

BÁO CÁO TỔNG KẾT ĐỒ ÁN MÔN HỌC

Môn học: **An toàn mạng**

Tên chủ đề: **Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System**

Giáo viên hướng dẫn: **ThS. Nghi Hoàng Khoa**

Lớp: **NT140.011.ANTT** Nhóm: **12**

1. THÔNG TIN THÀNH VIÊN NHÓM:

(Sinh viên liệt kê tất cả các thành viên trong nhóm)

STT	Họ và tên	MSSV	Email
1	Nguyễn Huy Hoàng	21522094	21522094@gm.uit.edu.vn
2	Nguyễn Đức Tài	21521395	21521395@gm.uit.edu.vn
3	Nguyễn Hoài Phương	21520408	21520408@gm.uit.edu.vn
4	Trần Minh Duy	21522010	21522010@gm.uit.edu.vn

2. TÓM TẮT NỘI DUNG THỰC HIỆN:¹

Phần này tóm tắt nội dung của đồ án, sinh viên báo cáo nội dung chi tiết ở phần **BÁO CÁO CHI TIẾT**

2.1. Chủ đề nghiên cứu trong lĩnh vực an toàn mạng:

- ☒ Phát hiện và phòng chống mã độc
- ☐ Bảo mật mạng không dây
- ☐ Bảo mật ứng dụng web
- ☐ Phân tích tấn công mạng
- ☐ Bảo mật IOT
- ☐ Khác:

2.2. Tên bài báo tham khảo chính:

¹ Ghi nội dung tương ứng theo mô tả

Tên tiếng Anh: *Fusion of statistical importance for feature selection in Deep Neural Network-based Intrusion Detection System*

Tên tiếng Việt (dịch): *Kết hợp dữ liệu thống kê để lựa chọn các đặc trưng trong Hệ thống Phát hiện Xâm nhập dựa trên Deep Neural Network*

2.3. Tóm tắt nội dung chính:

- Hệ thống Phát hiện Xâm nhập (IDS) là một phần cần thiết của mạng, đã có nhiều nghiên cứu toàn diện trong lĩnh vực IDS và đã phát triển các phương pháp khác nhau để thiết kế hệ thống phát hiện và phân loại xâm nhập. Việc sử dụng các kỹ thuật Học sâu (Deep Learning - DL) trở nên khá phổ biến vì khả năng học dữ liệu một cách toàn diện của chúng
- Trong nghiên cứu, các tác giả đề xuất một kỹ thuật lựa chọn đặc trưng mới (feature selection) cải thiện hiệu suất của IDS dựa trên DNN bằng cách xếp hạng các feature theo rank được tính ra từ Độ lệch chuẩn và Khoảng cách giữa Mean và Median, các đặc trưng được loại bỏ dựa trên thứ hạng của chúng giúp việc học dữ liệu tốt hơn.
- Phương pháp được đánh giá trên ba bộ dữ liệu phát hiện xâm nhập: NSL-KDD, UNSW_NB-15 và CIC-IDS-2017. Hiệu suất được đánh giá với các chỉ số độ accuracy, precision, recall, f -score, False Positive Rate (FPR) và thời gian thực thi. Hơn nữa, kết quả đạt được cũng được kiểm định thống kê bằng thử nghiệm xếp hạng Wilcoxon Signed.
- Kỹ thuật feature selection được đề xuất đạt được kết quả tốt hơn so với các kỹ thuật hiện có với IDS dựa trên DNN cho cả ba bộ dữ liệu phát hiện xâm nhập được sử dụng.

2.4. Code nhóm thực hiện lập trình và triển khai cho demo:

Code thực hiện được lưu tại [ATM project - Google Drive](#)

3. TỰ ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH SO VỚI KẾ HOẠCH THỰC HIỆN:

100%

4. NHẬT KÝ PHÂN CÔNG NHIỆM VỤ:

STT	Công việc	Phân công nhiệm vụ
1		
2		
3		

BÁO CÁO TỔNG KẾT CHI TIẾT

Phần bên dưới của báo cáo này là tài liệu báo cáo tổng kết - chi tiết của nhóm thực hiện cho đề tài này.

A. GIỚI THIỆU TỔNG QUAN

A.1. Ngữ cảnh

Hệ thống Phát hiện Xâm nhập (IDS) là một phần cần thiết của mạng, đã có nhiều nghiên cứu toàn diện trong lĩnh vực IDS và đã phát triển các phương pháp khác nhau để thiết kế hệ thống phát hiện và phân loại xâm nhập. Việc sử dụng các kỹ thuật Học sâu (Deep Learning - DL) trở nên khá phổ biến vì khả năng học dữ liệu một cách toàn diện của chúng.

A.2. Vấn đề

Phần lớn nghiên cứu đã thiết kế IDS sử dụng các kỹ thuật feature selection hiện có bằng cách sử dụng các công cụ trực quan hóa như WeKa, cũng như sử dụng các bộ dữ liệu lỗi thời thiếu kịch bản thực nghiệm.

Table 1
Comparative summary of existing Machine Learning (ML) approaches for IDS.

Ref	Technique	Feature selection	Dataset	Results
[16]	DT	CFS	NSL-KDD	· Accuracy for NSL-KDD: 90.30%
[15]	LS-SVM	FMI	Kyoto 2006, KDD CUP 99, and NSL-KDD	· DR for KDD CUP 99: 99.46% · DR for NSL-KDD: 98.76% · DR for Kyoto 2006: 99.64%
[17]	RF	Attribute evaluator, greedy stepwise, IG, and ranker	KDD CUP 99, UNSW_NB-15	Comparative analysis is presented in graphical format for feature selection technique considered.
[22]	RepTree	IG	NSL-KDD, UNSW_NB-15	· Accuracy for NSL-KDD: 89.85% · Accuracy for UNSW_NB-15: 88.95%
[14]	DT	GA	KDD CUP 99, UNSW_NB-15	· DR for KDD CUP 99: 99.90% · DR for UNSW_NB-15: 81.24%
[19]	kNN, DT, BME, XGBoost, and RF	Feature importance	UNSW_NB-15	· Accuracy for kNN: 71.01% Accuracy for DT: 74.22% Accuracy for BME: 74.64% Accuracy for XGBoost: 71.43% Accuracy for RF: 74.87%
[21]	RF	IG	UNSW_NB-15	· Accuracy for UNSW_NB-15: 85.78%
[24]	Bagging classifier	GR	NSL-KDD	Accuracy for NSL-KDD: 84.25%
[23]	IELM	APCA	NSL-KDD, UNSW_NB-15	· Accuracy for NSL-KDD: 81.22% · Accuracy for UNSW_NB-15: 70.51%
[20]	Rotation forest and Bagging classifier	PSO, ACO, and GA	KDD CUP 99	· Accuracy for KDD CUP 99: 72.52%
[25]	NB, MLP	Combined feature selection technique	KDD CUP 99	· Accuracy for NB: 93.00% · Accuracy for MLP: 97.00%
[18]	Rule-based multiple tree classifiers	IG	UNSW_NB-15	· Accuracy for UNSW_NB-15: 84.83%

A.3. Giải pháp

Trong nghiên cứu, các tác giả đề xuất một kỹ thuật lựa chọn đặc trưng mới (feature selection) cải thiện hiệu suất của IDS dựa trên DNN bằng cách xếp hạng các

feature theo rank được tính ra từ Độ lệch chuẩn và Khoảng cách giữa Mean và Median, các đặc trưng được loại bỏ dựa trên thứ hạng của chúng giúp việc học dữ liệu tốt hơn.

A.4. Kết quả

Kỹ thuật feature selection được đề xuất đạt được kết quả tốt hơn so với các kỹ thuật hiện có với IDS dựa trên DNN cho cả ba bộ dữ liệu phát hiện xâm nhập được sử dụng.

B. PHƯƠNG PHÁP THỰC HIỆN

B.1. Kiến thức nền tảng

B.1.1. Mối liên hệ giữa IDS và feature selection

IDS sử dụng dữ liệu từ các bộ dữ liệu xâm nhập mạng - Intrusion detection dataset. Đó là các thông tin được trích xuất từ file pcap hoặc tcpdump có được bằng cách ghi lại các gói dữ liệu truyền qua mạng (sniffing) sử dụng các công cụ như Wireshark và Nmap. Các thông tin được trích xuất bao gồm nhiều chi tiết khác nhau liên quan đến giao tiếp trong mạng và nó thường được gọi là các **feature** trong dataset.

Tuy nhiên, khi xem xét các feature của mạng, có khả năng các bộ dữ liệu phát hiện xâm nhập có thể bao gồm các feature dư thừa và không liên quan có thể ảnh hưởng hoặc không góp phần vào quá trình dự đoán và phân loại. Feature selection giúp lựa chọn các feature thích hợp và loại bỏ các feature dư thừa trong training IDS. Hình 1 thể hiện vai trò và tầm quan trọng của feature selection được đưa ra trong bài báo:

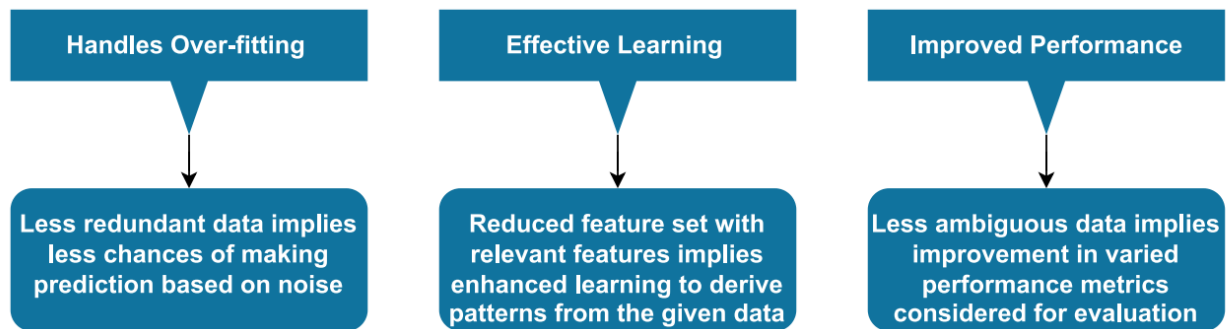


Fig. 1. Scientific contribution and importance of the proposed feature selection technique.

B.1.2. Nhắc lại một số kiến thức thống kê

1. Standard deviation (Độ lệch chuẩn)

Độ lệch chuẩn của các feature có thể được mô tả như một thước đo thống kê để đo mức độ biến thiên hoặc độ lệch của các feature so với giá trị trung bình. Độ lệch chuẩn có thể được tính bằng phương trình (1):

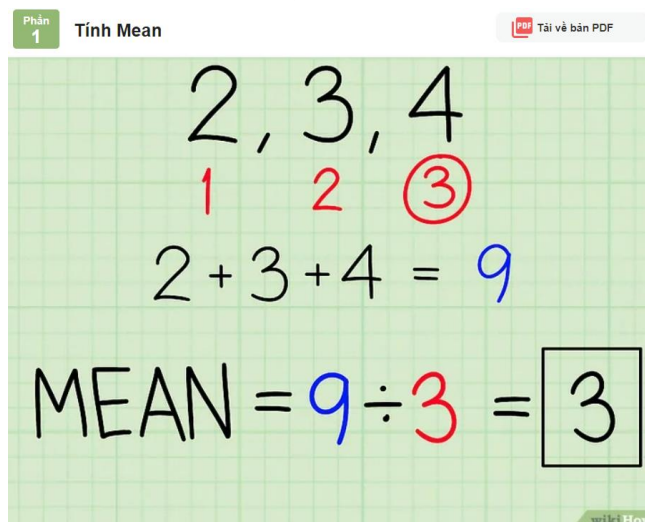
$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}} \quad (1)$$

Trong đó, σ là độ lệch chuẩn, N là số lượng mẫu, x_i là giá trị thứ i của feature, μ là giá trị trung bình.

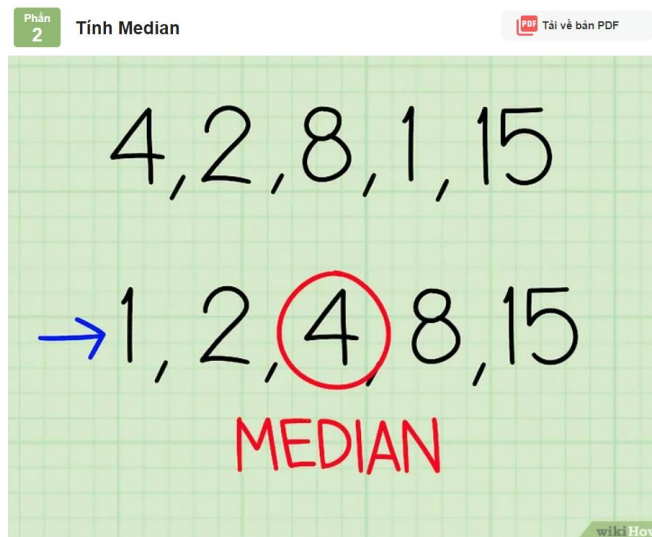
Giá trị độ lệch chuẩn cao cho thấy feature đó bị phân tán trên phạm vi giá trị lớn và giá trị độ lệch chuẩn thấp cho thấy các giá trị feature nằm gần nhau so với giá trị trung bình. Do đó, feature selection sử dụng độ lệch chuẩn sẽ chọn các feature có giá trị độ lệch chuẩn cao vì khi giá trị feature được mở rộng trên phạm vi lớn, có thể đạt được kết quả dự đoán hiệu quả vì nó thể hiện sự khác biệt của các feature trên tất cả các mẫu.

2. Mean và median

Mean là giá trị trung bình của các giá trị trong feature, được tính bằng cách cộng một nhóm các số rồi chia cho số lượng các số.



Median là giá trị trung vị của feature, là số nằm ở giữa một nhóm các số; có nghĩa là, phân nửa các số có giá trị lớn hơn số trung vị, còn phân nửa các số có giá trị bé hơn số trung vị.



Giá trị Mean và Median được sử dụng để thể hiện mức độ sai lệch tương đối trong phân phối dữ liệu, được biểu thị bằng biểu thức (2).

$$D = |Mean - Median| \quad (2)$$

Giá trị D chênh lệch cao biểu giá trị feature tập trung vào một phía và có độ phân tán cao, giá trị D thấp biểu thị giá trị feature đối xứng qua Mean và Median và có độ phân tán thấp, do đó, kĩ thuật feature selection đề xuất ưu tiên chọn feature có giá trị D tính ra lớn.

B.2. Phương pháp feature selection được đề xuất

- Tính độ lệch chuẩn (σ) của các feature trong dataset.
- Xếp hạng các feature dựa trên giá trị độ lệch chuẩn từ cao xuống thấp, gọi xếp hạng này là *Rank1*.
- Tính D là sự khác nhau giữa giá trị trung bình và trung vị cho các feature của dataset.
- Xếp hạng các feature dựa trên giá trị chênh lệch từ cao xuống thấp. gọi xếp hạng này là *Rank2*.
- Tính Combined Feature Rank = *Rank1* + *Rank2*.
- Thêm lần lượt các feature từ cao xuống thấp trong Combined Feature Rank vào tập hợp feature được chọn để training model cho đến khi độ chính xác không cao hơn so với tập feature trước đó.

Thuật toán đệ quy feature selection sử dụng kỹ thuật đề xuất được trình bày trong Thuật toán 1.

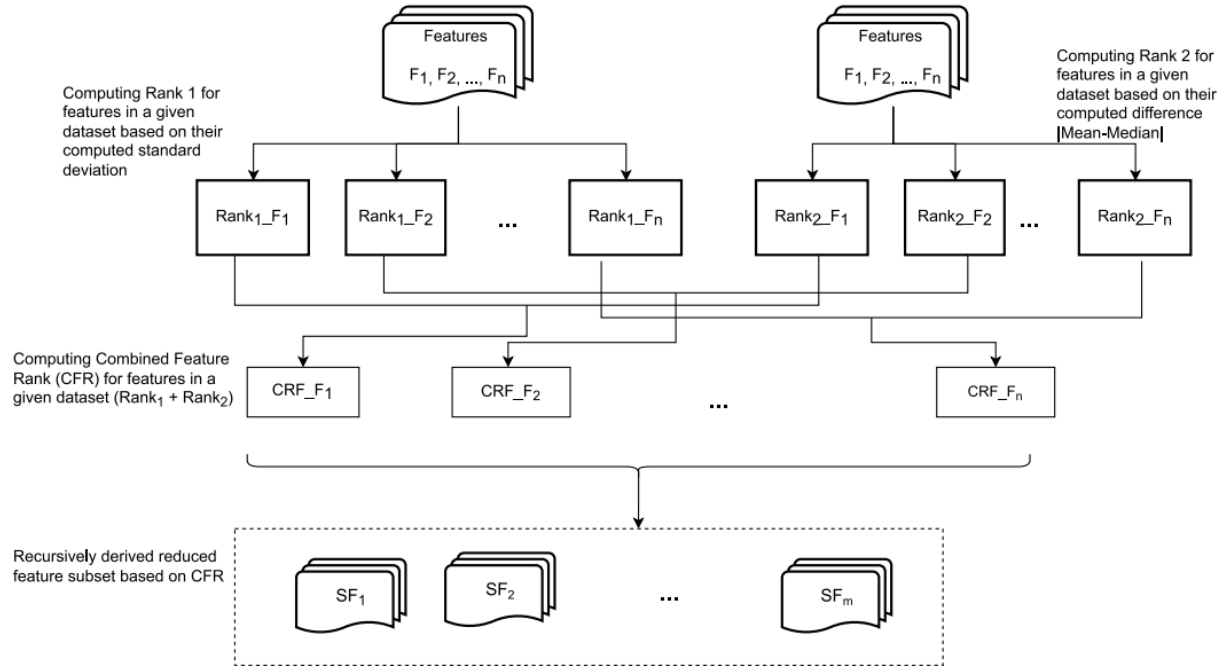


Fig. 2. Conceptualization strategy of the proposed feature selection technique.

Algorithm 1 Recursive Feature Selection Using Fusion of Standard Deviation and Absolute Difference of Mean and Median

- 1: Consider Dataset D_l for intrusion detection and classification where $D = \{NSL-KDD, UNSW_NB-15, CIC-IDS-2017\}$.
- 2: For features of dataset D_l , calculate standard deviation for each feature using equation (1).
- 3: Sort features from high to low based on their standard deviation and rank them. Consider the assigned rank as $Rank_1$.
- 4: For features of dataset D_l , calculate absolute value of difference between mean and median of each feature using equation (2).
- 5: Sort features from high to low based on the absolute value of the difference and rank them. Consider the assigned rank as $Rank_2$.
- 6: Compute combined feature rank R by summing $Rank_1$ and $Rank_2$.
- 7: For each feature $F_i \in F$ of dataset D_l do,
- 8: Remove the highest rank feature F_i from F and update S_l as $S_l = S_l \cup F_i$.
- 9: Train DNN model on training set with S_l features and compute model accuracy.
- 10: Repeat Steps [8-9], for features F_i until increase in accuracy is recorded more than previous computed accuracy.
- 11: Store the derived relevant features in subset S_l for the Dataset D_l .
- 12: Use feature subset S_l for training DNN-based IDS for the dataset D_l .

C. CHI TIẾT HIỆN THỰC VÀ THỰC NGHIỆM PHƯƠNG PHÁP

	ENVIRONMENT
Nhóm	Google Colab - GPU T4 RAM hệ thống 12,7 GB RAM GPU 15 GB Ổ đĩa 78.2 GB using Python3
PAPER	Intel(R) Core(TM) i5-8265U CPU processor 64-bit Windows10 operating system 8.00 GB RAM using Python

C.1. Hiện thực phương pháp

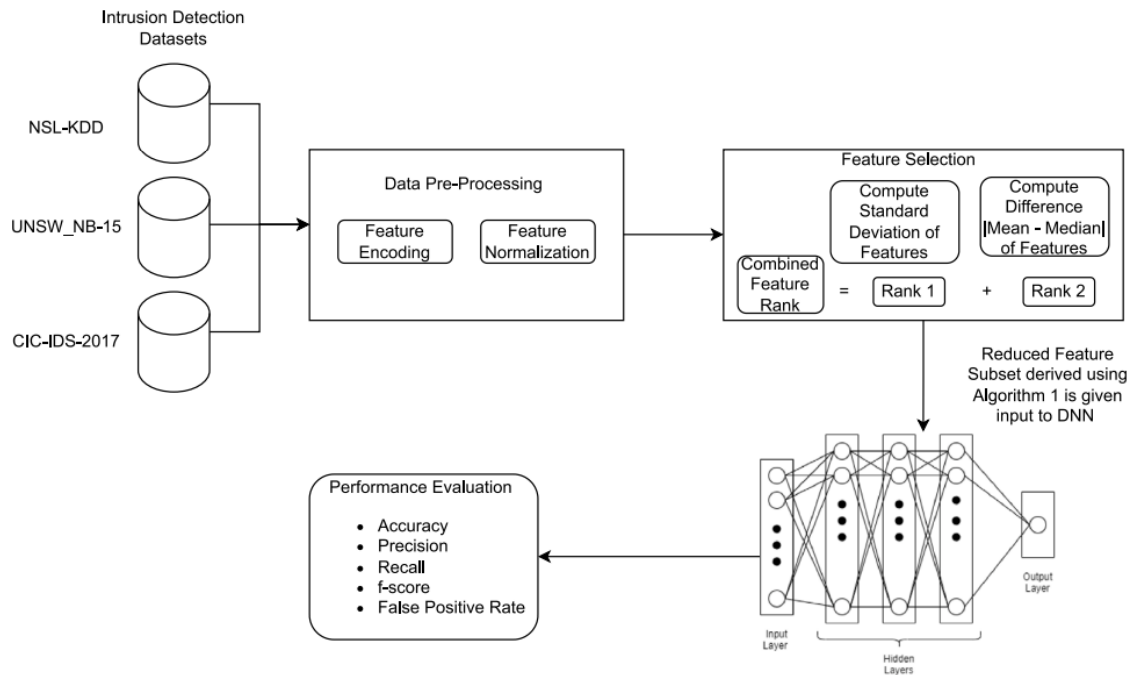


Fig. 3. Schematic of the proposed approach.

C.1.1. Dataset

Các dataset được sử dụng gồm NSL-KDD, UNSW_NB-15, và CICIDS-2017. Đây là các dataset tạo ra trong môi trường mạng khác nhau, có nhiều feature gồm cả dữ liệu thực tế và tổng hợp. Thông tin dataset và số mẫu sử dụng được thống kê trong bảng 2.



Table 2
 Statistics of the experimental datasets [8].

Criteria (↓)/Dataset (→)	NSL-KDD	UNSW_NB-15	CIC-IDS-2017
Type of network traffic	Real & Synthetic	Synthetic	Real
Number of features	41	42	79
Number of attack categories	4	9	7
Number of classes	5	10	15
Number of data samples	148 517	257 673	225 745
Number of samples in training set	125 973	175 341	165 730
Number of samples in test set	22 544	82 332	60 015

NSL-KDD

Drive của tôi > Dataset > NSL-KDD ▾

Loại ▾
 Người ▾
 Lần sửa đổi gần đây nhất ▾

Tên ↑	Chủ sở hữu	Sửa đổi lần cuối ▾	Kích cỡ tệp
KDDTest+.txt	tôi	23 thg 11, 2023 tôi	3,3 MB
KDDTrain+.txt	tôi	23 thg 11, 2023 tôi	18,2 MB

duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_shell	su_
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0
1	0	udp	other	SF	146	0	0	0	0	0	0	0	0	0
2	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0
3	0	tcp	http	SF	232	8153	0	0	0	0	1	0	0	0
4	0	tcp	http	SF	199	420	0	0	0	0	1	0	0	0
...
148512	0	tcp	smtp	SF	794	333	0	0	0	0	1	0	0	0
148513	0	tcp	http	SF	317	938	0	0	0	0	1	0	0	0
148514	0	tcp	http	SF	54540	8314	0	0	0	2	1	1	0	0
148515	0	udp	domain_u	SF	42	42	0	0	0	0	0	0	0	0
148516	0	tcp	sunrpc	REJ	0	0	0	0	0	0	0	0	0	0

148517 rows x 42 columns

UNSW_NB-15

Drive của tôi > Dataset > UNSW_NB15 ▾

Loại ▾
 Người ▾
 Lần sửa đổi gần đây nhất ▾

Tên ↑	Chủ sở hữu	Sửa đổi lần cuối ▾	Kích cỡ tệp
UNSW_NB15_testing-set.csv	tôi	21 thg 11, 2023 tôi	14,7 MB
UNSW_NB15_training-set.csv	tôi	21 thg 11, 2023 tôi	30,8 MB



	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	sinpkt	dinpkt
0	0.121478	tcp	-	FIN	6	4	258	172	74.087490	252	254	1.415894e+04	8495.365234	0	0	24.295600	8.375000
1	0.649902	tcp	-	FIN	14	38	734	42014	78.473372	62	252	8.395112e+03	503571.312500	2	17	49.915000	15.432865
2	1.623129	tcp	-	FIN	8	16	364	13186	14.170161	62	252	1.572272e+03	60929.230470	1	6	231.875571	102.737203
3	1.681642	tcp	ftp	FIN	12	12	628	770	13.677108	62	252	2.740179e+03	3358.622070	1	3	152.876547	90.235726
4	0.449454	tcp	-	FIN	10	6	534	268	33.373826	254	252	8.561499e+03	3987.059814	2	1	47.750333	75.659602
...
257668	0.000005	udp	-	INT	2	0	104	0	200000.005100	254	0	8.320000e+07	0.000000	0	0	0.005000	0.000000
257669	1.106101	tcp	-	FIN	20	8	18062	354	24.410067	254	252	1.241044e+05	2242.109863	7	1	55.880051	143.700000
257670	0.000000	arp	-	INT	1	0	46	0	0.000000	0	0	0.000000e+00	0.000000	0	0	60000.720000	0.000000
257671	0.000000	arp	-	INT	1	0	46	0	0.000000	0	0	0.000000e+00	0.000000	0	0	60000.732000	0.000000
257672	0.000009	udp	-	INT	2	0	104	0	111111.107200	254	0	4.622222e+07	0.000000	0	0	0.009000	0.000000

257673 rows x 43 columns

CICIDS-2017

Drive của tôi > Dataset > CIC-IDS-2017 ▾

Loại ▾ Người ▾ Lần sửa đổi gần đây nhất ▾

Tên ↑	Chủ sở hữu	Sửa đổi lần cuối ▾	Kích cỡ tệp
 Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	 tôi	25 thg 11, 2023 tôi	73,6 MB

	Destination Port	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	Bwd Packet Length Max	Bwd Packet Length Min	Bwd Packet Length Mean	Bwd Packet Length Std	Flow Bytes/s	Flow Packets/s	Flow IAT Mean	Flow IAT Std
0	54865	3	2	0	12	0	6	6	6.0	0.0	0	0	0.0	0.0	4.000000e+06	666666.66670	3.0	0
1	55054	109	1	1	6	6	6	6	6.0	0.0	6	6	6.0	0.0	1.100917e+05	18348.62385	109.0	0
2	55055	52	1	1	6	6	6	6	6.0	0.0	6	6	6.0	0.0	2.307692e+05	38461.53846	52.0	0
3	46236	34	1	1	6	6	6	6	6.0	0.0	6	6	6.0	0.0	3.529412e+05	58823.52941	34.0	0
4	54863	3	2	0	12	0	6	6	6.0	0.0	0	0	0.0	0.0	4.000000e+06	666666.66670	3.0	0
...
225740	61374	61	1	1	6	6	6	6	6.0	0.0	6	6	6.0	0.0	1.967213e+05	32786.88525	61.0	0
225741	61378	72	1	1	6	6	6	6	6.0	0.0	6	6	6.0	0.0	1.666667e+05	27777.77778	72.0	0
225742	61375	75	1	1	6	6	6	6	6.0	0.0	6	6	6.0	0.0	1.600000e+05	26666.66667	75.0	0
225743	61323	48	2	0	12	0	6	6	6.0	0.0	0	0	0.0	0.0	2.500000e+05	41666.66667	48.0	0
225744	61326	68	1	1	6	6	6	6	6.0	0.0	6	6	6.0	0.0	1.764706e+05	29411.76471	68.0	0

225745 rows x 79 columns

C.1.2. Data pre-processing

Các kỹ thuật tiền xử lý dữ liệu được áp dụng để dễ dàng chuyển đổi dữ liệu để xử lý và học một cách trơn tru. Trong bài báo, hai kỹ thuật tiền xử lý dữ liệu được áp dụng, đó là One-hot encoding và Standard scalar. One-hot encoding được dùng để chuyển đổi các categorical features thành các numerical features. Standard scalar để chuẩn hóa giá trị cho các feature bằng cách trừ giá trị trung bình và scale giá trị feature thành phương sai đơn vị.

Về One-Hot encoding:

One-hot encoding là quá trình biến đổi từng giá trị thành các feature nhị phân chỉ chứa giá trị 1 hoặc 0. Mỗi mẫu trong categorical feature sẽ được biến đổi thành một vector có

kích thước bằng feature ban đầu nhưng chỉ với một trong các giá trị là 0 (biểu thị nó là inactive) hoặc 1 (biểu thị nó là active). Ví dụ như hình dưới, có thể thấy rằng mẫu dữ liệu có id là 1 đang có màu đỏ (giá trị ở cột color_red = 1), xác định màu cho các mẫu khác được thực hiện tương tự

id	color			
1	red			
2	blue			
3	green			
4	blue			

One Hot Encoding

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

One hot encoding

```
df_oh = pd.get_dummies(df)
df_oh
```

service_uucp	service_uucp_path	service_vmnet	service_whois	flag_OTH	flag_REJ	flag_RSTO	flag_RSTOS0	flag_RSTR	flag_S0	flag_S1	flag_S2	flag_S3	flag_SF	flag_SH
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
...
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Hình 1. Thực hiện One-hot encoding với hàm pandas.get_dummies()

Về Standard Scaler:

Kĩ thuật chuẩn hoá được áp dụng đối với những biến không có phân phối chuẩn. Biến được biến đổi theo kì vọng và độ lệch chuẩn như sau:

$$\mathbf{x}' = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\sigma(\mathbf{x})}$$

Từ đó suy ra giá trị của biến sau khi được biến đổi ngược lại:

$$\mathbf{x} = \mathbf{x}' * \sigma(\mathbf{x}) + \bar{\mathbf{x}}$$

Các biến sau khi được chuẩn hoá sẽ có cùng một dạng phân phối chuẩn hoá với trung bình bằng 0 và phương sai bằng 1. Nhờ đó quá trình huấn luyện sẽ trở nên ổn định và hội tụ tới nghiệm tối ưu nhanh hơn.

Code thực hiện:

```

▼ Scaler Data

[ ] scaler = StandardScaler()
    data_scale = scaler.fit_transform(X)
    data_scale

array([[ -0.11248106, -0.00734564, -0.00461423, ..., -0.04483903,
         0.80839092, -0.04818309],
       [ -0.11248106, -0.00740942, -0.00461423, ..., -0.04483903,
         0.80839092, -0.04818309],
       [ -0.11248106, -0.00743641, -0.00461423, ..., -0.04483903,
        -1.23702527, -0.04818309],
       ...,
       [ -0.11248106,  0.00264568, -0.00236933, ..., -0.04483903,
         0.80839092, -0.04818309],
       [ -0.11248106, -0.00742865, -0.00460289, ..., -0.04483903,
         0.80839092, -0.04818309],
       [ -0.11248106, -0.00743641, -0.00461423, ..., -0.04483903,
        -1.23702527, -0.04818309]])
    
```

```

final_df = pd.concat([pd.DataFrame(data_scale, columns=X.columns), y], axis=1)
final_df
    
```

	duration	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_shell	su_attempted	num_root
0	-0.112481	-0.007346	-0.004614	-0.01468	-0.085488	-0.010403	-0.094071	-0.059832	-0.821249	-0.011473	-0.038865	-0.023032	-0.012064
1	-0.112481	-0.007409	-0.004614	-0.01468	-0.085488	-0.010403	-0.094071	-0.059832	-0.821249	-0.011473	-0.038865	-0.023032	-0.012064
2	-0.112481	-0.007436	-0.004614	-0.01468	-0.085488	-0.010403	-0.094071	-0.059832	-0.821249	-0.011473	-0.038865	-0.023032	-0.012064
3	-0.112481	-0.007394	-0.002413	-0.01468	-0.085488	-0.010403	-0.094071	-0.059832	1.217658	-0.011473	-0.038865	-0.023032	-0.012064
4	-0.112481	-0.007400	-0.004501	-0.01468	-0.085488	-0.010403	-0.094071	-0.059832	1.217658	-0.011473	-0.038865	-0.023032	-0.012064
...
148512	-0.112481	-0.007290	-0.004524	-0.01468	-0.085488	-0.010403	-0.094071	-0.059832	1.217658	-0.011473	-0.038865	-0.023032	-0.012064
148513	-0.112481	-0.007378	-0.004361	-0.01468	-0.085488	-0.010403	-0.094071	-0.059832	1.217658	-0.011473	-0.038865	-0.023032	-0.012064
148514	-0.112481	0.002646	-0.002369	-0.01468	-0.085488	-0.010403	0.899396	-0.059832	1.217658	0.033509	-0.038865	-0.023032	-0.012064
148515	-0.112481	-0.007429	-0.004603	-0.01468	-0.085488	-0.010403	-0.094071	-0.059832	-0.821249	-0.011473	-0.038865	-0.023032	-0.012064
148516	-0.112481	-0.007436	-0.004614	-0.01468	-0.085488	-0.010403	-0.094071	-0.059832	-0.821249	-0.011473	-0.038865	-0.023032	-0.012064

148517 rows x 123 columns

C.1.3. Feature selection

Các feature được chọn bằng kỹ thuật feature selection được đề xuất phần B.2:

- **Bước 1:** Tính độ lệch chuẩn (σ) và Khoảng cách giữa Mean và Median (D) của các feature trong dataset

✓ FEATURES SELECTION

✓ Create a dictionary to store the values for each feature

```
[ ] # @title Create a dictionary to store the values for each feature
Standard_Deviation = {}
Mean_Median_Difference = {}

# Calculate values for each feature
for fts in X_train.columns:

    standard = np.std(X_train[fts])
    Standard_Deviation[fts] = standard

    abs_mean_median = np.abs(np.mean(X_train[fts]) - np.median(X_train[fts]))
    Mean_Median_Difference[fts] = abs_mean_median
```

- **Bước 2:** Tính rank_1 và rank_2 dựa trên kết quả ở bước 1

✓ Compute rank_1 and rank_2

```
# @title Compute rank_1 and rank_2

df_rank = pd.DataFrame({
    'Feature': X_train.columns,
    'Standard_Deviation': Standard_Deviation.values(),
    'Mean_Median_Difference': Mean_Median_Difference.values()
})

df_rank['Rank_1'] = df_rank['Standard_Deviation'].rank(ascending=False)
df_rank['Rank_2'] = df_rank['Mean_Median_Difference'].rank(ascending=False)
df_rank
```

	Feature	Standard_Deviation	Mean_Median_Difference	Rank_1	Rank_2
0	duration	0.984646	0.110244	111.0	41.0
1	src_bytes	1.099030	0.008321	7.0	114.0
2	dst_bytes	1.117986	0.005561	6.0	117.0
3	land	0.988217	0.014336	107.0	106.0
4	wrong_fragment	1.008366	0.086666	28.0	44.0
...
117	flag_S1	1.011563	0.052238	24.0	80.0
118	flag_S2	1.027758	0.032679	13.0	94.0
119	flag_S3	0.982262	0.043259	113.0	89.0
120	flag_SF	0.999784	0.807386	73.0	2.0
121	flag_SH	1.014042	0.049549	22.0	83.0

122 rows × 5 columns

- **Bước 3:** Tính rank là Combined Feature Rank với $\text{rank} = \text{Rank}_1 + \text{Rank}_2$ và sắp xếp theo thứ tự tăng dần

```

v Compute combined feature rank

# @title Compute combined feature rank

df_rank['Rank'] = df_rank['Rank_1'] + df_rank['Rank_2']

[ ] df_rank.sort_values(by=['Rank'], ascending=[True])
    
```

	Feature	Standard_Deviation	Mean_Median_Difference	Rank_1	Rank_2	Rank
20	srv_count	1.004139	0.283214	38.0	25.0	63.0
32	dst_host_same_src_port_rate	1.000876	0.473907	52.0	12.0	64.0
8	logged_in	1.000097	0.821737	63.0	1.0	64.0
90	service_private	1.001182	0.468803	50.0	14.0	64.0
19	count	1.000297	0.601789	60.0	6.0	66.0
...
12	num_root	0.561610	0.011267	120.0	108.0	228.0
103	service_tim_i	0.944918	0.008669	117.0	113.0	230.0
9	num_compromised	0.527079	0.010501	121.0	109.0	230.0
91	service_red_i	0.968250	0.006881	116.0	115.0	231.0
16	num_outbound_cmds	0.000000	0.000000	122.0	122.0	244.0

122 rows x 6 columns

Kết quả giảm tổng số feature thu được thể hiện trong bảng sau:

	Tác giả	Nhóm thực hiện lại
NSL-KDD	21/41	16/41
UNSW_NB-15	21/42	8/42
CIC-IDS-2017	64/79	3/79

Tuy nhiên kết quả của nhóm trong bảng trên không ổn định, kết quả thu được bên trên là một lần chạy mẫu có kết quả khá tốt.

C.1.4. Deep neural network

Mô hình DNN sử dụng được thể hiện trong hình 4 và chi tiết về tham số cài đặt cho DNN được thể hiện ở bảng 3, với input layer có kích thước bằng với số lượng feature sau khi sử dụng feature selection, ba lớp hidden layer (fully connected dense) với số lượng nơ-ron khác nhau để chuyển đổi và học dữ liệu, cuối cùng là một lớp đầu ra có một nơ-ron để phân loại nhị phân – normal hay attack.

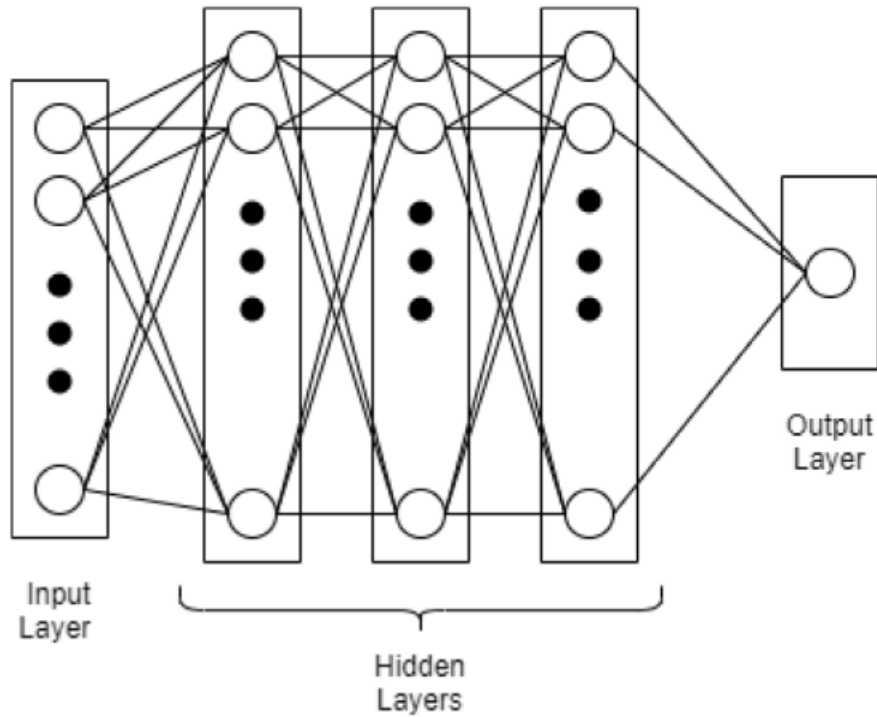


Fig. 4. Deep Neural Network.

Table 3

Neural network architecture and configuration details [4].

Criteria	Values
Model	Sequential
Number of hidden layers [4]	3
Size of input	NSL-KDD: 21, UNSW_NB: 21, CIC-IDS-2017: 64
Number of neurons in hidden layers [4]	1024, 768, 512
Activation function for hidden layer [4]	ReLU
Activation function for output layer [4]	Sigmoid
Dropout techniques	Standard dropout ($p = 0.1$) (Derived using GridSearchCV)
Batch-size [4]	1024
Epochs [4]	300


```

v define DNN model

# @title define DNN model

from keras import layers

def dnn_model(input_dim):

    model = Sequential()
    model.add(layers.Input(shape=(input_dim,), batch_size=1024, name="Input-Layer"))

    # 3 Hidden layers
    model.add(Dense(1024, activation='relu'))
    model.add(Dropout(0.1))
    model.add(Dense(768, activation='relu'))
    model.add(Dropout(0.1))
    model.add(Dense(512, activation='relu'))
    model.add(Dropout(0.1))

    # Output layer
    model.add(Dense(1, activation='sigmoid'))

    # Compile the model
    model.compile(
        optimizer=keras.optimizers.Adam(learning_rate=1e-3),
        loss=keras.losses.BinaryCrossentropy(),
        metrics=[keras.metrics.BinaryAccuracy()],
    )
    return model

model = dnn_model(input_dim = X_train.shape[1])
model.summary()
    
```

Hình: Định nghĩa mô hình DNN

Model: "sequential_7"

Layer (type)	Output Shape	Param #
dense_28 (Dense)	(1024, 1024)	125952
dropout_21 (Dropout)	(1024, 1024)	0
dense_29 (Dense)	(1024, 768)	787200
dropout_22 (Dropout)	(1024, 768)	0
dense_30 (Dense)	(1024, 512)	393728
dropout_23 (Dropout)	(1024, 512)	0
dense_31 (Dense)	(1024, 1)	513

=====
 Total params: 1307393 (4.99 MB)
 Trainable params: 1307393 (4.99 MB)
 Non-trainable params: 0 (0.00 Byte)

Hình: Summary Model

Code training model, sử dụng ModelCheckpoint để lưu lại trọng số tốt nhất, và EarlyStopping để kết thúc sớm quá trình training sau số epoch nhất định mà kết quả không cải thiện.

Hình (bên phải): Shape của mô hình DNN

```
%%time

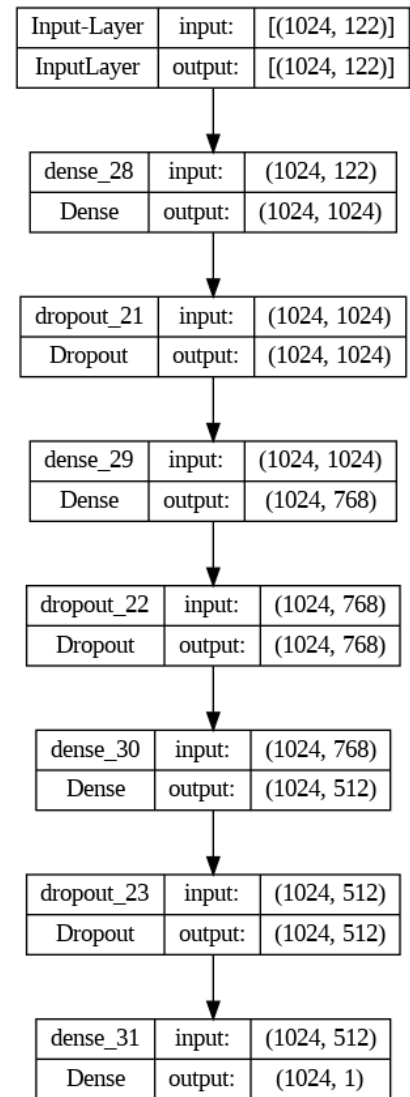
best_acc = -1
best_model = None
best_fts = []
accuracy_history = []

for i in range(1, len(X_train.columns) + 1):

    # Get i feature from high to low
    top_features = df_rank.nsmallest(i, 'Rank')
    print()
    print('-' * 30, 'Train model with', i, 'Feature', ('-' * 30))
    print("List index feature: ", top_features.index)

    # Get data from i feature
    column_names = X_train.columns
    top_feature_names = column_names[top_features.index]
    X_train_dnn = X_train[top_feature_names]
    X_test_dnn = X_test[top_feature_names]

    # Initial Base model
    checkpoint = ModelCheckpoint(
        filepath.format(i),
        monitor='val_loss',
        save_best_only=True,
        verbose=0
    )
    early_stopping = EarlyStopping(
        monitor='val_loss',
        patience=30,
        restore_best_weights=True
    )
```



Hình: Code thực hiện huấn luyện mô hình dựa trên thuật toán của bài báo

```
model = dnn_model(input_dim=X_train_dnn.shape[1])
print(' ' * 34, 'Start training')
model.fit(
    X_train_dnn, y_train,
    epochs=300, batch_size=1024,
    validation_split=0.3,
    callbacks=[checkpoint, early_stopping],
    verbose=0
)

# Evaluate model
model = keras.models.load_model(filepath.format(i))
dnn_scores = model.evaluate(X_test_dnn, y_test)

# Check model score to stop training
accuracy_history.append(dnn_scores[1])
if dnn_scores[1] >= best_acc:
    if best_model is not None:
        del best_model # Delete the previous best model

    best_acc = dnn_scores[1]
    best_model = model
    best_fts = top_features.index
    del model

elif dnn_scores[1] < best_acc:
    break

print()
print("Best current accuracy: ", best_acc)
print("Num fts: ", len(best_fts))
print("Index: ", best_fts)
print("Features: ", X_train.columns[best_fts])
```

Hình: Code thực hiện huấn luyện mô hình dựa trên thuật toán của bài báo (tt)

```
----- Train model with 64 features -----
List index feature: Int64Index([16, 66,  4, 17, 21, 63, 20,  1, 12, 40, 41, 54, 69, 68,  2,  5, 47,
 48, 52, 62, 65, 39, 70, 72, 26, 13, 71, 74, 76,  0, 73, 46, 10, 23,
  8, 18, 25, 53, 77,  3, 64, 34, 55, 22, 29, 42, 24, 30, 44, 38, 36,
  6, 75, 35, 19,  9, 11, 27, 28, 51,  7, 67, 37, 43],
 dtype='int64')
          Start training

Epoch 1: val_loss improved from inf to 0.00402, saving model to best-weights-64-fts.h5
Epoch 2: val_loss did not improve from 0.00402
Epoch 3: val_loss improved from 0.00402 to 0.00240, saving model to best-weights-64-fts.h5
Epoch 4: val_loss improved from 0.00240 to 0.00213, saving model to best-weights-64-fts.h5
Epoch 5: val_loss did not improve from 0.00213
Epoch 6: val_loss did not improve from 0.00213
Epoch 7: val_loss improved from 0.00213 to 0.00201, saving model to best-weights-64-fts.h5
Epoch 8: val_loss improved from 0.00201 to 0.00190, saving model to best-weights-64-fts.h5
Epoch 9: val_loss did not improve from 0.00190
Epoch 10: val_loss improved from 0.00190 to 0.00180, saving model to best-weights-64-fts.h5
Epoch 11: val_loss did not improve from 0.00180
Epoch 12: val_loss improved from 0.00180 to 0.00165, saving model to best-weights-64-fts.h5
Epoch 13: val_loss improved from 0.00165 to 0.00143, saving model to best-weights-64-fts.h5
Epoch 14: val_loss improved from 0.00143 to 0.00135, saving model to best-weights-64-fts.h5
Epoch 15: val_loss did not improve from 0.00135
```

Hình: Code thực hiện huấn luyện mô hình dựa trên thuật toán của bài báo (tt)

```

Epoch 58: val_loss did not improve from 0.00102
Epoch 59: val_loss did not improve from 0.00102
Epoch 60: val_loss did not improve from 0.00102
Epoch 61: val_loss did not improve from 0.00102
Epoch 62: val_loss did not improve from 0.00102
Epoch 63: val_loss did not improve from 0.00102
Epoch 64: val_loss did not improve from 0.00102
Epoch 65: val_loss did not improve from 0.00102
Epoch 66: val_loss did not improve from 0.00102
Epoch 67: val_loss did not improve from 0.00102
Epoch 68: val_loss did not improve from 0.00102
Epoch 69: val_loss did not improve from 0.00102
Epoch 70: val_loss did not improve from 0.00102
Epoch 71: val_loss did not improve from 0.00102

Epoch 72: val_loss did not improve from 0.00102
Restoring model weights from the end of the best epoch: 42.
Epoch 72: early stopping
1411/1411 [=====] - 4s 3ms/step - loss: 0.0012 - binary_accuracy: 0.9998
CPU times: user 1min 5s, sys: 3.69 s, total: 1min 9s
Wall time: 1min 27s

```

Hình: Code thực hiện huấn luyện mô hình dựa trên thuật toán của bài báo (tt)

```

Best current accuracy: 0.9726636409759521
Num fts: 16
Index: Int64Index([20, 8, 32, 90, 19, 31, 56, 39, 65, 112, 4, 28, 40, 120, 18, 47], dtype='int64')
Features: Index(['srv_count', 'logged_in', 'dst_host_same_src_port_rate',
                'service_private', 'count', 'dst_host_diff_srv_rate', 'service_ecr_i',
                'protocol_type_tcp', 'service_http', 'flag_REJ', 'wrong_fragment',
                'dst_host_count', 'protocol_type_udp', 'flag_SF', 'is_guest_login',
                'service_courier'],
                dtype='object')
Save Best model
Saved successfully!!!
CPU times: user 26min 11s, sys: 1min 38s, total: 27min 49s
Wall time: 33min 4s

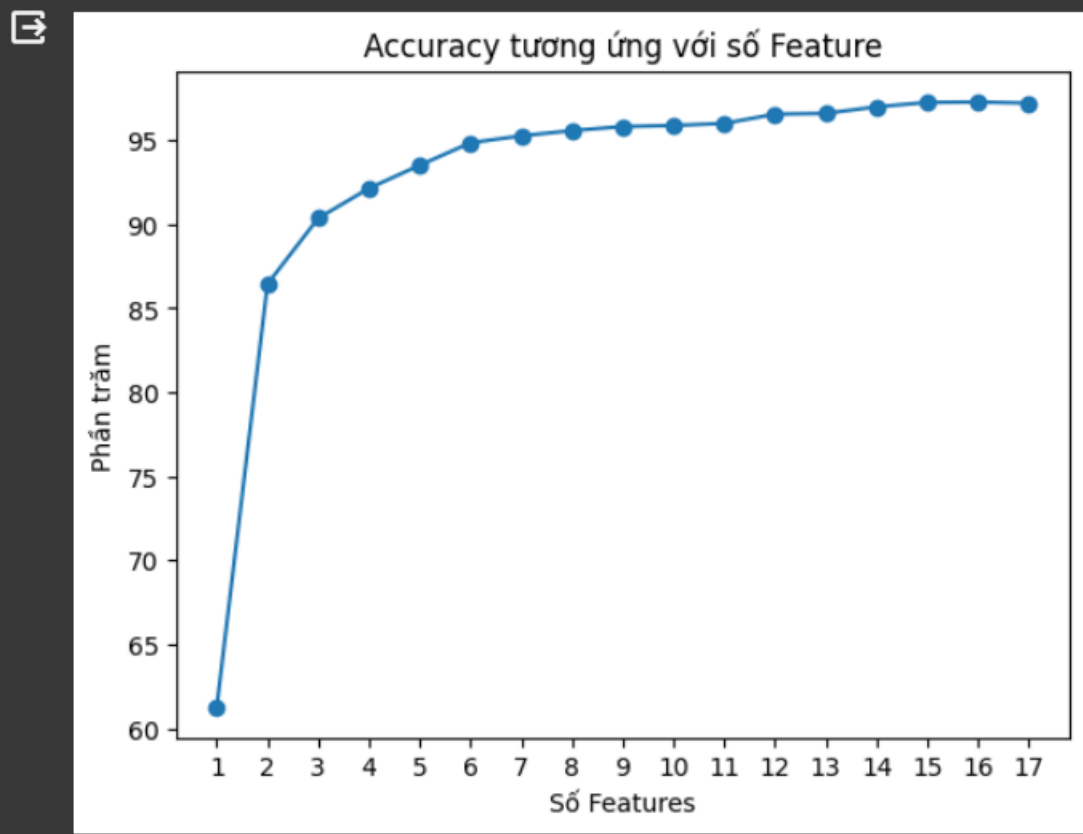
```

Hình: Kết quả chạy thuật toán với dataset NSL-KDD

```
import matplotlib.pyplot as plt

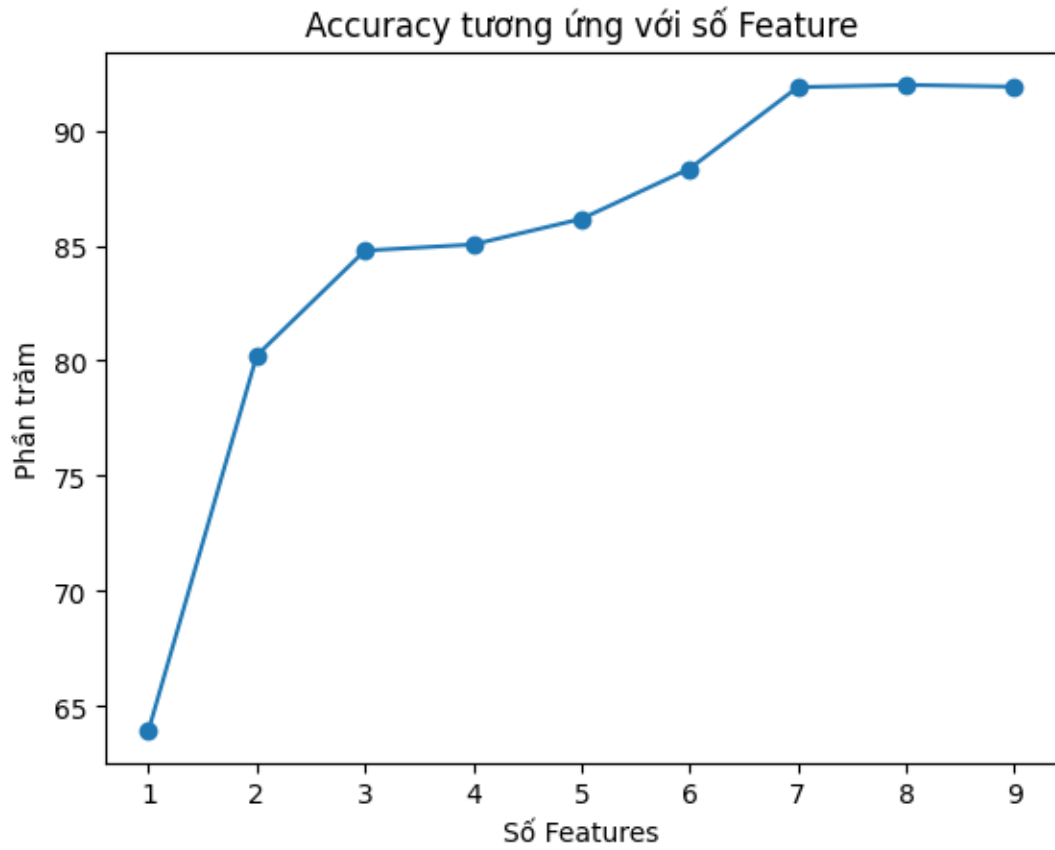
num_features = list(range(1, len(accuracy_history)+1))
acc_his = [x * 100 for x in accuracy_history]

plt.plot(num_features, acc_his, marker='o', linestyle='-')
plt.xlabel('Số Features')
plt.ylabel('Phần trăm')
plt.title('Accuracy tương ứng với số Feature')
plt.xticks(num_features)
# plt.figure(figsize=(20, 10))
plt.show()
```

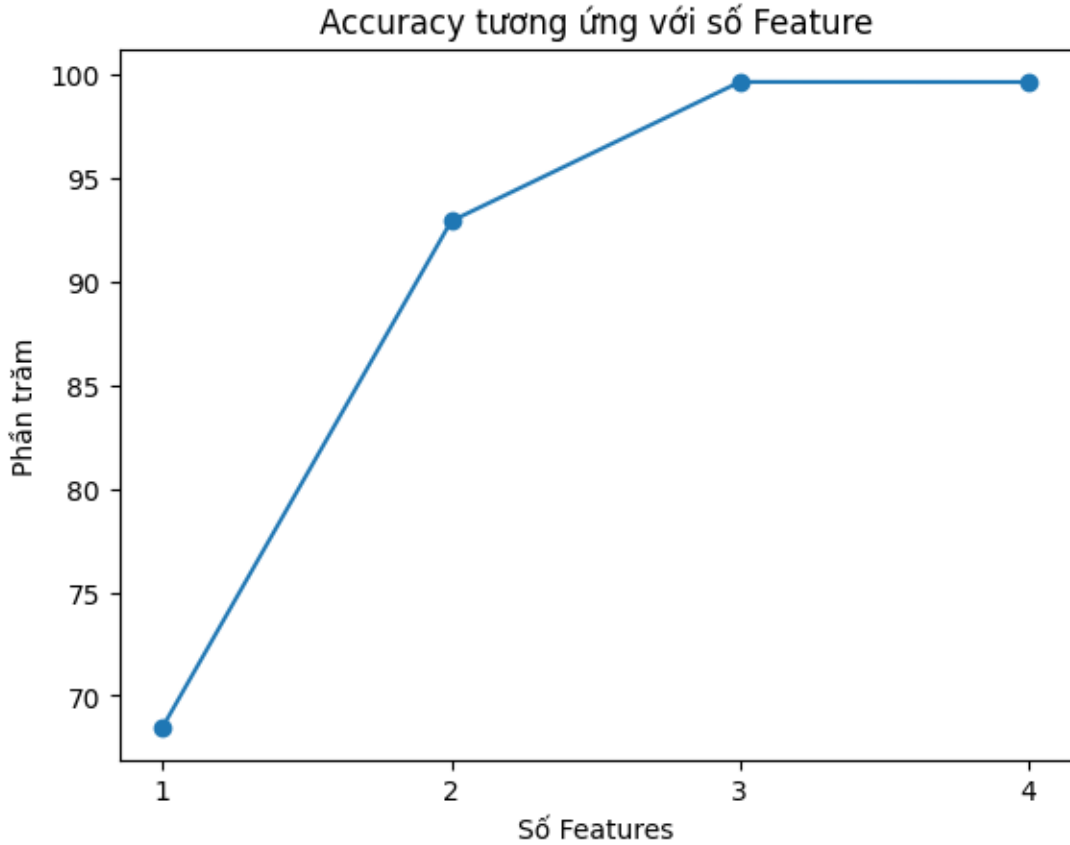


Hình: Kết quả chạy thuật toán với dataset NSL-KDD (tt)

Các dataset khác thực hiện tương tự.



Hình 2. Kết quả chạy thuật toán với dataset UNSW_NB15



Hình 3. Kết quả chạy thuật toán với dataset CIC-IDS-2017

C.1.5. Performance evaluation

Hiệu suất của mô hình được đánh giá với các chỉ số độ accuracy, precision, recall, f -score, False Positive Rate (FPR) và thời gian thực thi

$$Accuracy = \frac{T_p + T_n}{T_p + F_p + F_n + T_n} \quad (3)$$

$$Precision = \frac{T_p}{T_p + F_p} \quad (4)$$

$$Recall = \frac{T_p}{T_p + F_n} \quad (5)$$

$$f - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (6)$$

$$FPR = \frac{F_p}{F_p + T_n} \quad (7)$$

Code thực hiện:

```
#####
cm = confusion_matrix(y_test, y_pred_binary)

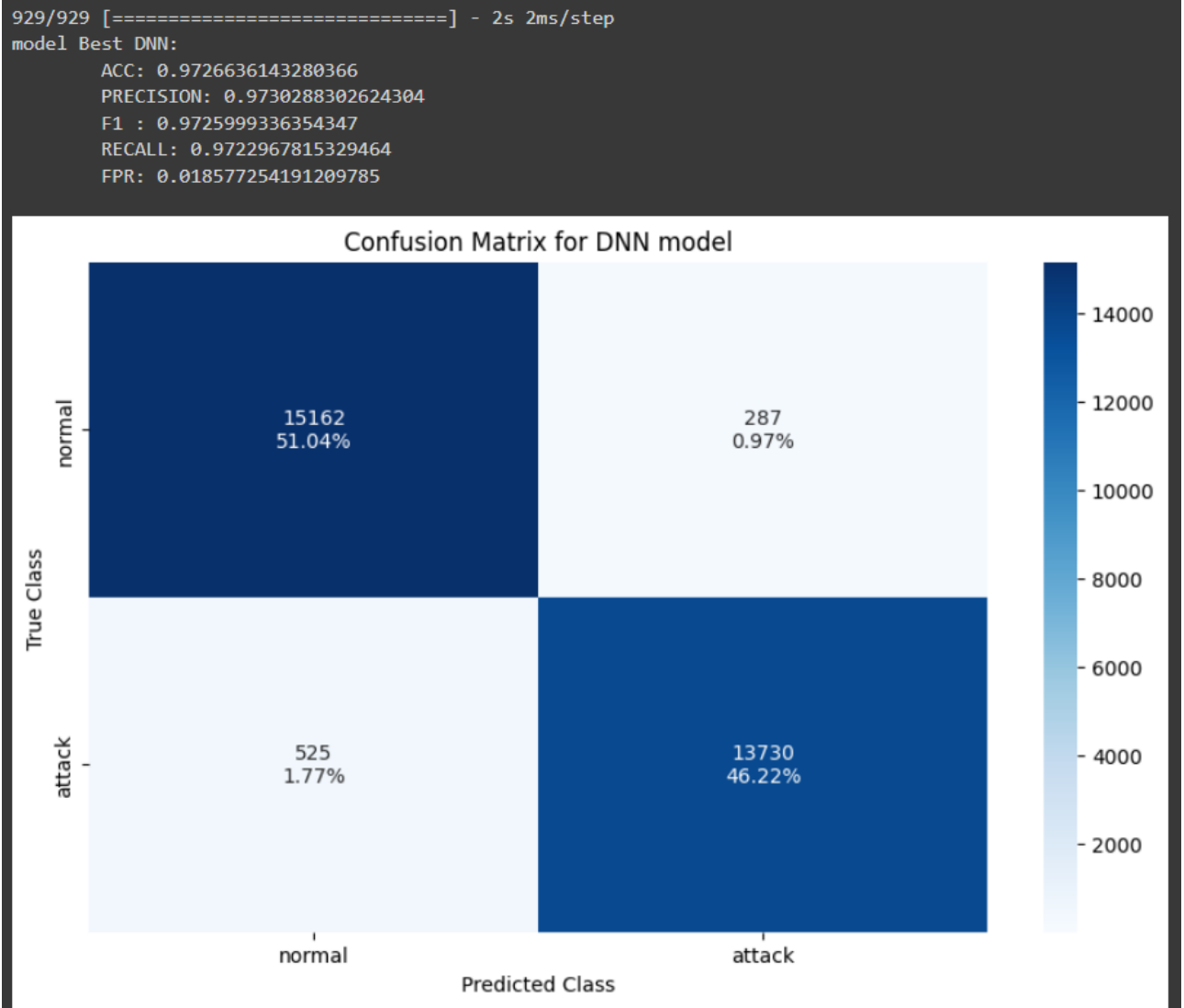
acc = accuracy_score(y_test, y_pred_binary)
precision = precision_score(y_test, y_pred_binary, average='macro')
f1 = f1_score(y_test, y_pred_binary, average='macro')
recall = recall_score(y_test, y_pred_binary, average='macro')
fpr = (cm[0, 1] / (cm[0, 0] + cm[0, 1]))

model_pred['n_fts'] = y_pred_binary
model_acc['n_fts'] = acc
model_ppv['n_fts'] = precision
model_f1['n_fts'] = f1
model_tpr['n_fts'] = recall
model_fpr['n_fts'] = fpr

print('model Best DNN:')
print(f'\tACC: {acc}')
print(f'\tPRECISION: {precision}')
print(f'\tF1 : {f1}')
print(f'\tRECALL: {recall}')
print(f'\tFPR: {fpr}')
print()

#####
```

Hình: Code tính các giá trị đánh giá model



Hình: Kết quả đánh giá + confusion matrix

	precision	recall	f1-score	support
0	0.97	0.98	0.97	15449
1	0.98	0.96	0.97	14255
accuracy			0.97	29704
macro avg	0.97	0.97	0.97	29704
weighted avg	0.97	0.97	0.97	29704

Hình: Classification report

Result table của NSL-KDD với kết quả 3 trường hợp:

- Chạy thuật toán minh họa đề xuất trong bài báo (dòng n_fts)

- Chạy thuật toán với số feature bằng với số feature tác giả là 21 feature (21_fts)
- Chạy thuật toán với toàn bộ dataset (all_fts)

Result table

```
[ ] summary_df = pd.DataFrame({
    'ACC': model_acc.values(),
    'PRECISION': model_ppv.values(),
    'F1': model_f1.values(),
    'RECALL': model_tpr.values(),
    'FPR': model_fpr.values(),
}, index=model_acc.keys())

# metrics_df_sorted = summary_df.sort_values(by=['TPR', 'FPR'], ascending=[False, True])
# metrics_df_sorted['Rank'] = range(1, len(metrics_df_sorted) + 1)
summary_df#.round(4)
```

	ACC	PRECISION	F1	RECALL	FPR
n_fts	0.972664	0.973029	0.972600	0.972297	0.018577
21_fts	0.982326	0.982333	0.982296	0.982260	0.016118
all_fts	0.991752	0.991705	0.991739	0.991775	0.008803

Hình: Tổng hợp kết quả với dataset NSL_KDD

UNSW_NB15

	ACC	PRECISION	F1	RECALL	FPR
n_fts	0.919686	0.913448	0.912853	0.912270	0.114436
21_fts	0.936121	0.929560	0.931030	0.932583	0.080159
all_fts	0.936703	0.931856	0.931334	0.930820	0.090367

Hình: Tổng hợp kết quả với dataset UNSW_NB15

CIC-IDS-2017

	ACC	PRECISION	F1	RECALL	FPR
n_fts	0.996567	0.996384	0.996499	0.996615	0.003040
64_fts	0.999756	0.999761	0.999751	0.999742	0.000361
all_fts	0.999756	0.999774	0.999751	0.999729	0.000464

Hình: Tổng hợp kết quả với dataset CIC-IDS-2017

C.2. So sánh, đánh giá

Kết quả tổng hợp cuối cùng được cho trong các bảng sau:

Bảng kết quả triển khai với dataset NSL-KDD

NSL-KDD	ACC	PRECISION	F1	RECALL	FPR	Execution time (s)
16_fts	97,27%	97,30%	97,26%	97,23%	0,0186	1.988,435
21_fts	98,23%	98,23%	98,23%	98,23%	0,0161	99,707
all_fts (41->122)	99,18%	99,17%	99,17%	99,18%	0,0088	58,309
PAPER (21_fts)	99,84%	99,94%	99,37%	98,81%	0,0110	22.318,015

Bảng kết quả triển khai với dataset UNSW_NB15

UNSW_NB15	ACC	PRECISION	F1	RECALL	FPR	Execution time (s)
8_fts	91,97%	91,34%	91,29%	91,23%	0,1144	1.182,227
21_fts	93,61%	92,96%	93,10%	93,26%	0,0802	75,745
all_fts (42->196)	93,67%	93,19%	93,13%	93,08%	0,0904	75,508
PAPER (21_fts)	89,03%	95,00%	96,93%	98,95%	0,0110	13.913,500

Bảng kết quả triển khai với dataset CIC-IDS-2017

CIC-IDS-2017	ACC	PRECISION	F1	RECALL	FPR	Execution time (s)
3_fts	99,66%	99,64%	99,65%	99,66%	0,0030	506,502
64_fts	99,98%	99,98%	99,98%	99,97%	0,0004	108,366
all_fts (78->76)	99,98%	99,98%	99,98%	99,97%	0,0005	126,455
PAPER (64_fts)	99,80%	99,85%	99,89%	99,94%	0,0120	27.719,360

Nhận xét: Nhìn chung phương pháp feature selection được đề xuất hoạt động khá ổn, đặc biệt trong trường hợp của CIC-IDS-2017, chỉ với 3/78 feature, kết quả chính xác đạt được khi phân loại lại khá ấn tượng với các giá trị đều cao hơn 99,6%.

Để kiểm tra thêm tính đúng đắn của phương pháp được đề xuất, nhóm thực hiện so sánh kết quả với một phương pháp feature selection khác trên **medium**:

<https://towardsdatascience.com/3-step-feature-selection-guide-in-sklearn-to-supercharge-your-models-e994aa50c6d2>

Giới thiệu sơ qua về phương pháp này, với tên gọi *3-Step Feature Selection*, nó gồm 3 bước cơ bản:

- Variance Threshold feature selection: loại bỏ feature có phương sai thấp.

- Pairwise Correlation feature selection: chỉ giữ lại 1 trong tập hợp những feature có mối liên hệ với nhau cao, nghĩa là khi biết giá trị của feature này có thể suy ra giá trị của những feature còn lại trong nhóm.
- Recursive Feature Elimination with Cross-Validation (RFECV): kết hợp Recursive Feature Elimination và Cross-Validation với Random Forests được sử dụng làm underlying model để xác định số lượng tính năng tối ưu cho một mô hình nhất định.

Kết quả thực hiện khi thay thế phương pháp đề xuất trong bài báo bằng phương pháp feature selection trên **medium** được cho trong các bảng sau:

Bảng kết quả triển khai với dataset NSL-KDD

NSL-KDD	ACC	PRECISION	F1	RECALL	FPR
16_fts	91.26%	91.71%	91.20%	91.06%	0.04075
21_fts	94.93%	95.35%	94.90%	94.76%	0.00996
all_fts	99.02%	99.01%	99.02%	99.02%	0.0108

Bảng kết quả triển khai với dataset UNSW_NB15

UNSW_NB15	ACC	PRECISION	F1	RECALL	FPR
8_fts	90.08%	90.70%	88.88%	87.70%	0.208671
21_fts	92.30%	91.58%	91.67%	91.77%	0.101381
all_fts	93.79%	93.36%	93.26%	93.16%	0.09128

Bảng kết quả triển khai với dataset CIC-IDS-2017

CIC-IDS-2017	ACC	PRECISION	F1	RECALL	FPR
3_fts	94.45%	94.17%	94.37%	94.73%	0.032363
64_fts	99.93%	99.93%	99.93%	99.93%	0.000567
all_fts	99.97%	99.97%	99.97%	99.97%	0.000412

So sánh với kết quả của tác giả:

NSL-KDD	ACC	PRECISION	F1	RECALL	FPR
3-Step Feature Selection	94.93%	95.35%	94.90%	94.76%	0.00996
Proposed Feature Selection	99,84%	99,94%	99,37%	98,81%	0,0110

UNSW_NB15	ACC	PRECISION	F1	RECALL	FPR
3-Step Feature Selection	92.30%	91.58%	91.67%	91.77%	0.101381
Proposed Feature Selection	89,03%	95,00%	96,93%	98,95%	0,0110

CIC-IDS-2017	ACC	PRECISION	F1	RECALL	FPR
3-Step Feature Selection	99.93%	99.93%	99.93%	99.93%	0.000567
Proposed Feature Selection	99,80%	99,85%	99,89%	99,94%	0,0120

➔ Phương pháp feature selection được tác giả đề xuất có kết quả cao hơn ở NSL-KDD và UNSW_NB15 nhưng thấp hơn với CIC-IDS-2017

So sánh với kết quả nhóm thực hiện:

NSL-KDD	ACC	PRECISION	F1	RECALL	FPR
3-Step Feature Selection	91.26%	91.71%	91.20%	91.06%	0.04075
Proposed Feature Selection	97,27%	97,30%	97,26%	97,23%	0,0186

UNSW_NB15	ACC	PRECISION	F1	RECALL	FPR
3-Step Feature Selection	90.08%	90.70%	88.88%	87.70%	0.208671
Proposed Feature Selection	91,97%	91,34%	91,29%	91,23%	0,1144

<i>CIC-IDS-2017</i>	<i>ACC</i>	<i>PRECISION</i>	<i>F1</i>	<i>RECALL</i>	<i>FPR</i>
<i>3-Step Feature Selection</i>	94.45%	94.17%	94.37%	94.73%	0.032363
<i>Proposed Feature Selection</i>	99,66%	99,64%	99,65%	99,66%	0,0030

➔ Phương pháp đề xuất cho kết quả tốt hơn ở cả 3 dataset

D. HƯỚNG PHÁT TRIỂN

Hướng phát triển được các tác giả đưa ra: Xem xét phân tích khả năng phục hồi của IDS bằng cách tối ưu hóa kiến trúc mạng neural network bằng thuật toán “nature-inspired” hoặc bằng cách sử dụng thuật toán “nature-inspired algorithms” làm kỹ thuật feature selection.

Còn về phía nhóm, nếu có thêm thời gian, nhóm dự định kiểm tra kết quả của phương pháp được đề xuất với những ngữ cảnh khác, kết hợp thêm các phương pháp tiền xử lý cùng các thuật toán khác dùng để training model và đánh giá kết quả...

Bài báo cáo của nhóm em đến đây là hết. Nhóm em xin cảm ơn thầy cô và các bạn đã đọc.

HẾT