HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN QUANG TUẤN

PHƯƠNG PHÁP NGHIÊN CỨU PHÁT HIỆN MÃ ĐỘC DỰA TRÊN PHÂN TÍCH ĐƯỜNG DẪN SỬ DỤNG HỌC SÂU

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI - 2026

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



NGUYỄN QUANG TUẤN

PHƯƠNG PHÁP NGHIÊN CỨU PHÁT HIỆN MÃ ĐỘC DỰA TRÊN PHÂN TÍCH ĐƯỜNG DẪN SỬ DỤNG HỌC SÂU

CHUYÊN NGÀNH: KHOA HỌC MÁY TÍNH

MÃ SỐ: 8.48.01.01 (Khoa học máy tính)

ĐỀ CƯƠNG TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC: PGS.TS. TRẦN QUANG ANH

HÀ NỘI - 2026

LỜI CẢM ƠN

Trước hết, tôi xin gửi lời cảm ơn chân thành đến PGS.TS Trần Quang Anh đã tận tình định hướng cho tôi trong việc lựa chọn đề tài, đưa ra những nhận xét quý báu và trực tiếp hướng dẫn tôi trong suốt quá trình nghiên cứu và hoàn thành luận văn tốt nghiệp.

Tôi cũng chân thành cảm ơn các thầy cô trong Khoa Công nghệ Thông tin và Khoa Đào tạo Sau Đại học - Học viện Công nghệ Bưu chính Viễn thông đã tận tâm giảng dạy và hỗ trợ tôi trong suốt thời gian học tập và nghiên cứu tại trường.

Tôi xin cảm ơn các cấp Lãnh đạo và toàn thể đồng nghiệp, gia đình, bạn bè đã chia sẻ, giúp đỡ, tạo điều kiện cho tôi hoàn thành luận văn này.

Hà Nội, ngày 29 tháng 7 năm 2026



Nguyễn Quang Tuấn

LỜI CAM ĐOAN

Tôi xin cam đoan Luận văn này là công trình nghiên cứu khoa học nghiêm túc của cá nhân, được thực hiện dưới dự hướng dẫn khoa học của PGS.TS Trần Quang Anh.

Các số liệu, kết quả nghiên cứu và kết luận được trình bày trong Luận văn là trung thực và chưa được công bố dưới bất kỳ hình thức nào.

Tôi xin chịu trách nhiệm về công trình nghiên cứu của mình.

Tác giả Luận văn



Nguyễn Quang Tuấn

MỤC LỤC

MỤC LỤC iii
DANH SÁCH THUẬT NGỮ VIẾT TẮT vi
DANH SÁCH HÌNH ẢNH vii
DANH SÁCH BẢNG BIỂU viii
MỞ ĐẦU 1
1. Lý do chọn đề tài: 1
2. Tổng quan về vấn đề nghiên cứu: 1
3. Mục đích nghiên cứu: 3
4. Đối tượng và phạm vi nghiên cứu: 3
5. Phương pháp nghiên cứu: 4
CHƯƠNG 1: TỔNG QUAN VỀ MÃ ĐỘC VÀ ĐƯỜNG DẪN (PATH FILE) 5
1.1. Tổng quan về mã độc 5
1.1.1. Khái niệm về mã độc 5
1.1.2. Phân loại mã độc 5
1.1.3. Cách thức lây nhiễm mã độc 9
1.1.4. Dấu hiệu phát hiện mã độc 10
1.1.5. Các phương pháp phòng chống và phát hiện mã độc 12
1.1.5.1. Các biện pháp phòng ngừa mã độc 12
1.1.5.2. Các kỹ thuật phát hiện mã độc 14
1.2. Tổng quan về đường dẫn (path file) 16
1.2.1. Khái quát về đường dẫn (path file) 16
1.2.2. Phân loại đường dẫn (path file) 18
1.2.3. Khái quát về đường dẫn (path file) độc 19
1.2.4. Các phương pháp phòng chống và phát hiện path file độc 20
1.3. Kết luận chương 1 21
CHƯƠNG 2: PHÁT HIỆN MÃ ĐỘC SỬ DỤNG TRÍ TUỆ NHÂN TẠO 22
2.1. Tổng quan về học máy (Machine Learning) 22
2.1.1. Khái niệm Machine Learning 22

2.1.2. Phân loại các mô hình Machine Learning 23
2.1.2.1. Học có giám sát (Supervised Learning) 23
2.1.2.2. Học không giám sát (Unsupervised Learning) 25
2.1.2.3. Học bán giám sát (Semi-Supervised Learning) 26
2.1.2.4. Học tăng cường – Reinforcement Learning 27
2.2. Một số thuật toán trong phân loại dữ liệu 27
2.2.1. Cây quyết định (Decision tree) 27
2.2.1.1. Khái niệm 27
2.2.1.2. Thuật toán CLS 28
2.2.1.3. Thuật toán ID3 29
2.2.1.4. Thuật toán C4.5 30
2.2.1.5. Ưu và hạn chế của cây quyết định 32
2.2.2. Random Forest 33
2.2.2.1. Giới thiệu về Random Forest 33
2.2.2.2. Cách thức hoạt động của Random Forest 34
2.2.2.3. Ưu và nhược điểm của thuật toán Random Forest 35
2.2.3. Logistic Regression (Hồi quy Logistic) 36
2.2.3.1. Giới thiệu về Logistic Regression 36
2.2.3.2. Mô hình Logistic 36
2.2.4. Support Vector Machines (SVM) 38
2.2.4.1. Giới thiệu về thuật toán Support Vector Machines 38
2.2.4.2. Các ưu điểm và nhược điểm của SVM 39
2.3. Ứng dụng học máy và phát hiện mã độc 40
2.3.1. Các phương pháp trích xuất đặc trưng trong xử lý văn bản 40
2.3.1.1. Bag of Words (BoW) 40
2.3.1.2. TF-IDF 42
2.3.1.3. Word2Vec 43
2.3.1.4. GloVe 44
2.3.1.5. FastText 45

2.4. Một số thuật toán học sâu 45
2.4.1. Mạng Perceptron đa tầng (MLP – Multi-layer Perceptron) 45
2.4.2. Mạng Nơ-ron Tích chập (CNN - Convolutional Neural Networks) 47
2.4.3. Bộ nhớ Ngắn hạn Dài (LSTM - Long Short-Term Memory) 49
2.4.4. Bộ nhớ Ngắn hạn Dài Hai chiều (BiLSTM - Bidirectional LSTM) 51
2.4. Kết luận chương 52
CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ 53
3.1. Dữ liệu thực nghiệm 53
3.2. Kịch bản thực nghiệm 54
3.3. Tiêu chí đánh giá 57
3.4. Kết quả thực nghiệm 61
3.5. Kết luận chương 61
KẾT LUẬN 62
1. Những đóng góp của luận văn 62
2. Hướng phát triển của luận văn 62
DANH MỤC TÀI LIỆU THAM KHẢO 63

DANH SÁCH THUẬT NGỮ VIẾT TẮT

Từ viết tắt	Tiếng Anh	Tiếng Việt
LSTM	Convolutional Neural Networks	Bộ nhớ ngắn hạn dài
CNN	Convolutional Neural Networks	Mạng noron tích chập
BiLSTM	BiLSTM - Bidirectional LSTM	Bộ nhớ ngắn hạn dài hai chiều
MLP	Multi-layer Perceptron	Mạng Perceptron đa tầng
BoW	Bag of words	Túi từ
SVM	Support Vector Machine	Máy vector hỗ trợ
ML	Machine learning	Học máy
DL	Deep learning	Học sâu

DANH SÁCH HÌNH ẢNH

Hình 1.1. Các loại mã độc phổ biến 6
Hình 1.2. Hình ảnh đường dẫn tệp tin "File Path" 17
Hình 1.3. Hình ảnh đường dẫn (path file) độc 19
Hình 2.1. Kiến trúc cơ bản của Học có giám sát 23
Hình 2.2. Kiến trúc cơ bản của Học không giám sát 25
Hình 2.4. Mô hình về thuật toán Random Forest 34
Hình 2.5. Cách thức hoạt động của Random Forest 35
Hình 2.6. Thuật toán Support Vector Machine 39
Hình 2.7. Ví dụ về mô hình túi từ (Bag of Words) 41
Hình 3.2. Kiến trúc mô hình học sâu MLP 46
Hình 3. 1. Kiến trúc truyền thống của một mạng CNN 47
Hình 3. 2. Ví dụ phép tích chập 48
Hình 3. 3. Ví dụ phép gộp cực đại 49
Hình 3. 4. Mô hình cấu trúc một ô nhớ của LSTM 50
Hình 3. 5. Mô hình mô tả 2 thành phần LSTM trong BiLSTM 52

DANH SÁCH BẢNG BIỂU

Bảng 1. Bảng ví dụ mô hình Bag of Words 41
Bảng 2: Bảng số liệu dữ liệu 54
Bảng 3: Bảng đánh giá từng chỉ số với LSTM 58
Bảng 4: Bảng đánh giá từng chỉ số với MLP 58
Bảng 5: Bảng đánh giá từng chỉ số với BiLSTM 59

MỞ ĐẦU

1. Lý do chọn đề tài:

Trong bối cảnh an ninh mạng ngày càng trở nên phức tạp, mã độc vẫn là một trong những mối đe dọa lớn nhất đối với hệ thống thông tin. Các phương pháp phát hiện mã độc truyền thống như dựa trên chữ ký hoặc hành vi đã biết, mặc dù phổ biến, nhưng vẫn còn những hạn chế. Chúng thường không đủ hiệu quả khi đối phó với các mã độc mới, biến thể chưa được nhận diện hoặc các mã độc sử dụng kỹ thuật né tránh phát hiện. Điều này đặc biệt đáng lo ngại trong một thế giới kỹ thuật số ngày càng đa dạng, nơi các cuộc tấn công mạng ngày càng trở nên tinh vi và khó lường.

Để khắc phục những hạn chế của các phương pháp truyền thống, nghiên cứu này hướng tới việc phát triển một phương pháp mới dựa trên phân tích đường dẫn. Phân tích đường dẫn cung cấp một cái nhìn sâu hơn về cách mà mã độc thực thi trong hệ thống, giúp nhận diện những hành vi bất thường ngay cả khi chúng không để lại dấu vết cụ thể, từ đó tăng khả năng phát hiện các mã độc phức tạp.

Việc kết hợp phân tích đường dẫn với học sâu mở ra những cơ hội mới trong việc tự động hóa và nâng cao độ chính xác của quá trình phát hiện mã độc. Học sâu, với khả năng học và nhận diện các mẫu phức tạp từ dữ liệu lớn, có thể phân tích và phát hiện các đặc trưng bất thường trong đường dẫn mà các phương pháp truyền thống không thể nhận ra. Sự kết hợp này không chỉ cải thiện hiệu suất phát hiện mà còn giảm thiểu tỉ lệ báo động sai, giúp hệ thống bảo mật trở nên hiệu quả hơn.

Với những lý do trên, học viên chọn đề tài " **Phương pháp nghiên cứu phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu** " làm đề án tốt nghiệp cao học bởi không chỉ mang tính thời sự mà còn hứa hẹn đóng góp những giá trị lớn cho cả lý thuyết và ứng dụng trong thực tế.

2. Tổng quan về vấn đề nghiên cứu:

Phát hiện mã độc là một lĩnh vực quan trọng trong an ninh mạng với mục tiêu bảo vệ hệ thống thông tin trước các mối đe dọa và các cuộc tấn công nguy hiểm. Các phương pháp truyền thống như phân tích chữ ký và hành vi đã được sử dụng rộng rãi, nhưng chúng có những hạn chế khi đối phó với các mã độc mới hoặc tinh vi. Do đó,

nghiên cứu này hướng đến việc phát triển một phương pháp mới dựa trên phân tích đường dẫn, nhằm nâng cao hiệu quả trong việc phát hiện mã độc.

Trong thời đại số hóa, khi mạng lưới máy tính và internet ngày càng trở nên phổ biến, vấn đề an ninh mạng đã trở thành một mối quan tâm hàng đầu. Đặc biệt, sự gia tăng của các cuộc tấn công mã độc (malware) đang đặt ra một thách thức lớn đối với việc bảo mật thông tin và dữ liệu cá nhân. Mã độc có thể lây lan và gây thiệt hại trên quy mô lớn, không chỉ phá hủy dữ liệu mà còn gây ra các cuộc tấn công mạng phức tạp, ảnh hưởng đến cả cá nhân và tổ chức. Theo báo cáo của Symantec [1], số lượng mã độc đã tăng đột biến trong những năm gần đây, với sự phát triển nhanh chóng của các biến thể mã độc mới.

Một trong những phương pháp phát hiện mã độc truyền thống là sử dụng các kỹ thuật phát hiện dựa trên chữ ký (signature-based detection). Phương pháp này đã chứng minh được tính hiệu quả trong việc phát hiện các loại mã độc đã biết. Tuy nhiên, nó gặp phải hạn chế lớn trong việc đối phó với các biến thể mã độc mới, chưa được xác định trước đó. Các nghiên cứu của McAfee [2] và Kaspersky Lab [3] đã nhấn mạnh rằng các phương pháp dựa trên chữ ký không còn đủ hiệu quả trong việc phát hiện các cuộc tấn công mã độc hiện đại.

Để giải quyết vấn đề này, các phương pháp phát hiện mã độc dựa trên phân tích hành vi và phân tích đường dẫn đã được nghiên cứu và áp dụng. Phân tích hành vi tập trung vào việc quan sát và phát hiện các hành động bất thường của các tiến trình hoặc ứng dụng trong hệ thống. Trong khi đó, phân tích đường dẫn, đặc biệt là phân tích các đường dẫn mà mã độc có thể sử dụng để xâm nhập và lây lan trong hệ thống, đã trở thành một hướng nghiên cứu tiềm năng. Các nghiên cứu của Kolbitsch et al. [4] và Christodorescu & Jha [5] đã chứng minh rằng việc phân tích các đường dẫn này có thể cung cấp thông tin quan trọng giúp nhận diện và ngăn chặn mã độc.

Gần đây, với sự phát triển của trí tuệ nhân tạo (AI) và học sâu (Deep Learning), việc ứng dụng các mô hình học sâu vào phát hiện mã độc đã trở thành một xu hướng nghiên cứu nổi bật. Các mô hình này có khả năng học và phân tích các mẫu phức tạp trong dữ liệu, từ đó phát hiện các hành vi bất thường hoặc các đường dẫn không hợp lệ mà mã độc có thể sử dụng. Nghiên cứu của Saxe & Berlin [6] và Hardy et al. [7]

đã khẳng định rằng học sâu không chỉ cải thiện độ chính xác của việc phát hiện mã độc mà còn giúp phát hiện các biến thể mã độc mới một cách hiệu quả hơn so với các phương pháp truyền thống.

Phương pháp phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu, trong bối cảnh này là một hướng nghiên cứu tiềm năng và đầy thách thức. Mặc dù đã có không ít nghiên cứu quan trọng trong lĩnh vực này, nhưng vẫn còn nhiều vấn đề chưa được giải quyết hoàn toàn. Một trong những thách thức lớn là việc làm sao để mô hình học sâu có thể phân tích và hiểu được các đường dẫn phức tạp trong hệ thống, đồng thời nhận diện các hành vi bất thường của mã độc một cách chính xác.

Trong nghiên cứu của mình, tôi sẽ tập trung vào nghiên cứu phát hiện mã độc dựa trên đường dẫn sử dụng học sâu. Nghiên cứu này không chỉ đóng góp vào việc phát hiện và ngăn chặn mã độc mà còn mở ra các hướng đi mới trong việc ứng dụng AI vào an ninh mạng.

3. Mục đích nghiên cứu:

Mục tiêu chính của đề tài " Phương pháp nghiên cứu phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu " là nghiên cứu và ứng dụng thuật toán học sâu để phát hiện mã độc dựa trên phân tích các đặc trưng bất thường của đường dẫn. Nghiên cứu này không chỉ tìm hiểu về các phương pháp phân tích và nhận diện đường dẫn mà còn khám phá cách tích hợp các kỹ thuật học sâu để cải thiện độ chính xác và hiệu quả trong việc phát hiện các mối đe dọa an ninh mạng. Đặc biệt, đề tài sẽ xem xét các yếu tố như tính khả dụng của mô hình, khả năng ứng dụng trong thực tế và cách các mô hình này có thể phát hiện được các biến thể mã độc mới mà chưa được nhận diện trước đó.

4. Đối tượng và phạm vi nghiên cứu:

a) Đối tượng nghiên cứu

- Nghiên cứu hành vi của mã độc: Đề tài tập trung vào việc phân tích cách thức mã độc tương tác với hệ thống, đặc biệt là thông qua các đường dẫn và quy trình của hệ điều hành. Điều này bao gồm cả các kỹ thuật mà mã độc sử dụng để lẩn tránh sự phát hiện, như việc ẩn mã, thay đổi đường dẫn hoặc sử dụng các kỹ thuật tàng hình khác.

- Nghiên cứu đặc trưng bất thường của đường dẫn. Đề tài tập trung vào việc phân tích các đường dẫn trong hệ điều hành Windows. Đây là hệ điều hành được sử dụng rộng rãi nhất và cũng là mục tiêu chủ yếu của nhiều loại mã độc hiện nay.

b) Phạm vi nghiên cứu

- Nghiên cứu sẽ tập trung vào các thuật toán học máy, học sâu tiên tiến như CNN, RNN hoặc LSTM trong việc phân tích và nhận diện các đường dẫn bất thường.
- Nghiên cứu dịch vụ VirusTotal để phát hiện xem tệp tin hay URL có chứa mã độc hay không .

5. Phương pháp nghiên cứu:

Nghiên cứu lý thuyết:

- Nghiên cứu tài liêu: Đọc và phân tích các tài liệu, bài báo khoa học liên quan đến mã độc và đường dẫn.
- Mô tả nguồn dữ liệu mã độc và dữ liệu sạch
- Phương pháp thu thập và xử lý dữ liệu đường dẫn.
- Tổng hợp lý thuyết: Áp dụng các mô hình học sâu để phát hiện mã độc dựa trên đường dẫn.

Xây dựng mô hình

- Giới thiệu về các mô hình học sâu sẽ được sử dụng (CNN, RNN, LSTM, hoặc các biến thể).
- Quá trình huấn luyện mô hình và tối ưu hóa các tham số:

Phân tích và đánh giá

- Phương pháp đánh giá hiệu suất của mô hình (sử dụng các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu, F1-score).
- So sánh hiệu quả giữa các mô hình khác nhau.

CHƯƠNG 1: TỔNG QUAN VỀ MÃ ĐỘC VÀ ĐƯỜNG DẪN (PATH FILE)

1.1. Tổng quan về mã độc

1.1.1. Khái niệm về mã độc

Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (NIST – National Institute of Standards and Technology) đưa ra khái niệm mã độc (Malware) như sau: đó là một chương trình máy tính được đưa vào hệ thống hoặc thiết bị điện tử một cách âm thầm, không có sự hay biết của người sử dụng, với ý đồ xâm hại đến tính bí mật, tính toàn vẹn hoặc khả năng vận hành bình thường của dữ liệu, các ứng dụng cũng như hệ điều hành.

Về bản chất, virus máy tính là những đoạn mã hoặc chương trình được viết ra nhằm khả năng tự nhân bản chính mình rồi phát tán sang các đối tượng khác như tệp tin, ổ đĩa hay toàn bộ hệ thống máy tính, qua đó xâm nhập vào những thành phần khác trong môi trường tính toán để tiến hành tấn công hoặc gây ra thiệt hại.

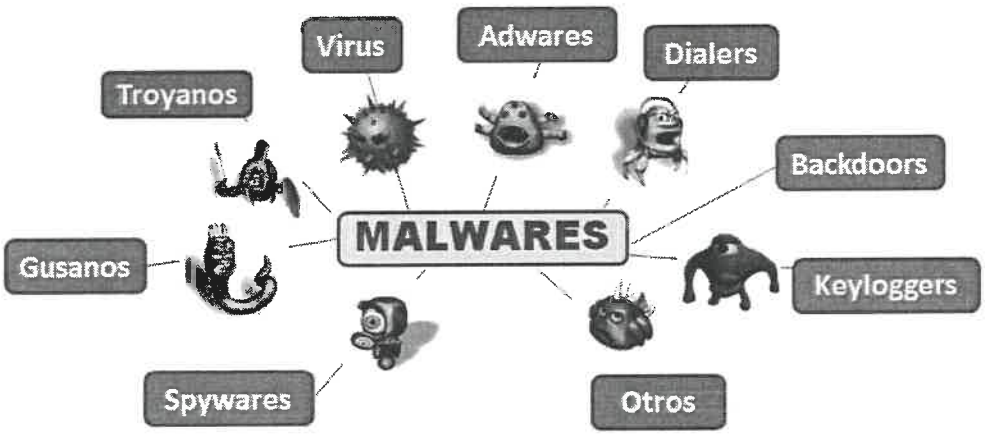
Từ cách hiểu nêu trên có thể thấy virus chỉ là một nhánh nhỏ trong bức tranh chung của mã độc. Phạm trù mã độc còn bao gồm hàng loạt hình thái khác như worm, trojan, spyware, ransomware, cùng những công cụ hỗ trợ tấn công mà tin tặc thường khai thác, ví dụ backdoor, rootkit hay keylogger. Dù cơ chế hoạt động và mục đích của từng loại là khác nhau, điểm chung của chúng vẫn là gây ra tổn hại cho hệ thống hoặc dữ liệu của người sử dụng.

Nhìn chung, những kẻ tạo ra mã độc đều nhắm đến việc thực hiện các hành vi phi pháp hoặc mang tính phá hoại, chẳng hạn xâm nhập trái phép vào hệ thống, đánh cắp dữ liệu cá nhân, phát tán thư rác, cho đến các hành vi tống tiền hay tấn công gây tổn hại trực tiếp lên hệ thống máy tính. Động cơ đằng sau việc phát triển mã độc thường xoay quanh lợi ích của cá nhân hoặc một nhóm người, có thể mang màu sắc kinh tế, chính trị, hoặc đôi khi chỉ đơn thuần là ý đồ chơi khăm, gây phiền toái cho người khác.

1.1.2. Phân loại mã độc

Cách nhìn nhận của NIST (National Institute of Standards and Technology), vốn cũng được đa số giới chuyên môn Công nghệ thông tin đồng thuận, cho rằng Virus,

Worm, Trojan horse, Adware, Spyware, Backdoor, Botnet, Launcher hay Rootkit thực chất chỉ là những biểu hiện cụ thể nằm trong tập hợp mã độc hại nói chung. Nói cách khác, thuật ngữ "Virus máy tính" mà nhiều người vẫn quen dùng không nên bị bó hẹp trong đúng nghĩa của Virus, mà cần được hiểu rộng ra bao trùm nhiều dạng thức mã độc khác nhau, từ những phần mềm tấn công như Worm, Trojan cho tới các phần mềm do thám như Spyware và Adware.



Hình 1.1. Các loại mã độc phổ biến

Phân loại các dạng mã độc phổ biến:

- Virus máy tính (Virus): là dạng chương trình hoặc đoạn mã sở hữu khả năng tự sao chép rồi lan truyền sang các đối tượng khác như tệp tin, ổ đĩa hay các máy tính khác. Tác hại của chúng thể hiện qua việc làm gián đoạn hoạt động của các chương trình, xóa bỏ dữ liệu, sửa đổi thông tin hoặc gây hỏng hóc phần cứng. Không dừng lại ở mục đích phá hoại thuần túy, các biến thể virus hiện đại còn nhắm tới việc đánh cắp những thông tin cá nhân nhạy cảm như mã số thẻ tín dụng, tài khoản ngân hàng, tài liệu mật, hoặc mở đường cho tin tặc chiếm quyền điều khiển hệ thống nhằm phục vụ lợi ích của kẻ phát tán.

- Sâu máy tính (Worm): là loại mã độc có thể tự nhân bản và tự động phát tán thông qua mạng lưới hoặc các điểm yếu bảo mật trong hệ thống mạng (phổ biến nhất là qua thư điện tử và các lỗ hổng của hệ điều hành). Đặc trưng của worm là không đòi hỏi tệp thực thi để hoạt động, nhờ đó tốc độ lây lan của chúng rất nhanh. Bên cạnh việc gây hại trực tiếp cho các máy bị nhiễm, đối tượng chính mà worm nhắm tới là các mạng thông tin, khiến hiệu năng hoạt động suy giảm, thậm chí có thể bị lợi dụng

để đánh cắp dữ liệu nhạy cảm trong phạm vi mạng đó. Một ví dụ điển hình là con sâu do Robert Morris tạo ra năm 1988, từng có sức công phá đủ để làm tê liệt hầu như mọi hệ điều hành UNIX có kết nối Internet vào thời điểm đó.

- Trojan horse: là dạng phần mềm độc hại được xây dựng nhằm len lỏi vào hệ thống của nạn nhân bằng cách đội lốt hoặc ẩn mình bên trong những phần mềm, tệp tin trông có vẻ đáng tin cậy. Cái tên "Trojan horse" bắt nguồn từ truyền thuyết Hy Lạp cổ đại về cuộc chiến thành Troia, khi quân Hy Lạp dùng một con ngựa gỗ khổng lồ để giấu binh lính bên trong và lừa đưa vào bên trong thành trì đối phương. Tương tự như vậy, Trojan horse đánh lừa người dùng để thâm nhập hệ thống, từ đó thực hiện các hành vi gây hại hoặc phục vụ cho mục tiêu tấn công mà kẻ đứng sau nhắm đến.

- Phần mềm gián điệp (Spyware): là một dạng mã độc được thiết kế với mục tiêu thu thập dữ liệu cá nhân của nạn nhân mà không hề xin phép hay được sự đồng ý của họ. Loại phần mềm này thường âm thầm cài vào máy tính hoặc thiết bị di động ngoài sự hay biết của người dùng, sau đó thực hiện các hành vi như ghi nhận lịch sử hoạt động trực tuyến, thu thập những thông tin cá nhân nhạy cảm, theo dõi thông tin đăng nhập cũng như tài khoản ngân hàng, thậm chí âm thầm giành quyền kiểm soát các hoạt động của máy tính.

- Scareware: là dạng mã độc thường đội lốt các thông báo hoặc cảnh báo giả mạo về những sự cố bảo mật không có thực trên máy tính, qua đó thúc ép người dùng thực hiện một số hành động như mua phần mềm bảo mật giả, cung cấp thông tin cá nhân, hoặc thanh toán chi phí để "khắc phục" mối đe dọa mà chúng bịa ra. Kiểu tấn công này thường khiến người dùng hoang mang, lo sợ, và trong nhiều trường hợp còn có thể gây hại thực sự cho máy tính hoặc bị lợi dụng để chiếm đoạt tài khoản thanh toán của họ.

- Phần mềm quảng cáo (Adware): là loại mã độc chuyên dùng để chèn các quảng cáo không mong muốn lên máy tính người dùng. Chúng hay đi kèm theo các phần mềm miễn phí hoặc bản dùng thử, và quảng cáo thường chỉ dừng hiển thị khi người dùng trả phí theo đúng chính sách của nhà cung cấp. Có một biến thể khá phổ biến của adware hoạt động theo cơ chế gián điệp, âm thầm cài vào trong lúc người dùng duyệt web để thu thập dữ liệu hành vi rồi dùng chính dữ liệu đó nhằm hiển thị quảng

cáo nhắm mục tiêu. Đáng chú ý, nhiều phần mềm quảng cáo hiện nay còn được tích hợp thêm các dạng mã độc khác như virus, worm hay Trojan horse, gây ra thiệt hại đáng kể cho thiết bị và hệ thống mạng.

- Downloader: là một dạng phần mềm độc hại có chức năng tải về và cài đặt thêm các phần mềm khác mà bản thân người dùng không hề hay biết hoặc không mong muốn. Downloader độc hại thường đóng vai trò là công cụ trung gian để phát tán những loại mã độc khác như virus, trojan, ransomware hay adware. Người dùng có thể vô tình cài đặt loại phần mềm này khi tải và cài phần mềm từ những nguồn không rõ ràng, hoặc bị dẫn dụ thông qua các thủ đoạn lừa đảo.

- Backdoor: là đoạn mã độc được cấy vào máy tính của nạn nhân, mở ra một "cửa hậu" cho phép tin tặc kết nối và điều khiển từ xa mà không phải trải qua bước xác thực nào. Nhờ có backdoor, kẻ tấn công có thể âm thầm thực thi các câu lệnh ngay trên máy nạn nhân mà không bị hệ thống phát hiện.

- Botnet: là tập hợp các máy tính đã bị lây nhiễm mã độc và chịu sự điều khiển từ xa của tin tặc. Cũng giống backdoor, botnet cho phép kẻ tấn công truy cập cũng như thao túng các hệ thống nạn nhân. Những máy tính nằm trong cùng một botnet sẽ nhận lệnh từ máy chủ điều khiển của tin tặc thông qua các kênh liên lạc như Internet Relay Chat (IRC) hoặc mạng ngang hàng peer-to-peer (P2P).

Một khi đã sở hữu một mạng botnet trong tay, tin tặc hay các tổ chức đứng sau hoàn toàn có thể biến nó thành vũ khí phục vụ cho các cuộc chiến tranh mạng, mà biểu hiện thường thấy nhất là những đợt tấn công từ chối dịch vụ nhằm làm tê liệt hạ tầng mạng của một tổ chức, thậm chí của cả một quốc gia.

- Launcher: là loại mã độc đóng vai trò khởi chạy hoặc triển khai các loại mã độc khác trên máy tính nạn nhân. Chúng có thể được che giấu bên trong những tệp tin gọi là "Launchers", với mục đích lặng lẽ xâm nhập hệ thống rồi tiến hành các hành vi gây hại như phá hỏng dữ liệu, đánh cắp thông tin cá nhân, hay chiếm quyền điều khiển từ xa máy tính nạn nhân mà không được phép.

- Rootkit: là mã độc chuyên dùng để che giấu sự hiện diện của chính nó trên hệ thống, đồng thời tạo quyền truy cập trái phép cho kẻ tấn công. Nó có khả năng giấu các tiến trình, tệp tin và dữ liệu nằm trong Registry của Windows, khiến phần lớn các

phần mềm diệt virus hay chống malware không thể phát hiện ra. Ngoài ra, rootkit còn có thể ghi lại các kết nối mạng cũng như giám sát thao tác bàn phím. Tùy theo mục đích sử dụng mà rootkit có thể bị lợi dụng cho những việc xấu, chẳng hạn nghe lén thông tin hoặc che giấu sự tồn tại của các phần mềm độc hại khác.

- Keylogger: là mã độc được xây dựng nhằm mục đích ghi lại từng phím bấm trên bàn phím máy tính mà không cần đến sự cho phép của người dùng. Nó lưu lại các ký tự, chữ số và câu lệnh được gõ vào, sau đó gửi những dữ liệu này về cho kẻ điều khiển thông qua email hoặc các giao thức mạng khác.

Mục đích phổ biến nhất khi triển khai keylogger là thu thập trái phép các thông tin của người dùng, bao gồm tên đăng nhập, mật khẩu, dữ liệu tài khoản ngân hàng và nhiều loại thông tin nhạy cảm khác. Tuy vậy, cũng tồn tại những trường hợp sử dụng keylogger một cách hợp pháp, chẳng hạn giám sát hoạt động máy tính khi có sự đồng thuận của người dùng, hoặc phục vụ công tác phân tích hành vi trong lĩnh vực an ninh mạng nhằm chủ động phòng ngừa các nguy cơ tấn công.

1.1.3. Cách thức lây nhiễm mã độc

Mã độc có thể phát tán sang các máy tính hay thiết bị khác thông qua nhiều con đường khác nhau, cụ thể như sau.

- Qua tệp tài liệu và phần mềm: Mã độc thường được giấu bên trong các tệp tài liệu hay phần mềm tải từ Internet, kể cả những phần mềm hợp pháp nhưng đã bị can thiệp để chèn thêm mã độc. Nhóm phần mềm bẻ khóa (Crack) là đối tượng có nguy cơ chứa mã độc cao nhất. Vì muốn dùng phần mềm miễn phí, không ít người dùng vẫn chấp nhận đánh đổi rủi ro nhiễm mã độc khi tải về và cài đặt các phần mềm dạng này.

- Qua trang web: Việc truy cập các website, nhất là những trang thiếu độ tin cậy, cũng có thể khiến hệ thống bị lây nhiễm mã độc. Ngay cả khi chỉ đơn thuần xem nội dung trang web, mã độc vẫn có cơ hội xâm nhập thông qua các lỗ hổng bảo mật tồn tại trong những plugin trình duyệt do bên thứ ba phát triển.

- Qua email: Do mức độ sử dụng email ngày càng phổ biến, đây trở thành một kênh phát tán mã độc rất hiệu quả. Sau khi đã xâm nhập được vào máy nạn nhân, mã

độc có khả năng tự động dò tìm danh bạ email rồi gửi bản sao của chính mình đến hàng loạt địa chỉ trong danh sách đó.

- Qua dịch vụ chat: Các nền tảng chat cho phép nhiều người trò chuyện trực tuyến với nhau qua mạng máy tính, và cũng giống như email, đây có thể trở thành kênh lây lan mã độc khi kẻ tấn công gửi tệp tin hoặc đường liên kết chứa mã độc tới người dùng.

- Qua Remote Desktop Protocol (RDP): RDP vốn là công nghệ hỗ trợ kết nối từ xa tới một máy tính khác thông qua mạng. Tuy được Microsoft phát triển ban đầu, giao thức này hiện đã được nhiều hệ điều hành khác hỗ trợ. Đây cũng chính là mục tiêu mà tội phạm mạng tìm cách khai thác: nếu RDP không được cấu hình bảo mật chặt chẽ, hoặc vẫn vận hành trên các hệ điều hành đã cũ, nó sẽ trở thành lỗ hổng để kẻ tấn công lợi dụng. Một khi phát hiện ra kết nối yếu và chiếm được quyền truy cập từ xa, chúng có thể âm thầm cài mã độc hoặc đánh cắp dữ liệu mà nạn nhân hoàn toàn không hay biết.

- Qua thiết bị USB: Trong trường hợp tin tặc tiếp cận trực tiếp được với thiết bị của nạn nhân, USB trở thành một công cụ lây nhiễm nhanh gọn và đơn giản. Những chiếc USB này thường được cài sẵn mã độc để thu thập dữ liệu ngay khi kết nối. Chẳng hạn, một ổ flash có thể cấy keylogger vào máy tính nhằm ghi lại toàn bộ thao tác nhập liệu, kể cả thông tin đăng nhập, dữ liệu thanh toán và các thông tin nhạy cảm khác. Mã độc lây qua đường này có thể là ransomware, spyware, virus hay worm, và hoàn toàn có khả năng lan rộng ra nhiều hệ thống khác nhau.

1.1.4. Dấu hiệu phát hiện mã độc

- Tốc độ máy tính bị suy giảm: Một trong những biểu hiện thường gặp khi nhiễm mã độc là hệ điều hành hoạt động chậm hẳn đi, bất kể đang lướt web hay chỉ chạy các ứng dụng cài sẵn trên máy, mức tiêu thụ tài nguyên hệ thống vẫn cao bất thường. Người dùng thậm chí có thể nhận ra quạt tản nhiệt quay hết công suất - dấu hiệu cho thấy có tiến trình nào đó đang ngầm chiếm dụng tài nguyên ở chế độ nền. Hiện tượng này hay xảy ra khi máy tính đã bị đưa vào một mạng botnet, tức mạng lưới các máy bị kiểm soát dùng để thực hiện tấn công DDoS, phát tán thư rác hoặc đào tiền điện tử.

- Xuất hiện dày đặc các quảng cáo gây khó chịu: Việc các cửa sổ quảng cáo bật lên một cách ngoài ý muốn là dấu hiệu điển hình cho thấy máy đã nhiễm mã độc, đặc biệt gắn liền với nhóm phần mềm quảng cáo (adware). Đáng lưu ý, những cửa sổ pop-up dạng này thường kéo theo các nguy cơ mã độc tiềm ẩn khác đi kèm. Do đó, nếu bắt gặp thông báo kiểu "CHÚC MỪNG, Bạn đã giành được một buổi đọc sách tâm linh miễn phí!" hiện lên trên màn hình, tuyệt đối không nên nhấp vào, bởi bất kể phần thưởng nào được hứa hẹn, cái giá phải trả sau đó thường không hề rẻ.

- Hệ thống thường xuyên gặp sự cố: Biểu hiện có thể là tình trạng treo máy hoặc hiện tượng BSOD (màn hình xanh chết chóc) - lỗi xuất hiện trên Windows sau khi hệ thống gặp phải một sự cố nghiêm trọng.

- Dung lượng ổ đĩa hao hụt một cách khó hiểu: Nguyên nhân có thể đến từ việc mã độc mang theo và ẩn giấu nhiều thành phần độc hại khác bên trong ổ cứng, dạng thức này thường được gọi là phần mềm đóng gói.

- Lưu lượng truy cập Internet tăng vọt bất thường: Ví dụ với Trojan, ngay sau khi xâm nhập vào máy mục tiêu, nó sẽ tìm cách kết nối tới máy chủ điều khiển của kẻ tấn công để tải về phần mềm lây nhiễm giai đoạn hai, thường là ransomware - đây chính là nguyên nhân khiến lưu lượng Internet tăng đột biến. Cơ chế tương tự cũng diễn ra với botnet, spyware và các mối đe dọa khác vốn cần duy trì liên lạc hai chiều với máy chủ điều khiển.

- Cấu hình trình duyệt bị thay đổi ngoài ý muốn: Nếu trang chủ mặc định bỗng nhiên bị đổi, hoặc xuất hiện thêm những thanh công cụ, tiện ích mở rộng, plugin lạ mà bản thân không chủ động cài đặt, rất có thể máy đã dính mã độc. Nguyên nhân có thể đa dạng, nhưng phần lớn xuất phát từ việc đã lỡ nhấp vào một cửa sổ "chúc mừng trúng thưởng" nào đó và vô tình tải về phần mềm không mong muốn.

- Phần mềm diệt virus tự dưng ngừng hoạt động và không thể khởi động lại được, khiến hệ thống mất đi lớp bảo vệ trước chính loại mã độc đã âm thầm vô hiệu hóa nó.

- Không thể truy cập được vào tệp tin hoặc toàn bộ máy tính: Đây là dấu hiệu đặc trưng của việc nhiễm ransomware. Kẻ tấn công thường công khai sự hiện diện của mình bằng cách để lại lời nhắn đòi tiền chuộc ngay trên màn hình, hoặc thay hẳn hình nền máy tính thành thông báo tống tiền. Nội dung thông báo thường cho biết dữ liệu

đã bị mã hóa và yêu cầu nạn nhân trả một khoản tiền chuộc để đổi lấy việc được giải mã lại các tệp tin.

Tuy nhiên, cần lưu ý rằng dù hệ thống có vẻ vận hành hoàn toàn bình thường thì cũng chưa thể kết luận chắc chắn là máy tính không hề dính mã độc. Có những loại mã độc tinh vi có khả năng ẩn náu sâu bên trong hệ thống, né tránh mọi sự phát hiện và âm thầm thực hiện các hoạt động mờ ám mà không để lộ bất kỳ dấu hiệu cảnh báo nào.

1.1.5. Các phương pháp phòng chống và phát hiện mã độc

1.1.5.1. Các biện pháp phòng ngừa mã độc

- Trang bị phần mềm diệt virus có bản quyền: Nên ưu tiên lựa chọn những phần mềm diệt virus uy tín trên thị trường như Kaspersky, Bitdefender, Avast, Norton, Bkav, CyStack..., đồng thời luôn tải về từ trang chủ chính thức của nhà sản xuất để đảm bảo an toàn.

- Sử dụng công cụ chuyên phát hiện spyware: Nhóm công cụ này được xây dựng riêng để nhận diện và xử lý phần mềm gián điệp tồn tại trong hệ thống, thông qua cách ly hoặc gỡ bỏ hoàn toàn. Điểm khác biệt so với phần mềm diệt virus thông thường là chúng có thể xử lý cả những loại mã độc không thuộc nhóm spyware. Nhìn chung, các công cụ phát hiện spyware thường có năng lực xử lý mạnh hơn, góp phần ngăn chặn các vấn đề nghiêm trọng như hệ thống bị chậm, ứng dụng hoạt động thiếu ổn định, đồng thời bảo vệ quyền riêng tư cho người dùng.

- Triển khai Hệ thống phòng chống xâm nhập mạng (IPS): IPS thực hiện việc kiểm tra từng gói tin và phân tích lưu lượng mạng nhằm phát hiện, ngăn chặn các hành vi khả nghi. Cách thức triển khai của nó khá tương đồng với tường lửa, và nó có thể chặn đứng hiệu quả những mối đe dọa đã được biết đến như worm tấn công dịch vụ mạng, worm lây qua email hay các virus có dấu hiệu nhận biết rõ ràng. Dù vậy, IPS lại không đủ khả năng ngăn chặn mã độc trên thiết bị di động hay Trojan, cho dù nó vẫn có thể phát hiện và chặn một số mối đe dọa chưa xác định thông qua việc phân tích các giao thức mạng.

- Sử dụng tường lửa và bộ định tuyến: Cả thiết bị mạng như tường lửa, bộ định tuyến lẫn phần mềm tường lửa cài trên máy chủ đều có chức năng kiểm soát lưu lượng mạng, cho phép hoặc chặn dựa trên các quy tắc đã cấu hình sẵn. Riêng bộ định tuyến sử dụng danh sách kiểm soát truy cập (Access-list) để lọc gói tin ra vào mạng. Tường lửa được chia thành hai loại: tường lửa mạng và tường lửa máy chủ. Trong đó, tường lửa mạng đặt giữa các hệ thống mạng nhằm hạn chế lưu lượng bất hợp lệ di chuyển từ mạng này sang mạng khác, còn tường lửa máy chủ đảm nhiệm việc lọc dữ liệu ra vào của chính máy chủ đó. Cả hai loại này đều đóng vai trò quan trọng trong việc phòng ngừa sự cố do mã độc gây ra.

- Duyệt web một cách an toàn: Nếu thiếu ý thức cảnh giác, chính thao tác truy cập web hàng ngày có thể trở thành con đường khiến máy tính dễ dàng nhiễm mã độc. Khi lướt web, cần tránh ghé vào các trang web đen, trang chứa mã độc hoặc nội dung thiếu lành mạnh; đồng thời không nên tùy tiện bấm vào các đường liên kết xuất hiện trong email, tin nhắn chat hay từ những trang web khác. Bởi lẽ, những website và URL dạng này thường tiềm ẩn mã độc, chỉ chờ người dùng nhấp vào là sẽ tự động tải về, cài đặt và thực thi ngay trên máy. Trên thực tế, khi rà soát máy tính của nhân viên văn phòng, phần lớn đều phát hiện các addon hoặc phần mềm quảng cáo đã được cài vào, nguyên nhân chính là do thói quen duyệt web đã vô tình cho phép các addon hay phần mềm này được thực thi.

- Đặt ra nguyên tắc xử lý tệp tin: Hạn chế tối đa việc mở hoặc tải về những tệp tin không rõ nguồn gốc, nhất là các tệp thực thi (chẳng hạn .exe, .dll,...). Với các tệp dạng này, cách làm an toàn nhất là quét qua phần mềm diệt virus hoặc kiểm tra trên trang VirusTotal trước khi mở. Một khi còn nghi ngờ về độ an toàn, tốt nhất nên dừng ngay việc thực thi tệp.

- Thiết lập chính sách quản lý thiết bị PnP: Những thiết bị như USB, CD/DVD hoàn toàn có thể bị virus lợi dụng để tự động thực thi mà không cần người dùng cho phép. Do đó, cần cấu hình lại chế độ hoạt động cho các thiết bị và chương trình liên quan để hạn chế tình trạng mã độc thực thi ngoài tầm kiểm soát. Cụ thể, khi cắm USB vào máy, không nên mở trực tiếp bằng cách bấm Enter hay nhấp đúp vào biểu tượng ổ đĩa, mà nên nhấp chuột phải rồi chọn "Explorer" để đảm bảo an toàn hơn.

- Cập nhật thường xuyên máy tính và phần mềm: Cần duy trì thói quen cài đặt các bản vá mới nhất cho hệ điều hành, các ứng dụng đang sử dụng, và đặc biệt là chương trình diệt virus. Đây là biện pháp then chốt để ngăn mã độc lợi dụng các lỗ hổng bảo mật nhằm lây lan, đồng thời giúp phần mềm diệt virus nâng cao khả năng nhận diện và xử lý những mẫu mã độc mới xuất hiện.

- Đào tạo, nâng cao nhận thức cho người dùng: Các tổ chức nên thường xuyên mở các lớp tập huấn để trang bị kiến thức về mã độc cho đội ngũ cán bộ, nhân viên. Thông qua đó, người tham gia sẽ nắm được các chính sách phòng chống mã độc cũng như quy trình xử lý phù hợp ứng với từng tình huống cụ thể. Nội dung đào tạo cần đề cập đến các biện pháp phòng tránh mã độc lẫn cách hạn chế thiệt hại khi sự cố đã xảy ra. Đồng thời, cán bộ nhân viên cần được hướng dẫn để nhận biết các nguy cơ tiềm ẩn, hiểu được cơ chế lây nhiễm và phát tán của mã độc trong hệ thống, cũng như thực hành thuần thục các biện pháp phòng ngừa.

- Xây dựng chính sách phòng chống mã độc ở cấp doanh nghiệp: Mỗi tổ chức cần ban hành một chính sách rõ ràng nhằm ngăn ngừa các sự cố liên quan đến mã độc, đảm bảo tính khả thi khi triển khai, đồng thời duy trì được sự nhất quán và hiệu quả trong công tác phòng ngừa. Chính sách nên được xây dựng theo hướng tổng quát nhất có thể, vừa tạo sự linh hoạt khi áp dụng vào thực tế, vừa hạn chế việc phải chỉnh sửa, cập nhật thường xuyên. Trên thực tế, không ít cơ quan, tổ chức đã có chính sách riêng để xử lý vấn đề mã độc, song cũng có những nơi để chính sách phòng chống mã độc chồng chéo với các chính sách an ninh khác.

1.1.5.2. Các kỹ thuật phát hiện mã độc

- Kỹ thuật phát hiện dựa trên chữ ký (signature-based)

Cách tiếp cận dựa trên chữ ký thực hiện việc nhận dạng mã độc bằng cách đối chiếu các đặc điểm riêng biệt của mẫu cần kiểm tra với những mã độc đã được biết đến từ trước. "Chữ ký" ở đây thường là chuỗi băm (hash) của tệp mã độc, ví dụ MD5 hay SHA1, hoặc có thể là một chuỗi ký tự cố định nằm trong nội dung tệp hay trong phần siêu dữ liệu (metadata) của nó. Quy trình thông thường là khi các hãng bảo mật xác nhận một tệp là độc hại, chữ ký tương ứng của tệp đó sẽ được bổ sung vào cơ sở dữ liệu. Về sau, mỗi khi phần mềm diệt virus quét một tệp tin trên hệ thống, nếu phát

hiện chữ ký trùng khớp đã tồn tại trong cơ sở dữ liệu, nó sẽ đưa ra cảnh báo tệp đó có khả năng là mã độc. Nhược điểm là phương pháp này chỉ phát huy hiệu quả với những loại mã độc phổ biến, đã có chữ ký được ghi nhận sẵn.

Điểm yếu lớn nhất của phương pháp này bộc lộ khi gặp phải các mã độc đã bị chỉnh sửa, biến thể, hoặc những mẫu hoàn toàn mới chưa từng được ghi nhận chữ ký. Bằng cách thay đổi một vài đặc điểm hoặc sinh ra các biến thể mới không khớp với bất kỳ chữ ký nào đang lưu trữ, mã độc hoàn toàn có thể qua mặt được cơ chế phát hiện dựa trên chữ ký.

- Phát hiện dựa trên phân tích hành vi

Khác với cách tiếp cận trên, phương pháp phát hiện dựa trên hành vi (còn được gọi là phương pháp heuristics) nhận diện các mẫu độc hại thông qua việc quan sát và phân tích cách thức mà phần mềm hay tệp tin vận hành trong một hệ thống hoặc mạng.

Thay vì chỉ căn cứ vào những đặc điểm cố định như chữ ký, phương pháp này chú trọng đánh giá cách một đối tượng hành xử trong hệ thống. Một đối tượng bị xem là khả nghi hoặc độc hại khi nó thực hiện những hành động bất thường hay vượt ra ngoài phạm vi quyền hạn được cấp. Một số hành vi có thể tiềm ẩn nguy hiểm là: vô hiệu hóa các cơ chế bảo mật, cài đặt rootkit, tự động khởi chạy cùng hệ thống, chỉnh sửa tệp host, hay thiết lập những kết nối mạng đáng ngờ. Từng hành vi riêng lẻ có thể chưa gây hại tức thì, nhưng khi chúng xuất hiện đồng thời, mức độ nghi vấn đối với đối tượng đó sẽ tăng lên đáng kể. Hệ thống thường đặt ra một ngưỡng nhất định, và một khi tệp tin nào vượt ngưỡng này sẽ bị gắn nhãn là mã độc. Nhờ vậy, phương pháp này có thể phát hiện được cả mã độc đa hình (thay đổi hình dạng) lẫn các mối đe dọa zero-day chưa từng xuất hiện trước đó.

Mức độ chính xác của kỹ thuật heuristics phụ thuộc nhiều vào quá trình tệp tin được thực thi. Giải pháp tối ưu là cho tệp chạy trong một môi trường ảo hóa, chẳng hạn sandbox, để quan sát hành vi của nó. Cách làm này tuy tốn nhiều thời gian hơn nhưng bù lại an toàn hơn hẳn, bởi tệp tin sẽ được kiểm tra kỹ trước khi được phép thực thi trên hệ thống thật.

Lợi thế nổi bật của phương pháp heuristics nằm ở khả năng phát hiện không chỉ mã độc đã biết mà cả những cuộc tấn công zero-day hay virus đa hình. Tuy nhiên,

một số mẫu mã độc tinh vi có thể nhận biết được mình đang chạy trong môi trường ảo, từ đó "án binh bất động", không thực hiện hành vi độc hại nào trong sandbox. Bên cạnh đó, trước tốc độ gia tăng chóng mặt của các biến thể mã độc hiện nay, phương pháp này cũng bộc lộ những hạn chế nhất định khi phải đối mặt với các mối đe dọa hoàn toàn mới.

- Phát hiện ứng dụng kỹ thuật học máy

Đây là hướng tiếp cận trong lĩnh vực an ninh mạng, sử dụng các thuật toán và mô hình học máy để nhận diện, xác định các loại mã độc khác nhau.

Không giống việc dựa vào chữ ký cố định hay các đặc điểm biểu hiện cụ thể như những phương pháp trước, cách làm này tận dụng khả năng tự học từ dữ liệu của học máy để nhận ra các mẫu bất thường hoặc mang tính độc hại. Nhờ đó, thuật toán học máy có thể tự động phát hiện những hành vi khác lạ, tiềm ẩn nguy cơ tấn công hay sự hiện diện của mã độc.

Các mô hình được sử dụng có thể xây dựng trên nhiều nền tảng thuật toán khác nhau như máy vector hỗ trợ (SVM), mạng nơ-ron, cây quyết định, hay các kỹ thuật học sâu như mạng nơ-ron sâu (deep learning). Sau khi được huấn luyện từ tập dữ liệu mẫu, mô hình sẽ được dùng để phân tích các mẫu mới xuất hiện và đưa ra kết luận về việc chúng có phải là mã độc hay không.

Thế mạnh của cách tiếp cận này là có thể nhận ra cả những mẫu độc hại mới lẫn các biến thể khác nhau của mã độc mà không cần phụ thuộc vào chữ ký hay đặc điểm cố định nào. Hơn nữa, năng lực nhận dạng của mô hình học máy còn có thể được cải thiện liên tục khi được huấn luyện thêm với dữ liệu mới, từ đó nâng cao hiệu quả phát hiện các mối đe dọa tiềm ẩn.

1.2. Tổng quan về đường dẫn

1.2.1. Khái quát về đường dẫn

Có thể hiểu path file (đường dẫn tệp tin) là một dãy ký tự dùng để biểu diễn nơi lưu trữ - dù là tuyệt đối hay tương đối - của một tệp hoặc một thư mục bên trong hệ thống lưu trữ của máy tính. Trong cấu trúc hệ thống tập tin, mỗi tầng của cây phân

cấp tương ứng với một thư mục, nhờ đó người dùng có thể dễ dàng thao tác, tìm kiếm và quản lý dữ liệu được tổ chức trong hệ thống của mình.

Nói cách khác, đường dẫn tệp tin cho biết chính xác nơi một tệp đang tồn tại trong cây thư mục của máy tính, và nó có thể được dùng để trỏ tới nhiều dạng tệp khác nhau, chẳng hạn như:

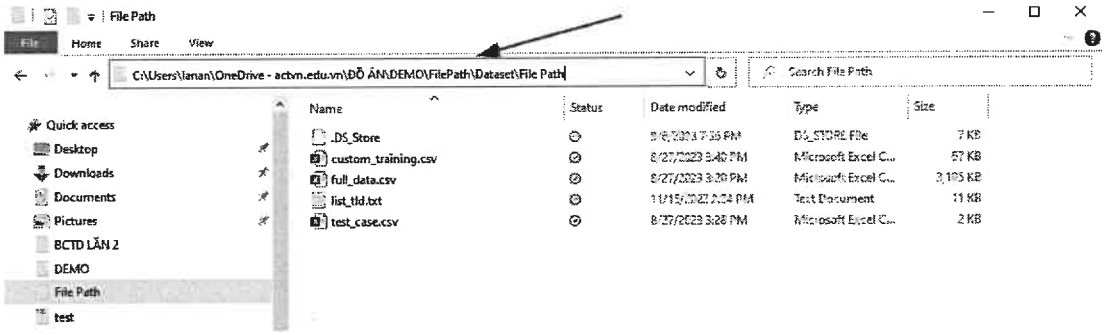
+ Tệp văn bản: lưu trữ nội dung dạng chữ viết, phổ biến theo chuẩn mã hóa ASCII hoặc Unicode, với các phần mở rộng thường gặp như .txt, .doc, .docx, .pdf (đối với tài liệu dạng văn bản), cùng một số định dạng khác nữa.

+ Tệp đa phương tiện: bao gồm hình ảnh, âm thanh và video.

+ Tệp thực thi: dùng để khởi chạy ứng dụng hoặc chương trình; trên Windows loại tệp này thường mang đuôi .exe, còn trên các hệ Unix/Linux thì có thể không kèm phần mở rộng hoặc sử dụng đuôi .sh.

+ Tệp dữ liệu: lưu thông tin phục vụ cho ứng dụng hoặc hệ thống vận hành, với một số định dạng thông dụng như .csv, .json, .xml và .sql.

Bên cạnh đó, còn tồn tại nhiều loại khác như tệp mã nguồn, tệp nén, tệp ảnh đĩa hay tệp cấu hình. Mỗi loại đều mang định dạng riêng và phục vụ mục đích khác nhau, được ứng dụng rộng rãi trong nhiều lĩnh vực của công nghệ thông tin và máy tính.



Hình 1.2. Hình ảnh đường dẫn tệp tin "File Path"

Tóm lại, có thể xem đường dẫn là một chuỗi ký tự xác định duy nhất một vị trí trong cấu trúc phân cấp - có thể là cây thư mục hoặc cây trang web. Với hệ thống tập tin thì mỗi tầng cấp bậc chính là một thư mục, còn đối với URL thì mỗi tầng lại tương ứng với một trang. Các thành phần trong đường dẫn được ngăn cách nhau bởi ký tự phân cách, ví dụ dấu gạch chéo lên (/), mỗi phần thể hiện một thư mục hoặc một trang riêng biệt trong cấu trúc phân cấp đó, minh họa như sau:

/home/user/python/test.py

Với ví dụ đường dẫn nêu trên, tệp test.py được đặt bên trong thư mục python; thư mục này lại nằm bên trong thư mục user, và user là con của thư mục home. Đây chính là một minh họa cụ thể cho khái niệm đường dẫn tệp, đồng thời đường dẫn cũng đóng vai trò không thể thiếu khi cấu thành nên một URL.

Đối với các hệ điều hành thuộc họ Unix như Linux hay macOS, ký tự dùng để phân cách các thành phần đường dẫn là dấu gạch chéo lên (/); trong khi đó, hệ điều hành Windows lại sử dụng dấu gạch chéo ngược (\) cho mục đích này.

1.2.2. Phân loại đường dẫn

Đường dẫn tệp tin thường được phân thành hai dạng cơ bản:

Đường dẫn tuyệt đối (Absolute Path): là dạng đường dẫn trình bày trọn vẹn hành trình từ thư mục gốc của hệ thống tệp cho tới nơi đặt tệp hoặc thư mục cần xác định, tức là liệt kê đầy đủ mọi thư mục cha nằm giữa gốc và đích đến. Minh họa cho đường dẫn tuyệt đối trong môi trường Windows:

C:\Users\Username\Documents\file.txt

Minh họa cho đường dẫn tuyệt đối trong môi trường Linux:

~/home/username/Documents/file.txt

Trên hệ điều hành Linux, ký hiệu dấu ngã (~) hay được dùng để đại diện cho thư mục cá nhân (home) của người dùng khi viết đường dẫn tệp.

Đường dẫn tương đối (Relative Path): trái ngược với loại trên, đây là cách xác định vị trí tệp hoặc thư mục dựa vào nơi người dùng đang đứng trong hệ thống tệp tại thời điểm đó; nó có thể xuất phát từ vị trí hiện hành hoặc từ một thư mục cha nhất định nào đó.

Chẳng hạn, giả sử thư mục làm việc hiện tại là '/home/username/Documents' và cần mở tệp 'file.txt' vốn đang nằm ngay trong thư mục đó, thì đường dẫn tương đối lúc này đơn giản chỉ là 'file.txt'.

Cú pháp và ký tự sử dụng trong đường dẫn tệp có thể khác nhau tùy theo từng nền tảng hệ điều hành: hệ thống Windows dùng dấu gạch chéo ngược '\' làm ký tự

ngăn cách thư mục, còn các hệ thống thuộc dòng Unix-like lại dùng dấu gạch chéo lên '/'.

1.2.3. Khái quát về đường dẫn độc

Có một dạng đường dẫn tệp tin đặc biệt nguy hiểm, gọi là đường dẫn tệp tin độc - khi bị truy cập, nó có khả năng gây hại cho máy tính của nạn nhân thông qua các hình thức như phát tán virus, mã độc, ransomware hay spyware. Để né tránh bị các công cụ bảo mật phát hiện, phần mềm độc hại thường áp dụng thủ thuật che giấu, đồng thời tạo ra những đường dẫn và tên tệp mang tính giả mạo.

Một số dấu hiệu nhận diện thường thấy ở đường dẫn tệp tin độc bao gồm:

- Có độ dài vượt quá 260 ký tự, tức là vượt ngưỡng giới hạn mà Windows cho phép.

- Có sự xuất hiện của các ký tự đặc biệt, ví dụ như %, $, #, @, ^, &, *, (,), [,], {, }, <, >...

- Mang phần mở rộng không tương xứng với bản chất tệp tin, chẳng hạn .exe, .bat, .cmd, .scr, .vbs...

- Đi kèm các đường link được rút gọn hoặc mang tính ẩn danh, ví dụ như bit.ly, goo.gl, tinyurl.com...

- Dẫn tới những trang web thiếu an toàn hoặc từng bị trình duyệt gắn cảnh báo.

Ví dụ:



Hình 1.3. Hình ảnh đường dẫn (path file) độc

C:\Users\Mr Binh\: đây chính là đường dẫn trỏ tới thư mục cá nhân của tài khoản người dùng "Mr Binh," nơi phần lớn tệp và thư mục bên trong thuộc quyền sở hữu của chính người dùng đó.

appData\Local\Temp\: đường dẫn này thường đóng vai trò chứa các tệp tạm thời; cụ thể thư mục Temp hay được các ứng dụng cũng như tiến trình hệ thống dùng để lưu tạm dữ liệu trong quá trình hoạt động.

csrss\smb\piaagseoqizukxx\: đây là một dãy thư mục con lồng nhau. Tên gọi csrss nhiều khả năng làm ngụy trang, đánh lừa người xem tưởng nhầm đây là tiến trình csrss.exe - vốn là một tiến trình hệ thống thiết yếu của Windows. Còn hai thư mục smb và piaagseoqizukxx dường như được sinh ra một cách tùy tiện nhằm khiến đường dẫn trông ngẫu nhiên, khó bị nhận diện.

ssleay32.dll: đây là tên tệp thuộc dạng DLL - một dạng thư viện liên kết động thường lưu trữ đoạn mã dùng chung cho nhiều ứng dụng khác nhau. Ở đây, cái tên ssleay32.dll có dấu hiệu là giả danh một tệp DLL vì DLL vốn thường liên quan tới chức năng mã hóa SSL/TLS.

1.2.4. Các phương pháp phòng chống và phát hiện path file độc

- Duy trì thói quen sao lưu định kỳ: Nhờ việc sao lưu được tiến hành đều đặn, dữ liệu quan trọng luôn có bản dự phòng và có thể phục hồi khi xảy ra sự cố, kể cả trường hợp bị nhiễm tệp độc hại hay thất thoát dữ liệu. Việc sao lưu diễn ra thường xuyên, kịp thời còn giúp xác minh được cả tính toàn vẹn của đường dẫn lẫn dữ liệu.

- Xây dựng quy tắc đặt tên hợp lý: Một quy tắc đặt tên chặt chẽ sẽ ngăn ngừa tình trạng tệp bị ghi đè ngoài ý muốn, đồng thời giữ cho đường dẫn lưu trữ tệp luôn đúng đắn, không bị thay đổi tùy tiện.

- Kiểm tra virus đối với tệp tin: Mọi tệp được tải lên cần trải qua bước quét bằng phần mềm diệt virus, và tệp nào không đạt yêu cầu thì phải bị ngăn chặn ngay. Biện pháp này góp phần đảm bảo tệp tải lên không mang mã độc, hạn chế nguy cơ lây lan tệp độc hại cũng như bảo vệ hệ thống trước hiểm họa virus.

- Ưu tiên dùng đường dẫn tuyệt đối: Nên tham chiếu tệp tin trên hệ thống bằng đường dẫn tuyệt đối thay vì đường dẫn tương đối, qua đó hạn chế được nguy cơ bị lợi dụng đường dẫn để tấn công.

- Hạn chế quyền truy cập tệp: Vận dụng nguyên tắc trao quyền tối thiểu, chỉ cho phép người dùng, tiến trình hay ứng dụng truy cập đúng những tệp tin thực sự cần thiết, không cấp thêm quyền vượt mức cần thiết.

- Tránh mở hoặc truy cập những đường dẫn có biểu hiện khả nghi, bất thường.

1.3. Kết luận chương 1

Nội dung chương 1 đã trình bày cái nhìn tổng quát về hai chủ đề chính: mã độc và đường dẫn tệp tin. Có thể hiểu mã độc (malware) là loại phần mềm được tạo ra nhằm mục đích phá hoại hoặc thực hiện những hành vi trái phép trên máy tính, hệ thống mạng mà không có sự đồng ý của chủ sở hữu; các dạng phổ biến của nó gồm virus, sâu máy tính, Trojan, ransomware cùng nhiều biến thể khác, có thể gây ra hậu quả từ đánh cắp thông tin quan trọng cho đến làm tê liệt cả hệ thống, do đó đòi hỏi phải hết sức thận trọng và có biện pháp phòng ngừa thích hợp nhằm giảm thiểu rủi ro cũng như thiệt hại có thể xảy ra. Chương này cũng đã phân tích về đường dẫn tệp tin độc hại, những rủi ro mà nó mang lại, cùng ý nghĩa của việc nhận diện và ngăn chặn kịp thời, đồng thời đưa ra một số giải pháp bảo vệ như sao lưu dữ liệu định kỳ, xây dựng quy ước đặt tên hợp lý và quét virus cho tệp tin. Tóm lại, việc nắm vững kiến thức về mã độc cũng như cách quản lý đường dẫn tệp tin đóng vai trò thiết yếu trong việc bảo vệ máy tính, dữ liệu trước các nguy cơ an ninh mạng, hướng đến sử dụng máy tính an toàn hơn, đồng thời góp phần ngăn ngừa thất thoát dữ liệu và bảo vệ quyền riêng tư khi hoạt động trực tuyến.

CHƯƠNG 2: PHÁT HIỆN MÃ ĐỘC SỬ DỤNG TRÍ TUỆ NHÂN TẠO

2.1. Tổng quan về học máy

Sự bùng nổ về khối lượng dữ liệu, cùng với chi phí lưu trữ ngày càng giảm và những bước tiến vượt bậc của công nghệ xử lý, chính là động lực thúc đẩy học máy phát triển mạnh mẽ trong thời đại hiện nay. Rất nhiều lĩnh vực đang triển khai xây dựng những mô hình học máy có năng lực xử lý tập dữ liệu lớn, phức tạp, cho ra kết quả nhanh chóng và chính xác ở quy mô lớn. Nhờ các công cụ học máy, tổ chức có thể nhận diện được cơ hội cũng như rủi ro tiềm ẩn một cách hiệu quả hơn.

Sức ảnh hưởng của học máy đã lan tỏa tới rất nhiều ngành nghề khác nhau. Riêng trong lĩnh vực bảo mật, học máy giữ vai trò then chốt trong việc phát hiện và ngăn chặn các nguy cơ đe dọa cả trực tuyến lẫn ngoại tuyến. Công nghệ này được ứng dụng để phát hiện xâm nhập trái phép, nhận diện các hành vi gian lận tín dụng, phân loại thư rác và đánh giá mức độ rủi ro, từ đó góp phần nâng cao an toàn thông tin và bảo vệ những dữ liệu quan trọng trong bối cảnh số hóa hiện nay.

2.1.1. Khái niệm Machine Learning

Học máy (Machine Learning) ra đời và phát triển như một nhánh của trí tuệ nhân tạo từ những năm 1950. Nền tảng của học máy là việc tìm kiếm mô hình phù hợp nhất áp dụng cho dữ liệu mới, dựa trên những dữ liệu đã có trước đó. Chính vì vậy, sự tiến bộ trong nghiên cứu học máy luôn gắn liền và song hành cùng sự gia tăng không ngừng của khối lượng dữ liệu.

Nhờ học máy, máy tính có khả năng tự học từ dữ liệu, không ngừng nâng cao hiệu suất qua từng trải nghiệm và đưa ra dự đoán mà không cần được lập trình một cách tường minh. Các ứng dụng của học máy trải rộng trên nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên, xử lý hình ảnh và thị giác máy tính, nhận dạng giọng nói và chữ viết, ngành ô tô, hàng không, sản xuất, sản sinh năng lượng, tính toán tài chính, sinh học,.... Bản chất của nó là phân tích và diễn giải các dữ liệu đã có từ trước. Tuy nhiên,

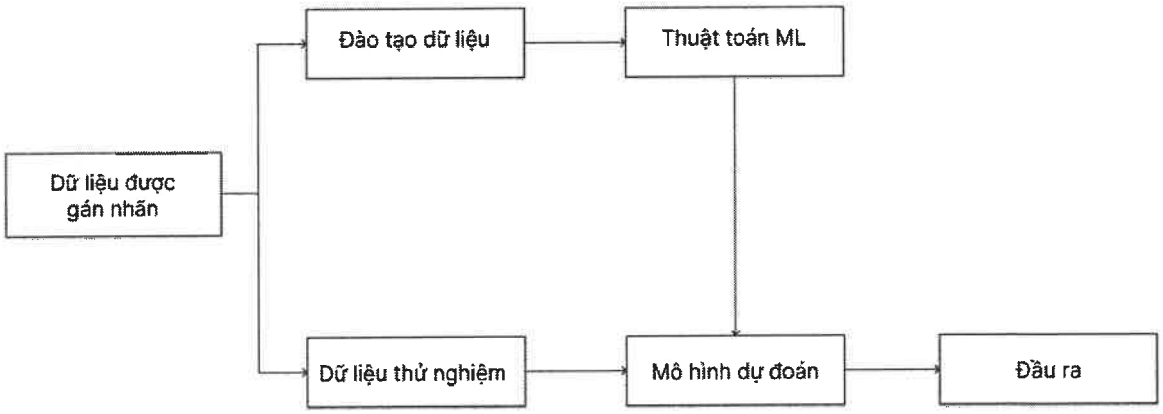
vì con người khó có thể tự mình phân tích và diễn giải khối lượng dữ liệu đó, các thuật toán và phương pháp học máy đã ra đời để đảm nhận công việc này. Đích đến quan trọng nhất của học máy là xây dựng nên những mô hình có khả năng tự huấn luyện, không ngừng cải thiện, nhận biết được các mẫu hình phức tạp và tìm ra lời giải cho những vấn đề mới dựa trên nguồn dữ liệu sẵn có.

2.1.2. Phân loại các mô hình Machine Learning

2.1.2.1. Học có giám sát (Supervised Learning)

Học có giám sát là phương pháp mà trong đó thuật toán dựa vào các cặp (input, outcome) đã biết trước để dự đoán outcome của một input mới. Các cặp dữ liệu này còn thường được gọi bằng tên khác là (dữ liệu, nhãn). Đây được xem là một trong những nhóm thuật toán được sử dụng phổ biến bậc nhất trong lĩnh vực Machine Learning.

Với học có giám sát, mô hình sẽ được huấn luyện trên nguồn dữ liệu đã gắn nhãn, sau đó được kiểm định trên nguồn dữ liệu chưa gắn nhãn. Quy trình cơ bản bắt đầu từ bước thu thập tập dữ liệu, rồi phân chia thành hai phần: dữ liệu huấn luyện và dữ liệu kiểm thử. Tiếp theo, các đặc trưng được trích xuất từ đầu vào sẽ đưa vào thuật toán để mô hình học được mối liên hệ giữa đặc trưng và nhãn tương ứng. Sau cùng, dữ liệu kiểm thử được đưa vào mô hình nhằm tạo ra các nhãn dự đoán tương ứng với từng đầu vào.



Hình 2.1. Kiến trúc cơ bản của Học có giám sát

Biểu diễn dưới dạng công thức toán học:

$X = \{x_1, x_2, \dots, x_N\}$: tập hợp biến đầu vào

$Y = \{y_1, y_2, \dots, y_N\}$: tập hợp nhãn tương ứng

Với x_i, y_i là các vector.

Tập hợp các cặp dữ liệu đã biết trước $(xi, yi) \in X \times Y$ được gọi là training data (tập dữ liệu huấn luyện). Dựa vào tập training data này, mục tiêu đặt ra là xây dựng một hàm số có khả năng ánh xạ từng phần tử của tập X sang phần tử tương ứng (mang tính xấp xỉ) thuộc tập Y:

$$y_i \approx f(x_i)\ \forall i = 1,2, \dots, N$$

Các thuật toán học có giám sát được phân chia thành hai nhóm chính:

- Classification (Phân loại): Dựa vào các đặc trưng riêng của dữ liệu để phân chia dữ liệu thành những lớp khác nhau.

- Regression (Hồi quy): Dựa trên một số đặc trưng đã có để đưa ra dự đoán hoặc kết luận về các đặc trưng còn lại của dữ liệu.

Classification

Phân loại là một kỹ thuật thuộc nhóm học có giám sát, được dùng để xác định hoặc dự đoán lớp/nhãn của một đầu vào dựa theo các đặc trưng vốn có của nó. Với phân loại, một chương trình sẽ học hỏi từ tập dữ liệu cho trước, để rồi từ đó phân loại dữ liệu mới vào một trong số các lớp hay nhóm nhất định, chẳng hạn Có hoặc Không, 0 hoặc 1, Thư rác hoặc Không phải Thư rác,... Những lớp này còn có thể được gọi bằng những tên khác như mục tiêu, nhãn hay danh mục.

Đích đến chính của thuật toán phân loại là xác định được danh mục ứng với một tập dữ liệu cho trước, và thuật toán này chủ yếu phục vụ mục đích dự đoán đầu ra cho dạng dữ liệu phân loại. Thuật toán đảm nhiệm việc phân loại trên tập dữ liệu được gọi là trình phân loại. Thuật toán phân loại thường được chia thành hai dạng sau:

- Binary Classifier: Trường hợp đầu ra của phân loại chỉ tồn tại hai khả năng, được gọi là Bộ phân loại nhị phân.

- Multi-class Classifier: Trường hợp đầu ra của phân loại có từ ba khả năng trở lên, được gọi là Trình phân loại nhiều lớp.

Regression

Hồi quy được xem là một công cụ quan trọng, được ứng dụng rộng rãi trong cả thống kê lẫn học máy. Đích đến chủ yếu của các bài toán hồi quy là đưa ra dự đoán về những giá trị liên tục cho nhãn đầu ra, xuất phát từ dữ liệu đầu vào. Đầu ra này được xác lập dựa trên những gì mà mô hình đã tiếp thu được trong suốt quá trình huấn luyện. Nói cách khác, mô hình hồi quy khai thác các đặc trưng của dữ liệu đầu vào (biến độc lập) cùng với giá trị đầu ra liên tục tương ứng (biến phụ thuộc), từ đó xác định mối tương quan giữa đầu vào và đầu ra.

Phân loại các mô hình hồi quy

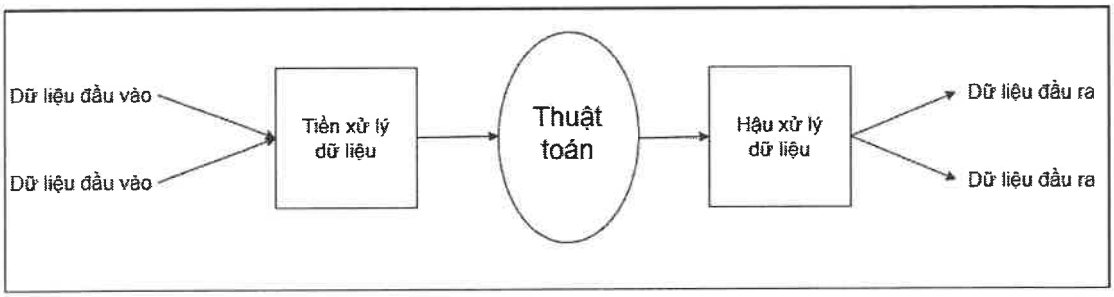
- Mô hình hồi quy đơn giản: Là dạng cơ bản nhất, trong đó dự đoán được xây dựng dựa trên duy nhất một đặc trưng của dữ liệu.

- Mô hình hồi quy bội: Ở dạng mô hình này, dự đoán được xây dựng dựa trên nhiều đặc trưng khác nhau của dữ liệu.

2.1.2.2. Học không giám sát (Unsupervised Learning)

Đối với thuật toán này, chỉ có dữ liệu đầu vào là được biết đến, còn đầu ra hay nhãn thì hoàn toàn không xác định trước. Thuật toán học không giám sát tập trung khai thác và phân tích cấu trúc bên trong của dữ liệu, phục vụ cho các tác vụ như phân nhóm (clustering) hay giảm chiều dữ liệu (dimension reduction), từ đó góp phần tối ưu hóa việc lưu trữ và xử lý thông tin.

Trong học không giám sát, hệ thống AI sẽ tự sắp xếp những thông tin chưa được phân loại thành các nhóm dựa trên điểm tương đồng và khác biệt giữa chúng, dù không hề có danh mục nào được cung cấp trước. Đích đến quan trọng nhất của phương pháp này là phát hiện ra những mẫu hình ẩn giấu, đặc thù bên trong nguồn dữ liệu chưa được gán nhãn.



Hình 2.2. Kiến trúc cơ bản của Học không giám sát

Thuật toán học không giám sát được phân chia thành hai nhóm chính:

- Clustering (Phân cụm): Dựa trên các đặc trưng tương đồng của dữ liệu để gộp các mẫu dữ liệu thành từng nhóm (cụm).

- Dimension reduction (Giảm chiều dữ liệu): Là quá trình thu giảm số lượng đặc trưng (hay số chiều) của một tập dữ liệu, song vẫn giữ lại được lượng thông tin có ý nghĩa nhiều nhất có thể.

Clustering

Đây là dạng bài toán thực hiện phân chia toàn bộ tập dữ liệu X thành từng nhóm nhỏ, dựa trên mức độ liên quan giữa các dữ liệu nằm trong cùng một nhóm. Chẳng hạn, việc phân nhóm khách hàng dựa theo hoạt động mua hàng của họ. Điều này cũng tương tự như việc chia một bó hoa thành từng nhóm loại hoa khác nhau, căn cứ vào hình dáng và màu sắc đặc trưng của chúng.

Dimension reduction

Giảm chiều dữ liệu (Dimension reduction) là quá trình chuyển đổi dữ liệu từ một không gian có số chiều lớn sang không gian có số chiều nhỏ hơn, trong khi vẫn giữ được bản chất vốn có của dữ liệu gốc, nhờ đó giúp giảm bớt độ phức tạp của mô hình, nâng cao hiệu suất của thuật toán học máy, hoặc khiến việc trực quan hóa dữ liệu trở nên dễ dàng hơn.

2.1.2.3. Học bán giám sát (Semi-Supervised Learning)

Học bán giám sát chính là sự giao thoa giữa học có giám sát và học không giám sát. Phương pháp này tận dụng một lượng nhỏ dữ liệu đã gắn nhãn cùng với một lượng lớn dữ liệu chưa gắn nhãn, qua đó thừa hưởng lợi ích của cả hai hướng tiếp cận, đồng thời tránh được khó khăn trong việc tìm kiếm số lượng lớn dữ liệu đã gắn nhãn. Nói cách khác, có thể huấn luyện được một mô hình có khả năng gán nhãn cho dữ liệu mà không đòi hỏi phải sử dụng quá nhiều dữ liệu huấn luyện đã gắn nhãn.

Trên thực tế, phần lớn các bài toán trong lĩnh vực Machine Learning đều rơi vào nhóm này, bởi việc thu thập dữ liệu có nhãn thường tốn kém cả về thời gian lẫn chi phí. Đặc biệt, có những loại dữ liệu, ví dụ như hình ảnh y học, chỉ có thể được gắn nhãn bởi những chuyên gia trong ngành. Trong khi đó, dữ liệu chưa gắn nhãn lại có thể dễ dàng thu thập với chi phí thấp, chẳng hạn từ internet.

2.1.2.4. Học tăng cường – Reinforcement Learning

Học tăng cường là một phương pháp học máy, trong đó tác nhân học cách ra quyết định trong môi trường bằng việc thực hiện các hành động rồi đánh giá kết quả mang lại từ những hành động đó. Khi thực hiện hành động đúng đắn, tác nhân sẽ nhận được phản hồi tích cực, còn khi thực hiện hành động sai, nó sẽ nhận phản hồi tiêu cực hoặc chịu hình phạt.

Trong phương pháp học tăng cường, tác nhân tự học hỏi thông qua phản hồi nhận được từ những kinh nghiệm tích lũy được, mà không cần đến bất kỳ dữ liệu gắn nhãn nào.

Tác nhân sẽ tương tác với môi trường xung quanh, từ đó tự mình khám phá và không ngừng cải thiện. Mục tiêu cốt lõi của một tác nhân trong học tăng cường chính là nâng cao hiệu suất bằng cách đạt được kết quả tối đa có thể.

2.2. Một số thuật toán trong phân loại dữ liệu

2.2.1. Cây quyết định

2.2.1.1. Khái niệm

Là một trong những mô hình học máy được ứng dụng rộng rãi, cây quyết định (Decision Tree) có thể giải quyết đồng thời cả bài toán phân loại (classification) lẫn bài toán dự đoán (regression). Về bản chất, đây là một cấu trúc dạng cây, trong đó dữ liệu được phân tách dần thành các nhánh thông qua các quyết định dựa trên đặc trưng hoặc thuộc tính vốn có của dữ liệu. Đích đến quan trọng nhất của mô hình này là tìm ra cách phân chia dữ liệu thành các nhóm nhỏ sao cho mức độ tách biệt hay độ tinh khiết giữa chúng đạt giá trị cao nhất, thông qua các tiêu chí tối ưu như sự gia tăng thông tin (information gain) hoặc mức giảm độ sai lệch (gini impurity).

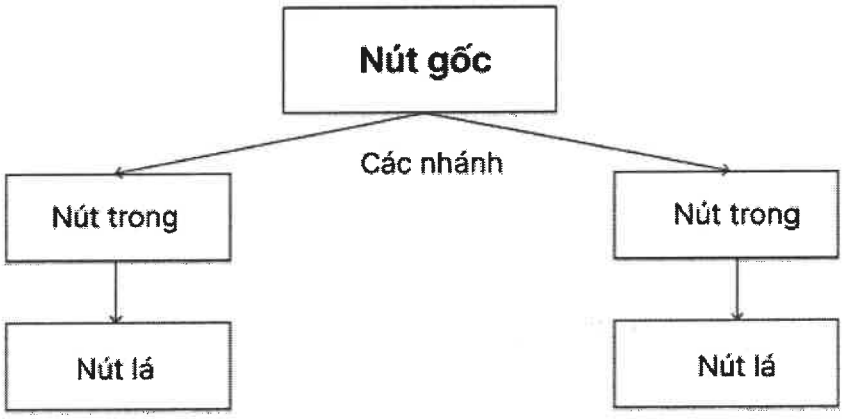
Xuất phát từ nút gốc, quá trình xây dựng cây sẽ liên tục tách dữ liệu thành các nhánh con bằng việc áp dụng những quy tắc quyết định căn cứ trên giá trị của từng đặc trưng. Quá trình phân tách này được lặp lại nhiều lần cho tới khi thỏa mãn một điều kiện dừng nào đó, chẳng hạn khi một nhóm dữ liệu đã đạt được mức độ tinh khiết cần thiết hoặc khi cây đã chạm tới độ sâu tối đa được quy định.

Cấu trúc hoàn chỉnh của một cây quyết định bao gồm ba thành phần: nút gốc, các nút trong và các nút lá.

- Nút gốc: là vị trí cao nhất của cây, thể hiện biến đặc trưng phù hợp nhất để thực hiện việc phân chia dữ liệu ban đầu.

- Nút quyết định (nút trong): biểu thị cho thao tác phân chia dữ liệu thành nhiều đoạn nhỏ hơn, với mục tiêu hướng tới là các nút con đạt được tính đồng nhất.

- Nút lá: là nơi thể hiện kết quả dự đoán sau cùng của mô hình.



Hình 2.3. Mô hình chung về cây quyết định

Một trong những điểm mạnh của cây quyết định là tính dễ hiểu và khả năng diễn giải cao, nhất là đối với những cây có kích thước nhỏ. Dù vậy, cần lưu ý rằng nếu không được kiểm soát chặt chẽ, mô hình này rất dễ rơi vào tình trạng quá khớp (overfitting) với dữ liệu huấn luyện, vì thế việc tối ưu hóa các siêu tham số, chẳng hạn như độ sâu của cây, là điều cần được quan tâm đúng mức.

2.2.1.2. Thuật toán CLS

Vào những năm 50 của thế kỷ 20, Hovland và Hint đã đề xuất thuật toán này trong hệ thống Concept Learning System (CLS), về sau được gọi ngắn gọn là thuật toán CLS. Chiến lược thiết kế của CLS dựa trên nguyên tắc chia để trị theo hướng từ trên xuống, với các bước thực hiện như sau:

- Bước 1: Khởi tạo một nút T chứa toàn bộ các mẫu thuộc tập huấn luyện.

- Bước 2: Trường hợp toàn bộ các mẫu trong T cùng thuộc một lớp và thuộc tính quyết định nhận giá trị:

+ "yes" thì gán nhãn cho nút T là "yes" và dừng lại. T lúc này là nút lá.

+ "no" thì gán nhãn cho nút T là "no" và dừng lại. T lúc này là nút lá.

- Bước 3: Ngược lại, nếu các mẫu của tập huấn luyện thuộc đồng thời cả hai lớp "yes" và "no" thì:

+ Chọn một thuộc tính X trong tập thuộc tính của tập mẫu dữ liệu, X có các giá trị v1, v2, …vn.

+ Chia tập mẫu trong T thành các tập con T1, T2,…., Tn. chia theo giá trị của X.

+ Tạo n nút con Ti (i=1,2…n) với nút cha là nút T.

+ Tạo các nhánh nối từ nút T đến các nút Ti (i=1,2…n) là các thuộc tính của X.

- Bước 4: Lặp lại quá trình trên đối với các nút con Ti (i=1,2..n) rồi quay trở lại bước 2.

2.2.1.3. Thuật toán ID3

Tên gọi ID3 là viết tắt của cụm từ Iterative Dichotomiser 3, xuất phát từ việc thuật toán này liên tục thực hiện phân đôi các tính năng thành hai hoặc nhiều nhóm tại mỗi bước xử lý. Được phát minh bởi Ross Quinlan, ID3 dùng để xây dựng cây quyết định từ tập dữ liệu và hiện là thuật toán được sử dụng phổ biến nhất cho mục đích này.

Những đặc điểm nổi bật của thuật toán ID3:

- ID3 có nguy cơ khớp quá mức với dữ liệu huấn luyện; do đó, để hạn chế hiện tượng quá khớp, nên ưu tiên lựa chọn cây quyết định có kích thước nhỏ thay vì cây lớn.

- Mặc dù thường cho ra kết quả là những cây có kích thước nhỏ, nhưng không phải trong mọi trường hợp thuật toán đều tạo ra được cây nhỏ nhất.

- Việc áp dụng ID3 cho dữ liệu liên tục gặp nhiều khó khăn hơn, bởi khi một thuộc tính nào đó mang giá trị liên tục, số lượng vị trí có thể phân chia trên thuộc tính đó sẽ tăng lên, khiến việc xác định điểm chia tối ưu trở nên tốn kém về mặt thời gian.

Những mặt lợi và hạn chế của thuật toán ID3:

- Ưu điểm:

+ Bản thân cây quyết định có khả năng tự giải thích và có thể được tái tạo lại một cách dễ dàng khi cần. Với số lượng nút lá ở mức hợp lý, ngay cả người không am hiểu chuyên môn cũng có thể nắm bắt được cấu trúc của cây. Ngoài ra, cây quyết

định còn có thể được quy đổi thành các tập luật nhằm phục vụ cho những ứng dụng khác.

+ Cây quyết định có khả năng xử lý được cả thuộc tính dạng tên lẫn thuộc tính dạng số trong dữ liệu đầu vào.

+ Cách biểu diễn của cây quyết định mang tính linh hoạt cao, có thể mô hình hóa được mọi dạng giá trị rời rạc.

+ Các thuộc tính dư thừa hay không liên quan có thể được xử lý một cách dễ dàng, ngoại trừ trường hợp giữa chúng tồn tại sự tương tác lẫn nhau.

- Nhược điểm:

+ Không gian tìm kiếm của cây quyết định có khả năng tăng trưởng theo cấp số nhân.

+ Đa số các thuật toán, điển hình như ID3 hay C4.5, đều yêu cầu thuộc tính mục tiêu bắt buộc phải mang giá trị rời rạc.

+ Sự tương tác qua lại giữa các thuộc tính không được tính đến.

+ Mỗi ranh giới quyết định chỉ gắn liền với duy nhất một thuộc tính.

Quy trình xây dựng cây quyết định

- Sử dụng toàn bộ dữ liệu hiện có làm đầu vào

- Xác định giá trị Entropy của biến mục tiêu cùng với các thuộc tính dùng để dự đoán

- Tính toán độ lợi thông tin cho toàn bộ các thuộc tính

- Lựa chọn thuộc tính có độ lợi thông tin lớn nhất để đặt làm nút gốc

- Tiếp tục lặp lại các bước tương tự trên từng nhánh cho tới khi nút quyết định của mỗi nhánh được xác lập hoàn chỉnh.

2.2.1.4. Thuật toán C4.5

Là một thuật toán phân lớp dữ liệu xây dựng trên nền tảng cây quyết định, C4.5 được ứng dụng khá phổ biến trong lĩnh vực khai phá dữ liệu đối với các cơ sở dữ liệu có quy mô nhỏ. Do vận hành theo cơ chế lưu trữ dữ liệu trong bộ nhớ, thuật toán này tỏ ra thích hợp hơn khi làm việc với các cơ sở dữ liệu có kích thước nhỏ. Điểm đáng chú ý của C4.5 nằm ở cơ chế tái sắp xếp dữ liệu tại từng nút trong suốt quá trình xây dựng cây. Ngoài ra, thuật toán còn áp dụng một kỹ thuật riêng để biến đổi cây quyết

định thành danh sách các quy tắc if-then đã được sắp xếp, qua đó giúp đơn giản hóa các luật phân lớp; kỹ thuật này vừa thu gọn kích thước của tập luật, vừa duy trì được độ chính xác tương đương so với khi sử dụng trực tiếp nhánh cây quyết định.

Nền tảng tư tưởng trong việc xây dựng cây quyết định của C4.5 bắt nguồn từ phương pháp Hunt đã được nghiên cứu từ trước đó. Thuật toán này áp dụng chiến lược phát triển cây theo độ sâu (depth-first strategy), tức là các nhánh của cây sẽ liên tục được phân tách cho đến khi không còn khả năng tách tiếp, khác với cách phát triển theo chiều rộng mà một số thuật toán khác sử dụng.

Mã giả của thuật toán C4.5:

(1) ComputerClassFrequency(T);

(2) if OneClass or FewCases

return a leaf;

Create a decision node N;

(3) ForEach Attribute A

ComputeGain(A);

(4) N.test=AttributeWithBestGain;

(5) if (N.test is continuous)

find Threshold;

(6) ForEach T' in the splitting of T

(7) If (T' is Empty)

Child of N is a leaf

else

(8) Child of N=FormTree(T');

(9) ComputeErrors of N;

return N

So với các thuật toán khác, C4.5 sở hữu một số đặc điểm riêng biệt, bao gồm: cơ chế lựa chọn thuộc tính để kiểm tra tại mỗi nút, cách thức xử lý đối với dữ liệu bị thiếu giá trị, biện pháp tránh hiện tượng "quá vừa" dữ liệu, phương pháp ước lượng độ chính xác và cơ chế cắt tỉa cây.

C4.5 dùng Gain-entropy làm độ đo lựa chọn thuộc tính “tốt nhất”.

Đa số các hệ thống đều hướng tới việc xây dựng cây có kích thước càng nhỏ càng tốt, bởi cây nhỏ vừa dễ hiểu hơn vừa dễ dàng đạt được độ chính xác dự đoán cao hơn. Vì tính cực tiểu của cây quyết định không thể được đảm bảo tuyệt đối, C4.5 vận dụng các nghiên cứu về tối ưu hóa, theo đó cách phân chia được lựa chọn là cách có độ đo lựa chọn thuộc tính đạt giá trị cực đại.

Hai độ đo được sử dụng trong C4.5 là information gain và gain ratio. RF(Cj,S) biểu diễn tần suất (Relative Frequency) các case trong S thuộc về lớp Cj.

Với là kích thước tập các case có giá trị phân lớp là Cj.là kích thước tập dữ liệu đào tạo.

Chỉ số thông tin cần thiết cho sự phân lớp: I(S) với S là tập cần xét sự phân phối lớp được tính bằng:

Sau khi S được phân chia thành các tập con S1, S2,…, St bởi test B thì information gain được tính bằng:

Test B sẽ được chọn nếu có G(S,B) đạt giá trị lớn nhất.

Tuy nhiên, việc sử dụng G(S,B) lại tồn tại một hạn chế là có xu hướng ưu tiên các phép kiểm tra cho ra số lượng kết quả lớn; chẳng hạn, G(S,B) sẽ đạt giá trị cực đại khi mỗi Si chỉ chứa đúng một case đơn lẻ. Để khắc phục vấn đề này, tiêu chuẩn gain ratio được đưa vào sử dụng thông qua việc bổ sung thêm thông tin tiềm năng (potential information) của chính mỗi phân hạch.

Test B sẽ được chọn nếu có tỉ số giá trị gain ratio lớn nhất.

Khi xây dựng mô hình phân lớp C4.5, việc xác định thuộc tính tối ưu có thể dựa trên một trong hai chỉ số là Information Gain hoặc Gain ratio, trong đó Gain ratio được xem là lựa chọn mặc định.

2.2.1.5. Ưu và hạn chế của cây quyết định

Thuộc nhóm các thuật toán học có giám sát, cây quyết định hướng tới mục tiêu xây dựng một mô hình được huấn luyện nhằm dự đoán lớp hoặc giá trị của biến mục tiêu, dựa trên các quy tắc học được rút ra từ dữ liệu huấn luyện. Mặc dù có thể áp dụng cho cả bài toán hồi quy lẫn phân loại, thuật toán này vẫn tồn tại song song cả những ưu điểm và hạn chế nhất định.

Ưu điểm

- Cây quyết định có khả năng dự đoán cả giá trị liên tục lẫn giá trị rời rạc, nghĩa là mô hình này hoạt động hiệu quả đối với cả nhiệm vụ hồi quy và phân loại.

- Cây quyết định xử lý được đồng thời cả dữ liệu dạng số lẫn dữ liệu mang giá trị tên thể loại, nhờ đó có tính linh hoạt cao hơn so với nhiều kỹ thuật khác vốn chỉ thích hợp với một dạng dữ liệu cố định. Chẳng hạn, các luật quan hệ chỉ áp dụng được cho biến dạng tên, còn mạng nơ-ron lại chủ yếu phục vụ cho việc xử lý các biến mang giá trị số.

- Mô hình này còn có thể được vận dụng để phân loại những dữ liệu không thể phân tách theo cách tuyến tính.

- Xét về bản chất, cây quyết định thuộc dạng mô hình hộp trắng, cho phép mỗi tình huống quan sát được trong mô hình dễ dàng lý giải bằng logic Boolean. Ngược lại, mạng nơ-ron lại là ví dụ tiêu biểu cho mô hình "hộp đen", do tiến trình tạo ra kết quả của nó vô cùng phức tạp, khiến việc diễn giải kết quả trở nên khó khăn và không dễ nắm bắt.

- Cây quyết định có thể xử lý một cách hiệu quả khối lượng dữ liệu lớn trong thời gian ngắn, nhờ đó ngay cả máy tính cá nhân cũng đủ khả năng phân tích một lượng dữ liệu đáng kể, giúp các nhà hoạch định chiến lược nhanh chóng đưa ra quyết định dựa trên kết quả phân tích thu được.

Nhược điểm

- Đối với các bài toán mà dữ liệu phụ thuộc liên tục vào yếu tố thời gian, cây quyết định lại tỏ ra không phù hợp, do gặp nhiều khó khăn khi xử lý các chuỗi thời gian mang tính phức tạp.

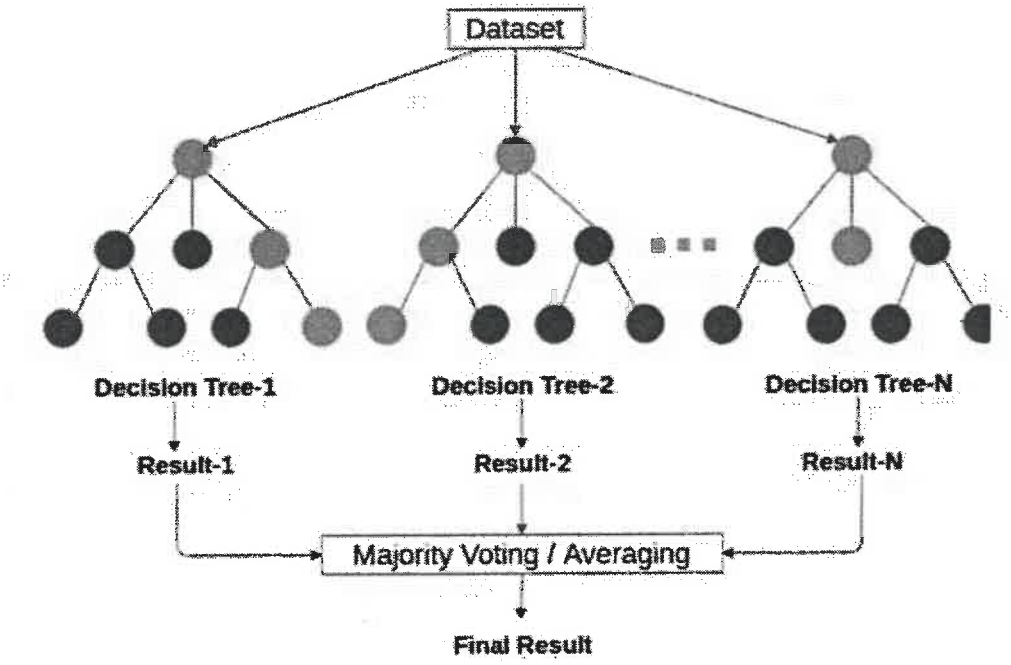
- Một khi số lượng lớp trong cây quyết định gia tăng, chi phí tính toán phục vụ cho việc xây dựng và duy trì mô hình cũng có thể tăng lên đáng kể, kéo theo hiệu suất suy giảm và nguy cơ phát sinh lỗi cao hơn.

2.2.2. Random Forest

2.2.2.1. Giới thiệu về Random Forest

Random Forest thuộc nhóm thuật toán học máy có giám sát, được xây dựng dựa trên nguyên lý ensemble learning (học kết hợp) - phương pháp phối hợp nhiều thuật

toán khác nhau, hoặc lặp lại cùng một thuật toán nhiều lần, nhằm tạo ra một mô hình dự đoán có độ chính xác và độ tin cậy cao hơn.



Hình 2.4. Mô hình về thuật toán Random Forest

Cụ thể, Random Forest xây dựng đồng thời nhiều cây quyết định (decision tree) - cùng một loại thuật toán - để hình thành nên một "khu rừng" các cây, và đây cũng chính là nguồn gốc của tên gọi "rừng ngẫu nhiên". Thuật toán này có khả năng ứng dụng cho cả bài toán phân loại lẫn bài toán hồi quy.

2.2.2.2. Cách thức hoạt động của Random Forest

Quy trình hoạt động của Random Forest có thể được mô tả qua đoạn mã giả (pseudocode) sau:

1. Lựa chọn ngẫu nhiên "k" đặc trưng (features) trong tổng số "m" đặc trưng, với k << m

2. Dựa trên tập "k" đặc trưng đó, xác định node "d" mang lại kết quả phân loại tốt nhất

3. Tiến hành phân chia các node con dựa theo node tốt nhất vừa xác định

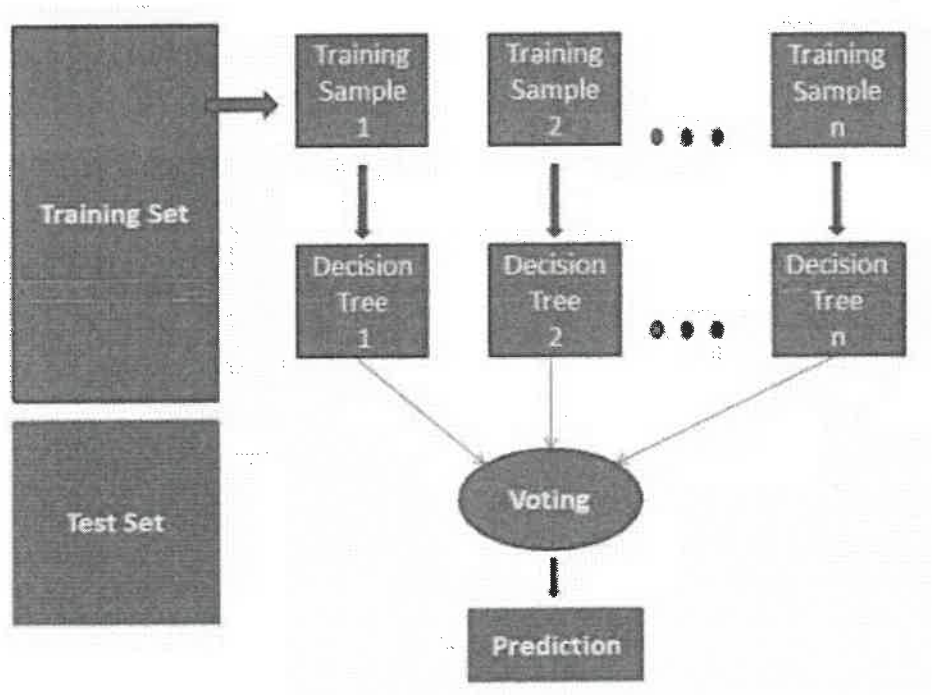
4. Lặp lại các bước 1 đến 3 cho tới khi đạt được k node

5. Lặp lại toàn bộ các bước 1 đến 4 nhằm tạo ra "n" cây quyết định

Sau khi Random Forest đã được huấn luyện, quá trình đưa ra dự đoán được thực hiện theo các bước sau:

1. Đưa các đặc trưng của tập kiểm tra (test features) vào từng Cây quyết định đã được xây dựng để thu được kết quả dự đoán, sau đó lưu lại toàn bộ các kết quả này vào một danh sách.

2. Thống kê số lượng phiếu bầu (vote) mà toàn bộ rừng cây dành cho mỗi kết quả dự đoán. 3. Chọn kết quả nhận được số phiếu bầu cao nhất làm kết quả dự đoán cuối cùng.



Hình 2.5. Cách thức hoạt động của Random Forest

2.2.2.3. Ưu và nhược điểm của thuật toán Random Forest

Ưu điểm

- Do được cấu thành từ nhiều cây quyết định, mỗi cây lại huấn luyện trên một tập con dữ liệu riêng, nên Random Forest hạn chế được hiện tượng sai lệch. Về bản chất, thuật toán này vận hành theo nguyên tắc "sức mạnh của số đông", nhờ đó độ chênh lệch chung được kéo giảm xuống.

- Đây là một thuật toán có tính ổn định cao: khi có thêm một điểm dữ liệu mới được bổ sung vào tập dữ liệu, kết quả tổng thể hầu như không bị xáo trộn đáng kể,

bởi dữ liệu mới đó nhiều lắm chỉ tác động tới một cây riêng lẻ, còn việc chi phối đồng thời toàn bộ các cây là điều khó xảy ra.

- Thuật toán này thể hiện hiệu quả tốt ở cả hai dạng bài toán: phân loại và hồi quy.

- Ngay cả khi dữ liệu tồn tại giá trị bị khuyết hoặc chưa được chuẩn hóa/chia tỷ lệ hợp lý, Random Forest vẫn cho kết quả xử lý khá tốt.

Nhược điểm

- Một hạn chế đáng kể của Random Forest là mức độ phức tạp trong tính toán: vì mô hình kết hợp một số lượng lớn cây quyết định, nên đòi hỏi tài nguyên tính toán nhiều hơn.

- Cũng chính vì độ phức tạp đó, thời gian cần thiết để huấn luyện mô hình thường lâu hơn so với các thuật toán khác có tính chất tương đương.

2.2.3. Logistic Regression

2.2.3.1. Giới thiệu về Logistic Regression

Hồi quy Logistic là một thuật toán học có giám sát được sử dụng để giải quyết bài toán phân loại dữ liệu. Mô hình này thích hợp với các biến phụ thuộc mang tính định tính hoặc định lượng nhưng chỉ nhận một trong hai giá trị (có/không), tức dạng nhị phân 0 hoặc 1. Đặc điểm này khiến nó trở nên phù hợp với bài toán phân loại bình luận của người dùng, cụ thể là phân tích quan điểm (sentiment analysis), trong đó đầu ra cần xác định là bình luận mang tính tích cực hay tiêu cực.

2.2.3.2. Mô hình Logistic

Trước tiên, hãy cùng xem xét mô hình logistic thông qua một ví dụ cụ thể: giả sử có hai yếu tố dự đoán x1 và x2, cùng với một biến phản hồi nhị phân Y, ký hiệu p = P (Y = 1). Ở đây, ta giả định rằng tồn tại một mối quan hệ tuyến tính giữa các biến dự đoán và tỷ lệ cược log (log-odds) của sự kiện Y = 1. Mối quan hệ tuyến tính đó có thể được biểu diễn bằng công thức toán học dưới đây, trong đó ℓ là tỷ lệ cược log, b là cơ số của logarit, còn lại là các thông số của mô hình:

$$\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

Có thể sử dụng lũy thừa để phục hồi tỉ lệ cược ta có:

$$\frac{p}{1-p} = b^{\beta_0+\beta_1 x_1+\beta_2 x_2}$$

Bằng thao tác đại số (chia cả tử và mẫu số cho $b^{\beta_0+\beta_1 x_1+\beta_2 x_2}$) ta thu được xác suất Y = 1 là:

$$p = x = \frac{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}}{b^{\beta_0+\beta_1 x_1+\beta_2 x_2}+1} = \frac{1}{1+b^{-(\beta_0+\beta_1 x_1+\beta_2 x_2)}} = S_b(\beta_0+\beta_1 x_1+\beta_2 x_2)$$

Trong đó Sb là hàm sigmoid với cơ sở b. Công thức trên cho thấy rằng sau khi β_i được sửa, chúng ta có thể dễ dàng tính toán tỷ lệ cược Y = 1 cho một quan sát nhất định, hoặc xác suất Y = 1 cho một quan sát nhất định. Trường hợp sử dụng chính của mô hình logistic là đưa ra một quan sát (x1, x2) và ước tính xác suất p mà Y = 1. Trong hầu hết các ứng dụng, cơ số b của logarit thường được coi là e. Tuy nhiên, trong một số trường hợp, việc truyền đạt kết quả có thể dễ dàng hơn bằng cách làm việc trong cơ sở 2 hoặc cơ sở 10.

Xét một ví dụ minh họa với cơ số b = 10, cùng các hệ số β0=−3, β1=1 và β2=2

Cụ thể, mô hình là:

$$\log_{10}\frac{p}{1-p} = \ell = -3 + x_1 + 2x_2$$

Ở đây, p biểu thị xác suất xảy ra sự kiện Y = 1.

Ý nghĩa cụ thể được diễn giải như sau:

- Hệ số β0=−3 chính là giao điểm với trục tung (chặn y), thể hiện tỷ lệ cược của sự kiện Y = 1 trong trường hợp các yếu tố dự đoán x1=x2=0. Thông qua phép tính lũy thừa, có thể thấy rằng khi x1=x2=0, tỷ lệ cược cho trường hợp Y = 1 bằng 1/ (1000 1) 1/1001. Tương tự, xác suất xảy ra sự kiện Y = 1 khi x1=x2=0 cũng được tính ra là 1/ (1000 1) 1/1001.

- Hệ số β1=1 cho biết rằng khi x1 tăng thêm 1 đơn vị, tỷ lệ cược sẽ tăng thêm 1 đơn vị tương ứng. Nói cách khác, mỗi khi x1 tăng 1, tỷ lệ cược cho sự kiện Y = 1 tăng theo hệ số 101. Cần lưu ý rằng xác suất của Y = 1 cũng tăng theo, nhưng mức tăng không lớn bằng mức tăng của tỷ lệ cược.

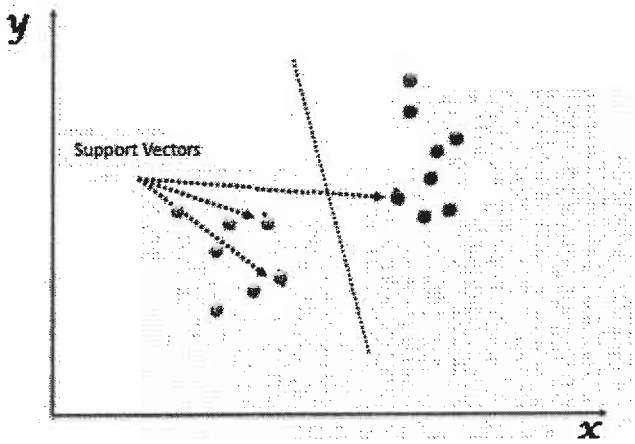
- Hệ số $\beta 2$=2 hàm ý rằng x2 tăng thêm 1 đơn vị sẽ khiến tỷ lệ tăng thêm 2 đơn vị; do đó, khi x2 tăng 1, tỷ lệ cược cho sự kiện Y = 1 sẽ tăng theo hệ số 102. Đáng chú ý, mức ảnh hưởng của x2 lên tỷ lệ cược log lớn gấp đôi so với x1, nhưng ảnh hưởng lên bản thân tỷ lệ cược lại lớn hơn tới 10 lần. Tuy nhiên, tác động lên xác suất của Y = 1 không lớn tới mức 10 lần như vậy - chỉ riêng tỷ lệ cược mới chịu ảnh hưởng lớn hơn 10 lần.

Việc xác định các tham số β dựa trên dữ liệu thực tế đòi hỏi phải tiến hành quá trình hồi quy logistic.

2.2.4. Support Vector Machines (SVM)

2.2.4.1. Giới thiệu về thuật toán Support Vector Machines

SVM (Support Vector Machine) là một thuật toán học có giám sát, được ứng dụng chủ yếu trong các bài toán phân loại, song cũng có thể mở rộng sang bài toán hồi quy. Cách thức vận hành của thuật toán này là biểu diễn dữ liệu dưới dạng các điểm nằm trong một không gian n chiều, với n tương ứng số lượng đặc trưng; mỗi điểm trong không gian đó tượng trưng cho một mẫu dữ liệu, còn các đặc trưng của nó chính là các giá trị tọa độ liên kết. Nhiệm vụ của SVM là xác định một "siêu phẳng" (hyperplane) - còn được gọi là đường phân chia - nhằm tách các lớp dữ liệu ra khỏi nhau, sao cho việc phân tách giữa hai nhóm đạt được mức độ rõ ràng cao nhất có thể. Trong trường hợp dữ liệu không thể được phân tách bằng một đường thẳng ngay trong không gian gốc, SVM có thể vận dụng kỹ thuật kernel để ánh xạ dữ liệu sang một không gian có số chiều cao hơn, nơi việc phân tách các lớp trở nên khả thi hơn.



Hình 2.6. Thuật toán Support Vector Machine

Có thể hiểu Support Vectors một cách đơn giản là những điểm dữ liệu nằm trên đồ thị tọa độ, còn nhiệm vụ của SVM là đi tìm một ranh giới tối ưu nhằm phân tách hai lớp dữ liệu đó.

Mô hình toán học

Khác với Logistic Regression, Support Vector Machine không đưa ra kết quả dưới dạng xác suất mà chỉ đơn thuần dự đoán nhãn kết quả là 0 hoặc 1.

2.2.4.2. Các ưu điểm và nhược điểm của SVM

Ưu điểm

- Cho hiệu quả cao đối với các bộ dữ liệu có số lượng đặc trưng lớn, chẳng hạn dữ liệu trong lĩnh vực tài chính hay y tế.

- Vẫn phát huy hiệu quả tốt ngay cả khi số lượng đặc trưng vượt trội hơn số lượng điểm dữ liệu.

- Chỉ sử dụng một tập con các điểm huấn luyện - được gọi là các vectơ hỗ trợ - trong hàm quyết định, nhờ đó giúp tiết kiệm bộ nhớ đáng kể.

- Có thể áp dụng nhiều loại hàm nhân (kernel) khác nhau cho hàm quyết định; ngoài các nhân phổ biến sẵn có, người dùng cũng có thể tự định nghĩa các nhân tùy chỉnh riêng.

Nhược điểm

- Trong trường hợp số lượng đặc trưng vượt trội hẳn so với số điểm dữ liệu, việc lựa chọn hàm nhân cùng các thành phần chính quy hóa sao cho tránh hiện tượng trùng lặp là điều hết sức quan trọng.

- SVM không trực tiếp tạo ra các ước lượng xác suất; những giá trị này chỉ có thể có được thông qua phương pháp xác nhận chéo (cross-validation) 5 lần, vốn khá tốn kém về mặt tính toán.

- Phát huy hiệu quả tốt nhất khi áp dụng cho các bộ mẫu có kích thước nhỏ, do thời gian huấn luyện khá cao.

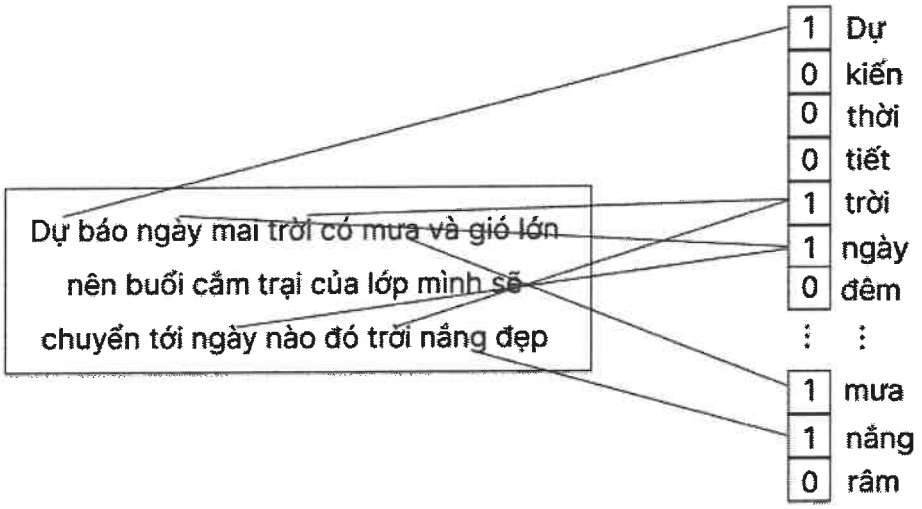
2.3. Ứng dụng học máy và phát hiện mã độc

2.3.1. Các phương pháp trích xuất đặc trưng trong xử lý văn bản

Về bản chất, máy tính không có khả năng nắm bắt trực tiếp ý nghĩa của ngôn từ trong văn bản. Do đó, muốn xử lý ngôn ngữ tự nhiên, cần một cơ chế chuyển đổi dữ liệu văn bản sang dạng biểu diễn mà máy tính có thể tiếp nhận và xử lý được. Cách tiếp cận được sử dụng rộng rãi nhất hiện nay là biểu diễn văn bản dưới dạng vector, theo đó mỗi từ hoặc cụm từ thuộc kho từ vựng sẽ được ánh xạ sang một vector trong không gian số thực, nhờ đó máy tính có thể phân tích và xử lý được thông tin ngôn ngữ.

2.3.1.1. Bag of Words (BoW)

Bag of Words (BoW), hay còn gọi là mô hình túi từ, là một trong các kỹ thuật nền tảng nhất trong xử lý văn bản, được dùng để trích xuất đặc trưng rồi vector hóa dữ liệu phục vụ cho việc huấn luyện các thuật toán học máy. Với mô hình BoW, mỗi mẫu văn bản được biểu diễn thành một vector sparse mà từng chiều tương ứng với một từ riêng biệt trong kho từ vựng. Giá trị tại mỗi chiều có thể đơn giản là 0 hoặc 1 thể hiện sự xuất hiện của từ, có thể là tần suất xuất hiện của từ trong văn bản, hoặc cũng có thể là một giá trị trọng số phản ánh mức độ quan trọng của từ đó trong ngữ cảnh.



*Hình 2.7. Ví dụ về mô hình túi từ (**Bag of Words**)*

Ví dụ về mô hình túi từ

Để hoạt động, mô hình túi từ đòi hỏi phải xây dựng một từ điển - tập hợp toàn bộ các từ đã xuất hiện trong văn bản - được hình thành bằng cách tách riêng từng từ theo cách tiếp cận unigram, mỗi từ trở thành một mục trong từ điển, sau đó tần suất xuất hiện của từ sẽ được tính toán dựa trên từ điển này. Điểm hạn chế của cách tiếp cận này là toàn bộ thứ tự sắp xếp cũng như ý nghĩa ngữ cảnh của từ trong câu đều bị bỏ qua; mô hình chỉ quan tâm từ đó có xuất hiện hay không, còn vị trí của nó trong câu hoàn toàn không được xét đến.

Một ví dụ cho mô hình Bag of Words:

Câu	Hôm	nay	trời	thật	đẹp	có	nắng
Hôm nay trời thật đẹp	1	1	1	1	1	0	0
Hôm nay trời có nắng	1	1	1	0	0	1	1

*Bảng 1. Bảng ví dụ mô hình **Bag of Words***

Áp dụng mô hình túi từ, hai câu "Hôm nay trời thật đẹp" và "Hôm nay trời có nắng" lần lượt được biểu diễn bằng các vector {1,1,1,1,1,0,0} và {1,1,1,0,0,1,1}, dựa trên từ điển ["Hôm", "nay", "trời", "thật", "đẹp", "có", "nắng"].

Điểm mạnh của mô hình túi từ nằm ở sự đơn giản trong triển khai, khả năng kết hợp thuận tiện với nhiều thuật toán học máy khác nhau, cùng tính dễ hiểu, chính vì thế nó được xem là mô hình cơ bản nhất trong lĩnh vực xử lý văn bản. Song, mô hình

này cũng tồn tại một số hạn chế nhất định như làm mất đi thông tin về trật tự cũng như ngữ nghĩa của từ, không thể hiện được mối tương quan giữa các từ với nhau, và dễ chịu tác động bởi những từ không mang nhiều giá trị thông tin.

2.3.1.2. TF-IDF

TF-IDF - viết tắt của "Term Frequency-Inverse Document Frequency" - là một đại lượng số học được ứng dụng trong truy xuất thông tin cũng như khai thác dữ liệu văn bản, nhằm đánh giá mức độ quan trọng của một từ trong một tài liệu cụ thể so với toàn bộ tập tài liệu (corpus). Kỹ thuật này được sử dụng phổ biến trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) cũng như phân tích văn bản nói chung. TF-IDF cho phép xác định mức độ quan trọng của từng từ trong tài liệu thông qua việc kết hợp giữa tần suất xuất hiện của từ đó trong tài liệu và mức độ hiếm gặp của nó xét trên toàn bộ corpus.

Giá trị TF được tính toán theo công thức sau:

$$tf(t,d) = \frac{f(t,d)}{max\{f(w,d): w \in d\}}$$

Với các ký hiệu như sau:

- tf(t,d): giá trị cần xác định, có miền giá trị trong khoảng [0,1]
- f(t,d): số lần từ t xuất hiện trong văn bản d
- max{f(w,d):w ∈ d): tần suất xuất hiện cao nhất của một từ bất kỳ trong văn bản đó

IDF phản ánh mức độ phổ biến của một từ xét trên toàn bộ tập văn bản, được xác định thông qua công thức:

$$idf(t,D) = log\frac{|D|}{|\{d \in D: t \in d\}|}$$

Với các thành phần:

- idf(t,D): giá trị cần xác định, mỗi từ tương ứng với một giá trị idf riêng biệt
- |D|: tổng số văn bản thuộc tập D
- $|\{d \in D: t \in d\}|$: số lượng văn bản d thuộc tập D có sự xuất hiện của từ t

Có thể nhận thấy rằng, từ nào xuất hiện với tần suất càng cao thì giá trị idf của nó càng tiến gần về 0, ngược lại, những từ ít phổ biến sẽ có giá trị idf tiến gần về 1.

Giá trị TF-IDF được tính bằng tích số giữa giá trị tf và giá trị idf:

tfidf(t,d,D) = tf(t,d) x idf(t,D)

Giá trị TF-IDF càng cao thì từ tương ứng càng được xem là quan trọng đối với văn bản đó.

Nhờ khả năng xác định mức độ quan trọng của từ trong văn bản cũng như xếp hạng các văn bản theo mức độ liên quan đến từ t, TF-IDF được áp dụng rộng rãi trong nhiều bài toán như truy xuất thông tin, khai thác văn bản và tìm kiếm.

TF-IDF có ưu điểm là việc tính toán và triển khai khá đơn giản, đồng thời phản ánh được mức độ quan trọng của từ trong văn bản. Tuy vậy, tương tự mô hình túi từ, kỹ thuật này vẫn bỏ qua thứ tự xuất hiện của từ, nên trong một số trường hợp vẫn chưa thể nắm bắt trọn vẹn ngữ nghĩa của văn bản.

2.3.1.3. Word2Vec

Word2Vec là một mô hình học máy được dùng để biểu diễn các từ vựng trong ngôn ngữ tự nhiên thành những vectơ số. Được Tomas Mikolov cùng nhóm nghiên cứu tại Google phát triển vào năm 2013, mô hình này đã nhanh chóng trở thành một công cụ then chốt trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) cũng như khai phá dữ liệu văn bản.

Nguyên lý hoạt động của Word2Vec dựa trên giả định rằng những từ mang ý nghĩa tương đồng thường có xu hướng xuất hiện gần nhau trong văn bản. Từ đó, mô hình xây dựng các vectơ biểu diễn sao cho những từ có ý nghĩa gần gũi sẽ có vị trí gần nhau trong không gian đa chiều. Việc huấn luyện Word2Vec có thể thực hiện theo hai phương pháp chính:

- Skip-gram: phương pháp này thực hiện dự đoán các từ ngữ cảnh (context words) xuất phát từ một từ mục tiêu (target word) cho trước, qua đó học được các biểu diễn từ có khả năng phản ánh cách từ đó được sử dụng trong ngữ cảnh.

- Continuous Bag of Words (CBOW): ngược lại với Skip-gram, phương pháp này dự đoán từ mục tiêu dựa trên các từ ngữ cảnh xung quanh, và thường cho hiệu quả tốt hơn trong trường hợp dữ liệu huấn luyện hạn chế.

Một khi đã hoàn tất huấn luyện, các vectơ từ do Word2Vec tạo ra có thể phục vụ cho nhiều ứng dụng khác nhau trong NLP, chẳng hạn như tìm kiếm từ tương đồng, phân loại văn bản, phát hiện các mối quan hệ ngữ nghĩa giữa từ ngữ, cùng nhiều tác vụ khác.

Nhờ vậy, Word2Vec đã khẳng định vai trò là một công cụ quan trọng trong biểu diễn từ vựng và khai phá dữ liệu văn bản, đóng góp đáng kể vào nhiều thành tựu của lĩnh vực NLP nói chung.

2.3.1.4. GloVe

GloVe - viết tắt của "Global Vectors for Word Representation" - là một mô hình dùng để biểu diễn từ vựng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), do Jeffrey Pennington, Richard Socher và Christopher Manning phát triển tại Đại học Stanford vào năm 2014. Chức năng chính của mô hình này là tạo ra các vectơ biểu diễn cho các từ trong ngôn ngữ tự nhiên.

GloVe hướng đến việc học các biểu diễn từ vựng sao cho các mối quan hệ về mặt ngữ nghĩa lẫn cú pháp giữa những từ được thể hiện rõ trong không gian vector. Ý tưởng nền tảng của mô hình là các từ có ý nghĩa tương đồng thường xuất hiện trong cùng ngữ cảnh văn bản. Để đạt mục tiêu đó, GloVe thực hiện tối ưu hóa hàm mất mát nhằm đảm bảo mức độ tương đồng ngữ nghĩa giữa các từ được phản ánh đúng trong vectơ biểu diễn tương ứng.

So với Word2Vec, GloVe sở hữu một số ưu điểm nổi bật như sau:

- Hiệu quả cao khi làm việc với các bộ dữ liệu lớn: GloVe thường cho kết quả vượt trội hơn Word2Vec khi lượng dữ liệu huấn luyện dồi dào.

- Kết hợp đồng thời thông tin cú pháp và ngữ nghĩa: các biểu diễn từ vựng do GloVe tạo ra thường thể hiện tốt hơn cả hai khía cạnh ngữ nghĩa và cú pháp.

- Có thể ứng dụng linh hoạt vào nhiều bài toán NLP khác nhau, như tìm kiếm từ tương đồng, phân loại văn bản, phát hiện quan hệ ngữ nghĩa, hay dịch máy. Nhờ những đặc điểm này, GloVe đã trở thành một công cụ quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên nói chung và biểu diễn từ vựng nói riêng.

2.3.1.5. FastText

FastText là một mô hình dùng để biểu diễn từ vựng cũng như các đoạn văn bản trong xử lý ngôn ngữ tự nhiên (NLP), do Facebook AI Research (FAIR) phát triển và giới thiệu vào năm 2016, nhanh chóng khẳng định vị thế là một công cụ quan trọng trong lĩnh vực NLP.

FastText sở hữu một số đặc điểm nổi bật như sau:

- Biểu diễn dựa trên subword: thay vì chỉ biểu diễn từng từ riêng lẻ, FastText còn khai thác thông tin từ các n-grams (chuỗi ký tự liên tiếp có độ dài n) cấu thành nên từ. Nhờ đó, FastText có thể xử lý hiệu quả các từ mới xuất hiện hoặc từ hiếm gặp, bằng cách suy ra biểu diễn của chúng dựa trên các n-grams sẵn có.

- Khả năng hỗ trợ đa nhiệm vụ: FastText có thể được áp dụng cho nhiều bài toán khác nhau trong NLP như phân loại văn bản, dự đoán từ khóa hay dịch máy, qua đó thể hiện tính đa năng của mô hình.

- Hiệu suất vượt trội với dữ liệu quy mô lớn: FastText thường đạt hiệu suất cao khi làm việc với các bộ dữ liệu lớn, đồng thời có tốc độ huấn luyện nhanh.

- Tính chất mã nguồn mở: là một dự án mã nguồn mở, FastText cho phép bất kỳ ai cũng có thể sử dụng cũng như tùy biến theo nhu cầu riêng.

Hiện nay, FastText được ứng dụng rộng rãi trong nhiều bài toán NLP như tìm kiếm thông tin, phân loại văn bản, phân loại tin tức, dự đoán từ khóa hay nhận diện ngôn ngữ. Đáng chú ý, chính cách biểu diễn từ dựa trên subword đã giúp FastText xử lý tốt sự đa dạng của ngôn ngữ và tạo ra các biểu diễn chất lượng cho những từ ngữ phức tạp.

2.4. Một số thuật toán học sâu

2.4.1. Mạng Perceptron đa tầng (MLP – Multi-layer Perceptron)

Là một dạng mạng nơ-ron nhân tạo phổ biến, MLP được sử dụng rộng rãi cho cả nhiệm vụ trích xuất đặc trưng lẫn phân loại dữ liệu. Kiến trúc của mạng MLP, được xây dựng dựa trên cái nguyên lý mô phỏng hoạt động của các nơ-ron thần kinh trong bộ não con người, đã được Daniel Svozil và các cộng sự [20] trình bày một cách chi tiết. Về cấu trúc, mạng MLP thường bao gồm tối thiểu 3 tầng, gồm 1 tầng đầu vào

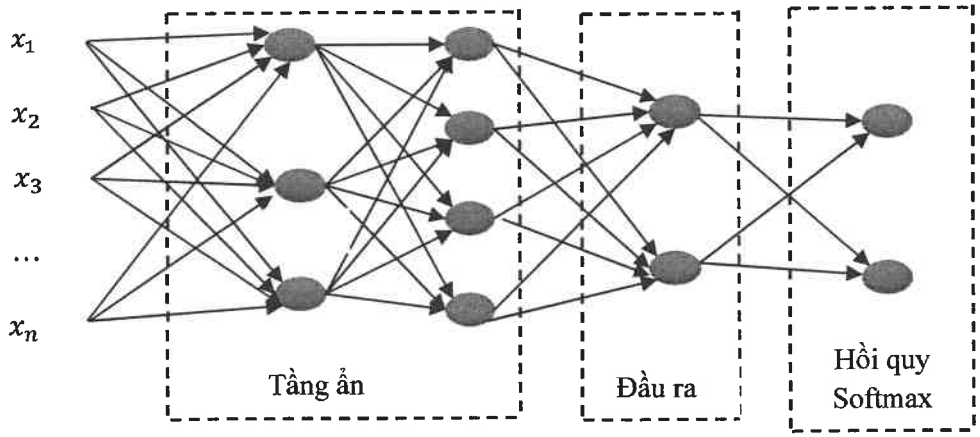
(input layer), 1 tầng đầu ra (output layer) và ít nhất 1 tầng ẩn (hidden layer) trở lên. Công thức tính toán tại các tầng ẩn được định nghĩa cụ thể tại công thức (3.1).

$$h_i^j = f(\sum_{k-1}^{n_{i-1}} w_{k,j}^{i-1} h_{i-1}^k) \quad i = 2,...,N and\ j = 1,...,n_i \tag{3.1}$$

Ký hiệu wk,ji thể hiện trọng số kết nối giữa nơ-ron k thuộc tầng ẩn $i-1$ với nơ-ron j thuộc tầng ẩn i, còn f đại diện cho hàm kích hoạt, có thể là đơn vị tuyến tính chỉnh lưu (ReLU - rectified linear unit) hoặc sigmoid [20, 21]. Trong phạm vi đề tài này, hàm sigmoid được lựa chọn sử dụng, với công thức thể hiện tại (3.2).

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.2}$$

Bên cạnh đó, nghiên cứu [21] của Hassan Ramchoun và các cộng sự cũng đã đưa ra những phân tích chuyên sâu về các phương pháp tối ưu hóa cấu trúc cũng như quá trình huấn luyện mạng MLP. Trong phạm vi đề án, mô hình MLP được sử dụng sẽ gồm nhiều tầng ẩn với số lượng nút (unit) được hiệu chỉnh linh hoạt, đồng thời kết hợp các kỹ thuật điều chuẩn (regularization) như dropout với nhiều tỉ lệ khác nhau nhằm xác định kiến trúc mang lại hiệu quả tối ưu nhất. Kết quả đầu ra của mạng MLP là một véc tơ hai chiều, tương ứng với hai lớp 0 và 1 của dữ liệu, sau đó được đưa qua mạng Softmax. Kiến trúc tổng quát của mô hình MLP được minh họa tại hình phía dưới.



Hình 3.2. Kiến trúc mô hình học sâu MLP

Hàm softmax [20] được biểu diễn cụ thể thông qua công thức (3.3) dưới đây:

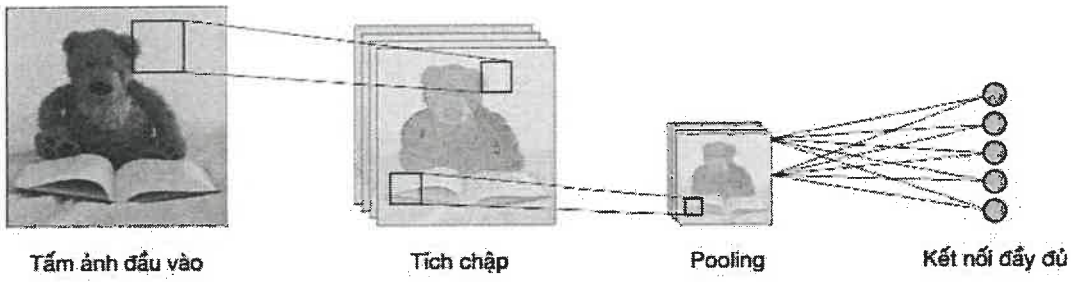
$$a_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \quad \forall i = 1, 2, ..., C$$

(3.3)

Trong đó: C là số lượng lớp $z = [z_1, z_2, ..., z_C]$ là véc tơ đầu ra của mạng MLP tương ứng với dữ liệu đầu vào cần được phân loại, a_i là xác suất để dữ liệu đầu vào đó rơi vào lớp thứ i được tính toán bằng hàm softmax. Lưu ý $\sum_{i=1}^{C} a_i = 1$ và hồi quy Softmax chỉ có nhiệm vụ tính toán xác suất rơi vào các lớp của đầu vào chứ không có ý nghĩa trong quá trình trích xuất đặc trưng, xác suất rơi vào lớp nào cao nhất thì lớp đó sẽ được gán cho đầu vào được dự đoán.

2.4.2. Mạng Nơ-ron Tích chập (CNN - Convolutional Neural Networks)

CNN [30, 31] là một dạng mạng nơ-ron sâu thuộc lĩnh vực học sâu, được sử dụng phổ biến cho các tác vụ phân tích hình ảnh trực quan. Phạm vi ứng dụng của CNN khá đa dạng, trải rộng từ phân loại ảnh (image classification) [22], phát hiện đối tượng (object detection) [23, 24], phân khúc hình ảnh (image segmentation) [25, 26], nhận dạng khuôn mặt (face recognition) [27], cho đến phân loại văn bản (text classify) [28] và mô hình hóa câu (modelling sentences) [29]. Kiến trúc truyền thống của một mạng CNN được minh họa tại Hình 3.3 dưới đây.



Hình 3. 1. Kiến trúc truyền thống của một mạng CNN

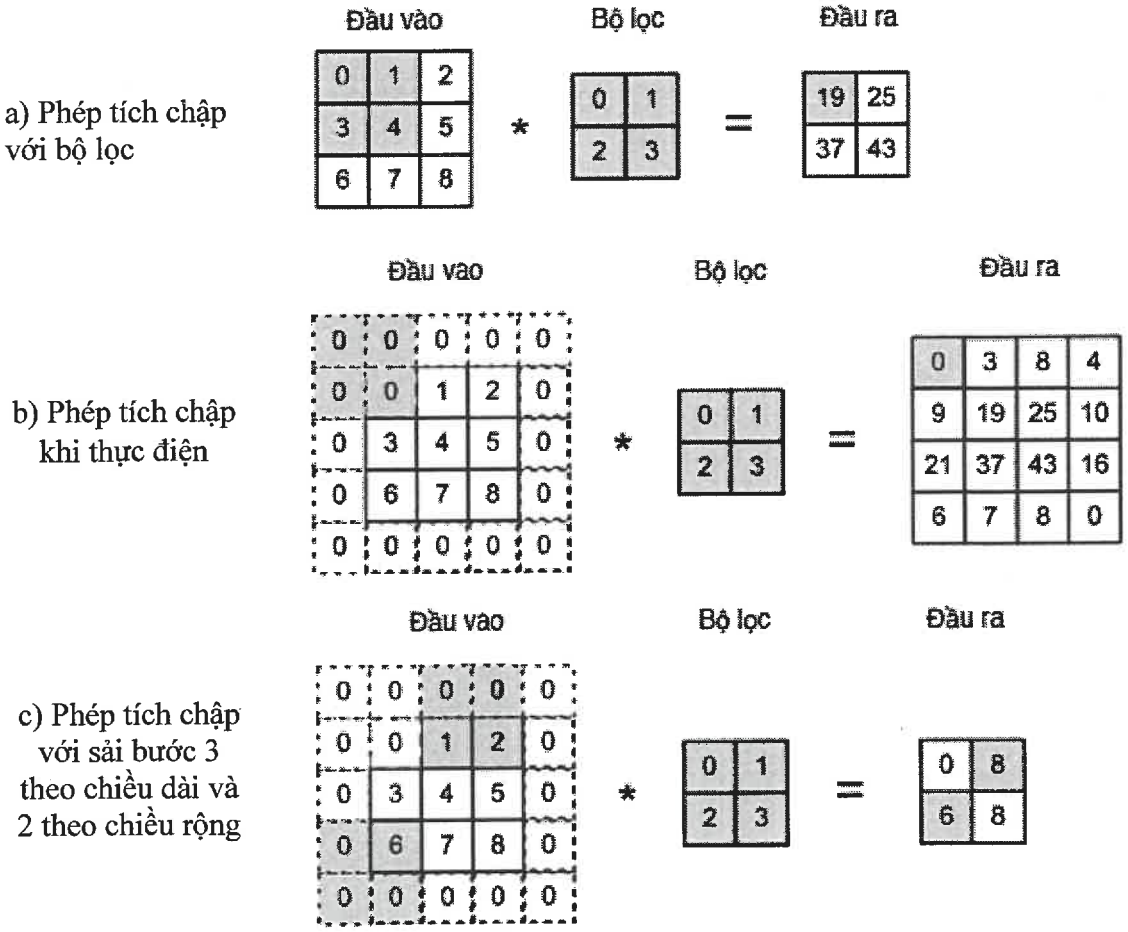
Về cơ bản, một mạng CNN được cấu thành từ các tầng chính sau đây:

- Tầng tích chập (Convolution): tầng này thực hiện phép tích chập bằng cách trượt các bộ lọc qua đầu vào theo từng chiều của nó, sau đó cộng thêm hệ số điều chỉnh để tạo ra đầu ra tương ứng. Ví dụ minh họa cho phép tích chập giữa

bộ lọc và đầu vào được thể hiện tại Hình 3.4.a. Tầng tích chập có các siêu tham số như sau:

+ Đệm (padding): Việc áp dụng tầng tích chập có thể gây ra hiện tượng mất mát một số điểm ảnh nằm ở biên của ảnh. Kỹ thuật đệm được sử dụng để khắc phục vấn đề này, thông qua việc bổ sung thêm các giá trị 0 vào phần biên của đầu vào. Cách thức áp dụng đệm trong phép tích chập được minh họa tại Hình 3.4.b.

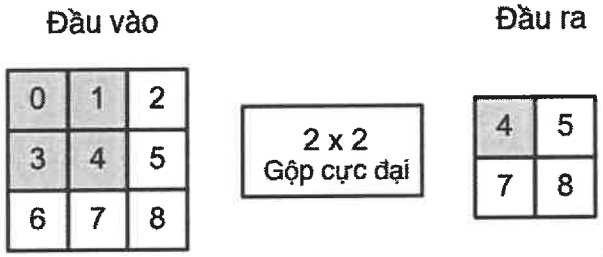
+ Sải bước (stride): Đây là đại lượng biểu thị số pixel mà cửa sổ trượt sẽ dịch chuyển qua sau mỗi phép tính, và việc tăng sải bước có thể làm giảm độ phân giải của đầu ra. Việc kết hợp sử dụng sải bước cùng với đệm trong phép tích chập được minh họa qua ví dụ tại Hình 3.4.c.



Hình 3. 2. Ví dụ phép tích chập

- Tầng gộp (Pooling): Vai trò của tầng gộp là làm giảm độ nhạy của các tầng tích chập trước sự thay đổi vị trí, đồng thời thu nhỏ kích thước các biểu diễn

dữ liệu. Giống với tầng tích chập, phép gộp cũng sử dụng một cửa sổ kích thước cố định trượt qua toàn bộ các vùng của đầu vào theo một bước nhảy xác định, và tại mỗi vị trí trượt qua sẽ cho ra một giá trị đầu ra duy nhất; tuy nhiên khác với tầng tích chập, tầng gộp không sử dụng bộ lọc. Hai kỹ thuật gộp thông dụng nhất là gộp cực đại (max pooling), lấy giá trị lớn nhất trong cửa sổ gộp, và gộp trung bình (average pooling hay mean pooling), lấy giá trị trung bình của các phần tử trong cửa sổ. Một ví dụ minh họa cho phép gộp cực đại được trình bày tại Hình 3.5.



Hình 3. 3. Ví dụ phép gộp cực đại

- Tầng kết nối đầy đủ (FC - Fully-connected): Đầu vào của tầng này là dữ liệu đã qua bước làm phẳng, trong đó từng phần tử đầu vào sẽ được kết nối với toàn bộ các nơ-ron thông qua phép nhân giữa ma trận và véc tơ. Trong các mô hình CNNs, tầng kết nối đầy đủ thường được đặt ở phần cuối của mạng, giữ vai trò tối ưu hóa mục tiêu của mô hình, ví dụ như độ chính xác phân lớp.

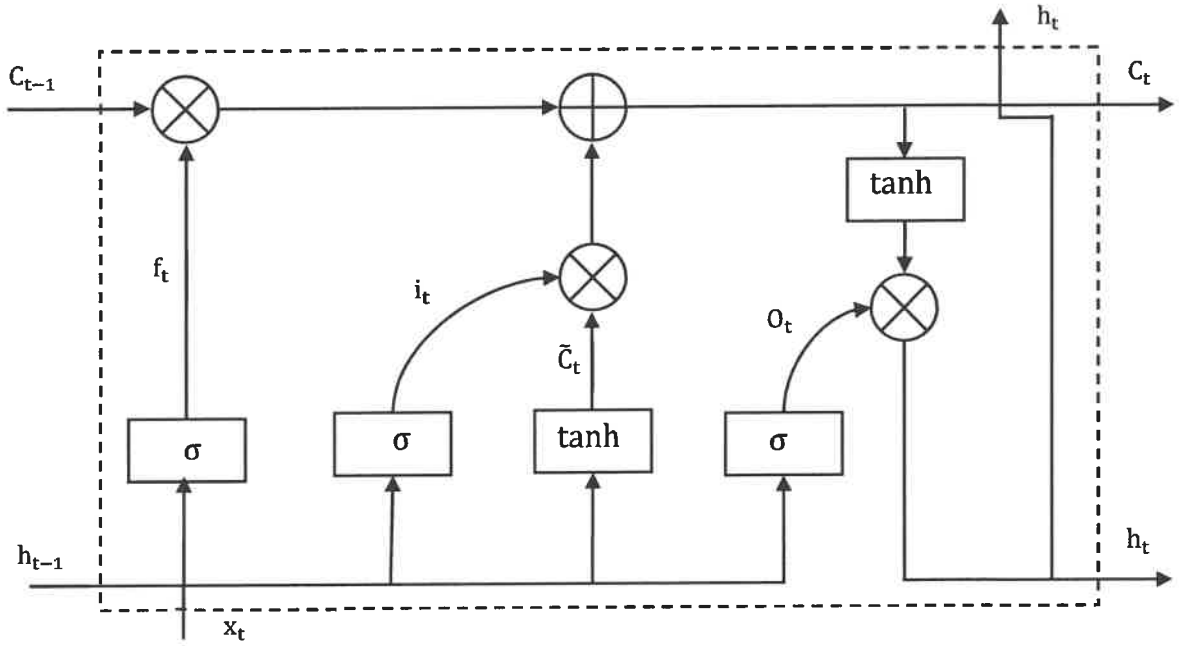
Hàm kích hoạt được lựa chọn sử dụng trong mạng CNN là ReLU, với công thức được biểu diễn tại (3.4).

$$f(x) = \max(0, x) \tag{3.4}$$

2.4.3. Bộ nhớ Ngắn hạn Dài (LSTM - Long Short-Term Memory)

Mạng LSTM cùng với khả năng ghi nhớ thông tin dài hạn đã được giới thiệu trong nghiên cứu [32], thể hiện qua cấu trúc các cổng bên trong từng ô nhớ. Cụ thể, việc kiểm soát một ô nhớ đòi hỏi ba loại cổng: cổng đầu vào (input gate), cổng quên (forget gate) và cổng đầu ra (output gate). Trước tiên, cổng quên có nhiệm vụ xác định thông tin nào cần loại bỏ khỏi trạng thái ô nhớ (cell state); sau đó, cổng đầu vào xác định thông tin nào sẽ được bổ sung vào trạng thái ô nhớ; và cuối cùng, cổng đầu

ra chịu trách nhiệm tính toán ra giá trị đầu ra mong muốn. Xuyên suốt quá trình này, trạng thái ô nhớ liên tục được truyền đi và cập nhật qua từng mắt xích. Cấu trúc chi tiết của một ô nhớ cơ bản trong mạng LSTM được minh họa tại Hình 3.6.



*Hình 3. 4. Mô hình cấu trúc một ô nhớ của **LSTM***

Ở mỗi thời điểm t, mạng LSTM sẽ tạo ra một trạng thái ẩn (hidden state) ht cùng một trạng thái ô nhớ Ct, với các công thức toán học cơ bản tương ứng được trình bày như sau:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (3.5)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (3.6)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t]) \quad (3.7)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3.8)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (3.9)$$

$$h_t = o_t * \tanh(C_t) \quad (3.10)$$

Trong số đó:

w biểu thị ma trận trọng số tương ứng của từng cổng,

b là véc tơ hệ số điều chỉnh (bias) gắn với từng cổng,

$*$ *ký hiệu phép nhân theo từng phần tử* $(element - wise)$.

2.4.4. Bộ nhớ Ngắn hạn Dài Hai chiều (BiLSTM - Bidirectional LSTM)

Tuy đã giải quyết được phần nào hạn chế về khả năng ghi nhớ so với các mạng nơ-ron hồi quy RNN truyền thống, mạng LSTM vẫn tồn tại nhược điểm là chỉ có thể học và ghi nhớ thông tin theo một chiều duy nhất, khiến cho một số thông tin quan trọng có thể bị bỏ sót trong quá trình xử lý. Nhằm khắc phục hạn chế này, nghiên cứu [33] đã đề xuất mạng học sâu BiLSTM, với mô hình được cấu thành từ hai thành phần: phần xuôi (forward LSTM) và phần ngược (backward LSTM). Nhờ đó, mô hình không những thừa hưởng được khả năng ghi nhớ dài hạn vốn có của LSTM, mà còn bổ sung thêm khả năng ghi nhớ thông tin theo cả hai chiều. Hai tầng LSTM này sinh ra hai trạng thái ẩn tương ứng là hfi (từ forward LSTM, mang thông tin từ các bước trước) và hbi (từ backward LSTM, mang thông tin từ các bước sau). Trạng thái ẩn cuối cùng hi được xác định bằng cách ghép nối hai trạng thái này theo công thức (3.11) dưới đây.

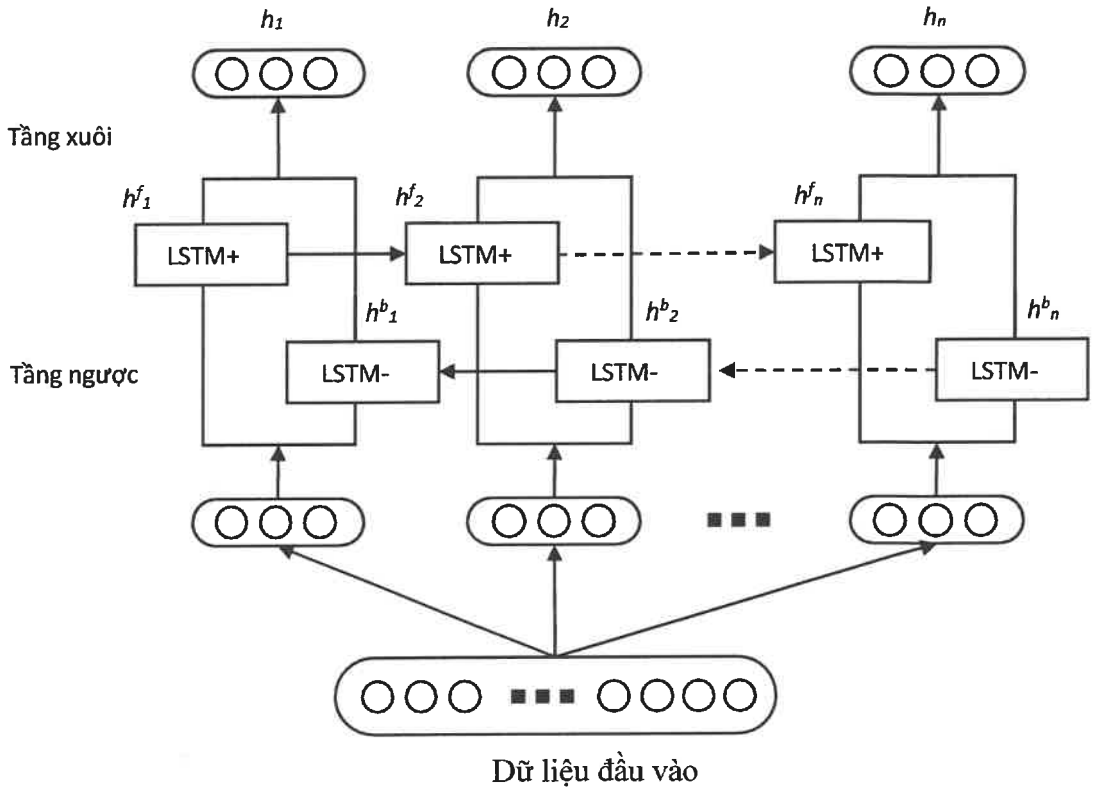
$$h_i = h_i^f \parallel h_i^b \tag{3.11}$$

Ở đây:

h_i biểu thị trạng thái ẩn tại bước thời gian i, tổng hợp thông tin từ cả hai hướng.

|| ký hiệu phép toán ghép nối (concatenation).

Kiến trúc của mạng BiLSTM, bao gồm hai thành phần forward LSTM và backward LSTM, được minh họa tại Hình 3.7 dưới đây.



Hình 3. 5. Mô hình mô tả 2 thành phần LSTM trong BiLSTM

Có thể thấy qua kiến trúc thể hiện tại rằng mạng BiLSTM có ưu điểm học thêm được một chiều thông tin bổ sung so với LSTM.

2.4. Kết luận chương

Ở chương mở đầu này, em đã tìm hiểu một cách tổng quan về mã độc cũng như con đường lây lan của chúng, nhằm nắm rõ vấn đề cần giải quyết và định hình các bước thực hiện của đề tài. Trước hết là phần tổng quan về mã độc, bao gồm các khái niệm và hành vi đặc trưng của từng loại, qua đó góp phần làm phong phú thêm và cung cấp những thông tin chi tiết phục vụ cho quá trình học sâu. Bởi tính linh hoạt và không ngừng biến đổi, các loại mã độc ngày càng che giấu mức độ nguy hiểm và tiềm ẩn tinh vi hơn khiến việc phát hiện trở nên khó khăn hơn, tuy nhiên khi đi sâu tìm hiểu bản chất thì có thể nắm bắt được cách thức hoạt động của chúng. Sau cùng là phương pháp lây nhiễm, mã độc có thể lan truyền dưới nhiều hình thức khác nhau, thông qua nhiều thiết bị trung gian để xâm nhập vào máy tính, thiết bị của người dùng. Nhờ đó, có thể khẳng định phần tổng quan lý thuyết này đóng vai trò hết sức cần thiết và quan trọng đối với mỗi đề tài nghiên cứu.

CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1. Dữ liệu thực nghiệm

Dữ liệu thực nghiệm sử dụng trong nghiên cứu là một tập hợp các đường dẫn dưới dạng chuỗi nhị phân được gán nhãn. Cụ thể, mỗi mẫu dữ liệu trong tập hợp bao gồm:

Chuỗi đường dẫn nhị phân: Biểu diễn các tệp hoặc đường dẫn trên hệ điều hành ở dạng nhị phân.

Nhãn (Label): Gán nhãn cho từng đường dẫn để phân loại:

- ✓ "*Clean*": Đường dẫn an toàn, không có dấu hiệu chứa mã độc.
- ✓ "*Malicious*": Đường dẫn độc hại, có chứa hoặc liên quan đến mã độc.

Chi tiết tập dữ liệu:

- Tổng số lượng mẫu: **[57596]**
- Số lượng mẫu gán nhãn "*Clean*": **[10000]**
- Số lượng mẫu gán nhãn "*Malicious*": **[47596]**

Định dạng dữ liệu:

Tập dữ liệu được lưu trữ trong file CSV với các cột chính:

- sha: Mã băm SHA đại diện cho từng đường dẫn.
- path_binary: Chuỗi đường dẫn dạng nhị phân.
- label: Nhãn tương ứng cho từng đường dẫn (Clean hoặc Malicious).

Dữ liệu được tiền xử lý để chuẩn hóa và đảm bảo tính nhất quán, bao gồm:

- Loại bỏ các mẫu bị lỗi hoặc không hợp lệ.
- Chuyển đổi đường dẫn sang định dạng nhị phân.
- Phân chia tập dữ liệu thành hai nhóm để huấn luyện và kiểm thử mô hình.

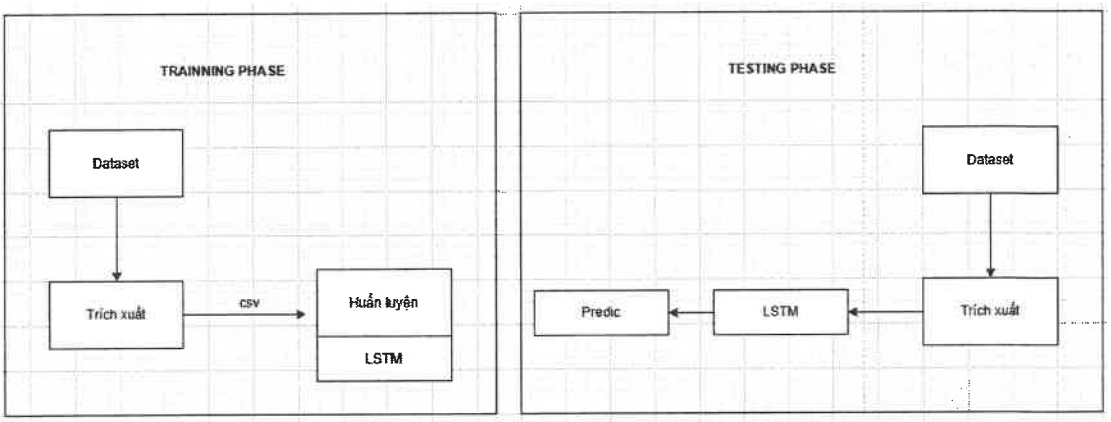
Bảng 2: Bảng số liệu dữ liệu

Category	Count
safe	10000
trojan	23624
worm	15003
backdoor	6385
downloader	814
ransomware	664
dropper	564
informationstealer	364
virus	112
pua	21
p2p-worm	16
cryptominer	14
trojan-gamethief	6
exploit	5
rootkit	3

3.2. Kịch bản thực nghiệm

+ Phát hiện mã độc dựa trên phân tích đường dẫn và sử dụng học sâu. Đồng thời, sử dụng VirusTotal như một công cụ hỗ trợ trong quá trình phân tích và so sánh kết quả. Quá trình xây dựng mô hình bao gồm hai giai đoạn chính:

- Giai đoạn 1: Huấn luyện mô hình (Trainning phase)
- Giai đoạn 2: Kiểm thử mô hình (Testing phase)



Hình 3.1: Quá trình xây dựng mô hình

✧ **Giai đoạn huấn luyện mô hinh (bao gồm 3 bước):**

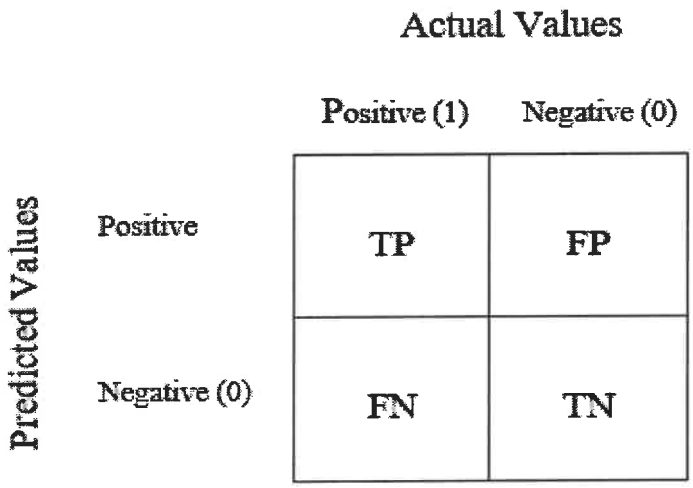
- Bước 1: Bộ dữ liệu chứa các đường dẫn mã độc. Tại bước này, thực hiện tính toán sự xuất hiện của các ký tự quan trọng mới và lưu chúng trong cơ sở dữ liệu.
- Bước 2: Mô-đun không gian vectơ được sử dụng để chuyển đổi dữ liệu chuỗi thành các vectơ. Sử dụng kĩ thuật trích chọn dữ liệu tác giả đã giới thiệu và mô tả cách tính ở chương 2.
- Bước 3: Mô-đun xử lý dữ liệu sử dụng thuật toán học sâu (lần lượt thay thế các thuật toán khác nhau để xác định mô hình tối ưu nhất cho bài toán: LSTM).

✧ **Giai đoạn kiểm thử mô hình:**

- Bước 1: Phần dữ liệu thử nghiệm được tiến hành loại bỏ nhãn dữ liệu.
- Bước 2: Thực hiện quá trình trích xuất đặc trưng dữ liệu tương tự bước 2 ở giai đoạn 1.
- Bước 3: Thử nghiệm các mô hình ứng với các thuật toán học sâu đã được xây dựng ở giai đoạn 1. Tác giả lựa chọn phương pháp đánh giá độ chính xác bằng cách sử dụng ma trận độ đo (confusion matrix) được mô tả như sau:

Confusion Matrix là một phương pháp đánh giá kết quả của những bài toán phân loại với việc xem xét cả những chỉ số về độ chính xác và độ bao quát của các dự đoán cho từng lớp. Một confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại:

- *TP (True Positive):* mẫu mang nhãn dương được phân lớp đúng vào lớp dương
- *TN (True Negative):* mẫu mang nhãn âm được phân lớp đúng vào lớp âm.
- *FP (False Positive - Type 1 Error):* mẫu mang nhãn âm bị phân lớp sai vào lớp dương.
- *FN (False Negative - Type 2 Error):* mẫu mang nhãn dương bị phân lớp sai vào lớp âm.



Hình 3.2: Ma trận độ đo (Conusion matrix)

Ký hiệu TP là True Positive; TN là True Negative; FP là False Positive và FN là False Negative. Thực hiện phép đo Precision – Recall, trong đó, Precision là tỉ lệ số điểm TP trong những điểm được phân loại Positive, còn Recall là tỉ lệ số điểm TP trong số điểm thực sự là Positive. Công thức như sau:

$$\text{precision} = \frac{TP}{TP+FP}$$

$$\text{recall} = \frac{TP}{TP+FN}$$

Ta thấy rằng, Precision và Recall phủ càng cao thì càng tốt. Nhưng trong thực tế, hai giá này không thể đạt được cực đại cùng một lúc và thông thường phải tìm kiếm sự cân bằng. Thước đo $F1_{score}$ là trung bình hài hòa giữa Precision và Recall. Nó có xu hướng bằng không nếu hai giá trị này có xu hướng bằng không.

$$F1_{score} = 2* \frac{precision \times recall}{precision + recall}$$

✧ **Kịch bản thực nghiệm chi tiết:**

Trong mô hình này đã sử dụng bộ dữ liệu bao gồm 57294 bản ghi với các đường dẫn chứa mã độc và có cả chứa mã sạch. Sau khi đã sử dụng học sâu để xử lý bài toán rồi thì đánh giá trên các chỉ số để đánh giá mô hình.

Trong quá trình thử nghiệm thêm một số phương pháp học sâu để so sánh các embedding size khác nhau cho phương pháp đã đề xuất.

Ngoài ra chúng ta còn sử dụng thêm trình duyệt VirusTotal để kiểm tra cho datase đó có chứa mã độc hay mã sạch.

3.3. Tiêu chí đánh giá

+ Sử dụng dataset cụ thể: Cần chọn hoặc xây dựng một dataset chứa các mẫu mã độc và mẫu mã sạch.

+ Sử dụng VirusTotal để xác minh và đối chiếu kết quả từ mô hình học sâu.

Tiêu chí đánh giá dựa trên các chỉ số:

- ✓ Accuracy: Khi dữ liệu cân bằng và tất cả lỗi đều quan trọng
- ✓ Sensitivity (recall): Khi muốn tối đa hoá phát hiện
- ✓ Specificity: Khi cần tránh báo động sai
- ✓ F1-score: Khi dữ liệu cân bằng hoặc cần cân nhắc giữa recall và precision.

Nguồn data: MalwareBazaar và Kaggle.

Link nguồn dữ liệu thực nghiệm: https://bazaar.abuse.ch/export/

https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset

+ Accuracy: tỷ lệ đoán đúng tổng thể - phù hợp vào dữ liệu cân bằng, mọi loại lỗi quan trọng như nhau.

Tỷ lệ dự đoán đúng trên tổng số mẫu = (TP+TN)/(TP+TN+FP+FN). Đo hiệu năng tổng thể.

+ Precion: trong số mẫu nhãn "malicious" bao nhiêu % thực sự đúng, thấp nghĩa là báo động nhầm. Trong số mẫu mô hình gắn nhãn Malicious, bao nhiêu % là đúng thật = TP/(TP+FP). Precision thấp → nhiều báo động nhầm (mã sạch bị coi là mã độc).

+ Recall/Sentitivity: trong số mẫu thực sự là mã độc, trong số mẫu thực sự là Malicious, mô hình bắt được bao nhiêu % = TP/(TP+FN). Recall thấp → bỏ sót mã độc thật (nguy hiểm hơn về mặt an ninh)

+ Specificity: khả năng nhận đúng mẫu

+ F1- score: trung bình điều hòa, dùng khi cần bằng hai chỉ số này- đặc biệt hữu ích với dữ liệu cân bằng.

Trung bình điều hoà giữa Precision và Recall = $2 \cdot (P \cdot R)/(P+R)$. Dùng để đánh giá cân bằng khi dữ liệu mất cân bằng lớp (ở đây Malicious chiếm 82,6%).

	Thực tế Malicious	Thực tế Clean
Dự đoán: Malicious	TP	FP
Dự đon	FN	TN

Giả sử tập test 1000 mẫu , mô hình dự đoán ra TP = 820, TN= 140, FP= 25, FN = 15. Ví dụ cách tính như sau.

accuracy = (820+140) / 1000 = 0.960

precision = 820 / (820+25) = 0.970

recall = 820 / (820+15) = 0.982

f1 = 2×0.970×0.982/(0.970+0.982) = 0.976

Từ việc thực hiện phân chia dữ liệu đầu vào và các embedding size khác nhau với tỷ lệ như trên, ta được bảng kết quả.

Bảng 3: Bảng đánh giá từng chỉ số với LSTM

Stt	LSTM	Accuracy	Precision	Recall	F1-score
1	32	0.8802	0.9045	0.8593	0.7721
2	64	0.9113	0.9360	0.8942	0.9122
3	128	0.9384	0.9445	0.9178	0.9309
4	256	0.9120	0.9212	0.9042	0.9191

Bảng 4: Bảng đánh giá từng chỉ số với MLP

Stt	MLP	Accuracy	Precision	Recall	F1-score
1	32	0.8482	0.8234	0.7905	0.8011
2	64	0.8923	0.8786	0.8590	0.8612
3	128	0.9201	0.9008	0.8899	0.8976
4	256	0.9107	0.8913	0.8745	0.8874

Bảng 5: Bảng đánh giá từng chỉ số với BiLSTM

Stt	BiLSTM	Accuracy	Precision	Recall	F1-score
1	32	0.8734	0.8504	0.8328	0.8464
2	64	0.9217	0.9134	0.8971	0.9013
3	128	0.9575	0.9523	0.9302	0.9106
4	256	0.9418	0.9374	0.9245	0.9281

Kết quả thống kê kết quả đối với bộ dữ liệu thu thập thể hiện ở bảng phía trên. Dựa vào chuỗi giá trị nhập vào chứa đường dẫn 32bit, 64bit, 128bit, 256 bit thì phân tích và đánh giá như sau:

✧ **Với Embedding Size 32:**

Với kích thước nhỏ này, mô hình chỉ có thể học các biểu diễn đơn giản, không đủ khả năng nắm bắt toàn bộ ngữ nghĩa của chuỗi đầu vào.

Kết quả accuracy, precision, recall, và F1-score vẫn đạt mức tốt (~88–90%), nhưng có thể bỏ sót các đặc điểm quan trọng của chuỗi, đặc biệt nếu các đường dẫn độc có chiều dài vượt quá khả năng biểu diễn của embedding size 32.

Embedding size 32 chỉ có thể biểu diễn không gian ngữ nghĩa trong 32 chiều. Nếu một chuỗi đường dẫn đầu vào có 100 ký tự, embedding này phải chia thành nhiều lần (tràn bộ nhớ), dẫn đến thông tin không đầy đủ hoặc sai lệch.

Hạn chế: Không gian ngữ nghĩa nhỏ, dẫn đến giảm khả năng phân loại chính xác cho các chuỗi phức tạp hoặc dài. Mô hình phù hợp hơn với dữ liệu đơn giản và ít biến thể.

✧ **Với Embedding Size 64:**

Với kích thước lớn hơn, mô hình học được nhiều biểu diễn phức tạp hơn và cải thiện khả năng phân loại so với size 32.

Accuracy (~91%) và precision (~93%) tăng lên do không gian embedding mở rộng, cho phép mô hình nắm bắt được các mối quan hệ ngữ nghĩa chi tiết hơn.

Embedding size 64 biểu diễn tốt hơn các chuỗi dài hơn (lên đến 64 bit), giảm hiện tượng mất mát thông tin hoặc tràn bộ nhớ khi xử lý đường dẫn có chiều dài trung bình.

*Ưu điểm:*Độ chính xác và khả năng phân loại vượt trội hơn so với size 32. Phù hợp với dữ liệu phức tạp hơn nhưng vẫn giữ được tốc độ huấn luyện tốt.

Hạn chế: Chưa đạt đến tối ưu hoàn toàn khi so với size 128 (có thể thiếu chi tiết trong các trường hợp rất phức tạp).

✧ **Với Embedding Size 128:**

Đây là kích thước lý tưởng nhất trong nhiều bài toán, bao gồm cả bài toán phát hiện mã độc dựa trên chuỗi đường dẫn.

Accuracy (~93%), precision (~94%), recall (~91%), và F1-score (~93%) đạt mức cao nhất, cho thấy mô hình có khả năng biểu diễn đầy đủ và chính xác nhất các đặc điểm ngữ nghĩa của chuỗi đầu vào.

Embedding size 128 có khả năng xử lý các chuỗi dài phức tạp, gần như không gặp hiện tượng mất mát thông tin.

*Ưu điểm:*Biểu diễn ngữ nghĩa đầy đủ, giảm thiểu tối đa lỗi do tràn hoặc mất thông tin. Đạt hiệu quả tối ưu cả về thời gian huấn luyện và chất lượng phân loại.

*Hạn chế:*Chi phí tính toán cao hơn so với size 32 và 64, nhưng vẫn được xem là cân bằng.

✧ **Với Embedding Size 256**:

Với kích thước lớn nhất, mô hình có khả năng biểu diễn không gian ngữ nghĩa rộng nhất, nhưng điều này không nhất thiết cải thiện hiệu suất.

Accuracy (~91%), precision (~92%), và F1-score (~91%) giảm nhẹ so với size 128. Điều này có thể do hiện tượng overfitting khi kích thước embedding quá lớn so với thông tin cần thiết trong dữ liệu.

Embedding size 256 phù hợp với các bài toán phức tạp hơn, nhưng với bài toán này, không gian biểu diễn lớn trở thành dư thừa.

Ưu điểm: Phù hợp cho dữ liệu cực kỳ phức tạp, chứa thông tin dài và nhiều biến thể.

Hạn chế: Tốn nhiều tài nguyên tính toán. Không gian biểu diễn dư thừa, có thể gây overfitting nếu không có điều chỉnh thích hợp (ví dụ: regularization).

Embedding Size 128 thường mang lại kết quả tốt nhất khi huấn luyện mô hình LSTM. Các tham số như Precision và Recall đạt mức cân bằng, trong khi F1-score

cao thể hiện khả năng phân loại tổng thể tốt. Việc thử nghiệm các giá trị embedding size khác nhau giúp xác định kích thước tối ưu dựa trên dữ liệu cụ thể.

3.4. Kết quả thực nghiệm

Trong phần này học viên trình bày kết quả thực nghiệm dựa trên dữ liệu thu thập được. Các kết quả thực nghiệm bao gồm thống kê và số lượng mã sạch và các loại mã độc trong tập dữ liệu.

Thông tin thống kê

- Tổng số lượng đường dẫn trong dữ liệu là : 57596
- Đường dẫn mã độc: 47596

Nhận xét:

- ✓ Loại mã độc phổ biến nhất trong dữ liệu là Trojan (23,624), theo sau là Worm (15,003).
- ✓ Các loại mã độc khác như CryptoMiner, Exploit, và Rootkit có tần suất xuất hiện rất thấp.
- ✓ Tỉ lệ đường dẫn mã độc chiếm khoảng 82.6% so với tỉ lệ mã sạch.

3.5. Kết luận chương

Trong chương 3 này luận văn đã xây dựng kịch bản thử nghiệm phân tích mã độc dựa trên phân tích đường dẫn. Với mỗi kịch bản đã xây dựng được mô hình học sâu như LSTM.

Sử dụng VirusTotal để đối chiếu và đánh giá kết quả. Việc tích hợp VirusTotal giúp đối chiếu kết quả phân tích từ mô hình với các cơ sở dữ liệu bảo mật hàng đầu, đảm bảo độ tin cậy và chính xác trong phát hiện mã độc. Điều này cũng giúp đánh giá được điểm mạnh và hạn chế của mô hình học sâu so với các công cụ bảo mật truyền thống.

Các kết quả thử nghiệm ban đầu cho thấy giải pháp phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu đề xuất có tính khả thi cao và phù hợp với các yêu cầu đề ra.

KẾT LUẬN

1. Những đóng góp của luận văn

Với mục đích nghiên cứu phương pháp phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu. Luận văn đã đi sâu nghiên cứu các vấn đề xung quanh đề tài nghiên cứu, các thuật toán học sâu để phân tích và phát hiện đường dẫn bất thường.

Những kết quả chính đã đạt được trong luận văn:

- Tìm hiểu về các các loại mã độc và hoạt động của mã độc từ đó đưa ra phương pháp học sâu phù hợp nhất và giải quyết bài toán tốt nhất.
- Nghiên cứu các mô hình học sâu và làm rõ ưu nhược điểm để lựa chọn phương pháp tối ưu nhất cho phân tích đường dẫn.
- Tìm hiểu phương pháp phát hiện đường dẫn độc sử dụng học sâu
- Lựa chọn và ứng dụng thuật toán học sâu dựa trên phân tích đường dẫn sử dụng học sâu LSTM, MLP, BiLSTM.
- Thực nghiệm xây dựng mô hình với bộ data mã độc/ mã sạch theo từng kịch bản để đưa ra mô hình tốt nhất với embedding là 32 bit, 64 bit, 128 bit, 256 bit.

2. Hướng phát triển của luận văn

Một số hướng phát triển tiếp theo của luận văn:

- Mặc dù thuật toán đề xuất cải thiện độ chính xác phân loại của các liên kết bất hợp pháp, nhưng với sự gia tăng số lượng tham số có trong các yêu cầu, độ chính xác phân loại sẽ giảm. Do đó, cần tìm sự kết hợp của các phương pháp phát hiện bất thường dựa trên học sâu nhằm cải thiện độ chính xác phân loại không chỉ của mã độc lạ, bất thường mới có đặc trưng các từng thiết bị.
- Thực hiện nghiên cứu phương pháp có thể xuất bản một phần mềm/ thương mại hoá, cảnh báo độc hại để đính kèm trên thiết luôn, có thể giống như các hãng công nghệ mới nhất hiện nay dùng cho phát hiện mã độc.

DANH MỤC TÀI LIỆU THAM KHẢO

Tiếng Việt

[1]. Vũ Hữu Tiệp (2016-2020) – "Machine Learning cơ bản"

[2]. Hoàng Xuân Dâu (2017) - "An toàn thông tin "

[3]. Nguyễn Thanh Tuấn (2019) - Sách Deep learning cơ bản

[4]. Từ Minh Phương (2014) - "Nhập môn trí tuệ nhân tạo"

Tiếng Anh

[1].Symantec. "Symantec Internet Security Threat Report". 2022.

[2].McAfee. "Automated Malware Signature Detection Develop a System for Signatures to detect malware".IJMTST. Jan.2024.

[3].Kaspersky Lab. "Modern Malware Threats and Detection"

[4].Kolbitsch, C., Comparetti, P. M., Kruegel, C., Zhou, X., & Wang, X. "Effective and efficient malware detection at the end host". In Proceedings of the 18th conference on USENIX security symposium. 2009.

[5].Christodorescu, M., & Jha, S.. "Static analysis of executables to detect malicious patterns". In Proceedings of the 12th USENIX Security Symposium. 2004.

[6].Saxe, J., & Berlin, K.. "Deep neural network based malware detection using two dimensional binary program features". IEEE. 2015.

[7].Hardy, W., Chen, L., Hou, S., Ye, Y., & Li, X.. "A deep learning framework for intelligent malware detection". In Proceedings of the International Conference on Data Mining Workshops. IEEE. 2016.

PHỤ LỤC

1. Sử dụng thuật toán LSTM cho mô hình học sâu

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

# 1. Read the data
file_path = r'C:\CTDL&GT\CodePycharm\data\bodmas_malware_category.csv'  #
Path to the CSV file
data = pd.read_csv(file_path)

# 2. Prepare the data
# Separate the path column and the label column
X = data['sha256']
y = data['category']

# Encode the labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# 3. Text preprocessing
# Tokenizer to convert text to numbers
tokenizer = Tokenizer(num_words=5000)  # Use the top 5000 most frequent
words
tokenizer.fit_on_texts(X_train)

# Convert text into sequences of numbers
X_train_seq = tokenizer.texts_to_sequences(X_train)
X_test_seq = tokenizer.texts_to_sequences(X_test)

# Pad sequences to ensure fixed length
max_length = 100  # Maximum length of sequences
```

```
X_train_pad = pad_sequences(X_train_seq, maxlen=max_length,
padding='post')
X_test_pad = pad_sequences(X_test_seq, maxlen=max_length, padding='post')

# 4. Build the model
model = Sequential([
    Embedding(input_dim=5000, output_dim=128, input_length=max_length),
# Embedding layer
    LSTM(128, return_sequences=False),  # LSTM layer
    Dropout(0.3),  # Dropout to reduce overfitting
    Dense(64, activation='relu'),  # Fully connected layer
    Dropout(0.3),
    Dense(1, activation='sigmoid')  # Output layer with sigmoid (binary
classification)
])

# 5. Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['accuracy'])

# 6. Train the model
history = model.fit(
    X_train_pad, y_train,
    validation_data=(X_test_pad, y_test),
    epochs=10,
    batch_size=32,
    verbose=1
)

# 7. Evaluate the model
loss, accuracy = model.evaluate(X_test_pad, y_test)

# 8. Save the model
model.save('malicious_path_detection_model.h5')
```

2. Training model

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing.sequence import pad_sequences
import pandas as pd
import pickle
import os
```

```
# 1. Load the saved model
try:
    model = load_model('malicious_path_detection_model.h5')
    print("The model has been successfully loaded.")
except Exception as e:
    print(f"Error loading the model: {e}")
    exit()

# 2. Load the Tokenizer from the saved file
from tensorflow.keras.preprocessing.text import Tokenizer
tokenizer = Tokenizer()
with open('tokenizer.pickle', 'wb') as handle:
    pickle.dump(tokenizer, handle, protocol=pickle.HIGHEST_PROTOCOL)

try:
    with open('tokenizer.pickle', 'rb') as handle:
        tokenizer = pickle.load(handle)
    print("Tokenizer has been successfully loaded.")
except Exception as e:
    print(f"Error loading the Tokenizer: {e}")
    exit()

# 3. Function to prepare input data
def prepare_data(data, tokenizer, max_length=100):
    """
    Convert text data into numerical sequences and add padding.
    """
    if data.isnull().any():
        print("Warning: Data contains missing values (NaN).")
        data = data.fillna("")  # Handle missing values

    sequences = tokenizer.texts_to_sequences(data)
    padded_sequences = pad_sequences(sequences, maxlen=max_length,
padding='post')
    return padded_sequences

# 4. Read the data for prediction
try:
    # Ensure the CSV file exists
    csv_file = r'C:\CTDL&GT\CodePycharm\data\bodmas_metadata_result.csv'
    if not os.path.exists(csv_file):
        raise FileNotFoundError(f"The file {csv_file} does not exist.")
```

```
    data = pd.read_csv(csv_file)
    print("Data has been successfully loaded.")

    # Check if the 'sha' column exists
    if 'sha' not in data.columns:
        raise KeyError("The 'sha' column does not exist in the data.")
except Exception as e:
    print(f"Error loading or checking the data: {e}")
    exit()

# 5. Prepare the input data
X_new = prepare_data(data['sha'], tokenizer)

# 6. Perform predictions
try:
    predictions = model.predict(X_new, verbose=1)
    data['predicted_label'] = (predictions > 0.5).astype(int)
    data['predicted_label'] = data['predicted_label'].replace({1:
"Malicious", 0: "Safe"})
except Exception as e:
    print(f"Error during prediction: {e}")
    exit()

# 7. Display the results
print(data[['sha', 'predicted_label']])

# 8. Save the prediction results to a new CSV file
output_file = 'predicted_results.csv'
data.to_csv(output_file, index=False)
print(f"The prediction results have been saved to the file:
{output_file}")
```

BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án qua phần mềm *Tailieu.vn* một cách trung thực và đạt kết quả mức độ tương đồng 17% toàn bộ nội dung đề án. Bản đề án kiểm tra qua phần mềm là bản cứng đề án 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.

Hà Nội, ngày 29 *tháng* 7 *năm 2026*

HỌC VIÊN CAO HỌC



Kết quả kiểm tra trùng lặp #68037311 KiemTraTaiLieu

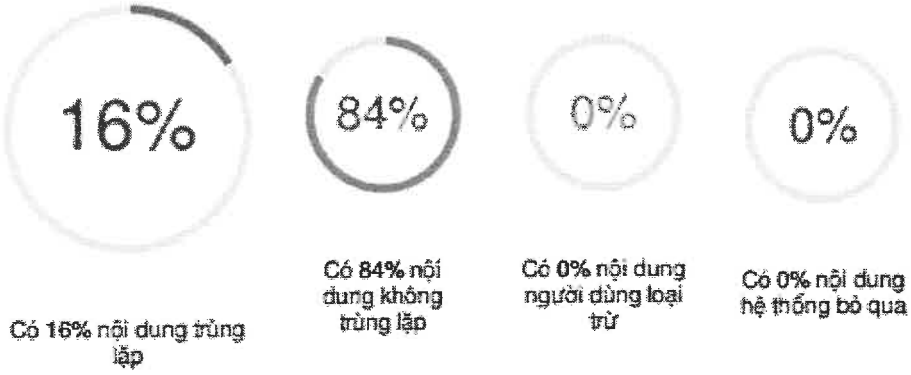
KiểmTraTàiLiệu

BÁO CÁO KIỂM TRA TRÙNG LẶP

Thông tin tài liệu

Tên tài liệu:	Nguyễn Quang Tuấn _B23CHCS023B
Tác giả:	Nguyen Quang Tuan
Điểm trùng lặp:	16
Thời gian tải lên:	06:23 10/07/2026
Thời gian sinh báo cáo:	06:27 10/07/2026
Các trang kiểm tra:	76/76 trang

Kết quả kiểm tra trùng lặp



Có 16% nội dung trùng lặp

Có 84% nội dung không trùng lặp

Có 0% nội dung người dùng loại trừ

Có 0% nội dung hệ thống bỏ qua

Nguồn trùng lặp tiêu biểu

tailieu.vn 123docz.net www.uet.vnu.edu.vn

GIẢNG VIÊN HƯỚNG DẪN **HỌC VIÊN**

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BÁO CÁO GIẢI TRÌNH
SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP

Họ và tên học viên: Nguyễn Quang Tuấn
Chuyên ngành: Khoa học máy tính
Khóa: 2023 đợt 1

Tên đề tài: Phương pháp nghiên cứu phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu

Người hướng dẫn khoa học: PGS.TS. Trần Quang Anh.

Ngày bảo vệ: 22/07/2025

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

TT	Ý kiến hội đồng	Sửa chữa của học viên
1	Chỉnh sửa lỗi soạn thảo, lỗi ngữ pháp, chính tả	Học viên đã rà soát, chỉnh sửa các lỗi soạn thảo, các lỗi ngữ pháp
2	Bỏ các từ mục tiếng anh	Tiếp thu góp ý của Hội đồng, tác giả đã bỏ từ mục tiếng anh tại chương 1
3	Bổ sung mô tả nguồn dữ liệu thực nghiệm	Tiếp thu góp ý của Hội đồng, tác giả đã bổ sung nguồn dữ liệu thực nghiệm tại tại mục 3.3 trong chương 3

Hà Nội, ngày 29 tháng 7 năm 2026

Ký xác nhận của

CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN	THƯ KÝ HỘI ĐỒNG	NGƯỜI HƯỚNG DẪN KHOA HỌC	HỌC VIÊN
PGS.TS.Nguyễn Trọng Khánh	TS. Đặng Ngọc Hùng	PGS.TS. Trần Quang Anh	Nguyễn Quang Tuấn

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BIÊN BẢN
HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 1949/QĐ-HV ngày 07 tháng 07 năm 2026 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi 11 giờ 00 phút, ngày 22 tháng 07 năm 2026 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Nguyễn Quang Tuấn**

Tên đề án tốt nghiệp: **Phương pháp nghiên cứu phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu**

Chuyên ngành: **Khoa học máy tính** Mã số: **8480101**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: 05 / 05

TT	HỌ VÀ TÊN	TRÁCH NHIỆM TRONG HĐ	GHI CHÚ
1	**PGS.TS. Nguyễn Trọng Khánh**	Chủ tịch	
2	**TS. Đặng Ngọc Hùng**	Thư ký	
3	**TS. Vũ Văn Thỏa**	Phản biện 1	
4	**TS. Trịnh Bảo Ngọc**	Phản biện 2	
5	**TS. Nguyễn Văn Vinh**	Uỷ viên	

Các nội dung thực hiện:

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên. (có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

1/ Giải thích lý do lựa chọn mô hình học sâu được áp dụng trong đề án

2/ Làm rõ nguồn gốc và thông tin của bộ dữ liệu dùng để train và test

3/ Tại sao lựa chọn LSTM do tác giả phân tích đường dẫn

7. Trả lời của học viên:

1/ Học viên có trả lời được các câu hỏi và lý do lựa chọn CNN, RNN, LSTM

2/ Học viên có nêu được và giải thích nguồn gốc của bộ dữ liệu.

3/ Học viên chưa thực sự hiểu và trả lời được lý do lựa chọn LSTM.

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo).*

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: PGS.TS. Nguyễn Trang Khánh
  2. Uỷ viên Ban kiểm phiếu: TS. Vũ Văn Thoả
  3. Uỷ viên Ban kiểm phiếu: TS. Đặng Ngọc Hưng
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 7,2

Kết luận:

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:

- Rà soát các sai sót, lỗi chính tả, in ấn theo nhận xét của Hội đồng.
- Bổ sung mã và nguồn dữ liệu thực nghiệm.
- Thay chỉnh bố cục để sử dụng phù hợp với tên đề án và góp ý của phản biện.

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên: đề nghị công nhận

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS.....Không.....

Buổi làm việc kết thúc vào..... cùng ngày.

Chủ tịch	Thư ký
Nguyễn Trang Khánh	Đặng Ngọc Hưng

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Tên đề tài đề án:	Phương pháp nghiên cứu phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu
Chuyên ngành:	Khoa học máy tính
Mã số:	8.48.01.01
Tên học viên:	Nguyễn Quang Tuấn
Họ tên người nhận xét:	Vũ Văn Thỏa
Học hàm, học vị:	Tiến sĩ **Chuyên ngành**: Toán học tính toán
Cơ quan công tác:	Học viện Công nghệ Bưu chính Viễn thông

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài

Hiện nay, cùng với sự bùng nổ của các dịch vụ được triển khai trên nền tảng Internet, vấn đề an ninh mạng diễn biến phức tạp khi mã độc (malware) liên tục gia tăng về số lượng và biến thể tinh vi. Một trong các hướng nghiên cứu phát hiện mã độc đang được quan tâm là phân tích đường dẫn tệp tin (path file) nhằm cung cấp dấu vết chuyên sâu về cách thức mã độc lây lan, ẩn náu và tương tác với hệ thống qua các chuỗi ký tự bất thường.

Đề án với đề tài "Phương pháp nghiên cứu phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu" của học viên Nguyễn Quang Huy với mục tiêu nghiên cứu phương pháp phát hiện mã độc dựa trên phân tích các bất thường của đường dẫn tập tin (Path File) sử dụng các mô hình mạng MLP, mạng LSTM và mạng BiLSTM. Vì vậy, đề tài đề án có tính cấp thiết, có ý nghĩa khoa học và thực tiễn.

II/ Về nội dung chất lượng của đề án, những kết quả đạt được

Đề án được trình bày trong 62 trang bao gồm phần mở đầu, 3 chương nội dung, phần kết luận và tài liệu tham khảo.

Chương 1 của đề án gồm 17 trang trình bày tổng quan về mã độc dựa trên tiêu chuẩn NIST bao gồm: khái niệm, phân loại các loại malware như Virus, Worm, Trojan, Spyware, Rootkit..., cách thức lây nhiễm, dấu hiệu nhận biết và biện pháp phòng chống. Đồng thời, chương này giới thiệu về cấu trúc đường dẫn hệ thống (tuyệt đối, tương đối) và định nghĩa thế nào là một "đường dẫn độc hại" và một số dấu hiệu nhận biết chúng như: độ dài > 260 ký tự, chứa ký tự đặc biệt, ngụy trang tiến trình hệ thống (ví dụ: csrss), đuôi tệp mập mờ, ... Tuy nhiên, một số nội dung trong chương 1 cần trình bày hợp lý hơn. Chẳng hạn, mục 1.2 nên hiệu chỉnh tên thành 1.2 Tổng quan về đường dẫn và đường dẫn độc. Cần nêu rõ bài toán cần giải

quyết trong đề án: bài toán phát hiện đường dẫn độc là một trường hợp đặc biệt của bài toán phân loại dữ liệu. Sau đó khảo sát tổng quan một số phương pháp giải quyết bài toán đặt ra.

Nội dung chính của chương 2 được trình bày trong 31 trang là nghiên cứu nền tảng lý thuyết cho toàn bộ đề án, bao gồm: Tổng quan về Học máy (Phân loại học có giám sát, không giám sát, bán giám sát, tăng cường); Các thuật toán phân loại kinh điển (Decision Tree - CLS, ID3, C4.5, Random Forest, Logistic Regression, SVM); Các phương pháp nhúng trích xuất văn bản (Bag of Words, TF-IDF, Word2Vec, GloVe, FastText) và một số thuật toán học sâu bao gồm mạng MLP, mạng CNN, mạng LSTM và mạng BiLSTM. Tuy nhiên, đối chiếu với mục tiêu của đề án, nội dung của chương 2 nên được hiệu chỉnh như sau:

- Tiêu đề chương 2 nên hiệu chỉnh thành: Chương 2: Nghiên cứu một số mô hình phát hiện đường dẫn độc.

- Phần trình bày tổng quan về học máy nên lược bớt nội dung vì các thuật toán học máy truyền thống không phải là chủ đề chính của đề án.

- Phần trình bày về học sâu cần trình bày chi tiết hơn để tăng tính học thuật của đề án. Có thể trình bày theo cấu trúc: mạng MLP, mạng CNN, mạng LSTM và mạng BiLSTM. Nên bổ sung các phân tích so sánh đánh giá các mô hình này.

Chương 3 của đề án gồm 7 trang trình bày thực nghiệm ba mô hình MLP, LSTM, BiLSTM trên tập dữ liệu đã lựa chọn theo một số kịch bản và đánh giá kết quả. Tuy nhiên, đề án cần hiệu chỉnh nội dung các tiêu chí đánh giá phù hợp hơn. Tập dữ liệu thử nghiệm cần nêu rõ nguồn gốc xuất xứ và đánh giá sự phù hợp khi lựa chọn để tiến hành thực nghiệm. Các kết quả thử nghiệm cần được phân tích đánh giá chi tiết hơn. Các bảng trình bày trong chương 3 cần đánh số thứ tự kèm số thứ tự của chương, chẳng hạn bảng 3.1, 3.2, ….

Đề án được chia thành các chương mục rõ ràng và cơ bản bám sát các nội dung theo đề cương đã được phê duyệt. Tuy nhiên, học viên cần hiệu chỉnh lại bố cục của chương 2 theo các góp ý trên để đảm bảo tính lô gic trong trình bày. Phần mở đầu cần hiệu chỉnh phù hợp hơn theo quy định của Học viện.

Trong đề án còn nhiều lỗi in ấn cần phải sửa chữa tại các trang 6, 12, 25, 26, 49, 54, 60….. Một số hình còn mờ nên hiệu chỉnh hoặc in màu. Số thứ tự của một số hình trong chương 2 cần hiệu chỉnh phù hợp.

III/ Những vấn đề cần giải thích thêm

(1) Giải thích rõ hơn lý do chọn các mô hình học sâu gồm các mạng MLP, LSTM và BiLSTM để giải quyết bài toán trong đề án?

(2) Giải thích cụ thể hơn các bộ dữ liệu thử nghiệm sử dụng trong đề án bao gồm các khía cạnh: nguồn gốc và các tiền xử lý cần thiết?

(3) Giải thích thêm về ý nghĩa của các tiêu chí đánh giá và kết quả thử nghiệm trình bày tại mục 3,3 của đề án?

IV/ Kết luận

Tôi đồng ý cho học viên Nguyễn Quang Tuấn được bảo vệ đề án với đề tài "Phương pháp nghiên cứu phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu" trước Hội đồng chấm đề án tốt nghiệp thạc sĩ của Học viện. Tuy nhiên, để đảm bảo chất lượng đề án tốt

nghiệp thạc sĩ chuyên ngành Khoa học máy tính, học viên cần chỉnh sửa đề án và giải trình rõ trước Hội đồng các vấn đề được góp ý trên đây.

Ngày 20 tháng 07 năm 2026

Người nhận xét

TS Vũ Văn Thỏa

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

(Dùng cho người phản biện)

Tên đề tài đề án tốt nghiệp: Phương pháp nghiên cứu phát hiện mã độc dựa trên phân tích đường dẫn sử dụng học sâu

Chuyên ngành: Hệ thống thông tin
Mã chuyên ngành: 8.48.01.04
Họ và tên học viên: Nguyễn Quang Tuấn
Họ và tên người nhận xét: Trịnh Bảo Ngọc
Học hàm, học vị: Tiến sĩ
Chuyên ngành: Kỹ thuật phần mềm
Cơ quan công tác: Trường đại học Hà Nội
Số điện thoại: 0987 355 799 E-mail: ngoctb@hanu.edu.vn

NỘI DUNG NHẬN XÉT

I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:

Cơ sở khoa học: Đề tài được xây dựng trên nền tảng khoa học vững chắc của lĩnh vực an ninh mạng, học máy và học sâu. Tác giả đã phân tích các khái niệm cơ bản về mã độc, phân loại, cách thức lây nhiễm và các phương pháp phát hiện truyền thống. Tiếp đó, đề tài đi sâu vào các thuật toán học máy như Decision Tree, Random Forest, SVM, Logistic Regression và các mô hình học sâu như MLP, CNN, LSTM, BiLSTM. Việc kết hợp phân tích đường dẫn với học sâu cho thấy cơ sở khoa học rõ ràng, phù hợp với xu hướng nghiên cứu hiện đại trong phát hiện mã độc.

Cơ sở thực tiễn: Đề tài xuất phát từ thực trạng mã độc ngày càng tinh vi, vượt qua các phương pháp phát hiện truyền thống dựa trên chữ ký hoặc hành vi. Phân tích đường dẫn file trong hệ điều hành Windows là hướng tiếp cận thực tiễn, vì đây là môi trường phổ biến và mục tiêu chính của nhiều loại mã độc. Tác giả đã xây dựng mô hình thử nghiệm với dữ liệu thực tế, áp dụng các thuật toán học sâu để phát hiện bất thường trong đường dẫn. Kết quả thực nghiệm cho thấy khả năng ứng dụng cao, giúp nâng cao hiệu quả bảo mật và giảm tỷ lệ báo động sai trong hệ thống.

Tính cấp thiết: Trong bối cảnh an ninh mạng ngày càng phức tạp, mã độc là mối đe dọa lớn đối với cá nhân và tổ chức. Các phương pháp truyền thống không còn đủ mạnh để phát hiện biến thể mới. Do đó, việc nghiên cứu phát hiện mã độc dựa trên phân tích đường dẫn bằng học sâu là cấp thiết. Nó không chỉ giúp phát hiện sớm các hành vi bất thường mà còn mở ra khả năng tự động hóa trong bảo mật. Đây là nhu cầu thực tế của các hệ thống thông tin hiện đại, đặc biệt trong bối cảnh số lượng mã độc ngày càng tăng và kỹ thuật tấn công ngày càng tinh vi.

II/ Nội dung của đề án tốt nghiệp, các kết quả đã đạt được:

Kết cấu đề án:

Đề án gồm ba chương được trình bày trong 77 trang: tổng quan về mã độc và đường dẫn, nghiên cứu các thuật toán học máy và học sâu trong phát hiện mã độc, thực nghiệm và đánh giá kết quả. Cuối cùng là phần kết luận và hướng phát triển, đảm bảo cấu trúc logic, đầy đủ và khoa học.

Tóm tắt các kết quả đạt được:

Đề tài đã đạt được nhiều kết quả quan trọng. Về lý luận, tác giả đã hệ thống hóa kiến thức nền tảng về mã độc, phân loại, cách thức lây nhiễm và các phương pháp phát hiện. Về kỹ thuật, đề tài đã phân tích và áp dụng nhiều thuật toán học máy và học sâu, bao gồm Decision Tree, Random Forest, Logistic Regression, SVM, MLP, CNN, LSTM và BiLSTM. Về thực nghiệm, tác giả đã xây dựng bộ dữ liệu đường dẫn, thiết kế kịch bản kiểm thử và đánh giá hiệu quả mô hình bằng các chỉ số như độ chính xác, độ nhạy, độ đặc hiệu và F1-score.

Kết quả cho thấy các mô hình học sâu, đặc biệt là LSTM và BiLSTM, có khả năng phát hiện mã độc với độ chính xác cao hơn so với các phương pháp truyền thống. Đề tài cũng chỉ ra ưu điểm của việc phân tích đường dẫn trong phát hiện bất thường, đồng thời nêu rõ hạn chế như yêu cầu dữ liệu lớn và chi phí tính toán cao. Đây là đóng góp thiết thực cho lĩnh vực an ninh mạng.

III/ Những vấn đề cần giải thích thêm:

Về hình thức trình bày:

Đề án trình bày rõ ràng, có mục lục chi tiết, danh mục thuật ngữ và hình vẽ đầy đủ, giúp người đọc dễ theo dõi và nắm bắt nội dung.

Một số lỗi định dạng xuất hiện trong tài liệu. Trang 5, bảng "Danh sách thuật ngữ viết tắt" có lỗi sai khi mục LSTM được ghi là "Convolutional Neural Networks" thay vì "Long Short-Term Memory", gây nhầm lẫn.

Trang 6, danh sách hình ảnh có lỗi đánh số không đồng bộ, ví dụ "Hình 3.1" và "Hình 3.2" bị ghi lệch thứ tự.

Trang 7, danh sách bảng biểu có lỗi căn chỉnh khi số trang không thẳng hàng. Trang 3, phần mục lục có lỗi trình bày khi các mục bị tách dòng không hợp lý, ví dụ "2.1.2.2. Học không giám sát" xuống dòng riêng biệt gây khó đọc.

Ngoài ra, một số bảng có phần mô tả dài gây mất cân đối giữa các cột, ví dụ bảng 2 ở trang 62. Các lỗi này tuy không ảnh hưởng lớn đến nội dung nhưng làm giảm tính chuyên nghiệp của hình thức trình bày.

Về nội dung:

Một số vấn đề về kiến thức và logic cần lưu ý. Trang 5, bảng thuật ngữ viết tắt có lỗi sai nghiêm trọng khi định nghĩa LSTM thành "Convolutional Neural Networks", gây nhầm lẫn về khái niệm.
Trang 14, phần phân loại mã độc chưa đề cập đến ransomware, trong khi đây là loại mã độc phổ biến hiện nay.
Trang 30, phần tổng quan về học máy chưa phân tích rõ sự khác biệt giữa học có giám sát và học bán giám sát, chỉ dừng ở mức liệt kê.
Trang 48, phần trích xuất đặc trưng văn bản chưa giải thích chi tiết cách áp dụng TF-IDF và Word2Vec trong phân tích đường dẫn, khiến người đọc khó hình dung. Trang 53, phần mô tả MLP chưa phân tích rõ ưu nhược điểm so với CNN trong bối cảnh dữ liệu đường dẫn.
Trang 57, phần mô tả LSTM chưa giải thích rõ cơ chế ghi nhớ chuỗi dài, chỉ dừng ở mức khái quát.
Trang 65, bảng đánh giá chỉ số với LSTM, MLP và BiLSTM chưa có so sánh định lượng cụ thể giữa các mô hình, chỉ liệt kê kết quả riêng lẻ.

Câu hỏi:

1. Vì sao LSTM phù hợp hơn CNN trong phân tích đường dẫn?
2. Cơ chế hoạt động của BiLSTM khác gì so với LSTM thường?
3. Ưu nhược điểm của Random Forest trong phân loại dữ liệu mã độc?

IV/ Kết luận:
Đồng ý.

Ngày 21 tháng 07 năm 2026
NGƯỜI NHẬN XÉT

TS Trịnh Bảo Ngọc