HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

LÊ HOÀNG BẢO

PHÂN LOẠI LƯU LƯỢNG MẠNG INTERNET DÙNG MACHINE LEARNING

LUẬN VĂN THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

TP. HỒ CHÍ MINH - NĂM 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

LÊ HOÀNG BẢO

PHÂN LOẠI LƯU LƯỢNG MẠNG INTERNET DÙNG MACHINE LEARNING

Chuyên ngành: HỆ THỐNG THÔNG TIN
Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT
(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC:
TS. NGUYỄN HỒNG SƠN

TP. HỒ CHÍ MINH - NĂM 2022

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu khoa học của riêng tôi. Các số liệu sử dụng phân tích trong luận án phải có nguồn gốc rõ ràng, đã công bố theo đúng quy định. Kết quả nghiên cứu trong luận án do tôi tự tìm hiểu, phân tích một cách trung thực, khách quan. Ngoài ra kết quả này phù hợp với thực tiễn của Việt Nam. Các kết quả này chưa từng được công bố trong bất kỳ nghiên cứu nào khác.

TP HCM, ngày 25 tháng 01 năm 2022

Học viên thực hiên luận văn

Lê Hoàng Bảo

LỜI CẢM ƠN

Trong quá trình thực hiện đề tài "**Phân Loại Lưu Lượng Internet Dùng Machine Learning**.", Tôi đã nhận được rất nhiều sự giúp đỡ, tạo điều kiện của tập thể lãnh đạo, cán bộ, giảng viên, cán bộ các phòng, ban chức năng Trường Học Viện Công Nghệ Bưu Chính Viễn Thông Cơ Sở Hồ Chí Minh.Tôi xin bày tỏ lòng cảm ơn chân thành về sự giúp đỡ đó.

Tôi xin bày tỏ lòng biết ơn sâu sắc tới **TS.Nguyễn Hồng Sơn** thầy giáo trực tiếp hướng dẫn và chỉ bảo cho Tôi hoàn thành luận án này.

Tôi xin chân thành cảm ơn bạn bè, đồng nghiệp của Tôi đang công tác tại VNPT Tây Ninh và gia đình đã động viên, khích lệ, tạo điều kiện và giúp đỡ Tôi trong suốt quá trình thực hiện và hoàn thành luận án này.

TP HCM, ngày 25 tháng 01 năm 2022

Học viên thực hiên luận văn

Lê Hoàng Bảo

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ,CHỮ VIẾT TẮT

Viết Tắt	Tiếng Anh	Tiếng Việt
VPN	Virtual Private Network	Mạng riêng ảo
ML	Machine learning	Học máy
ToS	Terms Of Service	Điều khoản sử dụng
DSCP	Differentiated Services Code	Điểm mã dịch vụ phân biệt
MPLS	Multiprotocol label switching	Chuyển đổi nhãn đa giao thức
IANA	Internet Assigned Numbers Authority	Tổ chức cấp phát số hiệu Internet
SMTP	Simple Mail Transfer Protocol	Giao thức chuyển thư đơn giản
POP3	Post Office Protocol 3	Giao thức Bưu điện 3
HTTPS	Hypertext transfer protocol	Giao thức truyền siêu văn bản an toàn
KNN	K-Nearest	K lân cận
ANN	Artificial Neural Networks	mạng neural
SMPTS	Simple Mail Transfer Protocol	Giao thức chuyển thư đơn giản Bảo mật
RF	Random Forest	Rừng ngẫu nhiên
IMAPS	Internet Message Access Protocol	Giao thức truy cập tin nhắn Internet

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

MỞ ĐẦU

Phân loại lưu lượng mạng Internet từ lâu đã là một trong những vấn đề được quan tâm hàng đầu trong cộng đồng nghiên cứu và phát triển mạng Internet. Có rất nhiều phương pháp khác nhau được đề xuất để phân loại lưu lượng mạng Internet nhằm quản lý vấn đề bảo mật cũng như đảm bảo chất lượng sử dụng dịch vụ (Quality of Service – QoS). Tuy nhiên, một số phương pháp phân loại truyền thống, ví dụ như phương pháp điều chỉnh Giao thức điều khiển truyền nhận/Giao thức liên mạng (TCP/IP), đã không còn phù hợp do mức độ phức tạp trong quy trình quản lý của mạng Internet. Ngoài ra, những phương pháp khác như phương pháp phân loại dựa trên cổng (port-based) và phương pháp kiểm tra gói chuyên sâu (Deep Packet Inspection - DPI) lại có những hạn chế trong việc xử lý các đặc điểm mới trong lưu lượng mạng (phân bố cổng động, VPN, mã hóa, …).

Thay vào đó, trong vài năm trở lại đây, việc áp dụng các phương pháp học máy (Machine Learning – ML) nhằm mục đích phân tích và phân loại lưu lượng mạng Internet đã đạt được những kết quả đáng chú ý. Các phương pháp này sử dụng các công cụ phân loại thống kê để xây dựng các mô hình phân loại dựa trên các tập dữ liệu huấn luyện đã được gắn nhãn. Kết quả đưa ra từ những mô hình này là nhóm đối tượng hoặc xác suất phân bố của các nhóm đối tượng. Các phương pháp máy học dùng đặc trưng đầu vào để tiến hành huấn luyện, mô hình phân loại dựa vào các thuật toán khác nhau. Với khả năng xử lý nhiều thông tin phức tạp từ nhiều đặc trưng khác nhau, các mô hình học máy có thể phân loại các dữ liệu đầu vào với độ chính xác khá cao. Điều này dẫn đến việc sử dụng mô hình học máy là một xu hướng ngày càng trở nên phổ biến và được áp dụng vào nhiều ứng dụng, lĩnh vực khác nhau.

Chương 1: NGHIÊN CỨU TỔNG QUAN

1.1 Nhu cầu phân tích lưu lượng mạng Internet

Trong lĩnh vực phân loại lưu lượng Internet, những phương pháp truyền thống có một số hạn chế nhất định. Đầu tiên, đánh dấu gói (packet marking) được đề xuất để phân biệt lưu lượng dựa trên lớp QoS của nó. Một số ví dụ về các trường được sử dụng để đánh dấu gói là Loại dịch vụ (Type of Sevice - ToS), Điểm mã dịch vụ phân biệt (Differentiated Services Code Point - DSCP) và Thông báo tắc nghẽn rõ ràng (Explicit Congestion Notification - ECN). Từ đấy, một số giao thức đã được đề xuất để phân loại lưu lượng bao gồm Dịch vụ khác biệt (DiffServ), Dịch vụ tích hợp (IntServ) và Chuyển mạch nhãn đa giao thức (MPLS). Tuy nhiên, các giao thức này không được triển khai và sử dụng một cách rộng rãi do sự phức tạp và các vấn đề tương thích với hệ thống của chúng.

Ngoài ra, có hai phương pháp phân loại truyền thống được ứng dụng rộng rãi, bao gồm phương pháp phân loại dựa trên cổng (Port – based) và phương pháp phân loại dựa trên tải trọng (Payload – based).

Phân loại dựa trên cổng (Port-based technique): Kỹ thuật phân loại dựa trên cổng là kỹ thuật phổ biến và thông dụng nhất để phân loại lưu lượng mạng Internet. Trong kỹ thuật này, mỗi một gói dữ liệu (packet) trong lưu lượng mạng IP đều mang số cổng (số cổng nguồn và số cổng đích) do tổ chức IANA (Internet Assigned Number Authority – Tổ chức cấp phát số hiệu Internet) ấn định. Các ứng dụng mạng Internet nổi tiếng đều đã đăng ký số cổng tại IANA, và bằng cách này, lưu lượng mạng được xác định tương ứng với số cổng đã đăng ký. Ví dụ: các ứng dụng Email sử dụng số cổng 25 (SMTP) để gửi email và cổng 110 (POP3) được sử dụng để nhận email, các ứng dụng web sử dụng số cổng 80 [4]. Tuy nhiên, không phải tất cả các ứng dụng sử dụng mạng Internet đều đã đăng ký số cổng. Một số ứng dụng thế hệ mới như mạng ngang hàng (Peer-to-Peer, hoặc P2P Network), ứng dụng 1 chơi game trực tuyến đều không có đăng ký số cổng cố định, mà sử dụng số cổng động (dynamic port number). Ngoài ra, một số dịch vụ mạng đường hầm (tunneling)

và ẩn danh (anonymization), lại ẩn đi thông tin số cổng của mình [1], [5]. Hơn nữa, trong các ứng dụng di động, hầu hết lưu lượng ứng dụng được truyền đi bằng đường hầm thông qua Giao thức truyền tải siêu văn bản an toàn (Hypertext Transfer Protocol Secure - HTTPS) [6]. Do đó, rất khó để phân loại loại ứng dụng như vậy bằng kỹ thuật dựa trên cổng.

Phân loại dựa trên nội dung truyền tải (Payload-based technique): thường được biết đến dưới cái tên phương pháp kiểm tra gói chuyên sâu (Deep Packet Inspection - DPI). Trong kỹ thuật này, nội dung của gói dữ liệu được kiểm tra dựa trên đặc trưng của các ứng dụng mạng trong lưu lượng Internet. Kỹ thuật này đặc biệt được đề xuất cho các ứng dụng Peer-to-Peer (P2P), hoặc cho những ứng dụng tương đương có sử dụng số cổng động nhằm xác định lưu lượng mạng Internet. Tuy nhiên, phương pháp này cũng có những hạn chế nhất định. Kỹ thuật này yêu cầu nhiều về phần cứng nhằm phát hiện những đặc trưng trong gói dữ liệu, DPI không thể xử lý được những gói lưu lượng truyền tải dữ liệu đã được mã hóa [7], [8], và cần được cập nhật liên tục những đặc trưng cụ thể của những ứng dụng mạng mới phát triển.

Những hạn chế trên đòi hỏi cần một giải pháp mới trong lĩnh vực phân loại lưu lượng mạng Internet nhằm đạt được những kết quả tích cực hơn. Điều này dẫn tới ứng dụng mô hình học máy được sử dụng như một giải pháp cho vấn đề này. Trên thế giới, đã từng có nhiều công trình nghiên cứu trong lĩnh vực phân loại lưu lượng Internet bằng cách áp dụng mô hình huấn luyện học máy. Trong đó, các loại thuật toán học máy khác nhau được sử dụng để phân loại lưu lượng nhằm đáp ứng những nhu cầu khác nhau trong ứng dụng phân loại lưu lượng truy cập mạng.

1.2 Các phương pháp tiền xử lý dữ liệu

Điều chỉnh thang đo đặc trưng (feature scaling) là một phương pháp được sử dụng để chuẩn hóa phạm vi của các biến độc lập hoặc các đặc trưng của dữ liệu. Trong lĩnh vực xử lý dữ liệu, quá trình này còn được gọi là chuẩn hóa dữ liệu (normalization) và thường được tiến hành trong bước tiền xử lý tập dữ liệu. Thang đo của các mẫu giá trị trong tập dữ liệu ban đầu đa phần thường rất phân tán, dẫn đến hiệu quả của các hàm mục tiêu (objective funciton) sẽ giảm sút nếu không áp dụng

quá trình chuẩn hóa. Do đó, thang đo đặc trưng của các mẫu dữ liệu cần được chuẩn hóa sao cho mỗi giá trị đều mang lại đóng góp tương ứng với vị trí của chúng trong phạm vi chuẩn hóa.

1.2.1 Phương pháp chuẩn hóa

Chuẩn hóa tối thiểu – tối đa (Min – Max Normalization): Chuẩn hóa tối thiểu – tối đa là phương án chuẩn hóa đơn giản nhất, nhưng lại được áp dụng khá nhiều trong các bài toán tiền xử lý dữ liệu nhằm mục đích đưa thang đo các giá trị trong tập dữ liệu về mức $[0,1]$. Tuy nhiên, tùy theo yêu cầu cũng như đặc trưng cơ bản của tập dữ liệu, thang đo mục tiêu để điều chỉnh cũng khác nhau. Công thức điều chỉnh thang đo của các giá trị về mức cơ bản $[0,1]$ được miêu tả như sau, trong đó min, max lần lượt là các giá trị nhỏ nhất và lớn nhất xuất hiện trong tập dữ liệu:

$$x' = \frac{x - min}{max - min}, \tag{1.1}$$

Để điều chỉnh lại phạm vi của các giá trị về mức $[a, b]$ tùy theo yêu cầu của bài toán tiền xử lý dữ liệu, công thức (1.1) có thể được điều chỉnh thành:

$$x' = a + \frac{(x - \min)\ (b - a)}{max - min}, \tag{1.2}$$

Trong cả hai công thức trên, x đại diện cho giá trị gốc của đặc trưng của dữ liệu, và x'đại diện cho giá trị tương ứng của đặc trưng đó sau khi chuẩn hóa.

Chuẩn hóa trung bình (Mean Normalization): Ngoài phương pháp chuẩn hóa tối thiểu – tối đa, một phương án khác được rất nhiều các chuyên gia trong lĩnh vực xử lý dữ liệu là chuẩn hóa trung bình, được miêu tả trong công thức (1.3) với $mean$ là giá trị trung bình của từng đặc trưng tương ứng có trong tập dữ liệu:

Chuẩn hóa Z – score (Độ lệch chuẩn): Trong lĩnh vực học máy, tập dữ liệu của bài toán phân loại có thể tồn tại nhiều loại dữ liệu khác nhau, ví dụ: tín hiệu âm thanh và giá trị pixel cho dữ liệu hình ảnh và dữ liệu này có thể bao gồm nhiều đặc trưng khác nhau với các giá trị tồn tại trong các phạm vi khác nhau.

Chuẩn hóa Z-score cho phép giá trị của các đặc trưng trong tập dữ liệu tập trung xung quanh vùng có trị trung bình là 0 và độ lệch chuẩn có giá trị là 1. Phương

pháp này được áp dụng rộng rãi tại bước chuẩn hóa trong nhiều thuật toán học máy (ví dụ: máy vectơ hỗ trợ SVM, hồi quy logistic và mạng neuron nhân tạo). Công thức tính toán cũng tương đương với (1.3), khác biệt duy nhất là giá trị độ lệch chuẩn sẽ thay thế tại vị trí mẫu thức:

$$x' = \frac{x - mean}{standard\ deviation}, \tag{1.4}$$

Phương pháp mã hóa (encode)

Trong lĩnh vực Học máy hoặc Khoa học dữ liệu, tập dữ liệu có thể chứa các giá trị văn bản hoặc phân loại mà không phải là định dạng số. Một số ít thuật toán như CATBOAST, cây quyết định có thể xử lý các bài toán phân loại rất tốt với dữ liệu đầu vào ở các định dạng nhãn hay lớp phân loại. Tuy nhiên, hầu hết các thuật toán và mô hình học máy đều mong muốn các giá trị số ở dữ liệu đầu vào của mô hình nhằm đạt được hiệu quả phân loại cao nhất. Do đó, thách thức chính mà các nhà nghiên cứu hoặc phân tích dữ liệu phải xử lý là chuyển đổi dữ liệu văn bản hoặc phân loại thành dữ liệu số, trong khi vẫn tạo ra một thuật toán phù hợp với định dạng đó.

Mã hóa nhãn (Label Encoding): phương pháp này vô cùng đơn giản vì chỉ liên quan đến quá trình gán một con số cho các giá trị chữ hoặc nhãn của lớp phân loại tương ứng. Tuy nhiên, điểm bất lợi của phương pháp này bao gồm việc gán các giá trị số có thể tạo ra một đặc trưng mới không mong muốn dựa trên mối liên quan về độ lớn nhỏ của các giá trị. Ví dụ như các loại thực phẩm khác nhau trong bảng 1.1, các lớp nhãn ban đầu không có sự phân biệt về độ lớn nhỏ.

Bảng 1.1. Minh họa quy trình mã hóa nhãn

Loại thực phẩm (định dạng chữ)	**Loại thực phẩm (định dạng số)**
Thịt	0
Cá	1
Rau	2
Củ	3
Trứng	4
Sữa	5

Tuy nhiên, sau khi gán các giá trị số tương ứng với mỗi lớp nhãn, đặc trưng mới này đã tạo ra sự khác biệt dựa trên độ lớn nhỏ của các giá trị. Ví dụ, ban đầu các loại thực phẩm như thịt, cá, và rau không có sự liên hệ trực tiếp nào theo như bảng 1.1. Sau khi gán các giá tri số tương ứng, thuật toán có thể hiểu rằng các đặc trưng có sự sắp xếp theo thứ tự từ lớn đến nhỏ. Các đặc trưng này sẽ được gán giá trị trọng số với độ lớn không mong muốn, từ đó dẫn đến sự sai lệch kết quả trong quá trình huấn luyện.

Mã hóa One-hot (One-hot Encoding): Mặc dù ưu điểm của mã hóa nhãn là tính đơn giản, ngược lại các thuật toán phân loại có thể hiểu nhầm đặc trưng ban đầu thành mối liên hệ phân cấp theo độ lớn nhỏ của giá trị số được gán. Nhược điểm này có thể được giải quyết bằng phương án tiếp cận khác được biết đến với cái tên mã hóa One-hot.

Trong mã hóa One-hot, mỗi đặc trưng định dạng phân loại lớp hoặc nhãn dữ liệu sẽ được phân thêm các cột dữ liệu và được mã hóa giá trị 1 hoặc 0, tương ứng với các ký hiệu True/False cho mỗi cột dữ liệu. Bảng giá trị 1.2 minh họa quá trình mã hóa One-hot sử dụng các đặc trưng giới thiệu trong bảng 1.1.

Bảng 1.2. Minh họa quá trình Mã hóa One-hot

Loại thực phẩm (định dạng chữ)	**Loại thực phẩm (Thịt)**	**Loại thực phẩm (Cá)**	**Loại thực phẩm (Rau)**	**Loại thực phẩm (Củ)**	**Loại thực phẩm (Trứng)**	**Loại thực phẩm (Sữa)**
Thịt	**1**	0	0	0	0	0
Cá	0	**1**	0	0	0	0
Rau	0	0	**1**	0	0	0
Củ	0	0	0	**1**	0	0
Trứng	0	0	0	0	**1**	0
Sữa	0	0	0	0	0	**1**

Trong bảng 1.2, chỉ những cột dữ liệu có nhãn phân loại tương ứng với các hàng có giá trị giống nhau mới được gán giá trị 1 (đúng nhãn phân loại), trong khi tất

cả các hàng còn lại có nhãn khác sẽ được gán giá trị 0 (sai nhãn phân loai. Quy trình này giải quyết được vấn đề về độ lớn nhỏ của các giá trị được gán cho nhãn phân loại, nhưng lại có nhược điểm là tạo thêm các cột dữ liệu cho tập dữ liệu. Điều này gây ảnh hưởng đặc biệt lớn khi tập dữ liệu chứa nhiều đặc trưng duy nhất cho các nhãn phân loại, dẫn đến hậu quả tiêu tốn tài nguyên tính toán và thời gian trong quá trình xử lý và huấn luyện mô hình.

1.2.2 Vấn đề dữ liệu bị khuyết (missing data)

Hiện tượng mất mát dữ liệu rất phổ biến trong lĩnh vực khoa học dữ liệu và học máy. Cho đến hiện nay, vẫn chưa có một phương án xử lý triệt để hiệu quả trong tất cả các vấn đề mà hiện tượng mất mát dữ liệu mang đến. Hiện tượng này có thể ảnh hưởng trực tiếp đến các mô hình huấn luyện tùy thuộc vào phương pháp xử lý được áp dụng, vì bản chất việc mất mát dữ liệu cũng có thể mang yếu tố quan trọng với mô hình huấn luyện hoặc thậm chí bản thân bài toán phân loại.

Nguyên nhân dẫn đến hiện tượng mất mát dữ liệu có thể là ngẫu nhiên do mất mát giá trị bên trong tập dữ liệu hoặc sai phạm trong quá trình ghi nhận dữ liệu. Về cơ bản, hiện tượng mất mát dữ liệu có thể chia làm 4 loại, bao gồm mất mát hoàn toàn ngẫu nhiên (Missing completely at random – MCAR), mất mát ngẫu nhiên (Missing at random – MAR), mất mát không ngẫu nhiên (Missing not at random – MNAR), và mất mát mang tính hệ thống (Structurally missing). Tùy thuộc vào bản chất của hiện tượng mất mát dữ liệu mà mô hình có thể áp dụng các phương án xử lý phù hợp.

Xóa toàn bộ hàng có dữ liệu thiếu: Phương án đơn giản nhất để xử lý vấn đề mất mát dữ liệu là xóa toàn bộ cột hoặc hàng có giá trị bi thiếu, nếu cột hoặc hàng dữ liệu đó có nhiều hơn một nửa số giá trị bị mất mát. Ưu điểm của phương pháp này là làm mạnh mô hình huấn luyện bằng cách loại bỏ hết tất cả các trường dữ liệu bị thiếu. Tuy nhiên, nhược điểm của phương pháp này có thể kể đến là làm mất đi rất nhiều thông tin trong tập dữ liệu, đồng thời làm giảm hiệu suất huấn luyện của mô hình nếu số lượng thông tin bị xóa đi chiếm lượng lớn trong tập dữ liệu.

Thay thế bằng giá trị Trung bình/trung vị: Nếu các giá trị được ghi nhận trong cột dữ liệu thuộc định dạng số, thì các vị trí có dữ liệu bị mất mát có thể được thay thế bằng các giá trị trung bình (mean) hoặc trung vị (median) của các giá trị còn lại. Phương pháp này phòng ngừa hiện tượng mất mát thông tin trong quá trình xóa toàn bộ hàng hoặc cột dữ liệu, có thể áp dụng đơn giản và hiệu quả với các tập dữ liệu nhỏ. Ngược lại, phương án này chỉ có thể áp dụng nếu định dạng dữ liệu là định dạng số, tạo nên độ sai lệch nhất định trong quá trình huấn luyện, đồng thời chưa cân nhắc đến sự ảnh hưởng của hiệp phương sai (covariance) giữa các điểm dữ liệu.

Thay thế bằng giá trị Yếu vị: Tương tự như phương pháp trên, phương án này thay thế các giá trị bị thiếu bằng giá trị yếu vị (mode) dựa trên thông tin của các giá trị khác trong cột hoặc hàng dữ liệu. Tuy nhiên, phương pháp này thường dược sử dụng với các tập dữ liệu có giá trị thuộc định dạng chữ hoặc mang giá trị phân loại. Ngoài các ưu và nhược so với phuong pháp trên, nếu các giá trị thay thế chiếm tỷ lệ cao trong một cột dữ liệu, mô hình thậm chí có thể xem đó là một đặc trưng phân loại dữ liệu mới.

Mô hình không bị ảnh hưởng bởi mất mát dữ liệu: Một số thuật toán học máy không bị ảnh hưởng bởi các điểm giá trị thiếu trong tập dữ liệu. Mô hình K-NN có thể bỏ qua các điểm bị thiếu từ một cột giá trị bằng thước đo khoảng cách đến cột dữ liệu đó. Naive Bayes cũng bao gồm thuật toán chống lại sự ảnh hưởng của các giá trị còn thiếu trong quá trình đưa ra dự đoán. Một thuật toán khác là Rừng ngẫu nhiên hoạt động tốt trên các tập dữ liệu phi tuyến tính và phân loại. Mô hình này thích ứng với cấu trúc dữ liệu có xem xét đến các giá trị phương sai hoặc độ lệch, cho ra kết quả tốt hơn trên các tập dữ liệu lớn.

1.3 Một số thuật toán học máy được áp dụng vào phân loại lưu lượng

Phân loại lưu lượng mạng Internet sử dụng phương pháp học máy đã được đề xuất để khắc phục các hạn chế của phương pháp DPI và phương pháp phân loại dựa trên cổng [9–11]. Các phương pháp này đã cho thấy hiệu quả của chúng để phân loại lưu lượng mạng Internet thậm chí trên những mảng dữ liệu đã được mã hóa. Trên thực tế, quá trình học máy chủ yếu dựa vào việc học các mẫu khác biệt đặc trưng từ

những đặc điểm trong lưu lượng truy cập. Trong bối cảnh này, một số công trình sơ bộ đã được báo cáo để phân tích lưu lượng nhằm mục đích mô tả tính chất các tập dữ liệu [12], mô hình hóa lưu lượng mạng Internet [13], và trích xuất các mẫu đặc trưng cụ thể [14].

Một số nhà nghiên cứu đang đặc biệt xem xét kỹ việc áp dụng kỹ thuật học máy (ML) (một phân nhánh của lĩnh vực Trí tuệ nhân tạo) để phân loại lưu lượng Internet. Việc áp dụng các kỹ thuật học máy bao gồm một số bước. Đầu tiên, các đặc trưng được ghi nhận bởi lưu lượng Internet không xác định mà trong tương lai có thể được xác định và phân biệt. Đặc trưng là các thuộc tính của các giao thức dữ liệu được tính trên nhiều gói (chẳng hạn như độ dài gói tối đa hoặc tối thiểu theo mỗi chiều truyền dữ liệu, thời lượng truyền tải dữ liệu hoặc thời gian đến giữa các gói truyền tải). Sau đó, bộ phân loại được huấn luyện để liên kết những tập hợp các tính năng với các lớp lưu lượng đã biết trước đặc điểm (tạo quy tắc) và áp dụng thuật toán học máy để phân loại lưu lượng chưa biết bằng cách sử dụng các quy tắc đã học trước đó.

Mỗi thuật toán học máy đều có một cách tiếp cận khác nhau để sắp xếp và phân loại các bộ đặc điểm, dẫn đến các hành vi khác nhau trong quá trình đào tạo và phân loại. Do đó, trong phạm vi đề tài, việc phân loại lưu lượng IP dựa trên nghiên cứu từ những bài báo hội nghị, tạp chí quốc tế có liên quan đến nội dung đề tài, xem xét các phương pháp tiếp cận hiện đại để phân loại lưu lượng, sau đó so sánh và đánh giá các kỹ thuật dựa trên học máy có tính phù hợp cao để phân loại lưu lượng IP. Sau cùng, từ những mô hình học máy được đề cập trong nội dung phân tích, chọn ra một mô hình học máy phù hợp để áp dụng vào đề tài nghiên cứu. Sau đây là một số thuật toán học máy nổi tiếng:

Cây quyết định: Các phương pháp dựa trên cây quyết định (Decision Tree - DT) đã từng được đề xuất nhằm phân loại lưu lượng mạng Internet. Phương pháp phân loại cây quyết định DT là một phương pháp dựa trên quy tắc. Nó chủ yếu bao gồm việc trả lời một chuỗi các câu hỏi ở các nút câu hỏi (non-leaf node) để đưa về phía các nút lá (leaf node), mà tại đó mỗi node lá đại diện cho một nhãn dán được dự

đoán. Tuy nhiên, phương pháp này lại có xu hướng tạo ra hiện tượng quá khớp (overfitting). Trong nghiên cứu của mình, Yuan và Wang áp dụng mô hình Cây quyết định Hadoop nhằm cải thiện độ chính xác (accuracy) của mô hình phân loại lưu lượng Internet trước đó trên cả 8 giao thức truy cập khác nhau [15]. Ở một nghiên cứu khác, bằng cách áp dụng mô hình cây quyết định C4.5, Yongli Ma và các đồng nghiệp của mình đã đạt được kết quả chính xác trung bình lên đến 88.67% trong việc phân loại lưu lượng mạng Internet [16].

Bộ phân loại Naïve Bayes (NB): NB là một phương pháp học máy khác đã được sử dụng để phân loại lưu lượng mạng Internet. NB là một phương pháp xác suất dựa vào định lý Bayes mà tại đó, NB là phương thức đơn giản nhất trong họ phương pháp Bayes. Các phương pháp khác dựa trên định lý Bayes được xây dựng nhằm mô hình hóa và xử lý các tình huống phân tích phức tạp hơn. Trong đó, có sự phụ thuộc giữa các đặc trưng khác nhau để phân loại lưu lượng mạng Internet. Trong bài báo cáo hội nghị năm 2016, Muhammad Shafiq và các đồng nghiệp đã sử dụng công cụ WireShark để tự tạo ra tập dữ liệu lưu lượng mạng Internet và áp dụng một số mô hình học máy để đánh giá khả năng phân loại lưu lượng mạng. Trong số những mô hình được áp dụng như Cây quyết định 4.5, máy vec-tơ hỗ trợ (Support vector machine - SVM),... NB đạt được kết quả đáng chú ý với độ chính xác trung bình ở mức 71.8919% [4].

K lân cận (KNN): KNN là một phương pháp huấn luyện học máy phi tham số. Là một thuật toán được phân vào loại mô hình huấn luyện lười học (lazy learning), KNN không bao gồm giai đoạn huấn luyện. Vì vậy, thời gian phân loại phụ thuộc vào kích thước dữ liệu. Ở giai đoạn phân loại, thuật toán tiến hành phân loại dữ liệu dựa trên việc đo khoảng cách giữa mẫu thử nghiệm với tất cả các mẫu được gắn nhãn. Mẫu thử nghiệm sẽ được gán cho lớp có K – lân cận gần nhất của nó. Dixit và các đồng tác giả đã áp dụng mô hình này cho tập dữ liệu UNSW NB-15, đồng thời so sánh kết quả phân loại song song với các mô hình huấn luyện tổng hợp từ NB như Gaussian Naïve Bayes và Đa thức Naïve Bayes. Kết quả đạt được từ mô hình học máy KNN là cao nhất khi cho ra kết quả chính xác tổng quan lên đến 81.647% [17].

Mạng Nơ-ron (NN): là một phương pháp huấn luyện học máy đã được thiết kế lấy nguồn gốc từ hệ thống thần kinh của con người. Một bài nghiên cứu áp dụng mô hình mạng nơ-ron học sâu trong phân loại lưu lượng Internet vào ứng dụng phát hiện xâm nhập mạng [18] đã cho thấy, các mô hình mạng nơ-ron cũng có thể tạo ra những kết quả đáng chú ý. Li Zhipeng và các đồng nghiệp của mình đã chứng minh 2 mô hình mạng nơ-ron học sâu ResNet 50 và GoogleNet khi so sánh với các kỹ thuật khác như NB, mô hình cây quyết định kết hợp với NB, máy vec-tơ hỗ trợ…vẫn có thể cho ra kết quả tốt nhất. Ví dự như trên tập dữ liệu NSL-KDD Test^{-21} [18], kết quả chính xác của 2 mô hình trên lần lượt là 81.57% và 81.84%.

Những mô hình trên đã liệt kê ra những khả năng thích hợp ứng dụng học máy vào phân loại lưu lượng mạng Internet. Trong phạm vi đề tài luận văn sẽ chỉ tập trung vào một số mô hình phù hợp dựa trên kỹ thuật học có giám sát như NB, K-NN hay ANN, CNN. Bảng 1.1 dưới đây tóm tắt lại các mô hình máy học được áp dụng trong việc phân loại lưu lượng mạng Internet cùng với kết quả độ chính xác tổng quan đạt được.

Chương 2: TỔNG QUAN VỀ HỌC MÁY

2.1 Giới thiệu

Học máy (machine learning hay ML) là một tập hợp con của trí tuệ nhân tạo và được coi là một trong những lĩnh vực trong khoa học máy tính với khả năng tự học dựa trên dữ liệu đầu vào mà không cần phải có sự lập trình cụ thể. Học máy được biết đến như một tập hợp các kỹ thuật khai thác dữ liệu, tìm kiếm và mô tả các mẫu cấu trúc hữu ích trong dữ liệu. Bên cạnh đó, học máy có nhiều ứng dụng, bao gồm công cụ tìm kiếm, chẩn đoán y tế, nhận dạng văn bản và chữ viết tay, sàng lọc hình ảnh, dự báo tài chính, dự báo doanh số, v.v.

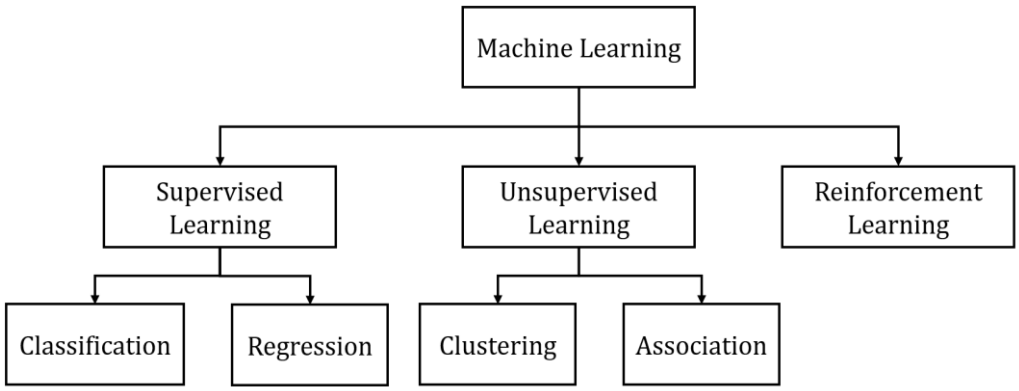
Một cách khái quát, học máy là quá trình tìm kiếm và mô tả các mẫu cấu trúc trong một tập dữ liệu được cung cấp. Máy học sẽ nhận đầu vào dưới dạng một tập hợp dữ liệu của các mẫu (samples). Một điểm dữ liệu có thể là một bức hình, một đoạn âm thanh, một tập hợp các đặt tính của một bài toán, … Các điểm dữ liệu thường được chuyển về dạng tập hợp các con số hay còn được gọi là đặc trưng (feature) mà máy tính có thể học được. Ngoài ra, có những loại dữ liệu được biểu diễn dưới dạng ma trận hoặc mảng nhiều chiều [19].

Trong quá trình xây dựng mô hình, bộ dữ liệu thường được chia thành 2 tập dữ liệu: tập huấn luyện và tập kiểm tra. Tập huấn luyện (training set) là tập dữ liệu bao gồm các đặc trưng của các mẫu dữ liệu được sử dụng trong việc xây dựng mô hình học máy. Tập kiểm tra (test set) gồm các đặc trưng của các dữ liệu được dùng để đánh giá khả năng và hiệu quả của mô hình học máy. Để đánh giá mô hình một cách trung thực, các điểm dữ liệu trong tập kiểm tra không sử dụng trong quá trình huấn luyện mô hình. Việc này đóng vai trò rất quan trọng, vì việc đánh giá dựa trên tập kiểm tra (độc lập so với tập huấn luyện) thể hiện khả năng học của mô hình nhằm thể hiện độ chính xác khi áp dụng thực tế. Mỗi bài toán học máy sẽ gồm có 2 pha lớn: pha huấn luyện (training phase) và pha kiểm tra (test phase). Pha huấn luyện sẽ xây dựng mô hình dựa trên dữ liệu huấn luyện và pha kiểm tra sẽ sử dụng dữ liệu kiểm tra để đánh giá hiệu quả của mô hình.

Đầu ra của mô hình học máy sẽ là mô tả kiến thức đã học. Đầu ra của mô hình sẽ tuỳ vào ứng dụng cụ thể thì đưa ra những hành động khác nhau. Đối với một số bài toán phân loại (classification), mô hình sẽ đưa ra sự tiên đoán về nhãn dựa vào điểm dữ liệu. Đối với một số bài toán hồi quy (regression), đầu ra sẽ là sự dự đoán của kết quả.

2.2 Các phương pháp học trong quá trình học máy

Thông thường, các bài toán học máy sẽ được phân loại như hình dưới đây [19]:



Hình 2.1. Phân loại thuật toán trong máy học

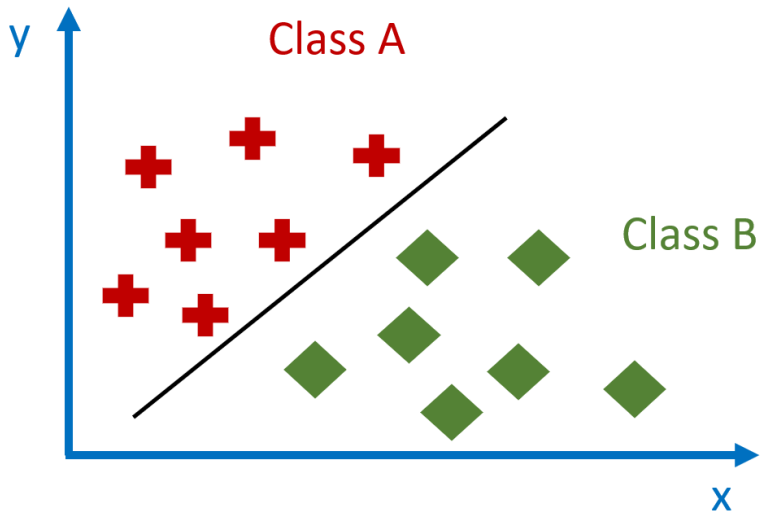
Học có giám sát (supervised learning) là việc xây dựng mô hình học dự đoán các mẫu dữ liệu mới thành các nhãn đã cho trước dựa trên các mẫu dữ liệu huấn luyện. Máy học được cung cấp một tập hợp các mẫu dữ liệu, được phân loại trước thành các lớp/nhãn. Đầu ra của quá trình học tập là một mô hình phân loại được xây dựng bằng cách tổng quát hóa từ tập huấn luyện. Trên thực tế, học có giám sát tập trung vào việc mô hình hóa để xây dựng các mối quan hệ đầu vào / đầu ra. Mục tiêu của phương pháp học này là xác định một ánh xạ từ các đặc trưng cho trước vào một phân lớp ở đầu ra.

Học không giám sát (unsupervised learning) là thuật toán học máy trích xuất được những thông tin quan trọng dựa trên mối liên hệ của các điểm dữ liệu. Nói cách khác, điểm dữ liệu trong phương pháp học này sẽ không được gán nhãn và không có đầu ra tương ứng. Học không giám sát được áp dụng trong các bài toàn phân cụm hay giảm chiều dữ liệu.

Học tăng cường (reinforcement learning) là lĩnh vực liên quan đến việc dạy cho máy (agent) thực hiện tốt một nhiệm vụ (task) bằng cách tương tác với môi trường (environment) thông qua hành động (action) và nhận được phần thưởng (reward). Các bài toán học tăng cường giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance).

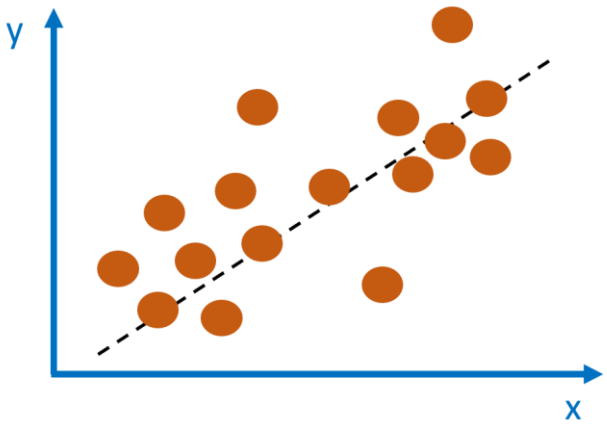
2.3 Các loại bài toán cơ bản trong học máy

Phân loại (classification) là một trong những bài toán được quan tâm và nghiên cứu nhiều nhất trong học máy. Học phân loại là liên quan đến việc học máy từ một tập hợp các mẫu được phân loại trước (còn được gọi là được gắn nhãn trước), từ đó nó xây dựng một tập hợp các quy tắc phân loại (một mô hình) để phân loại các mẫu khác trong tương lai. Đối với các bài toán này, nhiệm vụ được yêu cầu xác định nhãn của một điểm dữ liệu trong số các nhãn khác nhau. Các cặp dữ liệu sẽ được ký hiệu là (x, y) tương đương với (dữ liệu, nhãn). Số nhãn trong tập dữ liệu được ký hiệu là C, khi đó, việc xây dựng mô hình là việc tìm một hàm số f ánh xạ một điểm dữ liệu x vào một phần tử y. Ví dụ trong Hình 2.2 thể hiện minh hoạ cho bài toán phân loại. Việc huấn luyện bài toán phân loại thực chất là đi tìm đường phân chia giữa các nhãn dữ liệu. Từ đó, tạo thành ranh giới phân loại để tiến hành phân loại các điểm dữ liệu khác trong tương lai.



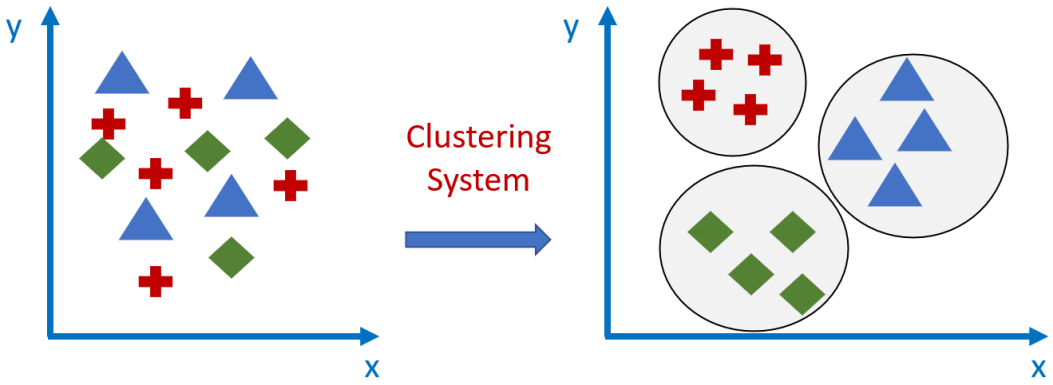
Hình 2.2. Hình minh hoạ cho bài toán phân loại [4]

Hồi quy (regression) bao gồm một tập hợp các phương pháp học máy cho phép chúng ta dự đoán một biến kết quả liên tục y dựa trên giá trị của một hoặc nhiều đặt trưng của điểm dữ liệu x. Hồi quy được dùng để ước tính mối quan hệ giữa mục tiêu (target) và biến (variable) độc lập. Điều này giúp cho việc dự đoán các biến liên tục và được ứng dụng vào các trường hợp thực tế như phân tích giá cả thị trường, khuynh hướng doanh số bán hàng, etc. Hình 2.3 thể hiện minh hoạ bài toán hồi quy. Với các dữ liệu cho trước, mô hình hồi quy sẽ tìm được mối liên hệ phân bố các điểm dữ liệu và từ đó có khả năng dự đoán được đầu ra.



Hình 2.3. Hình minh hoạ cho bài toán hồi quy [1]

Phân cụm (clustering) là thực hiện nhóm các đối tượng có các đặc điểm giống nhau thành các cụm mà không có sự tách động trước nào. Bài toán phân cụm sẽ chia toàn bộ dữ liệu thành các cụm nhỏ dựa trên sự tương quan giữa các dữ liệu trong mỗi cụm. Đối với loại bài toán này, tập dữ liệu huấn luyện sẽ không có gán nhãn và mô hình sẽ tự động thực hiện sự phân chia các điểm dữ liệu thành các cụm khác nhau [20]. Loại máy học này thường được sử dụng như một kỹ thuật phân tích dữ liệu để khám phá các mẫu dữ liệu với nhiều chiều, chẳng hạn như các nhóm khách hàng dựa trên hành vi của họ. Ví dụ trong Hình 2.4 thể hiện sự phân cụm sau khi qua hệ thống phân cụm của mô hình máy học.



Hình 2.4. Hình minh hoạ cho bài toán phân cụm [17]

Association là một phương pháp học không giám sát dựa trên quy tắc (rule-based) hướng tới việc khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước. Nó dựa trên các quy tắc khác nhau để khám phá các mối quan hệ giữa các biến trong tập dữ liệu. Phương pháp học máy này thường được áp dụng vào khai thác sử dụng web, hệ thống gợi ý khách hàng, hành vi mua hàng khách hàng v.v. Nhìn chung, bài toán phân loại là một trong những bài toán được quan tâm và nghiên cứu nhiều trong mảng học máy. Trong các mục kế tiếp, đề tài sẽ khảo sát những mô hình học máy và phân loại tiêu biểu trong lĩnh vực phân loại lưu lượng Internet.

Trích rút luật (Rule Extraction): là bài toán mà dữ liệu được sử dụng để tìm ra các quy luật (nhân - quả). Bài toán này tương tụ như hồi quy, về cơ sở bản chất đều xuất phát từ vấn đề học cách tổng quát hóa và dự đoán của con người. Tuy nhiên thay vì dự đoán các giá trị thực thì trích rút luật tổng quát hóa các luật nhân quả, các mẫu suy luận nếu - thì từ dữ liệu. Các luật này, có thể không trực tiếp thậm chí không thể hiểu được dưới dạng suy luận của con người. Việc tìm ra quy luật dựa trên các nguyên tắc thống kê quan hệ giữa các thuộc tính và dữ liệu mà không liên quan đến các tri thức tiền nghiệm. Một ví dụ kinh điển là bài toán khai phá luật kết hợp tìm được sự liên hệ giữa việc mua bia và mua bỉm (luật này có thể sai hoặc đúng tùy theo các dữ liệu được cung cấp).

2.3.1 K – Lân cận (KNN – K-Nearest Neighbors)

K lân cận (K-nearest neighbor hay KNN) là một trong những thuật toán học máy có giám sát cơ bản và đơn giản nhất. Trong quá trình huấn luyện, thay vì phải học từ những dữ liệu huấn luyện, KNN sẽ ghi nhớ một cách máy móc toàn bộ các dữ liệu đó. Các phép tính toán sẽ được tiến hành trong pha kiểm tra. KNN là thuật toán xác định đầu ra của một điểm dữ liệu mới dựa vào thông tin của K điểm dữ liệu gần nhất trong tập huấn luyện. Ưu điểm của KNN được biết là quá trình dự đoán kết quả của dữ liệu mới tương đối đơn giản sau khi xác định được các điểm lân cận.

Nguyên tắc hoạt động: Phương pháp KNN là một thuật toán phân loại bằng cách nhóm tất cả các mẫu thể hiện những đặc trưng tương tự của tập dữ liệu lại với nhau [21]. Thuật toán KNN sẽ lưu trữ tập dữ liệu huấn luyện trong bộ nhớ, và mỗi khi một mẫu của tập dữ liệu kiểm tra được đưa vào đầu vào để phân loại, thuật toán sẽ phân loại phần tử đó vào nhãn có độ tương tự tối đa và có ít sự khác biệt nhất với các đặc trưng của dữ liệu đầu vào [22]. Quy trình phân loại một mẫu $X = \{x_1, x_2, \dots, x_n\}$, trong đó $x_1, x_2, \dots, x_n$ là những đặc trưng của mẫu, được quy định bằng quy tắc đa số dựa trên số lượng vec-tơ tham chiếu k-lân cận nhất của mẫu X. Thuật toán KNN đặt giả thiết rằng tất cả mẫu thuộc tập dữ liệu đều tương ứng với những điểm tồn tại trong không gian n chiều được ký hiệu bằng $\Re^n$, và khoảng cách giữa những điểm trong chiều không gian trên được định nghĩa bằng chỉ số khoảng cách (distance metrics). Do đó, khoảng cách giữa mẫu X_i và X_j được định nghĩa bằng công thức sau:

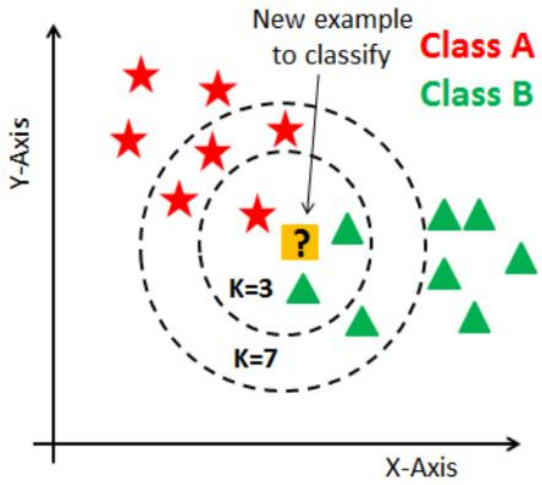
$$d(X_i, X_j) = \left(\sum_{f=1}^{n} \left| x_f^i - x_f^j \right|^p \right)^{1/p}, \tag{2.1}$$

trong dó, x_f^i ký hiệu giá trị tương ứng của số lượng đặc trưng f của mẫu dữ liệu X_i. Tiếp theo, thuật toán sẽ chọn ra k điểm tương ứng với số mẫu trong tập huấn luyện có khoảng cách gần nhất với mẫu cần phân loại tại đầu vào. Nhãn của mẫu X

sẽ được phân loại dựa trên số lượng lớp của k mẫu trên theo quy tắc bình chọn đa số (major voting), nghĩa là X thuộc về lớp có số lượng nhiều hơn trong k mẫu. Trong đó, nếu định nghĩa $C = \{c_i | 1 \leq i \leq m, m > 0\}$ là tập hợp số lượng lớp có trong tập dữ liệu được dán nhãn trước, thì mục tiêu phân loại mẫu X_q của mô hình KNN có thể được khái quát hóa bằng công thức (2.2), trong đó $\hat{f}(X_q)$ là kết quả phân loại của mẫu X_q.

$$\hat{f}(X_q) \leftarrow argmax \sum_{i=1}^{k} f(c, X_i) \tag{2.2}$$

.



Hình 2.5. Quá trình phân loại bằng phương pháp K-NN, khi $k = 3$ và $k = 7$

Mẫu thử nghiệm cần được phân loại thành 1 trong 2 lớp tam giác màu xanh hoặc ngôi sao màu đỏ. Nếu $k = 3$ (đường tròn nhỏ) thì nó sẽ thuộc lớp hình tam giác màu xanh vì có 2 hình tam giác và chỉ có 1 hình ngôi sao bên trong hình tròn có 3 điểm gần nhất. Nếu $k = 7$ (đường tròn lớn) thì nó được gán nhãn hình ngôi sao (4 hình ngôi sao so với 3 hình tam giác bên trong hình tròn bên ngoài).

Chỉ số khoảng cách: Để thuật toán KNN hoạt động hiệu quả nhất trên một tập dữ liệu cụ thể, chúng ta cần chọn phương án tính chỉ số khoảng cách tương ứng thích hợp nhất. Công thức số (2.1), hay còn được biết đến như là khoảng cách Minkowski, là chỉ số dành cho không gian vec-tơ có giá trị thực. Khoảng cách Minkowski chỉ có thể được áp dụng trong không gian vec-tơ chuẩn tắc, có nghĩa là trong không gian mà khoảng cách có thể được biểu diễn dưới dạng vectơ có độ dài có giá trị không âm. Giá trị p trong công thức (2.1) có thể được thay đổi để đạt được 1 trong 2 chỉ số: Khoảng cách Manhattan với giá trị $p = 1$, và khoảng cách Euclid, với giá trị $p = 2$.

Khoảng cách Manhattan giữa hai điểm là tổng của sự khác biệt tuyệt đối của tọa độ Descartes của chúng, được biểu diễn bằng công thức (2.3). Khoảng cách Manhattan được sử dụng khi tệp dữ liệu có số chiều không gian nhiều.

$$d(X_i, X_j) = \sum_{f=1}^{n} |x_f^i - x_f^j|, \tag{2.3}$$

Khoảng cách Euclid chỉ số được sử dụng rộng rãi nhất được sử dụng cho KNN. Nó là thước đo khoảng cách đường thẳng thực giữa hai điểm trong không gian Euclid, được biểu diễn bằng công thức (2.4).

$$d(X_i, X_j) = \sqrt{\sum_{f=1}^{n} \left(x_f^i - x_f^j\right)^2}, \tag{2.4}$$

Trọng số điểm lân cận: Bộ phân loại KNN có thể được xem như gán trọng số $1/k$ bằng nhau cho những điểm k lân cận nhất với mẫu kiểm tra, và tất cả những điểm còn lại trong tập huấn luyện đều có trọng số bằng 0. Tuy nhiên, nhằm mục tiêu gia tăng hiệu suất phân loại, đề tài có thể áp dụng Bộ phân loại trọng số điểm lân cận. Trong đó, trong k điểm lân cận, bộ phân loại sẽ gia tăng độ tin cậy của những mẫu huấn luyện có khoảng cách gần hơn với mẫu kiểm trả tại đầu vào so với những mẫu có khoảng cách xa hơn. Trọng số dựa vào khoảng cách của mẫu lân cận X_i có thể được biểu diễn bằng công thức:

$$\omega_i = \frac{1}{d(X_q, X_i)^2}, \tag{2.5}$$

Từ đó, quá trình phân loại mẫu kiểm tra X_q trong công thức (2.2) có thể được viết lại dưới ảnh hưởng của trọng số ω_i như sau:

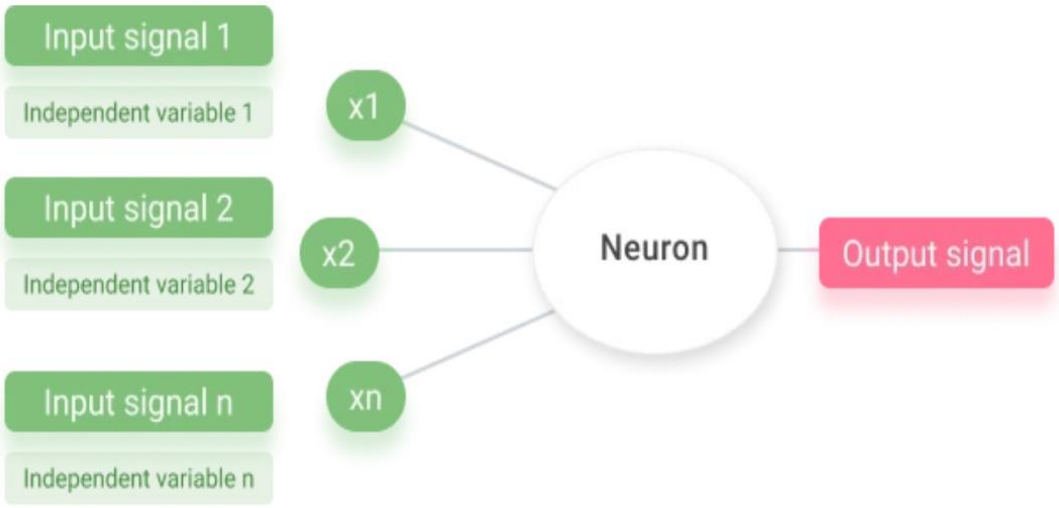
$$\hat{f}(X_q) \leftarrow argmax \sum_{i=1}^{k} \omega_i f(c, X_i) \tag{2.6}$$

2.3.2 Mạng Neuron nhân tạo (ANN – Artificial Neural Networks)

Mạng neuron nhân tạo (Artificial Neural Network hay ANN) là một thuật toán của máy học mô phỏng dựa trên hoạt động thần kinh sinh học. ANN bao gồm 3 lớp chính: lớp đầu vào (input layer), lớp ẩn (hidden layer) và lớp đầu ra (output layer). Mỗi layer gồm nhiều neuron được gắn nối với nhau để xử lý các thông tin. Mỗi neuron đều gồm có đầu vào dữ liệu (Inputs) để nhận và xử lý cho ra một kết quả ở đầu ra (Outputs). Bên cạnh đó, đầu ra của một neuron hay kết quả xử lý của một neuron có thể làm đầu vào cho các neuron khác. Mỗi neuron đều có trọng số (weight) và độ lệch (bias). Việc huấn luyện (training) mạng là quá trình tinh chỉnh những trọng số liên kế. Các trọng số này ban đầu sẽ được mặc định ngẫu nhiên, sau đó, quá trình huấn luyện được thực thi để tối ưu các trọng số trên. Một trong những lợi ích của việc sử dụng mạng ANN là có khả năng áp dụng ở những bài toán phi tuyến tính.

Bước đầu tiên để nắm được chức năng của mạng neuron nhân tạo (Artificial Neural Network) là hiểu được nguyên lý hoạt động của các neuron, hay còn được gọi là tế bào thần kinh. Mạng thần kinh trong khoa học máy tính bắt chước các tế bào thần kinh thực tế của não người, do đó có tên là mạng "thần kinh". Mỗi tế bào thần kinh có các nhánh dữ liệu đầu vào (input, ký hiệu là $\boldsymbol{x_i}$) để tiếp nhận thông tin (data) và xử lý cho kết quả đầu ra (output, ký hiệu là $\boldsymbol{y_i}$). Một tế bào thần kinh không thể xử lý được nhiều thông tin, nhưng khi hàng nghìn tế bào thần kinh kết nối và hoạt động cùng nhau, mạng thần kinh sẽ hoạt động rất mạnh mẽ và có thể xử lý các nhiệm vụ và khái niệm phức tạp. Tương tự như vậy, một nút mạng máy tính hoạt động giống như cách một tế bào thần kinh của con người hoạt động, và mạng máy tính tái tạo hệ

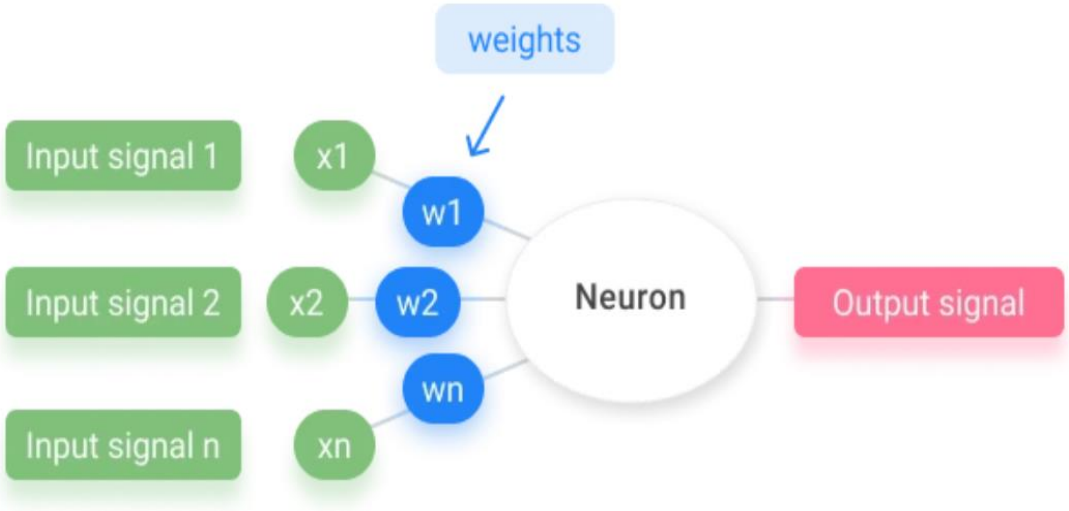
thống các tế bào thần kinh thực, giúp xử lý các tác vụ như tiếp nhận thông tin, truy xuất đặc trưng, xử lý thông tin đặc trưng, tiến hành phân loại…. trong các bài toán khác nhau. Hình 2.6 biểu diễn mối liên kết của giá trị đầu vào đến một nút mạng neuron, từ đó tạo ra giá trị đầu ra.



Hình 2.6. Hình ảnh thể hiện đặc điểm dữ liệu đầu vào và đầu ra của một neuron thần kinh

Đối với mạng neuron máy tính, các giá trị độc lập (tín hiệu đầu vào) được chuyển đến nút mạng neuron nhằm tạo ra giá trị phụ thuộc (tín hiệu đầu ra). Các biến độc lập này trong một lớp (layer), là tập hợp của một hàng dữ liệu cho một lần quan sát duy nhất. Trong một ví dụ ứng dụng mạng neuron xác định danh tính cá nhân, một lớp đầu vào sẽ biểu thị một biến - có thể là tuổi hoặc giới tính (biến độc lập) của một người có danh tính (phụ thuộc) mà bài toán cần xử lý và xác định danh tính. Mạng nơ-ron này sau đó được áp dụng nhiều lần với số lượng điểm dữ liệu mà bài toán cung cấp trên mỗi biến độc lập. Giá trị đầu ra có thể là biến liên tục, nhị phân hoặc biến phân loại. Điều kiện tiên quyết là những giá trị đầu ra đó phải tương ứng với nhóm dữ liệu đầu vào dưới dạng các biến độc lập. Về bản chất, một loại biến độc lập tương ứng với một loại biến đầu ra. Các biến đầu ra đó có thể giống nhau đối với các hàng dữ liệu khác nhau trong khi các biến đầu vào thì không. Kết quả của các biến đầu ra cũng có thể được sử dụng để làm biến đầu vào cho các neuron khác.

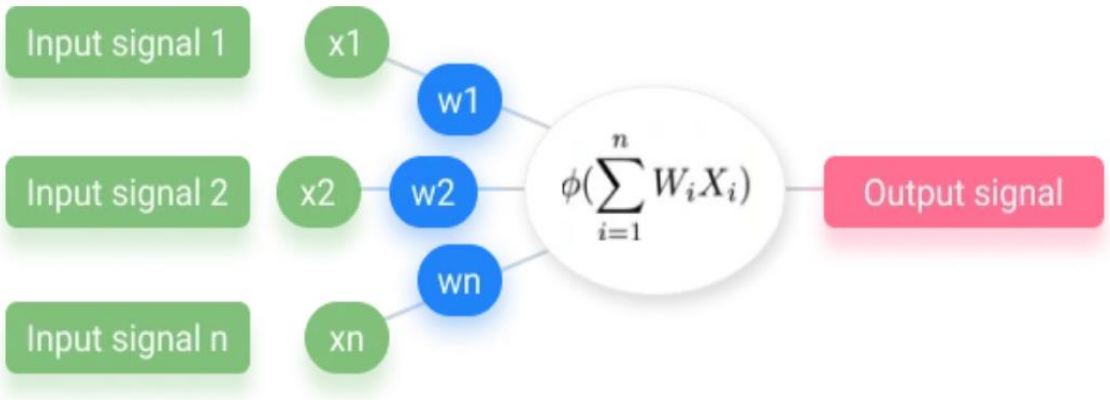
Trọng số (Weight): Các tín hiệu đầu vào được nối với các nút neuron qua các khớp thần kinh (synapses). Mỗi giá trị tín hiệu đầu vào được đính kèm với một giá trị trọng số tương ứng, ký hiệu là $\boldsymbol{w_i}$, thể hiện tầm quan trọng của giá trị đầu vào đó tại nút neuron, so với các giá trị đầu vào khác. Các trọng số quyết định tín hiệu đầu vào nào không quan trọng - tín hiệu nào được truyền đi và tín hiệu nào không. Trọng số là các giá trị rất quan trọng đối với mạng neuron nhân tạo vì chúng cho phép các mạng "học hỏi" tầm quan trọng của các đặc trưng của mẫu dữ liệu. Hình 2.7 miêu tả vị trí của các trọng số tương ứng với từng giá trị dữ liệu đầu vào trong liên kết với nút mạng neuron.



Hình 2.7. Minh họa giá trị trọng số tương ứng với các giá trị đầu vào của một neuron

Hàm kích hoạt (Activation function): Bên trong mỗi tế bào neuron, mạng máy tính sẽ lấy tổng tất cả giá trị dữ liệu đầu vào, kèm với trọng số tương ứng với mỗi giá trị đó. Sau đó, mỗi một nút mạng neuron sẽ áp dụng một hàm kích hoạt lên trên giá trị tổng trọng số (weighted sum) tính toán được. Hàm kích hoạt là một hàm ánh xạ các giá trị đầu vào của một nút neuron với đầu ra tương ứng. Tùy thuộc vào kết quả của hàm được áp dụng, neuron sẽ truyền tín hiệu hoặc không truyền tín hiệu đến nút mạng kế tiếp. Hầu hết các thuật toán học máy đều có cấu trúc được miêu tả ở hình 2.8, với một loạt các tín hiệu đầu vào đi kèm giá trị trọng số, tổng trọng số

được áp dụng một hàm kích hoạt (có thể là ReLU, Signmoid, hoặc Tanh,..) và một tín hiệu đầu ra ở cuối một neuron.



Hình 2.8. Ứng dụng hàm kích hoạt đối với tổng trọng số tại một nút mạng neuron

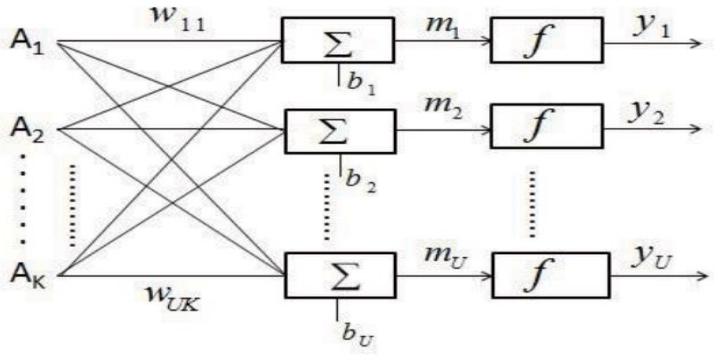
Trong mạng ANN đề xuất với mục đích phân loại lưu lượng mạng Internet, hàm ReLU (Rectified Linear Units) được lựa chọn để làm hàm kích hoạt trong các tầng ẩn (hidden layer). Hàm ReLU được sử dụng rộng rãi trong các mô hình huấn luyện học sâu vì thuật toán chỉ bao gồm phép tính đơn giản trong quá trình tính toán, đồng thời làm giảm và khắc phục khả năng xảy ra hiện tượng các nút neuron bị vô hiệu hóa. Công thức của hàm ReLU được diễn tả qua công thức (2.7):

$$R(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases} \tag{2.7}$$

Ngoài hàm ReLU, một hàm kích hoạt khác thường được sử dụng ở lớp đầu ra dưới dạng các phép tính xác suất là Softmax, nhằm mục tiêu phân loại mẫu dữ liệu đến các lớp lưu lượng mạng. Hàm Softmax sẽ nhận giá trị đầu vào dưới dạng vec-tơ, và tiến hành chuẩn hóa dưới dạng xác suất nhằm dự đoán tỷ lệ một mẫu dữ liệu lần lượt thuộc về các lớp khác nhau, từ đó phân loại mẫu về lớp có xác suất cao nhất. Công thức của hàm Softmax được biểu diễn trong công thức (2.8):

$$\sigma(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, i = 1, \dots, K \; and \; \boldsymbol{z} = (z_1, \dots, z_K) \tag{2.8}$$

Nguyên tắc hoạt động: Hoạt động của mạng neuron là một hệ thống phức tạp có tính phi tuyến tính, mô phỏng lại hoạt động của hệ thống thần kinh con người. Trên thực tế, mạng máy tính nhân tạo có cấu trúc bao gồm sự liên kết của những giá trị dữ liệu đầu vào, đầu ra, và một số lượng lớn các tế bào neuron thần kinh đan xen. Mặc dù một mạng máy tính có thể được điều chỉnh để thích nghi với những tập dữ liệu đầu vào khác nhau, hoặc phụ thuộc vào mục đích của bài toán ứng dụng, cấu trúc của chúng thường bao gồm lớp dữ liệu đầu vào, lớp dữ liệu ẩn và lớp dữ liệu đầu ra. Hình 2.9 khái quát hóa mối quan hệ của dữ liệu đầu vào, trọng số và dữ liệu đầu ra trong một lớp dữ liệu ẩn.



Hình 2.9. Cấu trúc của một lớp dữ liệu ẩn [23]

Trong hình 2.9, $\boldsymbol{A_i}$ $(1 \leq i \leq K)$ đại biểu các giá trị dữ liệu đầu vào, $\boldsymbol{w_{ij}}$ $(1 \leq i \leq K; 1 \leq j \leq U)$ là những giá trị trọng số, $\boldsymbol{b_j}$ $(1 \leq j \leq U)$ là giá trị độ lệch tương ứng với các nút mạng neuron, và $\boldsymbol{y_j}$ $(1 \leq j \leq U)$ là các giá trị đầu ra lớp dữ liệu ẩn này. Tổng trọng số của các giá trị đầu vào tại một neuron có thể được miêu tả theo công thức sau:

$$m_j = \sum_{i=1}^{K} w_{ij} A_i + b_j \tag{2.9}$$

Như vậy từ công thức (2.7), với mỗi hàm kích hoạt $\boldsymbol{f}$ tùy theo nhu cầu của thuật toán, giá trị dữ liệu đầu ra của một nút mạng neuron được liên hệ trong công thức (2.10):

$$y_j = f(m_j) \tag{2.10}$$

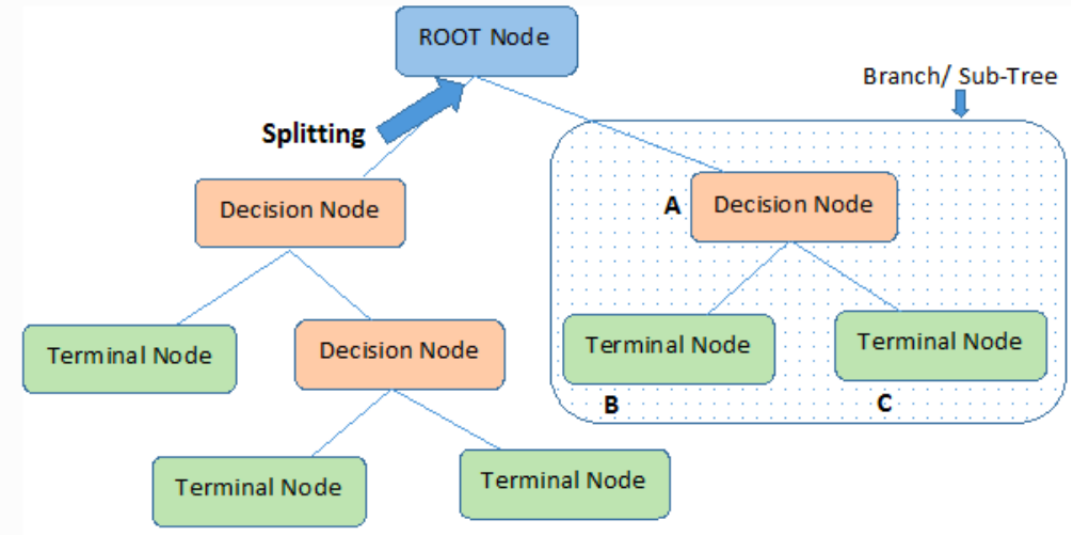
2.3.3 Rừng ngẫu nhiên (RF – Random Forest)

Trong lĩnh vực học máy, mô hình Rừng ngẫu nhiên là một loại mô hình phi tham số có thể được sử dụng cho cả bài toán hồi quy (regression) và bài toán phân loại (classification). Chúng là một trong những phương pháp học máy tập hợp (ensemble learning) phổ biến nhất, thuộc nhóm phương pháp đóng gói (Bagging method).

Phương pháp tập hợp liên quan đến việc sử dụng nhiều thuật toán học (leaners) để nâng cao hiệu suất của bất kỳ mô hình nào trong số đó một cách riêng lẻ. Các phương pháp này có thể được mô tả như là kỹ thuật sử dụng một nhóm những mô hình học yếu với nhau (những mô hình học mà trong đó, chúng chỉ đạt được kết quả phân loại tốt hơn giá trị trung bình một chút so với việc sử dụng một mô hình học ngẫu nhiên), để tạo ra một mô hình tổng hợp, mạnh mẽ hơn. Trong trường hợp này, mô hình Rừng ngẫu nhiên là sự kết hợp của rất nhiều mô hình học Cây quyết định (Decision Trees) để áp dụng vào các bài toán phân loại hoặc hồi quy.

Cây quyết định: Trong lĩnh vực học máy, tương tự như Rừng ngẫu nhiên, Cây quyết định là một loại mô hình học phi tham số, có thể được sử dụng cho cả bài toàn phân loại và hồi quy. Điều này có nghĩa là Cây quyết định là mô hình học linh hoạt không làm tăng số lượng tham số của chúng khi mô hình thêm nhiều đặc trưng hơn (nếu mô hình được xây dựng một cách chính xác) và từ đó đưa ra dự đoán phân loại.

Mục tiêu của việc sử dụng Cây quyết định là tạo ra một mô hình huấn luyện nhằm mục đích dự đoán nhãn, lớp hoặc giá trị của mẫu dữ liệu bằng cách học các quy tắc quyết định đơn giản được suy ra từ các tập dữ liệu trước đó (dữ liệu huấn luyện). Trong mô hình Cây quyết định, để dự đoán nhãn hoặc lớp cho một mẫu dữ liệu, mô hình bắt đầu từ gốc (root) của cây. Thuật toán huấn luyện so sánh các giá trị của đặc trưng tại gốc của mô hình với các đặc trưng của bản ghi. Dựa trên cơ sở của phép so sánh, mô hình phân nhánh cho mẫu tương ứng với giá trị đó và nhảy đến nút (node) tiếp theo. Hình 2.10 miêu tả quá trình phân nhánh dựa trên đặc trưng của một điểm dữ liệu tại các nút khác nhau.



Hình 2.10. Phân nhánh cho 1 điểm dữ liệu
tại các nút của mô hình Cây quyết định

Dựa trên hình 2.10, những điểm cần lưu ý trong mô hình Cây quyết định bao gồm:

Nút gốc (Root node): đại diện cho toàn bộ tập hoặc mẫu dữ liệu (ví dụ như tập huấn luyện), từ đó tiếp tục được chia thành hai hoặc nhiều tập hợp đồng nhất.

Nút quyết định (Decision node): Khi một nút con tách thành các nút con thấp hơn, thì nó được gọi là nút quyết định.

Nút lá (Leaf/Terminal Node): Các nút không phân chia được gọi là nút Lá.

Phân nhánh (Splitting): quá trình phân chia một điểm nút thành hai hoặc nhiều nút con.

Cắt tỉa (Pruning): Khi mô hình loại bỏ các nút con của một nút quyết định, quá trình này đối ngược với quá trình phân nhánh.

Nhánh/Cây phụ (Branch/Sub-Tree): Một tập phụ thuộc của toàn bộ cây được gọi là nhánh hoặc cây con.

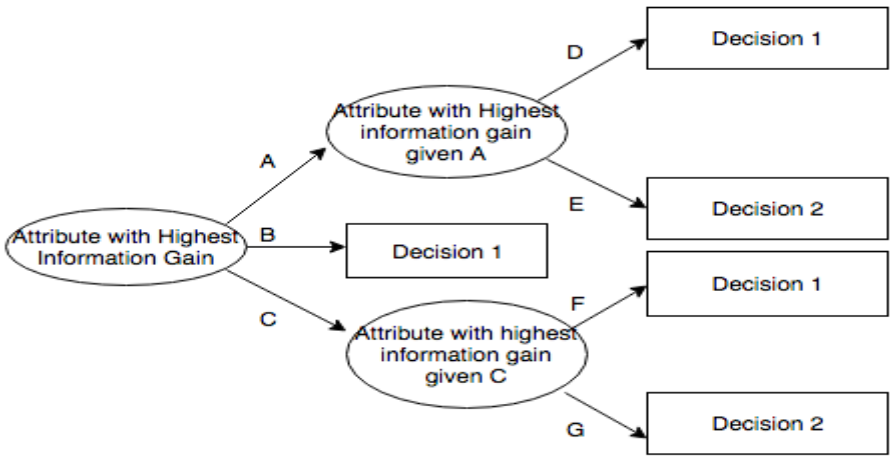
Cây quyết định phân loại các mẫu dữ liệu bằng cách sắp xếp chúng theo chiều từ gốc đến các số nút lá, trong đó nút lá là điểm phân loại nhãn của mẫu dữ liệu. Mỗi nút trong mô hình hoạt động như một điểm thử nghiệm cho các đặc trưng của mẫu, và cứ mỗi một nút quyết định phân nhánh xuống các nút con sẽ tương ứng với quá trình phân loại có thể có cho điểm thử nghiệm của mẫu dữ liệu đó. Tùy thuộc vào độ

phức tạp của tập dữ liệu huấn luyện, số lượng các đặc trưng của mẫu dữ liệu, và tài nguyên tính toán khả dụng, độ sâu của mô hình cây quyết định cũng đóng một vai trò quan trọng trong quá trình thiết lập thuật toán.

Nguyên tắc hoạt động của Cây quyết định: Thuật toán quyết định trong chiến lược phân chia của mô hình ảnh hưởng rất nhiều đến độ chính xác của Cây quyết định. Các thuật toán quyết định khác nhau được áp dụng tùy thuộc theo mô hình cây phân loại và cây hồi quy.

Mô hình cây quyết định sử dụng nhiều thuật toán khác nhau để quyết định phân nhánh một nút thành hai hoặc nhiều nút con. Việc tạo ra các nút con làm tăng tính đồng nhất của các nút con ở các nhánh thấp hơn. Nói cách khác, chúng ta có thể nói rằng độ tinh khiết của nút tăng lên tương ứng với mẫu dữ liệu. Cây quyết định phân chia các nút trên tất cả các mẫu dữ liệu có sẵn trong tập huấn luyện, từ đó lựa chọn nhánh có sự phân tách dẫn đến các nút con có sự đồng nhất cao nhất.

Một trong những mô hình cây quyết định tiêu biểu là ID3. Thuật toán ID3 xây dựng cây quyết định bằng cách sử dụng cách tiếp cận tìm kiếm tham lam (greedy search) từ gốc xuống các nút con thông qua các nhánh cây mà không áp dụng thuật toán quay lui (backtracking). Thuật toán tham lam luôn đưa ra lựa chọn có vẻ là tốt nhất ở nút con tại thời điểm đó. Hình 2.11 thể hiện sơ đồ khối các bước thực hiện của thuật toán ID3.

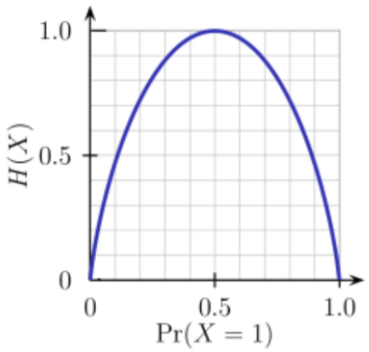


Hình 2.11. Quá trình đưa ra quyết định của mô hình ID3

- Thuật toán bắt đầu với tập dữ liệu huấn luyện ***S*** tại nút gốc.
- Trong mỗi vòng lặp của thuật toán, mô hình sẽ kiểm tra và chọn ra các đặc trưng chưa được sử dụng trong tập ***S*** và tính toán giá trị Entropy **H** và Độ lợi thông tin **IG** của đặc trưng đó.
- Mô hình lựa chọn đặc trưng có giá trị Entropy nhỏ nhất hoặc có độ lợi thông tin lớn nhất. Từ đó, phân chia tập ***S*** thành các tập dữ liệu con dựa trên đặc trưng đó.
- Thuật toán tiếp tục phân loại tập dữ liệu thành các tập con nhỏ hơn dựa trên các đặc trưng chưa được lựa chọn.

Nếu tập dữ liệu huấn luyện bao gồm **N** đặc trưng thì việc quyết định thuộc tính nào để đặt ở gốc hoặc ở các cấp khác nhau của mô hình cây tại các nút phụ thuộc là một quy trình phức tạp. Do đó, mô hình giải quyết vấn đề lựa chọn đặc trưng dựa trên các giá trị tiêu chí khác nhau như Entropy, độ lợi thông tin, Gini index,…Mỗi đặc trưng đều sẽ được tính toán cho từng tiêu chí này, từ đó sắp xếp đặt vị trí của các đặc trưng trong cây dựa trên thứ tự kết quả tính toán. Ngoài ra, nếu không đặt ra giới hạn cho thuật toán cây quyết định, độ sâu của cây quyết định trong quá trình phân nhánh giữa các đặc trưng sẽ tiếp tục tiến hành cho đến khi đạt được nhánh có kết quả phân loại thống nhất giữa các đặc trưng. Điều này sẽ tạo ra áp lực rất lớn đến tài nguyên tính toán cũng như độ phức tạp của thuật toán. Vì vậy, độ sâu của cây quyết định cũng cần được cân nhắc trong quá trình thiết kế mô hình.

Entropy: Entropy là thông số đại diện cho tính ngẫu nhiên của đặc trưng đang được phân tích và xử lý. Giá trị của Entropy càng cao thì càng khó rút ra kết luận từ đặc trưng đó và ngược lại, giá trị Entropy thấp mang lại tính khẳng định chắc chắn về quyết định phân nhóm dữ liệu của đặc trưng. Hình 2.12 minh họa giá trị Entropy theo tỷ lệ xác suất của đặc trưng phân loại dữ liệu.



Hình 2.12. Giá trị Entropy

Từ hình 2.12 trên, entropy **H**(X) bằng 0 khi xác suất có giá trị là 0 hoặc 1. Entropy chạm mốc cực đại khi xác suất là 0,5, vì giá trị này phản ánh tính ngẫu nhiên trong tập dữ liệu và rất khó để phân chia tập dữ liệu dựa trên đặc trưng này. Từ đó, mô hình ID3 tuân theo quy tắc nhánh cây có entropy bằng 0 là nút lá, và nhánh cây có entropy lớn hơn 0 cần được phân nhánh thêm. Công thức Entropy cho một và nhiều đặc trưng khác nhau được miêu tả như sau:

$$E(S) = \sum_{i=1}^{c} -p_i log_2(p_i), \tag{3.11}$$

$$\boldsymbol{E}(S, X) = \sum_{c \in X} P(c)E(c), \tag{3.12}$$

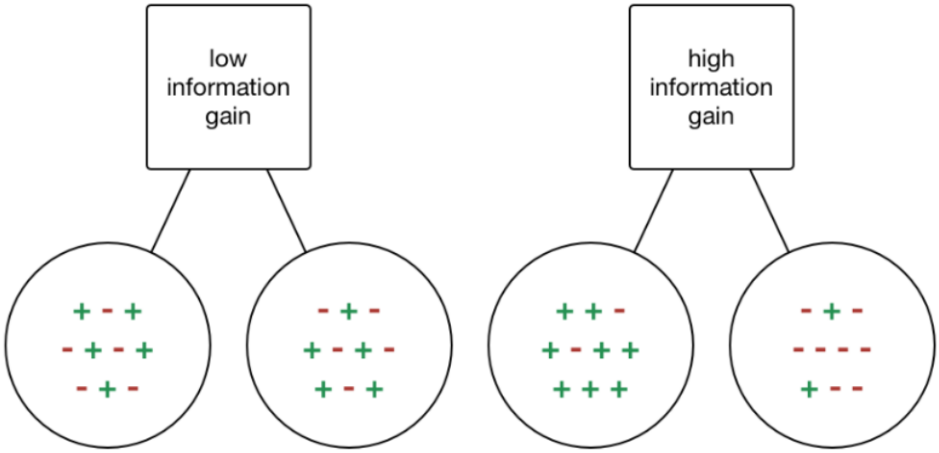
trong đó, S là trạng thái hiện tại của đặc trưng, X là đặc trưng được chọn, E là đại giá trị entropy của đặc trưng đó, và c là tổng tất cả các lựa chọn con có trong đặc trưng X.

Độ lợi thông tin: một đại lượng thống kê nhằm đo lường khả năng phân tách tập dữ liệu của một đặc trưng nhất định tương ứng với các nhãn phân loại có trong tập dữ liệu đó. Quá trình xây dựng mô hình cây quyết định chủ yếu xoay quanh quá trình lựa chọn các đặc trưng mang độ lợi thông tin lớn nhất và entropy nhỏ nhất. Về cơ bản, giá trị độ lợi thông tin là sự khác biệt về giá trị Entropy trước và sau khi tập dữ liệu huấn luyện được phân chia dựa trên đặc trưng đang phân tích. Về mặt toán học, công thức tính độ lợi thông tin có thể được khái quát hóa như sau:

$$\mathbf{IG}(S) = E(before) - \sum_{j=1}^{k} E(j, after), \tag{2.13}$$

trong đó, $E(before)$ và $E(after)$ là giá trị Entropy của tập dữ liệu trước và sau khi phân chia thành các tập dữ liệu con dựa trên đặc trưng hiện tại, k là số lượng tập dữ liệu con tạo ra được sau quá trình phân chia.

Hình 2.13 miêu tả khả năng phân chia tập dữ liệu với 2 giá trị độ lợi thông tin khác nhau.



Hình 2.13: Phân chia tập dữ liệu dựa trên giá trị độ lợi thông tin

Chỉ số Gini (Gini index): Chỉ số Gini là một hàm giá trị được sử dụng để đánh giá sự phân chia trong tập dữ liệu. Giá trị này được tính bằng cách lấy một trừ đi tổng bình phương các xác suất của mỗi lớp, được miêu tả trong công thức (2.14). Ngược lại với giá trị độ lợi thông tin mà tại đó quá trình phân chia tập dữ liệu thường được phân thành các nhóm dữ liệu nhỏ mang tính phân biệt cao, chỉ số Gini thường dùng để phân chia các tập dữ liệu thành các nhóm lớn và dễ được ứng dụng trong các mô hình phân loại hơn.

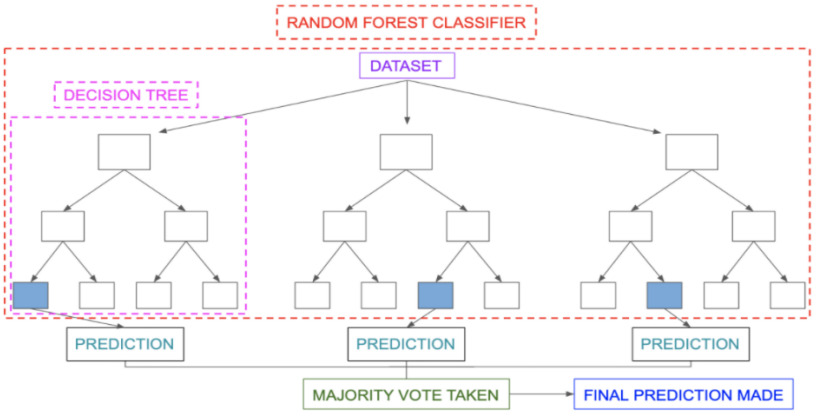
$$\mathbf{Gini} = 1 - \sum_{i=1}^{c} (p_i)^2, \tag{2.14}$$

Dựa trên công thức (3.14), giá trị chỉ số Gini cao đồng nghĩa với tính không đồng nhất giữa các mẫu dữ liệu cao hơn. Từ đó, tiến hành phân loại tập dữ liệu thành 2 nhóm dựa trên phép phân loại nhị phân.

Nguyên tắc hoạt động của Rừng ngẫu nhiên: Sự khác biệt chính giữa mô hình huấn luyện cây quyết định và mô hình rừng ngẫu nhiên là việc thiết lập các nút gốc và các nút quyết định được thực hiện ngẫu nhiên trong những thuật toán sau. Mô hình rừng ngẫu nhiên sử dụng phương pháp đóng gói (bagging) để tạo ra các phép dự đoán cần thiết.

Quá trình thực hiện phương pháp đóng gói bao gồm việc sử dụng các tập dữ liệu khác nhau (dữ liệu huấn luyện) thay vì chỉ áp dụng một tập huấn luyện duy nhất. Tập dữ liệu huấn luyện bao gồm các mẫu dữ liệu và các đặc trưng tương ứng được sử dụng để đưa ra dự đoán. Các mô hình cây quyết định khác nhau tạo ra các kết quả phân loại nhãn đầu ra khác nhau, tùy thuộc vào dữ liệu huấn luyện được cung cấp tại đầu vào của thuật toán rừng ngẫu nhiên. Hình 2.14 miêu tả đơn giản quá trình đưa ra kết quả phân loại dựa trên ứng dụng mô hình huấn luyện rừng ngẫu nhiên.

Các kết quả đầu ra này sẽ được xếp hạng, và kết quả cao nhất sẽ được chọn làm kết quả phân loại đầu ra cuối cùng. Việc lựa chọn kết quả dự đoán cuối cùng tuân theo nguyên tắc đa số. Do đó, kết quả phân loại được lựa chọn bởi số lượng lớn các cây quyết định sẽ trở thành kết quả đầu ra cuối cùng của mô hình rừng ngẫu nhiên. Nói cách khác, việc tăng số lượng cây quyết định trong việc thiết kế mô hình rừng ngẫu nhiên gia tăng độ chính xác trong quá trình phân loại kết quả dữ liệu đầu ra. Ngoài ra, mô hình rừng ngẫu nhiên cũng đạt hiệu quả cao hơn trong việc đưa ra dự đoán mà không cần quá trình tinh chỉnh siêu tham số. Kết quả phân loại cũng giải quyết được hiện tượng quá khớp hay gặp trong mô hình cây huấn luyện.



Hình 3.14. Sơ đồ khối quá trình phân loại nhãn của mô hình Rừng ngẫu nhiên

Chương 3: PHÁT TRIỂN MÔ HÌNH

3.1. Tập dữ liệu

Trong quá trình áp dụng các mô hình học máy, một trong những yếu tố quan trọng nhất là lựa chọn tập dữ liệu (dataset) phù hợp theo đúng mục đích đề tài. Việc phân tích và xử lý cơ sở dữ liệu cho phép làm rõ những tính chất đặc thù của từng dữ liệu, từ đó đưa ra những phương án, kỹ thuật xử lý cũng như những mô hình học máy phù hợp nhằm nâng cao khả năng phân loại, đáp ứng yêu cầu của nhà cung cấp mạng.

Trong nghiên cứu này, tập dữ liệu VPN-nonVPN (ISCXVPN2016) sẽ được sử dụng cho quá trình huấn luyện và quá trình kiểm tra. Tập dữ liệu VPN-nonVPN (ISCXVPN2016) được đề xuất bởi [24]. Dữ liệu dược thu thập từ Đại học New Brunswick ở Canada và được tạo ra bằng việc tạo 2 tài khoản người dùng để sử dụng các dịch vụ như Skype, Facebook, v.v. Bảng 3.1 thể hiện các loại lưu lượng và ứng dụng khác nhau trong tập dữ liệu.

Bảng 3.1: Tóm tắt của tập dữ liệu [24]

Loại lưu lượng	**Chi tiết**
Web Browsing	Firefox và Ch
Email	SMPTS, POP3S và IMAPS
Chat	ICQ, AIM, Skype, Facebook và Hangouts
Streaming	Vimeo và Youtube
File Transfer	Skype, FTPS và SFTP
VoIP	Facebook, Skype và Hangouts voice calls (trong 1 giờ)
P2p	uTorrent và Bittorrent

Ngoài ra, trong tập dữ liệu này, các loại lưu lượng đã thu tập còn được chia thành 2 loại: được mã hoá bởi VPN và không được mã hoá bởi VPN. Như vậy, với mỗi trường hợp trên, ta sẽ có 7 loại lưu lượng được thu thập. Vì vậy, tập dữ liệu sẽ được chia thành 2 bối cảnh để tiến hành thử nghiệm. Những đặc trưng sử dụng trong tập dữ liệu sẽ được miêu tả cụ thể trong mục 3.2.3.

3.2 Mô hình phân loại lưu lượng

3.2.1 Xây dựng mô hình

Dựa trên mô hình trong [17], mô hình phân loại lưu lượng Internet có thể được khái quát hóa như sau:

Nhập tập dữ liệu gốc ISCXVPN2016: Trong đề tài này, 7 loại lưu lượng Internet được miêu tả và gắn nhãn tương ứng nhằm phục vụ mục tiêu phân loại lưu lượng Internet.

Tiền xử lý tập dữ liệu: Tại bước này, các giá trị trong mẫu dữ liệu được chuẩn hóa, đồng thời nhãn dán về lớp lưu lượng mạng cũng được mã hóa nhằm chuẩn bị cho các bước huấn luyện và phân loại kế tiếp trong các mô hình huấn luyện.

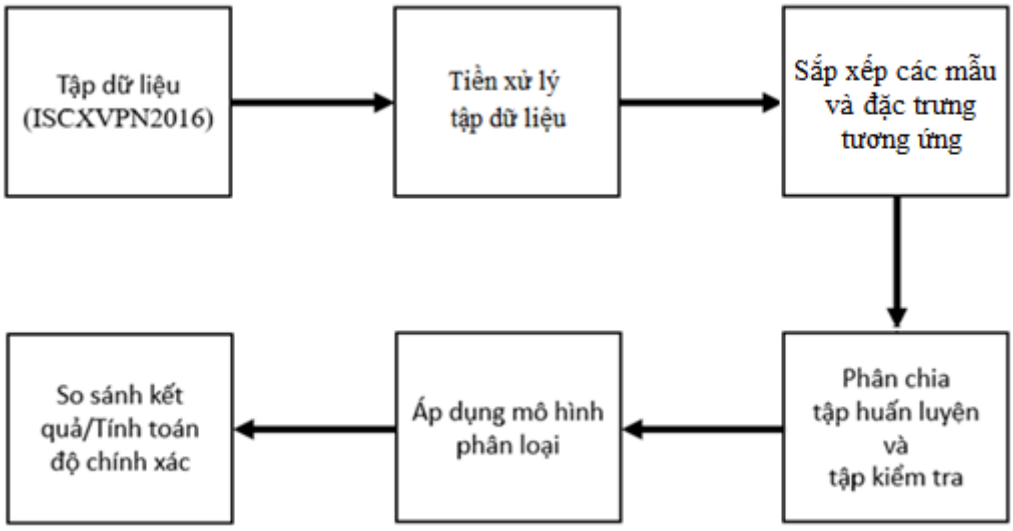
Sắp xếp mẫu dữ liệu và đặc trưng tương ứng từ tập dữ liệu: Trong đề tài này, những đặc trưng cụ thể trong mỗi mẫu dữ liệu chứa các thông tin khác nhau được sử dụng nhằm phân loại lưu lượng Internet. Tổng hợp toàn bộ các đặc trưng có trong tập dữ liệu, bao gồm tên và định nghĩa của chúng được mô tả trong mục 3.2.3.

Phân chia tập huấn luyện và tập kiểm tra: Từ tập dữ liệu ISCXVPN2016 trên, tập huấn luyện và tập kiểm tra được phân chia một cách ngẫu nhiên làm dữ liệu đầu vào nhằm đảm bảo kết quả dự đoán của mô hình phân loại mang đầy đủ tính khách quan. Để dự đoán nhãn cho các lớp lưu lượng mạng tại đầu ra, bộ dữ liệu huấn luyện được cung cấp làm dữ liệu đầu vào cho mô hình huấn luyện. Tỷ lệ phân chia cụ thể giữa tập huấn luyện và tập kiểm tra và phương án phân chia sẽ được miêu tả thêm trong mục 3.2.4.

Áp dụng mô hình phân loại: áp dụng những mô hình huấn luyện bao gồm KNN, ANN, và RF để dự đoán các nhãn trong tập kiểm tra. Nguyên tắc hoạt động, công thức và những bước áp dụng cụ thể của những mô hình huấn luyện cho quy trình phân loại được miêu tả cụ thể hơn trong những mục kế tiếp.

So sánh kết quả/Tính toán độ chính xác: bằng những thông số như Accuracy, Precision, Recall, F1 score…

Sơ đồ khối tổng quan của mô hình phân loại lưu lượng Internet được biểu diễn bằng hình 3.1.



Hình 3.1. Sơ đồ khối mô hình phân loại lưu lượng Internet

3.2.2 Tiền xử lý dữ liệu

Chuẩn hóa Tối đa – Tối thiểu: Dựa trên bản chất của tập dữ liệu vốn chứa rất nhiều các đặc trưng khác nhau với tổng cộng 23 đặc trưng cụ thể, mỗi đặc trưng lại chứa các giá trị trong phạm vi khác nhau, đề tài quyết định chuẩn hóa các trường dữ liệu khác nhau về phạm vi giá trị chuẩn $[0,1]$. Trong ứng dụng cụ thể cho quá trình tiền xử lý dữ liệu, biến *scaler* được dùng để thiết lập chuẩn hóa tối đa – tối thiểu *MinMaxScaler*. Biến *scaler* sau đó được dùng để áp dụng chung quá trình chuẩn hóa cho cả tập huấn luyện và tập kiểm tra.

```
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Quá trình chuẩn hóa tối đa – tối thiểu trong tập dữ liệu này được thực hiện nhằm làm bước chẩn bị cho quá trình huấn luyện cho các mô hình học máy ở các bước sau, đồng thời làm giảm độ phức tạp trong quá trình tính toán.

Mã hóa nhãn: Tương tự, nhằm hướng đến mục tiêu giảm tải khối lượng tính toán và phân loại các nhãn nhóm mạng Internet các nhãn phân loại được mã hóa dựa trên biến *encoder* trong quá trình áp dụng tiền xử lý dữ liệu. Biến *encoder* sẽ mã hóa

nhãn dán các lớp lưu lượng mạng, hỗ trợ quá trình xử lý dữ liệu tại bước phân loại mẫu dữ liệu được nhanh chóng hơn, giảm tải thời gian tính toán lên mô hình.

```
# encode the class name as int 64 for training
encoder = LabelEncoder()
encoder.fit(Y)
Y = encoder.transform(Y)
```

Kết quả mã hóa tương ứng cho từng lớp lưu lượng mạng được miêu tả trong bảng 3.2.

Bảng 3.2. Mã hóa nhãn các lớp lưu lượng mạng Internet

Lưu lượng Internet	**Giá trị mã hóa tương ứng**
VoIP	0
Web Browsing	1
File transfer	2
P2P	3
Chat	4
Streaming	5
Email	6

3.2.3 Mẫu dữ liệu và đặc trưng tương ứng

Như đã đề cập sơ bộ tại mục 3.1, tập dữ liệu ISCXVPN2016 bao gồm các mẫu dữ liệu ghi nhận từ 7 lớp lưu lượng mạng Internet khác nhau. Mỗi một mẫu dữ liệu được miêu tả 8 loại đặc trưng chính, bao gồm *fiat, biat, flowiat, active, idle, fb_psec* và *fp_psec*. Trong đó các đặc trưng fiat, biat, flowiat, active và idle được miêu tả cụ thể hơn bằng 4 thông số như giá trị trung bình (mean), giá trị nhỏ nhất (min), giá trị lớn nhất (max) và độ lệch chuẩn (standard deviation). Tổng hợp 23 đặc trưng của tập dữ liệu được tóm tắt trong bảng 3.3.

Bảng 3.3. Tổng hợp nhóm 23 đặc trưng của tập dữ liệu ISCXVPN2016

Đặc trưng	***Miêu tả***
duration	khoảng thời gian lưu lượng
fiat_mean	Giá trị trung bình thời gian giữa hai gói được gửi theo hướng đi
fiat_max	Giá trị lớn nhất thời gian giữa hai gói được gửi theo hướng đi
fiat_min	Giá trị nhỏ nhất thời gian giữa hai gói được gửi theo hướng đi
fiat_std	Giá trị độ lệch chuẩn thời gian giữa hai gói được gửi theo hướng đi
biat_mean	Giá trị trung bình thời gian giữa hai gói được gửi theo hướng ngược về
biat_max	Giá trị lớn nhất thời gian giữa hai gói được gửi theo hướng ngược về
biat_min	Giá trị nhỏ nhất thời gian giữa hai gói được gửi theo hướng ngược về
biat_std	Giá trị độ lệch chuẩn thời gian giữa hai gói được gửi theo hướng ngược về
flowiat_mean	Giá trị trung bình thời gian giữa hai gói được gửi theo một trong hai hướng
flowiat_max	Giá trị lớn nhất thời gian giữa hai gói được gửi theo một trong hai hướng
flowiat_min	Giá trị nhỏ nhất thời gian giữa hai gói được gửi theo một trong hai hướng
flowiat_std	Giá trị độ lệch chuẩn thời gian giữa hai gói được gửi theo một trong hai hướng
active_mean	Giá trị trung bình khoảng thời gian lưu lượng hoạt động trước khi ngừng hoạt động
active_max	Giá trị lớn nhất khoảng thời gian lưu lượng hoạt động trước khi ngừng hoạt động
active_min	Giá trị nhỏ nhất khoảng thời gian lưu lượng hoạt động trước khi ngừng hoạt động
active_std	Giá trị độ lệch chuẩn khoảng thời gian lưu lượng hoạt động trước khi ngừng hoạt động
idle_mean	Giá trị trung bình khoảng thời gian luồng không hoạt động trước khi bắt đầu hoạt động lại
idle_max	Giá trị lớn nhất khoảng thời gian luồng không hoạt động trước khi bắt đầu hoạt động lại
idle_min	Giá trị nhỏ nhất khoảng thời gian luồng không hoạt động trước khi bắt đầu hoạt động lại
idle_std	Giá trị độ lệch chuẩn khoảng thời gian luồng không hoạt động trước khi bắt đầu hoạt động lại
fb_psec	lưu lượng byte trên một giây
fp_psec	lưu lượng gói tin trên một giây

Mỗi mẫu dữ liệu trong tập dữ liệu ISCXVPN2016 đều sở hữu 23 loại đặc trưng miêu tả như trên. Tương tự, mỗi một lớp lưu lượng mạng bao gồm nhiều mẫu dữ liệu khác nhau và chúng được sử dụng làm dữ liệu đầu vào cho các mô hình huấn luyện, từ đó các mô hình sẽ phân tích các đặc trưng có trong mẫu dữ liệu làm căn cứ phân loại các lớp lưu lượng mạng khác nhau.

3.2.4 Dữ liệu đầu vào – phân chia tập huấn luyện và kiểm tra

Để đánh giá hiệu suất của mô hình học máy, thuật toán đánh giá cần sử dụng các mẫu dữ liệu không tham gia vào trong quá trình huấn luyện. Nếu không, việc đánh giá mô hình sẽ không mang tính khách quan và dễ dẫn đến sai sót trong quá trình đánh giá. Phương pháp đơn giản nhất là chia toàn bộ tập dữ liệu thành hai tập dữ liệu bao gồm tập huấn luyện và tập kiểm tra. Sau đó, sử dụng một tập dữ liệu để huấn luyện mô hình học máy, và một để đánh giá khả năng phân loại của mô hình được chọn. Đây được gọi là phương pháp giữ lại (hold-out method). Hình 3.2 minh họa các mẫu dữ liệu đại diện có sẵn trong tập dữ liệu sau khi hoàn thành quá trình sắp xếp các mẫu dữ liệu với 23 đặc trưng được miêu tả như trên, tương ứng với từng lớp lưu lượng mạng được mã hóa trong khâu tiền xử lý dữ liệu.

	duration	min_fiat	min_biat	max_fiat	max_biat	mean_fiat	mean_biat	...	std_flowiat	min_active	mean_active	max_active	std_active	min_idle	mean_idle	max_idle	std_idle	class1
0	14993462.0	0.0	0.0	823486.0	854818.0	873.134288	287.914517	...	7050.781273	-1.0	0.0	-1.0	0.000000	-1.0	0.0	-1.0	0.000000	b'BROWSING'
1	14463281.0	0.0	0.0	742368.0	742339.0	1321.330258	312.290951	...	6982.036846	-1.0	0.0	-1.0	0.000000	-1.0	0.0	-1.0	0.000000	b'BROWSING'
2	14997099.0	1.0	0.0	537201.0	565232.0	1850.116210	344.855753	...	5560.260369	-1.0	0.0	-1.0	0.000000	-1.0	0.0	-1.0	0.000000	b'BROWSING'
3	14999980.0	2.0	0.0	954084.0	954052.0	1796.827863	382.837498	...	9375.105249	-1.0	0.0	-1.0	0.000000	-1.0	0.0	-1.0	0.000000	b'BROWSING'
4	14989090.0	2.0	0.0	1014690.0	1016593.0	1668.792029	394.588556	...	9205.461338	9578088.0	9578088.0	9578088.0	0.000000	1014624.0	1014624.0	1014624.0	0.000000	b'BROWSING'
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8960	14997258.0	69.0	203.0	21131.0	28475.0	19659.708661	19785.300792	...	4198.398294	-1.0	0.0	-1.0	0.000000	-1.0	0.0	-1.0	0.000000	b'VOIP'
8961	400609.0	400609.0	200451.0	400609.0	200451.0	400609.000000	200451.000000	...	115497.999657	-1.0	0.0	-1.0	0.000000	-1.0	0.0	-1.0	0.000000	b'VOIP'
8962	11257986.0	66.0	168.0	225830.0	63675.0	19960.968085	19753.841355	...	6399.053203	-1.0	0.0	-1.0	0.000000	-1.0	0.0	-1.0	0.000000	b'VOIP'
8963	4009751.0	154.0	331.0	2005165.0	2004379.0	801903.800000	801930.800000	...	810535.993475	2003874.0	2004518.0	2005162.0	910.752985	2003446.0	2003912.5	2004379.0	659.729869	b'VOIP'
8964	8272911.0	62.0	1.0	1030079.0	1029669.0	459606.166667	288368.000000	...	360925.668879	1002717.0	1213655.6	2003627.0	441785.036352	1001616.0	1012422.4	1029669.0	14330.336844	b'VOIP'

8965 rows × 24 columns

Hình 3.2. Một số giá trị đại diện từ những đặc trưng của mẫu trong tập dữ liệu

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2)
```

Trong quá trình áp dụng mô hình học máy, đề tài sử dụng tỷ lệ 8:2 nhằm phân chia tập dữ liệu huấn luyện và tập kiểm tra. Trong đó, *X_train* và *X_test* là tập hợp mẫu dữ liệu lần lượt có trong tập huấn luyện và tập kiểm tra. Kế tiếp, *y_train* và *y_test* là các nhãn dán tương ứng cho các mẫu dữ liệu trong 2 tập trên, trong đó *y_train* sẽ

được sử dụng để huấn luyện mô hình, và sau đó *y_test* sẽ được dùng để so sánh kết quả phân loại, từ đó đánh giá hiệu suất phân loại của mô hình.

Tỷ lệ này được áp dụng sau khi tham khảo tỷ lệ phân chia tập dữ liệu từ các mô hình học máy khác nhau trong quá trình khảo sát trong mục 1.3. Ngoài ra, tỷ lệ này đảm bảo độ chênh lệch giữa tập huấn luyện và tập kiểm tra không quá lớn, đảm bảo tính khách quan trong quá trình phân loại và đánh giá mô hình huấn luyện.

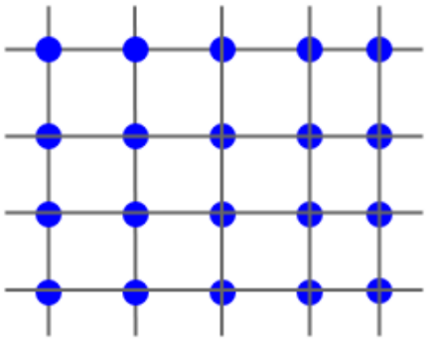
3.2.5 Điều chỉnh siêu tham số (Hyperparamters Tuning)

Trong quá trình xây dựng những mô hình học máy được miêu tả trên các mục 2.3.1, 2.3.2 và 2.3.3, việc xác định giá trị những tham số và siêu tham số đóng một vai trò hết sức quan trọng cho các mô hình tương ứng. Ví dụ như việc xác định lựa chọn phương pháp cũng như trọng số trong chỉ số khoảng cách thuộc mô hình KNN, hoặc những chỉ số như độ lợi thông tin và Entropy của Rừng ngẫu nhiên, v.v... Ngoài việc thúc đẩy quá trình huấn luyện của các mô hình được diễn ra nhanh chóng, việc xác định những giá trị tham số và siêu tham số phù hợp với tập dữ liệu tương ứng với các mô hình khác nhau cũng gia tăng hiệu suất phân loại. Trong đó, tham số là một thành phần của mô hình huấn luyện, và giá trị của chúng được tính toán một cách tự động bằng việc tham chiếu ứng với các mẫu dữ liệu như các vectơ hỗ trợ trong mô hình huấn luyện máy vectơ hỗ trợ.

Ngược lại, siêu tham số là những giá trị có thể được thiết lập trước nhằm cải thiện hiệu suất của mô hình huấn luyện, ví dụ như tốc độ học (learning rate) của một mô hình học sâu. Những giá trị siêu tham số sẽ định hình một mô hình huấn luyện. Để xác định được giá trị của các siêu tham số này, mô hình cần trải qua quá trình điều chỉnh (tuning) vì những giá trị lý tưởng ứng với một tập dữ liệu có thể không phù hợp với các tập dữ liệu khác. Trong các phương pháp điều chỉnh siêu tham số thuộc lĩnh vực học máy, những phương pháp nổi tiếng nhất là tìm kiếm lưới (Grid Search) và tìm kiếm ngẫu nhiên (Randomized Seach)

Tìm kiếm lưới: Phương pháp này khảo sát tất cả các tổ hợp siêu tham số khác nhau xác định trong không gian tìm kiếm. Điều này sẽ tốn một lượng đáng kể tài nguyên tính toán và thường tiêu tốn thời gian thực thi cao khi không gian tìm kiếm

có chiều lớn hơn tương ứng với nhiều siêu tham số, đồng thời chứa nhiều tổ hợp các giá trị siêu tham số khác nhau. Tuy nhiên phương pháp này lại rất lý tưởng khi mô hình chỉ chứa số lượng nhỏ siêu tham số và một số hữu hạn (cố định) các giá trị của chúng. Đồng thời, tìm kiếm lưới đảm bảo cung cấp tổ hợp siêu tham số tốt nhất cho mô hình huấn luyện, nhất là khi quá trình khảo sát các mô hình nổi tiếng trong cùng lĩnh vực được thực hiện tốt và cung cấp các giá trị phù hợp cho các siêu tham số.



Hình 3.3. Không gian tìm kiếm siêu tham số - Tìm kiếm lưới

Trong ví dụ điều chỉnh siêu tham số bằng 1 tập dữ liệu nhỏ từ *sklearn*, các giá trị siêu tham số được khai báo lần lượt trong mảng không gian tìm kiếm *hyperparameter_space*. Tiếp theo đó, các siêu tham số sẽ được đưa vào hàm *GridSearchCV* nhằm tìm kiếm tổ hợp giá trị các siêu tham số phù hợp với dữ liệu huấn luyện. Hình 3.4 báo cáo các giá trị tương ứng với từng siêu tham số và kết quả độ chính xác tương ứng với các giá trị này.

```
Defining 3-dimensional hyperparameter space as a Python dictionary
hyperparameter_space = {'max_depth': [2,3,4,6,8,10,12,15,20],
                        'min_samples_leaf': [1,2,4,6,8,10,20,30],
                        'min_samples_split': [1,2,3,4,5,6,8,10]}

from sklearn.model_selection import GridSearchCV
gs = GridSearchCV(dtclf, param_grid = hyperparameter_space,
                  scoring = "accuracy",
```

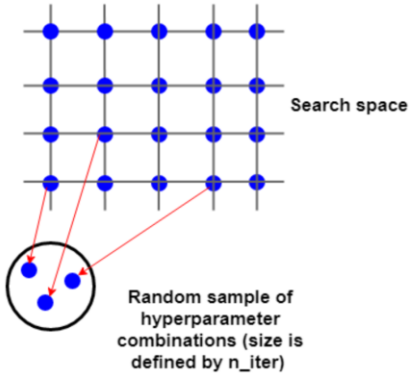
```
        n_jobs = -1, cv = 10, return_train_score = True)
```

```
Optimal hyperparameter combination: {'max_depth': 6, 'min_samples_leaf': 6, 'min_samples_split': 2}

Mean cross-validated training accuracy score: 0.7723544973544973
Test accuracy: 0.87
Execution time of Grid Search (in Seconds): 22.51905846595764
```

Hình 3.4. Kết quả các giá trị siêu tham số khi áp dụng Tìm kiếm lưới

Tìm kiếm ngẫu nhiên: Tìm kiếm ngẫu nhiên có xác suất cao trong việc tìm ra tổ hợp siêu tham số tối ưu trong các kết hợp được chọn ngẫu nhiên với số lần thử nghiệm các tổ hợp được đặt ra trước bằng giá trị *n_iter*. Phương pháp này rất hữu ích để tìm tổ hợp siêu tham số phù hợp một cách nhanh chóng và hiệu quả khi không gian tìm kiếm có số chiều nhiều hơn và chứa nhiều tổ hợp các giá trị khác nhau. Tuy nhiên, trong nhiều trường hợp, tổ hợp giá trị của các siêu tham số tìm có thể không phải là tổ hợp tối ưu nhất, vì nó không qua thử nghiệm hết tất cả các tổ hợp.



Hình 3.5. Không gian tìm kiếm siêu tham số - Tìm kiếm ngẫu nhiên

Tương tự với ví dụ trên, khi áp dụng quá trình tìm kiếm ngẫu nhiên cho các siêu tham số với tập hợp các giá trị trong chiều không gian tìm kiếm giống nhau, hình 3.5 miêu tả kết quả tổ hợp khác với giá trị từ tìm kiếm lưới.

```
# Defining 3-dimensional hyperparameter space as a Python dictionary

hyperparameter_space = {'max_depth': [2,3,4,6,8,10,12,15,20],
```

```
                'min_samples_leaf': [1,2,4,6,8,10,20,30],
                'min_samples_split': [1,2,3,4,5,6,8,10]}

from sklearn.model_selection import RandomizedSearchCV
rs = RandomizedSearchCV(dtclf, param_distributions = hyperparameter_space,
                n_iter =10, scoring = "accuracy", random_state =0,
                n_jobs =-1, cv = 10, return_train_score =True)
```

```
Optimal hyperparameter combination: {'min_samples_split': 8, 'min_samples_leaf': 6, 'max_depth': 12}

Mean cross-validated training accuracy score: 0.7723544973544973
Test accuracy: 0.87
Execution time of Random Search (in Seconds): 0.5163977146148682
```

Hình 3.6. Kết quả các giá trị siêu tham số khi áp dụng Tìm kiếm ngẫu nhiên

Từ những phân tích trên, kết hợp với độ lớn của tập dữ liệu đề xuất ISCXVPN2016 và các siêu tham số quan trọng đề cập trong các mô hình miêu tả trong mục 2.3.1, 2.3.2 và 2.3.3, GridSearchCV sẽ là một lựa chọn phù hợp với giới hạn và mục tiêu của đề tài. Trong đó, mục tiêu của đề tài là đề xuất các mô hình học máy phù hợp nhằm phân loại hiệu quả lưu lượng mạng Internet. Số lượng các siêu tham số được khảo sát trong các mục 2.3 khá ít, tài nguyên thư viện cũng như tính toán đủ khả năng sử dụng hàm GridSearchCV nhằm tìm ra các tổ hợp giá trị siêu tham số phù hợp nhất.

3.2.6 K – Lân cận (KNN – K-Nearest Neighbors)

Như đã nhắc đến trong mục 2.3.1, do đặc thù của quy tắc bình chọn đa số, chỉ số k trong mô hình KNN của đề tài chỉ gồm các giá trị lẻ, tương ứng với $k = 3, 5, 7, 11, 21$.

```
params = {'n_neighbors': [3, 5, 7, 11, 21],
   'weights': ['uniform', 'distance'],
   'metric': ['euclidean', 'manhattan']}
gs = GridSearchCV(KNeighborsClassifier(), params, cv = 10, n_jobs = -1)
```

Trong đó, Khoảng cách Manhattan được thể hiện bằng đoạn code python như sau.

```
def Manhattan_distance(point1, point2):
   return np.sum(np.absolute(point1 - point2))
```

Khoảng cách Euclid đo khoảng cách đường thẳng thực giữa hai điểm trong không gian như sau.

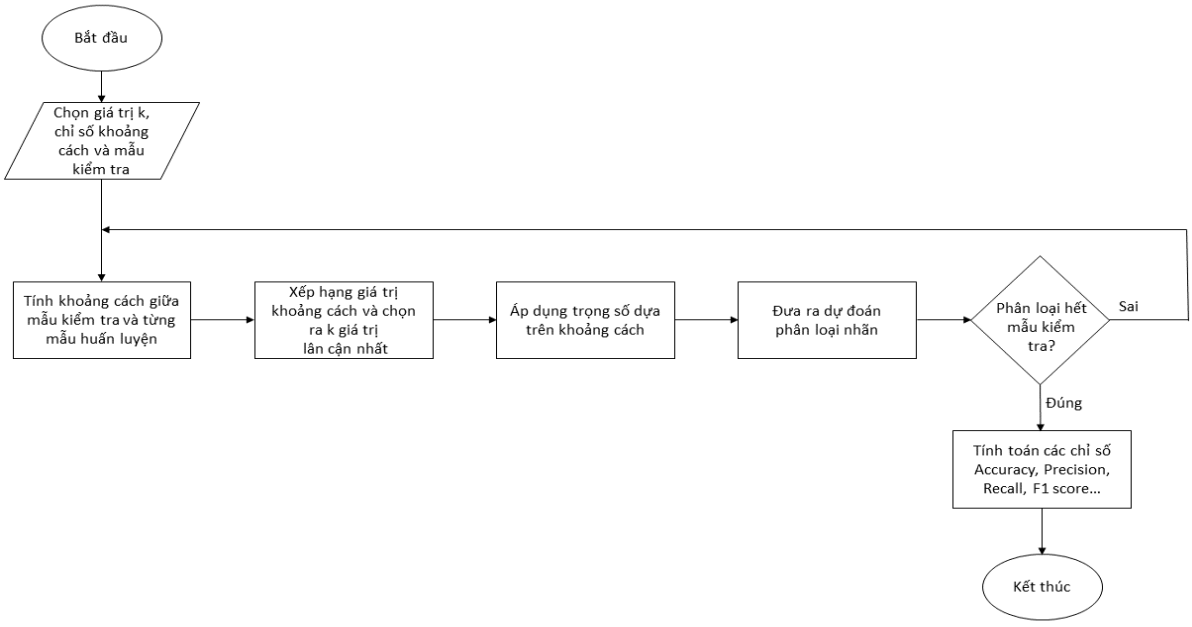
```
def Euclidean_distance(point1, point2):
   return np.sqrt(np.sum((point1 - point2) ** 2))
```

Tương ứng với khái niệm trên, khi thiết kế chỉ số cho mô hình huấn luyện KNN, chỉ số *n_neighbors* được khai báo dưới dạng mảng nhằm huấn luyện và kiểm tra khả năng phân loại của mô hình với các giá trị k khác nhau. Tương tự, những chỉ số khác được sử dụng để phân loại trong mô hình KNN bao gồm *metric* đại diện cho chỉ số khoảng cách và *weights* đại diện cho giá trị trọng số. Dựa theo kết quả mô phỏng từ công thức số (2.6), đề tài đề xuất sử mô hình KNN dưới ảnh hưởng của bộ phân loại trọng số điểm lân cận (weighted nearest neighbour classifier) nhằm tăng hiệu suất phân loại của tập dữ liệu.

```
params = {'n_neighbors': [3, 5, 7, 11, 21],
   'weights': ['uniform', 'distance'],
   'metric': ['euclidean', 'manhattan']}
gs = GridSearchCV(KNeighborsClassifier(), params, cv = 10, n_jobs = -1)
```

Như vậy, trong giá trị chỉ số khoảng cách, mô hình có thể áp dụng 2 trường hợp Euclidian hoặc Manhattan. Tương tự, chỉ số khoảng cách cũng có những lựa chọn bao gồm Uniform hoặc Distance đại diện cho 2 loại chỉ số khoảng cách là chia đều trọng số khoảng cách cho các mẫu dữ liệu, hoặc phân bổ phụ thuộc vào khoảng cách giữa 2 mẫu dữ liệu. Từ đó, thuật toán *GridSearchCV* sẽ nhập toàn bộ các lựa chọn này và đề xuất các chỉ số phù hợp nhất cho thuật toán phân loại KNN nhằm áp dụng cho bài toán phân loại. Kết quả các chỉ số được áp dụng trong mô hình KNN được lựa chọn bao gồm số lượng mẫu so sánh *k* "*n_neighbors*", chỉ số khoảng cách "*metric*", và trọng số của các điểm lân cận "*weights*" từ kết quả trả về của hàm *GridSearchCV*.

Sơ đồ khối: Quy trình phân loại lưu lượng Internet áp dụng cho mô hình KNN trọng số điểm lân cận được miêu tả bằng mô hình sơ đồ khối tại hình 3.7.



Hình 3.7. Sơ đồ khối mô hình phân loại KNN trọng số điểm lân cận

3.2.7 Mạng Neuron nhân tạo (ANN – Artificial Neural Networks)

Việc xây dựng thuật toán ANN sẽ được thực hiện qua việc dàn xếp cái lớp (layer) với số lượng neuron tương ứng với mỗi lớp. Với mô hình ANN, ta sẽ huấn luyện dựa vào các đặc trưng đầu vào với các nhãn tương ứng. Gọi X là ma trận chứa các điểm dữ liệu huấn luyện với $\mathcal{X} = [x_1, x_2, \dots, x_N] \in \mathbb{R}^{d \times N}$. Mỗi mỗi điểm dữ liệu tương ứng là mỗi cột x_i và số lượng đặc trưng d tương ứng. Trong đề tài này, 23 đặt trưng được áp dụng vì vậy d sẽ có giá trị là 23. Các nhãn tương ứng của các điểm dữ liệu được biểu diễn bằng vector $\mathcal{y} = [\mathcal{y}_1, \mathcal{y}_2, \dots, \mathcal{y}_N] \in \mathbb{R}^{1 \times N}$. Để xây dựng môt hình, thực chất là việc tìm ranh giới là một siêu mặt phẳng có phương trình (3.1) với vector hệ số và b là số hạng tự do bias.

$$f_w(x) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b = 0 \tag{3.1}$$

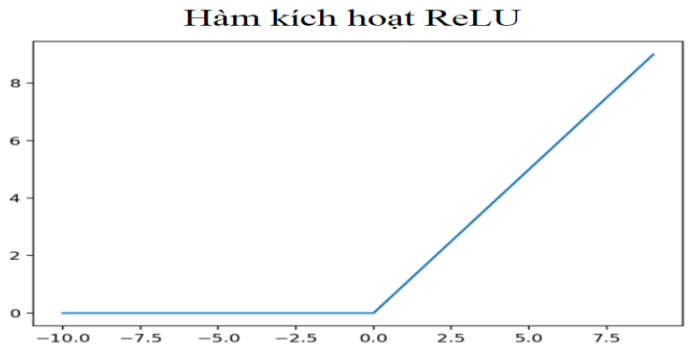
```
import numpy as np
def predict(w, X):


return np.sign(X.dot(w))


def perceptron(X, y, w_init):
    while True:
        pred = predict(w, X)
        mis_idxs = np.where(np.equal(pred, y) == False)[0]
        num_mis = mis_idxs.shape[0]
        if num_mis == 0: # no more misclassified points
return w


# random pick one misclassified point
random_id = np.random.choice(mis_idxs, 1)[0]
# update w
w = w + y[random_id]*X[random_id]
```

Ngoài ra, hàm kích hoạt được áp dụng sau mỗi lớp neuron với mục đích tăng tốc tốc đạo hàm góp phần tăng tốc độ huấn luyện và hạn chế vấn đề bốc hơi gradient (vanishing gradient). Trong đề tài này hàm ReLU được áp dụng trong tất cả các lớp neuron trong mô hình. Hàm trả về 0 nếu đầu vào là âm, nhưng đối với bất kỳ đầu vào dương nào, nó sẽ trả về giá trị đó và được thể hiện ở hình 3.8.



Hình 3.8. Hàm ReLU

```
import numpy as np

def Relu(z):
    if z > 0:
        return z
    else:
        return 0
```

Để thực hiện quá trình phân loại, mô hình xác suất cần được thiết lập ứng với mỗi đầu vào input x thì sẽ có a_n thể hiện xác suất để input đó rơi vào lớp thứ n. Vì vậy a_n sẽ có điều kiện cần là tổng giá trị chúng bằng một. Bên cạnh đó, giá trị đầu tra của neuron $z_n = w_n^T x$ sẽ tỉ lệ thuận với xác xuất dữ liệu rơi vào lớp thứ n.

Vì giá trị z_n là một tổ hợp tuyến tính của các thành phần của vector đặt trưng x. Tuy hàm kích hoạt được áp dụng trong phạm vi đề tài này vì vậy giá trị

của z_n sẽ có giá trị lớn 0 và thuận lợi cho việc đạo hàm. Việc áp dụng hàm số khả vi và chắc chắn chuyển z_n thành các giá trị dương được thực hiện bởi hàm số $\exp(z_n) = e^{z_n}$. Để đảm bảo tổng các xác suất a_n bằng một, chúng ta có thể áp dụng công thức (3.2).

$$a_n = \frac{\exp(z_n)}{\sum_{m=1}^{C} \exp\,(z_m)}, \forall m = 1,2,\dots,7 \tag{3.2}$$

Và xác suất a_n chúng ta có thể xem rằng như công thức (3.3)

$$p(y_k = n|x_k) = a_n \tag{3.3}$$

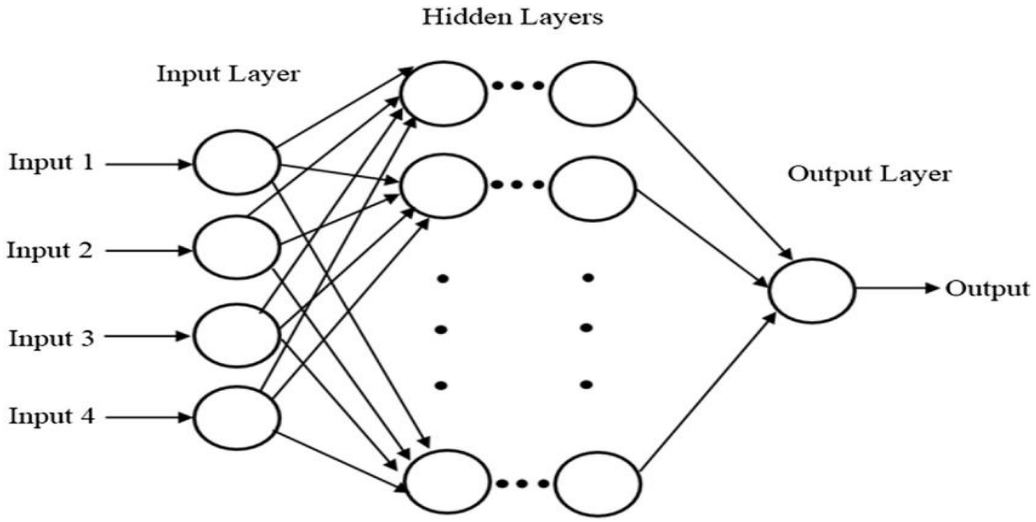
Hàm số trên được gọi là softmax function và được thể hiện bằng doạn code python như sau. Đầu vào input là một ma trận với các hàng là các vector z và đầu ra sẽ là một ma trận với mỗi hàng là giá trị xác suất a tương ứng.

```
import numpy as np
def softmax(Z):
        e_Z = np.exp(Z)
        A = e_Z / e_Z.sum(axis = 1, keepdims = True)
return A
```

Trong quá trình áp dụng mô hình ANN nhằm xây dựng mạng học sâu, các chỉ số *Dense* đại diện cho số lượng nút tại một lớp mạng ẩn, *activation* đại diện cho hàm kích hoạt được sử dụng tại lớp tương ứng, bao gồm hai giá trị là '*relu*' và 'softmax'. Mô hình huấn luyện sẽ được xây dựng dựa trên các chỉ số sau, từ đó áp dụng tập huấn luyện làm dữ liệu đầu và tiến hành huấn luyện. Sau đó, mô hình sẽ áp dụng nhóm dữ liệu kiểm tra, từ đó tiến hành phân loại và so sánh kết quả đầu ra nhằm đánh giá khả năng phân loại của mô hình.

```
#Define model
model = Sequential()
model.add(Dense(1024, input_shape=input_shape, activation = 'relu'))
model.add(Dense(512, activation = 'relu'))
model.add(Dense(256, activation = 'relu'))
model.add(Dense(256, activation = 'relu'))
model.add(Dense(128, activation = 'relu'))
model.add(Dense(64, activation = 'relu'))
model.add(Dense(num_classes, activation = 'softmax'))
```

Sơ đồ khối: Cấu trúc của mô hình huấn luyện ANN dung để phân loại lưu lượng Internet được miêu tả bằng mô hình sơ đồ tại hình 3.9.



Hình 3.9. Cấu trúc mô hình mạng ANN được áp dụng

3.2.8 Rừng ngẫu nhiên (RF - Random Forest):

Trong mô hình huấn luyện rừng ngẫu nhiên, như đã miêu tả tại mục 2.3.3, kết quả phân loại của các mẫu kiểm tra phụ thuộc vào kết quả bầu chọn của các mô hình cây quyết định thành phần tồn tại trong quá trình thiết kế mô hình rừng ngẫu nhiên. Kết hợp với ví dụ về quá trình áp dụng phương pháp điều chỉnh siêu tham số sử dụng hàm *GridSearchCV*, trong khi xây dựng mô hình huấn luyện, điều đầu tiên là tiến hành khai báo mô hình *RandomForestClassifier()* và các chỉ số tính toán cho quá trình phân nhánh của các cây quyết định thành phần với biến *param_grid*.

Trong quá trình thiết kế mô hình nhánh cây quyết định, chỉ số *max_depth* đại diện cho độ sâu tối đa mà một mô hình cây quyết định có thể phân nhánh đến. Các chỉ số *max_features* và *criterion* là các phép tính đại diện làm căn cứ để tiến hành phân chia tập dữ liệu dựa trên đặc trưng đang được đánh giá. Từ đó, thuật toán *GridSearchCV* sẽ tiến hành lần lượt thử nghiệm các chỉ số này từ đó đề xuất lựa chọn phù hợp nhất cho bài toán phân loại. Ý nghĩa của các chỉ số trên được giải thích trong mục 2.3.3.

```
model = RandomForestClassifier()
param_grid = {'max_depth': [40,50,60],
              'n_estimators': [90,100],
              'max_features': ['auto','log2'],
              'criterion': ['gini','entropy']}
GR = GridSearchCV(estimator = model, param_grid = param_grid, scoring = 'accuracy', cv = 6)
```

Với việc sử dụng hàm *GridSearchCV*, thuật toán sẽ ghi nhận mô hình huấn luyện rừng ngẫu nhiên và tiến hành điều chỉnh các siêu tham số được khai báo tại biến *param_grid* trong không gian điều chỉnh và tìm ra các giá trị siêu tham số phù hợp nhất dựa trên chỉ số *'accuracy'*. Lấy ví dụ điều chỉnh siêu tham số cho mô hình rừng ngẫu nhiên với tập dữ liệu NonVPN nhóm thời gian 15s, thuật toán sẽ khai báo tập dữ liệu mục tiêu và tiến hành điều chỉnh cho phù hợp.

```
dataset = arff.loadarff('TimeBasedFeatures-Dataset-15s-NO-VPN.arff')
df = pd.DataFrame(dataset[0])
```

Sau khi phân chia tập dữ liệu thành 2 tập hợp con là dữ liệu huấn luyện và kiểm tra, mô hình sẽ tiến hành tìm kiếm lưới bằng lệnh *fit* để ghi nhận các giá trị siêu tham số phù hợp nhất cho tập huấn luyện, từ đó thiết kế mô hình dựa trên các chỉ số tìm được và phân loại mẫu dữ liệu cho tập kiểm tra .Trong đó, *X_Train* là các mẫu dữ liệu cho tập huấn luyện và *y_train* là các nhãn dán tương với các mẫu dữ liệu huấn luyện.

```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.3, random_state = 42)
GR.fit(X_train, y_train)
```

Hình 3.10 ghi nhận giá trị các siêu tham số là kết quả phù hợp nhất trong tập hợp các giá trị đề xuất, theo quá trình tìm kiếm lưới. Theo đó, giả định sử dụng mô hình rừng ngẫu nhiên với các mô hình cây quyết định thành phần sử dụng những chỉ số trên, giá trị chính xác tham chiếu chéo trong tập dữ liệu huấn luyện có thể lên đến 93.8166%. Từng giá trị siêu tham số cụ thể được áp dụng trong mô hình cây quyết định có thể được truy xuất dựa vào lệnh *GR.best_params_*.

```
GR.fit(X_train, y_train)
```

```
GridSearchCV(cv=6, estimator=RandomForestClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [40, 50, 60],
                         'max_features': ['auto', 'log2'],
                         'n_estimators': [90, 100]},
             scoring='accuracy')
```

```
GR.best_score_
```

```
0.9381660521893993
```

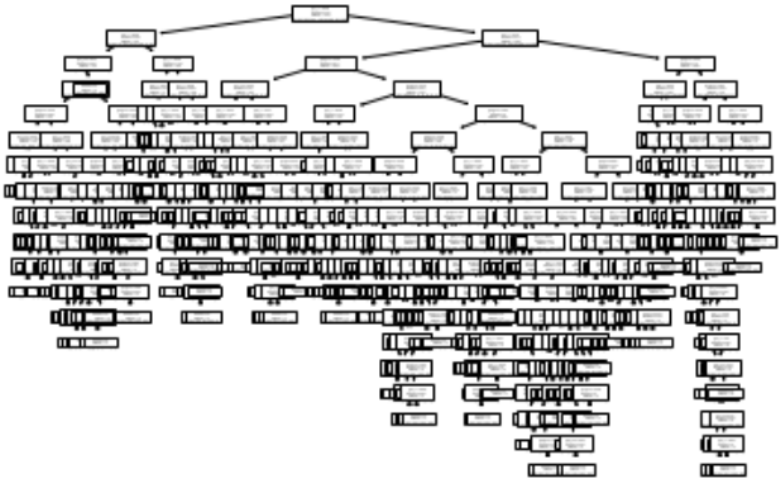
```
GR.best_params_
```

```
{'criterion': 'entropy',
 'max_depth': 40,
 'max_features': 'log2',
 'n_estimators': 100}
```

Hình 3.10. Kết quả quá trình tìm kiếm lưới cho giá trị siêu tham số - NonVPN – 15s

Dựa vào kết quả của quá trình điều chỉnh theo phương pháp tìm kiếm lưới, khi áp dụng mô hình rừng ngẫu nhiên cho mục đích phân loại cho dữ liệu NonVPN nhóm 15s, các mô hình cây quyết định thành phần sẽ có số lượng tối đa có thể đạt được là 100. Trong mỗi mô hình cây quyết định, độ sâu phân nhánh tối đa có thể có là 40 nút, và thuật toán quyết định phân nhánh dựa vào chỉ số độ lợi thông tin Entropy và thuật toán tính toán số lượng đặc trưng để phân nhánh là logarithm bậc 2. Hình 3.11 minh họa sơ đồ phân nhánh cụ thể của mô hình thứ 1 trong 100 cây quyết định thành phần áp dụng cho tập dữ liệu trên khi áp dụng lệnh *plot_tree*. Vì độ phức tạp của mỗi một mô hình cây quyết định là cực kỳ lớn, do đó việc sử dụng phương pháp điều chỉnh siêu tham số tự động tìm kiếm lưới có sự hỗ trợ vô cũng lớn trong quá trình phân loại.

```
from sklearn import tree
tree.plot_tree(GR.best_estimator_.estimators_[1])
```

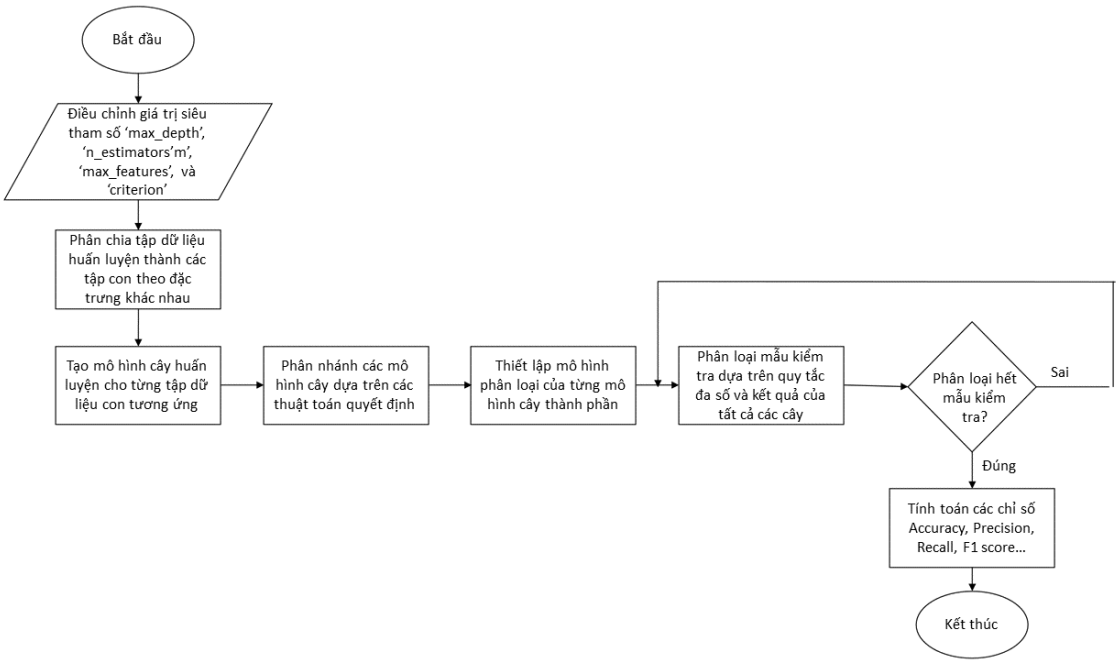


Hình 3.11. Minh họa mô hình cây quyết định thứ 1 trong mô hình rừng ngẫu nhiên

Trong mô hình thiết kế phạm vi đề tài này, sau quá trình thử nghiệm cho ra kết quả khả quan nhất đồng thời tham khảo từ các báo cáo nổi tiếng cùng lĩnh vực, số lượng mô hình cây quyết định tối đa được đặt trong mô hình rừng ngẫu nhiên, tương ứng với các giá trị tồn tại trong chỉ số *n_estimator*. Tương tự, số lượng tối đa nút có thể phân nhánh của một nhóm cây được khai báo bằng biến *max_depth*. Đồng thời, thuật toán đánh giá chỉ tiêu phân nhánh trước khi đưa ra quyết định phân nhóm dữ liệu dựa trên giá trị Độ lợi thông tin ứng với *criterion* và *max_feature*. Từ đó, thuật toán *GridSearchCV* lần lượt áp dụng các thông số được liệt kê trong các biến trên và luân phiên áp dụng chúng nhằm tìm ra giá trị tối ưu nhất cho từng chỉ số, qua đó áp dụng vào mô hình Rừng Ngẫu nhiên để giải quyết bài toán phân loại cho từng tập dữ liệu khác nhau, bao gồm tập dữ liệu lớn NonVPN và VPN với các nhóm thời gian khác nhau bao gồm 15,30,60 và 120s.

```
model = RandomForestClassifier()
param_grid = {'max_depth': [40,50,60],
              'n_estimators': [90,100],
              'max_features': ['auto','log2'],
              'criterion': ['gini','entropy']}
GR = GridSearchCV(estimator = model, param_grid = param_grid, scoring = 'accuracy', cv = 6)
```

Sơ đồ khối: Quy trình phân loại lưu lượng Internet áp dụng cho mô hình KNN trọng số điểm lân cận được miêu tả bằng mô hình sơ đồ khối tại hình 3.12.



Hình 3.12. Sơ đồ khối mô hình phân loại RF rừng ngẫu nhiên

Chương 4 : KẾT QUẢ THỰC NGHIỆM

4.1 Môi trường thực hiện

Môi trường phần mềm bao gồm Phần mềm Anaconda, Giao diện lập trình ứng dụng (Application programming interface – API) học sâu và API học máy cho ngôn ngữ lập trình Python như Tensorflow, Keras và Scikit Learn. Anaconda là một nền tảng mã nguồn mở về khoa học dữ liệu trên Python và R thông dụng nhất hiện nay với hơn 11 triệu người dùng. Anaconda bao gồm các gói thư viện Python quan trọng cho tính toán khoa học (khoa học dữ liệu, ứng dụng học máy, xử lý dữ liệu quy mô lớn, phân tích dự đoán, v.v.), nhằm mục đích đơn giản hóa việc triển khai và quản lý gói dữ liệu.

Scikit-learning (trước đây là scikits.learnm hay còn được gọi là sklearn) là một thư viện máy học phần mềm miễn phí cho ngôn ngữ lập trình Python. Scikit Learn hỗ trợ các thuật toán phân loại, hồi quy và phân cụm khác nhau bao gồm máy vec-tơ hỗ trợ, rừng quyết định ngẫu nhiên (Random Forest), Gredient boosting, K – trung bình và DBSCAN, đồng thời được thiết kế để tương tác với các thư viện số học và khoa học của Python như NumPy và SciPy.

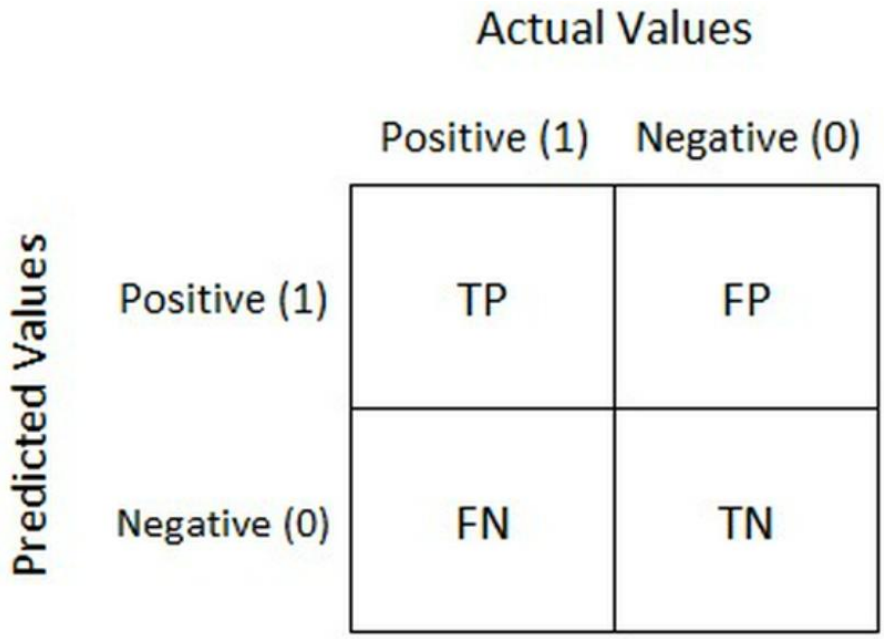
Scikit-learning phần lớn được viết bằng ngôn ngữ Python và sử dụng nhiều hàm có trong thư viện NumPy, nhằm thực hiện các phép toán mảng và đại số tuyến tính mang hiệu suất cao. Một số thuật toán cốt lõi được viết bằng Python để cải thiện hiệu suất tính toán. Ngoài ra, Scikit-learning tích hợp tốt với nhiều thư viện Python khác, chẳng hạn như Matplotlib và plotly để vẽ biểu đồ, NumPy để vec-tơ hóa mảng, khung dữ liệu Pandas, SciPy, và nhiều thư viện hơn nữa.

4.2 Các chỉ số đánh giá (Evaluation metrics)

4.2.1 Ma trận nhầm lẫn (Confusion Matrix)

Trong lĩnh vực máy học và cụ thể là trong các bài toán phân loại thống kê, ma trận nhầm lẫn, hay còn được gọi là ma trận lỗi, là một ma trận thể hiện bố cục cụ thể cho phép người dùng hình dung được hiệu suất phân loại của một thuật toán. Ma trận

nhầm lẫn thường được sử dụng cho các mô hình Học có giám sát (trong mô hình Học không giám sát thường được gọi là ma trận so khớp). Mỗi hàng của ma trận đại diện cho các nhãn thực tế (ground truth) của mẫu trong một lớp, trong khi mỗi cột đại diện cho các kết quả dự đoán (prediction) mẫu trong một lớp, hoặc ngược lại. Cái tên ma trận nhầm lẫn bắt nguồn từ thực tế đây là một công cụ, hoặc chỉ số lượng hóa để đánh giá, giúp người dùng dễ dàng nhận ra liệu thuật toán có sự nhầm lẫn trong quá trình phân loại mẫu giữa hai lớp hay không. Ma trận nhầm lẫn được minh họa bằng hình 4.1



Hình 4.1. Ma trận hỗn loạn

Một ma trận hỗn loạn có bao gồm 4 thông số, trong đó giả định mẫu kiểm tra thuộc lớp P2P hoặc không thuộc lớp P2P.

- TP (True Positive): Số lần dự đoán đúng kết quả lớp của mẫu khớp với lớp thực tế. Ví dụ như mẫu kiểm tra thuộc lớp P2P và mô hình dự đoán chính xác mẫu thuộc lớp đó.
- TN (True Negative): Số lần gián tiếp dự đoán đúng kết quả lớp của mẫu. Từ ví dụ trên, mẫu kiểm tra không thuộc lớp P2P và mô hình dự đoán chính xác mẫu thuộc các lớp khác như Streaming hay Web Browsing.

- FP (False Positive – Type 1 Error): Số lần mô hình dự đoán sai lệch, nghĩa là mẫu kiểm tra thuộc nhóm Streaming nhưng lại phân loại vào lớp P2P.
- FN (False Negative – Type 2 Error): Số lần mô hình dự đoán sai lệch gián tiếp, tức là mẫu kiểm tra thuộc nhóm P2P nhưng lại phân vào các lớp khác.

Dựa trên các thông số trên, những chỉ số đánh giá sau có thể được tính toán như Độ chính xác Accuracy, Precision, Recall, F1 score, v v…

4.2.2 Các chỉ số đánh giá

Độ chính xác (Accuracy): là tổng số lần mẫu kiểm tra được phân loại đúng lớp của mình, được minh họa bằng công thức sau:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (4.1)$$

Trong trường hợp tập dữ liệu có kích thước gần bằng nhau, độ chính xác Accuracy có thể được sử dụng để cung cấp cho chúng ta các giá trị được phân loại chính xác. Tuy nhiên, để đánh giá tổng quan kết quả phân loại của mô hình, cần có những chỉ số khác nhằm đánh giá cả những chỉ số cho những kết quả phân loai sai.

Precision: Chỉ số đánh giá khi mẫu kiểm tra được phân loại thuộc về một lớp, số lần thực sự mà mẫu thuộc về lớp đó. Ví dụ trên tổng số lần mẫu kiểm tra được phân loại về lớp P2P, chỉ số Precision chỉ ra được tỷ lệ bao nhiêu mẫu thật sự thuộc về lớp đó.

$$Precision = \frac{TP}{TP + FP}, \quad (4.2)$$

Trong trường hợp số lượng mẫu trong các lớp bị mất cân bằng, chỉ số Precision hỗ trợ trong việc đánh giá một cách chính xác số lần mô hình thực sự dự đoán đúng nhãn của mẫu kiểm tra.

Recall: thông số chỉ ra khi mẫu kiểm tra thực sự thuộc về một lớp, số lần mẫu kiểm tra được phân loại thuộc về lớp đó trong những lần dự đoán. Ví dụ như, khi mà

một mẫu kiểm tra thuộc về lớp Web Browsing, chỉ số Recall tính toán tỷ lệ số lần hệ thống thường xuyên phân loại mẫu kiểm tra thuộc về lớp đó.

$$Recall = \frac{TP}{TP + FN}, \tag{4.3}$$

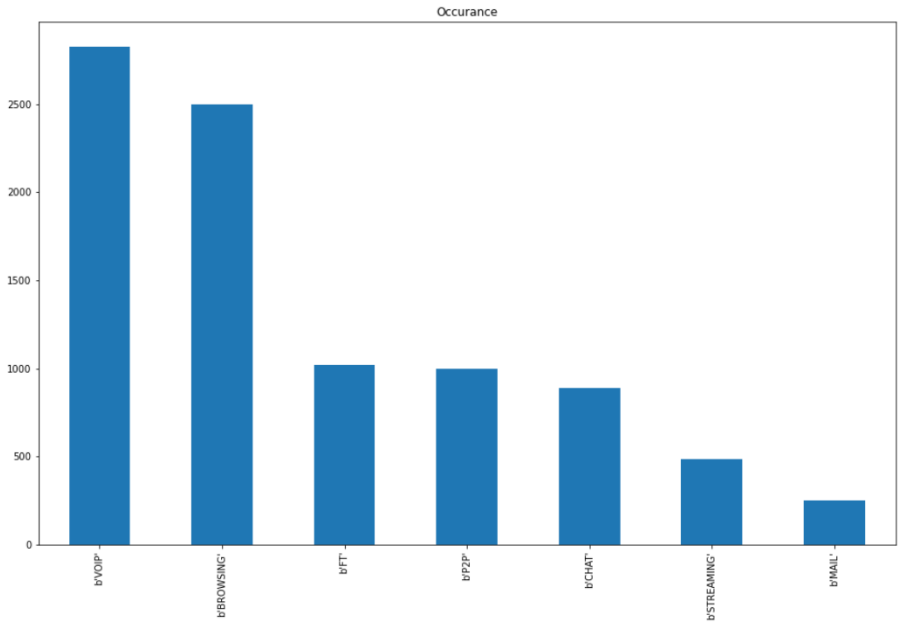
F1 score: Tổng hợp 2 chỉ số Precision và Recall, một mô hình có chỉ số F-score cao chỉ khi cả 2 chỉ số Precision và Recall đều cao. Một trong hai chỉ số này thấp đều sẽ kéo điểm F1-score xuống. Trường hợp xấu nhất khi 1 trong hai chỉ số Precison và Recall bằng 0 sẽ kéo điểm F-score về 0. Trường hợp tốt nhất khi cả điểm chỉ số đều đạt giá trị bằng 1, khi đó điểm F-score sẽ là 1. Qua việc sử dụng chỉ số F-score, bài toán phân loại sẽ là một thước đo đáng tin cậy về hiệu năng của mô hình trong các bài toán phân loại, đặc biệt khi dữ liệu về một lớp lớn hơn gấp nhiều lần so với dữ liệu về lớp còn lại.

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall}, \tag{4.4}$$

4.3 Kết quả đạt được

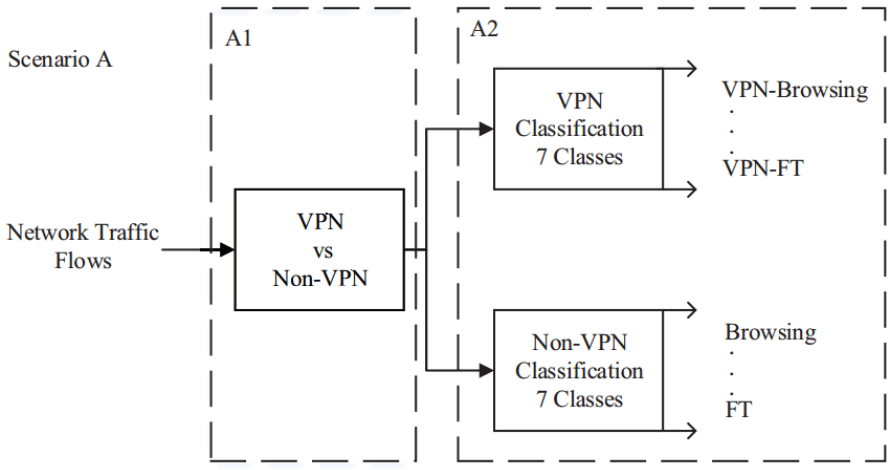
4.3.1 Miêu tả các bối cảnh thí nghiệm

Như đã miêu tả trong mục 3.1, tập dữ liệu ISCXVPN2016 bao gồm 7 loại lưu lượng khác nhau, mật độ phân bổ của từng lớp lưu lượng được miêu tả trong hình 4.2. Dựa trên mật độ phân bổ mẫu dữ liệu không đồng đều của các lớp, những chỉ số được sử dụng để đánh giá hiệu suất của mô hình đề xuất sẽ là chỉ số Accuracy, Precision, Recall and F1 score.



Hình 4.2. Mật độ phân bố mẫu dữ liệu của các lớp lưu lượng Internet

Ngoài ra, trong tập dữ liệu ISCXVPN2016, có hai trướng hợp khác nhau trong tập dữ liệu được sử dụng phục vụ cho mục đích nghiên cứu là lưu lượng mạng Internet VPN và Non-VPN. Do đó, có 2 bối cảnh thử nghiệm được đề xuất, ký hiệu bằng bối cảnh A1 và A2. Trong bối cảnh A, mục tiêu trước hết là phân loại 2 loại dữ liệu VPN và Non-VPN. Sau đó, trong mỗi loại dữ liệu, mô hình sẽ phân loại từng lớp trong 7 loại lưu lượng Internet khác nhau. Như vậy, tổng cộng bối cảnh A sẽ chia làm 2 bước A1, phân loại dữ liệu VPN và NonVPN, và bước A2, phân loại 7 lớp lưu lượng mạng Internet tương ứng của 2 loại dữ liệu trên.



Hình 4.3. Mô hình minh họa 2 bối cảnh thí nghiệm A1 và A2

Tuy nhiên, tập dữ liệu 2 trường hợp trên bao gồm cả dữ liệu được ghi nhận trong các khung thời gian là 15, 30, 60 và 120s [21]. Dựa trên tính chất này, mô hình phân loại có thể tiến hành thực hiện các phân tích dựa trên dữ liệu khung thời gian khác nhau, ngoài phân tích dựa trên những lớp lưu lượng Internet. Từ đó, các thuật toán KNN và ANN được áp dụng riêng biệt theo từng bối cảnh và từng tập dữ liệu dựa trên các khung thời gian khác nhau, từ đó tiến hành phân tích và so sánh những kết quả đạt được.

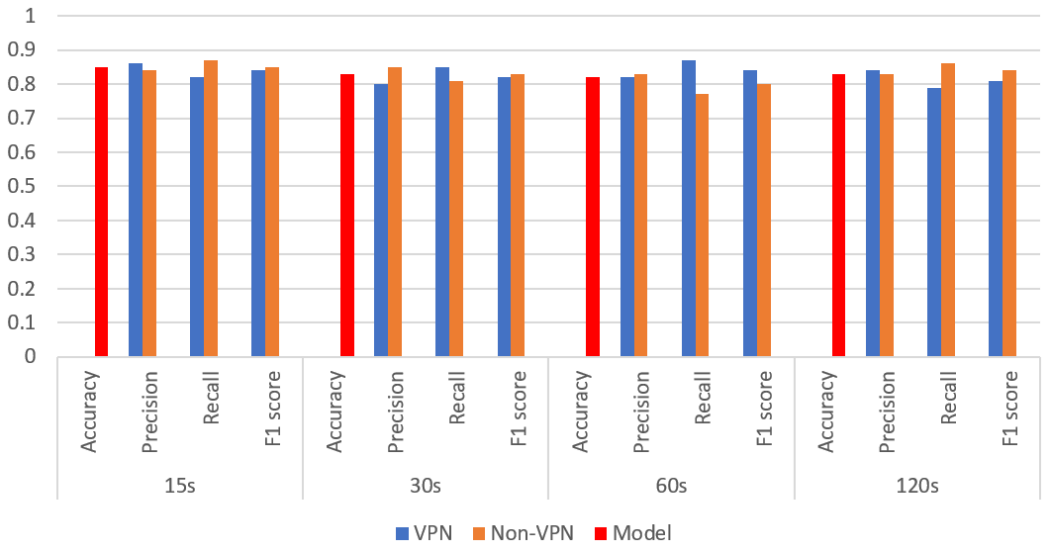
Ngoài ra, do tập dữ liệu ISCXVPN2016 có các nhãn đã được phân lớp cụ thể, phương án huấn luyện có thể được tối ưu hóa bằng việc phương pháp học có giám sát (supervised learning), phân chia nhóm dữ liệu thành 2 phần bao gồm tập huấn luyện (training set) và tập kiểm tra (test set). Trong đó, mô hình huấn luyện sẽ ghi nhận các đặc trưng và nhãn của tập huấn luyện và tiến hành học tập các đặc trưng của mẫu dữ liệu. Tiếp theo, mô hình sẽ tiến hành phân loại các mẫu trong tập kiểm tra, tìm kiếm những đặc trưng tương đồng với các mẫu trong tập huấn luyện, từ đó đưa ra kết quả phân loại dự đoán (predicted results) và so sánh kết quả dự đoán trên với kết quả nhãn thực tế (actual hoặc annotation labels).

Ưu điểm của phương án huấn luyện này dựa trên việc các nhãn của mẫu dữ liệu trong tập kiểm tra đều đã được phân lớp trước, giúp tối ưu hóa hiệu suất trong quá trình huấn luyện mô hình. Trong phạm vi đề tài này, dữ liệu ISCXVPN2016 sẽ

được chia thành tập huấn luyện và tập kiểm tra theo tỷ lệ 7:3, trong đó tập huấn luyện chiếm 7 phần mẫu dữ liệu, và tập kiểm tra chiếm 3 phần còn lại. Tỷ lệ phân chia 2 tập dữ liệu này được lựa chọn dựa trên việc các mô hình huấn luyện trong ứng dụng phân loại đều lựa chọn tỷ lệ sao cho số lượng mẫu dữ liệu huấn luyện đủ nhiều. Tuy nhiên, số lượng mẫu trong tập kiểm tra cũng phải đủ lớn để tránh hiện tượng kết quả phân loại nhãn dán có độ lệch lớn (bias).

4.3.2 Kết quả thu được – Mô hình KNN

Bối cảnh A1: Trong bước phân loại này, mục tiêu của mô hình nhằm đến việc phân loại liệu các lớp dữ liệu đầu vào có được mã hóa VPN hoặc không, tương đương với một mô hình phân loại nhị phân. Tại bối cảnh này, tổng cộng có 4 nhóm dữ liệu dựa theo khung thời gian 15, 30, 60 và 120s. Trong hình 4.4, những giá trị Accuracy, Precision, Recall và F1 score, được báo cáo và biểu diễn dưới dạng biểu đồ cột.

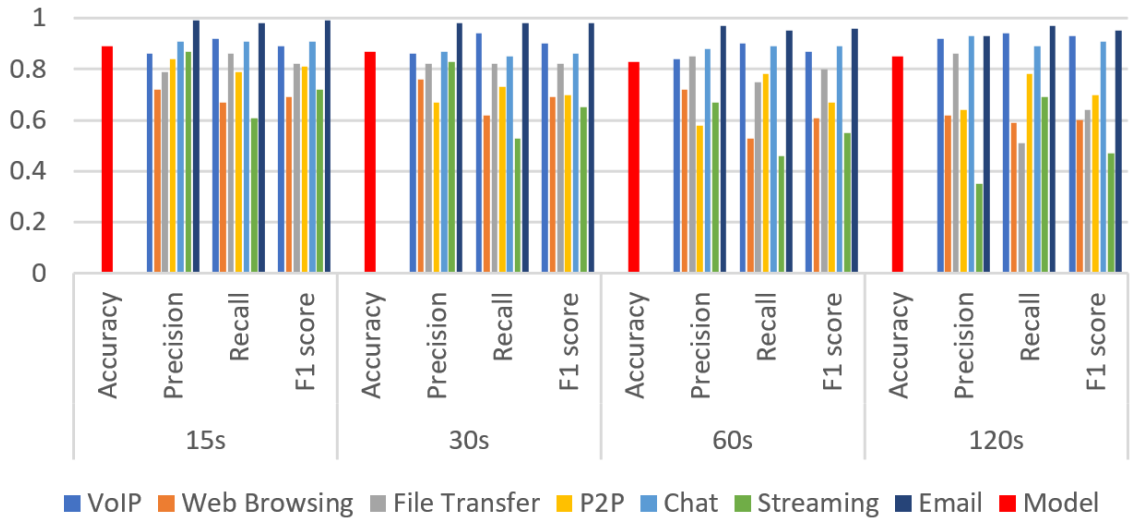


Hình 4.4. Biểu đồ chỉ số đánh giá kết quả phân loại bằng KNN - Bối cảnh A1

Từ biểu đồ trên, có thể thấy được kết quả phân loại của mô hình KNN đạt được kết quả tốt nhất tại chuỗi dữ liệu có thời gian ghi nhận 15s. Trong đó, mô hình ghi nhận các chỉ số đánh giá lần lượt là Accuracy đạt 84.86%, Precision đạt 86% cho lớp VPN và 84% cho lớp Non-VPN, Recall có mức 82% cho lớp VPN và 87% cho lớp Non-VPN, cuối cùng F1 score đạt 84% lớp VPN và 85% cho lớp Non-VPN. Các chỉ

số đánh giá kết quả phân loại thành công mang chiều hướng giảm nhẹ 2-3% cho các mục khi chuyển qua các tập dữ liệu gia tăng số thời gian ghi nhận dữ liệu lên 30, 60 và 120s. Tiêu biểu như chỉ số độ chính xác Accuracy, từ mức 84.86% giảm xuống lần lượt những giá trị 82.58%, 82.02%, 82.44% cho các gói dữ liệu 30, 60 và 120s.

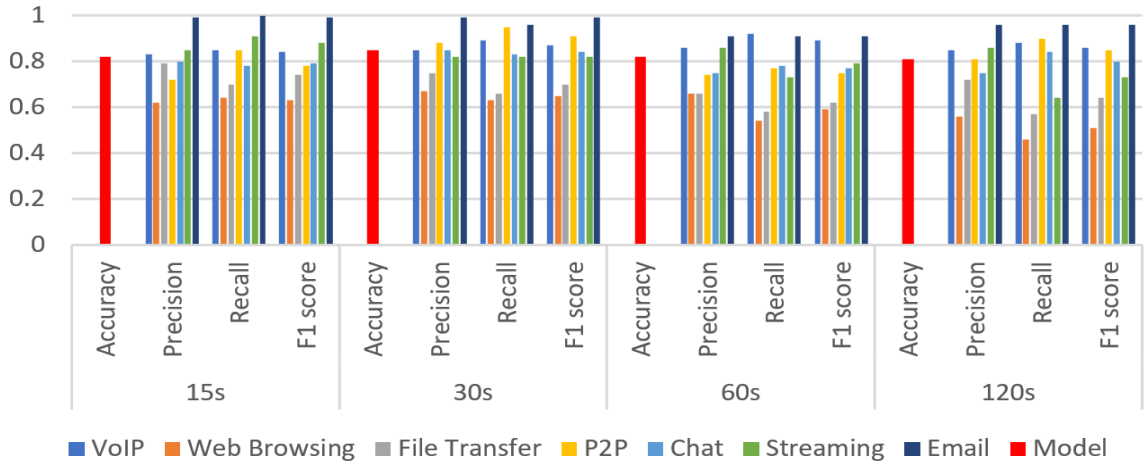
Bối cảnh A2: Trong bối cảnh A2, dữ liệu mạng Internet được mã hóa VPN và không mã hóa Non-VPN được phân loại thành 7 lớp lưu lượng mạng tương ứng. Biểu đồ so sánh chỉ số phân loại cho từng loại dữ liệu mạng được cung cấp lần lượt trong hình 4.5 và 4.6.



Hình 4.5. Biểu đồ chỉ số đánh giá không mã hóa Non-VPN bằng KNN - Bối cảnh A2

Trong bối cảnh A2 tập dữ liệu không mã hóa Non-VPN, chỉ số Accuracy trung bình của các nhóm dữ liệu đều cho kết quả khả quan trên 80%. Chỉ số Accuracy trung bình cao nhất thuộc về tập dữ liệu 15s là 87.81% và giá trị của nhóm dữ liệu 120s là 88.29%. Tuy nhiên, kết quả khảo sát độ chính xác khi phân lớp mẫu kiểm tra chỉ dao động ở mức 60-70%. Các chỉ số Precision và Recall có sự dao động nhẹ giữa các tập dữ liệu cho kết quả tốt nhất. Ví dụ như chỉ số Precision tốt nhất thuộc về nhóm dữ liệu 30s với 73.86%, cao hơn 10% so với nhóm 120s. Chỉ số Recall của nhóm 120s đạt mức cao nhất 68.92%. Tuy nhiên, kết quả F1-score đại diện cho 2 chỉ số trên đạt kết quả cao nhất thuộc tập dữ liệu 15s với mức 68.24%, trong khi chỉ số này đạt mức 66.86%, 61.82%, 64.97%, thuộc các nhóm dữ liệu tương ứng 30, 60 và 120s.

Trong bước phân loại các lớp dữ liệu lưu lượng mạng, lớp Email đạt được kết quả phân loại chính xác gần như tuyệt đối các chỉ số Precision, Recall, F1-score ở mức 95-98%. Kết quả của lớp VoIP và Chat xấp xỉ đạt mức cao thứ 2 qua các gói dữ liệu thời gian khác nhau, chỉ thấp hơn lớp Email khoảng 5-10%. Lớp dữ liệu Web Browsing cho kết quả thấp nhất ở tất cả các chỉ số và các nhóm dữ liệu, chỉ số Precision và Recall chỉ xoay quanh mức 60%, dẫn đến kết quả chỉ số F1-score ở các nhóm dữ liệu cũng đạt kết quả thấp, lần lượt là 68%, 67%, 62% và 65% các nhóm 15,30,60 và 120s.



Hình 4.6. Biểu đồ chỉ số đánh giá mã hóa VPN bằng KNN - Bối cảnh A2

Tại tập dữ liệu mã hóa VPN, chỉ số Accuracy trung bình của các nhóm dữ liệu tuy đều trên 80%, nhưng đều thấp hơn so với nhóm dữ liệu Non-VPN, chỉ đạt khoảng 82-83%. Giá Accuracy trung bình cao nhất chỉ đạt mức 83.28%, thấp hơn cả giá trị Accuracy thấp nhất 84.77% thuộc nhóm 60s dữ liệu Non-VPN. Các chỉ số giá trị trung bình của Precision và Recall cũng có xu hướng giảm ở toàn bộ các nhóm dữ liệu thời gian khác nhau, dao động xung quanh mức 52-67%. Điều này dẫn đến kết quả F1-score cũng giảm một khoảng lớn khi mà kết quả trung bình F1-score cao nhất thuộc về nhóm 30s đạt 63.09%, trong khi mức thấp nhất thuộc nhóm 120s chỉ đạt mức 56.47%.

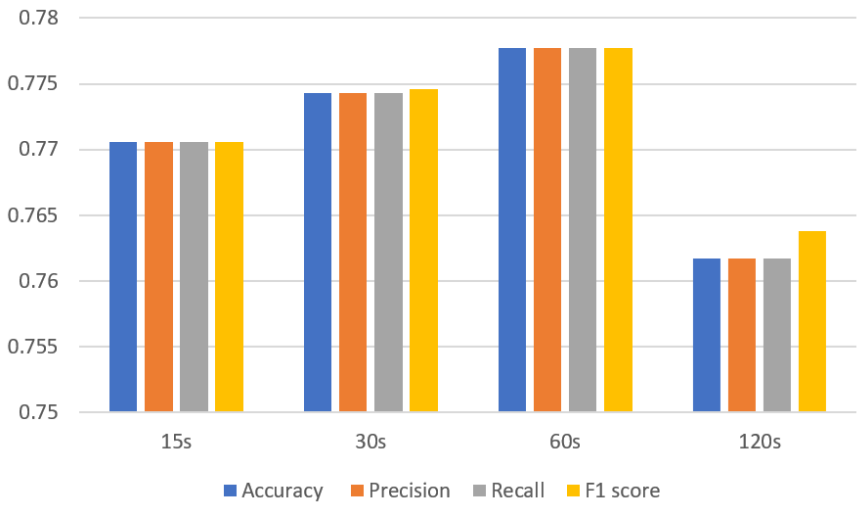
Trong nhóm các lớp dữ liệu lưu lượng mạng, lớp Email một lần nữa đạt được kết quả phân loại chính xác cao nhất. Kết quả Recall và Precision của các lớp VoIP,

Chat, P2P và Streaming dao động khá nhiều qua các nhóm dữ liệu thời gian khác nhau, nhìn chung đạt kết qua cao hơn 80%. Lớp dữ liệu Web Browsing cho kết quả thấp nhất ở tất cả các chỉ số, đặc biệt tại nhóm dữ liệu 120s, chỉ số Precision và Recall chỉ đạt mức 61% và 53%, dẫn đến kết quả chỉ số F1-score cũng chỉ đạt mức 56% thấp nhất trong tất cả các chỉ số trong 14 lớp phân loại lưu lượng mạng, kể cả loại dữ liệu mã hóa VPN và không mã hóa Non-VPN.

4.3.3 Kết quả thu được – Mô hình ANN

Tương tự với mô hình KNN, mô hình ANN cũng được đánh giá qua 2 bối cảnh A1 và A2 nhằm đánh giả khả năng phân loại lưu lượng Internet.

Bối cảnh A1: Tại bối cảnh phân loại dữ liệu không được mã hóa Non-VPN và có mã hóa VPN, tổng cộng có 4 nhóm dữ liệu dựa theo khung thời gian 15, 30, 60 và 120s. Kết quả phân loại tương tự được báo cáo dựa trên 4 chỉ số Accuracy, Precision, Recall và F1 score và ghi nhận trong biểu đồ hình 4.7. Các chỉ số đánh giá khả năng phân loại trong mỗi khung thời gian đều mang giá trị xấp xỉ nhau, và có chiều hướng tăng dần theo khung thời gian ghi nhận dữ liệu. Trong đó, giá trị các chỉ số tại khung thời gian 15s đạt 77.06%, 30s là 77.43% và đạt đỉnh cao nhất tại khung thời gian 60s với giá trị 77.77%. Tuy nhiên, khi gia tăng gấp đôi khung thời gian ghi nhận dữ liệu, các chỉ số đánh giá có sự suy giảm nhẹ khi các chỉ số Accuracy, Precision và Recall đạt còn 76.17%, chỉ có chỉ số F1 score đạt mức nhỉnh hơn ở khoảng 76.38%.

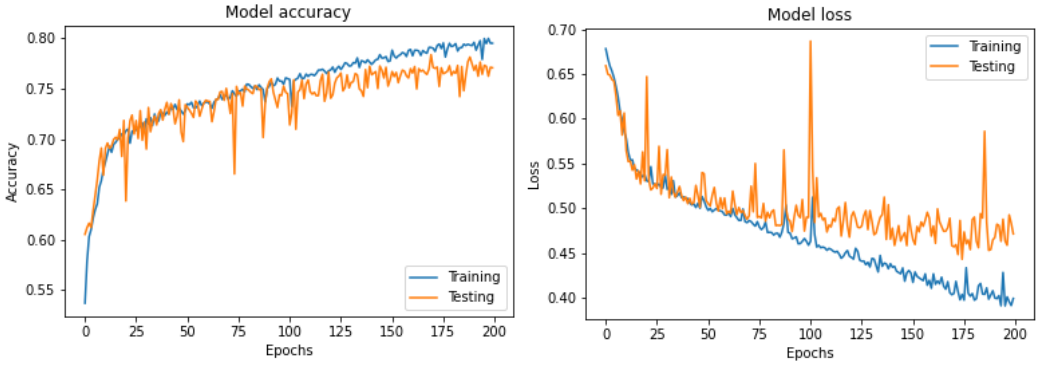


Hình 4.7. Biểu đồ chỉ số đánh giá kết quả phân loại bằng ANN - Bối cảnh A1

Ngoài 4 chỉ số đánh giá trên, mô hình ANN còn được đánh giá dựa trên việc phân tích đồ thị hàm chính xác (Accuracy function) và đồ thị hàm mất mát (Loss function), biểu diễn khả năng học qua số epoch mà mô hình được huấn luyện. Trong định nghĩa học máy ANN, epoch xác định thời điểm thuật toán huấn luyện (learning algorithm) đã hoàn tất việc đưa toàn bộ dữ liệu huấn luyện vào mô hình huấn luyện mạng máy tính một lần. Dựa trên biểu hiện phản ánh thông qua các đồ thị Accuracy function và Loss function, khả năng huấn luyện của mô hình ANN cũng được đánh giá trực quan hơn. Hình 4.8 minh họa 2 hàm số trên của khung thời gian ghi nhận dữ liệu 15s, trong đó số epoch huấn luyện mô hình là 200.

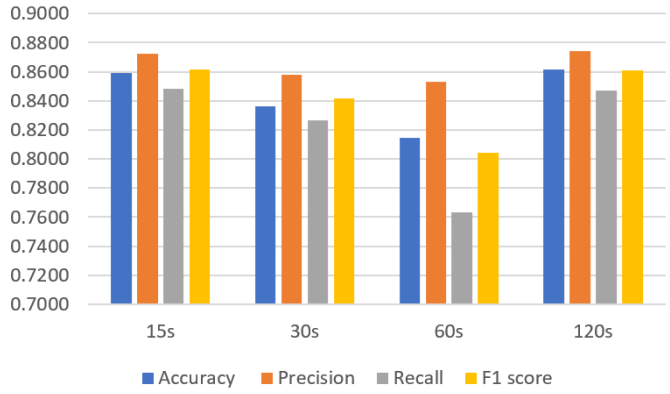
Trong mô hình hàm Chính xác, tỷ lệ phân loại của cả hai tập dữ liệu huấn luyện và kiểm tra đều khớp với nhau sau khoảng thời gian huấn luyện 100 epoch. Sau đó, 2 hàm kết quả bắt đầu phân kỳ và tạo khoảng cách lớn dần đến hết khoảng thời gian huấn luyện cuối là 200 epoch. Hiện tượng này thậm chí còn được quan sát rõ hơn trong đồ thị hàm mất mát, trong đó kết quả tập kiểm tra có biên độ dao động rất lớn bắt đầu từ epoch 100, từ đó hàm số biểu diễn của 2 tập dữ liệu phân kỳ mạnh đến cuối thời gian huấn luyện. Từ đó, mô hình huấn luyện có thể đưa ra dự đoán sai lệch trên kết quả của các mẫu trong tập kiểm tra, hay còn được gọi là mô hình huấn luyện quá khớp (overfitting). Hiện tượng quá khớp nó chỉ ra rằng mô hình này thiếu khả

năng tổng quát hóa khi có dữ liệu mới được đưa vào mô hình huấn luyện. Nếu một mô hình không thể tổng quát hóa dữ liệu mới, mô hình đó sẽ không thể thực hiện bài toán phân loại hoặc dự đoán khớp với yêu cầu ban đầu của mô hình.



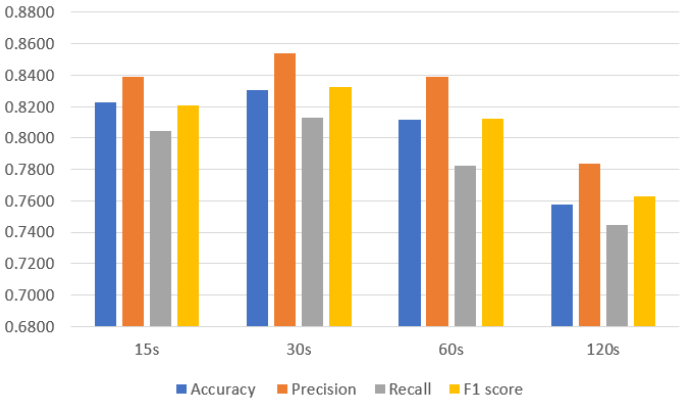
Hình 4.8. Đồ thị hàm Chính xác và hàm Mất mát của mô hình ANN – Khung 15s

Bối cảnh A2: Trong bối cảnh A2, dữ liệu mạng Internet được mã hóa VPN và không mã hóa Non-VPN được phân loại thành 7 lớp lưu lượng mạng tương ứng. Hình 4.9 báo cáo kết quả phân loại 7 lớp lưu lượng không mã hóa VPN bằng mô hình huấn luyện ANN. Trong cả 4 khung thời gian ghi nhận dữ liệu, chỉ số Precision luôn đạt mức cao nhất, dao động lần lượt là 87.27%, 85.8%, 85.32% và 87.45% tương ứng với khung 15s, 30s, 60s và 120s. Trái lại, chỉ số Recall lại luôn ở mức thấp nhất trong tất cả các trường hợp, với giá trị giảm đến mức thấp nhất trong khung 60s chỉ đạt mức 76.34%. Chiều hướng đối ngược của 2 chỉ số Precision và Recall trực tiếp ảnh hưởng đến giá trị F1 score, khi mà trong khung 60s, giá trị F1 score giảm mạnh chỉ còn ở 80.44%, thấp hơn khá nhiều so với các khung thời gian khác, trong đó 2 khung thời gian 15s và 120s đạt giá trị tương tự nhau lần lượt ở mức 86.16% và 86.11%. Tuy nhiên, nhìn chung các chỉ số đánh giá trong nhóm dữ liệu không mã hóa VPN đặt ở mức ổn định và đồng đều cao.



Hình 4.9. Biểu đồ chỉ số đánh giá kết quả phân loại bằng ANN – bối cảnh A2 Non-VPN

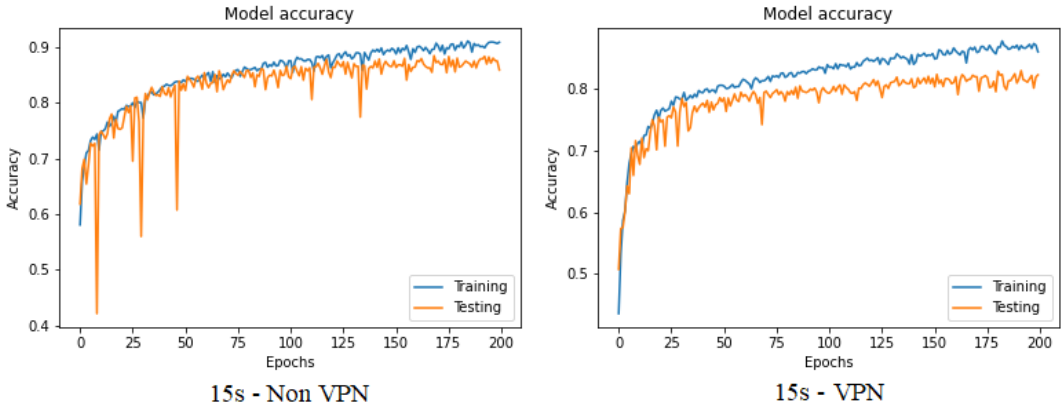
Ngược lại với chiều hướng độ chính xác có dấu hiệu ổn định lại nếu được ghi nhận trong khung thời gian dài hơn là 120s, các chỉ số kết quả phân loại tập dữ liệu mã hóa VPN trên hình 4.10 cho thấy xu hướng giảm mạnh theo thời gian. Nhóm chỉ số Accuracy và F1 score ở các khung thời gian khác nhau đều có giá trị xấp xỉ, điều này chỉ ra rằng khả năng phân loại lưu lượng các nhóm dữ liệu có mã hóa của mô hình ANN đạt mức độ ổn định cao, nhưng hiệu quả phân loại lại giảm theo thời gian ghi nhận dữ liệu.



Hình 4.10. Biểu đồ chỉ số đánh giá kết quả phân loại bằng ANN – bối cảnh A2 Non-VPN

Về mặt hiệu quả, kết quả của 4 loại chỉ số ở mức 120s giảm mạnh còn khoảng 75% - 78%. Trong đó, chỉ số Accuracy là 75.74%, Precision là 78.38%, Recall là 74.44% và F1 score là 76.29%, thấp nhất trong tất cả các tập dữ liệu dù có mã hóa

hay không mã hóa VPN. Trong đồ thị minh họa hàm số chính xác và mất mát, kết quả chỉ ra được nhóm dữ liệu không mã hóa VPN ít bị ảnh hưởng bởi hiện tượng quá khớp hơn là nhóm dữ liệu mã hóa VPN, minh họa bởi các hàm số chính xác trong nhóm dữ liệu khung thời gian 15s hình 4.11.

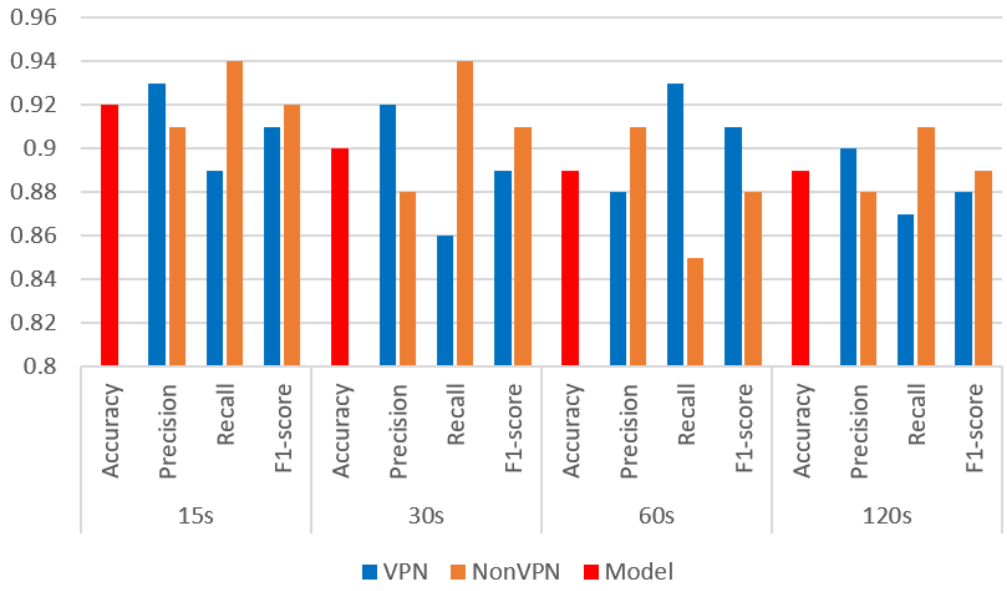


Hình 4.11. Hàm số chính xác của 2 nhóm dữ liệu không mã hóa VPN và có mã hóa VPN - Khung 15s

4.3.4 Kết quả thu được – Mô hình RF

Tương đương với các phép đánh giá mô hình KNN và mô hình ANN, mô hình Rừng ngẫu nhiên RF cũng được đánh giá qua 2 bối cảnh A1 và A2 nhằm kiểm tra khả năng phân loại lưu lượng Internet. Đánh giá tổng quan cho thấy, kết quả thu được từ mô hình phân loại Rừng ngẫu nhiên RF cho ra kết quả phân loại tốt hơn cả hai mô hình KNN và ANN.

Bối cảnh A1: Tại bối cảnh phân loại nhị phân cho tập dữ liệu gồm 2 nhóm dữ liệu không được mã hóa Non-VPN và có mã hóa VPN, tổng cộng có 4 nhóm dữ liệu dựa theo khung thời gian 15, 30, 60 và 120s. Kết quả phân loại tương tự được báo cáo dựa trên 4 chỉ số Accuracy, Precision, Recall và F1 score và ghi nhận trong biểu đồ hình 4.12.



Hình 4.12. Biểu đồ chỉ số đánh giá kết quả phân loại bằng RF - Bối cảnh A1

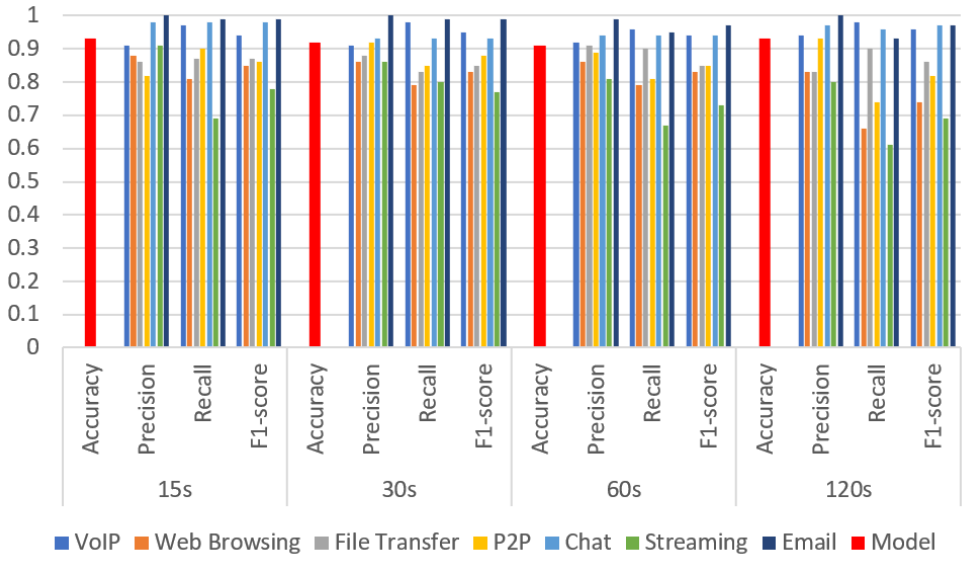
Nhìn chung, chỉ số đo lường kết quả phân loại Accuracy giảm dần theo sự gia tăng của thời gian ghi nhận dữ liệu, trong đó nhóm dữ liệu 15s đầu đầu với giá trị Accuracy lên đến 92%, và giảm còn 88% tại 2 nhóm dữ liệu 60s và 120s. Về chỉ số Precision, tập dữ liệu VPN phần lớn cho kết quả cao hơn tập Non-VPN với giá trị nằm trong khoảng 90-92%. Chỉ duy nhất trong nhóm dữ liệu 60s thì nhóm dữ liệu Non-VPN chiếm tỷ số cao hơn là 91% so với con số 88% nhóm mã hóa VPN.

Ngược lại trong các giá trị chỉ số Recall, tập dữ liệu mã hóa VPN lại cho cung cấp kết quả thấp hơn 4-8% so với nhóm Non-VPN. Tuy nhiên, một lần nữa, trong nhóm dữ liệu thời gian 60s, lại cho kết quả ngược lại hoàn toàn khi chỉ số Recall nhóm VPN chiếm tới 93% so với 85% Non-VPN. Sự đối nghịch này mang lại giá trị cân bằng trong chỉ số F1-score cho cả 2 tập dữ liệu mã hóa và không mã hóa, nhưng trung bình chung giá trị phân loại đạt kết quả cao từ 89-91% đều qua cả 4 khung thời gian. Kết quả cho thấy mô hình Rừng ngẫu nhiên này rất phù hợp để phân loại tập dữ liệu trong bối cảnh A1 này.

Bối cảnh A2: Tương tự với 2 mô hình trên, trong bối cảnh này, mô hình Rừng ngẫu nhiên sẽ phân loại từng tập dữ liệu mạng mã hóa VPN và không mã hóa Non-

VPN thành 7 lớp lưu lượng mạng khác nhau. Báo cáo chỉ số phân loại từng nhóm lưu lượng mạng được liệt kê trong hình 4.13 và 4.14.

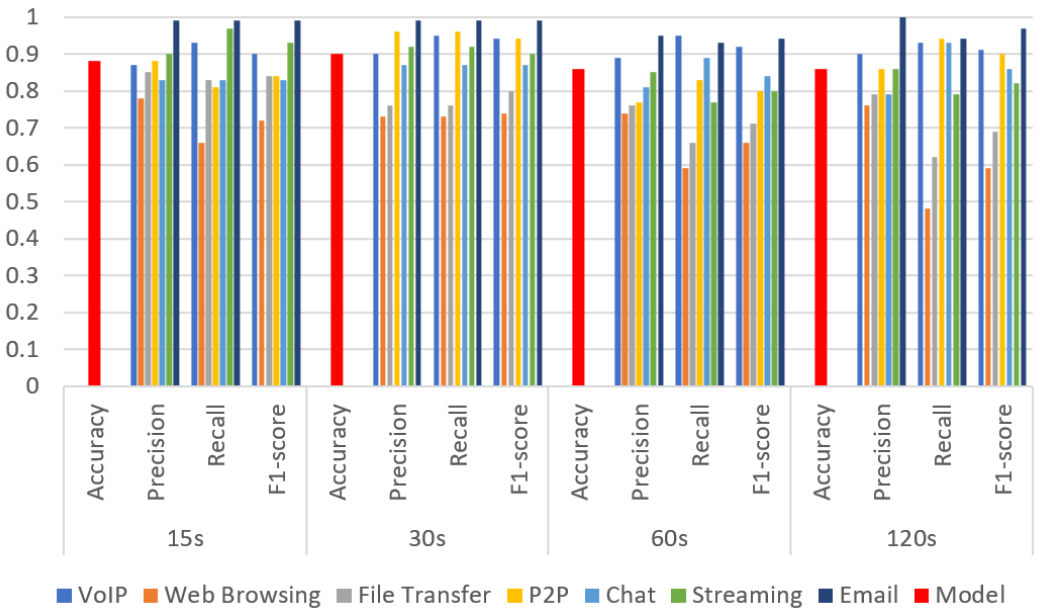
Về chỉ số đo lường Accuracy, tất cả các nhóm dữ liệu theo thời gian đều cho kết quả rất cao trên 90% khi so sánh với 2 mô hình phân loại trên, lần lượt là 93, 91, 92 và 93% cho các nhóm dữ liệu 15, 30, 60 và 120s. Tương tự, chỉ số Precision cũng cho thấy kết quả phân loại tương đối cao cho cả 7 lớp dữ liệu khi phân bổ đều từ 80 – 100%. Trong đó, lớp dữ liệu Email thuộc nhóm phù hợp nhất khi đạt kết quả phân loại tuyệt đối là 100% trong cả 4 nhóm dữ liệu thời gian. Hai lớp dữ liệu có kết quả ổn gần kề là VoIP và Chat khi phân bố kết quả trong khoảng từ 93-98% trong suốt tập dữ liệu. Lớp dữ liệu cho kết quả thấp nhất trong 3 trên 4 nhóm là Streaming, với kết quả phân loại chính xác giảm dần theo sự gia tăng của lượng thời gian ghi nhận thông tin, từ 91% ở 15s xuống đạt mức chỉ còn 80% tập 120s, thấp nhất trong toàn bộ bảng dữ liệu Precision.



Hình 4.13. Biểu đồ chỉ số đánh giá không mã hóa Non-VPN bằng RF – Bối cảnh A2

Ở nhóm chỉ số Recall, Email vẫn đạt vị trí có khả năng phân loại tốt nhất song song với lớp VoIP và Chat, duy trì ở mức 93-99%, cao hơn hẳn so với 4 lớp dữ liệu khác. Ngược lại, kết quả ghi nhận của lớp Streaming đứng cuối và gần như tách biệt hẳn so với các nhóm dữ liệu còn lại luôn đạt mức trên 80%. Trong chỉ số Recall này,

kết quả của lớp Streaming lần lượt là 69, 80, 67 và 61% trên cả 4 nhóm dữ liệu. Điều này ảnh hưởng trực tiếp đến giá trị F1-score của lớp Streaming, đưa đến kết luận là lớp dữ liệu này không đạt kết quả ghi nhận thấp nhất trong toàn mô hình với giá trị F1-score trung bình chỉ khoảng 74.25%. Các nhóm dữ liệu cho kết quả cao nhất lần lượt là Email, VoIP và Chat khi luôn giữ mức điểm F1-score trên 93% trên các khung thời gian. Xu hướng giá trị độ chính xác khi phân loại có chiều hướng giảm từ 15s đến 120s.



Hình 4.14. Biểu đồ chỉ số đánh giá mã hóa VPN bằng RF - Bối cảnh A2

Nhìn chung, chỉ số độ chính xác Accuracy có sự giảm nhẹ khi ghi nhận kết quả cho nhóm dữ liệu mã hóa VPN so với nhóm dữ liệu Non-VPN, đạt mức 86-88% và chỉ duy nhất đạt mức 90% tại nhóm dữ liệu thời gian ghi nhận 30s. Với chỉ số Precision, có một sự sụt giảm đáng kể trong các nhóm dữ liệu Web Browsing và File Transfer, tách biệt hẳn với các lớp dữ liệu khác chứ không giữ mức ổn định gần nhau như trên bối cảnh Non-VPN. Giá trị trung bình của các nhóm qua các khung thời gian cũng có xu hướng giảm theo chiều tăng của nhóm thời gian ghi nhận là 60s và 120s. Các lớp dữ liệu có kết quả thấp hơn 80% được ghi nhận thuộc 3 lớp Web Browsing, File Transfer và P2P. Trong đó, thấp nhất thuộc nhóm 60s với kết quả 74, 76, 77% tương ứng với 3 lớp dữ liệu trên.

Chỉ số Recall đánh dấu sự sụt giảm rõ rệt trong khả năng phân loại lớp Web Browsing, thấp nhất trong toàn bộ 7 lớp dữ liệu, ghi nhận ở mức 66, 73, 59 và 48% ứng với các nhóm thời gian 15, 30, 60 và 120s. Ngược lại, lớp Email luôn luôn giữ vị trí đứng đầu hoàn toàn so với các lớp dữ liệu cần phân loại trong cả 2 chỉ số Recall và F1-score. Hai lớp dữ liệu kế tiếp có độ chính xác cao được kể đến là VoIP và Streaming trong chỉ số F1-score, ghi nhận các giá trị trung bình trong khoảng hơn 90% ở 2 khung thời gian 15-30s. Trong khung thời gian 60-120s, chỉ số F1-score đứng thứ 2 thuộc về các lớp VoIP và P2P, với kết quả nằm trong giai đoạn 85-91%. Trong toàn bộ tập dữ liệu, kết quả tổng hợp của Web Browsing là thấp nhất trong 7 lớp dữ liệu, ghi nhận vấn đề cần khắc phục trong việc xây dựng các mô hình phân loại trong tương lai.

4.3.5 Kết quả tổng quan từ 3 mô hình

Bảng 4.1 đưa ra các chỉ số đánh giá kết quả phân loại tổng hợp của 3 mô hình huấn luyện KNN, ANN và RF. Trong 3 nhóm mô hình, giá trị cao nhất được đánh dấu đỏ.

Bảng 4.1. Kết quả tổng hợp phân loại mạng Internet của 3 mô hình KNN, ANN và RF – Bối cảnh A1

	Time	15s				30s			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
Scenario A1	KNN	84.86%	84.97%	86.68%	85.82%	82.58%	83.06%	84%	83.53%
	ANN	77.06%	77.06%	77.06%	77.06%	77.43%	77.43%	77.43%	77.46%
	RF	91.61%	90.60%	93.84%	92.19%	89.92%	88.01%	93.54$	90.70%
	Time	60s				120s			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
	KNN	82.02%	82.14%	76.33%	79.13%	82.44%	80.60%	86.79%	83.58%
	ANN	77.77%	77.77%	77.77%	77.77%	76.17%	76.17%	76.17%	76.38%
	RF	89.45%	91.02%	84.99%	87.90%	88.87%	88.14%	90.77%	89.44%

Dựa vào bảng tổng kết trên, có thể thấy được trong bối cảnh A1, nhìn chung mô hình Rừng ngẫu nhiên RF cho kết quả phân loại cao nhất, tiếp đó là mô hình KNN và thấp nhất là ANN. Xuyên suốt tất cả chỉ số Accuracy, Precision, Recall và F1-score trong 4 nhóm dữ liệu thời gian, mô hình Rừng ngẫu nhiên luôn cho giá trị từ 88-93%, là một trong những kết quả cao nhất cho tập dữ liệu này. Tương tự, mô hình KNN cung cấp kết quả tương đối với giá trị vào khoảng 80-86%, với 2 giá trị Recall

và F1-score 76.33% và 79.13%. Thấp nhất trong nhóm bối cảnh này là mô hình ANN khi phần lớn giá trị đánh giá nằm trong khoảng 77%.

Bảng 4.2. Kết quả tổng hợp phân loại mạng Internet của 3 mô hình KNN, ANN và RF – bối cảnh A2 - NonVPN

Scenario A2 Non-VPN	Time	15s				30s			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
	KNN	87.81%	71.02%	65.66%	68.24%	85.55%	73.86%	61.08%	66.86%
	ANN	85.91%	87.27%	84.83%	86.16%	83.62%	85.80%	82.66%	84.17%
	RF	93.46%	88.35%	81.48%	84.78%	92.10%	86.31%	79.23%	82.62%
	Time	60s				120s			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
	KNN	84.77%	62.67%	60.99%	61.82%	92.76%	82.69%	66.15%	73.50%
	ANN	81.47%	85.32%	76.34%	80.44%	86.16%	87.45%	84.73%	86.11%
	RF	90.87%	86.38%	79.31%	82.70%	92.76%	82.69%	66.15%	73.50%

Ngược lại trong bối cảnh A2, các giá trị cao nhất được phân chia cho 2 mô hình ANN và RF. Trong tập dữ liệu Non-VPN, mô hình RF chiếm giá trị cao nhất đa phần trong các chỉ số Accuracy và Precision. Cụ thể hơn, chỉ số Accuracy qua các nhóm dữ liệu thời gian khác nhau đều lớn hơn 90%, và chỉ số Precision cũng đạt giá trị từ 86-88%. Còn lại trong mô hình ANN, nhóm chỉ số đạt giá trị cao nhất là Recall trong khoảng 82-84%, với ngoại lệ duy nhất trong nhóm dữ liệu 60s thuộc mô hình RF có chỉ số Recall trung bình là 79.31%. Tương tự, chỉ số F1-score đều mang giá trị cao trên 80% trên tất cả các nhóm dữ liệu thời gian ghi nhận thông tin.

Ngoài ra, trong nhóm dữ liệu mã hóa VPN. Chỉ số Accuracy đạt mức cao nhất trong mô hình huấn luyện RF trong tất cả các nhóm dữ liệu thời gian khác nhau, giá trị dao động từ thấp nhất 86.35% đến cao nhất 90.22%. Trường hợp ngoại lệ duy nhất thuộc mô hình huấn luyện KNN đạt giá trị cao nhất 88.29% trong tập dữ liệu thời gian 120s. Còn lại trong các chỉ số Precision, Recall và F1-score, mô hình huấn luyện ANN đạt giá trị cao nhất trong khoảng 80-83% ở hầu hết các nhóm dữ liệu thời gian. Một lần nữa, tập thời gian 120s tuy giá trị 3 chỉ số trên của mô hình ANN tuy cũng đạt giá trị cao nhất, nhưng chỉ đạt mức dao động cụ thể là 78.38%, 74.44%, và 76.29% ứng với các chỉ số trên.

Bảng 4.3. Kết quả tổng hợp phân loại mạng Internet của 3 mô hình KNN, ANN và RF – bối cảnh A2 - VPN

Scenario A2 VPN	Time	15s				30s			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
	KNN	83.05%	61.98%	62.67%	62.33%	83.28%	67.11%	59.53%	63.09%
	ANN	82.30%	83.88%	80.43%	82.09%	83.02%	85.38%	81.30%	83.24%
	RF	87.99%	77.63%	66.48%	71.62%	90.22%	76.15%	72.81%	74.44%
	Time	60s				120s			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
	KNN	82.75%	63.93%	54.93%	59.09%	88.29%	61.45%	68.92%	64.97%
	ANN	81.16%	83.92%	78.23%	81.23%	75.74%	78.38%	74.44%	76.29%
	RF	86.35%	73.95%	59.06%	65.67%	86.45%	76.06%	48.21%	59.02%

Như vậy, có thể kết luận mô hình Rừng ngẫu nhiên RF phù hợp nhất với mục tiêu phân loại Internet theo thứ tự bối cảnh A1 và A2. Kết quả phân loại của mô hình ANN cũng đồng thời cung cấp những kết quả phân loại có giá trị đáng kể. Từ những đặc trưng trên của cả 3 mô hình, có thể nhận thấy mô hình RF mang những ưu điểm có thể xử lý những mặt hạn chế của các mô hình ANN hoặc KNN như : giảm thiểu hiện tượng quá khớp có trong các mô hình huấn luyện như cây quyết định hay ANN, giảm độ phương sai và nâng cao hiệu suất phân loại, tự động xử lý vấn đề mất mát dữ liệu,độ ổn định cao, xử lý với hiệu suất cao những dữ liệu bị nhiễu, không đồng bộ với tập dữ liệu hoặc không tuyến tính và phù hợp với yêu cầu phân loại của đề tài. Tuy nhiên, ANN và KNN cũng mang những ưu điểm riêng. Đồng thời, tập dữ liệu được chọn có thể mang những đặc trưng phù hợp với một trong hai mô hình ANN hoặc KNN, từ đó đạt được hiệu suất phân loai cao hơn dự kiến.

Do đó, tuy RF mang những đặc diểm có ưu thế so với 2 mô hình trên, đề tài vẫn cần phải áp dụng cả 3 mô hình huấn luyện, sau đó dựa trên kết quả phân loại và đánh giá phân tích, từ đó mới đưa ra kết quả cho đề tài về việc mô hình RF là phù hợp nhất cho vấn đề phân loại lưu lượng mạng Internet sử dụng tập dữ liệu được chọn.

KẾT LUẬN

1. Kết quả đạt được

Trong phạm vi nghiên cứu của đề tài này, các mô hình huấn luyện học máy khác nhau đã được đề xuất nhằm mục tiêu phân loại lưu lượng mạng Internet sử dụng tập dữ liệu mở ISCXVPN2016, bao gồm mô hình huấn luyện K – lân cận, Mạng Neuron nhân tạo và Rừng ngẫu nhiên. Trong đó, dựa trên các chỉ số đánh giá ghi nhận được cho mỗi mô hình huấn luyện cho 2 bối cảnh A1 và A2, Rừng ngẫu nhiên là mô hình đạt được kết quả phân loại cao và phù hợp nhất so với 2 mô hình huấn luyện còn lại. Nhìn chung, việc phân loại lưu lượng được mã hóa khó hơn nhiều so với việc phân loại lưu lượng không được mã hóa, chủ yếu vì các chuyên gia phân tích không thể thực hiện phương pháp phân tích gói chuyên sâu.

Điều này cũng được phản ánh rõ ràng trong kết quả phân loại lưu lượng Internet bối cảnh A2. Các chỉ số đánh giá kết quả phân chia đều cho các mô hình phân loại Rừng ngẫu nhiên và Mạng Neuron nhân tạo, thậm chí ở chỉ số phân loại F1-score kết quả của mô hình Rừng ngẫu nhiên tỏ ra yếu kém hơn khá nhiều so với mô hình mạng neuron. Tuy nhiên, khi cân nhắc toàn bộ bối cảnh thí nghiệm, xét từ kết quả bối cảnh A1 nối tiếp sau đó là bối cảnh A2, Rừng ngẫu nhiên có thể được xem là mô hình phù hợp nhất cho bài toán phân loại lưu lượng Internet trong phạm vi đề tài này. Khi kết quả phân loại lưu lượng Internet được áp dụng với kết quả thành công cao, những chuyên gia an ninh mạng có thể xây dựng các chính sách bảo mật tiến hành tự động ngăn chặn khi phát hiện loại có sự truy cập của lưu lượng không mong muốn.

2. Phương hướng nghiên cứu

Với sự gia tăng tỷ lệ sử dụng internet, các ứng dụng mới và giao thức mới đang xuất hiện. Các ứng dụng mã hóa lưu lượng truy cập qua internet bằng các phương pháp và giao thức mã hóa để đảm bảo giao tiếp bí mật và an toàn. Trong các nghiên cứu trong tương lai, một tập dữ liệu mới có thể được thu thập để đánh giá các

ứng dụng và giao thức mới, qua đó tiến hành thực hiện phân loại lưu lượng Internet. Khi các nghiên cứu trong lĩnh vực này tiếp tục được mở rộng với càng nhiều mô hình huấn luyện được thiết kế phù hợp hơn cho bài toán phân loại, phương hướng nghiên cứu tiếp theo là áp dụng các mô hình huấn luyện phức tạp hơn với khả năng tinh chỉnh siêu tham số. Từ đó, những mô hình phù hợp nhất cho bài toán này có thể được phát hiện và ứng dụng phù hợp với nhu cầu nghiên cứu và phát triển.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, and S. Wolff, "A Brief History of the Internet," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 5, pp. 22–31, Oct. 2009.

[2] O. Salman, I. Elhajj, A. Chehab, and A. Kayssi, "IoT survey: An SDN and fog computing perspective," *Computer Networks*, vol. 143, 2018.

[3] L. Stewart, G. Armitage, P. Branch, and S. Zander, "An Architecture for Automated Network Control of QoS over Consumer Broadband Links," 2005, pp. 1–6.

[4] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, pp. 2451–2455.

[5] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Challenges in Network Application Identification," 2012.

[6] A. Razaghpanah, A. Akhavan Niaki, N. Vallina-Rodriguez, S. Sundaresan, J. Amann, and P. Gill, "Studying TLS Usage in Android Apps," 2017, pp. 350–362.

[7] B. Park, Y. Won, J. Chung, M. Kim, and J. W.-K. Hong, "Fine-grained traffic classification based on functional separation," *International Journal of Network Management*, vol. 23, no. 5, pp. 350–381, 2013.

[8] G. Aceto, A. Dainotti, W. Donato, and A. Pescapè, "PortLoad: Taking the Best of Two Worlds in Traffic Classification," 2010, pp. 1–5.

[9] D. Qin, J. Yang, J. Wang, and B. Zhang, "IP traffic classification based on machine learning," 2011.

[10] J. Dromard, P. Owezarski, V. Mozo, A. Ordozgoiti, and B. Vakaruk, "Delivrable Algorithms Description: Traffic pattern evolution and unsupervised network anomaly detectionONTIC D4.2," 2016.

[11] N. Namdev, S. Agrawal, and S. Silkari, “Recent Advancement in Machine Learning Based Internet Traffic Classification,” *Procedia Computer Science*, vol. 60, pp. 784–791, 2015.

[12] k. claffy, “Internet traffic characterization,” UC San Diego, 1994.

[13] V. Paxson, “Empirically derived analytic models of wide-area TCP connections,” *IEEE/ACM Transactions on Networking*, vol. 2, no. 4, pp. 316–336, 1994.

[14] C. Dewes, A. Wichmann, and A. Feldmann, “An analysis of Internet chat systems,” *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, 2003.

[15] Z. Yuan and C. Wang, “An improved network traffic classification algorithm based on Hadoop decision tree,” *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*. pp. 53–56, 2016.

[16] Y. Ma, Z. Qian, G. Shou, and Y. Hu, “Study of Information Network Traffic Identification Based on C4.5 Algorithm,” in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, 2008, pp. 1–5.

[17] M. Dixit, R. Sharma, S. Shaikh, and K. Muley, “Internet Traffic Detection using Naïve Bayes and K-Nearest Neighbors (KNN) algorithm,” in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 1153–1157.

[18] L. Zhipeng, Z. Qin, K. Huang, X. Yang, and S. Ye, “Intrusion Detection Using Convolutional Neural Networks for Representation Learning,” 2017, pp. 858–866.

[19] I. Witten and I. H. Frank, *Data Mining - Practical Machine Learning Tools and Techniques with JAVA Implementations*, vol. 31. Morgan Kaufmann Publishers, 2005.

[20] I. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*, 4th Ed. Morgan Kaufmann Publishers, 2016.

[21] L. Jun, Z. Shunyi, L. Yanqing, and Z. Zailong, “Internet Traffic Classification Using Machine Learning,” in *2007 Second International Conference on Communications and Networking in China*, 2007, pp. 239–243.

[22] A. Moldagulova and R. B. Sulaiman, “Using KNN algorithm for classification of textual documents,” in *2017 8th International Conference on Information Technology (ICIT)*, 2017, pp. 665–671.

[23] W. Zhou, L. Dong, L. Bic, M. Zhou, and L. Chen, “Internet traffic classification using feed-forward neural network,” in *2011 International Conference on Computational Problem-Solving (ICCP)*, 2011, pp. 641–646.

[24] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, “Characterization of Encrypted and VPN Traffic using Time-related Features,” in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP2016)*, 2016, pp. 407–414.

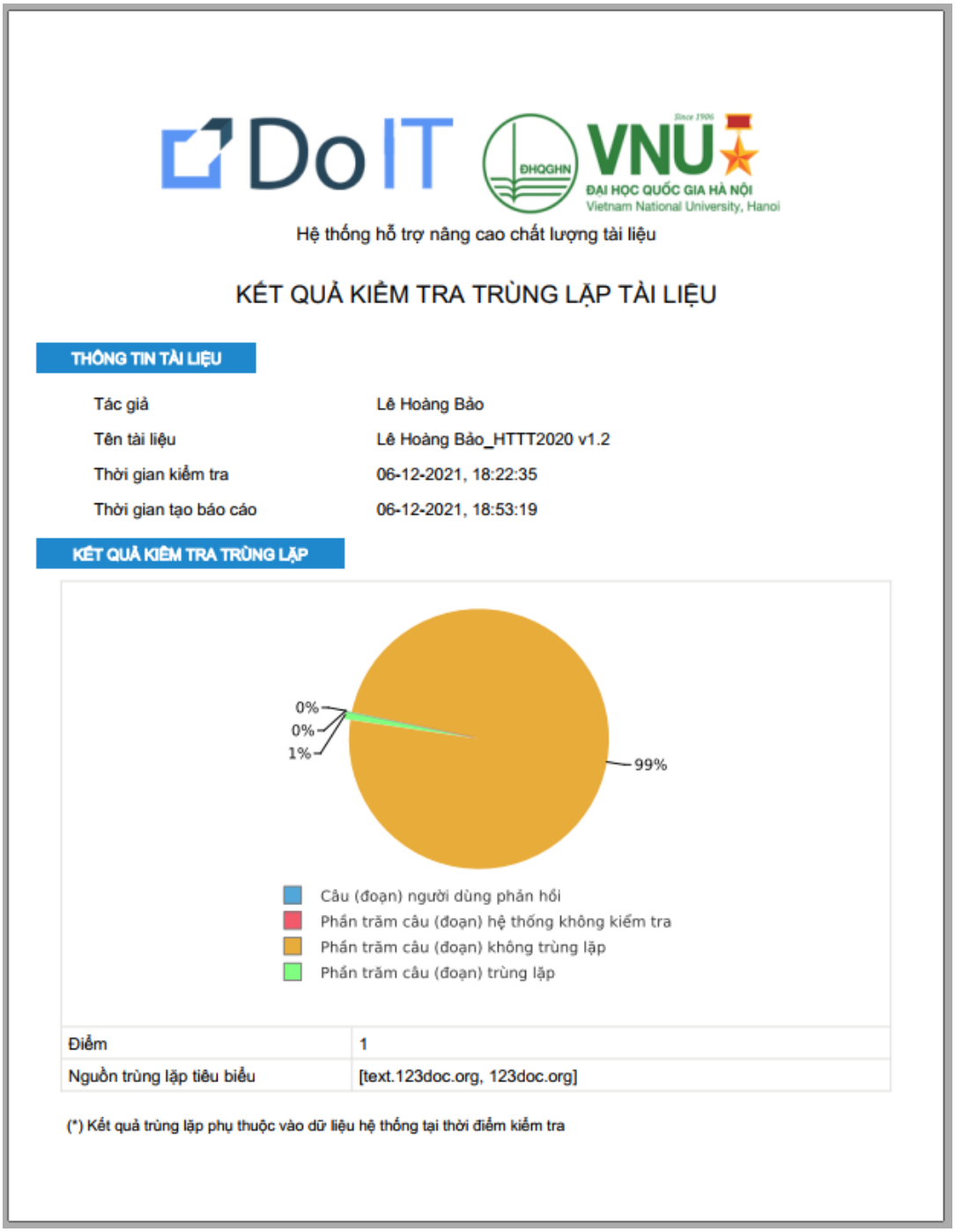
BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung luận văn qua phần mềm DoIT một cách trung thực và đạt kết quả mức độ tương đồng 1% toàn bộ nôi dung luận văn . Bản luận văn kiểm tra qua phần mềm là bản cứng luận văn đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học Viện.

TP,HCM ngày 25 tháng 01 năm 2022

HỌC VIÊN CAO HỌC

Lê Hoàng Bảo



Hệ thống hỗ trợ nâng cao chất lượng tài liệu

KẾT QUẢ KIỂM TRA TRÙNG LẶP TÀI LIỆU

THÔNG TIN TÀI LIỆU

Tác giả	Lê Hoàng Bảo
Tên tài liệu	Lê Hoàng Bảo_HTTT2020 v1.2
Thời gian kiểm tra	06-12-2021, 18:22:35
Thời gian tạo báo cáo	06-12-2021, 18:53:19

KẾT QUẢ KIỂM TRA TRÙNG LẶP

Điểm	1
Nguồn trùng lặp tiêu biểu	[text.123doc.org, 123doc.org]

(*) Kết quả trùng lặp phụ thuộc vào dữ liệu hệ thống tại thời điểm kiểm tra

HỌC VIÊN | **NGƯỜI HƯỚNG DẪN KHOA HỌC**

Lê Hoàng Bảo | **TS.Nguyễn Hồng Sơn**