Tạp chí Khoa học và Kinh tế Phát triển
Trường Đại học Nam Cần Thơ

Website: jsde.nctu.edu.vn

Sử dụng kỹ thuật học sâu trong lĩnh vực trí tuệ nhân tạo để nâng cao khả năng phát hiện các cuộc tấn công xâm nhập máy tính qua mạng Internet

Trần Thanh Nam[1*], Trương Hùng Chen[1], Nguyễn Văn Linh[1]

[1] *Trường Đại học Nam Cần Thơ*

** Người chịu trách nhiệm bài viết: Trần Thanh Nam (email: ttnam@nctu.edu.vn)*

Ngày nhận bài: 10/5/2023
Ngày phản biện: 18/6/2023
Ngày duyệt đăng: 25/7/2023

Title: *Using Deep Learning techniques in Artificial Intelligence to advanced warning of computer attacks from the internet*

Keywords: *artificial intelligence, artificial neural network, deep learning technique, internet security*

Từ khóa: *an ninh mạng, kỹ thuật học sâu, mạng nơ-ron nhân tạo, trí tuệ nhân tạo,*

ABSTRACT

Nowadays, there is an explosion of information and communication with the indispensable role of the Internet in modern life. Computer attacks are becoming more and more sophisticated and increasing in scale and quantity. The effectiveness of current traditional IDS intrusion protection systems has many limitations. With the aim of providing a network security solution with a new trend, this paper introduced a network intrusion detection system applying a deep learning method based on Artificial Neural Network (ANN) and Multi-layers Perceptron (MLP). Model evaluation methods showed that the proposed solution worked effectively.

TÓM TẮT

Trong thời đại bùng nổ thông tin và truyền thông hiện nay, mạng Internet đóng vai trò không thể thiếu trong cuộc sống hiện đại, vấn đề an toàn thông tin khi sử dụng môi trường mạng cần được đặc biệt quan tâm với tình hình các cuộc tấn công mạng ngày càng trở nên tinh vi và tăng dần về quy mô, số lượng. Hiệu quả của các hệ thống bảo vệ tấn công xâm nhập mạng IDS truyền thống hiện tại đã có nhiều hạn chế. Với mục đích cung cấp một giải pháp an ninh mạng với xu hướng mới, bài viết này giới thiệu một hệ thống phát hiện xâm nhập mạng áp dụng phương pháp học sâu dựa trên mạng nơ-ron nhân tạo (Artificial Neural Network – ANN) và đa lớp (Multi-layers Perceptron- MLP). Các phương pháp đánh giá mô hình cho thấy rằng giải pháp được đề xuất hoạt động hiệu quả.

1. GIỚI THIỆU

Sự phổ biến ngày càng tăng theo cấp số nhân của máy tính và mạng Internet đã khiến chúng trở thành mục tiêu thường xuyên của các cuộc tấn công và xâm nhập mạng (Vinayakumar et al., 2019) [1]. Thiệt hại về kinh tế do các cuộc tấn công mạng gây ra là rất lớn. 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) [2]. 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) [3]. Thời gian qua đã có một số nghiên cứu (Mafra et al., 2010; Vinayakumar et al., 2019) [4],[1] áp dụng các kỹ thuật máy học (Machine learning) với mục đích nâng cao hiệu quả và khắc phục các nhược điểm của các giải pháp IDS truyền thống. Tuy nhiên kỹ thuật máy học áp dụng các thuật toán khá phức tạp 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. Nhằm giải quyết hạn chế của những giải pháp trên, bài viết này thiết kế một hệ thống phát hiện xâm nhập mạng sử dụng kỹ thuật học sâu (Deep learning) là một bước đột phá của máy học, có nhiều cải tiến mạnh mẽ trong lĩnh vực trí tuệ nhân tạo (AI). Khi ứng dụng kỹ thuật học sâu, các hệ thống IDS có thể "học" các mối liên hệ ẩn từ các dữ liệu mạng của những hành vi tấn công và xâm nhập đã biết trước đây. Từ đó có thể phát hiện các cuộc tấn công mạng trong tương lai hoặc các cuộc tấn công mạng không lường trước (unknown attacks) một cách linh hoạt và hiệu quả.

1.1 CƠ SỞ LÝ THUYẾT

1.1.1 Khái niệm tấn công, xâm nhập mạng

Tấn công mạng, xâm nhập mạng là các tác động hoặc là trình tự liên kết giữa các tác động với nhau để hiện thực hóa các nguy cơ gây hại bằng cách lợi dụng lỗ hổng của các hệ thống thông tin. Theo một cách khác xâm nhập mạng có thể được định nghĩa là hành động cố gắng phá hủy sự toàn vẹn, bí mật của một tài nguyên hoặc đi ngược lại mục tiêu bảo mật của một tài nguyên nào đó.

1.1.2 Các kiểu tấn công xâm nhập mạng phổ biến hiện nay

Spyware, Keylogger, Backdoor: Hay còn gọi là tấn công thành lập "cửa sau", là cách tấn công cài đặt các ứng dụng cho phép truy cập từ xa vào máy tính nạn nhân. Trong loại vi phạm này, hacker sử dụng các loại phần mềm gián điệp được thiết kế, lập trình tinh vi, hoạt động ở chế độ ẩn có các chức năng như: điều khiển từ xa, ghi thông tin gõ bàn phím, duyệt và lấy cắp dữ liệu, chụp ảnh màn hình, mở micro ghi tín hiệu thoại,… mã hóa dữ liệu và gửi vào email được lưu trên một máy chủ trung gian để giấu nguồn gốc của hacker.

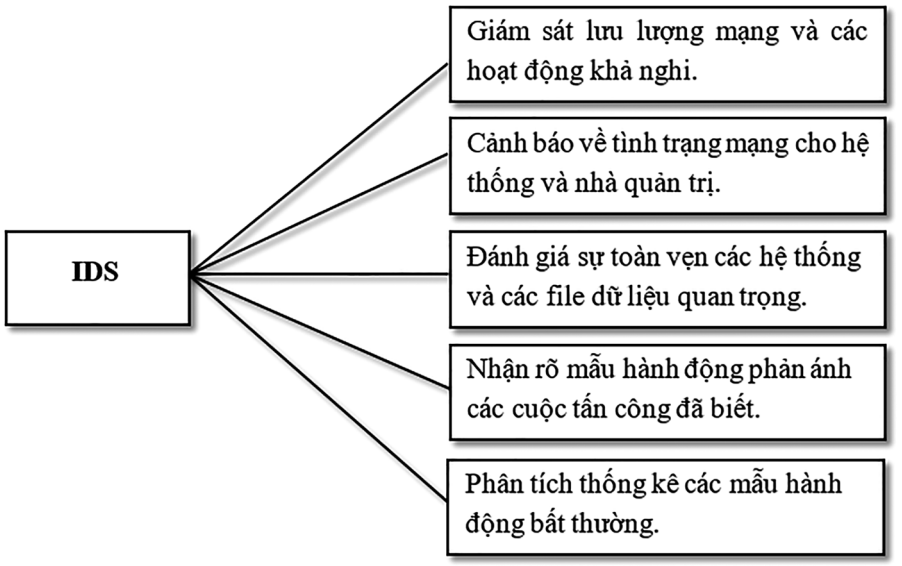
Tấn công từ chối dịch vụ DoS- Botnet: Một cuộc tấn công từ chối dịch vụ (DoS) ngăn cản người dùng hợp pháp truy cập vào các dịch vụ hoặc thông tin. Nó thành công khi một kẻ tấn công làm quá tải một máy chủ với yêu cầu nhiều hơn các máy chủ có thể xử lý. DoS là chỉ một kẻ nào đó tấn công kiểm soát máy tính và sử dụng chúng để làm tràn một email đặc biệt với các tin nhắn hoặc một trang web có khối dữ liệu khổng lồ. Các tấn công DoS bao gồm TCP-SYN Flood, ICMP/UDP Flood, Smurf, Ping of Death, Teardrop, Mailbomb, Apache2,…

Tấn công thăm dò (Reconnassance): Là hình thức tấn công nhằm thu thập các thông tin về hệ thống mục tiêu, từ đó phát hiện ra các điểm yếu. Tấn công do thám thường để làm bàn đạp cho cuộc tấn công truy cập hoặc tấn công từ chối dịch vụ về sau. Để tấn công thăm dò, hacker thường dùng các công cụ truy vấn thông tin Internet, Ping sweep, Port Scan, Packet sniffer,…

Tấn công khai thác mối quan hệ tin cậy: Khi hệ thống A và hệ thống B có mối quan hệ tin cậy nhau, các điểm yếu của hệ thống B có thể bị hacker lợi dụng để tấn công vào hệ thống A, vì các truy cập từ hệ thống B vào hệ thống A được xem là hợp lệ. Để ngăn chặn tấn công lợi dụng mối quan hệ tin cậy, người quản trị hệ thống phải hạn chế các mối quan hệ tin cậy từ hệ thống mạng bên trong với hệ thống mạng bên ngoài.

1.1.3 Hệ thống phát hiện xâm nhập mạng IDS

Hệ thống phát hiện xâm nhập mạng (Intrusion Detection System - IDS) là một hệ thống giám sát lưu lượng mạng nhằm phát hiện các hiện tượng bất thường và các hoạt động xâm nhập trái phép vào hệ thống máy tính. Các tính năng quan trọng nhất của một hệ thống IDS được mô tả trong Hình 1.



Hình 1. Các tính năng của một hệ thống IDS

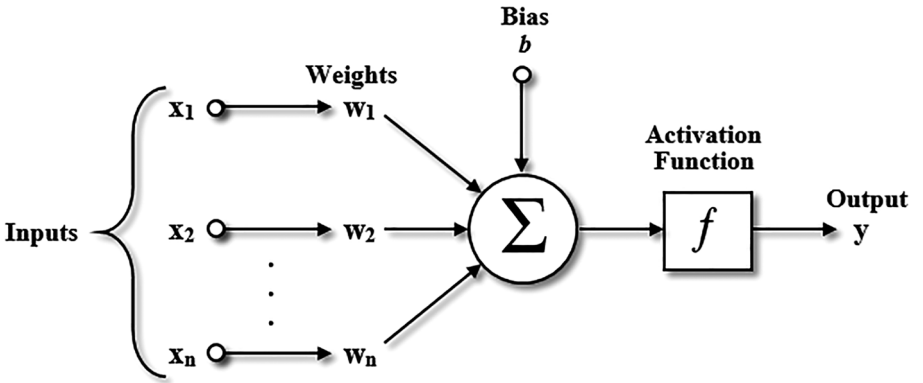
Trong thực tế, các IDS thường được kết hợp với các công cụ giám sát, tường lửa, chương trình phát hiện mã độc,... để tạo thành một hệ thống bảo mật hoàn chỉnh cho các hệ thống công nghệ thông tin. Các hệ thống IDS có thể triển khai theo hai mô hình chính bao gồm: Network-based (NIDS) được cài đặt trên 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) [5]. Mặc dù khá hiệu quả trong việc phát hiện các hình tấn công đã biết và đã 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) [3].

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) [1] đã áp dụng các thuật toán máy học và DNN để phát hiện các cuộc tấn công xâm nhập mạng không lường trước. Nghiên cứu của Mafra et al. (2010) [4] 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 (Mafra et al., 2010; Mishra et al., 2019; Vinayakumar et al., 2019) [4],[3],[1] 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.

1.1.4 Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN)

Mạng nơ-ron nhân tạo mô phỏng một tập hợp các tế bào thần kinh kết nối với nhau, đầu ra của nơ-ron này có thể là đầu vào của nơ-ron khác. Hình 2 mô tả mạng nơ-ron nhân tạo được tạo từ các node xếp chồng các lớp lên nhau giữa vector đặc trưng và vector đích. Mạng nơ-ron nhân tạo đơn giản nhất tạo từ một node được gọi

là "perceptron". Giống như các tế bào thần kinh sinh học có các nhánh và sợi trục, mạng nơ-ron nhân tạo đơn là một cấu trúc cây đơn giản có các node đầu ra kết nối với mỗi node đầu vào.



Hình 2. Cấu trúc mạng nơ-ron nhân tạo

Trong đó:

x_1, x_2,… x_n: Các tín hiệu vào của nơ-ron, được biểu diễn dưới dạng vector N chiều.

w_1, w_2,… w_n: Các trọng số tương ứng với các tín hiệu vào. Đây là thành phần quan trọng của một mạng nơ-ron, nó thể hiện mức độ quan trọng (độ mạnh) của dữ liệu đầu vào đối với quá trình xử lý thông tin. Quá trình học của mạng nơ-ron thực chất là quá trình điều chỉnh trọng số (weight) của các dữ liệu đầu vào để có được kết quả mong muốn.

Hàm tổng (Summation Function): Tính tổng trọng số của tất cả các input được đưa vào mỗi nơ-ron. Hàm tổng của một nơ-ron đối với n input được tính theo công thức sau:

$$y = \sum_{i=1}^{n} x_i w_i \quad (2.1)$$

b: Độ lệch (bias). Tất cả các nơ-ron đều cho sẵn một độ lệch (b). Độ lệch là một tham số điều chỉnh vô hướng của nơ-ron, nó không phải là một đầu vào, song hằng số phải được xem như đầu vào và nó cần được coi như vậy khi xem xét độ phụ thuộc tuyến tính của các vector đầu vào.

y: Đầu ra của nơ-ron.

f: Hàm kích hoạt (activation function). Được dùng để giới hạn phạm vi đầu ra của mỗi nơ-ron. Một số hàm kích hoạt thường được sử dụng là:

Hàm sigmoid (Sigmoid function): Hàm này đặc biệt thuận lợi khi sử dụng cho các mạng được huấn luyện bởi thuật toán lan truyền ngược (back-propagation) bởi vì nó dễ lấy đạo hàm, do đó có thể giảm đáng kể tính toán trong quá trình huấn luyện. Hàm này được ứng dụng cho các chương trình ứng dụng mà các đầu ra mong muốn rơi vào khoảng [0,1].

Hàm tanh (x): Là một phiên bản thay đổi kích thước của hàm sigmoid, và phạm vi đầu ra của nó là [-1, 1] thay vì [0, 1].

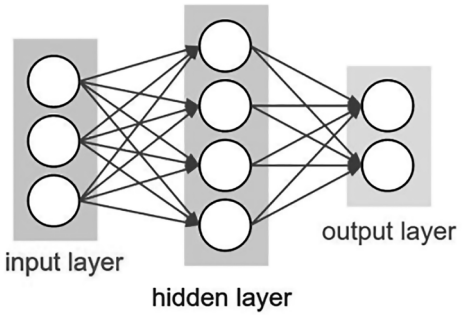
Hàm ReLU (Rectified Linear Unit): Được sử dụng rộng rãi gần đây vì tính đơn giản của nó. Ưu điểm chính của hàm này là giúp cho việc huấn luyện các mạng sâu (Deep Networks) nhanh hơn rất nhiều. Sự tăng tốc này được cho là vì ReLU được tính toán gần như tức thời và gradient của nó cũng được tính cực nhanh với gradient bằng 1 nếu đầu vào lớn hơn 0, bằng 0 nếu đầu vào nhỏ hơn 0.

Kiến trúc chung của mạng nơ-ron nhân tạo mô tả ở Hình 3 bao gồm *3* thành phần: Lớp đầu vào (Input Layer), Lớp ẩn (Hidden Layer) và Lớp đầu ra (Output Layer). Cụ thể:

Input Layer: Mỗi input tương ứng với một thuộc tính của dữ liệu đầu vào.

Output Layer: Lớp này sẽ cung cấp các kết quả đầu ra mà chúng ta mong muốn tính được.

Hidden Layer: Gồm các nơ-ron nhận dữ liệu input từ các nơ-ron ở lớp trước đó và chuyển đổi các input này cho các lớp xử lý tiếp theo. Trong một ANN có thể có nhiều Hidden Layer.



Hình 3. Kiến trúc mạng nơ-ron nhân tạo

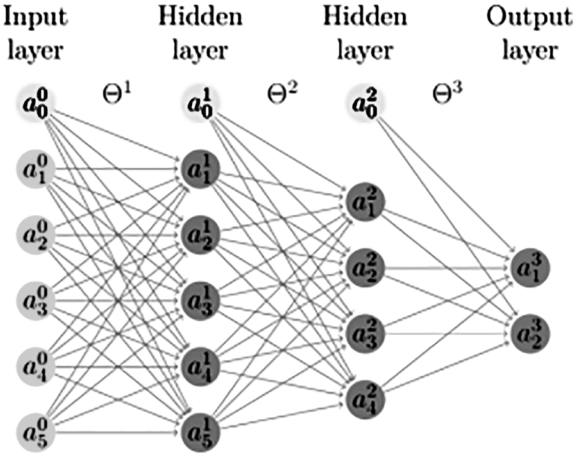
1.1.5 Kỹ thuật học sâu và kiến trúc mạng MLP

Học sâu (deep learning) là một bước đột phá của máy học (machine learning), có nhiều cải tiến mạnh mẽ trong lĩnh vực trí tuệ nhân tạo (AI). Kỹ thuật học sâu dựa trên một tập hợp các thuật toán để trừu tượng hóa mô hình dữ liệu ở mức cao bằng cách sử dụng nhiều lớp xử lý với cấu trúc phức tạp hoặc bằng nhiều biến đổi phi tuyến. Mô hình học sâu được áp dụng trong các lĩnh vực như: thị giác máy tính, xử lý ngôn ngữ tự nhiên... và cũng có thể áp dụng để cải thiện khả năng phát hiện xâm nhập mạng đạt tỷ lệ cao và tỷ lệ báo động giả thấp.

Học sâu sử dụng mạng nơ-ron nhân tạo để tự học và cải thiện các thuật toán máy tính, kết hợp các yếu tố xử lý được gọi là nơ-ron thông qua các trọng số liên kết. Mạng nơ-ron được xây dựng cho một chức năng cụ thể thông qua việc học từ dữ liệu đào tạo. Quá trình huấn luyện là sự điều chỉnh trọng số liên kết giữa các nơ-ron.

Mô hình mạng nơ-ron thường được sử dụng rộng rãi nhất là mô hình mạng truyền thẳng nhiều lớp (Multi-layers Perceptron- MLP). Một mạng MLP tổng quát có n lớp ($n \geq 2$) trong đó bao gồm một lớp vào, một lớp ra và một hoặc nhiều lớp ẩn như Hình 4.

Hoạt động của mạng MLP như sau: Tại lớp đầu vào, các nơ-ron nhận tín hiệu vào xử lý (tính tổng trọng số, gửi tới hàm truyền) rồi cho ra kết quả (là kết quả của hàm truyền); kết quả này sẽ được truyền tới các nơ-ron thuộc lớp ẩn thứ nhất; các nơ-ron tại đây tiếp nhận như là tín hiệu đầu vào, xử lý và gửi kết quả đến lớp ẩn thứ 2… quá trình tiếp tục cho đến khi các nơ-ron thuộc lớp ra cho kết quả.



Hình 4. Kiến trúc mạng MLP 4 lớp

(Nguồn: https://dlapplications.github.io/2018-06-15-MLP)

Mạng MLP được sử dụng rất hiệu quả trong phương pháp học sâu. Tùy theo yêu cầu của bài toán mà tùy chỉnh số nơ-ron đầu vào, đầu ra và số lớp ẩn. Để có độ chính xác cao, tránh hiện tượng quá khớp (overfitting) thì số lượng lớp ẩn và số nơ-ron trên nó là yếu tố quyết định.

2. PHƯƠNG PHÁP THIẾT KẾ MÔ HÌNH

2.1 Tổng quan thiết kế mô hình

Bài viết này mô tả phương pháp thiết kế để xây dựng mô hình mạng nơ-ron sâu với các lớp ẩn để tự động tìm hiểu các đặc tính của dữ liệu trước khi phát hiện các hành vi xâm nhập. Các đặc tính được học từ mô hình này có thể làm tăng khả năng phân biệt các hành vi khác nhau. Mô hình sử dụng mạng nơ-ron dùng để học có giám sát, học cách biểu diễn cho một tập các dữ liệu thông thường với mục đích giảm chiều dữ liệu, giúp dự đoán đầu ra từ một đầu vào ban đầu.

Hình 5 mô tả các bước áp dụng phương pháp học sâu để phát hiện tấn công và xâm nhập mạng. Tất cả các bước được mô tả cụ thể như sau: Bước đầu bao gồm tập dữ liệu thô đầu vào, qua các bước tiền xử lý dữ liệu, chia tập dữ liệu gốc ban đầu thành hai tập dữ liệu mới, một tập dùng để huấn luyện (training) và một tập dùng để kiểm thử (testing), dùng các thuật toán học sâu để huấn luyện ra mô hình (model), cuối cùng là đưa ra kết quả đánh giá tổng thể về tập dữ liệu.

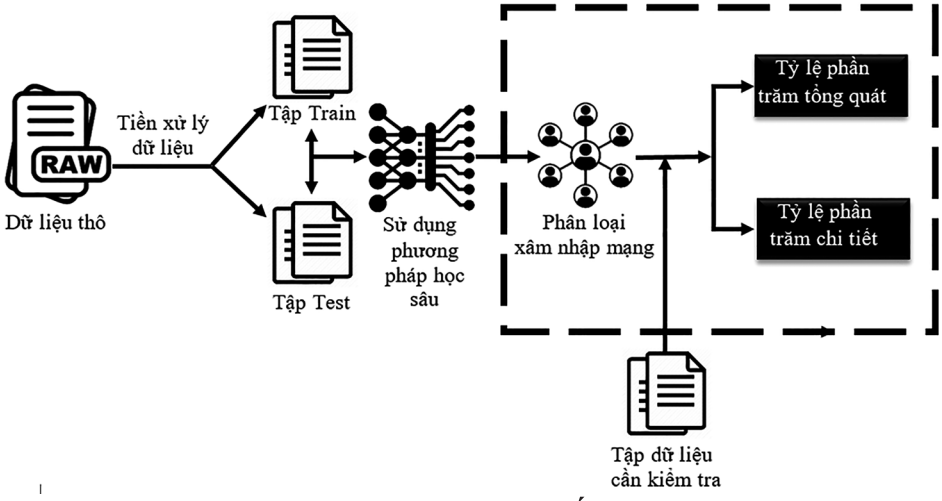
Mô hình sẽ được huấn luyện để học các đặc tính dữ liệu dựa trên mạng nơ-ron nhiều lớp. Các đặc tính đã được học sẽ được mô hình phân loại để phát hiện các hành vi xâm nhập. Mô hình phát hiện xâm nhập mạng này bao gồm hai giai đoạn:

Giai đoạn 1: Huấn luyện cho việc học và phân loại đặc tính. Việc huấn luyện mô hình học các đặc tính được thực hiện bằng phương pháp học có giám sát. Mô hình được huấn luyện phù hợp có thể được áp dụng trực tiếp để tìm hiểu đặc tính của dữ liệu mới. Tiền xử lý dữ liệu trong giai đoạn này mục đích để chuẩn hóa dữ liệu đầu vào.

Giai đoạn 2: Phát hiện các hành vi xâm nhập mạng. Đầu tiên cũng cần phải tiền xử lý dữ liệu đầu vào, sau đó cho mô hình đã học các đặc tính được huấn luyện trong giai đoạn 1 được sử dụng để tìm hiểu các đặc tính của dữ liệu này. Cuối cùng, các đặc tính này được nhập vào trình phân loại để dự đoán là xâm nhập hay bình thường.

2.2 Thiết kế kiến trúc mô hình

Mô hình học sâu sử dụng mạng nơ-ron MLP với *1* lớp đầu vào, *1* lớp đầu ra và *2* lớp ẩn. Do mô hình ở đây là học có giám sát với dữ liệu đầu vào là một vectơ có kích thước cố định nên mạng nơ-ron MLP (đầu ra của một lớp là đầu vào của lớp kế tiếp) là mô hình thích hợp. Mô hình được thiết kế với những lựa chọn như Hình 6

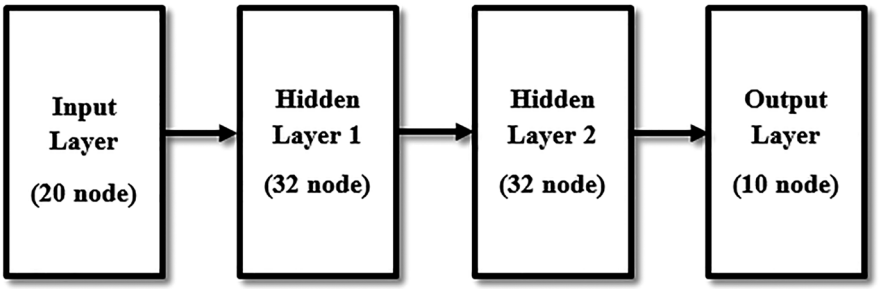


Hình 5. Phương pháp học sâu phát hiện tấn công xâm nhập mạng

Lớp đầu vào (Input Layer): Số node đầu vào là số thuộc tính của đối tượng cần phân lớp. Ở đây tập dữ liệu đầu vào có *20* thuộc tính nên chọn số node là *20*.

Lớp đầu ra (Output Layer): Số node ra là số đặc tính cần hướng tới của đối tượng (giá trị học có giám sát). Số node đầu ra ở đây tương ứng từ 0 đến 9 là *10* nodes (*9* thuộc tính tấn công và *1* thuộc tính bình thường "Normal").

Các lớp ẩn (Hidden Layers): Số node ẩn không xác định trước được, thường là do kinh nghiệm của người thiết kế mạng, nếu số node ẩn quá nhiều mạng sẽ cồng kềnh phức tạp, quá trình học sẽ chậm, còn nếu số node ẩn quá ít sẽ học không chính xác. Qua nhiều lần nghiên cứu thiết kế cài đặt số lớp ẩn và số node khác nhau, cuối cùng đã lựa chọn được mô hình có *2* lớp ẩn, mỗi lớp có *32* nodes là tối ưu nhất.



Hình 6. Mô hình học sâu được đề xuất

Hàm kích hoạt (Activation Functions): Được sử dụng để tính toán cho đầu ra của mỗi lớp mạng nơ-ron. Mô hình sử dụng các hàm kích hoạt:

Hàm ReLU: f(s) = max(0,s) cho đầu ra của lớp input và các lớp ẩn.

Hàm Softmax cho đầu ra lớp ouput. Công thức hàm Softmax như sau:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \qquad (3.1)$$

Trong đó: z là vector đầu vào liền kề với lớp đầu ra, K là tổng số các node mạng nơ-ron của đầu ra.

2.3 Tập dữ liệu huấn luyện

Bài viết này sử dụng bộ dữ liệu UNSW-NB15 (Moustafa et al., 2017) [6] được tạo bởi phòng thí nghiệm của Trung tâm An ninh mạng Úc (ACCS) để huấn luyện cho mô hình 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 tác giả Zoghi et al. (2021) [7] 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. 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 và thuộc tính *proto* được chuyển sang giá trị số theo Assigned Numbers Internet Protocol.

Chi tiết về các kiểu tấn công có trong tập dữ liệu UNSW-NB15 được mô tả trong Bảng 2.

Bảng 1. Các đặc tính (features) của tập dữ liệu UNSW-NB15

#	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

Bảng 2. Các kiểu tấn công trong tập dữ liệu UNSW-NB15

Loại tấn công	Mô tả
Normal	Gói tin mạng bình thường.
Denial of Service (DoS)	Kẻ tấn công gửi một lượng lớn yêu cầu làm hệ thống bị tràn ngập, mất kết nối dịch vụ.
Exploit	Tấn công khai thác vào lỗ hổng bảo mật của hệ thống hoặc phần mềm ứng dụng để khai thác lỗ hổng.
Worm	Tấn công vào điểm đầu cuối hoặc những lỗ hổng đã có sẵn. Worms tự động sao chép chính nó và xâm nhập đến hệ thống khác thông qua mạng máy tính.
Fuzzers	Kẻ tấn công gửi một lượng lớn dữ liệu đầu vào không hợp lệ hoặc bán hợp lệ nhằm mục đích phát hiện ra các lỗ hổng bảo mật trong mạng.
Backdoors	Tấn công cửa sau là một kỹ thuật bỏ qua các xác thực bảo mật của hệ thống mục tiêu và cho phép kẻ tấn công giành được quyền truy cập trái phép vào hệ thống.
Generic	Tấn công dựa trên một kỹ thuật thiết lập chống lại mọi mật mã bằng cách sử dụng hàm băm để gây ra xung đột mà không liên quan đến cấu hình của mật mã.
Shellcode	Là một phần con của Exploit, trong đó kẻ tấn công sử dụng một đoạn mã nhỏ để xâm nhập kiểm soát hệ thống.
Reconnaissance	Tấn công thăm dò để thu thập thông tin nhằm xác định lỗ hổng của một mạng máy tính.
Analysis	Là loại xâm nhập vào các ứng dụng web bằng cách thực hiện các hoạt động như quét cổng (port-scan), email (spam), web scripts (html files).

Mạng nơ-ron MLP chỉ cho phép các đặc tính hay thuộc tính của hành vi người dùng được thể hiện dưới dạng giá trị số ở lớp đầu vào. Do vậy cần phải xử lý dữ liệu đầu vào bằng cách sử dụng vectơ để chuyển tất cả chuỗi và số cho về giá trị số là [0,1]. Chuyển đổi mỗi thuộc tính trên thành số bằng cách sử dụng mô-đun tiền xử lý của thư viện Sklearn. Đầu tiên, sử dụng mô-đun *preprocessing.LabelEncoder()* để đảm bảo rằng trạng thái sẽ được mã hóa theo trạng thái danh sách. Sau đó, chúng ta chuyển đổi giao thức là kiểu chuỗi sang giá trị số bằng cách sử dụng Assigned Numbers Internet Protocol. Việc đánh giá độ chính xác (accuracy) của bộ phân lớp rất quan trọng, bởi vì nó cho phép dự đoán được độ chính xác của các kết quả phân lớp những dữ liệu tương lai. Độ chính xác còn giúp so sánh các mô hình phân lớp khác nhau. Phương pháp đánh giá Hold-out (Yadav & Shukla, 2016) [8] được sử dụng trong bài viết này 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 3.

Bảng 3. Số lượng dữ liệu dùng để huấn luyện và kiểm thử 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

2.4 Tiến hành thiết kế

Mô hình được huấn luyện sử dụng thư viện Keras (Chollet et al., 2015) [9] có *4* lớp. Lớp đầu vào gồm *20* nodes và hai lớp ẩn, mỗi lớp ẩn gồm *32* nodes sử dụng hàm kích hoạt ReLU. Lớp đầu ra gồm *10* nodes với hàm kích hoạt Softmax. Đầu tiên khởi tạo cho quá trình huấn luyện mô hình là sử dụng trình tối ưu hóa Adam với tốc độ học (learning_rate) *0,01*. Hàm lỗi (loss function) là *sparse_categorical_crossentropy*. Số lần mô hình "học" qua tất cả các dữ liệu trong tập huấn luyện (epochs) là *100* với số lượng mẫu huấn luyện được gửi đến mô hình cùng một lúc (batch_size) là *32*.

3. KẾT QUẢ VÀ THẢO LUẬN

Huấn luyện mô hình là quá trình thay đổi các tham số (epoch, batch-size, learning rate,…) để điều chỉnh mô hình đạt được kết quả tốt nhất. Mô hình được huấn luyện càng nhiều thì giá trị loss (độ lỗi) sẽ giảm xuống và tương ứng là giá trị accuracy (độ chính xác) sẽ tăng lên. Khi huấn luyện 70 lần *(Epochs=70)* độ chính xác đã tăng lên xấp xỉ *87,41%*, nhưng khi tăng số lần huấn luyện lên từ *80* đến *100* thì độ chính xác không cải thiện thêm mà chỉ dao động quanh *85-86%* do hiện tượng quá khớp (overfiting). Do đó, mô hình đạt độ chính xác tốt nhất là *87,41%* với *Epochs = 70, Batch-size = 2048* như Bảng 4.

Bảng 4. Huấn luyện mô hình với các tham số

Epoch	Batch-size	Accuracy train	Loss train
1	10	0.8401	0.4668
1	32	0.8418	0.4619
10	10	0.8530	0.4243
10	32	0.8622	0.4026
20	32	0.8629	0.3978
20	64	0.8637	0.3941
30	128	0.8655	0.3848
40	256	0.8667	0.3804
50	512	0.8680	0.3754
60	1024	0.8707	0.3364
70	2048	0.8741	0.3431
70	128	0.8661	0.3822
70	1024	0.8675	0.3751
70	4096	0.8668	0.3816
80	4096	0.8670	0.3787
80	2048	0.8673	0.3758
90	1024	0.8684	0.3768
90	64	0.8642	0.3916
100	32	0.8530	0.4220
100	128	0.8669	0.3798

Đánh giá mô hình:

Các tiêu chí được sử dụng để đánh giá hiệu quả mô hình bao gồm:

Accuracy: Là tỷ lệ số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm tra.

$$A = \frac{TP+TN}{TP+FN+FP+TN} \quad (3.2)$$

Recall: Là tỷ lệ số điểm True Positive (TP) trong số những điểm thực sự là Positive (TP+FN).

$$Recall = TPR = \frac{TP}{TP+FN} \quad (3.3)$$

Precision (P): Là thước đo một hệ thống có khả năng phát hiện bình thường hay tấn công.

$$P = \frac{TP}{TP+FP} \quad (3.4)$$

F1- score: Là *harmonic mean* của Precision và Recall, sử dụng để đánh giá bộ phân lớp.

$$F_1 = 2\frac{1}{\frac{1}{\text{precision}}+\frac{1}{\text{recall}}} = 2\frac{\text{precion}\cdot\text{recall}}{\text{precision+recall}} \quad (3.5)$$

Trong đó: TP (True Positives) là số lượng các tấn công được phân loại đúng. TN (True Negatives) là số lượng các bình thường được phân loại đúng. FP (False Positives) là số lượng các bình thường được phân loại sai. FN (False Negatives) là số lượng các tấn công được phân loại sai. Kết quả đánh giá mô hình của tiêu chí Precision ≈ *0.87796*. Tỷ lệ % của giá trị Precision là tỷ lệ số lượng tấn công được dự đoán đúng trên tổng số lượng bản ghi trong tập dữ liệu test. Precision ≈ *87.796%* thể hiện được hiệu quả của mô hình, tỷ lệ phát hiện tấn công xâm nhập mạng của mô hình đạt kết quả rất khả quan.

4. KẾT LUẬN VÀ ĐỀ NGHỊ

Bài viết đã giới thiệu ứng dụng của mạng nơ-ron nhân tạo và phương pháp học sâu trong việc xây dựng một mô hình phát hiện tấn công xâm nhập mạng, bên cạnh đó nâng cao kiến thức về an ninh mạng. Kết quả đánh giá mô hình cho thấy rằng giải pháp được đề xuất hoạt động hiệu quả và có tỷ lệ phát hiện tương đối cao. Hướng phát triển của nghiên cứu là ứng dụng mô hình này vào thực tế để xây dựng một hệ thống có khả năng phát hiện tấn công xâm nhập mạng đạt hiệu quả cao hơn so với các công cụ truyền thống khác.

TÀI LIỆU THAM KHẢO

[1] Vinayakumar, R., & Alazab (2019). Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access*, 7, 41525–41550.https://doi.org/10.1109/ACCESS.2019.2895334.

[2] Albin, E., & Rowe, N. C. (2012). A Realistic Experimental Comparison of the Suricata and Snort Intrusion-Detection Systems. 2012 26th *Advanced Networking,* 122–127.

[3] 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.

[4] 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.

[5] Hall, J., Barbeau, M., & Kranakis, E. (2005). Anomaly-based intrusion detection using mobility profiles of public transportation users. *IEEE International Conference on Wireless And Mobile Computing, Networking And Communications*, 17-24. https://doi.org/10.1109/WIMOB.2005.1512845. Truy cập ngày 25/1/2023.

[6] Moustafa, N., Creech, G., & Slay, J. (2017). *Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models (pp. 127–156)*. Springer International Publishing. https://doi.org/10.1007/978-3-319-59439-2_5.

[7] Zoghi, Z., & Serpen, G. (2021). *UNSW-NB15 Computer Security Dataset: Analysis through Visualization*. ArXiv:2101.05067 [Cs]. http://arxiv.org/abs/2101.05067. Truy cập ngày 20/1/2023.

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

[9] Chollet, F. (2015). *Keras, GitHub*. https://github.com/keras-team/keras. Truy cập ngày 20/1/2023.