

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



NGUYỄN THỊ THU HỒNG – 52100962
NGUYỄN THỊ ANH ĐÀO – 52000830

**ỨNG DỤNG MÔ HÌNH HỌC MÁY VÀ
HỌC SÂU TRONG VIỆC PHÁT HIỆN
URL ĐỘC HẠI**

DỰ ÁN CÔNG NGHỆ THÔNG TIN

**MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ
LIỆU**

THÀNH PHỐ HỒ CHÍ MINH, 2025

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



NGUYỄN THỊ THU HỒNG – 52100962

NGUYỄN THỊ ANH ĐÀO – 52000830

ỨNG DỤNG MÔ HÌNH HỌC MÁY VÀ
HỌC SÂU TRONG VIỆC PHÁT HIỆN
URL ĐỘC HẠI

DỰ ÁN CÔNG NGHỆ THÔNG TIN

MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG DỮ LIỆU

Người hướng dẫn

TS. Trương Đình Tú

THÀNH PHỐ HỒ CHÍ MINH, 2025

LỜI CẢM ƠN

Trước tiên, chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô Khoa Công nghệ thông tin Trường Đại học Tôn Đức Thắng đã trang bị cho chúng em kiến thức nền tảng, cùng với những kinh nghiệm quý báu trong suốt quá trình học tập. Đặc biệt, chúng em xin cảm ơn TS. Trương Đình Tú, người đã tận tình hướng dẫn, đóng góp ý kiến để giúp chúng em hoàn thành Dự án Công nghệ thông tin này.

Tuy kiến thức và cách tiếp cận còn giới hạn, nhưng sự hỗ trợ từ quý thầy cô, bạn bè và gia đình đã giúp chúng em có thể nỗ lực hơn và hoàn thành dự án một cách tốt nhất.

Cuối cùng, chúng em kính chúc quý thầy cô, gia đình và bạn bè nhiều sức khỏe và thành công.

TP. Hồ Chí Minh, ngày 20 tháng 02 năm 2025

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Thị Thu Hồng

Nguyễn Thị Anh Đào

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của TS. Trương Đình Tú. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 20 tháng 02 năm 2025

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Thị Thu Hồng

Nguyễn Thị Anh Đào

ỨNG DỤNG MÔ HÌNH HỌC MÁY VÀ HỌC SÂU TRONG VIỆC PHÁT HIỆN URL ĐỘC HẠI

TÓM TẮT

Giữa sự bùng nổ nhanh chóng của hàng ngàn trang web mới mỗi ngày, việc phân biệt các trang web an toàn với những trang có thể gây hại ngày càng trở nên phức tạp. Những trang web này thường thu thập dữ liệu người dùng và nếu không có các biện pháp an ninh mạng đầy đủ như phát hiện và phân loại hiệu quả các URL độc hại, thông tin nhạy cảm của người dùng có thể bị xâm phạm và sử dụng với mục đích xấu.

Chính vì vậy, trong khuôn khổ dự án này, chúng tôi cung cấp một so sánh trực tiếp giữa các mô hình học sâu phổ biến (Multi-Layer Perceptron, Long Short-Term Memory) với các thuật toán học máy (Logistic Regression, Random Forest, XGBoost) trong việc phát hiện và phân loại các URL độc hại. Bên cạnh đó, chúng tôi mở rộng tài liệu bằng cách sử dụng bộ dữ liệu công khai ISCX-URL-2016, chứa các đặc trưng từ vựng của các URL độc hại. Chúng tôi đã phân tích và chia bộ dữ liệu thành các bộ khác nhau để có cái nhìn chi tiết và khách quan hơn. Bộ dữ liệu ISCX-URL-2016 sẽ được thảo luận chi tiết trước khi trình bày và giải thích các thí nghiệm và kết quả. Cuối cùng, dự án sẽ kết luận và đề xuất hướng nghiên cứu tiếp theo.

APPLICATION OF MACHINE LEARNING AND DEEP LEARNING MODELS IN DETECTING MALICE URLS

ABSTRACT

Amid the rapid proliferation of thousands of new websites each day, distinguishing safe websites from potentially harmful ones has become increasingly complex. These websites often collect user data, and without adequate cybersecurity measures such as effective detection and classification of malicious URLs, sensitive user information can be compromised and misused.

Therefore, within the scope of this project, we provide a direct comparison between popular deep learning models (Multi-Layer Perceptron, Long Short-Term Memory) and machine learning algorithms (Logistic Regression, Random Forest, XGBoost) in detecting and classifying malicious URLs. Additionally, we extend the literature by utilizing the publicly available ISCX-URL-2016 dataset, which contains lexical features of malicious URLs. We analyzed and partition the dataset into different subsets to gain a more detailed and objective perspective. The ISCX-URL-2016 dataset will be discussed in detail before presenting and explaining the experiments and results. Finally, the project will conclude with recommendations for future research directions.

Mục lục	
LỜI CẢM ƠN	2
TÓM TẮT	4
ABSTRACT	5
MỤC LỤC	6
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT	9
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ	10
CHƯƠNG 1 - TỔNG QUAN ĐỀ TÀI	13
1.1 Giới thiệu	13
1.2 Tính cấp thiết của đề tài	14
1.3 Mục tiêu nghiên cứu	14
1.4 Đối tượng và phạm vi nghiên cứu	15
1.5 Phương pháp nghiên cứu	15
1.6 Các nghiên cứu liên quan	16
CHƯƠNG 2 - CƠ SỞ LÝ THUYẾT	18
2.1 URLs	18
2.1.1 Scheme (Giao thức))	18
2.1.2 Authority (Thẩm quyền)	19
2.1.3 Path (Đường dẫn)	19
2.1.4 Query (Truy vấn)	20
2.1.5 Fragment (Phân đoạn)	20
2.2 URLs độc hại	20
2.3 Các loại URL độc hại	21

2.3.1 Phishing URL (URL lừa đảo)	21
2.3.2 Malware Distribution URL (URL phát tán mã độc)	21
2.3.3 Command and Control (CC) Server URL	22
2.3.4 URL chứa các lỗ hổng tấn công	22
2.3.5 URL sử dụng Shortened Links (Liên kết rút gọn)	22
2.4 Học máy (Machine Learning)	23
2.4.1 Khái niệm	23
2.4.2 Lịch sử và phát triển	23
2.4.3 Phân loại học máy	26
2.4.4 Một số mô hình học máy sử dụng	33
2.5 Học sâu (Deep Learning)	39
2.5.1 Khái niệm	39
2.5.2 Cấu trúc mạng Nơ-ron nhân tạo	40
2.5.3 Một số mô hình học sâu	40
2.6 Các kỹ thuật lựa chọn đặc trưng	44
2.6.1 Random Forest (Tree-based Feature Importance)	44
2.6.2 Mutual Information	45
2.6.3 Recursive Feature Elimination (RFE)	46
2.6.4 SelectKBest	47
2.6.5 So sánh các mô hình	48
2.6.6 Lợi ích khi kết hợp sử dụng bốn mô hình	50
2.6.7 Chọn đặc trưng dựa trên phiếu bầu (Voting-based Feature Selection)	51
CHƯƠNG 3 - NỘI DUNG NGHIÊN CỨU	54
3.1 Bộ dữ liệu	54

3.2 Khai phá bộ dữ liệu	55
3.2.1 Phân tích đặc trưng của URL	56
3.2.2 Trực quan hóa dữ liệu	59
3.3 Lựa chọn đặc trưng	62
3.3.1 Random Forest (Tree-based Feature Importance)	62
3.3.2 Mutual Information	63
3.3.3 Recursive Feature Elimination (RFE)	64
3.3.4 SelectKBest	64
3.3.5 Kết hợp 4 phương pháp	64
3.4 Huấn luyện mô hình	66
3.4.1 Huấn luyện với bộ dữ liệu gốc	67
3.4.2 Huấn luyện với bộ dữ liệu 1	69
3.4.3 Huấn luyện với bộ dữ liệu 2	71
3.4.4 Huấn luyện với bộ dữ liệu 3	72
3.4.5 Huấn luyện với bộ dữ liệu 4	74
3.4.6 Huấn luyện với bộ dữ liệu 5	76
3.4.7 Huấn luyện với bộ dữ liệu 6	77
CHƯƠNG 4 - KẾT LUẬN VÀ KIẾN NGHỊ	80
4.1 Kết luận	80
4.2 Ưu điểm	81
4.3 Nhược điểm	82
4.4 Hướng phát triển	83
TÀI LIỆU THAM KHẢO	84

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

σ - sigma

β - beta

CÁC CHỮ VIẾT TẮT

URL - Uniform Resource Locator

MLP - Multi-Layer Perceptron

LSTM - Long Short-Term Memory

KDN - Knowledge Discovery in Databases

CNN - Convolutional Neural Network

HTTP - HyperText Transfer Protocol

HTTPS - HyperText Transfer Protocol Secure

FTP - File Transfer Protocol

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

1	Phương pháp phát hiện và phân loại URL độc hại	14
2	Thành phần của URL	18
3	Lịch sử và phát triển của học máy	23
4	Phân loại học máy	26
5	Học có giám sát (Supervised Learning)	26
6	Học không giám sát (Unsupervised Learning)	28
7	Học bán giám sát (Semi-Supervised Learning)	30
8	Học tự giám sát (Self-Supervised Learning)	31
9	Logistic Regression	33
10	Random Forest	35
11	XGBoost	37
12	Học sâu (Deep Learning)	39
13	Multi-Layer Perceptron	41
14	Long Short-Term Memory (LSTM)	43
15	Phân phối nhân trong tập dữ liệu	55
16	Ma trận tương quan	60
17	Phân phối KDE của Entropy_URL, pathLength	61
18	Phân phối các đặc trưng quan trọng	62
19	Lựa chọn đặc trưng bằng Random Forest	63
20	Lựa chọn đặc trưng bằng Mutual Information	63
21	Lựa chọn đặc trưng bằng Recursive Feature Elimination	64
22	Lựa chọn đặc trưng bằng SelectKBest	65

DANH MỤC BẢNG

1	So sánh các phân loại của học máy	32
2	So sánh các mô hình chọn đặc trưng	49
3	Ví dụ phân tích đặc trưng URL (1)	58
4	Ví dụ phân tích đặc trưng URL (2)	59
5	Huấn luyện bộ dữ liệu gốc với Logistic Regression	67
6	Huấn luyện bộ dữ liệu gốc với Random Forest	67
7	Huấn luyện bộ dữ liệu gốc với XGBoost	68
8	Huấn luyện bộ dữ liệu gốc với MLP	68
9	Huấn luyện bộ dữ liệu gốc với LSTM	68
10	Huấn luyện bộ dữ liệu 1 với Logistic Regression	69
11	Huấn luyện bộ dữ liệu 1 với Random Forest	70
12	Huấn luyện bộ dữ liệu 1 với XGBoost	70
13	Huấn luyện bộ dữ liệu 1 với MLP	70
14	Huấn luyện bộ dữ liệu 1 với LSTM	71
15	Huấn luyện bộ dữ liệu 2 với Logistic Regression	71
16	Huấn luyện bộ dữ liệu 2 với Random Forest	71
17	Huấn luyện bộ dữ liệu 2 với XGBoost	72
18	Huấn luyện bộ dữ liệu 2 với MLP	72
19	Huấn luyện bộ dữ liệu 2 với LSTM	72
20	Huấn luyện bộ dữ liệu 3 với Logistic Regression	73
21	Huấn luyện bộ dữ liệu 3 với Random Forest	73
22	Huấn luyện bộ dữ liệu 3 với XGBoost	73
23	Huấn luyện bộ dữ liệu 3 với MLP	74
24	Huấn luyện bộ dữ liệu 3 với LSTM	74

25	Huấn luyện bộ dữ liệu 4 với Logistic Regression	74
26	Huấn luyện bộ dữ liệu 4 với Random Forest	75
27	Huấn luyện bộ dữ liệu 4 với XGBoost	75
28	Huấn luyện bộ dữ liệu 4 với MLP	75
29	Huấn luyện bộ dữ liệu 4 với LSTM	76
30	Huấn luyện bộ dữ liệu 5 với Logistic Regression	76
31	Huấn luyện bộ dữ liệu 5 với Random Forest	76
32	Huấn luyện bộ dữ liệu 5 với XGBoost	77
33	Huấn luyện bộ dữ liệu 5 với MLP	77
34	Huấn luyện bộ dữ liệu 5 với LSTM	77
35	Huấn luyện bộ dữ liệu 6 với Logistic Regression	78
36	Huấn luyện bộ dữ liệu 6 với Random Forest	78
37	Huấn luyện bộ dữ liệu 6 với XGBoost	78
38	Huấn luyện bộ dữ liệu 6 với MLP	79
39	Huấn luyện bộ dữ liệu 6 với LSTM	79

CHƯƠNG 1 - TỔNG QUAN ĐỀ TÀI

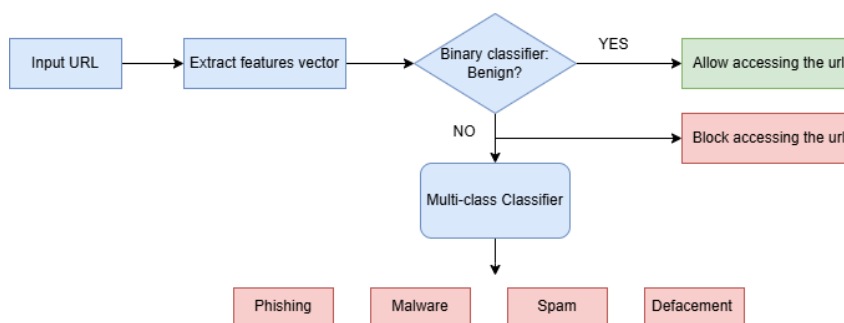
Chương này trình bày tổng quan về đề tài nghiên cứu, trình bày tính cấp thiết của đề tài, phương pháp nghiên cứu, phạm vi nghiên cứu, các đối tượng nghiên cứu và các nghiên cứu có liên quan.

1.1 Giới thiệu

Trong thời đại số hóa hiện nay, Internet đã trở thành một phần không thể thiếu trong đời sống cá nhân và doanh nghiệp. Cùng với sự phát triển của các nền tảng trực tuyến, thương mại điện tử, và mạng xã hội, các cuộc tấn công mạng cũng ngày càng tinh vi hơn. Một trong những hình thức tấn công phổ biến nhất là sử dụng **URL độc hại (Malicious URL)** nhằm đánh cắp thông tin cá nhân, phát tán mã độc hoặc thực hiện các cuộc tấn công có chủ đích. Các URL này có thể xuất hiện trong email lừa đảo (phishing email), quảng cáo độc hại (malvertising), tin nhắn trên mạng xã hội, hoặc các trang web bị xâm nhập.

Hiện nay, các phương pháp truyền thống như danh sách đen (blacklist) hoặc phân tích theo quy tắc (rule-based detection) không còn đủ hiệu quả trước sự gia tăng và biến đổi liên tục của các kỹ thuật tấn công mạng. Do đó, việc nghiên cứu và phát triển các phương pháp phát hiện URL độc hại dựa trên **học máy (Machine Learning)** và **học sâu (Deep Learning)** đang nhận được sự quan tâm lớn từ cộng đồng nghiên cứu và các tổ chức bảo mật.

Dựa vào các phương pháp học máy và học sâu, chúng ta có thể phát hiện và phân loại các URL lành tính và độc hại. Có thể theo mô tả dưới đây:



Hình 1: Phương pháp phát hiện và phân loại URL độc hại

1.2 Tính cấp thiết của đề tài

Các cuộc tấn công mạng sử dụng URL độc hại không chỉ gây thiệt hại về tài chính mà còn ảnh hưởng nghiêm trọng đến an ninh thông tin cá nhân và doanh nghiệp. Theo báo cáo của các tổ chức an ninh mạng, hàng triệu URL độc hại được tạo ra mỗi ngày, khiến việc phát hiện và ngăn chặn trở thành một thách thức lớn. Đặc biệt, các chiến thuật tấn công như **tấn công lừa đảo (phishing)**, **phát tán mã độc (malware distribution)**, **tấn công điều khiển từ xa (CC server attack)** đang ngày càng tinh vi, vượt qua các phương pháp bảo mật truyền thống.

Do đó, nghiên cứu các phương pháp hiện đại để **phát hiện URL độc hại một cách chính xác, nhanh chóng và hiệu quả** là một vấn đề cấp thiết. Việc ứng dụng các kỹ thuật **học máy, học sâu và xử lý ngôn ngữ tự nhiên (NLP)** trong phát hiện URL độc hại có tiềm năng cải thiện đáng kể độ chính xác so với các phương pháp truyền thống.

1.3 Mục tiêu nghiên cứu

- Nghiên cứu này nhằm đạt được các mục tiêu chính sau:
 - Tổng quan và phân loại các loại URL độc hại, bao gồm URL lừa đảo, URL phát tán mã độc, phishing, ...
 - Ứng dụng các mô hình học máy và học sâu nhằm đề xuất mô hình tốt nhất

cho bài toán phát hiện URL độc hại.

- Đánh giá hiệu quả của các mô hình đề xuất thông qua các bộ dữ liệu thực tế và so sánh với các phương pháp hiện có.

1.4 Đối tượng và phạm vi nghiên cứu

- **Đối tượng nghiên cứu:**

- Các phương pháp phát hiện URL độc hại, bao gồm kỹ thuật truyền thống và hiện đại dựa trên học máy và học sâu.

- **Phạm vi nghiên cứu:**

- Phân tích các đặc trưng của URL độc hại thông qua dữ liệu thực tế.
- Ứng dụng các thuật toán học và học sâu trong phát hiện URL độc hại.
- So sánh hiệu suất giữa các phương pháp khác nhau.

1.5 Phương pháp nghiên cứu

- Trong khuôn khổ dự án, nghiên cứu này sử dụng các phương pháp sau:

- **Phương pháp tổng hợp tài liệu:** Thu thập và phân tích các nghiên cứu trước đây về phát hiện URL độc hại.
- **Phương pháp phân tích dữ liệu:** Xây dựng bộ dữ liệu từ các nguồn công khai như OpenPhish, PhishTank, và sử dụng các kỹ thuật tiền xử lý dữ liệu.
- **Phương pháp thực nghiệm:** Triển khai, huấn luyện và đánh giá các mô hình học máy và học sâu trên tập dữ liệu thu thập được.
- **Phương pháp đánh giá:** Sử dụng các chỉ số như độ chính xác (Accuracy), độ nhạy (Recall), độ đặc hiệu (Precision), và F1-score để đo lường

hiệu suất mô hình.

1.6 Các nghiên cứu liên quan

Choi và cộng sự đã tiến hành thí nghiệm với việc phát hiện và phân loại các URL độc hại bằng cách sử dụng SVM cho phân loại nhị phân và các bộ phân loại C4.5, Label Powerset, và kNN cho phân loại đa lớp. Các đặc trưng được sử dụng cho hệ thống này bao gồm các đặc điểm từ vựng, cấu trúc liên kết, DNS, lưu lượng mạng, và thông tin thành phần nội dung. Việc phát hiện lưu lượng độc hại đã cho thấy hiệu suất cao với độ chính xác của SVM vượt quá 98%. Công trình này được cho là báo cáo đầu tiên về phân loại các URL độc hại trong các danh mục lừa đảo, spam, và phần mềm độc hại. Các mô hình của họ cho phân loại đa lớp đạt độ chính xác trên 93%. Ngoài ra, bài báo còn nêu rằng do các chi tiết mà mô hình đã xem xét, các kỹ thuật che giấu như chuyển hướng, thao tác liên kết, và lưu trữ nhanh sẽ làm giảm hiệu quả trong việc phát hiện các URL độc hại.

Doyen Sahoo và cộng sự đã cung cấp một khảo sát và hiểu biết cấu trúc về các kỹ thuật Phát hiện URL độc hại sử dụng học máy. Tác giả trình bày công thức chính thức của Phát hiện URL độc hại như một nhiệm vụ của ML, phân loại cũng như đánh giá các đóng góp của các nghiên cứu trong tài liệu liên quan đến các khía cạnh khác nhau của vấn đề này. Bài viết cung cấp một khảo sát cho nhiều đối tượng khác nhau.

Ucar và cộng sự đã áp dụng hai mô hình học sâu, Long Short-Term Memory (LSTM) và Convolutional Neural Network (CNN), trong việc phát hiện và phân loại các URL độc hại. Việc phát hiện các URL độc hại cho thấy hiệu suất cao, với độ chính xác lần lượt là 97.25% và 98.86% cho các mô hình LSTM và CNN. Mô hình CNN đạt độ chính xác 95.37% trong việc phân loại các URL độc hại, cao

hơn so với 91.13% của LSTM. Công trình này có ý nghĩa quan trọng vì hai lý do. Thứ nhất, có rất ít ứng dụng của học sâu trong vấn đề nghiên cứu này và thứ hai, ít công trình đã sử dụng bộ dữ liệu ISCX-URL-2016.

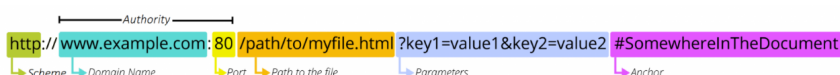
CHƯƠNG 2 - CƠ SỞ LÝ THUYẾT

Mục tiêu của chương "Cơ sở lý thuyết" là cung cấp nền tảng kiến thức cần thiết để hiểu rõ các khái niệm, lý thuyết và công nghệ liên quan đến nghiên cứu. Chương này sẽ trình bày những nguyên lý cơ bản, từ đó xây dựng cơ sở cho việc phát triển và triển khai các phương pháp, mô hình trong nghiên cứu.

2.1 URLs

URL (Uniform Resource Locator) là một chuỗi ký tự có cấu trúc, được sử dụng để xác định vị trí của một trang web trên Internet. Mỗi URL là một địa chỉ duy nhất dẫn đến một trang web, hình ảnh, video hoặc tài nguyên khác trên mạng. Mỗi URL được tạo từ các thành phần khác nhau, ảnh hưởng đến khả năng bảo mật và khả năng tối ưu hóa công cụ tìm kiếm của trang web đó.

Một URL thường có các thành phần chính sau:



Hình 2: Thành phần của URL
(Nguồn: <https://oriagency.vn/url-la-gi/>)

2.1.1 Scheme (Giao thức)

- Xác định giao thức được sử dụng để truy cập tài nguyên.
- Một số giao thức phổ biến:
 - http (HyperText Transfer Protocol)
 - https (HTTP Secure – phiên bản bảo mật của HTTP)
 - ftp (File Transfer Protocol)
 - mailto (Gửi email)

- file (Truy cập tệp cục bộ)
- Ví dụ:
 - *https://example.com* – sử dụng giao thức HTTPS
 - *ftp://ftp.example.com* – sử dụng giao thức FTP

2.1.2 Authority (Thẩm quyền)

- Thành phần này gồm **username, password, hostname, và port**.
- Cú pháp: [username:password@]hostname[:port]
 - **Hostname:** Địa chỉ của máy chủ chứa tài nguyên (ví dụ: *www.example.com*).
 - **Port (Cổng):** Xác định cổng mạng được sử dụng để kết nối đến máy chủ (ví dụ: *:8080*).
 - **Username và Password:** ít dùng do vấn đề bảo mật.
- Ví dụ:
 - *https://user:pass@example.com:8080/* → Truy cập example.com qua cổng 8080 với username là user và password là pass.

2.1.3 Path (Đường dẫn)

- Chỉ định vị trí của tài nguyên trên máy chủ.
- Đường dẫn có thể chứa nhiều cấp thư mục.
- Ví dụ:
 - *https://example.com/products/list* → Truy cập thư mục /products/list trên máy chủ example.com.

2.1.4 Query (Truy vấn)

- Chứa các tham số truyền vào tài nguyên.
- Được bắt đầu bằng dấu ? và chứa các cặp key=value, cách nhau bằng .
- Ví dụ:
 - *https://example.com/search?q=URL+theorypage=2*
 - *q=URL+theory* → tìm kiếm "URL theory"
 - *page=2* → chuyển đến trang số 2

2.1.5 Fragment (Phân đoạn)

- Được bắt đầu bằng # và xác định một phần cụ thể của tài nguyên.
- Thường dùng để di chuyển đến một phần trong trang web.
- Ví dụ:
 - *https://example.com/docsection3* → Truy cập section3 trong tài liệu.

2.2 URLs độc hại

- Malicious URL (URL độc hại) là các địa chỉ web được thiết kế để thực hiện hành vi tấn công hoặc gây hại cho người dùng, hệ thống hoặc mạng. Các URL này có thể được sử dụng để:
 - Phát tán phần mềm độc hại (malware) như virus, trojan, ransomware.
 - Lừa đảo (phishing) để đánh cắp thông tin cá nhân, tài khoản ngân hàng.
 - Tấn công mạng (cyber attacks) như SQL Injection, Cross-Site Scripting (XSS), Man-in-the-Middle (MITM).

- Chuyển hướng người dùng (URL redirection) đến các trang web không an toàn.

- Ví dụ:

- *http://example.com/steal-password?user=adminpass=12345*

2.3 Các loại URL độc hại

2.3.1 Phishing URL (URL lừa đảo)

- Được thiết kế để giả mạo các trang web hợp pháp nhằm đánh cắp thông tin đăng nhập, thẻ tín dụng, hoặc dữ liệu nhạy cảm.
- Thường có giao diện giống với các trang web phổ biến như Google, Facebook, ngân hàng trực tuyến.

- Ví dụ:

- *https://bank-login.secure.com/login*

- (URL này có vẻ hợp lệ nhưng thực chất không phải là trang ngân hàng chính thức).

2.3.2 Malware Distribution URL (URL phát tán mã độc)

- Lưu trữ hoặc tự động tải về phần mềm độc hại như virus, trojan, ransomware.
- Thường ẩn trong email spam, quảng cáo giả mạo hoặc tệp đính kèm.

- Ví dụ:

- Arduino

- *http://malware-site.com/download-virus.exe*

- (Tự động tải xuống phần mềm độc hại khi người dùng truy cập).

2.3.3 Command and Control (CC) Server URL

- Các URL này thuộc về máy chủ điều khiển của hacker, dùng để gửi lệnh đến máy tính bị nhiễm malware.
- Được sử dụng trong tấn công botnet, ransomware.
- Ví dụ:

– *http://botnet-control.com/command?attack=DDOS*

2.3.4 URL chứa các lỗ hổng tấn công

- **SQL Injection (SQLi):** Chèn mã độc vào URL để tấn công cơ sở dữ liệu.
- **Cross-Site Scripting (XSS):** Chèn mã JavaScript độc hại để đánh cắp cookie hoặc tấn công trình duyệt người dùng.
- Ví dụ:

– *http://example.com/search?q=<script>alert('Hacked!')</script>*

2.3.5 URL sử dụng Shortened Links (Liên kết rút gọn)

- Các URL được rút gọn bằng dịch vụ như bit.ly, tinyurl.com để che giấu địa chỉ thực sự.
- Tin tặc có thể lợi dụng để dụ người dùng nhấp vào các trang web độc hại.
- Ví dụ:

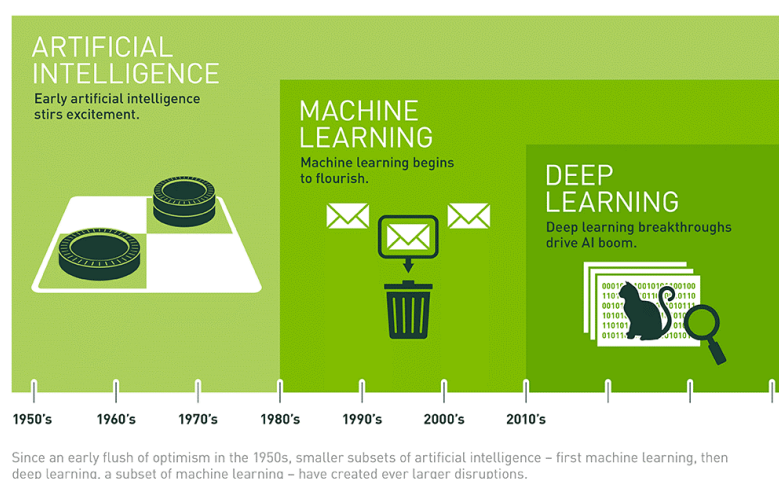
– *https://bit.ly/3abcXYZ*

2.4 Học máy (Machine Learning)

2.4.1 Khái niệm

Học máy là một nhánh con của trí tuệ nhân tạo (Artificial Intelligence - AI) cho phép máy tính học từ dữ liệu mà không cần được lập trình thông minh cụ thể. Học máy tập trung vào việc phát triển các thuật toán và mô hình cho phép máy tính nhận diện mẫu, dự đoán và có khả năng ra quyết định tự động.

2.4.2 Lịch sử và phát triển



Hình 3: Lịch sử và phát triển của học máy

(Nguồn: <https://bell24vietnam.vn/en/cong-nghe/tong-hop-kien-thuc-ve-machine-learning/>)

- 1950: Alan Turing, nhà bác học tiên phong trong lĩnh vực máy tính, đã tạo ra "Turing Test" (Phép thử Turing) nhằm xác định xem máy tính có sở hữu trí tuệ nhân tạo thực sự hay không. Để vượt qua bài kiểm tra này, máy tính phải có khả năng đánh lừa con người tin rằng nó cũng là một con người.
- 1952: Arthur Samuel phát triển chương trình học máy (computer learning) đầu tiên – một trò chơi cờ đam. Hãng IBM sau đó đã tiếp tục cải tiến chương trình này để nó có khả năng tự học và cải thiện chiến thuật để giành chiến thắng.

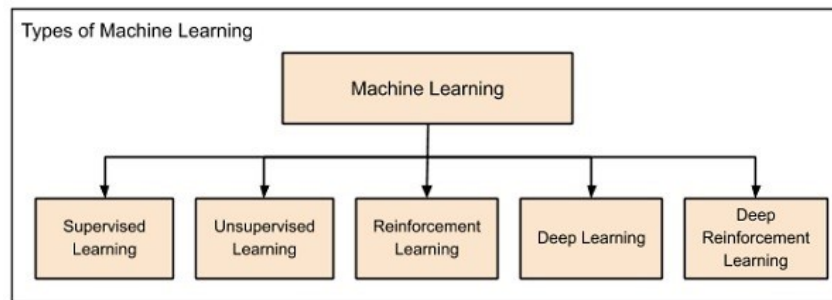
- 1957: Frank Rosenblatt thiết kế ra mạng nơ-ron (neural network) đầu tiên cho máy tính, mô phỏng quá trình tư duy của bộ não con người, đánh dấu bước tiến lớn trong việc phát triển trí tuệ nhân tạo.
- 1967: Thuật toán "Nearest Neighbor" được thiết kế, cho phép máy tính thực hiện nhận dạng mẫu cơ bản (pattern recognition). Một ứng dụng thực tế là sử dụng để vẽ ra lộ trình cho một người bán hàng có thể bắt đầu đi từ một địa điểm ngẫu nhiên nhưng đảm bảo anh ta sẽ đi qua tất cả các địa điểm khác theo một quãng đường ngắn nhất.
- 1979: Tại Đại học Stanford, sinh viên đã chế tạo thành công "Stanford Cart", một chiếc xe có khả năng tự điều hướng và tránh chướng ngại vật trong phòng.
- 1981: Gerald Dejong giới thiệu khái niệm Explanation-Based Learning (EBL), cho phép máy tính phân tích dữ liệu huấn luyện và từ đó rút ra các quy tắc tổng quát bằng cách loại bỏ thông tin không cần thiết.
- 1985: Terry Sejnowski phát triển "NetTalk", một chương trình có khả năng học cách phát âm các từ tương tự như quá trình trẻ em học nói.
- 1990s: Học máy chuyển từ phương pháp tiếp cận hướng tri thức "knowledge-driven" sang tiếp cận hướng dữ liệu "data-driven", cho phép máy tính học từ các tập dữ liệu lớn và rút ra kết luận một cách tự động.
- 1997: "Deep Blue" của IBM đã đánh bại nhà vô địch cờ vua thế giới.
- 2006: Geoffrey Hinton giới thiệu thuật ngữ "deep learning", đề cập đến các thuật toán mới cho phép máy tính "nhìn thấy" và phân biệt các đối tượng và văn bản trong hình ảnh và video.
- 2010: Microsoft Kinect là thiết bị có khả năng theo dõi 20 chuyển động của cơ thể người với tốc độ 30 lần mỗi giây, mở ra khả năng tương tác với máy

tính thông qua cử chỉ.

- 2011: Siêu máy tính Watson của IBM, đánh bại các đối thủ con người trong chương trình truyền hình Jeopardy, chứng minh khả năng xử lý vượt trội.
- 2011: Google phát triển “Google Brain”, một mạng nơ-ron sâu (deep neural network) có thể tự học cách nhận diện các đối tượng như mèo từ các video và hình ảnh.
- 2012: X Lab của Google xây dựng một thuật toán học máy có khả năng tự động quét hàng triệu video trên YouTube và phát hiện video có chứa mèo, đánh dấu bước ngoặt trong khả năng nhận diện hình ảnh của máy tính.
- 2014: Facebook giới thiệu “DeepFace”, một hệ thống nhận diện khuôn mặt có độ chính xác gần như ngang bằng với con người.
- 2015: Amazon ra mắt nền tảng học máy riêng.
- 2015: Microsoft phát triển “Distributed Machine Learning Toolkit”, cho phép giải các bài toán học máy trên quy mô lớn bằng cách phân phối xử lý trên nhiều máy tính.
- 2015: Hơn 3.000 nhà nghiên cứu trong lĩnh vực AI và Robotics, bao gồm các tên tuổi lớn như Stephen Hawking, Elon Musk và Steve Wozniak, đã ký vào bức thư ngỏ cảnh báo về nguy cơ từ vũ khí tự động có khả năng tự lựa chọn và tấn công mục tiêu mà không cần sự can thiệp của con người.
- 2016: AlphaGo - thuật toán trí tuệ nhân tạo của Google DeepMind, đánh bại Lee Sedol - nhà vô địch thế giới trò chơi Cờ Vây, một trò chơi được xem là phức tạp hơn nhiều so với cờ vua. AlphaGo thắng 4/5 ván đấu, đánh dấu bước tiến vượt bậc của AI.

2.4.3 Phân loại học máy

Học máy được chia thành năm nhóm chính:



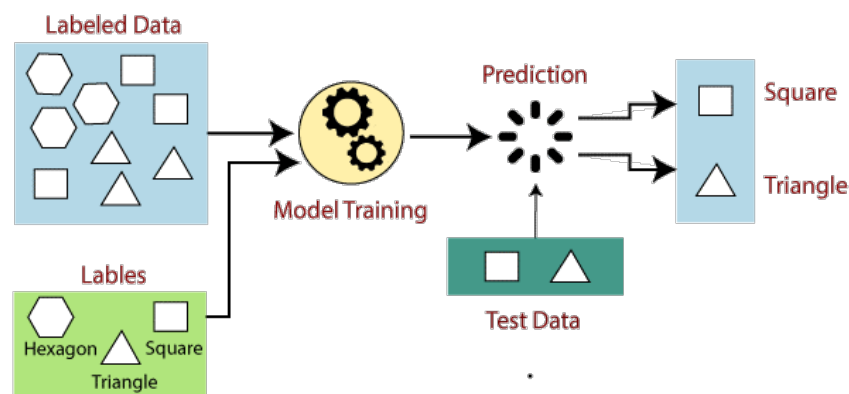
Hình 4: Phân loại học máy

(Nguồn: <https://newnet.vn/ai-machine-learning-co-ban/bai-5-machine-learning-phan-loai>)

1. Học có giám sát (Supervised Learning):

(a) Khái niệm:

- Học có giám sát là phương pháp học mà mô hình được huấn luyện trên một tập dữ liệu đã có nhãn (labeled data). Mỗi đầu vào trong tập dữ liệu tương ứng với một đầu ra mong muốn. Mục tiêu là để mô hình học mối quan hệ giữa đầu vào và đầu ra, từ đó dự đoán kết quả cho các dữ liệu mới.



Hình 5: Học có giám sát (Supervised Learning)

(Nguồn: <https://www.tpointtech.com/supervised-machine-learning>)

(b) Cách hoạt động:

- Mô hình nhận dữ liệu đầu vào cùng với nhãn đúng.

- So sánh đầu ra dự đoán với nhãn thực tế.
- Tính toán sai số (loss) và điều chỉnh trọng số để giảm sai số.

(c) Các bài toán chính trong học có giám sát:

- Phân loại (Classification):
 - Dự đoán đầu ra thuộc một trong các nhóm hay lớp cụ thể.
 - Ví dụ: Nhận diện email là spam hay không spam, phân loại hình ảnh thành mèo hay chó.
- Hồi quy (Regression):
 - Dự đoán giá trị liên tục.
 - Ví dụ: Dự đoán giá nhà, dự báo nhiệt độ.

(d) Các thuật toán phổ biến:

- Hồi quy tuyến tính (Linear Regression)
- Hồi quy logistic (Logistic Regression)
- Cây quyết định (Decision Trees)
- Rừng ngẫu nhiên (Random Forest)
- Máy vector hỗ trợ (Support Vector Machines - SVM)
- K-Láng giềng gần nhất (K-Nearest Neighbors - KNN)

(e) Ưu điểm:

- Hiệu quả với dữ liệu có cấu trúc rõ ràng.
- Dễ đánh giá hiệu suất thông qua các chỉ số như accuracy, precision, recall.

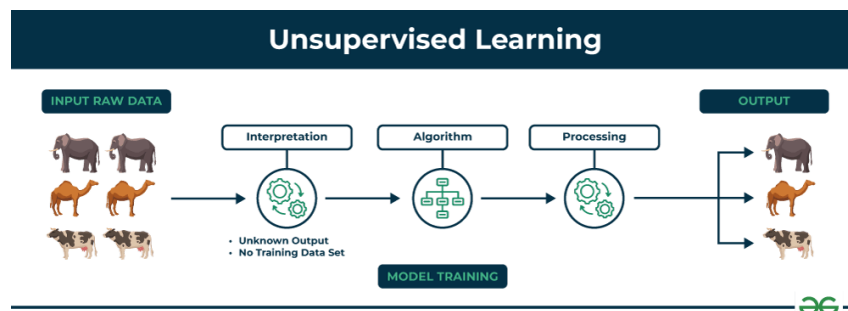
(f) Nhược điểm:

- Cần một lượng lớn dữ liệu đã được gán nhãn.
- Đòi hỏi chất lượng dữ liệu cao và nhãn chính xác.

2. Học không giám sát (Unsupervised Learning):

(a) Khái niệm:

- Học không giám sát là phương pháp học mà mô hình được huấn luyện trên tập dữ liệu chưa có nhãn (unlabeled data). Mục tiêu là tìm ra các cấu trúc ẩn hoặc mẫu trong dữ liệu.



Hình 6: Học không giám sát (Unsupervised Learning)
(Nguồn: <https://www.geeksforgeeks.org/unsupervised-learning/>)

(b) Cách hoạt động:

- Mô hình tự tìm kiếm các mẫu, nhóm hoặc cấu trúc trong dữ liệu mà không có hướng dẫn cụ thể.
- Thường được sử dụng để giảm chiều dữ liệu hoặc phân cụm các đối tượng tương tự.

(c) Các bài toán chính trong học không giám sát:

- Phân cụm (Clustering):
 - Nhóm các dữ liệu giống nhau vào cùng một cụm.
 - Ví dụ: Phân nhóm khách hàng dựa trên hành vi mua sắm.
- Giảm chiều (Dimensionality Reduction):
 - Giảm số lượng biến đầu vào trong khi vẫn giữ được các đặc trưng quan trọng của dữ liệu.
 - Ví dụ: Phân tích thành phần chính (PCA), dùng để nén dữ liệu hình ảnh.

(d) Các thuật toán phổ biến:

- K-means Clustering
- Hierarchical Clustering (Phân cụm phân cấp)
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
- Principal Component Analysis (PCA)
- t-Distributed Stochastic Neighbor Embedding (t-SNE)

(e) Ưu điểm:

- Không cần dữ liệu có nhãn.
- Tốt cho việc khám phá dữ liệu mới và tìm hiểu cấu trúc ẩn.

(f) Nhược điểm:

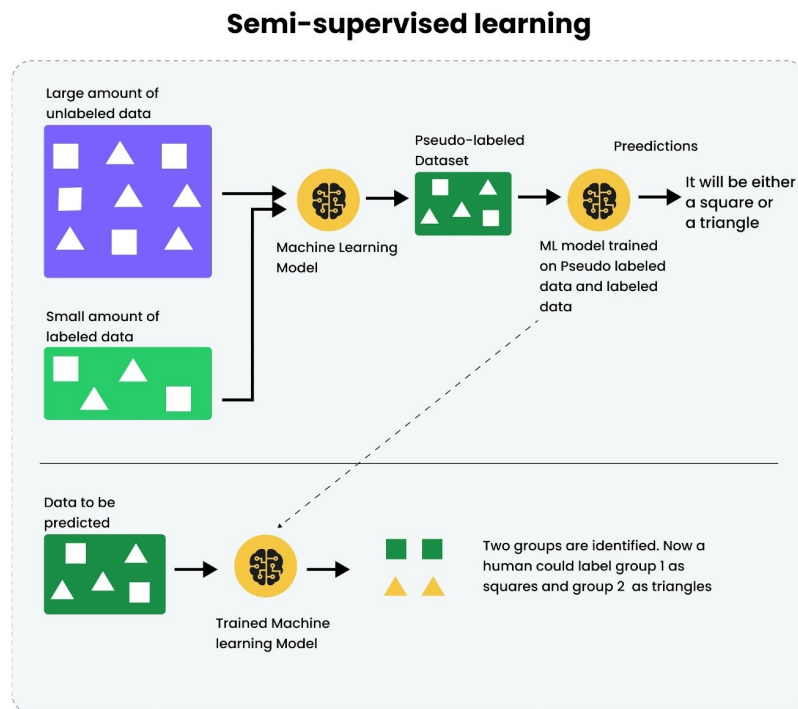
- Khó đánh giá hiệu suất do thiếu nhãn.
- Dễ bị ảnh hưởng bởi nhiễu và dữ liệu ngoại lai (outliers).

3. Học bán giám sát (Semi-Supervised Learning):**(a) Khái niệm:**

- Học bán giám sát kết hợp cả dữ liệu có nhãn và dữ liệu không có nhãn. Thông thường, một lượng nhỏ dữ liệu có nhãn được sử dụng cùng với một lượng lớn dữ liệu không có nhãn để cải thiện hiệu quả học tập.

(b) Cách hoạt động:

- Tác nhân (Agent): Thực hiện hành động trong môi trường.
- Môi trường (Environment): Cung cấp phản hồi (phần thưởng hoặc hình phạt) dựa trên hành động của tác nhân.
- Chính sách (Policy): Chiến lược mà tác nhân sử dụng để quyết định hành động.



Hình 7: Học bán giám sát (Semi-Supervised Learning)
(Nguồn: <https://maddevs.io/blog/semi-supervised-learning-explained/>)

- Hàm giá trị (Value Function): Đánh giá lợi ích dài hạn của một trạng thái.

(c) Ví dụ ứng dụng:

- Trò chơi điện tử (Game AI): AlphaGo của Google đã đánh bại nhà vô địch cờ vây thế giới nhờ vào học tăng cường.
- Xe tự lái: Điều chỉnh hành vi lái xe để tối ưu hóa sự an toàn và hiệu quả.

(d) Thuật toán phổ biến:

- Q-learning
- Deep Q Network (DQN)
- Policy Gradient Methods
- Actor-Critic Methods

(e) Ưu điểm:

- Khả năng học hỏi từ môi trường động và phức tạp.
- Hiệu quả trong các bài toán tối ưu hóa dài hạn.

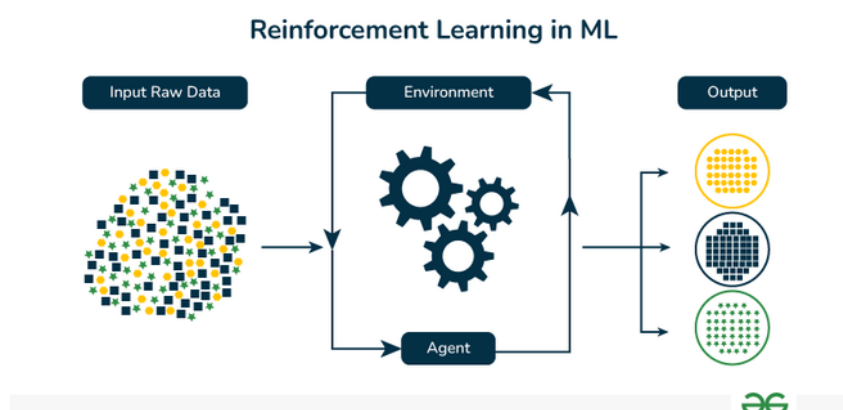
(f) **Nhược điểm:**

- Quá trình huấn luyện có thể tốn kém thời gian và tài nguyên.
- Khó khăn trong việc xác định chính sách tối ưu.

4. Học tự giám sát (Self-Supervised Learning):

(a) **Khái niệm:**

- Học tự giám sát là phương pháp nằm giữa học có giám sát và học không giám sát. Mô hình tự tạo nhãn từ dữ liệu thô để học các đặc trưng quan trọng mà không cần sự can thiệp của con người.



Hình 8: Học tự giám sát (Self-Supervised Learning)
(Nguồn: <https://www.geeksforgeeks.org/self-supervised-learning-ssl/>)

(b) **Cách hoạt động:**

- Mô hình tự động sinh ra các bài toán phụ (pretext tasks) từ dữ liệu chưa có nhãn và học từ các bài toán đó.
- Sau khi học các đặc trưng từ bài toán phụ, mô hình có thể được sử dụng cho các bài toán chính.

(c) **Ví dụ ứng dụng:**

- Xử lý ngôn ngữ tự nhiên (NLP): BERT và GPT học từ văn bản bằng cách dự đoán từ bị che giấu.
- Thị giác máy tính (Computer Vision): Dự đoán phần còn thiếu của hình ảnh.

(d) **Ưu điểm:**

- Giảm sự phụ thuộc vào dữ liệu có nhãn.
- Tốt cho các bài toán với dữ liệu lớn và phức tạp.

(e) **Nhược điểm:**

- Cần thiết kế các bài toán phù hợp lý để đạt hiệu quả cao.
- Đòi hỏi tài nguyên tính toán lớn trong quá trình huấn luyện.

5. So sánh các phương pháp

Phương pháp	Dữ liệu cần thiết	Ứng dụng chính	Ưu điểm	Nhược điểm
Học có giám sát (Supervised Learning)	Dữ liệu có nhãn	Phân loại, hồi quy	Độ chính xác cao nếu dữ liệu tốt	Độ chính xác cao nếu dữ liệu tốt
Học không giám sát (Unsupervised Learning)	Dữ liệu không có nhãn	Phân cụm, giảm chiều	Khám phá dữ liệu mới, không cần nhãn	Khó đánh giá hiệu suất, dễ bị nhiễu
Học bán giám sát (Semi-Supervised Learning)	Dữ liệu pha trộn (có và không có nhãn)	Nhận diện khuôn mặt, phân loại văn bản	Giảm chi phí gán nhãn	Phụ thuộc vào chất lượng dữ liệu có nhãn
Học tăng cường (Reinforcement Learning)	Phản hồi từ môi trường	Trò chơi, xe tự lái	Tối ưu hóa dài hạn, thích nghi tốt	Huấn luyện lâu, tài nguyên tốn kém
Học tự giám sát (Self-Supervised Learning)	Dữ liệu thô (chưa gán nhãn)	Xử lý ngôn ngữ tự nhiên, thị giác máy tính	Giảm phụ thuộc vào nhãn, khai thác dữ liệu lớn	Thiết kế bài toán phụ phức tạp

Bảng 1: So sánh các phân loại của học máy

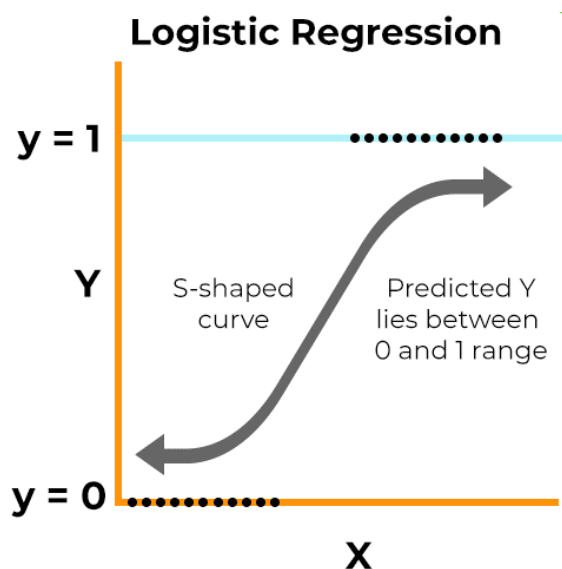
2.4.4 Một số mô hình học máy sử dụng

Trong dự án này, nhóm tác giả sử dụng một số mô hình học máy quen thuộc để thực hiện bài toán. Các mô hình như sau:

1. Logistic Regression

(a) Khái niệm:

Logistic Regression là một thuật toán học có giám sát được sử dụng phổ biến trong các bài toán phân loại nhị phân (binary classification). Mặc dù tên gọi là "Regression," nhưng mô hình này thực chất giải quyết các bài toán phân loại thay vì hồi quy



Hình 9: Logistic Regression

(Nguồn: <https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/>)

Thuật toán hoạt động bằng cách mô hình hóa xác suất của một quan sát thuộc về một lớp cụ thể. Logistic Regression sử dụng hàm sigmoid (hoặc hàm logistic) để chuyển đổi đầu ra tuyến tính thành một giá trị xác suất nằm trong khoảng từ 0 đến 1.

(b) Công thức:

- Hàm sigmoid được định nghĩa như sau:

$$\sigma(z) = \frac{1}{1+e^{-z}}$$

- Trong đó, $z = w^T x + b$ là tổ hợp tuyến tính của các đặc trưng đầu vào x , trọng số w , và hệ số chệch b .
- Dựa trên giá trị xác suất này, mô hình sẽ quyết định nhãn phân loại bằng cách so sánh với một ngưỡng (thường là 0.5).
- Mô hình Logistic Regression có dạng:

$$P(Y = 1 | X) = \sigma(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)$$

- Trong đó, $P(Y = 1 | X)$ là xác suất của biến phụ thuộc Y bằng 1 khi biết các biến độc lập X .
- $\beta_0, \beta_1, \dots, \beta_n$ là các hệ số hồi quy cần ước lượng.
- Các tham số của mô hình được ước lượng bằng phương pháp tối đa hóa hàm hợp lý (Maximum Likelihood Estimation - MLE). Mục tiêu là tìm các giá trị của sao cho xác suất dự đoán của mô hình khớp với dữ liệu quan sát.

(c) Ứng dụng:

- Logistic Regression không chỉ là một công cụ thống kê mà còn là một thuật toán học máy cơ bản. Nó thường được sử dụng như một baseline model trong các bài toán phân loại và có thể mở rộng thành các mô hình phức tạp hơn như Neural Networks.

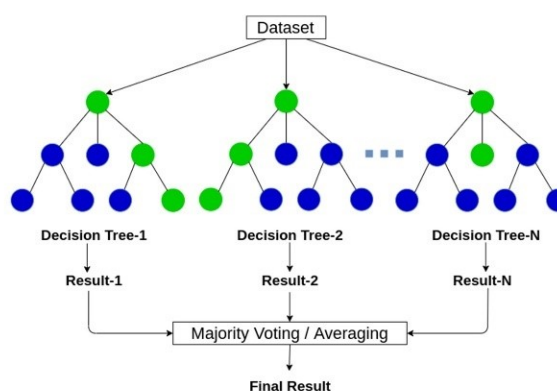
2. Random Forest

(a) Khái niệm:

Random Forest là một thuật toán học có giám sát thuộc nhóm *ensemble learning*, sử dụng kỹ thuật *bagging* (*Bootstrap Aggregating*) để xây dựng nhiều cây quyết định (decision trees) và kết hợp kết quả của chúng để đưa ra dự đoán chính xác hơn. Đối với bài toán phân loại, Random Forest lấy

kết quả dự đoán theo đa số phiếu bầu (*majority voting*) từ các cây; còn với bài toán hồi quy, kết quả là trung bình giá trị dự đoán từ các cây. Một trong những lý do chính khiến Random Forest hoạt động hiệu quả là tính đa dạng của các cây quyết định. Bằng cách sử dụng bootstrap sampling và chọn ngẫu nhiên các đặc trưng tại mỗi nút, các cây trong rừng sẽ có sự khác biệt đáng kể. Điều này giúp giảm thiểu hiện tượng *overfitting*, vì các lỗi của các cây đơn lẻ sẽ được trung hòa khi kết hợp kết quả.

Random Forest



Hình 10: Random Forest

(Nguồn: <https://anasbrital98.github.io/blog/2021/Random-Forest/>)

Mỗi cây quyết định trong Random Forest được xây dựng từ một tập con ngẫu nhiên của dữ liệu huấn luyện (với phép lấy mẫu có hoàn lại). Đồng thời, tại mỗi nút chia (*split node*), chỉ một tập con ngẫu nhiên các đặc trưng được xem xét để tìm ra điểm chia tốt nhất, giúp tăng tính đa dạng giữa các cây và giảm nguy cơ *overfitting*.

(b) Cách hoạt động:

- Dựa vào đặc điểm trên, có thể cấu trúc cách hoạt động của mô hình Random Forest như sau:
 - Đầu tiên, tạo nhiều tập dữ liệu con từ tập dữ liệu gốc bằng cách lấy

mẫu ngẫu nhiên tái lập lại bằng phương pháp bootstrap sampling.

Điều này có nghĩa là từ tập dữ liệu gốc, chúng ta lấy mẫu ngẫu nhiên có hoàn lại để tạo ra các tập con. Mỗi tập con này sẽ được sử dụng để huấn luyện một cây quyết định riêng biệt.

- Với mỗi tập dữ liệu con, xây dựng một cây quyết định. Mỗi nút trong cây chỉ xem xét một tập con ngẫu nhiên của các đặc trưng (features) để tìm ra đặc trưng tốt nhất để chia tách.

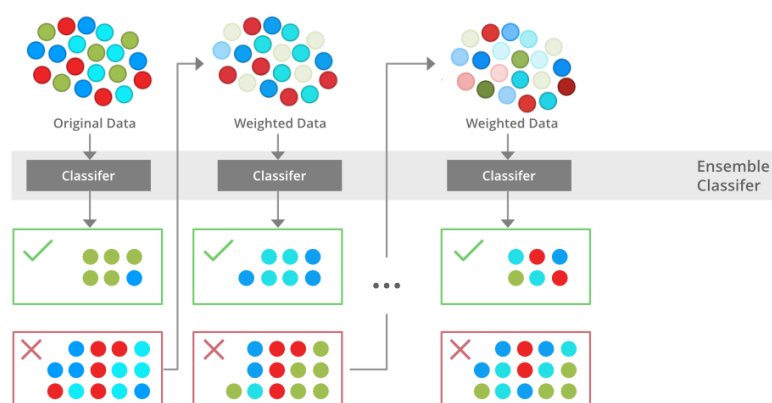
(c) Quá trình xây dựng cây quyết định:

- **Chọn ngẫu nhiên một tập con của các đặc trưng:** Tại mỗi nút của cây, chỉ một tập con ngẫu nhiên của các đặc trưng được xem xét để tìm ra đặc trưng tốt nhất để chia tách. Điều này giúp giảm tương quan giữa các cây và tăng tính đa dạng của rừng.
- **Chia tách nút:** Đối với mỗi nút, thuật toán tìm ra đặc trưng và ngưỡng chia tách tốt nhất để tối đa hóa độ thuần khiết (purity) của các nút con. Độ thuần khiết có thể được đo bằng các chỉ số như Gini impurity hoặc entropy.
- **Lặp lại quá trình:** Quá trình chia tách tiếp tục lặp lại cho đến khi đạt đến một điều kiện dừng, chẳng hạn như độ sâu tối đa của cây hoặc số lượng mẫu tối thiểu tại một nút.
- **Kết hợp kết quả:** Đối với bài toán phân loại, Random Forest sử dụng phương pháp bỏ phiếu đa số (majority voting) để quyết định nhãn cuối cùng. Đối với bài toán hồi quy, nó tính trung bình các dự đoán của các cây.
- Sau khi xây dựng xong tất cả các cây quyết định, Random Forest kết hợp kết quả của các cây này để đưa ra dự đoán cuối cùng:

- **Đối với bài toán phân loại:** Random Forest sử dụng phương pháp bỏ phiếu đa số (majority voting). Mỗi cây quyết định đưa ra một dự đoán, và nhãn cuối cùng được quyết định dựa trên nhãn được dự đoán nhiều nhất.
- **Đối với bài toán hồi quy:** Random Forest tính trung bình các dự đoán của các cây để đưa ra giá trị dự đoán cuối cùng.

3. XGBoost (Extreme Gradient Boosting)

- (a) **Khái niệm:** XGBoost là một thuật toán thuộc nhóm *ensemble learning*, sử dụng kỹ thuật *boosting* để cải thiện hiệu suất dự đoán. *Boosting* là phương pháp xây dựng các mô hình yếu (*weak learners*), thường là các cây quyết định nhỏ (decision trees), theo trình tự liên tiếp. Mỗi mô hình mới sẽ cố gắng sửa chữa các lỗi mà mô hình trước đó mắc phải, từ đó nâng cao độ chính xác của dự đoán tổng hợp.



Hình 11: XGBoost

(Nguồn: <https://medium.com/@techynilesh/xgboost-algorithm-explained-in-less-than-5-minutes-b561dcc1ccee>)

XGBoost mở rộng từ thuật toán *Gradient Boosting* bằng cách tối ưu hóa cả về hiệu suất tính toán lẫn độ chính xác dự đoán. Nó sử dụng *gradient descent* để giảm thiểu hàm mất mát (loss function) và bổ sung nhiều cải tiến như regularization (chính quy hóa) để ngăn ngừa *overfitting*, *parallel*

processing (xử lý song song) để tăng tốc độ huấn luyện, và *tree pruning* (cắt tỉa cây) để tối ưu cấu trúc cây

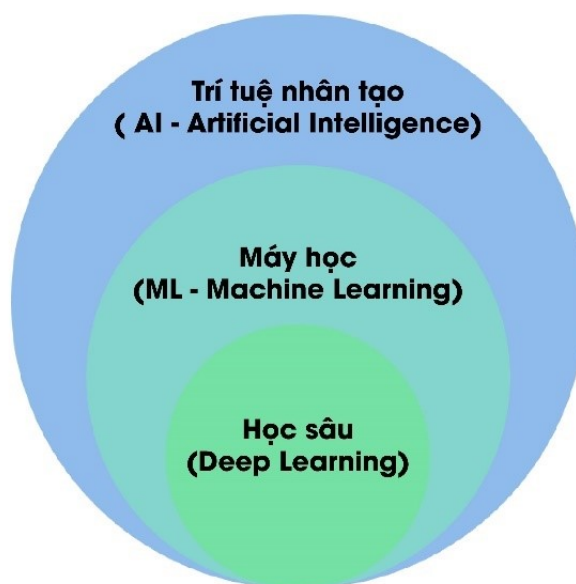
(b) **Bước hoạt động:**

- **Khởi tạo mô hình:** bằng việc bắt đầu với một mô hình đơn giản (thường là một giá trị trung bình cho hồi quy hoặc một dự đoán ngẫu nhiên cho phân loại).
- **Xây dựng cây quyết định:** Tại mỗi bước, một cây quyết định mới được xây dựng để giảm thiểu lỗi của mô hình hiện tại. Cây này được thêm vào mô hình hiện tại để cải thiện dự đoán. Quá trình này bao gồm:
 - **Chọn ngẫu nhiên một tập con của các đặc trưng:** Tại mỗi nút của cây, chỉ một tập con ngẫu nhiên của các đặc trưng được xem xét để tìm ra đặc trưng tốt nhất để chia tách.
 - **Chia tách nút:** Đối với mỗi nút, thuật toán tìm ra đặc trưng và ngưỡng chia tách tốt nhất để tối đa hóa độ thuần khiết (purity) của các nút con. Độ thuần khiết có thể được đo bằng các chỉ số như Gini impurity hoặc entropy.
 - **Lặp lại quá trình:** Quá trình chia tách tiếp tục lặp lại cho đến khi đạt đến một điều kiện dừng, chẳng hạn như độ sâu tối đa của cây hoặc số lượng mẫu tối thiểu tại một nút.
- **Cập nhật mô hình:** Mô hình hiện tại được cập nhật bằng cách thêm cây mới vào. Quá trình này lặp lại cho đến khi đạt được số lượng cây quyết định mong muốn hoặc lỗi giảm xuống dưới một ngưỡng nhất định.

2.5 Học sâu (Deep Learning)

2.5.1 Khái niệm

Học sâu (Deep Learning) là một nhánh của Trí tuệ nhân tạo (Artificial Intelligence - AI) và Học máy (Machine Learning - ML), sử dụng các mạng nơ-ron nhân tạo nhiều tầng (Artificial Neural Networks - ANN) để mô phỏng cách bộ não con người học từ dữ liệu. Học sâu cho phép máy tính tự động trích xuất đặc trưng từ dữ liệu mà không cần sự can thiệp của con người, nhờ vào khả năng học biểu diễn dữ liệu phức tạp.



Hình 12: Học sâu (Deep Learning)

(Nguồn: <https://www.elcom.com.vn/deep-learning-la-gi-ung-dung-cua-hoc-sau-trong-thuc-tien-1666582364>)

Sự phát triển của học sâu được thúc đẩy nhờ ba yếu tố chính: sự gia tăng về dữ liệu (Big Data), cải tiến về phần cứng (GPU, TPU), và những tiến bộ trong thuật toán tối ưu hóa.

Học sâu có khả năng học từ dữ liệu với quy mô lớn và tự động trích xuất các đặc trưng mà không cần sự can thiệp của con người.

2.5.2 Cấu trúc mạng Nơ-ron nhân tạo

- Mạng nơ-ron nhân tạo trong Deep Learning gồm nhiều lớp, có thể chia thành:
 - **Lớp đầu vào (Input Layer):** Nhận dữ liệu thô (hình ảnh, văn bản, âm thanh, ...) và truyền đến các lớp ẩn.
 - **Lớp ẩn (Hidden Layers):** gồm nhiều lớp Nơ-ron xử lý dữ liệu, có nhiệm vụ trích xuất đặc trưng, càng nhiều tầng thì mô hình càng sâu.
 - **Lớp đầu ra (Output Layer):** Dự đoán kết quả (ví dụ: phân loại hình ảnh là "mèo" hoặc "chó").
- Mỗi nơ-ron trong mạng có một trọng số (weight) và một độ lệch (bias), giúp điều chỉnh quá trình tính toán. Hàm kích hoạt (Activation Function) được sử dụng để đưa ra các quyết định phi tuyến, phổ biến nhất là ReLU, Sigmoid, và Softmax.

2.5.3 Một số mô hình học sâu

1. MLP (Multi-Layer Perceptron)

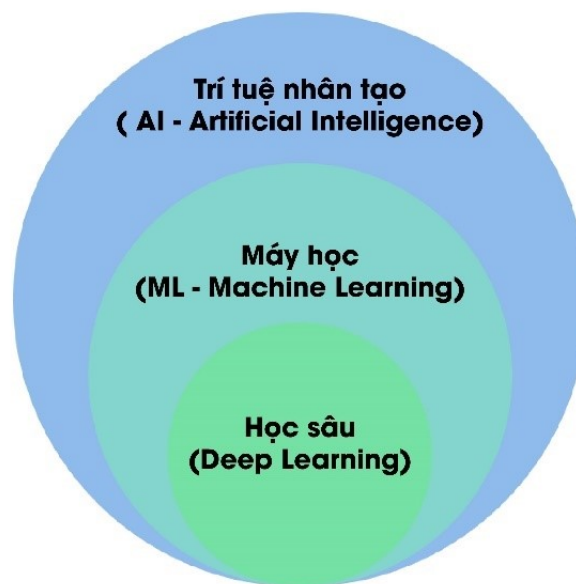
(a) **Khái niệm:** Multi-Layer Perceptron (MLP) là một loại mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) thuộc nhóm Deep Learning, được sử dụng phổ biến cho các bài toán phân loại và hồi quy. MLP bao gồm ít nhất ba lớp: một lớp đầu vào (input layer), một hoặc nhiều lớp ẩn (hidden layers), và một lớp đầu ra (output layer). Mỗi lớp bao gồm các nơ-ron (neurons) được kết nối với nhau thông qua các trọng số (weights).

Điểm nổi bật của MLP là khả năng mô hình hóa các mối quan hệ phi tuyến tính nhờ vào việc sử dụng các hàm kích hoạt (activation functions) như *ReLU*, *Sigmoid*, hay *Tanh*. Các kết nối trong mạng được điều chỉnh

thông qua *quá trình lan truyền ngược (backpropagation)* và *tối ưu hóa gradient* để giảm thiểu hàm mất mát.

(b) **Cấu trúc của MLP:**

- **Lớp đầu vào (Input Layer):** Nhận dữ liệu đầu vào, mỗi đặc trưng của dữ liệu tương ứng với một nơ-ron.
- **Lớp ẩn (Hidden Layers):** Mỗi nơ-ron trong lớp ẩn tính toán đầu ra bằng cách áp dụng hàm kích hoạt lên tổng có trọng số của các đầu vào. Số lượng lớp ẩn và số nơ-ron trong mỗi lớp có thể được điều chỉnh để tối ưu hóa hiệu suất mô hình.
- **Lớp đầu ra (Output Layer):** Đưa ra kết quả cuối cùng của mô hình, tùy thuộc vào bài toán là phân loại (sử dụng Softmax hoặc Sigmoid) hay hồi quy (sử dụng hàm tuyến tính).



Hình 13: Multi-Layer Perceptron

(Nguồn: <https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f>)

(c) **Cơ chế hoạt động:**

- **Lan truyền tiến (Forward Propagation):** Dữ liệu đầu vào được nhân

với trọng số của từng nơ-ron và cộng với độ chệch. Tiếp đó, hàm kích hoạt được áp dụng để tạo ra đầu ra cho từng nơ-ron. Quá trình này tiếp tục qua các tầng ẩn đến khi đạt được đầu ra cuối cùng.

- **Tính toán hàm mất mát (Loss Function):** Đánh giá mức độ sai lệch giữa đầu ra dự đoán và giá trị thực tế. Các hàm mất phổ biến: MSE (Mean Squared Error) cho bài toán hồi quy, Cross-Entropy cho bài toán phân loại.
- **Lan truyền ngược (Backward Propagation):** Sử dụng thuật toán tối ưu như Gradient Descent hoặc Adam để cập nhật trọng số bằng cách giảm dần lỗi dựa trên đạo hàm của hàm mất mát. Quá trình lặp lại nhiều lần để tối ưu hóa mô hình.

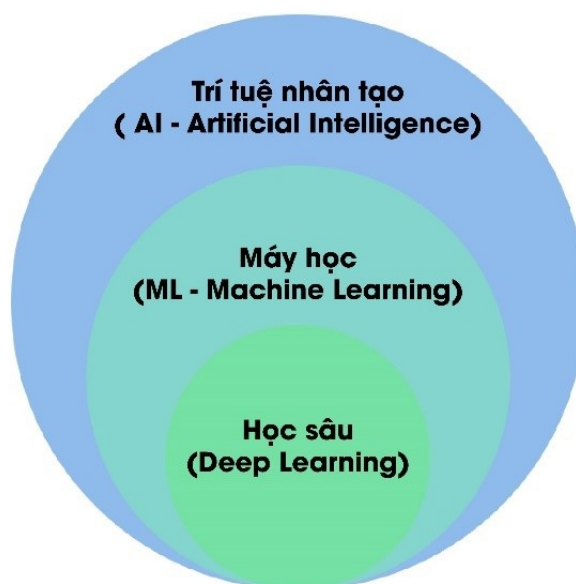
2. LSTM (Long Short-Term Memory)

(a) **Khái niệm:** Long Short-Term Memory (LSTM) là một loại mạng nơ-ron hồi tiếp (Recurrent Neural Network - RNN), được thiết kế đặc biệt để giải quyết vấn đề ghi nhớ dài hạn trong mô hình học sâu. LSTM xử lý và dự đoán các chuỗi dữ liệu có thứ tự theo thời gian, ví dụ như chuỗi văn bản, tín hiệu âm thanh, hoặc dữ liệu tài chính. LSTM được giới thiệu bởi Sepp Hochreiter và Jürgen Schmidhuber vào năm 1997 để khắc phục các vấn đề chính của RNN truyền thống, đặc biệt là vấn đề biến mất và bùng nổ gradient (vanishing and exploding gradient problem) khi xử lý các chuỗi dài.

Đối với RNN truyền thống, việc học và xử lý chuỗi dữ liệu có một vai trò quan trọng và là một bước tiến lớn trong học sâu. Tuy nhiên, RNN truyền thống gặp phải vấn đề vanishing gradient (mất dần gradient) khi huấn luyện trên chuỗi dài. Điều này khiến mạng khó học được các phụ

thuộc dài hạn trong dữ liệu, làm giảm hiệu suất trên các tác vụ yêu cầu ghi nhớ lâu dài.

LSTM giải quyết vấn đề này bằng cách sử dụng các cổng điều khiển để quản lý thông tin lưu trữ và truyền đi. Nhờ đó, LSTM có thể học được mối quan hệ xa trong chuỗi dữ liệu một cách hiệu quả hơn. LSTM nổi bật với khả năng ghi nhớ thông tin lâu dài và quên thông tin không quan trọng thông qua các cấu trúc gọi là bộ nhớ (memory cell) và các cổng điều khiển (gates). Điều này giúp mô hình nắm bắt các mối quan hệ dài hạn trong dữ liệu thời gian, điều mà các RNN truyền thống gặp khó khăn.



Hình 14: Long Short-Term Memory (LSTM)
(Nguồn: <https://www.mathworks.com/discovery/lstm.html>)

(b) Cấu trúc:

- LSTM bao gồm ba cổng chính:
 - **Forget Gate (f_t):** Quyết định lượng thông tin từ trạng thái trước đó cần loại bỏ.
 - **Input Gate (i_t):** Kiểm soát lượng thông tin mới cần được thêm vào bộ nhớ.

– **Output Gate (o_t):** Xác định phần nào của bộ nhớ sẽ được sử dụng để tạo đầu ra cho bước hiện tại.

- Ngoài ra, LSTM duy trì một trạng thái ô nhớ (c_t) để lưu trữ thông tin quan trọng qua các bước thời gian.

2.6 Các kỹ thuật lựa chọn đặc trưng

- Để chọn được các đặc trưng có độ quan trọng cao, ảnh hưởng nhiều đến việc học tập để phân biệt URL là bình thường, độc hại hay thuộc loại độc hại nào từ 80 đặc trưng trong bộ dữ liệu ISCX-URL-2016. Chúng tôi chọn sử dụng bốn mô hình sau:

- Random Forest (Tree-based Feature Importance)
- Mutual Information
- Recursive Feature Elimination (RFE)
- SelectKBest

2.6.1 Random Forest (Tree-based Feature Importance)

1. Cơ sở lý thuyết:

Random Forest là một thuật toán học máy thuộc nhóm các mô hình học có giám sát, dựa trên việc xây dựng nhiều cây quyết định (Decision Trees) và kết hợp kết quả của chúng để cải thiện độ chính xác và giảm thiểu overfitting. Khi áp dụng Random Forest để chọn đặc trưng, thuật toán đo tầm quan trọng của từng đặc trưng dựa trên mức độ giảm *impurity* (thường là *Gini impurity* hoặc *entropy*) khi đặc trưng đó được sử dụng để phân chia dữ liệu trong các cây. Tổng mức giảm *impurity* của từng đặc trưng qua tất cả các cây được sử dụng để đánh giá tầm quan trọng của đặc trưng đó.

2. Ưu điểm:

- Random Forest tự động xử lý mối quan hệ phi tuyến giữa các đặc trưng và biến mục tiêu.
- Có khả năng xử lý dữ liệu lớn với nhiều đặc trưng mà không yêu cầu chuẩn hóa dữ liệu.
- Cung cấp thứ hạng rõ ràng cho các đặc trưng dựa trên mức độ đóng góp của chúng trong việc cải thiện mô hình.

3. Nhược điểm:

- Khó giải thích với dữ liệu có nhiều đặc trưng phụ thuộc.
- Thời gian tính toán lâu với tập dữ liệu lớn.

2.6.2 Mutual Information

1. Cơ sở lý thuyết:

Mutual Information (MI) là thước đo mức độ phụ thuộc lẫn nhau giữa hai biến ngẫu nhiên. Trong quá trình chọn lọc đặc trưng, MI được sử dụng để đánh giá mức độ liên quan giữa từng đặc trưng và biến mục tiêu. Khác với các phương pháp chỉ phát hiện mối quan hệ tuyến tính, MI có khả năng phát hiện cả các mối quan hệ phi tuyến giữa đặc trưng và nhãn. Giá trị MI càng cao chứng tỏ đặc trưng đó càng cung cấp nhiều thông tin giá trị về biến mục tiêu.

2. Ưu điểm:

- **Không giả định quan hệ tuyến tính:** MI không yêu cầu mối quan hệ giữa đặc trưng và biến mục tiêu phải là tuyến tính, giúp phát hiện các mối quan hệ phức tạp hơn.

- **Phát hiện quan hệ phi tuyến hiệu quả:** Phương pháp đặc biệt hữu ích trong việc nhận diện các đặc trưng có mối liên kết phi tuyến với biến mục tiêu mà các thuật toán khác có thể bỏ qua.
- **Dễ triển khai và tính toán:** MI có thể được tính toán nhanh chóng và dễ dàng, kể cả với các tập dữ liệu lớn, nhờ vào các công cụ thư viện sẵn có trong Python như sklearn.
- **Bổ sung cho các phương pháp khác:** MI có thể phát hiện ra các đặc trưng quan trọng mà các phương pháp dựa trên độ tinh khiết (impurity) như Random Forest không nhận diện được.

3. Nhược điểm:

- Có thể gặp khó khăn khi tính toán trên tập dữ liệu rất lớn hoặc nhiều đặc trưng.
- Cần xử lý đặc biệt với dữ liệu có nhiều giá trị thiếu.

2.6.3 Recursive Feature Elimination (RFE)

1. Cơ sở lý thuyết:

Recursive Feature Elimination (RFE) là một kỹ thuật chọn lọc đặc trưng dựa trên mô hình học máy. Phương pháp này hoạt động bằng cách huấn luyện một mô hình (ví dụ: SVM, Logistic Regression) để đánh giá tầm quan trọng của từng đặc trưng, sau đó loại bỏ những đặc trưng ít quan trọng nhất. Quá trình này được lặp đi lặp lại một cách đệ quy cho đến khi chỉ còn lại số lượng đặc trưng mong muốn. Mức độ quan trọng của đặc trưng được xác định dựa trên các *hệ số (coefficients)* hoặc *trọng số (feature importance)* của mô hình đã huấn luyện.

2. Ưu điểm:

- **Tính linh hoạt cao:** RFE có thể áp dụng với nhiều thuật toán học máy khác nhau, từ các mô hình tuyến tính như Logistic Regression đến các mô hình phức tạp như Random Forest hoặc SVM.
- **Kiểm soát số lượng đặc trưng:** Phương pháp cho phép người dùng chỉ định chính xác số lượng đặc trưng cần giữ lại, giúp tối ưu hóa mô hình theo yêu cầu cụ thể.
- **Chọn đặc trưng chính xác:** RFE loại bỏ đặc trưng dựa trên kết quả thực tế từ mô hình huấn luyện, giúp đảm bảo rằng các đặc trưng giữ lại thực sự có giá trị trong việc cải thiện hiệu suất mô hình.

3. Nhược điểm:

- Tốn thời gian và tài nguyên tính toán khi số lượng đặc trưng và dữ liệu lớn.
- RFE không phải lúc nào cũng đạt được kết quả tốt khi đặc trưng có mối quan hệ phi tuyến tính.

2.6.4 SelectKBest

1. Cơ sở lý thuyết:

- SelectKBest là một kỹ thuật lọc đặc trưng dựa trên các phép kiểm định thống kê. Phương pháp này chọn ra 'k' đặc trưng tốt nhất từ tập dữ liệu dựa trên điểm số thống kê, phản ánh mức độ liên quan giữa từng đặc trưng với biến mục tiêu. Các kiểm định phổ biến được sử dụng trong SelectKBest bao gồm:
 - **Chi-squared (Kiểm định chi-bình phương):** Thường dùng cho dữ liệu phân loại, kiểm tra sự độc lập giữa các đặc trưng và biến mục tiêu.

- **F-test (Kiểm định ANOVA):** Phù hợp với dữ liệu liên tục để đánh giá sự khác biệt về phương sai giữa các nhóm.
- **Mutual Information (Thông tin tương hỗ):** Đo lường lượng thông tin mà một đặc trưng cung cấp về biến mục tiêu, hiệu quả với cả dữ liệu liên tục và phân loại.
- Các đặc trưng có điểm số thống kê cao nhất sẽ được giữ lại, vì chúng thể hiện mối liên hệ chặt chẽ với biến mục tiêu.

2. Ưu điểm:

- **Đơn giản và dễ áp dụng:** SelectKBest rất dễ triển khai và không yêu cầu nhiều cấu hình phức tạp, lý tưởng cho cả người mới và các bài toán lớn.
- **Hiệu quả trong việc loại bỏ đặc trưng không liên quan:** Phương pháp này giúp nhanh chóng lọc ra các đặc trưng ít quan trọng, giảm thiểu nhiễu và tăng tốc độ huấn luyện mô hình.
- **Tính linh hoạt cao:** Người dùng có thể chọn phương pháp thống kê phù hợp với loại dữ liệu cụ thể (phân loại hoặc liên tục), từ đó tối ưu hóa quá trình chọn đặc trưng.

3. Nhược điểm:

- Có thể không hoạt động tốt với các mối quan hệ phi tuyến tính giữa các đặc trưng và biến mục tiêu.
- Nếu K không được chọn một cách hợp lý, mô hình có thể thiếu đi những thông tin quan trọng.

2.6.5 So sánh các mô hình

Tiêu chí	Random Forest (Tree-based Feature Importance)	Mutual Information (MI)	Recursive Feature Elimination (RFE)	SelectKBest
Cơ sở lý thuyết	Đánh giá tầm quan trọng đặc trưng dựa trên độ giảm <i>impurity</i> khi xây dựng cây quyết định.	Đo lường mức độ phụ thuộc lẫn nhau giữa đặc trưng và biến mục tiêu, không giả định mối quan hệ tuyến tính.	Lặp lại quá trình huấn luyện mô hình, loại dần các đặc trưng ít quan trọng nhất.	Chọn 'k' đặc trưng tốt nhất dựa trên các kiểm định thống kê (chi-squared, F-test, MI).
Loại phương pháp	Phương pháp dựa trên mô hình (ensemble learning).	Phương pháp lọc (filter method).	Phương pháp bọc (wrapper method).	Phương pháp lọc (filter method).
Ưu điểm	Hiệu quả với dữ liệu phức tạp, mối quan hệ phi tuyến. Tự động xử lý tương tác giữa các đặc trưng. Chống overfitting tốt.	Phát hiện mối quan hệ phi tuyến giữa đặc trưng và biến mục tiêu. Dễ triển khai, tính toán nhanh. Không yêu cầu giả định về dữ liệu.	Linh hoạt với nhiều mô hình khác nhau. Tối ưu hóa số lượng đặc trưng thông qua lặp lại.	Đơn giản, dễ áp dụng. Nhanh chóng loại bỏ đặc trưng không liên quan. Linh hoạt trong lựa chọn kiểm định thống kê.
Nhược điểm	Khó giải thích với dữ liệu có nhiều đặc trưng phụ thuộc. Thời gian tính toán lâu với tập dữ liệu lớn.	Không đo lường được mối quan hệ đa chiều giữa các đặc trưng. Kết quả dễ bị ảnh hưởng bởi nhiễu dữ liệu.	Thời gian huấn luyện dài do cần lặp lại nhiều lần. Dễ bị ảnh hưởng bởi overfitting nếu không điều chỉnh đúng mô hình.	Không xem xét sự tương tác giữa các đặc trưng. Hiệu quả giảm khi đặc trưng có mối quan hệ phức tạp với biến mục tiêu.
Tính toán và thực thi	Trung bình đến cao (do yêu cầu huấn luyện mô hình cây quyết định).	Nhanh chóng, phù hợp với tập dữ liệu lớn.	Chậm hơn do yêu cầu huấn luyện lặp đi lặp lại. Thích hợp với tập dữ liệu nhỏ đến trung bình.	Rất nhanh, phù hợp với tập dữ liệu lớn và nhiều đặc trưng.
Khả năng xử lý quan hệ phi tuyến	Rất tốt, do khả năng chia nhỏ không gian dữ liệu của cây quyết định.	Tốt, có thể phát hiện các mối quan hệ phi tuyến.	Tùy thuộc vào mô hình sử dụng (SVM, Logistic Regression, ...)	Hạn chế, phụ thuộc vào phương pháp kiểm định được chọn.
Khả năng giải thích	Trung bình, khó giải thích khi số lượng đặc trưng lớn.	Tốt, dễ hiểu với mối quan hệ rõ ràng giữa đặc trưng và nhãn.	Tốt, dễ hiểu khi số lượng đặc trưng nhỏ.	Rất tốt, trực quan do dựa trên điểm số thống kê.
Kết quả mong đợi	Danh sách đặc trưng quan trọng nhất dựa trên tầm quan trọng từ mô hình cây.	Các đặc trưng có mức độ phụ thuộc cao nhất với biến mục tiêu.	Tập đặc trưng tối ưu sau khi loại bỏ những đặc trưng không cần thiết.	Top 'k' đặc trưng tốt nhất dựa trên kiểm định thống kê.

Bảng 2: So sánh các mô hình chọn đặc trưng

• Nhận xét chung:

- **Random Forest** phù hợp với các bài toán phức tạp và dữ liệu có mối quan hệ phi tuyến. Tuy nhiên, nó đòi hỏi tài nguyên tính toán lớn và khó giải thích khi số lượng đặc trưng nhiều.
- **Mutual Information** dễ triển khai và phù hợp để phát hiện các đặc trưng có mối quan hệ phi tuyến, nhưng không thể hiện rõ mối quan hệ đa chiều

giữa các đặc trưng.

- **RFE** cung cấp kết quả tối ưu khi kết hợp với mô hình mạnh như SVM hay Logistic Regression, tuy nhiên tốn thời gian và dễ bị overfitting nếu không điều chỉnh cẩn thận.
- **SelectKBest** là lựa chọn tốt cho các bài toán yêu cầu sự đơn giản và nhanh chóng, nhưng hạn chế trong việc phát hiện các mối quan hệ phức tạp giữa đặc trưng và nhãn.

2.6.6 Lợi ích khi kết hợp sử dụng bốn mô hình

- **Tăng độ tin cậy:**

- Khi một đặc trưng được nhiều phương pháp khác nhau lựa chọn (ví dụ: vừa được Random Forest đánh giá cao, vừa có điểm Mutual Information lớn), khả năng cao đặc trưng đó thực sự có ý nghĩa quan trọng với mô hình.

- **Khai thác đa chiều mối quan hệ đặc trưng:**

- Random Forest và RFE giúp phát hiện các đặc trưng quan trọng qua mô hình học máy, kể cả các mối quan hệ phi tuyến và tương tác giữa các đặc trưng.
- Mutual Information phát hiện các đặc trưng có mối quan hệ phi tuyến với biến mục tiêu mà mô hình cây có thể bỏ qua.
- SelectKBest nhanh chóng lọc ra các đặc trưng có liên quan trực tiếp về mặt thống kê với biến mục tiêu.

- **Giảm nguy cơ Overfitting:**

- Một số phương pháp như RFE có thể bị overfitting nếu không được điều

chính cẩn thận. Kết hợp với các phương pháp lọc như Mutual Information và SelectKBest sẽ giúp giảm thiểu rủi ro này.

- **Hiệu suất cao hơn khi huấn luyện mô hình:**

- Khi chỉ giữ lại những đặc trưng quan trọng nhất, mô hình sẽ huấn luyện nhanh hơn và chính xác hơn do giảm được nhiễu từ các đặc trưng không liên quan.

2.6.7 Chọn đặc trưng dựa trên phiếu bầu (Voting-based Feature Selection)

Sử dụng Voting-based Feature Selection (Chọn đặc trưng dựa trên phiếu bầu) để chọn được những đặc trưng xuất hiện nhiều nhất sau khi gộp kết quả của bốn mô hình bằng cách dựa vào tần suất xuất hiện. Phương pháp này giúp đảm bảo rằng các đặc trưng được chọn không chỉ quan trọng theo một tiêu chí nhất định mà còn có sự nhất quán khi đánh giá bằng nhiều phương pháp khác nhau.

1. Các bước thực hiện

(a) Thu thập kết quả từ 4 phương pháp:

- Lấy danh sách 20 đặc trưng quan trọng nhất từ mỗi phương pháp:
 - Random Forest Feature Importance
 - Mutual Information
 - Recursive Feature Elimination (RFE)
 - SelectKBest

(b) Gộp danh sách đặc trưng:

- Kết hợp các đặc trưng từ 4 phương pháp thành một danh sách tổng hợp.

- Có thể sẽ có các đặc trưng trùng nhau giữa các phương pháp.

(c) **Đếm tần suất xuất hiện của từng đặc trưng:**

(d) **Chọn 20 đặc trưng có tần suất xuất hiện cao nhất:**

- Sắp xếp các đặc trưng theo số lần xuất hiện giảm dần.
- Chọn 20 đặc trưng xuất hiện nhiều nhất. Nếu có nhiều đặc trưng có cùng số lần xuất hiện, có thể ưu tiên dựa trên mức độ quan trọng hoặc đánh giá thử nghiệm.

(e) **Kiểm tra hiệu suất của mô hình với tập đặc trưng được chọn:**

- Sử dụng các đặc trưng đã chọn để huấn luyện mô hình và đánh giá hiệu quả dự đoán bằng các chỉ số như Accuracy, Precision, Recall, và F1-score.

2. Ưu điểm

- **Độ tin cậy cao:** Đặc trưng xuất hiện nhiều lần thường là những đặc trưng thực sự có giá trị, giúp cải thiện khả năng dự đoán của mô hình.
- **Giảm thiểu rủi ro bỏ sót đặc trưng quan trọng:** Sự đồng thuận từ nhiều phương pháp làm giảm khả năng bỏ sót những đặc trưng quan trọng mà một phương pháp đơn lẻ có thể bỏ qua.
- **Cân bằng giữa tính đơn giản và hiệu quả:** Thay vì chỉ phụ thuộc vào một phương pháp phức tạp, bạn kết hợp nhiều phương pháp để có cái nhìn toàn diện nhưng vẫn giữ được số lượng đặc trưng ở mức tối ưu.

3. Nhược điểm

- **Có thể giữ lại đặc trưng dư thừa:** Một số đặc trưng có thể được chọn do tần suất cao nhưng lại không đóng góp nhiều vào hiệu suất mô hình.

Để khắc phục, bạn có thể thử nghiệm thêm với việc loại bỏ một vài đặc trưng ít quan trọng sau khi kiểm tra hiệu suất.

- **Tổn thời gian xử lý ban đầu:** Việc chạy và kết hợp nhiều phương pháp có thể mất thêm thời gian, nhưng bù lại bạn sẽ có kết quả tối ưu hơn.

CHƯƠNG 3 - NỘI DUNG NGHIÊN CỨU

Dựa trên cơ sở lý thuyết đã tìm hiểu, cũng như những nghiên cứu khoa học đã được thực hiện, kế thừa những tư liệu đang có, dự án phân tích bộ dữ liệu và thực hiện huấn luyện trên các mô hình học máy và học sâu. Chương này làm rõ cách triển khai và đánh giá nghiên cứu.

3.1 Bộ dữ liệu

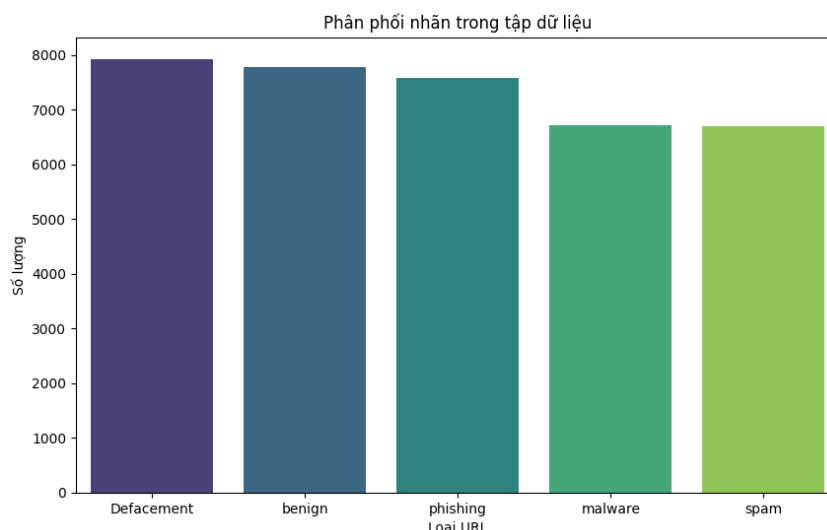
Bộ dữ liệu **ISCX-URL-2016** được phát triển bởi Viện An ninh mạng Canada (Canadian Institute for Cybersecurity) nhằm hỗ trợ nghiên cứu và phát triển các phương pháp phát hiện URL độc hại. Bộ dữ liệu này phản ánh các hoạt động tấn công mạng phổ biến thông qua việc sử dụng URL độc hại.

- **Thành phần của bộ dữ liệu:**

- **URL lành tính (Benign URLs):** Hơn 35.300 URL được thu thập từ các trang web hàng đầu theo xếp hạng của Alexa. Các tên miền này được thu thập thông qua công cụ Heritrix web crawler để trích xuất các URL. Ban đầu, khoảng nửa triệu URL duy nhất được thu thập và sau đó loại bỏ các URL trùng lặp và chỉ chứa tên miền.
- **URL độc hại (Malicious URLs):** Bao gồm các loại URL sau:
 - * **Defacement:** URL dẫn đến các trang web bị thay đổi giao diện hoặc nội dung bởi tin tặc.
 - * **Malware:** URL phân phối phần mềm độc hại nhằm xâm nhập và gây hại cho hệ thống người dùng.
 - * **Phishing:** URL giả mạo các trang web hợp pháp nhằm lừa đảo và thu thập thông tin nhạy cảm của người dùng.

- * **Spam:** URL được sử dụng trong các chiến dịch gửi thư rác, thường nhằm quảng cáo hoặc phát tán nội dung không mong muốn.

Với các nhãn là các loại URL độc hại và URL bình thường, chúng ta có bảng phân phối nhãn như sau:



Hình 15: Phân phối nhãn trong tập dữ liệu

Bộ dữ liệu ISCX-URL-2016 cung cấp một nguồn tài nguyên cho cộng đồng nghiên cứu an ninh mạng, đặc biệt trong việc phát triển và đánh giá các mô hình học máy và học sâu nhằm phát hiện và phân loại URL độc hại. Việc sử dụng bộ dữ liệu này đã được đề cập trong nhiều nghiên cứu và dự án khác nhau.

3.2 Khai phá bộ dữ liệu

- Để hiểu rõ cấu trúc, phân bố và đặc điểm của dữ liệu trước khi áp dụng các phương pháp phân tích hoặc mô hình học máy. Với bộ dữ liệu ISCX-URL-2016, quá trình khai phá giúp chúng ta xác định:
 - Số lượng và phân bố của các loại URL (benign/malicious).
 - Các đặc trưng chính của URL có thể được sử dụng trong mô hình học máy.

- Các xu hướng và đặc điểm quan trọng trong tập dữ liệu.

3.2.1 Phân tích đặc trưng của URL

- **Đặc trưng dựa trên cú pháp URL (Lexical Features)**

- Các đặc trưng về cú pháp URL có thể giúp nhận diện URL độc hại, bao gồm:
 - * **Độ dài URL:** URL độc hại thường dài hơn URL lành tính.
 - * **Số lượng dấu “.” và “/”:** URL lừa đảo có xu hướng chứa nhiều dấu “/” hơn.
 - * **Số lượng tham số trong query string:** URL spam thường có nhiều tham số.
 - * **Tần suất xuất hiện các từ khóa đáng ngờ:** Các từ như “login”, “verify”, “secure”, “update” thường xuất hiện trong phishing URL.

- **Đặc trưng về tên miền (Host-based Features)**

- **Tuổi đời của tên miền:** URL độc hại thường được đăng ký trong thời gian ngắn.
- **Địa chỉ IP của tên miền:** URL độc hại thường sử dụng IP thay vì tên miền hợp lệ.
- **Tần suất xuất hiện trên danh sách đen:** URL độc hại có thể được phát hiện bằng cách so sánh với danh sách đen (blacklist).

- **Đặc trưng về hành vi (Behavioral Features)**

- **Tốc độ chuyển hướng (Redirects):** URL độc hại thường có nhiều lớp chuyển hướng.

- **Sự thay đổi nội dung trang web:** URL phishing có thể thay đổi giao diện thường xuyên.
- **Mức độ tương tác của người dùng:** URL độc hại có tỷ lệ thoát (bounce rate) cao.

- Ví dụ trực quan:

- Chúng tôi đưa ra ví dụ về một URL:

<http://fanpage.gr/family/gineka/%ce%b1%cf%85%cf%84%ce%bf%ce%af-%ce%b5%ce%bf%ce%b9-13-%cf%89%cf%81%ce%b1%ce%b9%cf%8c%cf%84%ce%b5%cf%84%ce%ad%ce%bb%ce%bb%ce%b7%ce%bd%ce%b5%cf%82-%ce%b1%ce%bd-2/>

- Chúng tôi phân tích URL trên theo ba nhóm đặc trưng:
 - Đặc trưng dựa trên cú pháp URL (Lexical Features)
 - Đặc trưng về tên miền (Host-based Features)
 - Đặc trưng về hành vi (Behavioral Features)
- Trong dự án này, chúng tôi sẽ phân tích đặc trưng dựa trên cú pháp URL để lấy những thành phần quan trọng để huấn luyện.

Đặc trưng dựa trên cú pháp	
Độ dài URL	161 ký tự
Số lượng dấu /	6
Số lượng dấu .	2
Số lượng dấu -	3
Số lượng dấu % (mã hóa UTF-8)	34
Số lượng từ trong đường dẫn	15
Độ dài từ trung bình trong đường dẫn	~6.2 ký tự
Chứa từ đáng ngờ?	Không có từ khóa đáng ngờ (như "login", "verify", "update", v.v.)
Đặc trưng về tên miền	
Tên miền	fanpage.gr
Độ dài tên miền	10 ký tự
Số lượng dấu . trong miền	1
Tên miền cấp cao nhất (TLD)	.gr (Hy Lạp)
Tên miền con (subdomain)	Không
Tên miền có trong danh sách trắng?	Không rõ
Tuổi đời tên miền	Cần kiểm tra WHOIS
Tên miền có xuất hiện trong danh sách blacklist?	Cần kiểm tra trên các dịch vụ như Phishtank, VirusTotal
Đặc trưng về hành vi	
URL có chuyển hướng (redirect)?	Cần kiểm tra
Trang web có yêu cầu thông tin nhạy cảm?	Cần kiểm tra
Trang web có chứa mã độc hay không?	Cần kiểm tra trên VirusTotal
Có sử dụng chứng chỉ SSL (HTTPS)?	Không (HTTP)
Lưu lượng truy cập trang web	Cần kiểm tra
Nội dung trang web có hợp lệ không?	Cần kiểm tra

Bảng 3: Ví dụ phân tích đặc trưng URL (1)

- URL này có một số yếu tố cần kiểm tra thêm (mã hóa UTF-8, không có HTTPS).
- Có thể kiểm tra trên VirusTotal, PhishTank, hoặc Google Safe Browsing để xác minh thêm.
- Nếu URL này dẫn đến một trang web có nội dung lạ hoặc yêu cầu nhập thông tin cá nhân, cần cảnh giác.

Trong dự án này, qua việc phân tích bộ dữ liệu ISCX-URL-2016, chúng tôi nhận thấy các đặc trưng URL tương ứng với các đặc trưng của bộ dữ liệu như sau:

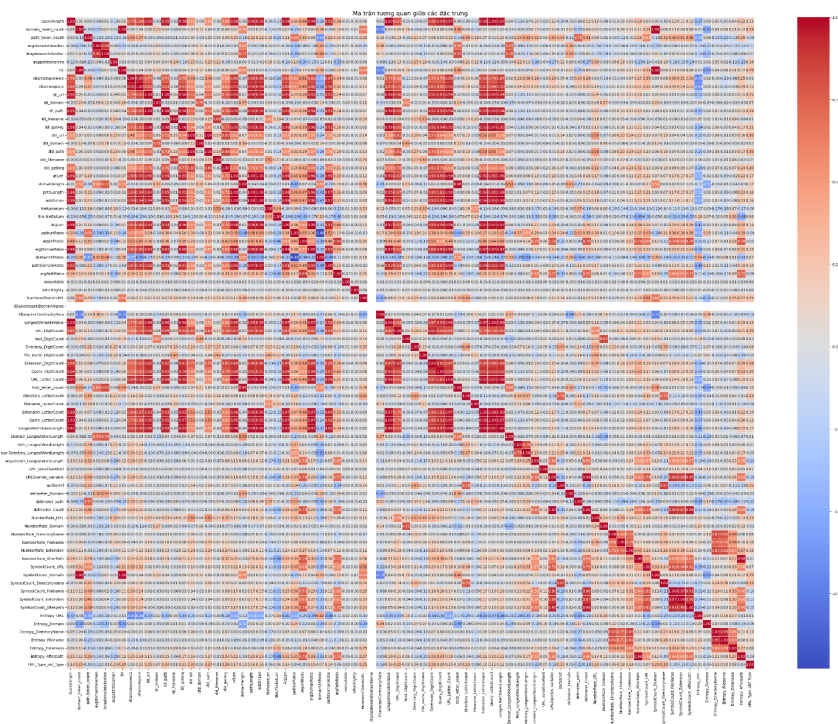
Nhóm	Đặc trưng giữ lại từ bộ dữ liệu
Lexical	ldl_url, domain_token_count, path_token_count, avgdomain-tokenlen, longdomaintokenlen, avgpathtokenlen, ldl_filename, ldl_getArg, tld
Host-based	ldl_domain, tld
Behavioral	Không có

Bảng 4: Ví dụ phân tích đặc trưng URL (2)

Vì trong bộ dữ liệu này, Behavioral Features không có sẵn, chúng tôi tập trung vào Lexical Features và Host-based Features. Hướng tiếp cận hiện tại là huấn luyện mô hình với các đặc trưng hiện có và sau đó thử nghiệm thêm đặc trưng mới để đánh giá mức độ cải thiện.

3.2.2 Trực quan hóa dữ liệu

- Để dữ liệu được phân tích một cách đầy đủ và trực quan nhất, chúng tôi đã thực hiện trực quan hóa dữ liệu theo các phương pháp sau:
- Sử dụng ma trận tương quan để trực quan hóa mối quan hệ giữa các đặc trưng:

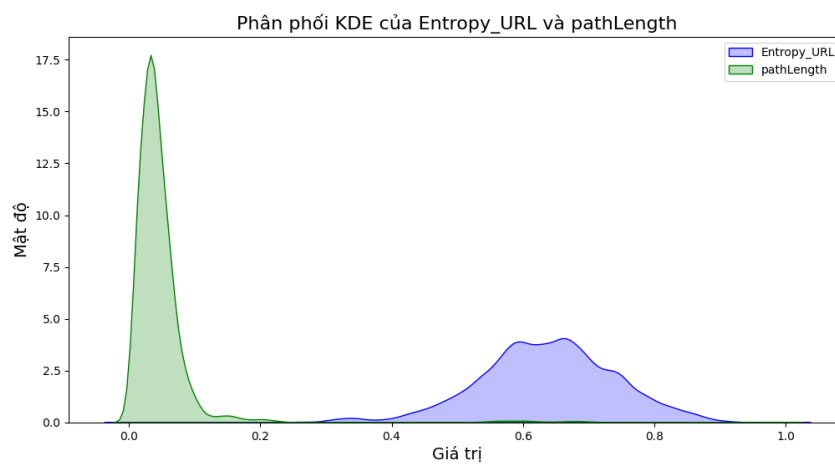


Hình 16: Ma trận tương quan

- Dựa trên ma trận tương quan giữa các đặc trưng trong bộ dữ liệu URL, hiển thị mối quan hệ giữa các biến dưới dạng hệ số tương quan Pearson, có thể nhận xét như sau:
 - Dọc theo đường chéo chính (từ góc trên bên trái đến góc dưới bên phải) là các giá trị tương quan hoàn hảo ($=1.0$), vì đây là mối quan hệ của một biến với chính nó.
 - Một số đặc trưng có mối tương quan chặt chẽ với nhau, đặc biệt là các đặc trưng liên quan đến cú pháp URL, độ dài, số lượng ký tự đặc biệt, entropy, ...
 - Các đặc trưng liên quan đến độ dài URL, số lượng dấu chấm ($.$), dấu gạch chéo ($/$), ký tự đặc biệt như $@$, $_$, $\%$, ... thường có tương quan cao với nhau, điều này có thể giúp phân biệt URL lành tính và độc hại.
 - Một số đặc trưng về entropy và sự phức tạp của tên miền có mối tương quan mạnh với nhau, điều này có thể cho thấy URL độc hại thường có cấu trúc tên miền phức tạp.

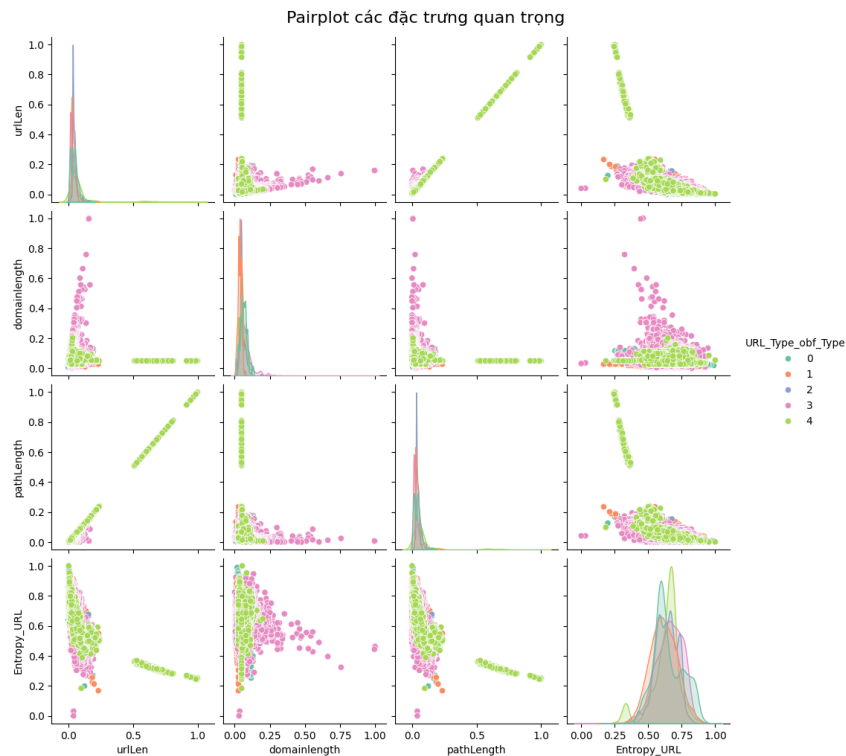
trúc phức tạp hơn để tránh bị phát hiện.

- Các đặc trưng về tần suất xuất hiện ký tự hoặc từ khóa có mối liên hệ nhất định với các đặc trưng về cấu trúc URL.
- Một số quan sát khác từ dữ liệu:
 - Hiển thị phân phối của các cột Entropy_URL, pathLength, ... bằng KDE để quan sát dạng dữ liệu:



Hình 17: Phân phối KDE của Entropy_URL, pathLength

– Trực quan hóa mối quan hệ giữa nhiều đặc trưng quan trọng với nhãn:



Hình 18: Phân phối các đặc trưng quan trọng

3.3 Lựa chọn đặc trưng

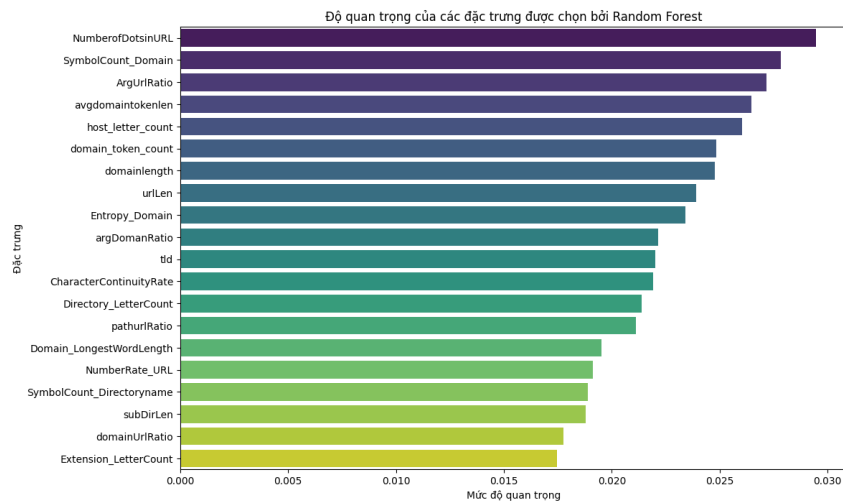
Có thể thấy, việc lựa chọn các đặc trưng trong bộ dữ liệu đóng vai trò quan trọng trong việc cho ra kết quả bài toán. Chính vì vậy, chúng tôi tập trung vào việc lựa chọn các đặc trưng liên quan nhất của bộ dữ liệu để tiến hành huấn luyện. Sau đó, chúng tôi sẽ so sánh với kết quả huấn luyện của bộ dữ liệu gốc.

Trong dự án này, dựa vào các phân tích trên, chúng tôi sẽ thực nghiệm chọn 20 đặc trưng trong bộ dữ liệu để phân tích. Với mỗi phương pháp, chúng tôi sẽ cho ra mỗi bộ dữ liệu khác nhau.

3.3.1 Random Forest (Tree-based Feature Importance)

Random Forest cung cấp thông tin quan trọng của đặc trưng dựa trên độ giảm impurity (Gini hoặc entropy) qua tất cả các cây trong mô hình.

Phù hợp với cả dữ liệu phi tuyến và có thể xử lý mối quan hệ phức tạp giữa các đặc trưng.

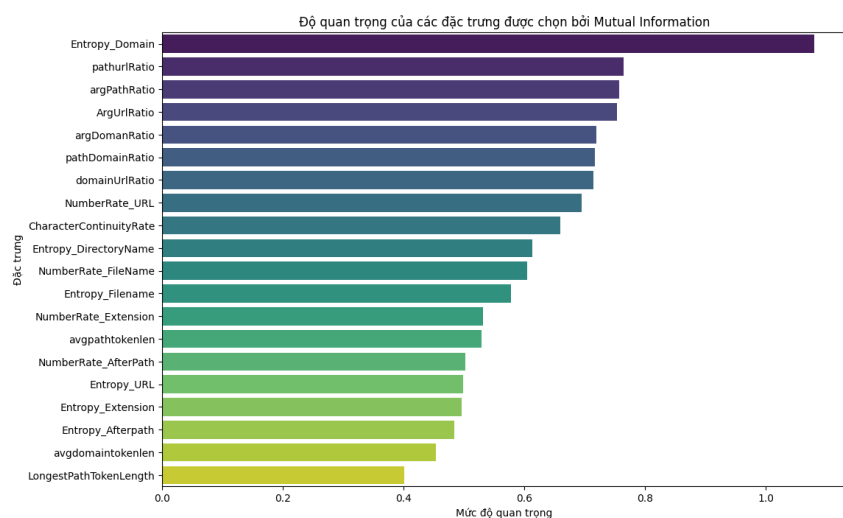


Hình 19: Lựa chọn đặc trưng bằng Random Forest

3.3.2 Mutual Information

Mutual Information đo lường sự phụ thuộc (phi tuyến) giữa các đặc trưng và nhãn đầu ra.

Có thể phát hiện các đặc trưng quan trọng mà các phương pháp dựa trên impurity (như Random Forest) bỏ qua.

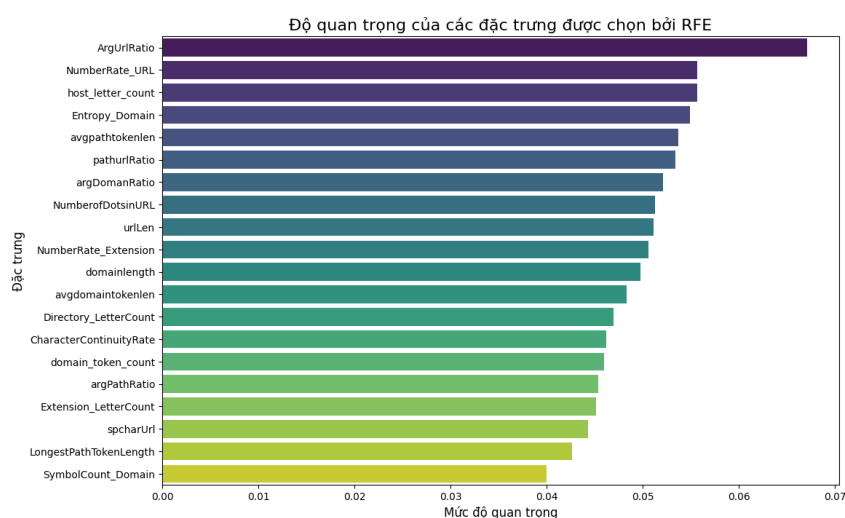


Hình 20: Lựa chọn đặc trưng bằng Mutual Information

3.3.3 Recursive Feature Elimination (RFE)

RFE giúp chọn các đặc trưng bằng cách lặp lại quá trình huấn luyện mô hình và loại bỏ những đặc trưng kém quan trọng nhất.

Phương pháp này tối ưu hóa dựa trên một mô hình cụ thể (ví dụ: Random Forest hoặc Logistic Regression).



Hình 21: Lựa chọn đặc trưng bằng Recursive Feature Elimination

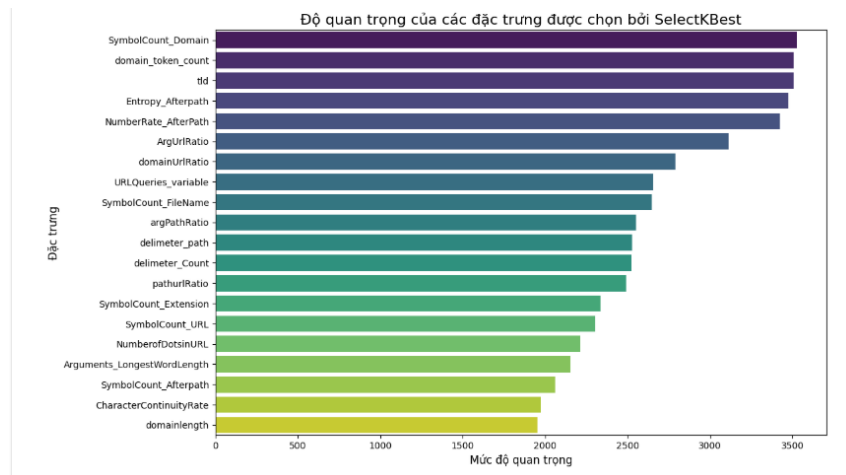
3.3.4 SelectKBest

SelectKBest tính toán điểm số thống kê giữa từng đặc trưng và nhãn (label). Sau đó, nó chọn ra **K đặc trưng có điểm số cao nhất** để giữ lại.

Các đặc trưng còn lại sẽ bị loại bỏ khỏi tập dữ liệu huấn luyện.

3.3.5 Kết hợp 4 phương pháp

- Chúng tôi cũng tiến hành thực nghiệm bằng cách kết hợp bốn phương pháp lựa chọn đặc trưng gồm Random Forest, Mutual Information, Recursive Feature Elimination (RFE), và SelectKBest. Việc này nhằm tận dụng ưu điểm của từng phương pháp, từ đó tạo ra một bộ đặc trưng tối ưu hơn cho mô hình.



Hình 22: Lựa chọn đặc trưng bằng SelectKBest

- Sau quá trình chọn lọc, chúng tôi thu được 20 đặc trưng quan trọng nhất, bao gồm:

- ArgUrlRatio
- CharacterContinuityRate
- athurlRatio
- NumberofDotsinURL
- SymbolCount_Domain
- avgdomaintokenlen
- domain_token_count
- domainlength
- Entropy_Domain
- argDomanRatio
- NumberRate_URL
- domainUrlRatio
- argPathRatio
- host_letter_count

- urlLen
- tld
- Directory_LetterCount
- Extension_LetterCount
- NumberRate_Extension
- avgpathtokenlen

3.4 Huấn luyện mô hình

Với các bộ dữ liệu có được, chúng tôi tiến hành huấn luyện thông qua các mô hình học máy và học sâu. Mỗi bộ dữ liệu được chia 80% dùng để huấn luyện và 20% dùng để kiểm tra.

- Quy ước các bộ dữ liệu selt-buit như sau:
 - Bộ dữ liệu 1: Kết quả từ phương pháp Random Forest
 - Bộ dữ liệu 2: Kết quả từ phương pháp Mutual Information
 - Bộ dữ liệu 3: Kết quả từ phương pháp Recursive Feature Elimination
 - Bộ dữ liệu 4: Kết quả từ phương pháp SelectKBest
 - Bộ dữ liệu 5: Kết quả của 4 phương pháp Random Forest, Mutual Information, Recursive Feature Elimination, SelectKBest
 - Bộ dữ liệu 6: Kết quả của 4 phương pháp Random Forest, Mutual Information, Recursive Feature Elimination, SelectKBest trích xuất 60 đặc trưng
- Các mô hình sử dụng:
 - Học máy: Logistic Regression, Random Forest, XGBoost (Extreme Gradient Boosting)

- Học sâu: MLP (Multi-Layer Perceptron), LSTM (Long Short-Term Memory)

Trong mô hình LSTM, chúng tôi sử dụng 4 lớp chính: **Embedding Layer** để chuyển đổi từ thành vector 128 chiều (giúp mô hình hiểu ngữ nghĩa). Tiếp theo, **LSTM Layer** học mối quan hệ trong chuỗi dữ liệu, trích xuất đặc trưng thời gian. Với **Dense Layer (64 neurons, ReLU)**, chúng tôi sử dụng hàm kích hoạt ReLU giúp mô hình học các biểu diễn phi tuyến tính tốt hơn, giảm chiều dữ liệu, giúp mô hình học tốt hơn. **Output Layer (Softmax)**: dùng hàm softmax để chuyển đổi kết quả thành xác suất cho từng lớp đầu ra.

3.4.1 Huấn luyện với bộ dữ liệu gốc

Logistic Regression

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		79%	78%	78%
Benign		79%	87%	83%
Malware		75%	60%	67%
Phishing		77%	85%	80%
Spam		87%	85%	86%
Tổng thể	79%			

Bảng 5: Huấn luyện bộ dữ liệu gốc với Logistic Regression

Random Forest

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		99%	98%	99%
Benign		98%	99%	98%
Malware		100%	98%	99%
Phishing		94%	96%	95%
Spam		99%	97%	98%
Tổng thể	98%			

Bảng 6: Huấn luyện bộ dữ liệu gốc với Random Forest

XGBoost

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		99%	99%	99%
Benign		98%	99%	99%
Malware		99%	99%	99%
Phishing		96%	97%	97%
Spam		100%	98%	99%
Tổng thể	99%			

Bảng 7: Huấn luyện bộ dữ liệu gốc với XGBoost

MLP

Chỉ số	Giá trị
Train Accuracy	92.98%
Train Loss	0.2064
Validation Accuracy	95.56%
Validation Loss	0.1580

Bảng 8: Huấn luyện bộ dữ liệu gốc với MLP

LSTM

Chỉ số	Giá trị
Train Accuracy	94.40%
Train Loss	0.1581
Validation Accuracy	91.64%
Validation Loss	0.2949

Bảng 9: Huấn luyện bộ dữ liệu gốc với LSTM

- **Kết luận:**
 - **Logistic Regression** đạt độ chính xác tổng thể 79%, nhưng hiệu suất không đồng đều giữa các loại URL. Đặc biệt, mô hình gặp khó khăn trong việc phát hiện malware (F1-score: 67%), cho thấy hạn chế của mô hình tuyến tính khi xử lý dữ liệu phức tạp.
 - **Random Forest** cải thiện đáng kể độ chính xác, đạt 98% tổng thể, với

khả năng phân loại chính xác các loại URL độc hại, đặc biệt là malware và spam với F1-score lên đến 99%.

- **XGBoost** tiếp tục nâng cao hiệu suất, đạt độ chính xác tổng thể 98%, thể hiện khả năng học tốt hơn nhờ việc tối ưu hóa các cây quyết định và kiểm soát overfitting hiệu quả.
- **MLP (Multi-Layer Perceptron)** là một phương pháp Deep Learning cho kết quả ấn tượng với Validation Accuracy đạt 95.56%. Mặc dù không đạt mức chính xác như XGBoost, nhưng mô hình này có tiềm năng cao trong việc học đặc trưng phi tuyến tính và có thể cải thiện hơn nữa bằng cách tinh chỉnh siêu tham số hoặc thử nghiệm với các kiến trúc phức tạp hơn.
- **LSTM** cho kết quả bất ngờ với độ chính xác lên đến 94%.

3.4.2 Huấn luyện với bộ dữ liệu 1

Logistic Regression

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		69%	72%	70%
Benign		72%	86%	78%
Malware		61%	35%	45%
Phishing		63%	72%	67%
Spam		70%	70%	70%
Tổng thể	78%			

Bảng 10: Huấn luyện bộ dữ liệu 1 với Logistic Regression

Random Forest

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		98%	98%	98%
Benign		97%	99%	98%
Malware		99%	99%	99%
Phishing		95%	95%	95%
Spam		99%	98%	98%
Tổng thể	98%			

Bảng 11: Huấn luyện bộ dữ liệu 1 với Random Forest

XGBoost

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		99%	98%	99%
Benign		98%	99%	99%
Malware		98%	99%	98%
Phishing		96%	95%	95%
Spam		99%	99%	99%
Tổng thể	98%			

Bảng 12: Huấn luyện bộ dữ liệu 1 với XGBoost

MLP

Chỉ số	Giá trị
Train Accuracy	88.43%
Train Loss	0.3182
Validation Accuracy	92.47%
Validation Loss	0.2169

Bảng 13: Huấn luyện bộ dữ liệu 1 với MLP

LSTM

Chỉ số	Giá trị
Train Accuracy	84.27%
Train Loss	0.3665
Validation Accuracy	83.55%
Validation Loss	0.4219

Bảng 14: Huấn luyện bộ dữ liệu 1 với LSTM

3.4.3 Huấn luyện với bộ dữ liệu 2

Logistic Regression

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		65%	72%	69%
Benign		72%	79%	75%
Malware		40%	27%	33%
Phishing		65%	72%	68%
Spam		70%	66%	68%
Tổng thể	64%			

Bảng 15: Huấn luyện bộ dữ liệu 2 với Logistic Regression

Random Forest

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		98%	98%	98%
Benign		97%	99%	98%
Malware		99%	98%	99%
Phishing		93%	96%	95%
Spam		99%	96%	98%
Tổng thể	97%			

Bảng 16: Huấn luyện bộ dữ liệu 2 với Random Forest

XGBoost

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		99%	98%	99%
Benign		98%	99%	98%
Malware		99%	98%	99%
Phishing		94%	96%	95%
Spam		99%	98%	98%
Tổng thể	98%			

Bảng 17: Huấn luyện bộ dữ liệu 2 với XGBoost

MLP

Chỉ số	Giá trị
Train Accuracy	87.94%
Train Loss	0.3450
Validation Accuracy	91.71%
Validation Loss	0.2313

Bảng 18: Huấn luyện bộ dữ liệu 2 với MLP

LSTM

Chỉ số	Giá trị
Train Accuracy	79.23%
Train Loss	0.5256
Validation Accuracy	76.56%
Validation Loss	0.5882

Bảng 19: Huấn luyện bộ dữ liệu 2 với LSTM

3.4.4 Huấn luyện với bộ dữ liệu 3

Logistic Regression

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		68%	72%	70%
Benign		71%	83%	77%
Malware		47%	31%	37%
Phishing		65%	73%	69%
Spam		71%	68%	69%
Tổng thể	66%			

Bảng 20: Huấn luyện bộ dữ liệu 3 với Logistic Regression

Random Forest

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		98%	98%	98%
Benign		98%	99%	98%
Malware		99%	98%	98%
Phishing		94%	95%	95%
Spam		100%	97%	98%
Tổng thể	98%			

Bảng 21: Huấn luyện bộ dữ liệu 3 với Random Forest

XGBoost

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		99%	99%	99%
Benign		98%	99%	99%
Malware		98%	99%	99%
Phishing		96%	95%	95%
Spam		99%	99%	99%
Tổng thể	98%			

Bảng 22: Huấn luyện bộ dữ liệu 3 với XGBoost

MLP

Chỉ số	Giá trị
Train Accuracy	88.77%
Train Loss	0.3185
Validation Accuracy	92.22%
Validation Loss	0.2225

Bảng 23: Huấn luyện bộ dữ liệu 3 với MLP

LSTM

Chỉ số	Giá trị
Train Accuracy	84.08%
Train Loss	0.3861
Validation Accuracy	83.03%
Validation Loss	0.4465

Bảng 24: Huấn luyện bộ dữ liệu 3 với LSTM

3.4.5 Huấn luyện với bộ dữ liệu 4

Logistic Regression

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		66%	69%	68%
Benign		71%	84%	77%
Malware		51%	28%	36%
Phishing		62%	78%	69%
Spam		74%	65%	69%
Tổng thể	66%			

Bảng 25: Huấn luyện bộ dữ liệu 4 với Logistic Regression

Random Forest

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		97%	97%	97%
Benign		97%	98%	97%
Malware		97%	98%	98%
Phishing		93%	93%	93%
Spam		98%	97%	97%
Tổng thể	97%			

Bảng 26: Huấn luyện bộ dữ liệu 4 với Random Forest

XGBoost

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		98%	97%	97%
Benign		97%	97%	97%
Malware		96%	97%	97%
Phishing		93%	92%	93%
Spam		98%	98%	98%
Tổng thể	96%			

Bảng 27: Huấn luyện bộ dữ liệu 4 với XGBoost

MLP

Chỉ số	Giá trị
Train Accuracy	86.38%
Train Loss	0.3658
Validation Accuracy	89.50%
Validation Loss	0.2767

Bảng 28: Huấn luyện bộ dữ liệu 4 với MLP

LSTM

Chỉ số	Giá trị
Train Accuracy	76.09%
Train Loss	0.5582
Validation Accuracy	75.07%
Validation Loss	0.5990

Bảng 29: Huấn luyện bộ dữ liệu 4 với LSTM

3.4.6 Huấn luyện với bộ dữ liệu 5

Logistic Regression

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		70%	71%	70%
Benign		70%	84%	77%
Malware		50%	32%	39%
Phishing		65%	72%	68%
Spam		71%	70%	71%
Tổng thể	67%			

Bảng 30: Huấn luyện bộ dữ liệu 5 với Logistic Regression

Random Forest

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		98%	98%	98%
Benign		98%	99%	98%
Malware		99%	98%	98%
Phishing		95%	95%	95%
Spam		99%	97%	98%
Tổng thể	98%			

Bảng 31: Huấn luyện bộ dữ liệu 5 với Random Forest

XGBoost

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		99%	99%	99%
Benign		98%	99%	99%
Malware		98%	99%	99%
Phishing		97%	95%	96%
Spam		99%	99%	99%
Tổng thể	98%			

Bảng 32: Huấn luyện bộ dữ liệu 5 với XGBoost

MLP

Chỉ số	Giá trị
Train Accuracy	88.88%
Train Loss	0.3111
Validation Accuracy	92.30%
Validation Loss	0.2188

Bảng 33: Huấn luyện bộ dữ liệu 5 với MLP

LSTM

Chỉ số	Giá trị
Train Accuracy	80.02%
Train Loss	0.4762
Validation Accuracy	78.64%
Validation Loss	0.5540

Bảng 34: Huấn luyện bộ dữ liệu 5 với LSTM

3.4.7 Huấn luyện với bộ dữ liệu 6

Logistic Regression

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		76%	74%	75%
Benign		76%	80%	78%
Malware		71%	54%	61%
Phishing		68%	83%	75%
Spam		78%	74%	76%
Tổng thể	67%			

Bảng 35: Huấn luyện bộ dữ liệu 6 với Logistic Regression

Random Forest

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		98%	97%	98%
Benign		97%	98%	97%
Malware		99%	98%	98%
Phishing		92%	96%	94%
Spam		99%	95%	97%
Tổng thể	98%			

Bảng 36: Huấn luyện bộ dữ liệu 6 với Random Forest

XGBoost

Loại URL	Accuracy	Precision	Recall	F1-score
Defacement		98%	98%	98%
Benign		97%	98%	98%
Malware		98%	99%	98%
Phishing		95%	95%	95%
Spam		99%	96%	98%
Tổng thể	98%			

Bảng 37: Huấn luyện bộ dữ liệu 6 với XGBoost

MLP

Chỉ số	Giá trị
Train Accuracy	90.09%
Train Loss	0.2611
Validation Accuracy	93.49%
Validation Loss	0.2108

Bảng 38: Huấn luyện bộ dữ liệu 6 với MLP

LSTM

Chỉ số	Giá trị
Train Accuracy	86.72%
Train Loss	0.3405
Validation Accuracy	84.72%
Validation Loss	0.4216

Bảng 39: Huấn luyện bộ dữ liệu 6 với LSTM

CHƯƠNG 4 - KẾT LUẬN VÀ KIẾN NGHỊ

Sau khi hoàn thành các thực nghiệm, chương 4 sẽ trình bày các kết luận, giá trị đạt được, và đề xuất các hướng phát triển trong tương lai.

4.1 Kết luận

Dựa trên bộ dữ liệu ISCX-URL-2016, chúng tôi thực hiện một so sánh trực tiếp giữa các mô hình học sâu phổ biến (Multi-Layer Perceptron, Long Short-Term Memory) và các thuật toán học máy (Logistic Regression, Random Forest, XGBoost) trong việc phát hiện và phân loại các URL độc hại. Đồng thời, chúng tôi mở rộng nghiên cứu bằng cách sử dụng bộ dữ liệu công khai ISCX-URL-2016, chứa các đặc trưng từ vựng của các URL độc hại. Chúng tôi đã phân tích và chia bộ dữ liệu thành các nhóm con khác nhau để có cái nhìn chi tiết và khách quan hơn.

Với 6 bộ dữ liệu tự tạo, khi huấn luyện các mô hình học máy và học sâu nói trên, chúng tôi nhận thấy rằng XGBoost đạt kết quả cao nhất, lên đến 99% (Mục 3.4) và ổn định hơn so với các mô hình còn lại. Các đặc trưng của URL trong bộ dữ liệu và mối quan hệ giữa chúng đóng vai trò quan trọng, ảnh hưởng đến kết quả phát hiện các URL độc hại. Chúng tôi phát hiện ra rằng một số cặp đặc trưng kết hợp với nhau mang lại kết quả tối ưu hơn (Hình 3.5 Lựa chọn đặc trưng bằng Random Forest) như *avgdomaintokenlen* và *domain_token_count*. Tuy nhiên, điều này cũng cho thấy không phải tất cả các đặc trưng đều quan trọng; một số đặc trưng không liên quan đến việc phân loại có thể gây nhiễu trong quá trình huấn luyện và ảnh hưởng đến kết quả (Bảng 3.24), ví dụ: *domain_token_count* (số lượng token trong domain) với *tld* (TLD) \rightarrow Domain có nhiều token thường liên

quan đến TLD (ví dụ: subdomain dài trong .xyz).

Khi áp dụng các mô hình học máy và học sâu kể trên với bộ dữ liệu gồm 60 đặc trưng (Mục 3.4.7), chúng tôi nhận thấy MLP, LSTM (các mô hình học sâu) đạt được kết quả tốt hơn nhiều so với các bộ dữ liệu khác (Bảng 3.35 và Bảng 3.36).

Ngoài ra, việc áp dụng các kỹ thuật lựa chọn đặc trưng riêng biệt hoặc kết hợp các kỹ thuật này giúp chúng tôi chọn ra các đặc trưng có giá trị tốt hơn, đồng thời có cái nhìn tổng quan hơn về khả năng lựa chọn của các phương pháp. Các kỹ thuật này sử dụng các tiêu chí khác nhau để lựa chọn những đặc trưng có liên quan đến việc phát hiện và phân loại nhãn. Số lượng đặc trưng cũng đóng vai trò quan trọng, vì vậy, chúng tôi đã tiến hành chia bộ dữ liệu với các số lượng đặc trưng khác nhau để kiểm tra ảnh hưởng của chúng.

4.2 Ưu điểm

Dự án này đã đạt được những thành công ban đầu đáng khích lệ nhờ vào cách tiếp cận từ bộ dữ liệu ISCX-URL-2016. Việc so sánh trực tiếp giữa các mô hình học sâu (Multi-Layer Perceptron, Long Short-Term Memory) và các thuật toán học máy (Logistic Regression, Random Forest, XGBoost) giúp chúng tôi đánh giá rõ ràng hiệu quả của từng mô hình trong việc phát hiện và phân loại các URL độc hại. Điều này cung cấp cái nhìn tổng thể về ưu nhược điểm của mỗi phương pháp, giúp lựa chọn mô hình phù hợp nhất cho bài toán.

Bộ dữ liệu ISCX-URL-2016 là một bộ dữ liệu công khai, giúp đảm bảo tính minh bạch và khả năng tái sử dụng của nghiên cứu. Việc sử dụng bộ dữ liệu này cũng cho phép so sánh kết quả nghiên cứu của mình với các nghiên cứu khác đã thực hiện trên cùng bộ dữ liệu. Việc sử dụng các đặc trưng từ vệt của URL độc hại làm tăng khả năng phát hiện và phân loại chính xác các URL. Các đặc trưng

này có thể cung cấp thông tin sâu sắc về cấu trúc của URL, hỗ trợ mô hình học máy và học sâu trong việc phân loại.

Khi phân tích và chia bộ dữ liệu thành 6 bộ khác nhau, với mỗi bộ sử dụng các kỹ thuật lựa chọn đặc trưng khác nhau (Random Forest, Mutual Information, Recursive Feature Elimination, và SelectKBest) chúng tôi làm rõ sự ảnh hưởng của mỗi kỹ thuật chọn đặc trưng đến kết quả mô hình. Điều này cung cấp cái nhìn sâu sắc về khả năng lựa chọn các đặc trưng quan trọng đối với vấn đề phát hiện URL độc hại.

4.3 Nhược điểm

Việc thực hiện so sánh trực tiếp giữa nhiều mô hình học sâu và thuật toán học máy đòi hỏi một quá trình phân tích phức tạp, đặc biệt là khi phải kết hợp nhiều kỹ thuật lựa chọn đặc trưng và bộ dữ liệu khác nhau. Điều này làm cho quá trình nghiên cứu trở nên tốn thời gian và dễ gây nhầm lẫn nếu không quản lý tốt các biến số.

Mặc dù sử dụng nhiều phương pháp lựa chọn đặc trưng giúp giảm thiểu sự can thiệp của các đặc trưng không liên quan, nhưng vẫn có thể có một số đặc trưng không thực sự ảnh hưởng đến việc phân loại. Việc loại bỏ những đặc trưng này không phải lúc nào cũng dễ dàng và có thể ảnh hưởng đến kết quả của mô hình nếu không thực hiện chính xác.

Mặc dù ISCX-URL-2016 là bộ dữ liệu công khai phổ biến, nhưng kết quả nghiên cứu có thể không hoàn toàn tổng quát hóa được cho các bộ dữ liệu khác. Các kết quả chỉ có thể áp dụng cho các bài toán tương tự nếu có những yếu tố tương đồng về dữ liệu.

4.4 Hướng phát triển

Tương tự Ucar và cộng sự, chúng tôi thực hiện bài toán trên bộ dữ liệu ISCX-URL-2016, nhưng sử dụng nhiều cách tiếp cận khác nhau. Chúng tôi định hướng trong tương lai sẽ mở rộng bài toán trên bộ dữ liệu tự xây dựng. Điều này sẽ phù hợp hơn với xu hướng phát triển khi ngày càng có nhiều cách tấn công người dùng tinh vi dưới một URL có vẻ như vô hại. Chúng tôi sẽ kế thừa các đặc trưng ngữ nghĩa được phân tích như bộ dữ liệu ISCX-URL-2016, có tính chỉnh các đặc trưng như đã thực hiện ở dự án này, từ đó cho ra một bộ dữ liệu đáp ứng nhu cầu phát hiện – phân loại URL độc hại và giảm thiểu sai sót.

Bên cạnh đó, chúng tôi hướng cách tiếp cận tiếp theo trong việc phân tích đặc trưng của URL ở đặc trưng về tên miền (Host-based Features) và đặc trưng về hành vi (Behavioral Features). Điều này giúp cho bài toán trở nên đa dạng và được phân tích một cách sâu sắc hơn.

TÀI LIỆU THAM KHẢO

- [1] (2017). Retrieved from Machine Learning cơ bản. [Truy cập ngày 28/01/2025].
<https://machinelearningcoban.com/2017/01/27/logisticregression/-mot-vai-tinh-chat-cua-logistic-regression>
- [2] (2024). Retrieved from oriagency. [Truy cập ngày 03/02/2025].
<https://oriagency.vn/url-la-gi/>
- [3] Toan, N. T. (2018). Retrieved from Viblo. [Truy cập ngày 08/02/2025].
<https://viblo.asia/p/gioi-thieu-ve-deep-learning-deep-learning-hoat-dong-nhu-the-nao-3Q75w2nDIWb>
- [4] A. Saleem Raja, R. V. (2021). Lexical features based malicious URL detection using machine learning techniques. Mater. Today, Proc. [Truy cập ngày 03/02/2025].
- [5] Ahammad, S., Kale, S., Upadhye, G., Pande, S., Babu, E., Dhumane, A., & Bahadur, M. (2022). Phishing URL Detection Using Machine Learning Methods. [Truy cập ngày 05/02/2025].
- [6] al., M. A. (2022). International Management Information Systems Conference “Connectedness and Cybersecurity”. IEEE Acess. [Truy cập ngày 05/02/2025].
- [7] Uçar, E. &. (2019). A DEEP LEARNING APPROACH FOR DETECTION OF MALICIOUS URLS. International Management Information Systems Conference “Connectedness and Cybersecurity”. [Truy cập ngày 13/02/2025].

- [8] Alam, M., & Vuong, S. (2013). Random Forest Classification for Detecting Android Malware. 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber. Washington. [Truy cập ngày 08/02/2025].
- [9] Cybersecurity, C. I. (n.d.). URL 2016. [Truy cập ngày 08/02/2025].
- [10] Gupta, B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., & Chang, X. (2021). A Novel Approach for Phishing URLs Detection Using Lexical Based Machine Learning in a Real-Time Environment. [Truy cập ngày 10/02/2025].
- [11] Patil, D. R. (2022). Feature-based malicious URL and attack type detection using multi-class classification. [Truy cập ngày 11/02/2025].
- [12] Sarma, B. a. (2018). Phishing URL detection system based on URL features using SVM. [Truy cập ngày 11/02/2025].
- [13] Shayan Abad, H. G. (2023). Classification of Malicious URLs Using Machine Learning. [Truy cập ngày 11/02/2025].
- [14] Singh, A., Thakur, N., & Sharma, A. (2016). A Review of Supervised Machine Learning Algorithms. n Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). New Delhi. [Truy cập ngày 13/02/2025].