol. 2020, p. 14, 2020.

[16] N. Moustafa, "The bot-iot dataset," *IEEE Dataport,* vol. 5, 2019.

[17] M. A. Al-Garadi, A. Mohamed, A. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," *EEE Communications Surveys & Tutorials,* vol. 22, no. 3, 2018.

[18] X. Xie, D. Wu, S. Liu, and R. Li, "IoT Data Analytics Using Deep Learning," https://arxiv.org/abs/1708.03854, 2017.

[19] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of eight data mining algorithms for smarter internet of things (IoT)," *Procedia Computer Science,* vol. 98, p. 437–442, 2016.

[20] P. Rodr´ıguez, M. A. Bautista, J. Gonzalez, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Image and Vision Computing,* vol. 75, pp. 21 - 31, 2018.

[21] S. Ji, W. Xu, M. Yang, and K. Yu, "3D Convolutional Neural Networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 35, no. 1, p. 221–231, 2013.

[22] L. Jun, G. Wang, Li..Y. Duan, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Transactions on Image Processing,* vol. 27, no. 4, pp. 1586-1599, 2017.

[23] Hasan Alkahtani, Theyazn H. H. Aldhyani, "Botnet Attack Detection by Using CNN-LSTM Model for Internet of Things Applications," *Security and Communication Networks,* vol. 2021, p. 23, 2011.

[24] A. Ivić, The theory of Hardy's Z-function, Cambridge University Press: Cambridge Tracts in Mathematics, 2013, p. 245.

FEATURE EXTRACTION METHOD FOR CONVOLUTIONAL NEURAL NETWORK IN THE PROBLEM OF DECTECT NETWORK ATTACKS

Nguyen Nang Hung Van, Do Phuc Hao, Pham Minh Tuan

ABSTRACT: *Today, the development of smart devices and computer networks has led to increasingly common cyber attacks. Hackers use a variety of cyberattack techniques to gain unauthorized access to computer systems and IoT to steal information or encrypt important information and demand ransom. Therefore, the issue of network security is very important and has a great impact on the performance of computer networks and IoT devices. One of the most popular measures today to ensure the security of computer network systems is the intrustion detector system. However, these measures are ineffective, unreliable and do not update themselves to detect newer intrusions. A new approach increasingly showing its superiority and overcoming the above limitations is the application of machine learning techniques to detect network attacks. This paper proposes a new method of feature extraction to classify computer network attacks based on packet characteristics using deep learning. In addition, the paper shows important features in the Bot-IoT data set that can be set up as a matrix for the convolutional neural network to classify and improve the detection accuracy of the network attack. First, the study proposes a data pre-processing method to optimize the data. In the next step, use the feature extraction method to create vectors and matrices. Finally, the study uses CNN to classify based on feature matrices. The proposed model is tested on the Bot-IoT dataset and the best result has an accuracy of 99 %.*