ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA KHOA HỌC VÀ KĨ THUẬT THÔNG TIN

BÁO CÁO ĐỒ ÁN CUỐI KÌ

IE105 - NHẬP MÔN BẢO ĐẢM VÀ AN NINH THÔNG TIN

ĐỀ TÀI:

INTRUSION DETECTION BY USING DEEP LEARNING

Nhóm sinh viên thực hiện

Nguyễn Phúc Khang - 21522194

Đỗ Nguyễn Anh Khoa - 21522219

Giảng viên hướng dẫn: **TS. Nguyễn Tấn Cầm**

MỤC LỤC

MỤC LỤC BẢNG

MỤC LỤC HÌNH ẢNH

LỜI MỞ ĐẦU

Trong thời kí phát triên của công nghệ thông tin, cùng với đó các mối đe doạn an ninh ngày càng gia tăng, gây ra vô số các thiệt hại đáng kể của các tổ chức, các nhân và doanh nghiệp, không chỉ gây ảnh hưởng về kinh tế mà nó còn ảnh hưởng đến thông tin thậm chí là an ninh quốc gia.

Để bảo đảm dự liệu không bị rò rỉ, các tổ chức, doanh nghiệp cần triển khai các phương phát bảo vệ an ninh hiệu quả. Một trong những giải pháp đã và đang được sử dụng rộng rải nhất hiện nay là phát hiện xâm nhập bằng học xâu (Instusion Detection by using Deep Learning).

Học sâu (Deep Learning) là một phương pháp của trí tuệ nhân tạo (AI) cho phép máy tính có khả năng học hỏi từ dữ liệu đã có sẳn mà không cần lập trình rõ ràng, mang lại hiệu quả rất cao và vô cùng nhanh chóng. So với các phương pháp phát hiện xâm nhập truyền thống thì phát hiện xâm nhập bằng máy học có nhiều ưu điểm nổi bật hơn hoàn toàn. Trong đồ án này, chúng em sẽ nghiên cứu về phát hiện xâm nhập bằng học sâu. Cụ thể chúng em sẽ xây dựng các hệ phát thống phát hiện xâm nhập bằng các sử dụng mạng Neuron Network (NN), Convolutional Neural Network (CNN) và Long – short Term Memory (LSTM). Kết quả của đồ án này có thể giúp nâng cao hiệu quả bảo vệ dữ liệu khỏi các cuộc tấn công nhằm cho các mục đích xấu.

Cuối cúng nhóm chúng em dành lời cảm ơn chân thành nhất đến TS. Nguyễn Tấn Cầm – người đã trực tiếp giảng dạy và hỗ trợ nhóm chúng em trong quá trình thực hiện đồ án. Tuy nhiên, do vốn kiến thức của nhóm còn nhiều hạn chế nên mặc dù đã cố gắng hết sức nhưng chắc chắn đồ án khó có thể tránh những điểm thiếu sót và nhiều chổ chưa chính xác, chúng em kính mong thầy xem xét và góp ý để đồ án nhóm chúng em được hoàn thiện hơn. Hơn thế nữa, nhóm chúng em hi vọng có thể phát triển đồ án môn học lên Paper Nghiên cứu Khoa học dưới sự hướng dẫn của thầy Nguyễn Tấn Cầm.

Lời cuối, chúng em xin kính chúc thầy sức khỏe, thành công và hạnh phúc trong sự nghiệp nhà giáo của mình

Chúng em chân thành cảm ơn thầy

Thành phố Hồ Chí Minh, ngày 24 tháng 11 năm 2023

Nhóm thực hiện

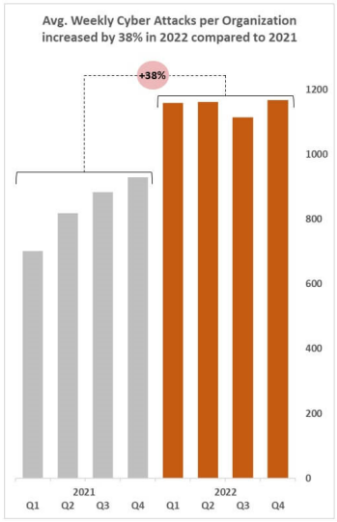
TÓM TẮT

- Cùng với sự phát triển mạnh mẽ của công nghệ thông tin và cơ sở hạ tầng công nghệ như điện toán đám mây, hệ thống xe cột, Internet vạn vật vất nối (IoT), v.v , lượng thông tin được truyền gửi ngày càng tăng theo cấp số nhân. Bên cạnh đó cũng là mảnh đất màu mỡ cho các đối tượng xấu với mục đích đánh cắp dữ liệu, tài nguyên sử dụng cho mục đích xấu. Vì vậy, việc nâng cao tính bảo mật của hệ thống mạng trở nên cấp thiết hơn bao giờ hết. Việc phát hiện và phân loại các thư mục độc hại để ngăn ngừa và tránh tấn công là vô cùng quan trọng. Trong nghiên cứu này, chúng em đề xuất hệ thống phát hiện xâm nhập (Intrusion Detection System - IDS) sử dụng các mô hình kĩ thuật Deep Learning (DL) cho phép trích xuất các đặc trưng của gói tin, từ các đặc trưng này được sử dụng làm đầu vào để đào tạo các model nhằm phát hiện và phân loại file độc hại với độ chính xác cao nhất
- Bộ dữ liệu chúng em sử dụng là NSL-KDD, là một dữ liệu đã được tinh chỉnh từ bộ dữ liệu KDD99, bộ dữ liệu đã được sử dụng cho các cuộc thi quốc tế về công cụ khai thác tri thức và khai phá dữ liệu (KKDCup – 1999) và đã đạt được những kết quả nhất định.

1. GIỚI THIỆU

1.1. Bối cảnh

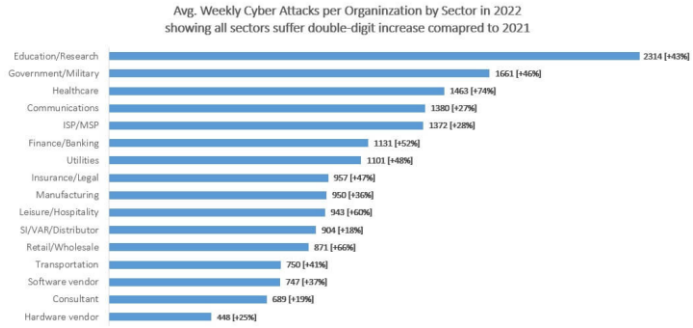
- Theo Check Point Rearch (CPR), số lượng các cuộc tấn công mạng toàn cầu trong 2022 đã tăng 38% sao với năm 2021.



Hình 1.1 : Số lượng các vụ tấn công theo quý của năm 2021 so với 2022

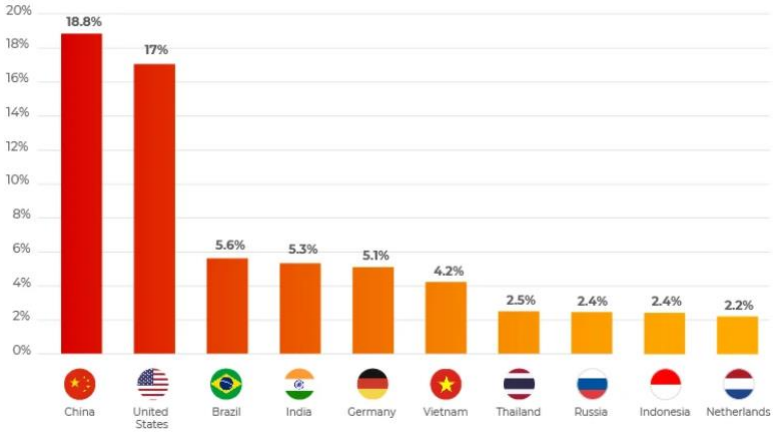
- Lĩnh vực giáo dục và nghiên cứu là lĩnh vực ghi nhận số lượng tấn công nhiều nhất, số lượng các cuộc tấn công tăng 43% so với năm 2021. Trung bình ghi nhận 2.314 cuộc tấn công mỗi tuần. Vì sự chuyển dịch của mô hình học tập trực tuyến ngày càng phát triển, nhiều cơ sở giáo dục chưa chú trọng nhiều vào vấn đề bảo mật dẫn đến tạo nên mãnh đất màu mỡ cho các hacker xâm nhập vào mạng thông qua nhiều phương tiện khác nhau. Việc mọi người học tập từ khắp

nơi, thông qua các thiết bị riêng và Wifi công cộng đã tạo cơ hội cho các hacker thực hiện các mưa đồ xấu.



Hình 1.2 : Số các vụ tấn công ghi nhận trong các lĩnh vực trong năm 2022

- Cung với sự bùng nổ của trào lưu AI trong năm 2023, đặt biệt là ChatGPT, các thông tin cá nhân hay doanh nghiệp nếu không được che dấu và bảo mật tốt sẽ dễ dàng bị đánh cắp và khai thác phục vụ cho các mục đích xấu.
- Đáng nguy ngại hơn, Việt Nam cũng là nước nằm trong 10 nước có số lượng tấn công mạng nhiều nhất thế giới theo CyberProof .



Hình 1.3 : Top 10 quốc gia có số lượng tấn công mạng lớn nhất

1.2. Phương hướng giải quyết

- Tuy nhiên những công nghệ như AI hoặc Deep Learning cũng có thể được dùng để tổng hợp và phân tích những dữ liệu để có thể dự đoán được các mối đe dọa. Đó cũng là phương hướng mà nhóm chúng em đã và đang thực hiện, nhằm mang lại sự an toàn và tránh các cuộc tấn công mạng không mong muốn.
- Trong bài báo cáo này, chúng em đã sử dụng ngôn ngữ lập trình python và các thư viện hỗ trợ như numpy, pandas, matplotlib... để làm sạch dữ liệu, trực quan hóa dư liệu, rút trích các đặc trưng cần thiết, chúng em đã thử nghiệm huấn luyện qua nhiều thuật toán máy học (CVM, RandomForestClassifier) và Deep Learning (CNN, neural network..,) nhằm tìm ra thuật toán tối ưu nhatas cho vấn đề. Chúng em đã huấn luyện qua nhiều bộ dữ liệu khác như NSL-KDD, KDD99... nhằm có được hiểu quả mô hình cao nhất.
- Bào báo cáo của nhóm chúng em gồm 5 phần chính:
    - Phần 1: Giới thiệu

- Phần 2: Nghiên cứu liên quan
- Phần 3: Hệ thống đề xuất
- Phần 4: Đánh giá
- Phần 5: Nhận xét

2. NGHIÊN CỨU LIÊN QUAN

- Việc sử dụng Deep Learning và Machine Learning trong việc phát hiện xâm nhập không phải là một chủ đề mới mà nó đã được quan tâm và nghiên cứu từ trước những năm 2000. Tuy nhiên nó vẫn là bài toán cấp thiết và đang tìm ra lời giải vì sự phát triển công của nghệ, bên cạnh đó ngày càng có nhiều mã độc ngày càng được phát triển nhiều hơn. Sau đâu là những nghiên cứu khóa học tiêu biểu trong lĩnh vực này :
    1. "A Survay on Deep Learning for instusion Detection – 2022",
    2. Deep Learning-Based Instrusion Detection: A Survay – 2021"
    3. "Deep Learning-Based Intrusion Detection: A Review – 2020"
    4. "Deep Learning-Based Intrusion Detection: A Survay and a Comparetive Study – 2019"
    5. "An Implementation of Instrusion Detection System Using Genetic Algorithm – 2012"
    6. "A Deep Learning Approach for NetWork Intrusion Detection System - 2016"
    7. "Deep Learning Approach for Intelligent Instrusion Detection System – 2018"
    8. "Importance of Instrusion Detection System (IDS) – 2011"
    9. "Intrusion Detection Systems: A Survey and Taxonomy – 2000"
    10. "The MINDS – Minnesota Instrusion Detection System – 2000"

3. HỆ THỐNG ĐỀ XUẤT

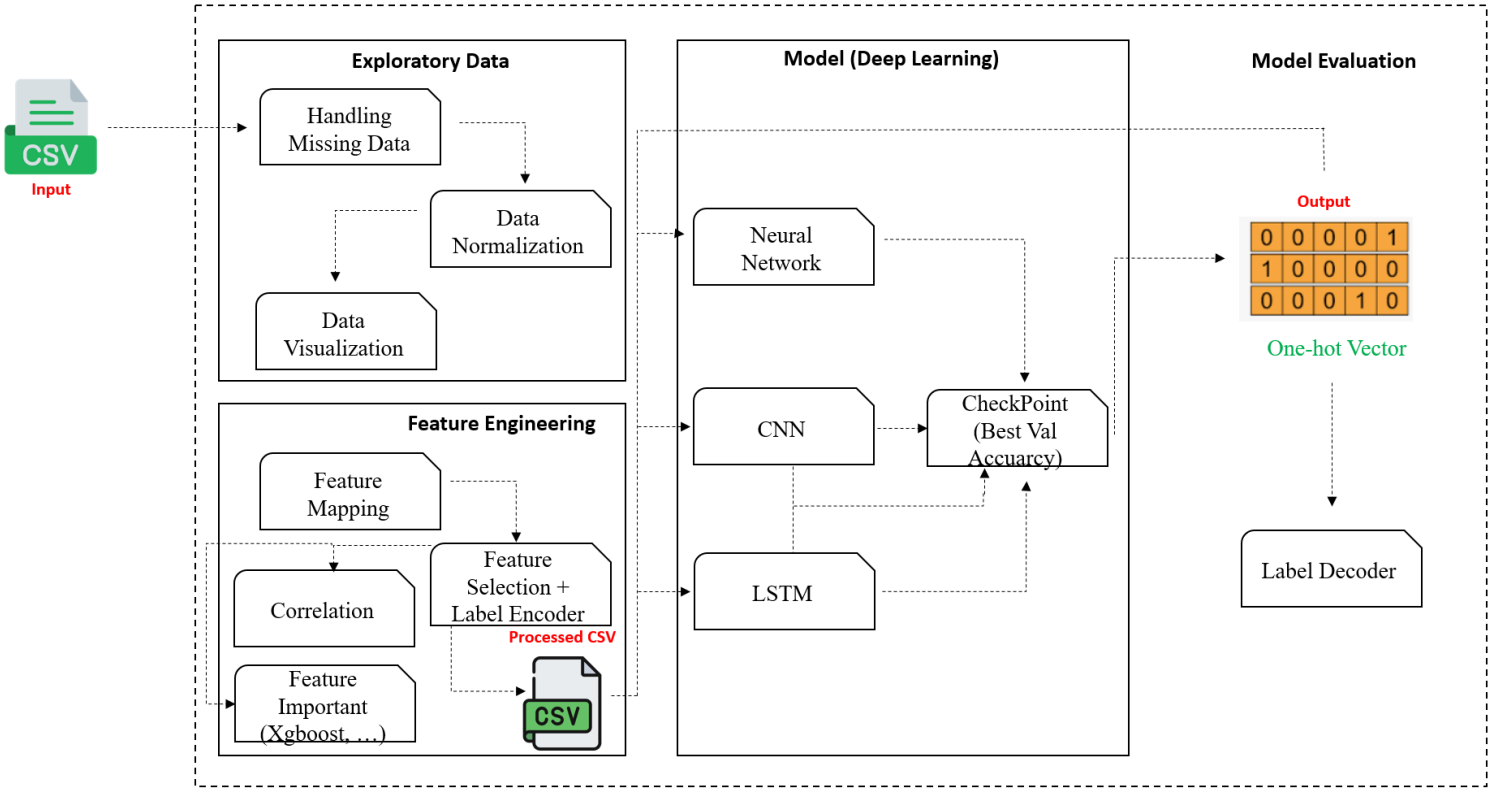
3.1. Mô tả hệ thống

Hệ thống bao gồm 4 phần chính: Exploratory Data (Khám phá dữ liệu), Feature Engineering (Các kĩ thuật đặt trưng), Model (Xây dựng mô hình Deep Learning) và Model Evaluate (Đánh giá mô hình).

- Exploratory Data (Khám phá dữ liệu) : Từ file CSV ban đầu, trước khi tiến hành kĩ thuật đặt trưng và xây dựng mô hình, nhóm em ưu tiên khám phá dữ liệu để có cái nhìn tổng quát nhất về dữ liệu từ đó lên ý tưởng cho các phần tiếp theo. Trước tiên, nhóm em tiến hành xử lý các dữ liệu thiếu (Handling Missing Data) bằng 2 cách: xem xét bỏ nếu không cần thiết hoặc điền giá trị Median của cột đó. Tiếp theo, nhóm tiến hành chuẩn hóa dữ liệu (Data Normalization) điều này đảm bảo rằng các đặc trưng có cùng phạm vi giá trị và sẽ giúp mô hình học tốt hơn và tăng tốc quá trình huấn luyện. Cuối cùng trong bước khám phá dữ liệu là trực quan hóa dữ liệu (Data Visualization) bằng thư viện Matplotlib của Python (thể hiện Data thông qua ảnh và biểu

đồ), điều này giúp chúng ta có cái nhìn dễ dàng và trực quan hơn là Data dạng số vốn dĩ khá dài và khó nhìn.

- Feature Engineering (Kĩ thuật đặt trưng) : Sau khi khám phá dữ liệu, nhóm đã nhận thấy rằng hầu hết các features (đặt trưng) đều đã là dạng Numeric (38/41) đúng với yêu cầu trước khi huấn luyện mô hình. Tuy nhiên còn khoảng tầm 3 feature vẫn là dạng Category do đó nhóm đã sử dụng Feature Mapping để map các dữ liệu dạng Category này về dạng Numeric theo đúng yêu cầu bài toán. Sau quá trình Mapping nhóm tiến hành đến bước Feature Selection (Lưa chọn đặt trưng) để chọn ra những đặt trưng quan trọng nhất cho bài toán. Nhóm em đã tiến hành 2 bước để chọn ra các đặt trưng quan trọng bao gồm: Feature Important (Xgboost & Random Forest) để chọn ra các Features quan trọng theo đánh giá của 2 phương pháp Xgboost và Random Forest, Correlation với Label chọn ra những features có sự tương quan mạnh với nhãn. Sau đó kết hợp cả 2 nhóm tìm ra được bộ CSV đã được xử lý và tối ưu trước khi qua bước tiếp theo.
- Model (Xây dựng mô hình Deep Learning) : Với phần xây dựng mô hình nhóm em ưu tiên sử dụng 4 mô hình sau bao gồm: Neural Network (NN), Convolutional Neural Network (CNN), Long – short term Memory (LSTM) và CNN + LSTM. Với mỗi mô hình nhóm sẽ dùng CheckPoint của Tensorflow để lưu lại mô hình có Val Accuracy cao nhất và lưu lại trong Colab. Sau mỗi lần train nhóm đều lưu lại để phát thảo lên Model Accuracy và Model Loss để chứng minh Model không bị Overfit và Underfit.
- Model Evaluate (Đánh giá mô hình) : Sau khi tiến hành Train Model, nhóm tiến hành đưa Data Test vào để kiểm thử và kết quả trả ra của mô hình luôn là vector one-hot với số chiều phụ thuộc vào số lượng nhãn, từ đó nhóm tiến hành đánh giá kết quả predict với tập grouth-truth dựa trên 2 độ đo quan trọng bao gồm Accuracy và F1-Score. Dựa vào 2 độ đó giúp nhóm biết được khả năng của mô hình từ đó giúp nhóm biết cách nên tỉnh chỉnh tham số đầu vào. Ngoài ra sau khi có vector one-hot nhóm có thể tiến hành đến bước Label – Decoder để chuyển số về lại thành nhãn ban đầu giúp người dùng có thể hiểu kết quả dự đoán.



Hình 3.1 Hệ thống đề xuất phát hiện xâm nhập bằng Deep Learning

3.2. Mô tả dữ liệu huấn luyện

- Bộ dữ liệu NSL-KDD là một bộ dữ liệu được Tavallaee công bố vào năm 2009, là một phiên bản được rút gọn từ bộ dữ liệu KDD Cup năm 1999 với việc loại bỏ đi một số bản ghi bị thừa, các cột không cần thiết và một số thông tin bị trùng lắp. Bộ dữ liệu được sử dụng rất nhiều trong các nghiên cứu khoa học và huấn luyện mô hình.
- Bộ dữ liệu gồm 2 tập csv là train và test
  - Tập train gồm 125973 và 41 cột
  - Tập test gồm 22544 và 41 cột
- Dưới đây là tên các thuộc tính có trong dataset

Bảng 3.1 : Các thuộc tính có trong dataset

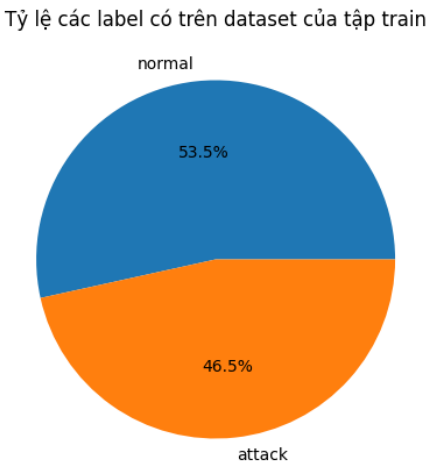
1	duration	11	num_failed_logins	21	is_host_login	31	srv_diff_host_rate
2	protocol_type	12	logged_in	22	is_guest_login	32	dst_host_count
3	service	13	num_compromised	23	count	33	dst_host_srv_count
4	flag	14	root_shell	24	srv_count	34	dst_host_same_srv_rate
5	src_bytes	15	su_attempted	25	serror_rate	35	dst_host_diff_srv_rate
6	dst_bytes	16	num_root	26	srv_serror_rate	36	dst_host_same_src_port_rate
7	land	17	num_file_creations	27	rerror_rate	37	dst_host_srv_diff_host_rate
8	wrong_fragment	18	num_shells	28	srv_rerror_rate	38	dst_host_serror_rate
9	urgent	19	num_access_files	29	same_srv_rate	39	dst_host_srv_serror_rate
10	hot	20	num_outbound_cmds	30	diff_srv_rate	40	dst_host_rerror_rate
						41	dst_host_srv_rerror_rate

- Phân loại các thuộc tính

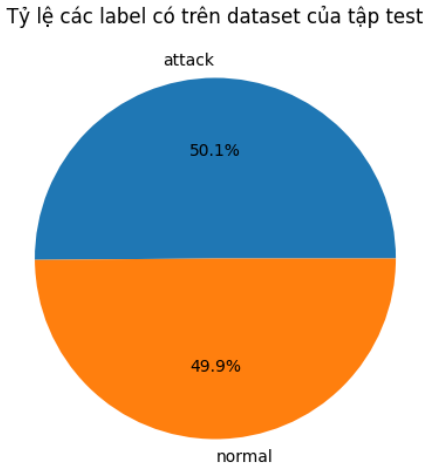
Bảng 3.2 : Gôm nhóm các thuộc tính có trong dataset

Kiểu	Thuộc tính
Catagory	Protocol_type(2), service(3), flag(4), Land(7), logged_in(12), root_shell(14), su_attempted(15), is_host_login(21),, is_guest_login(22)
Numeric	Duration(1), src_bytes(5), dst_bytes(6), wrong_fragment(8), urgent(9), hot(10), num_failed_logins(11), num_compromised(13), num_root(16), num_file_creations(17), num_shells(18), num_access_files(19), num_outbound_cmds(20), count(23) srv_count(24), serror_rate(25), srv_serror_rate(26), rerror_rate(27), srv_rerror_rate(28), same_srv_rate(29) diff_srv_rate(30), srv_diff_host_rate(31), dst_host_count(32), dst_host_srv_count(33), dst_host_same_srv_rate(34), dst_host_diff_srv_rate(35), dst_host_same_src_port_rate(36), dst_host_srv_diff_host_rate(37), dst_host_serror_rate(38), dst_host_srv_serror_rate(39), dst_host_rerror_rate(40), dst_host_srv_rerror_rate(41)

- Tập dữ liệu gồm 23 nhãn kiểu tấn công và có thêm 37 nhãn trong tập test, được rút gọn lại thành 3 trường hợp phân lớp: 2 nhãn, 5 nhãn và 8 nhãn
- Trường hợp 2 nhãn, gồm 2 nhãn Normal và Attack:
  (1) Normal : là những tập dữ liệu bình thường, không chứa mã độc
  (2) Attack : là những mà độc mang mục đích tấn công mạng



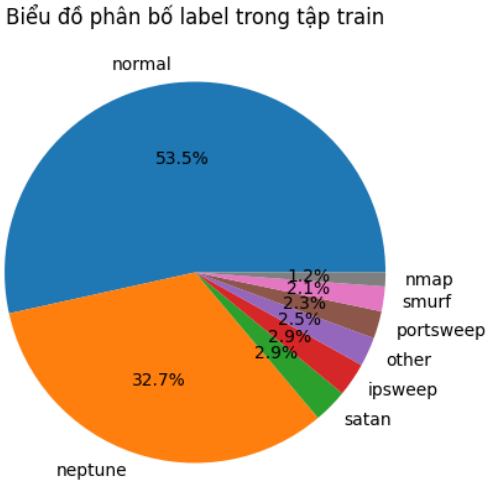
Hình 3.1 Sự phân bố nhãn normal và attack trong tập train



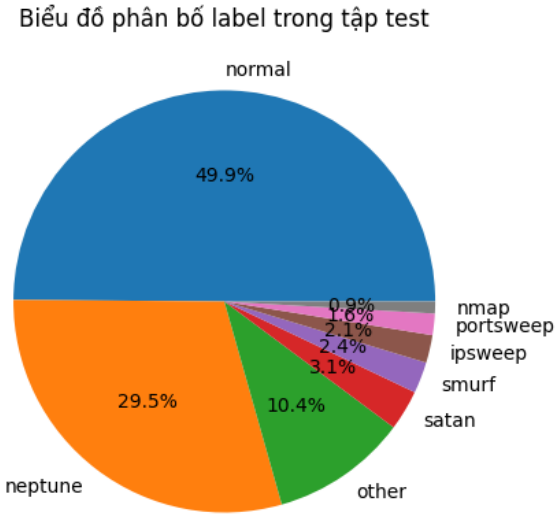
Hình 3.2 Sự phân bố nhãn normal và attack trong tập test

- Trường hợp 8 nhãn, nhận diện chính xác tên loại mã độc tấn công
  (1) Normal : Là những tệp dữ liệu bình thường, không nhằm mục đích tấn công.
  (2) Neptune : Là một phương thúc tấn công mạng (man-in-the-middle) sử dụng lỗ hỏng bảo mật trong DNS để thay đổi lưu lượng truy cập, điều này cho phép hacker tấn công thay đổi địa chỉ IP của máy chủ DNS, dẫn đến việc nạn nhân sẽ bị dẫn đến các trang web giả mạo hoặc bị nhiễm mã đọc.
  (3) Satan : Là loại mã độc nhằm mục đích đánh cắp thông tin nạn nhân (tên, sdt, mật khẩu ...), thường được phát tán qua email hay các nên tảng xã hội.
  (4) Ipsweep : Là một loại mã độc nhằm quét mạng để xác định các máu chủ hoạt động trên mạng sử dụng các phương thức kết nối hoặc ping nhằm để thăm dò máy chủ.
  (5) Portsweep : Là một lại mã đọc dùng để quét và nhận diện các cổng (port) đang mở của host. Nó hoạt động bằng cách gửi các gói tin TCP hoặc UDP đến các cổng port đích nằm để xác định cổng có mở hay không.
  (6) Smurf : Là một loại tấn công sử dụng các máy tính bị nhiễm để tạo ra lưu lượng truy cập ảo, mục đích gây ra tắt nghẽn máy chủ hoặc mạng, làm chậm hoặc vô hiệu hóa mạng.

(7) Nmap : Được dùng để nhận diện dịch vụ mạng máy tính và máy chủ, bằng cách gửi các gói tin và phân tích phải hồi, từ đó có thể thăm dò mạng máy tính, phát hiện máy chủ, dịch vụ và hệ điều hành đang sử dụng.
(8) Others: Bao gồm tất cả các loại mã độc khác.



Hình 3.3 Phân bố số lượng 8 nhãn trong tập train



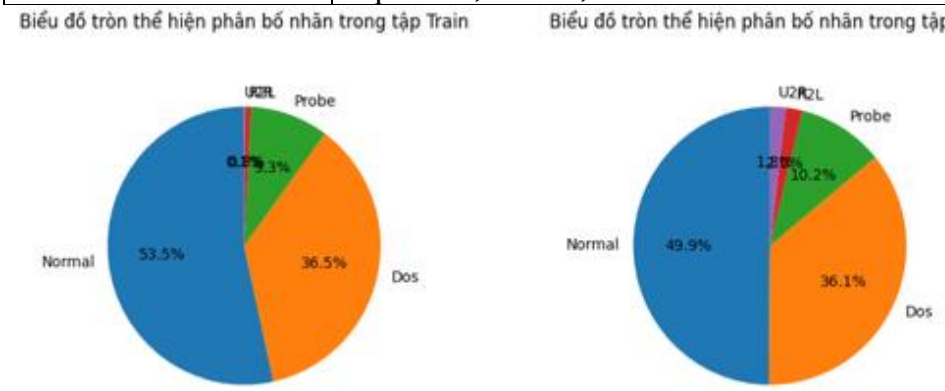
Hình 3.4 Phân bố số lượng 8 nhãn trong tập test

- Trường hợp 5 nhãn, phân loại nhóm (lớp) của mã độc tấn công :
(1) Normal : là những tệp dữ liệu bình thường, không nhằm mục đích tấn công
(2) DoS : là một loại tấn công làm cạn kiệt tài nguyên mạng, tắt nghẹn mạng, do đó làm cho không thể xử lý các tác vụ hợp pháp
(3) Probe : là một loại tấn công nhằm mục đích giám sát và thăm dò. Mục tiêu chính là thu thập thông tin của đối tượng
(4) U2R : là một loại mã độc giúp hacker truy cập trái phép vào các đặc quyền siêu người dùng, bằng cách sử dụng một tài khoảng bình thường để đăng nhập vào hệ thống nạn nhân và cố gắng giành được đặc quyền quản trị viên bằng cách khai thác các lỗ hỏng bảo mật của hệ thống

(5) R2L : là loại mã độc giúp truy cập trái phép từ một máy từ xa, kẻ tấn công xâm nhập vào máy từ xa và giành truy cập cục bộ vào máy nạn nhân

Bảng 3.2 Bảng phân loại 4 nhãn kiểu tấn công

Loại hình tấn công	Loại tấn công
DoS	Back, Land, Neptune, Pod, Smurf,Teardrop,Apache2, Udpstorm, Processtable, Worm
Probe	Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named
U2R	Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps



Hình 3.5 Phân bố số lượng nhã các kiểu tấn trong tập train và test

3.3. Xử lý dữ liệu

3.3.1. Xử lý dữ liệu chung cho 3 trường hợp

- Do bộ dữ liệu rút trích từ KDD99 nên bộ dữ liệu không có giá trị null hoặc NaN, không có cột nào có các giá trị bất thường nên công đoạn làm sạch dữ liệu không xử lý nhiều.

  (1) Phân loại dữ liệu thành các thuộc tính phù hợp như bảng ... để thuận tiện cho việc máy học

  (2) Xóa những cột dữ liệu không cần thiết như là
  - Num_outbound_cmds (chỉ có giá trị 0)
  - Is_host_login (chỉ có giá trị 1)
  - Num_root (tương quan với cột num_compromised đến 100%)

  (3) Áp dụng kĩ thuật StandardScaler đối với những cột có thuộc tính kiểu numeric (duration, src_bytes, dst_bytes, hot, num_compromised, num_file_creations..). Điều này giúp cho các đặc trưng có cùng một tầm quan trọng với mô hình và tiamr thiểu sự ảnh hưởng của các đặc trưng có thang đó lớn đến quá trình học của mô hình

(4) Áp dụng kĩ thuật OneHotEncoder giúp chuyển đổi mỗi giá trị đặc trưng phân loại thành một vector có số chiều bằng với số lượng giá trị có thể có của đặc trưng đó

3.3.2. Xử lý dữ liệu cho phân loại 2 nhãn

(1) Chuyển các dòng có label khác normal thành attack

(2) Chuyển normal thành 0, attack thành 1

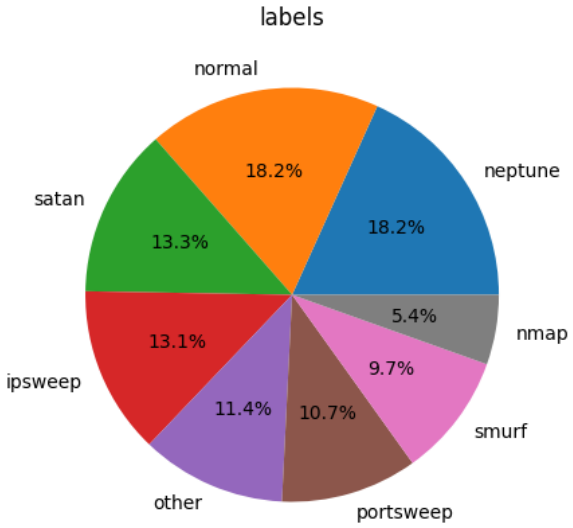
3.3.3. Xử lý dữ liệu cho phân loại 5 nhãn

(1) Phân loại các nhãn tấn công thành các nhãn phù hợp theo bảng

3.3.4. Xử lý dữ liệu cho phân loại 8 nhãn

(1) Chỉ giữ lại các nhãn label có tỷ lệ phần trăm số lượng trên tập lớn hơn 1% và phần còn lại chuyển thành other chúng ta được 8 nhãn

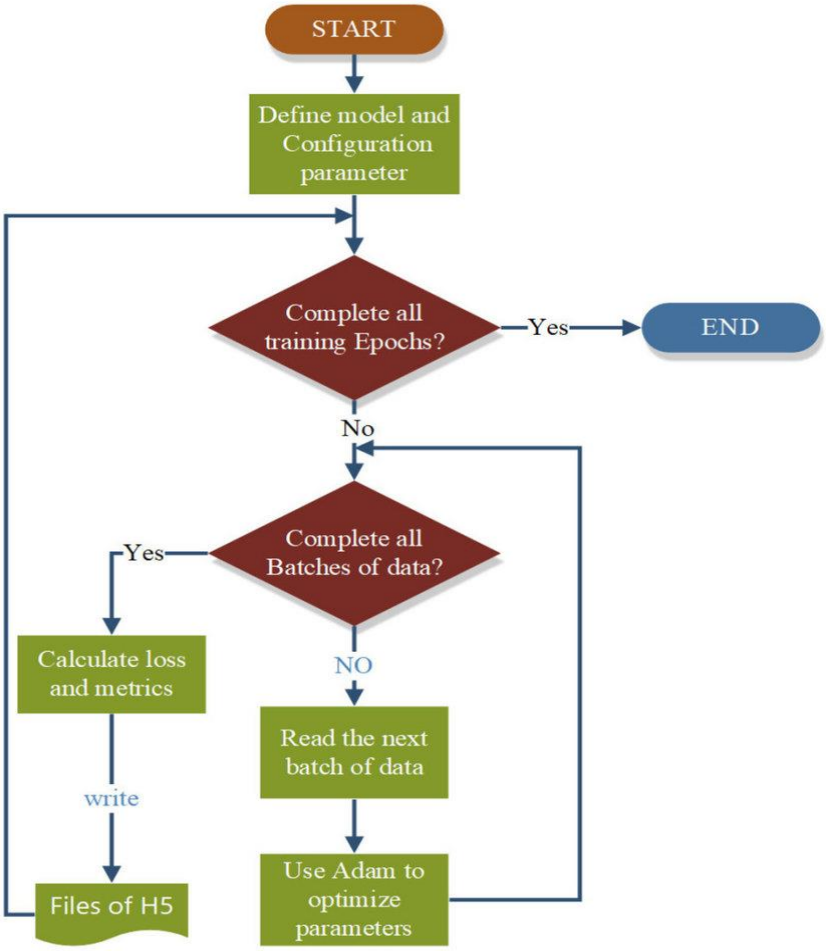
(2) Vì số lượng nhãn normal và neptune nhiều hơn so với số lượng nhãn còn lại nên cắt ngẫu nhiên 5000 dòng có nhãn label là normal và neptune ở bảng train tránh việc mất cân bằng dữ liệu



Hình 3.6 Phân bố số lượng 8 nhãn sau khi cân bằng dữ liệu

3.4. Huấn luyện mô hình

3.4.1. Các bước huấn luyện mô hình



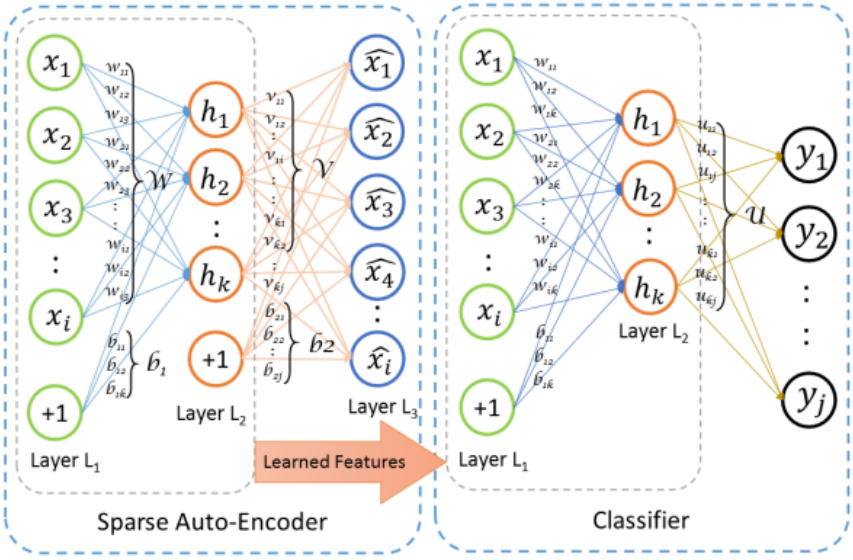
Hình 3.7 Sơ đồ Flow Chart mô hình

Giải thích: Sau bước tiền xử lý dữ liệu và feature engineering, nhóm tiến hành đến huấn luyện mô hình. Đầu tiên nhóm sẽ xác định mô hình sẽ được áp dụng cho bài toán (NN, LSTM, CNN) và thiết lập các tham số đầu vào cho mô hình (num layers, loss, acc, adam ….) cũng như số lần huấn luyện mô hình (Epochs). Với Loss và Adam tùy bài toán nhóm sẽ sử dụng Loss riêng, BinaryLossEntropy với bài toán 2 nhãn , CategoryLossEntropy với bài toán 5 và 8 nhãn. Riêng Adam nhóm đã tìm được con số phù hợp cho bài toán {0.1, 0.01}. Cứ sau mỗi lần train nếu lần train đó có Val Accuracy nhóm em sẽ lưu vào file H5 được lưu trong Colab.

Cuối cùng sau toàn bộ Epochs train sẽ tìm ra được mô hình có độ chính xác Val Accuracy cao nhất và nhóm sẽ lấy mô hình đó ra dự đoán cho tập Test Dataset. Nếu kết quả predict cao hơn kết quả cũ nhóm sẽ tiến hành vẽ Confuse Matrix để xác định độ chính xác cho từng nhãn trong từng bài toán. Sau đó nhóm sẽ trực quan hóa toàn bộ Loss Val và Acc Val trong toàn bộ các lần train (Epochs) để kiểm nghiệm xem mô hình có bị Overfit hoặc Underfit hay không từ đó ra các giải pháp cho lần thử nghiệm tiếp theo.

3.4.2. Mô hình huấn luyện Neuron Network (NN)

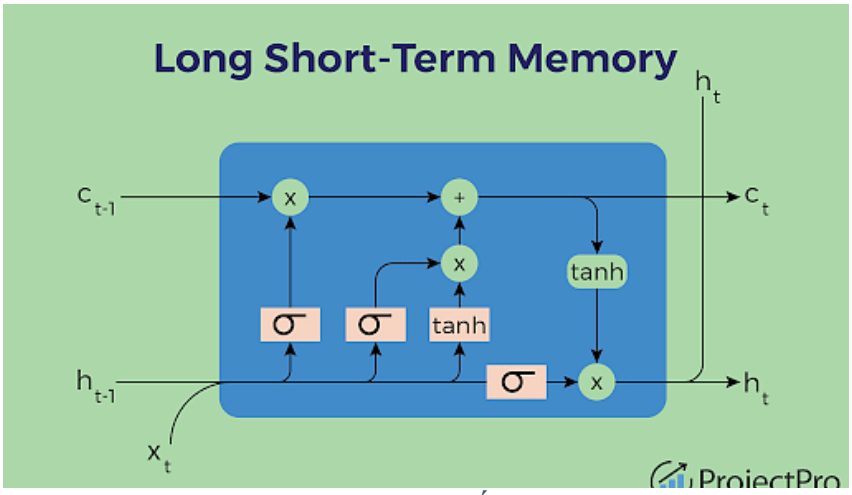
- Neural Network (NN) là một phương thức trong lĩ vực trí tuệ nhân tạo, thường được sử dụng để dạy máy tính xử lý dữ liệu theo cách xử lý của não bộ. Đây là một loại quy trình máy học, được gọi là deep learning, sử dụng các nút hoặc nơ-ron liên kết với nhau trong một cấu trúc phân lớp tương tự như một bộ não người. Phương thức này tạo ra một hệ thống thích ứng được máy tính sử dụng để học hỏi từ sai lầm của chúng và liên tục được cải thiện. Vì vậy mà mạng nơ ron nhắm tới giải quyết các vấn đề phức tạp, chẳng hạn như tóm tắt tài liệu hoặc nhận diện khuôn mặt với độ chính xác cao hơn máy học truyền thống.



Hình 3.8 Mô hình huấn luyện Neuron Network

3.4.3. Mô hình Long – short Term Memory (LSTM)

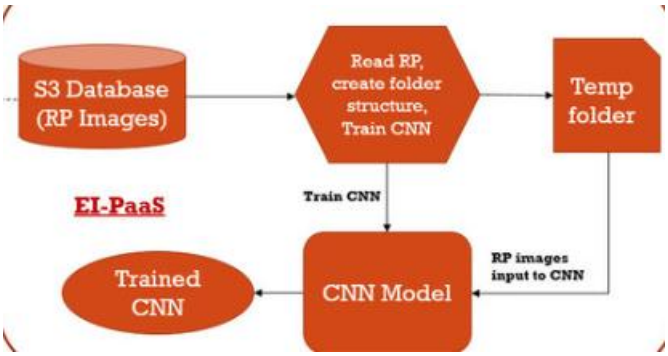
- Long Short-Term Memory (LSTM) là một kiến trúc mạng nơ-ron thuộc loại mạng hồi quy (Recurrent Neural Network - RNN) được thiết kế để giải quyết vấn đề biến mất gradient trong quá trình học của mạng hồi quy truyền thống. Do đặc tính này, LSTM thường được ưa chuộng trong các ứng dụng yêu cầu khả năng "nhớ" thông tin quan trọng trong thời gian dài.



Hình 3.9 Mô hình huấn luyện LSTM

3.4.4. Mô hình Convolutional Neural Network (CNN)

- Convolutional Neural Network (CNN) là một loại mạng nơ-ron được thiết kế đặc biệt để xử lý và phân loại dữ liệu không gian. CNN nổi tiếng với khả năng trích xuất và học các đặc trưng từ dữ liệu ảnh, giúp nó trở thành công cụ quan trọng trong lĩnh vực Computer Vision và nhiều ứng dụng khác.



Hình 3.10 Mô hình huyến luyện CNN

4. ĐÁNH GIÁ

4.1. Phần cứng

- Toàn bộ quá trình sử lý nhóm chúng em thực hiện trên google colab (GPU T4) với thông số của hệ thống máy tính là :
    - Kiến trúc GPU : Turing
    - Số lõi CUPA : 16,384
    - Bộ nhớ GPU : 16 GB GDDR6
    - Kiến trúc bộ nhớ : 256-bit
    - Tốc độ sử lý của GPU : 585 MHz(cơ bản), 1590 (boost)

4.2. Đánh giá mô hình – Độ đánh giá Accuracy Score

Bảng 4.1 Bảng đánh giá các mô hình dự theo độ đo Accuracy Score

Nhãn / Model	2 nhãn	5 nhãn	8 nhãn
Neuron Network	94,37%	93.19%	92.19%
CNN	92.52%	90.96%	90.8%
LSTM	92.46%	91.34%	91.7%
CNN + LSTM	92.32%	92.69%	91.31%

Nhận xét :

- *Dựa vào bảng tổng hợp kết quả của tất cả 12 mô hình mà nhóm đã thực hiện. Có thể rút ra nhận xét rằng mô hình 2 nhãn (labels) cho kết quả luôn cao hơn 5 và 8 nhãn. Để giải thích cho việc này có thể nói rằng bộ Dataset 2 nhãn có sự ổn định không có hiện tượng mất cân bằng nhãn do đó mô hình có thể học tốt cả 2 nhãn. Còn với TH 5 và 8 nhãn thì có hiện tượng chênh lệch nhãn trầm trọng điều đó dẫn đến mô hình học tốt phần lớn các nhãn tuy nhiên có thể sót 1 số nhãn có thể do dữ liệu quá ít khiến mô hình học không nhiều về nhãn đó. Tuy nhiên, nhóm đã cố gắng và cả 12 mô hình đều cho kết quả tương đối ổn phần lớn đều trên 90% (dựa trên độ đo Accuracy)*

- *Qua 12 mô hình nhóm đã nghiên cứu và cài đặt, nhóm em nhận ra rằng mô hình NN cho kết quả cao nhất trong cả 3 trường hợp 2 5 và 8 nhãn. Sau đây là một số lý do giải thích tại sao mô hình NN có kết quả cao nhất:*
    1. Dữ liệu chưa thật sự có nhiều có tính chất không gian và thời gian: Neural Network (NN) thường là lựa chọn tốt khi dữ liệu không có hoặc ít có các tính chất về thơi gian (như Time Series - LSTM) hoặc không gian (như ảnh - CNN).

    2. Kích thước dữ liệu trung bình:
    Neural Network (NN) thường cần ít dữ liệu hơn so với LSTM và CNN để đạt được hiệu suất tốt. LSTM và CNN thường yêu cầu một lượng lớn dữ liệu để học các mô hình phức tạp, trong khi NN có thể làm việc tốt với kích thước dữ liệu nhỏ hơn.

3. Khả năng tinh chỉnh và đơn giản:
   Neural Network (NN) thường có ít siêu tham số hơn so với LSTM và CNN, điều này có thể làm cho quá trình đào tạo và tinh chỉnh mô hình trở nên dễ dàng hơn. Trong khi LSTM và CNN vẫn cần nhiều thời gian hơn để nghiên cứu nhằm tìm ra bộ tham số tối ưu nhất cho bài toán phát hiện xâm nhập.

4. Đặc trưng đơn giản hoặc ít có cấu trúc đặc biệt:
   Các đặc trưng trong bộ Dataset ít có cấu trúc đặc biệt hoặc đặt trưng đơn giản, do đó NN có thể thực hiện tốt hơn so với các mô hình chuyên sâu như LSTM hoặc CNN. Tuy nhiên LSTM và CNN vẫn rất đáng và nên được sử dụng trong bài toán phát hiện xâm nhập.

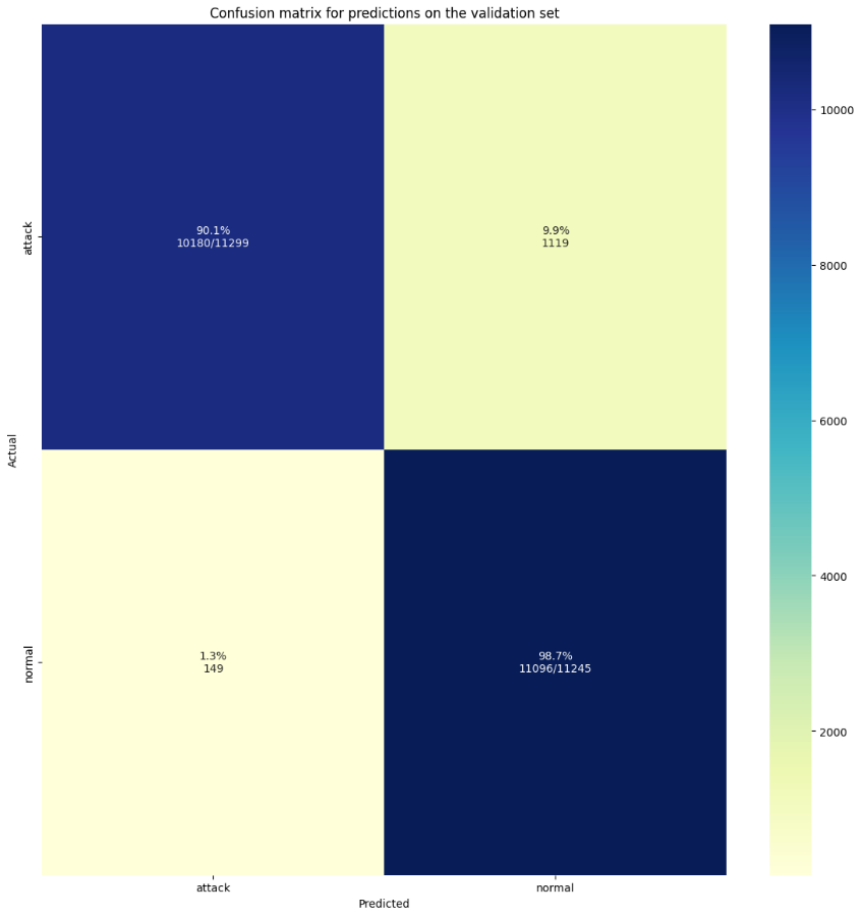
Tổng kết : *Trong quá trình nghiên cứu và xây dựng mô hình, nhóm em đã sử dụng 4 mô hình bao gồm Neural Network (NN), Long – short term Memory (LSTM), Convolutional Neural Network (CNN) và LSTM + CNN. Trong đó mỗi mô hình đều có những điểm mạnh riêng để nhóm quan tâm và sử dụng :*

- Neural Network (NN) : Là mô hình căn bản cầu nối giữa Machine Learning (ML) và Deep Learning (DL), trong đó các kiến trúc đều xây dựng dựa trên cấu trúc mạng Neural, đặt biệt với Dataset dạng CSV thì NN luôn là sự lựa chọn đầu tiên của rất nhiều nhà nghiên cứu. Ngoài ra NN còn có nhiều điểm mạnh như đơn giản, ít tham số nên giảm thời gian huấn luyện và đặt biệt là hoạt động ổn trên bộ Data “thuần bảng”.
- Long – short term Memory (LSTM) : Là mô hình thiên hướng mạnh cho các bài toán về Time Series. Sau khi phân tích Data nhóm nhận ra rằng có những đặt trưng có yêu cầu về mặt thời gian như: Duration (thời gian truyền tải dữ liệu giữa 2 máy) , …. Nên nhóm đã nghĩ đến LSTM như 1 phương pháp rất đáng để thử nghiệm trong bài toán phát hiện xâm nhập.
- Convolutional Neural Network (CNN) : Là mô hình mạnh về mặt không gian (như các tác vụ về ảnh), ngoài ra CNN cũng hỗ trợ tốt cho các bài toán có dạng Data tuần tự và cũng như không tuần tự. Và trong các bài toán ứng dụng Deep Learning trong lĩnh vực An toàn thông tin thì CNN rất thường được sử dụng và cho ra kết quả khá tốt nên nhóm đã chọn CNN sau NN và LSTM để thử nghiệm và CNN cũng cho ra kết quả dự đoán rất ổn (trên 90% Test Dataset).

4.2.1. Đánh giá với bộ dữ liệu 2 nhãn

4.2.1.1. Mô hình neuron network

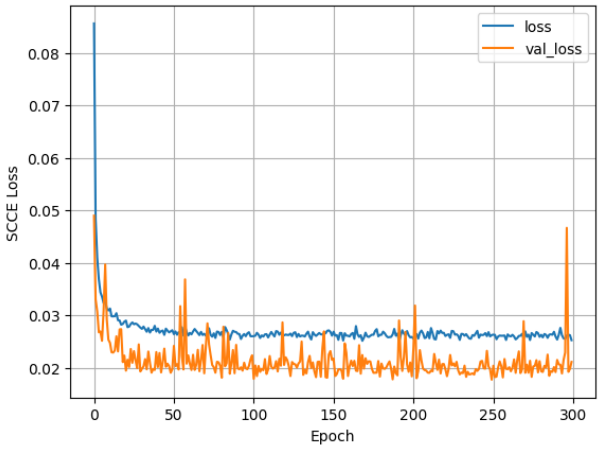
- Độ chinh xác accuracy : 94.37%
- Ma trận nhầm lẫn cho các dự đoán trên tập test



Hình 4.2 Ma trận nhầm lẫn của thuật toán Neural Network (NN) cho trường hợp 2 nhãn

Nhận xét : *Mô hình cho kết quả dự đoán đều trên 90% cho cả 2 nhãn Attack và Normal trong đó mô hình học khá tốt nhãn Normal với chính xác trên nhãn này tận 98.7%, trong đó nhãn Attack được cho kết quả khá ổn với chính xác 90.1%*

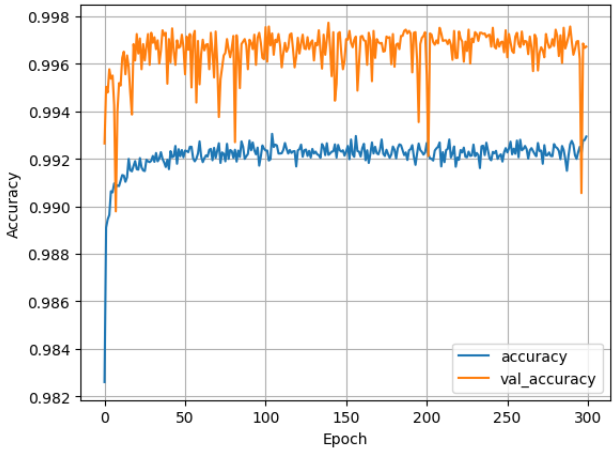
- Biểu đồ hàm loss



Hình 4.1 Biểu đồ hàm loss của thuật toán NN cho trường hợp 2 nhãn

Nhận xét : *Dựa vào biểu đồ hàm loss của mô hình Neural Network trong bài toán 2 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy

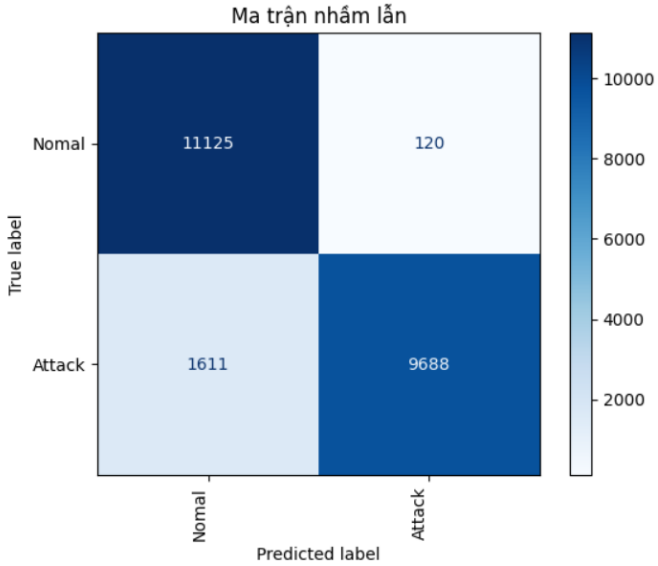


Hình 4.2 Biểu đồ hàm accuracy của thuật toán NN cho mô hình 2 nhãn

Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình Neural Network trong bài toán 2 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

4.2.1.2. Mô hình LSTM + CNN

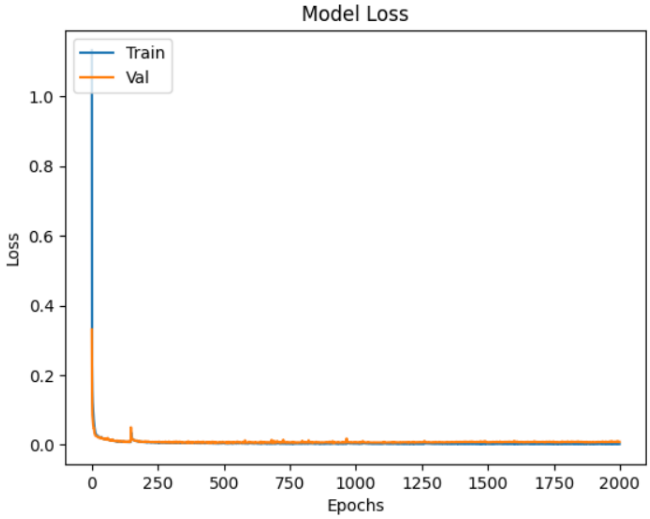
- Độ chinh xác accuracy : 92.32%
- Ma trận nhầm lẫn cho các dự đoán trên tập test



Hình 4.3 Ma trận nhầm lẫn của thuật toán LSTM + CNN cho trường hợp 2 nhãn

Nhận xét : *Mô hình LSTM + CNN cho dự đoán chính xác nhiều dựa trên đường chéo chính của đồ thị (màu xanh đậm). Trong đó nhãn Normal dự đoán chính xác 11125 / 11245 chiếm tỉ lệ 98.93%, nhãn Attack cho dự đoán chính xác 9688 / 11299 chiếm tỉ lệ 86%.*

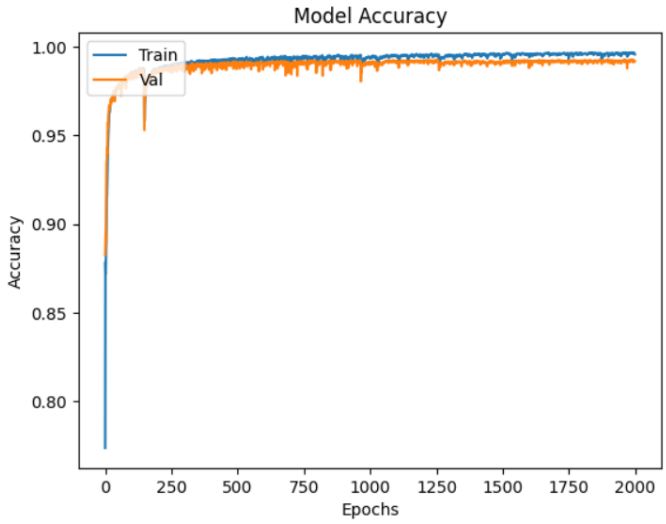
- Biểu đồ hàm loss



Hình 4.4 Biểu đồ hàm loss của thuật toán LSTM + CNN cho trường hợp 2 nhãn

Nhận xét : *Dựa vào biểu đồ hàm loss của mô hình LSTM + CNN trong bài toán 2 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy

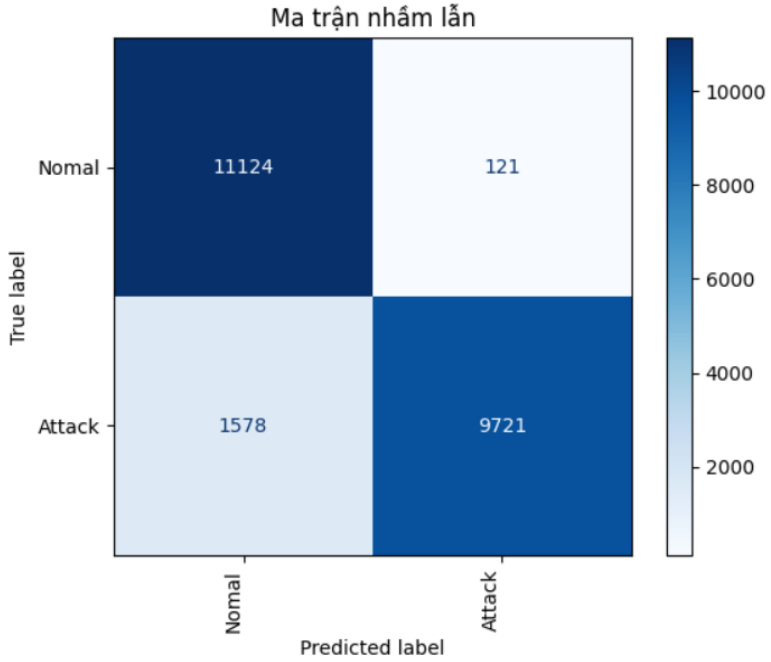


Hình 4.5 Biểu đồ hàm accuracy của thuật toán LSTM + CNN cho trường hợp 2 nhãn

Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình LSTM + CNN trong bài toán 2 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

4.2.1.3. Mô hình LSTM

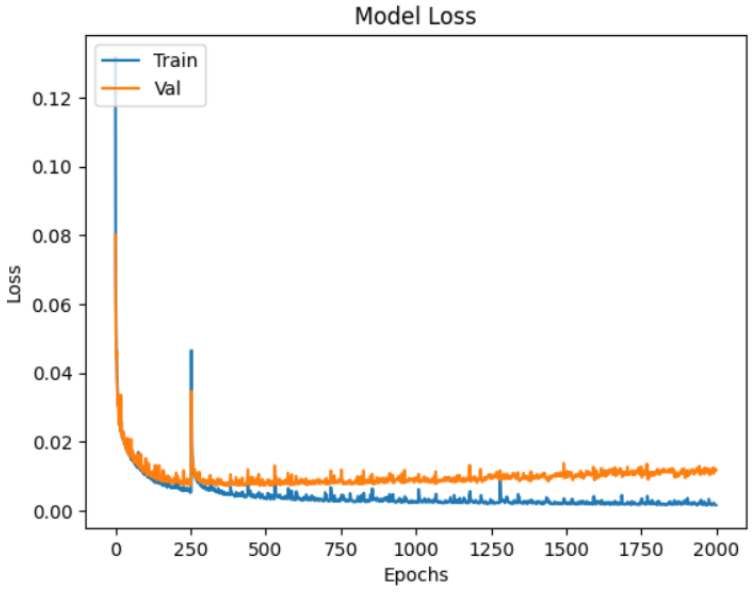
- Độ chinh xác accuracy : 92.46%
- Ma trận nhầm lẫn cho các dự đoán trên tập test



Hình 4.6 Ma trận nhầm lẫn của thuật toán LSTM cho trường hợp 2 nhãn

***Nhận xét** : Mô hình LSTM cho dự đoán chính xác nhiều dựa trên đường chéo chính của đồ thị (màu xanh đậm). Trong đó nhãn Normal được dự đoán chính xác 11124 / 11245 chiếm tỉ lệ 98.9%, nhãn Attack được dự đoán chính xác 9721 / 11299 chiếm tỉ lệ 86.03%*

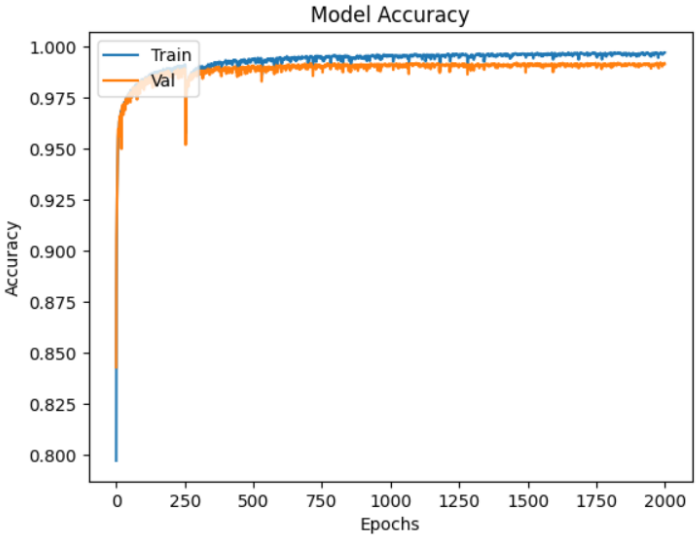
- Biểu đồ hàm loss



Hình 4.7 Biểu đồ hàm loss cho thuật toán LSTM cho trường hợp 2 nhãn

***Nhận xét :** Dựa vào biểu đồ hàm loss của mô hình LSTM trong bài toán 2 nhãn ta có thể thấy rằng khoảng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy

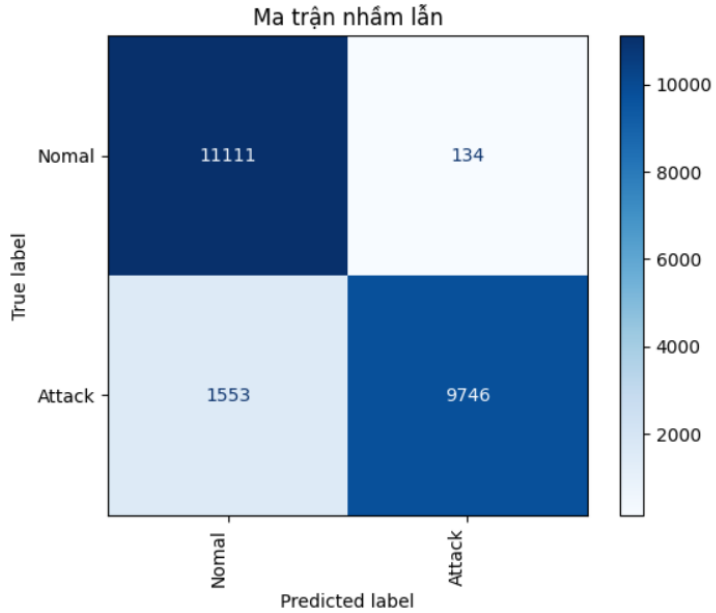


Hình 4.8 Biểu đồ hàm accuracy thuật toán LSTM của trường hợp 2 nhãn

***Nhận xét** : Dựa vào biểu đồ hàm Accuracy của mô hình LSTM trong bài toán 2 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit*

4.2.1.4. Mô hình CNN

- Độ chính xác accuracy: 92,52%
- Ma trận nhầm lẫn cho các dự đoán trên tập test

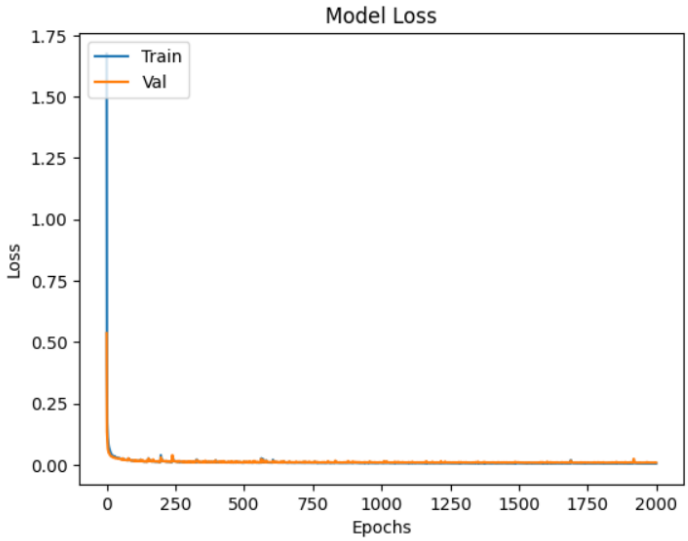


Hình 4.9 Ma trận nhầm lẫn thuật toán CNN trường hợp 2 nhãn

***Nhận xét :** Mô hình CNN cho dự đoán chính xác nhiều dựa trên đường chéo chính của đồ thị (màu xanh đậm). Trong đó nhãn Normal được dự đoán chính xác 11111 /*

11245 chiếm tỉ lệ 98.8%, nhãn Attack được dự đoán chính xác 9746 / 11299 chiếm tỉ lệ 86.26%

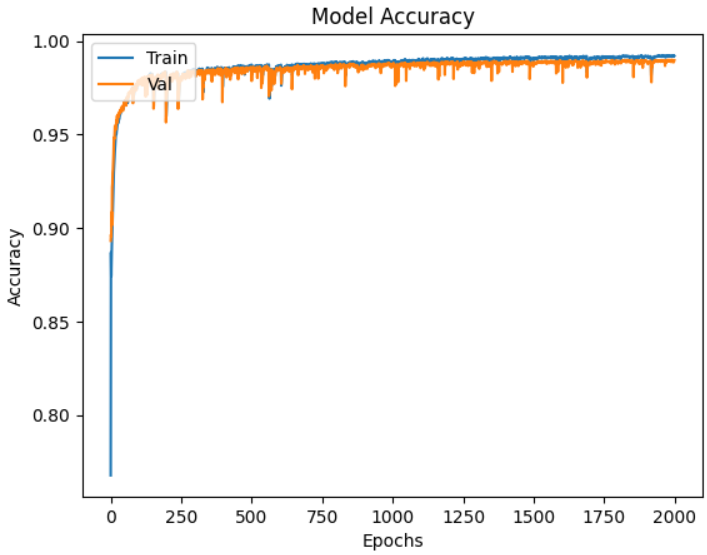
- Biểu đồ hàm loss



Hình 4.10 Biểu đồ ma trận hàm loss thuật toán CNN trường hợp nhãn

Nhận xét : *Dựa vào biểu đồ hàm loss của mô hình CNN trong bài toán 2 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy



Hình 4.11 Biểu đồ ma trận hàm accuracy thuật toán CNN trường hợp 2 nhãn

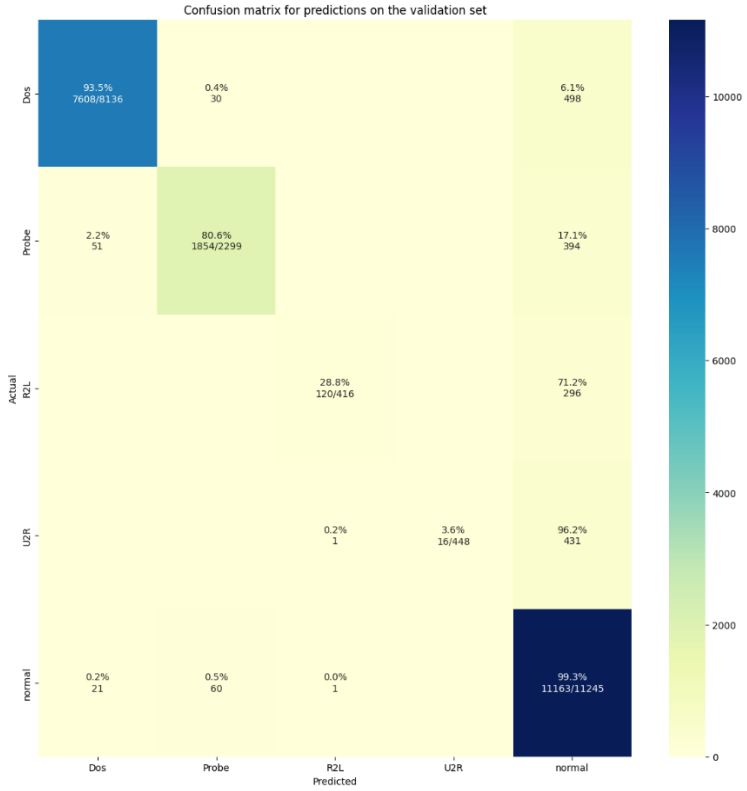
Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình CNN trong bài toán 2 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự*

cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.'

4.2.2. Đánh giá với bộ dữ liệu 5 nhãn

4.2.2.1. Mô hình neuron network

- Độ chính xác accuracy : 93.19%
- Ma trận nhầm lẫn cho các dự đoán trên tập test

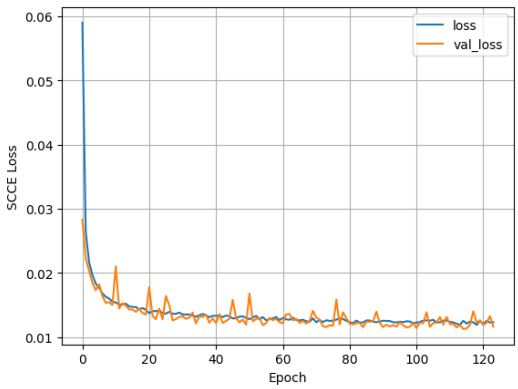


Hình 4.12 Ma trận nhầm lẫn thuật toán Neuron Network trường hợp 5 nhãn

***Nhận xét** : Mô hình NN trong bài toán 5 nhãn cho kết quả dự đoán lần lượt là DOS : 93.6%, Probe : 80.6%, R2L : 53.5%, U2R : 3.6% và Normal : 99%. Nhóm nhận thấy rằng mô hình cho dự đoán khá tốt trên 3 nhãn DOS, Probe và Normal và dự đoán trung bình trên 1 nhãn R2L và dự đoán chưa tốt trên 1 nhãn U2R. Có thể do data của*

nhãn U2R quá ít so với phần còn lại dẫn đến việc mô hình chưa thật sự học tốt. Do đó nhóm đang NC focal loss giúp mô hình học sâu hơn về những data tối tiểu (data có số lượng ít trên 1 nhãn).

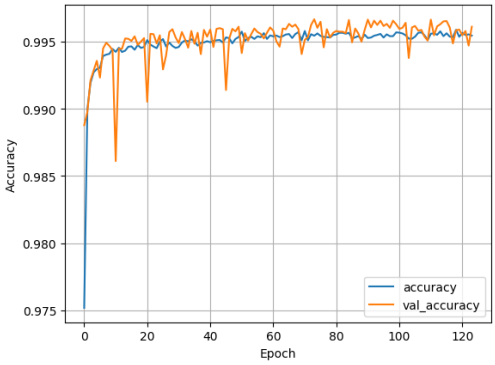
- Biểu đồ hàm loss



Hình 4.13 Biểu đồ hàm loss thuật toán Neuron Network trường hợp 5 nhãn

Nhận xét : *Dựa vào biểu đồ hàm loss của mô hình Neural Network trong bài toán 5 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy

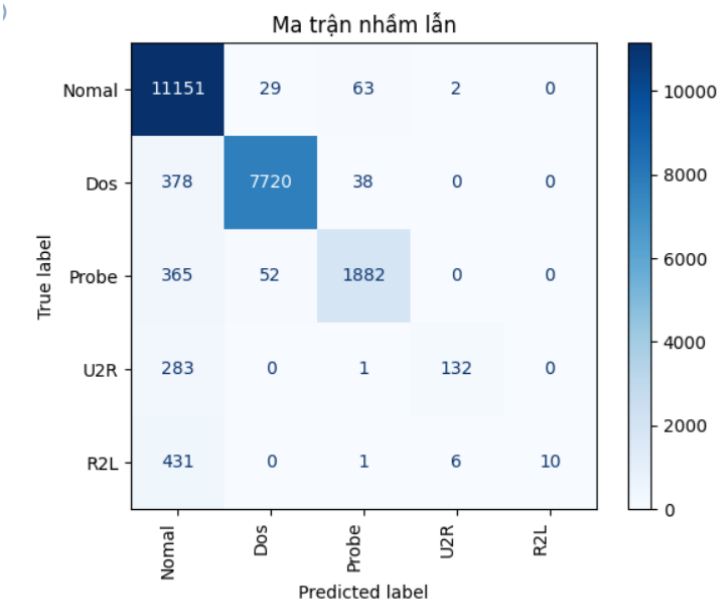


Hình 4.14 Biểu đồ hàm accuracy thuật toán Neuron Network trường hợp 5 nhãn

Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình Neural Network trong bài toán 5 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

4.2.2.2. Mô hình LSTM + CNN

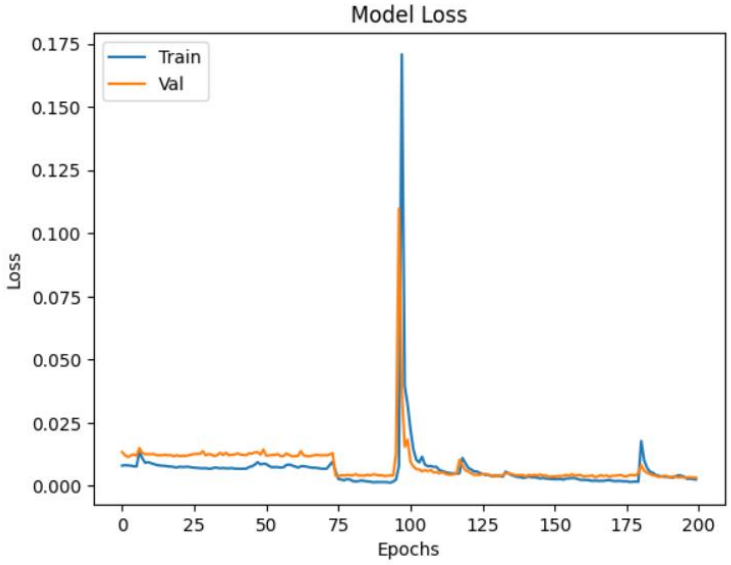
- Độ chính xác: 92.69%
- Ma trận nhầm lẫn cho các dự đoán trên tập test



Hình 4.15 Ma trận nhầm lẫn thuật toán LSTM + CNN trường hợp 5 nhãn

Nhận xét : *Mô hình LSTM + CNN trong bài toán 5 nhãn cho kết quả dự đoán lần lượt là DOS : 94.9%, Probe : 81.86%, R2L : 2.23%, U2R : 31.7% và Normal : 99.16%. Nhóm nhận thấy rằng mô hình cho dự đoán khá tốt trên 3 nhãn DOS, Probe và Normal và dự đoán trung bình trên 1 nhãn U2R và dự đoán chưa tốt trên 1 nhãn R2L. Có thể do data của nhãn R2L quá ít so với phần còn lại dẫn đến việc mô hình chưa thật sự học tốt. Do đó nhóm đang nghiên cứu focal loss giúp mô hình học sâu hơn về những data tối tiểu (data có số lượng ít trên 1 nhãn).*

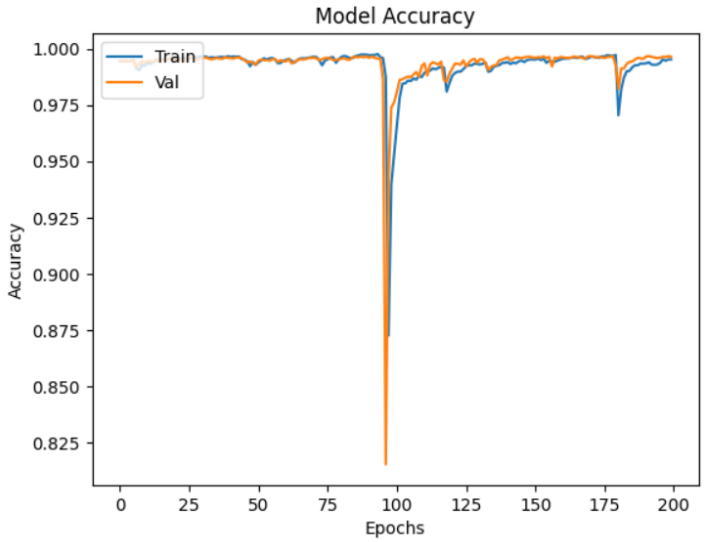
- Biểu đồ hàm loss



Hình 4.16 Biểu đồ hàm loss thuật toán LSTM + CNN trường hợp 5 nhãn

Nhận xét : *Dựa vào biểu đồ hàm loss của mô hình LSTM + CNN trong bài toán 5 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy

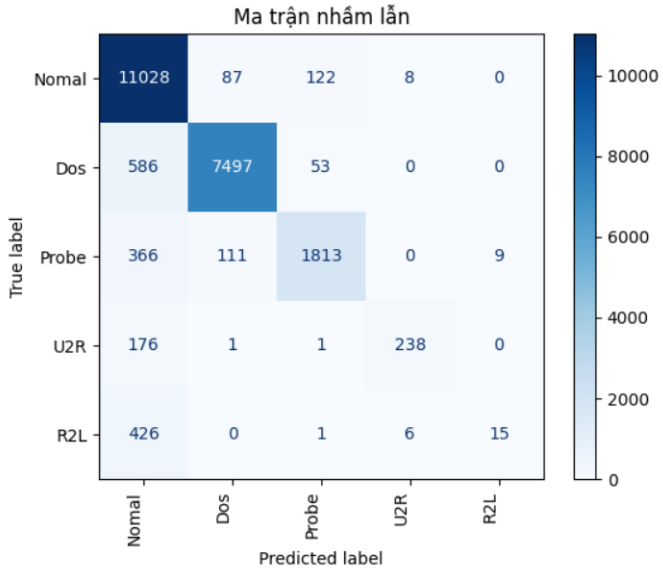


Hình 4.17 Biểu đồ hàm accuracy thuật toán LSTM + CNN trường hợp 5 nhãn

Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình LSTM + CNN trong bài toán 5 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit*

4.2.2.3. Mô hình LSTM

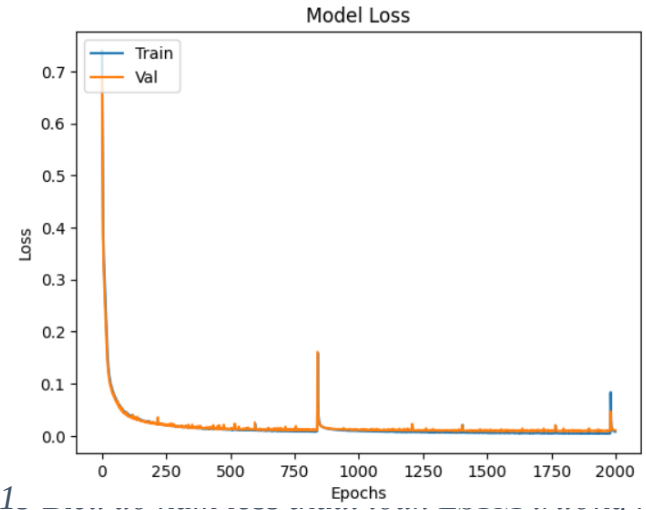
- Độ chính xác accuracy : 91.34%
- Ma trận nhầm lẫn cho các dự đoán trên tập test



Hình 4.18 Ma trận nhầm lẫn thuật toán LSTM trường hợp 5 nhãn

Nhận xét : *Mô hình LSTM trong bài toán 5 nhãn cho kết quả dự đoán lần lượt là DOS : 92.14%, Probe : 78.86%, R2L : 3.35%, U2R : 57.21% và Normal : 98.07%. Nhóm nhận thấy rằng mô hình cho dự đoán khá tốt trên 3 nhãn DOS, Probe và Normal và dự đoán trung bình trên 1 nhãn U2R và dự đoán chưa tốt trên 1 nhãn R2L. Có thể do data của nhãn R2L quá ít so với phần còn lại dẫn đến việc mô hình chưa thật sự học tốt. Do đó nhóm đang nghiên cứu focal loss giúp mô hình học sâu hơn về những data tối tiểu (data có số lượng ít trên 1 nhãn).*

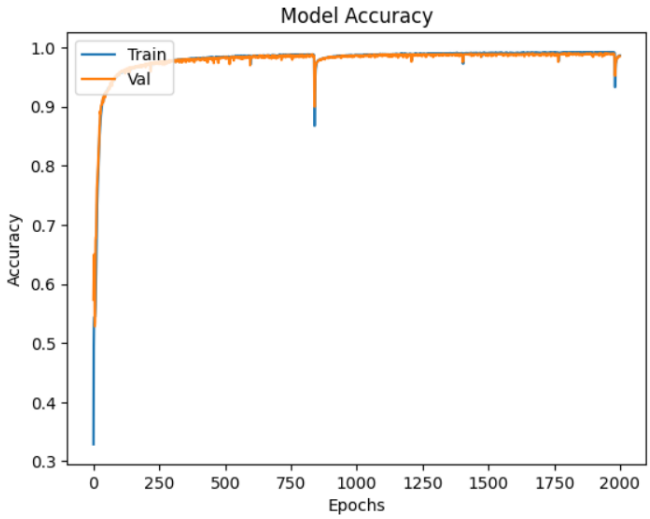
- Biểu đồ hàm loss



Hình 4.1 [illegible] trường hợp 5 nhãn

Nhận xét *: Dựa vào biểu đồ hàm loss của mô hình LSTM trong bài toán 5 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy

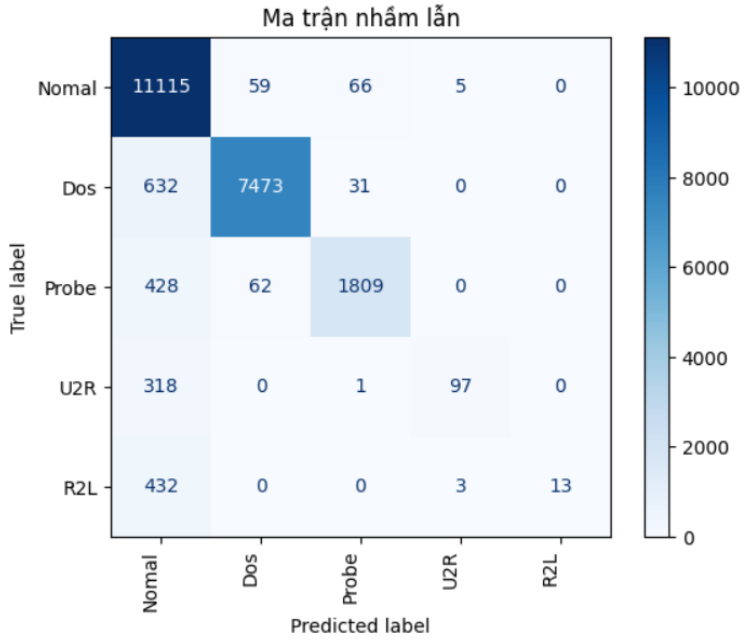


Hình 4.20 Biểu đồ hàm accuracy thuật toán LSTM trường hợp 5 nhãn

Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình LSTM trong bài toán 5 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit*

4.2.2.4. Mô hình CNN

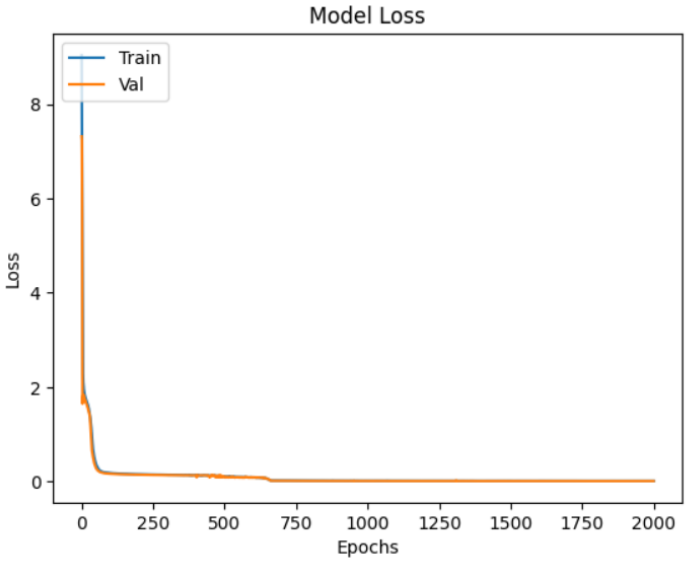
- Độ chính xác accuray : 90.96%
- Ma trận nhầm lẫn cho các dự đoán trên tập test



Hình 4.21 Ma trận nhầm lẫn thuật toán CNN trường hợp 5 nhãn

Nhận xét : *Mô hình CNN trong bài toán 5 nhãn cho kết quả dự đoán lần lượt là DOS : 91.85%, Probe : 78.69%, R2L : 2.9%, U2R : 23.31% và Normal : 98.84%. Nhóm nhận thấy rằng mô hình cho dự đoán khá tốt trên 3 nhãn DOS, Probe và Normal và dự đoán trung bình trên 1 nhãn U2R và dự đoán chưa tốt trên 1 nhãn R2L. Có thể do data của nhãn R2L quá ít so với phần còn lại dẫn đến việc mô hình chưa thật sự học tốt. Do đó nhóm đang nghiên cứu focal loss giúp mô hình học sâu hơn về những data tối tiểu (data có số lượng ít trên 1 nhãn).*

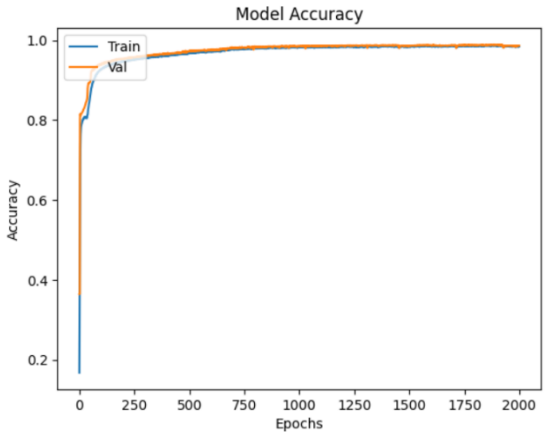
- Biểu đồ hàm loss



Hình 4.22 Biểu đồ hàm loss thuật toán CNN trường hợp 5 nhãn

Nhận xét : *Dựa vào biểu đồ hàm loss của mô hình CNN trong bài toán 5 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy



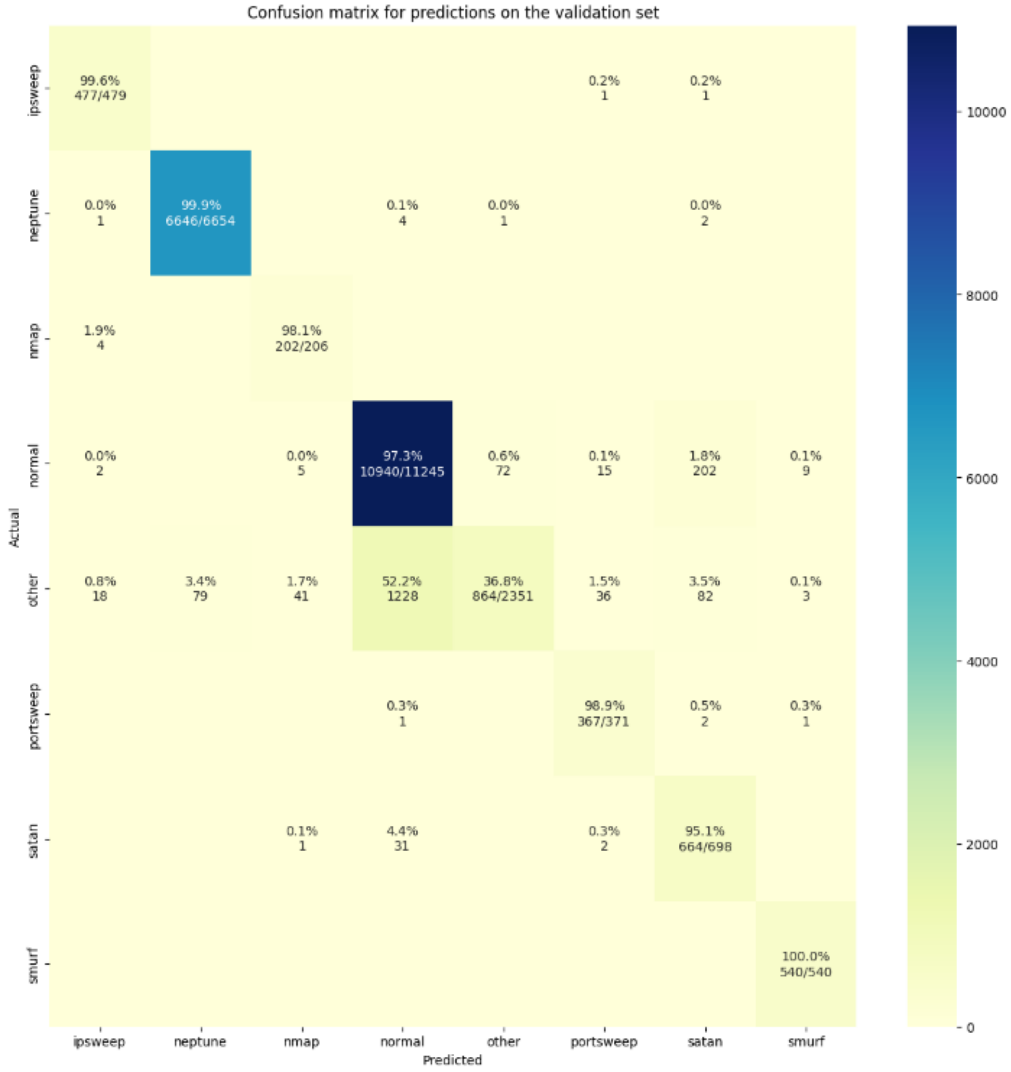
Hình 4.23 Biểu đồ hàm accuracy thuật toán CNN trường hợp 5 nhãn

Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình CNN trong bài toán 5 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

4.2.3. Đánh giá với bộ dữ liệu 8 nhãn

4.2.3.1. Mô hình neuron network

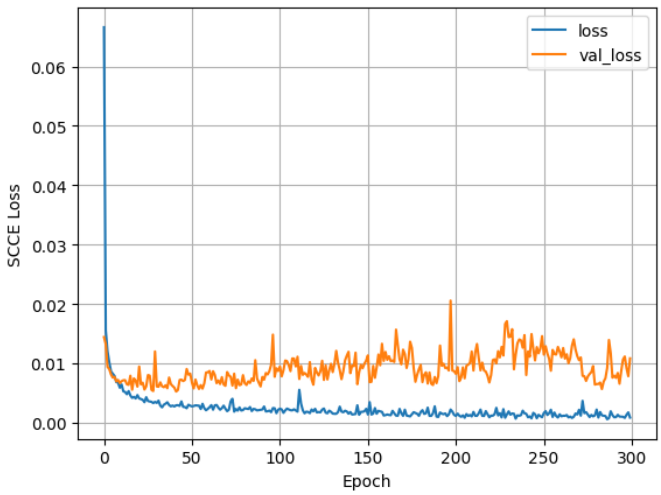
- Độ chính xác accuracy : 92.19%
- Ma trận nhầm lẫn cho các dự đoán trên tập test



Hình 4.24 Ma trận nhầm lân thuật toán Neuron Network trường hợp 8 nhãn

***Nhận xét** : Mô hình NN trong bài toán 8 nhãn cho kết quả dự đoán lần lượt là Ipsweep : 99.6%, Neptune : 99.9%, Nmap : 98.1%, Normal : 97.3% , Other : 36.8%, PortSweep : 98.9%, Satan : 95.1% và Smuft : 100 %. Phần lớn các nhãn đều cho độ chính xác cao thể hiện mô hình có khả năng học và phân loại tốt. Tuy nhiên có 1 TH nhãn dự đoán chỉ ở giá trị trung bình là Other với 36.8%, với trường hợp nhãn other cho kết quả thấp hơn các nhãn khác nhưng không đến nỗi quá tệ, nhóm đang nghiên cứu và quyết định cần tăng cường dữ liệu cho TH nhãn Other để giúp mô hình học nhiều hơn các đặt trưng về nhãn.*

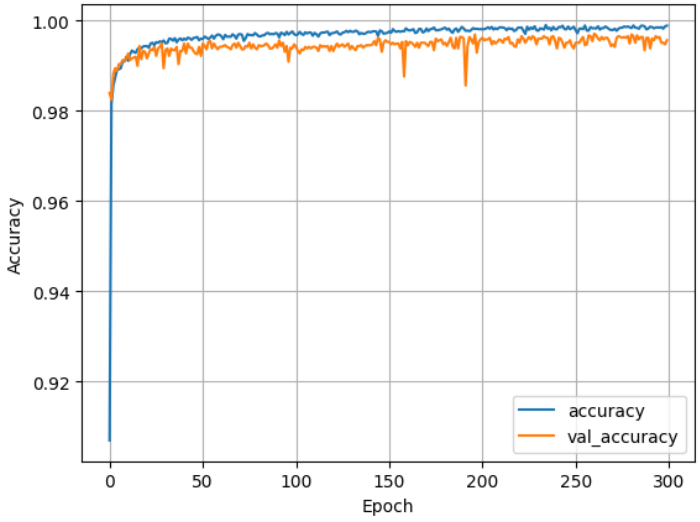
- Biểu đồ hàm loss



Hình 4.25 Biểu đồ hàm loss thuật toán Neuron Network trường hợp 8 nhãn

Nhận xét : *Dựa vào biểu đồ hàm loss của mô hình Neural Network trong bài toán 8 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy

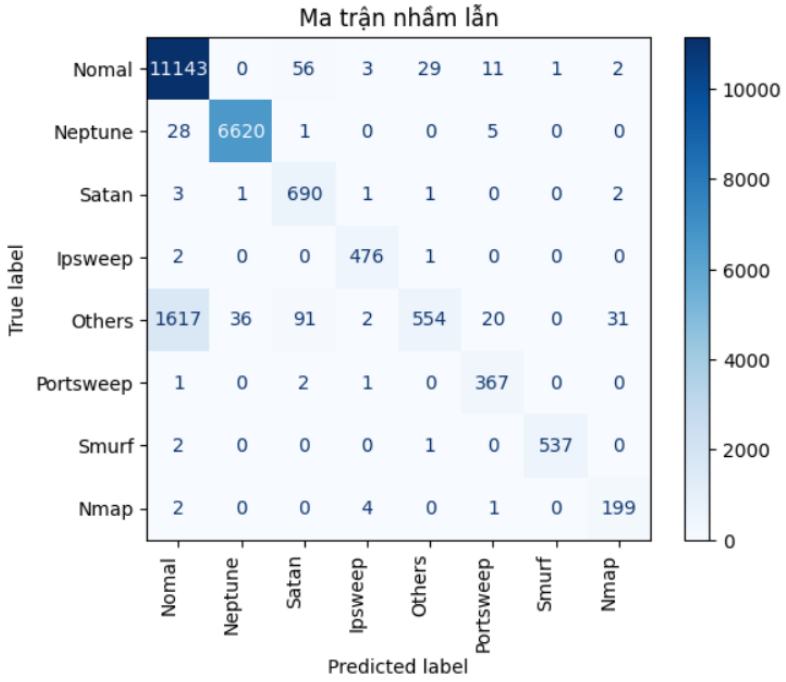


Hình 4.26 Biểu đồ hàm accuracy thuật toán Neuron Network trường hợp 8 nhãn

Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình NN trong bài toán 8 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit*

4.2.3.2. Mô hình LSTM + CNN

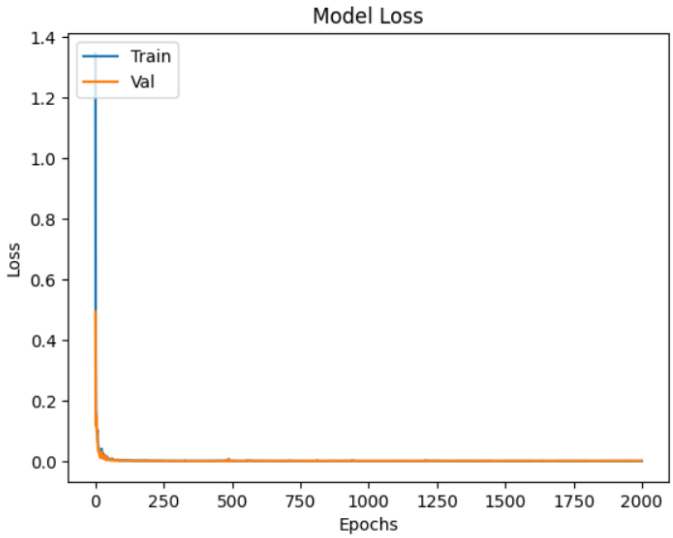
- Độ chính xác accuracy : 91.31%



Hình 4.27 Ma trận nhầm lẫn thuật toán LSTM + CNN trường hợp 8 nhãn

Nhận xét : *Mô hình LSTM + CNN trong bài toán 8 nhãn cho kết quả dự đoán lần lượt là Ipsweep : 99.37%, Neptune : 99.5%, Nmap : 96.6%, Normal : 99.12% , Other : 23.56%, PortSweep : 98.92%, Satan : 98.85% và Smuft : 99.44 %. Phần lớn các nhãn đều cho độ chính xác cao thể hiện mô hình có khả năng học và phân loại tốt. Tuy nhiên có 1 TH nhãn dự đoán chỉ ở giá trị trung bình là Other với 36.8%, với trường hợp nhãn other cho kết quả thấp hơn các nhãn khác nhưng không đến nỗi quá tệ, nhóm đang nghiên cứu và quyết định cần tăng cường dữ liệu cho TH nhãn Other để giúp mô hình học nhiều hơn các đặt trưng về nhãn*

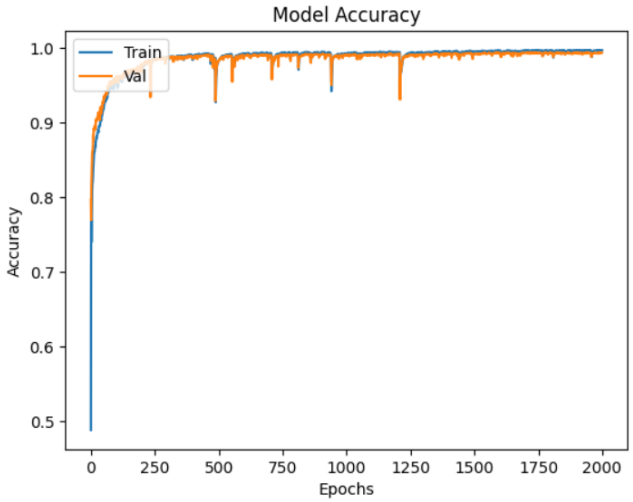
- Biểu đồ hàm loss



Hình 4.28 Biểu đồ hàm loss thuật toán LSTM + CNN trường hợp 8 nhãn

Nhận xét : *Dựa vào biểu đồ hàm loss của mô hình LSTM + CNN trong bài toán 8 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy

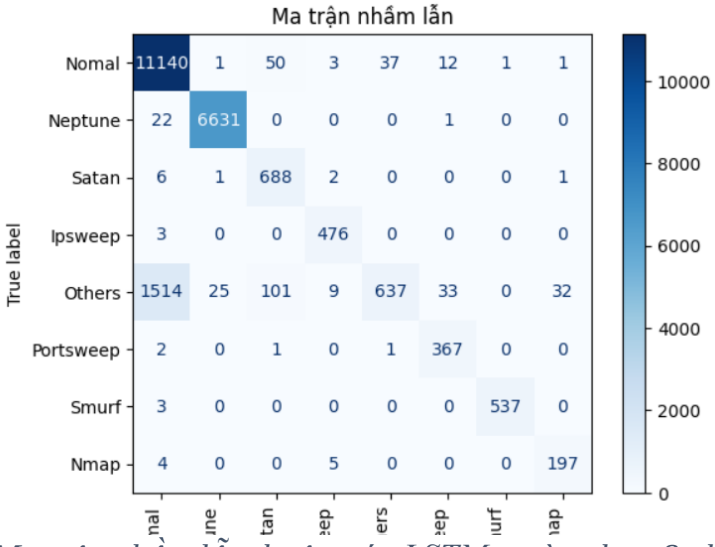


Hình 4.29 Biểu đồ hàm accuracy thuật toán LSTM + CNN trường hợp 8 nhãn

Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình LSTM + CNN trong bài toán 8 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit*

4.2.3.3. Mô hình LSTM

- Độ chính xác accuracy : 91.7%
- Ma trận nhầm lẫn cho các dự đoán trên tập test

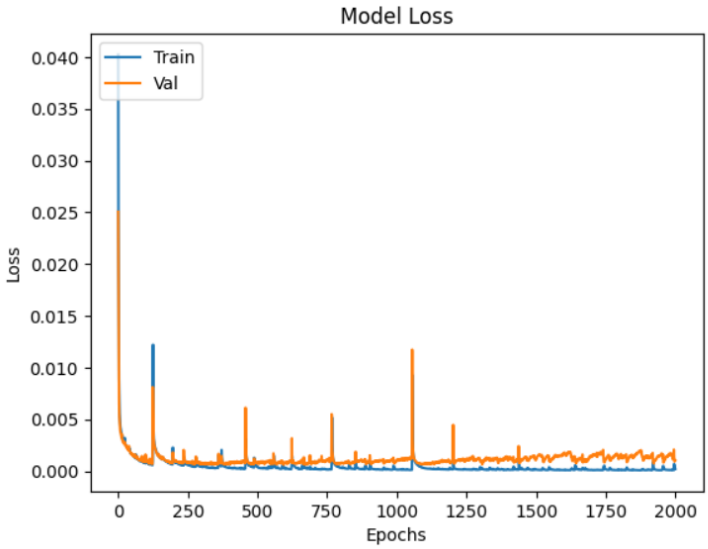


Hình 4.30 Ma trận nhầm lẫn thuật toán LSTM trường hợp 8 nhãn

Nhận xét : *Mô hình LSTM trong bài toán 8 nhãn cho kết quả dự đoán lần lượt là Ipsweep : 99.37%, Neptune : 99.65%, Nmap : 95.63%, Normal : 99.09% , Other : 27.09%, PortSweep : 98.92%, Satan : 98.56% và Smuft : 99.44 %. Phần lớn các nhãn*

đều cho độ chính xác cao thể hiện mô hình có khả năng học và phân loại tốt. Tuy nhiên có 1 TH nhãn dự đoán chỉ ở giá trị trung bình là Other với 36.8%, với trường hợp nhãn other cho kết quả thấp hơn các nhãn khác nhưng không đến nỗi quá tệ, nhóm đang nghiên cứu và quyết định cần tăng cường dữ liệu cho TH nhãn Other để giúp mô hình học nhiều hơn các đặt trưng về nhãn.

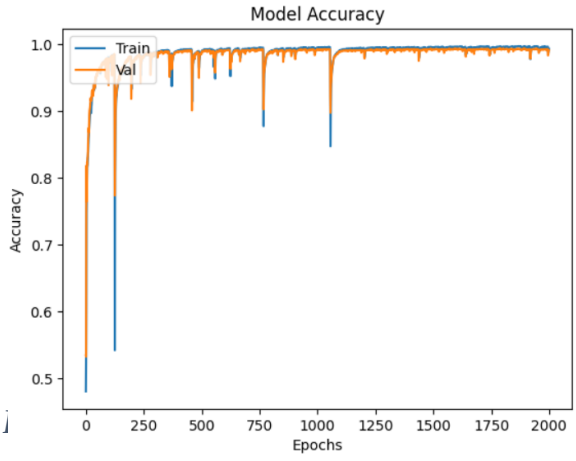
- Biểu đồ hàm loss



Hình 4.31 Biểu đồ hàm loss thuật toán LSTM trường hợp 8 nhãn

***Nhận xét :** Dựa vào biểu đồ hàm loss của mô hình LSTM trong bài toán 8 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy

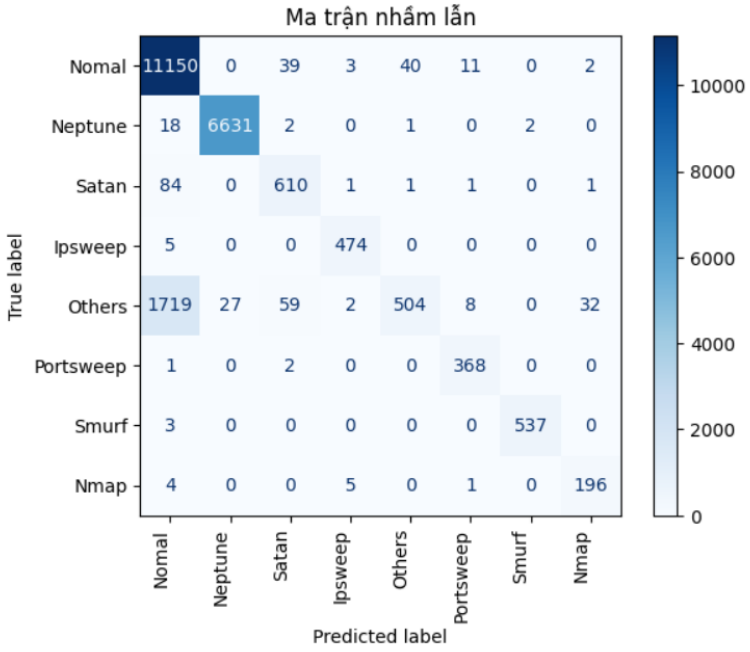


Hình 4.32 ường hợp 8 nhãn

***Nhận xét :** Dựa vào biểu đồ hàm Accuracy của mô hình LSTM trong bài toán 8 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit*

4.2.3.4. Mô hình CNN

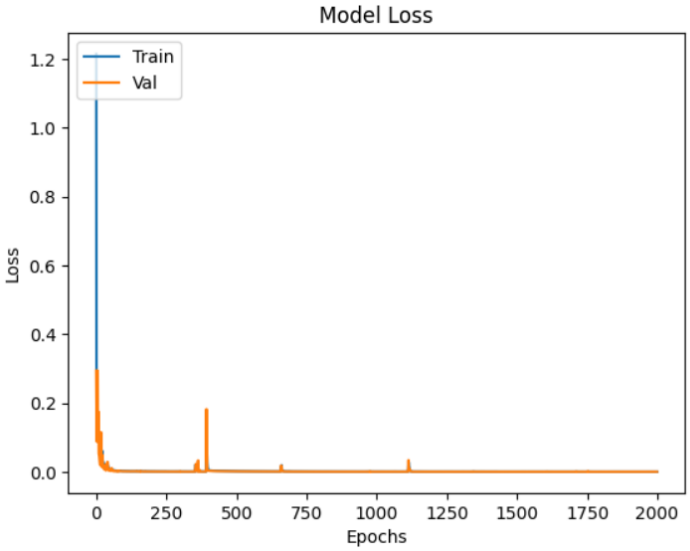
- Độ chính xác accuracy : 90.8%
- Ma trận nhầm lẫn cho các dự đoán trên tập test



Hình 4.33 Ma trận nhầm lẫn thuật toán CNN trường hợp 8 nhãn

Nhận xét : *Mô hình CNN trong bài toán 8 nhãn cho kết quả dự đoán lần lượt là Ipsweep : 98.95%, Neptune : 99.65%, Nmap : 95.14%, Normal : 99.18% , Other : 21.43%, PortSweep : 99.19%, Satan : 87.39% và Smuft : 99.44 %. Phần lớn các nhãn đều cho độ chính xác cao thể hiện mô hình có khả năng học và phân loại tốt. Tuy nhiên có 1 TH nhãn dự đoán chỉ ở giá trị trung bình là Other với 36.8%, với trường hợp nhãn other cho kết quả thấp hơn các nhãn khác nhưng không đến nỗi quá tệ, nhóm đang nghiên cứu và quyết định cần tăng cường dữ liệu cho TH nhãn Other để giúp mô hình học nhiều hơn các đặt trưng về nhãn.*

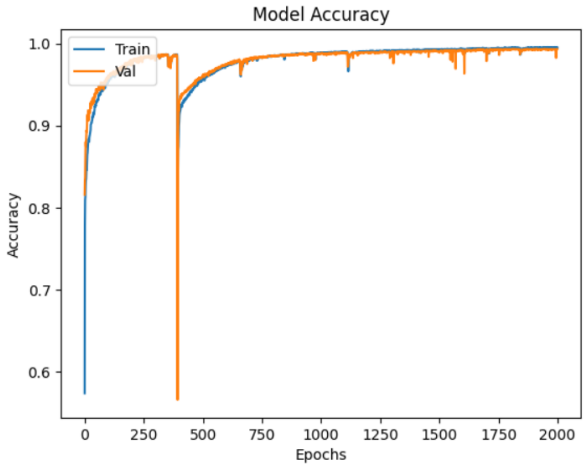
- Biểu đồ hàm loss



Hình 4.34 Biểu đồ hàm loss thuật toán CNN trường hợp 8 nhãn

Nhận xét : *Dựa vào biểu đồ hàm loss của mô hình CNN trong bài toán 8 nhãn ta có thể thấy rằng khoãng cách giữa độ mất mát giữ loss train và loss val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit.*

- Biểu đồ hàm accuracy



Hình 4.35 Biểu đồ hàm accuracy thuật toán CNN trường hợp 8 nhãn

Nhận xét : *Dựa vào biểu đồ hàm Accuracy của mô hình CNN trong bài toán 8 nhãn ta có thể thấy rằng khoãng cách giữa độ chính xác giữ acc train và acc val không có sự cách nhau quá lớn. Điều đó nói lên rằng mô hình không rơi vào 2 TH overfit và underfit*

4.3. So sánh mô hình so với các nghiên cứu khác

4.3.1. Đánh giá với bộ dữ liệu 5 nhãn

- Bài báo khoa học so sánh : An Implementation of Intrusion Detection System Using Genetic Algorithm – 2012
- Tác giả : Mohammad Sazzadul Hoque , Md. Abdul Mukit và Md. Abu Naser Bikas

	Normal	Probe	Dos	U2R	R2L
Bài báo	69.5%	71.1%	99.4%	18.9%	5.4%
Chúng tôi	99%	80.6%	93.6%	10%	53.5%

Nhận xét: Về tổng thể kết quả của nhóm có phần ổn định và tốt hơn so với bài báo của tác giả Mohammad Sazzadul Hoque. Với kết quả được công bố trên bài báo của tác giả Mohammad Sazzadul Hoque có thể thấy rằng 2 TH nhãn U2R và R2L cho kết quả ở mức trung bình thấp và kết quả mức khá với 2 nhãn Normal và Probe và tốt tại nhãn Dos. Với nhóm chúng em nhóm cho kết quả mức trung bình thấp chỉ mỗi nhãn U2R và mức khá với R2L và mức tốt tại Normal, Probe, DOS. Để giải thích cho sự khác biệt này có thể do mô hình nhóm làm và nghiên cứu hiện tại (2023) phần lớn là Deep Learning do đó cho khả năng học và dự đoán của mô hình tốt hơn là Machine Learning mà tác giả Mohammad Sazzadul Hoque đã sử dụng vào năm 2012.

5. KẾT LUẬN

Trong bài báo cáo này, chúng em đã trình bày và đề xuất triển khai hệ thống phát hiện xâm nhận bằng cách áp dụng kĩ thuật học sâu (deep-learning) để phát hiện nhiều loại và trường hợp xâm nhập khác nhau. Để triển khai và đo luờng hiệu suất hệ thống, chúng em đã sử dụng bộ dữ liệu NLS-KDD và đã đạt được tỷ lệ phát hiện khá cao trong các trường hợp. Trong 4 thuật toán chúng em triển khai, neuron network là mô hình phù hợp nhất và cho ra kết quả tốt nhất so với 3 thuật toán còn lại, tuy nhiên neuron network cũng có tương đối nhiều hạn chế trong việc dự đoán và triển khai. Chúng em hi vọng rằng bài báo cáo này có thể phần nào đó góp phần bảo đảm hệ thống an ninh mạng ở Việt Nam và thế giới. Trong tương lai, chúng em sẽ tiếp tục cải tiến và phát triển mô hình ngày càng tối ưu hóa hơn, bên cạnh đó chúng em sẽ cải thiện mô hình làm sao hệ thống có thể tự lấy dữ liệu trực tiếp về và tự học nhằm giúp hệ thống có thể nhanh chóng cập nhập và cải tiến để có khả năng phát hiện nhiều cuộc tấn công mạng đang ngày càng nhiều và khó để phát hiện hơn.

REFERENCS

[1] Check Point Research Team (2023). Check Point Research Report a 38% Increase in 2022 Global Cyverattacks[online], view 23 December 2023. From Check Point

[2] Niv DavidPur (2022). Which countries are most dengerous? Cyber attack origin [online], view 23 December 2023. From : CyberProof

[3] http://nsl.cs.unb.ca/NSL-KDD/

[4] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani “A Detailed Analysis of the KDD CUP 99 Data Set”, Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)