NÂNG CAO HIỆU QUẢ PHÁT HIỆN XÂM NHẬP MẠNG BẰNG HUẤN LUYỆN DSD

Huỳnh Trọng Thưa*, Nguyễn Hoàng Thành*

**Học viện Công nghệ Bưu chính Viễn thông Cơ sở tại Thành phố Hồ Chí Minh*

Tóm tắt: **Hầu hết các mô hình phát hiện xâm nhập hiện đại đều ứng dụng học máy để cho ra kết quả phát hiện và phân loại tấn công xâm nhập với độ chính xác cao. Nghiên cứu này đề xuất mô hình kết hợp mạng nơ-ron nhiều lớp với huấn luyện nhiều giai đoạn DSD để cải tiến đồng thời các tiêu chí liên quan đến hiệu quả thực thi của các hệ thống phát hiện xâm nhập trên tập dữ liệu UNSW-NB15, là tập được cập nhật thường xuyên các đặc trưng dữ liệu với nhiều hình thức tấn công mới. Chúng tôi tiến hành thực nghiệm trên 3 mô hình mạng nơ-ron RNN, LSTM, và GRU để đánh giá hiệu quả kết hợp với từng mô hình thông qua nhiều tiêu chí như độ chính xác, tỷ lệ phát hiện, tỷ lệ cảnh báo giả, Precision và F1-Score.**

Từ khóa: an ninh mạng, học máy, học sâu, IDS, mạng nơ-ron.

I. GIỚI THIỆU

Hệ thống phát hiện xâm nhập (Intrusion Detection System - IDS) là hệ thống giám sát lưu thông mạng, có khả năng nhận biết những hoạt động khả nghi hay những hành động xâm nhập trái phép trên hệ thống mạng trong tiến trình tấn công, từ đó cung cấp thông tin nhận biết và đưa ra cảnh báo cho hệ thống và nhà quản trị. Sự phát triển của mã độc (malware) đặt ra một thách thức quan trọng đối với việc thiết kế các hệ thống phát hiện xâm nhập (IDS). Các cuộc tấn công bằng mã độc đã trở nên tinh vi và nhiều thách thức hơn. Những kẻ tạo ra mã độc có khả năng sử dụng nhiều kỹ thuật khác nhau để che giấu hành vi và ngăn chặn sự phát hiện của IDS, từ đó chúng dễ dàng lấy cắp dữ liệu quan trọng cũng như phá hoại hoạt động sản xuất kinh doanh và cung cấp dịch vụ của nhiều cá nhân tổ chức.

IDS hoạt động theo ba phương thức chính gồm Signature-base, Abnormaly-base, và Stateful Protocol Analysis. Signature-base IDS so sánh các dấu hiệu của đối tượng quan sát với các dấu hiệu của các mối nguy hại đã biết. Abnormaly-base IDS so sánh định nghĩa của những hoạt động bình thường và đối tượng quan sát nhằm xác định các độ lệch để đưa ra cảnh báo. Stateful Protocol Analysis IDS so sánh các profile định trước về hoạt động của mỗi giao thức được coi là bình thường với đối tượng quan sát từ đó xác định độ lệch. Ngoại trừ phương thức Signature-base, hai phương thức còn lại rất cần phải học để nhận biết các dấu hiệu bất thường. Vì vậy mà trong vài thập kỷ qua, học máy đã được sử dụng để cải thiện phát hiện xâm nhập.

Hiệu quả của các IDS được đánh giá dựa trên hiệu suất thực thi của chúng trong việc xác định các cuộc tấn công. Điều này đòi hỏi một tập dữ liệu hoàn chỉnh với đầy đủ các hành vi bình thường và bất thường. Các tập dữ liệu chuẩn trước đây như KDDCUP 99 [1] và NSLKDD [2] đã được áp dụng rộng rãi để đánh giá khả năng thực thi của các IDS. Mặc dù hiệu quả của các tập dữ liệu này đã được ghi nhận trong nhiều nghiên cứu trước đó, việc đánh giá IDS bằng cách sử dụng các tập dữ liệu này không phản ánh đúng hiệu suất đầu ra thực tế do một vài lý do. Lý do đầu tiên là tập dữ liệu KDDCUP 99 chứa một số lượng lớn các bản ghi dư thừa trong tập huấn luyện. Các bản ghi dư thừa ảnh hưởng đến kết quả của các độ lệch bias trong phát hiện xâm nhập đối với các bản ghi thường xuyên. Thứ hai, nhiều bản ghi bị thiếu là một yếu tố thay đổi bản chất của dữ liệu. Thứ ba, tập dữ liệu NSLKDD là phiên bản cải tiến của KDDCUP 99, nó giải quyết một số vấn đề như mất cân bằng dữ liệu giữa các bản ghi bình thường và bất thường cũng như các giá trị bị thiếu [3].Tuy nhiên, tập dữ liệu này không phải là đại diện toàn diện cho môi trường tấn công thực tế hiện đại. Tập dữ liệu chuẩn UNSW-NB15 [4] được tạo ra nhằm giải quyết các hạn chế của các tập dữ liệu trước đó, đặc biệt là cập nhật thường xuyên các đặc trưng dữ liệu và bám sát các hình thức tấn công mới.

Đã có nhiều phương pháp học máy và mô hình mạng nơ-ron áp dụng vào phát hiện xâm nhập mạng như RNN, LSTM, GRU. Việc chọn mô hình phù hợp với bộ dữ liệu UNSW-NB15 để cải thiện kết quả đánh giá cũng là vấn đề đang được quan tâm [5]. Bên cạnh đó, một trong những nghiên cứu gần đây để cải tiến chất lượng huấn luyện [6], mô hình huấn luyện DSD (Dense Sparse Dense) đã được áp dụng hiệu quả vào một số bài toán xử lý hình ảnh, nhận dạng giọng nói. Trong nghiên cứu này, chúng tôi tập trung vào việc đánh giá hiệu quả phát hiện xâm nhập mạng dựa trên các mô hình mạng nơ-ron sâu như RNN, LSTM, GRU bằng cách đề xuất mô hình kết hợp phương pháp huấn luyện DSD vào từng mô hình mạng nơ-ron này để cải tiến hiệu quả thực thi cho các hệ thống phát hiện xâm nhập mạng.

Bài viết này có 6 phần, các phần còn lại được trình bày như sau. Phần II trình bày các nghiên cứu liên quan gồm mô hình học sâu và phương pháp huấn luyện DSD. Phần III trình bày mô hình kết hợp đề xuất. Phần IV và V trình bày thực nghiệm, kết quả và đánh giá các mô hình đề xuất. Phần VI sẽ kết luận cho nghiên cứu này.

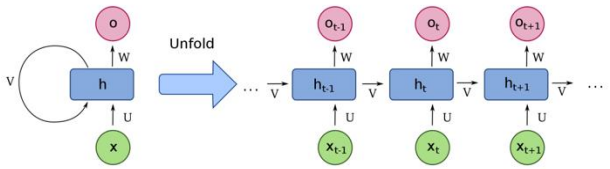
II. CÁC NGHIÊN CỨU LIÊN QUAN

Trong phần này, chúng tôi trình bày các mô hình học sâu liên quan và phương pháp huấn luyện DSD.

Tác giả liên hệ: Huỳnh Trọng Thưa,
Email: htthua@ptithcm.edu.vn
Đến tòa soạn: 08/2020, chỉnh sửa: 10/2020, chấp nhận đăng: 10/2020.

A. *Mạng nơ-ron hồi quy – Recurrent Neural Network (RNN)*

RNN [7] là một phần mở rộng của mạng nơ-ron chuyển tiếp thẳng, được thiết kế để nhận dạng các mẫu trong chuỗi dữ liệu. Các mạng nơ-ron được gọi là hồi quy vì chúng thực hiện cùng một nhiệm vụ cho mọi thành phần của chuỗi với đầu ra phụ thuộc vào các tính toán trước đó [8].



Hình 1. Kiến trúc RNN đơn giản

Có thể xem RNN là một cách để chia sẻ trọng số theo thời gian, như được minh họa trong Hình 1. Các phương trình (1) và (2) sau dùng để tính toán trạng thái $h_t$và đầu ra ẩn O_t theo RNN:

$$h_t = \sigma(W_i h_{t-1} + U x_t + b_t) \quad (1)$$

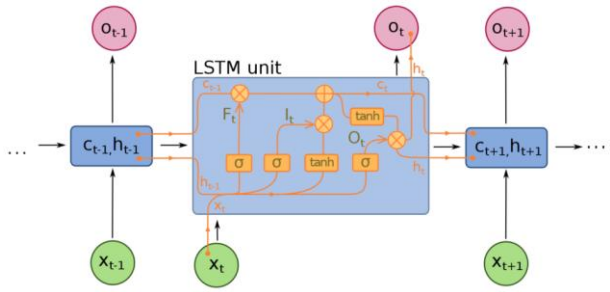
$$O_t = \tau(W_o h_t) \quad (2)$$

Trong đó σ và τ tương ứng là hàm kích hoạt *sigmoid* và *softmax*, x_t là một vectơ đầu vào tại thời điểm t, h_t là vectơ trạng thái ẩn ở thời điểm t, W_i là ma trận trọng số đầu vào, U là ma trận trọng số giữa các lớp ẩn, W_o là ma trận trọng số đầu ra và b_t là hệ số *bias*.

Mô hình RNN có một nhược điểm lớn, đó là tại mỗi thời điểm trong quá trình huấn luyện, các trọng số tương tự nhau được sử dụng để tính toán đầu ra O_t, điều này khiến cho kết quả tạo ra không còn chính xác. Mô hình Long Short-Term Memory (LSTM) và Gated Recurrent Unit (GRU) đã được đề xuất để giải quyết vấn đề này.

B. *Mạng Long Short-Term Memory (LSTM)*

Mạng bộ nhớ dài-ngắn LSTM (Long Short Term Memory) [7] là một biến thể của mạng nơ-ron hồi quy được đề xuất như một trong những kỹ thuật máy học để giải quyết nhiều vấn đề dữ liệu tuần tự. LSTM giúp duy trì lỗi có thể lan truyền ngược qua các lớp theo thời gian. LSTM được dùng để tăng độ chính xác của kết quả đầu ra, cũng như làm cho RNN trở nên hữu ích hơn dựa trên các tác vụ bộ nhớ dài hạn. Kiến trúc LSTM như được minh họa trong Hình 2, bao gồm bốn thành phần chính; cổng đầu vào *(i)*, cổng quên *(f)*, cổng đầu ra *(o)* và ô nhớ *(c)*.



Hình 2. Kiến trúc lõi của khối LSTM [7]

Khối LSTM đưa ra quyết định về việc lưu trữ, đọc và ghi thông qua các cổng mở hoặc đóng, và mỗi khối bộ nhớ tương ứng với một thời điểm cụ thể. Các cổng truyền thông tin dựa trên một tập các trọng số. Một số trọng số, như trạng thái đầu vào và ẩn, được điều chỉnh trong quá trình học. Các phương trình từ (3) đến (8) dùng để biểu diễn mối quan hệ giữa đầu vào và đầu ra tại thời điểm *t* trong kiến trúc khối LSTM:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (4)$$

$$j_t = \omega(W_j[h_{t-1}, x_t] + b_j) \quad (5)$$

$$c_t = f_t \times c_{t-1} + i_t \times j_t \quad (6)$$

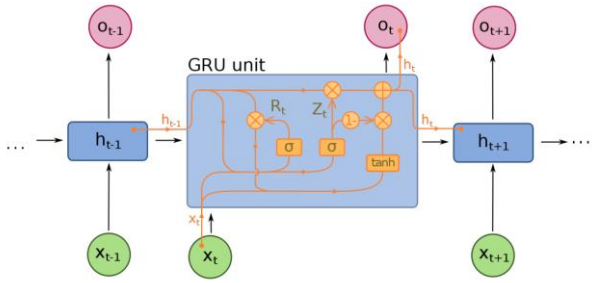
$$z_t = \sigma(W_j[h_{t-1}, x_t] + b_j) \quad (7)$$

$$h_t = z_t \times \omega(c_t) \quad (8)$$

Trong đó σ và ω tương ứng là hàm kích hoạt *sigmoid* và *tanh*, x_t là một vectơ đầu vào tại thời điểm t, h_t là vectơ đầu ra ở thời điểm t, W và b lần lượt là là ma trận trọng số và hệ số *bias*, f_t là hàm quên dùng để lọc bỏ các thông tin không cần thiết, i_t và j_t dùng để chèn thông tin mới vào ô nhớ, z_t xuất ra thông tin liên quan.

C. *Mạng Gated Recurrent Unit (GRU)*

GRU là một biến thể của LSTM được giới thiệu bởi K.Cho [8]. GRU về cơ bản là LSTM không có cổng đầu ra, do đó nó ghi tất cả nội dung từ bộ nhớ vào mạng lớn hơn ở từng thời điểm. Tuy nhiên nó được tinh chỉnh bằng cách sử dụng một cổng cập nhật được bổ sung vào trong cấu trúc khối GRU. Cổng cập nhật là sự kết hợp giữa cổng đầu vào và cổng quên. Mô hình GRU được đề xuất để đơn giản hóa kiến trúc của mô hình LSTM. Cấu trúc của GRU được thể hiện trong Hình 3 và các phương trình từ (9) đến (12) thể hiện mối quan hệ giữa đầu vào và kết quả dự đoán.



Hình 3. Kiến trúc lõi của khối GRU [7]

$$v_t = \sigma(W_v[o_{t-1}, x_t] + x_t) \quad (9)$$

$$s_t = \sigma(W_v[o_{t-1}, x_t]) \quad (10)$$

$$o'_t = \omega(W_v[s_t \times o_{t-1}, x_t]) \quad (11)$$

$$o_t = (1 - v_t) \times o_{t-1} + v_t \times o'_t \quad (12)$$

Trong đó không gian đặc trưng (đầu vào) được đại diện bởi *x* và dự đoán được đại diện bởi o_t, v_t là hàm cập nhật. W là trọng số được tối ưu hóa trong quá trình huấn luyện. σ và ω tương ứng là hàm kích hoạt *sigmoid* và *tanh* để giữ cho thông tin đi qua GRU trong một phạm vi cụ thể.

D. *Mô hình huấn luyện Dense Sparse Dense (DSD)*

Các mô hình nơ-ron nhiều lớp phức tạp cho kết quả tốt và có thể thu được các quan hệ phi tuyến cao giữa các

dữ liệu đặc trưng và đầu ra. Nhược điểm của các mô hình lớn này là chúng dễ bị nhiễu trong tập dữ liệu huấn luyện. Việc này dẫn đến tình trạng quá khớp (over-fitting) [9] và phương sai cao (high variance) [11].

Bảng I. Cấu hình của mô hình RNN đề xuất

Lớp (Kiểu)	Dạng đầu ra	# tham số
rnn_1 (SimpleRNN)	(None, None, 128)	21.888
drop_out_1 (Dropout)	(None, None, 128)	0
rnn_2 (SimpleRNN)	(None, None, 128)	32.896
drop_out_2 (Dropout)	(None, None, 128)	0
rnn_3 (SimpleRNN)	(None, None, 128)	32.896
drop_out_3 (Dropout)	(None, None, 128)	0
rnn_4 (SimpleRNN)	(None, 128)	32.896
drop_out_4 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129
activation_1(Activation)	(None, 1)	0
Tổng số tham số		**120.705**

Tuy nhiên, nếu đưa mô hình trở về dạng ít phức tạp sẽ khiến hệ thống máy học có thể bỏ lỡ các quan hệ liên quan giữa các đặc trưng và đầu ra, dẫn đến vấn đề under-fitting [9] và bias cao (high bias) [10]. Đây là vấn đề vô cùng thách thức vì độ lệch bias và phương sai rất khó để tối ưu hóa cùng một lúc.

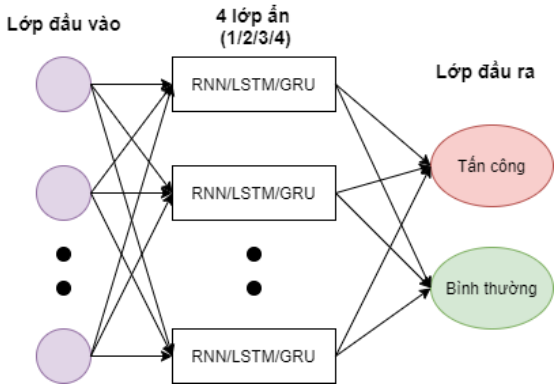
Trong nghiên cứu [6], Song và các đồng sự giới thiệu DSD, một mô hình huấn luyện "dày đặc – thưa thớt – dày đặc" bằng cách lựa chọn các kết nối để loại bỏ và phục hồi các kết nối khác sau đó. Nhóm tác giả đã thử nghiệm mô hình DSD của mình với GoogLeNet, VGGNet và ResNet trên tập dữ liệu ImageNet, NeuralTalk BLEU trên tập dữ liệu Flickr-8K, và DeepSpeech-1&2 trên tập dữ liệu WSJ'93. Kết quả thực nghiệm cho thấy mô hình huấn luyện DSD đã mang lại hiệu quả đáng kể trên các tập dữ liệu xử lý ảnh và nhận dạng giọng nói được áp dụng trên các mạng rơ-ron sâu.

III. MÔ HÌNH MẠNG NƠ-RON KẾT HỢP

Nhận thấy rằng các mạng nơ-ron hồi quy (RNN/LSTM/GRU) có thể học hiệu quả để tạo ra các quan hệ phi tuyến cao giữa các đặc trưng đầu vào và đầu ra, nghiên cứu này đề xuất một mô hình kết hợp RNN/LSTM/GRU với phương pháp huấn luyện 3 giai đoạn DSD để nâng cao hiệu quả phát hiện xâm nhập mạng. Chúng tôi đặt số lượng nơ-ron của tất cả các lớp ẩn là 128. Mô hình đề xuất gồm:

- Mô hình mạng nơ-ron RNN/LSTM/GRU gốc 4 lớp ẩn được thể hiện trong Hình 4;
- Mô hình huấn luyện lai DSD-RNN/DSD-LSTM/DSD-GRU sử dụng một quá trình ba giai đoạn: dày đặc (Dense - D), thưa thớt (Sparse - S), tái dày đặc (reDense - D) được áp dụng trên mô hình mạng nơ-ron gốc 4 lớp ẩn. Các giai đoạn được minh họa trong Hình 5;

Cấu hình đề nghị của các mô hình tương ứng được mô tả trong các Bảng I, II và III.



Hình 4. Kiến trúc RNN/LSTM/GRU gốc với 4 lớp ẩn

Bảng II. Cấu hình của mô hình LSTM đề xuất

Lớp (Kiểu)	Dạng đầu ra	# tham số
lstm_1 (LSTM)	(None, None, 128)	87.552
drop_out_1 (Dropout)	(None, None, 128)	0
lstm_2 (LSTM)	(None, None, 128)	131.584
drop_out_2 (Dropout)	(None, None, 128)	0
lstm_3 (LSTM)	(None, None, 128)	131.584
drop_out_3 (Dropout)	(None, None, 128)	0
lstm_4 (LSTM)	(None, 128)	131.584
drop_out_4 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129
activation_1(Activation)	(None, 1)	0
Tổng số tham số		**482.433**

Huấn luyện thưa thớt chuẩn hóa mô hình và huấn luyện dày đặc sau cùng khôi phục các trọng số đã được cắt tỉa (màu đỏ), giúp tăng hiệu suất mô hình mà không bị overfitting. Ngoài ra, để đánh giá đúng hiệu quả của việc kết hợp, chúng tôi áp dụng kỹ thuật Cross Validation [11] cho từng mô hình trên tập dữ liệu UNSW-NB15.

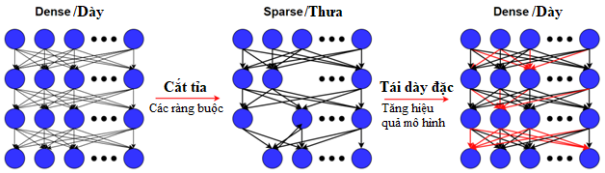
Bảng III. Cấu hình của mô hình GRU đề xuất

Lớp (Kiểu)	Dạng đầu ra	# tham số
gru_1 (GRU)	(None, None, 128)	65.644
drop_out_1 (Dropout)	(None, None, 128)	0
gru_2 (GRU)	(None, None, 128)	98.688
drop_out_2 (Dropout)	(None, None, 128)	0
gru_3 (GRU)	(None, None, 128)	98.688
drop_out_3 (Dropout)	(None, None, 128)	0
gru_4 (GRU)	(None, 128)	98.688
drop_out_4 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129
activation_1(Activation)	(None, 1)	0
Tổng số tham số		**361.857**

IV. THỰC NGHIỆM

A. Mô tả tập dữ liệu

Để đánh giá được hiệu quả của các IDS cần có một bộ dữ liệu xâm nhập chuẩn. Các bộ dữ liệu này rất quan trọng để thử nghiệm và đánh giá các phương pháp phát hiện xâm nhập. Chất lượng của dữ liệu thu thập ảnh hưởng đến hiệu quả của các kỹ thuật phát hiện bất thường. Một trong số tập dữ liệu chuẩn được dùng rộng rãi là KDD Cup 1999 [1] và NLS-KDD [2]. Tuy nhiên, chúng vẫn tồn tại một số hạn chế nhất định. Tập dữ liệu chuẩn UNSW-NB15 [4] được tạo ra nhằm giải quyết các thách thức trên. Các gói mạng thô của tập dữ liệu UNSW-NB15 được tạo bởi công cụ IXIA PerfectStorm trong Phòng thí nghiệm Cyber Range của Trung tâm An ninh mạng Úc (ACCS) để tạo ra một hỗn hợp các hoạt động bình thường thực sự và các hành vi tấn công tổng hợp hiện đại. Tập dữ liệu đầy đủ chứa tổng cộng 2.540.044 bản ghi.



Hình 5. Mô hình huấn luyện lại DSD-RNN/DSD-LSTM /DSD-GRU với 3 giai đoạn

Tập dữ liệu được rút gọn được mô tả trong Bảng IV chỉ có 44 đặc trưng kèm theo nhãn phân lớp, loại bỏ 6 đặc trưng (*dstip, srcip, sport, dsport, Ltime và Stime*) khỏi tập dữ liệu đầy đủ.

Trong Bảng IV, *dur* là thông tin về tổng thời gian; *proto* là giao thức; *service* là là loại dịch vụ (http, ftp, smtp, ssh,...); *state* biểu thị trạng thái giao thức; *spkts* là số gói từ nguồn đến đích; *dpkts* là số gói từ đích đến nguồn; *sbytes* là số byte từ nguồn tới đích; *dbytes* là số bytes từ đích đến nguồn; *sttl* là giá trị TTL từ nguồn tới đích; *dttl* là giá trị TTL từ đích tới nguồn; *sload* là số bit nguồn; *dload* là số bit đích; *sloss* là các gói nguồn được truyền lại hoặc bị loại bỏ; *dloss* là các gói đích được truyền lại hoặc bị loại bỏ (mili giây); *sinpkt* là thời gian đến giữa các gói nguồn (mili giây); *dinpkt* là thời gian đến giữa các gói đích; *sjit* là jitter nguồn (mili giây); *djit* là jitter đích (mili giây); *swin* là giá trị của kích thước TCP quảng bá nguồn; *stcpb* là số thứ tự của TCP nguồn; *is_sm_ips_ports* cho biết nếu địa chỉ IP nguồn và đích bằng nhau và số cổng nguồn và đích bằng nhau thì biến này nhận giá trị 1 ngược lại biến này nhận giá trị 0; *dtcpb* là số thứ tự của TCP đích; *dwin* là giá trị của kích thước TCP quảng bá đích; *tcprtt* là thời gian khứ hồi thiết lập kết nối TCP, là tổng của synack và ackdat; *synack* là thời gian thiết lập kết nối TCP giữa gói SYN và SYN_ACK; *ackdat* là thời gian thiết lập kết nối TCP, là thời gian giữa SYN_ACK và các gói ACK; *smean* là giá trị trung bình kích thước gói luồng được truyền bởi *src*; *dmean* là giá trị trung bình kích thước gói luồng được truyền bởi *dst*; *trans_depth* đại diện cho chiều sâu liên kết trong kết nối của giao dịch yêu cầu/phản hồi http; *response_body_len* là kích thước nội dung thực tế không nén của dữ liệu được truyền từ dịch vụ http máy chủ; *ct_srv_src* là số lượng kết nối chứa cùng một dịch vụ và địa chỉ nguồn trong 100 kết nối gần đây nhất; *ct_state_ttl* là số cho mỗi trạng thái theo phạm vi giá trị cụ thể cho TTL nguồn/đích; *ct_dst_ltm* là số kết nối có cùng địa chỉ đích trong 100 kết nối theo lần gần đây nhất; *ct_src_dport_ltm* là số kết nối có cùng địa chỉ nguồn và cổng đích trong 100 kết nối theo lần cuối cùng; *ct_dst_sport_ltm* là là số kết nối có cùng địa chỉ đích và cổng nguồn trong 100 kết nối theo lần cuối cùng; *is_ftp_login* cho biết nếu phiên ftp được truy cập bởi người dùng và mật khẩu thì nhận vào giá trị 1, ngược lại thì nhận giá trị 0; *ct_ftp_cmd* là số luồng có lệnh trong phiên ftp; *ct_flw_http_mthd* là số luồng có các phương thức như Get và Post trong dịch vụ http; *ct_src_ltm* là số kết nối có cùng địa chỉ nguồn trong 100 kết nối theo lần gần đây nhất; *ct_srv_dst* là số lượng kết nối có cùng dịch vụ và địa chỉ đích trong 100 kết nối theo lần gần đây nhất; *attack_cat* là tên của từng loại tấn công. Trong dữ liệu này, tập hợp chín loại; *label* nhận giá trị 0 cho bình thường và 1 cho các bản ghi tấn công.

Bảng IV. Mô tả đặc trưng tập dữ liệu rút gọn UNSWNB-15

#	Đặc trưng	Kiểu	#	Đặc trưng	Kiểu
1	dur	float	23	dtcpb	integer
2	proto	nominal	24	dwin	integer
3	service	nominal	25	tcprtt	float
4	state	nominal	26	synack	float
5	spkts	integer	27	ackdat	float
6	dpkts	integer	28	smean	integer
7	sbytes	integer	29	dmean	integer
8	dbytes	integer	30	trans_depth	integer
9	rate	float	31	response_body_len	integer
10	sttl	integer	32	ct_srv_src	integer
11	dttl	integer	33	ct_state_ttl	integer
12	sload	float	34	ct_dst_ltm	integer
13	dload	float	35	ct_src_dport_ltm	integer
14	sloss	integer	36	ct_dst_sport_ltm	integer
15	dloss	integer	37	ct_dst_src_ltm	integer
16	sinpkt	float	38	is_ftp_login	binary
17	dinpkt	float	39	ct_ftp_cmd	integer
18	sjit	float	40	ct_flw_http_mthd	integer
19	djit	float	41	ct_src_ltm	integer
20	Swin	integer	42	ct_srv_dst	integer
21	stcpb	integer	43	attack_cat	nonimal
22	is_sm_ips_ports	binary	44	label	binary

Một phần của tập dữ liệu đầy đủ được chia thành tập huấn luyện và tập kiểm thử, cụ thể là *UNSW_NB15_training-set.csv* và *UNSW_NB15_testing-set.csv*. Tập dữ liệu huấn luyện bao gồm 175.341 bản ghi trong khi tập dữ liệu kiểm thử chứa 82.332 bản ghi. Số lượng đặc trưng trong tập dữ liệu rút gọn khác với số lượng đặc trưng trong tập dữ liệu đầy đủ.

B. Tiền xử lý dữ liệu

1) Chuyển đổi đặc trưng nominal thành dạng số

Quá trình chuyển đổi đặc trưng nominal thành dạng số còn gọi là Numericalization. Trong bộ dữ liệu rút gọn UNSW-NB15 có 40 đặc trưng dạng numeric và 4 đặc trưng nominal. Vì giá trị đầu vào của RNN, LSTM, GRU phải là một ma trận số nên chúng tôi phải chuyển đổi một số đặc trưng nominal, chẳng hạn như các đặc trưng "proto", "service" và "state" thành dạng số. Chúng tôi chuyển các bộ số này bằng thư viện scikit-sklearn LabelEncoder [12]. Chẳng hạn, đặc trưng *proto* trong tập dữ liệu có các giá trị không trùng gồm *tcp, udp, rdp* thì sẽ được mã hóa nhãn tương ứng thành số 1, 2, 3.

Tập dữ liệu rút gọn chứa 10 loại, một loại là *normal* và 9 loại tấn công gồm: *generic, exploits, fuzzers, DoS, reconnaissance, analysis, backdoor, shellcode* và *worms*. Bảng V cho thấy chi tiết phân loại lớp của tập dữ liệu rút gọn UNSW-NB15.

2) Chuẩn hóa dữ liệu Min-Max

Chuẩn hóa là một kỹ thuật chia tỷ lệ trong đó các giá trị được dịch chuyển và định cỡ lại sao cho chúng kết thúc trong khoảng từ 0 đến 1. Chuẩn hóa đòi hỏi chúng ta phải biết hoặc có thể ước tính chính xác các giá trị tối thiểu và tối đa có thể quan sát được.

Do phạm vi giá trị của dữ liệu thô rất rộng và khác nhau, trong một số thuật toán học máy, các hàm mục tiêu sẽ không hoạt động đúng nếu không được chuẩn hóa. Một lý do khác cho việc chuẩn hóa đặc trưng được áp dụng là độ chính xác dự đoán sẽ tăng lên so với không có chuẩn hóa [13].

Bảng V. Phân loại tấn công trong tập dữ liệu rút gọn UNSW-NB15

Phân loại	Tập dữ liệu huấn luyện	Tập dữ liệu kiểm thử
	UNSW_NB_training-set	UNSW_NB_testing-set
Normal	56.000	37.000
Generic	40.000	18.871
Exploits	33.393	11.132
Fuzzers	18.184	6.062
DoS	12.264	4.089
Reconnaissance	10.491	3.496
Analysis	2.000	677
Backdoor	1.746	583
Shellcode	1.133	378
Worms	130	44
Tổng cộng	175.341	82.332

Trong thực nghiệm này, chúng tôi sử dụng chuẩn hóa Min-Max được cho bởi phương trình (13):

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad (13)$$

3) Các phương pháp đánh giá

Chúng tôi sử dụng 5 metric phổ biến để đánh giá hiệu suất phát hiện xâm nhập bao gồm độ chính xác (Accuracy), tỷ lệ phát hiện (Detection Rate), Precision, tỷ lệ cảnh báo giả (False Alarm Rate) và F1-score. Bảng VI thể hiện ma trận nhầm lẫn bao gồm dương tính thật (TP), âm tính thật (TN), dương tính giả (FP) và âm tính giả (FN). TP và TN chỉ ra rằng trạng thái bị tấn công và trạng thái bình thường được phân loại chính xác. FP chỉ ra rằng một bản ghi bình thường được dự đoán không chính xác, nghĩa là IDS cảnh báo một cuộc tấn công không đúng thực tế. FN chỉ ra rằng một bản ghi tấn công được phân loại không chính xác, nghĩa là IDS không cảnh báo gì mà ghi nhận đây vẫn là bản ghi bình thường.

Bảng VI. Ma trận nhầm lẫn

	Dự đoán - Tấn công	Dự đoán - Bình thường
Thực tế - Tấn công	TP	FN
Thực tế - Bình thường	FP	TN

Độ chính xác (Accuracy) – là mức độ gần của các phép đo với một giá trị cụ thể, thể hiện số lượng các trường hợp dữ liệu được phân loại chính xác trên tổng số dự đoán. Độ chính xác có thể không phải là thước đo tốt nếu tập dữ liệu không được cân bằng (cả hai lớp âm và dương có số lượng dữ liệu khác nhau). Công thức tính độ chính xác được định nghĩa trong công thức (14).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (14)$$

Tỷ lệ cảnh báo giả (False Alarm Rate - FAR) - còn được gọi là **False Positive Rate** (tỷ lệ dương tính giả). Thước đo này được tính theo công thức (15). Tỷ lệ lý tưởng cho thước đo này càng thấp càng tốt, tức là số phân loại nhầm một phân loại bình thường sang dự đoán tấn công (FP) càng thấp càng tốt.

$$FAR = \frac{FP}{FP + TN} \quad (15)$$

Độ chính xác phép đo (Precision) – là mức độ gần của các phép đo, có giá trị gần với 1 khi kết quả là một tập phân loại tốt. Precision là 1 chỉ khi tử số và mẫu số bằng nhau (TP = TP + FP), điều này cũng có nghĩa là FP bằng 0. Khi FP tăng giá trị dẫn đến mẫu số lớn hơn tử số và giá trị chính xác giảm. Công thức tính Precision được định nghĩa trong công thức (16).

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

Tỷ lệ phát hiện (Detection Rate – DR hay Recall) – Giá trị DR càng gần với 1 sẽ cho một phân loại tốt. DR là 1 chỉ khi tử số và mẫu số bằng nhau (TP = TP + FN), điều này cũng có nghĩa là FN bằng 0. Khi FN tăng giá trị dẫn đến mẫu số lớn hơn tử số và giá trị DR giảm. Chỉ số này nhằm đánh giá mức độ tổng quát hóa mô hình tìm được và được xác định theo công thức (17).

$$Detection\ Rate = \frac{TP}{TP + FN} \quad (17)$$

Chúng ta luôn mong muốn cả Precision và DR đều tốt, nghĩa là một trong hai giá trị FP và FN phải gần bằng

0 càng tốt. Do đó, chúng ta cần một tham số đo có tính đến cả Precision và DR, đó chính là F1-Score, được xác định theo công thức (18).

F1-Score được gọi là một trung bình điều hòa (harmonic mean) của các tiêu chí Precision và DR. Nó có xu hướng lấy giá trị gần với giá trị nào nhỏ hơn giữa 2 giá trị Precision và DR và đồng thời nó có giá trị lớn nếu cả 2 giá trị Precision và DR đều lớn. Chính vì thế F1-Score thể hiện được một cách khách quan hơn hiệu quả của một mô hình học máy. So với độ chính xác (Accuracy), F1-Score phù hợp hơn để đánh giá hiệu suất phát hiện của các mẫu dữ liệu không cân bằng.

$$F1 - Score = \frac{2(Precision \times DR)}{Precision + DR} = \frac{2TP}{2TP + FP + FN} \quad (18)$$

V. KẾT QUẢ VÀ ĐÁNH GIÁ

Trong phạm vi nghiên cứu này, mô hình phân loại nhị phân dựa trên mạng RNN, LSTM, GRU được lựa chọn. Mô hình này được huấn luyện trên bộ dữ liệu rút gọn UNSW-NB15. Thuật toán được xây dựng trên ngôn ngữ Python và thư viện Keras, Sklearn, chạy trên nền tảng Tensorflow và môi trường của Anaconda.

Thử nghiệm được thực hiện trên laptop Acer Nitro5, có cấu hình CPU Intel Core i5-9300 2.4 GHz, bộ nhớ 16 GB và GPU 3 Gibytes. Thực nghiệm đã được thiết kế để nghiên cứu hiệu suất giữa 3 mô hình mạng nơ-ron RNN, LSTM và GRU trong phân loại nhị phân (bình thường, bất thường).

Cả ba mô hình RNN, LSTM, GRU đều được huấn luyện với số *epoch* là 100, trong quá trình compile mô hình với thông số hàm optimize là Adam với learning rate theo mặc định (0.001), batch_size là 256, hàm *loss* là *binary_crossentropy*, tiêu chí tối ưu là theo độ chính xác (accuracy).

Các mô hình RNN, LSTM, GRU cơ bản giống nhau, chỉ khác nhau về kiến trúc lõi của nơ-ron, cụ thể:

- Các lớp RNN/LSTM/GRU gồm 128 đơn vị nơ-ron;
- Các lớp Dropout với hệ số là 0.1;
- Lớp Dense với hàm kích hoạt là sigmoid.

Trong đó, lớp Dropout giúp tránh tình trạng overfitting, lớp Dense cuối cùng là lớp đầu ra để đánh giá đầu ra là 1 hay 0, tức là bất thường hay bình thường.

Mỗi pha đều áp dụng DSD cho cả 3 mô hình DSD-RNN/DSD-LSTM/DSD-GRU. Tuy nhiên ở pha cuối cùng là pha tái dày đặc khôi phục kết nối thì tỷ lệ học được giảm xuống còn 0.0001.

A. Kết quả

Thử nghiệm mô hình huấn luyện được thực hiện với 2 trường hợp sau:

Trường hợp 1: Mô hình RNN, LSTM, GRU trên tập dữ liệu rút gọn UNSW-NB15.

Trường hợp 2: Mô hình RNN, LSTM, GRU trên tập dữ liệu rút gọn UNSW-NB15 kết hợp với mô hình huấn luyện DSD.

Các trường hợp trên đều được huấn luyện với Cross validation [11]. Kết quả đánh giá được trình bày trong Bảng VII (trường hợp 1) và Bảng VIII (trường hợp 2).

Bảng VII. Đánh giá Mô hình RNN, LSTM, GRU trên bộ dữ liệu rút gọn UNSW-NB15

	FAR%	Acc%	Prec%	DR%	F1-S%
RNN	34.17	83.54	77.87	**98.01**	86.78
LSTM	**33.37**	**83.70**	**78.23**	97.95	**86.85**
GRU	34.22	82.90	77.67	96.88	86.19

Kết quả thực nghiệm ba mô hình RNN, LSTM và GRU như trong Bảng VII cho thấy mô hình mạng nơ-ron LSTM có kết quả tốt nhất với Accuracy, Precision, F1 Score có chỉ số cao nhất, tỷ lệ dương tính giả FAR thấp nhất, chỉ có thông số DR là thấp hơn RNN không đáng kể. Mô hình cho kết quả xấu nhất trong thực nghiệm là mô hình mạng nơ-ron GRU với tất cả các thông số Accuracy, Precision, F1 Score đều có kết quả thấp nhất và tỷ lệ dương tính giả FAR cao.

Bảng VIII. Đánh giá mô hình huấn luyện DSD-RNN, DSD-LSTM, DSD-GRU trên bộ dữ liệu rút gọn UNSW-NB15

	FAR%	Acc%	Prec%	DR%	F1-S%
DSD-RNN	**32.62**	**84.27**	**78.88**	98.06	**87.36**
DSD-LSTM	33.10	84.21	78.67	**98.35**	87.35
DSD-GRU	32.63	84.16	78.80	97.87	87.25

Kết quả thực nghiệm ba mô hình DSD-RNN, DSD-LSTM và DSD-GRU như trong Bảng VIII cho thấy mô hình kết hợp DSD-RNN cho kết quả tốt nhất với Accuracy, Precision, F1-Score có chỉ số cao nhất, tỷ lệ dương tính giả FAR thấp nhất, chỉ có thông số DR là thấp hơn mô hình kết hợp DSD-LSTM. Mô hình kết hợp cho kết quả xấu nhất trong thực nghiệm là DSD-GRU với tất cả các thông số Accuracy, Precision, F1-Score đều có kết quả thấp nhất và tỷ lệ dương tính giả FAR cao.

Qua thực nghiệm, so sánh hiệu quả giữa mô hình mạng nơ-ron RNN, LSTM, GRU với mô hình kết hợp huấn luyện 3 giai đoạn DSD-RNN, DSD-LSTM, DSD-GRU như trong Bảng IX, Bảng X và Bảng XI đều giúp tăng hiệu quả mô hình với các thông số Accuracy, Precision, DR, F1 Score đều tăng và tỷ lệ dương tính giả FAR giảm.

B. Đánh giá

Giữa ba mô hình mạng nơ-ron RNN, LSTM, GRU có kết quả khá tương đồng nhau. RNN cho kết quả tốt nhất với tiêu chí tỷ lệ phát hiện 98.01%. LSTM cho kết quả tốt nhất với tỷ lệ cảnh báo giả (FAR) 33.37%, độ chính xác 83.70%, Precision 78.23% và F1 Score 86.85%. GRU không có kết quả nào là tốt nhất trong thử nghiệm này.

Tương tự, chúng tôi cũng so sánh giữa 3 mô hình DSD-RNN, DSD-LSTM và DSD-GRU. Kết quả giữa ba mô hình này tương đồng nhau. DSD-RNN cho tỷ lệ cảnh báo giả tốt nhất với 32.62%, độ chính xác 84.27%, Precision 78.88%, F1-Score 87.36%. DSD-LSTM cho

kết quả tốt nhất với tỷ lệ phát hiện 98.35%. DSD-GRU không có kết quả nào là tốt nhất trong thử nghiệm này.

Bảng IX. Đánh giá mô hình huấn luyện DSD-RNN và RNN trên bộ dữ liệu rút gọn UNSW-NB15

	FAR%	Acc%	Prec%	DR%	F1-S%
DSD-RNN	**32.62**	**84.27**	**78.88**	**98.06**	**87.36**
RNN	34.17	83.54	77.87	98.01	86.78

Bảng X. Đánh giá mô hình huấn luyện DSD-LSTM và LSTM trên bộ dữ liệu rút gọn UNSW-NB15

	FAR%	Acc%	Prec%	DR%	F1-S%
DSD-LSTM	**33.10**	**84.21**	**78.67**	**98.35**	**87.35**
LSTM	33.37	83.70	78.23	97.95	86.85

Bảng XI. Đánh giá mô hình huấn luyện DSD-GRU và GRU trên bộ dữ liệu rút gọn UNSW-NB15

	FAR%	Acc%	Prec%	DR%	F1-S%
DSD-GRU	**32.63**	**84.16**	**78.80**	**97.87**	**87.25**
GRU	34.22	82.90	77.67	96.88	86.19

RNN có vấn đề về "vanishing gradient" (gradient được sử dụng để cập nhật giá trị của weight matrix trong RNN và nó có giá trị nhỏ dần theo từng layer khi thực hiện back propagation). Khi gradient trở nên rất nhỏ (có giá trị gần bằng 0) thì giá trị của weight matrix sẽ không được cập nhật thêm và do đó mạng nơ-ron sẽ dừng việc học tại lớp này. Tuy nhiên khi dùng DSD, mạng RNN sẽ được cắt tỉa trọng số và được huấn luyện lại lần nữa. Điều này giúp khôi phục trọng số, khắc phục được vấn đề "vanishing gradient" và làm tăng hiệu quả huấn luyện ở pha Dense cuối cùng. Trong khi đó, LSTM và GRU đã thêm các cổng (cổng quên, cổng cập nhật và hàm *tanh*) để khắc phục vấn đề "vanishing gradient", vì vậy cải tiến hiệu quả phát hiện xâm nhập của LSTM và GRU bằng huấn luyện DSD không bằng khi áp dụng DSD vào RNN.

Ngoài ra, qua thực nghiệm cũng cho thấy khi áp dụng mô hình huấn luyện DSD vào 3 mạng nơ-ron RNN, LSTM, GRU đều cho kết quả khả quan hơn. Bảng XII minh họa kết quả của mô hình huấn luyện DSD-(RNN/LSTM/GRU) so với mô hình gốc RNN/LSTM/GRU. Với tiêu chí FAR, "-" là tốt hơn và các tiêu chí còn lại thì "+" là tốt hơn.

Bảng XII. Bảng đánh giá mô hình huấn luyện DSD (DSD-RNN, DSD-LSTM, DSD-GRU) so với mô hình mạng nơ-ron gốc (RNN, LSTM, GRU)

	FAR (%)	Acc (%)	Prec (%)	DR (%)	F1-S (%)
DSD-RNN	- 1.556	+ 0.7249	+ 1.0044	+ 0.0463	+ 0.5760
DSD-LSTM	- 0.270	+ 0.5123	+ 0.4422	+ 0.3944	+ 0.5009
DSD-GRU	- 1.595	+ 1.2646	+ 1.1272	+ 0.9949	+ 1.0612

VI. KẾT LUẬN

Từ kết quả thực nghiệm cho thấy cả 3 mô hình mạng nơ-ron RNN, LSTM, GRU đều cho kết quả tốt trên tập dữ liệu rút gọn UNSW-NB15. Mô hình ©mạng nơ-ron LSTM cho thấy hiệu quả tốt nhất. Việc đưa mô hình huấn luyện DSD vào mạng nơ-ron RNN, LSTM, GRU giúp cải thiện hiệu suất với hầu hết các tiêu chí. Trong đó mô hình mạng nơ-ron RNN được cải thiện hiệu quả nhất sau khi áp dụng mô hình huấn luyện DSD. Trong phạm vi thực nghiệm của nghiên cứu này, chúng tôi dùng độ cắt tỉa là 25%. Hướng tiếp cận trong tương lai, chúng tôi sẽ nghiên cứu việc thay đổi độ cắt tỉa bao nhiêu là phù hợp trên tập dữ liệu đầy đủ UNSW-NB15 và áp dụng mô hình huấn luyện vào các mạng nơ-ron khác với thiết kế chi tiết hơn nhằm đánh giá đầy đủ hơn nữa về hiệu quả của sự kết hợp này.

TÀI LIỆU THAM KHẢO

[1] Hettich, S. and Bay, S. D., "KDD Cup 1999 Data," 28 October 1999. [Online]. Available: http://kdd.ics.uci.edu. [Accessed 02 Feb 2020].

[2] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," 2009. [Online]. Available: https://www.unb.ca/cic/datasets/.

[3] Nour Moustafa, Jill Slay, "UNSW-NB15: A Comprehensive Data set for Network," in *Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, 2015.

[4] Moustafa, Nour Moustafa Abdelhameed, "The UNSW-NB15 Dataset Description," 14 November 2018. [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/. [Accessed 02 August 2020].

[5] Vinayakumar R et al., "A Comparative Analysis of Deep learning Approaches for Network Intrusion Detection Systems (N-IDSs)," *International Journal of Digital Crime and Forensics*, vol. 11, no. 3, p. 25, July 2019.

[6] Song Han∗, Huizi Mao, Enhao Gong, Shijian Tang, William J. Dally, "DSD: Dense-Sparse-Dense Traning For Deep Neural Networks," in *ICLR 2017*, 2017.

[7] Anani, Wafaa, "Recurrent Neural Network Architectures Toward Intrusion Detection," in *Recurrent Neural Network Architectures Toward Intrusion Detection*, Electronic Thesis and Dissertation Repository. 5625, 2018.

[8] K. Cho, J. Chung, C .Gulcehre, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *Computing Research Repository (CoRR)*, 2014.

[9] Brownlee, Jason, "Overfitting and Underfitting With Machine Learning Algorithms," 12 August 2019. [Online]. Available: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/. [Accessed 03 August 2020].

[10] Brownlee, Jason, "Gentle Introduction to the Bias-Variance Trade-Off in Machine Learning," 25 October 2019. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-to-the-bias-variance-trade-off-in-machine-learning/. [Accessed 03 August 2020].

[11] Gupta, Prashant, "Cross-Validation in Machine Learning," Towards Data Science, 05 June 2017. [Online]. Available: https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f. [Accessed 02 August 2020].

[12] Pedregosa et al., "Scikit-learn: Machine Learning in Python," 2011. [Online]. Available: https://scikit-learn.org/stable/modules/.

[13] Sergey Ioffe, Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.

ENHANCING NETWORK INTRUSION DETECTION EFFICIENCY USING DSD

***Abstract*:** Most modern intrusion detection models are applying machine learning to produce intrusion detection and classification results with high accuracy. This study proposes a model combining multi-layered neural networks with DSD multi-stage training method to simultaneously improve many criteria related to the performance of intrusion detection systems on the UNSW - NB15 dataset which is regularly updated for the features and follows new attack patterns. We conduct experiments on 3 neural network models RNN, LSTM, and GRU to evaluate the efficiency associated with each model through many criteria such as accuracy, detection rate, false alarm rate, precision and F1-Score.

Keywords: *network security, machine learning, deep learning, IDS, neural network.*

Huỳnh Trọng Thưa, Trưởng Bộ môn An toàn Thông tin, Khoa Công nghệ Thông tin 2, Học viện Công nghệ Bưu chính Viễn thông cơ sở tại TP. Hồ Chí Minh. Thưa nhận bằng Cử nhân Công nghệ Thông tin của Đại học Khoa học Tự nhiên TP. Hồ Chí Minh, bằng Thạc sĩ Kỹ thuật Máy tính tại Đại học Kyung Hee, Hàn Quốc và bằng Tiến sĩ Khoa học Máy tính tại Đại học Bách Khoa - Đại học Quốc gia TP. Hồ Chí Minh. Lĩnh vực nghiên cứu: An toàn và bảo mật thông tin, blockchain, mật mã học, điều tra.

Email: htthua@ptithcm.edu.vn

Nguyễn Hoàng Thành, Nhận bằng Kỹ sư Hệ thống thông tin của Học viện Công nghệ Bưu chính Viễn thông năm 2013. Hiện tại, Thành đang là học viên cao học của Học viện Công nghệ Bưu chính Viễn thông cơ sở tại TP. Hồ Chí Minh. Lĩnh vực nghiên cứu chính: An toàn thông tin, học máy

Email: hthanhsg@gmail.com