HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Trần Huỳnh Tiến

ỨNG DỤNG REPRESENTATION LEARNING PHÁT HIỆN TẤN CÔNG PHISHING

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

TP. HỒ CHÍ MINH – 2023

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Trần Huỳnh Tiến

ỨNG DỤNG REPRESENTATION LEARNING PHÁT HIỆN TẤN CÔNG PHISHING

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC

TS. NGUYỄN HỒNG SƠN

TP. HỒ CHÍ MINH – 2023

LỜI CAM ĐOAN

Tôi cam đoan rằng luận văn: *"Ứng dụng Representation Learning phát hiện tấn công Phishing"* là công trình nghiên cứu của chính tôi.

Tôi cam đoan các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Không có sản phẩm/nghiên cứu nào của người khác được sử dụng trong luận văn này mà không được trích dẫn theo đúng quy định.

TP. Hồ Chí Minh, ngày 28 tháng 02 năm 2023

Học viên thực hiện luận văn

Trần Huỳnh Tiến

LỜI CẢM ƠN

Trong suốt quá trình học tập và nghiên cứu thực hiện luận văn, ngoài nỗ lực của bản thân, tôi đã nhận được sự hướng dẫn nhiệt tình quý báu của quý Thầy Cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp. Với lòng kính trọng và biết ơn sâu sắc, tôi xin gửi lời cảm ơn chân thành tới:

Ban Giám hiệu , Phòng đào tạo sau đại học và quý Thầy Cô đã tạo mọi điều kiện thuận lợi giúp tôi hoàn thành luận văn.

Tôi xin chân thành cảm ơn Thầy **TS. Nguyễn Hồng Sơn**, người Thầy kính yêu đã hết lòng giúp đỡ, hướng dẫn, động viên, tạo điều kiện cho tôi trong suốt quá trình thực hiện và hoàn thành luận văn.

Tôi xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành luận văn. Mặc dù đã có nhiều cố gắng, nỗ lực, nhưng do thời gian và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý của quý Thầy Cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn.

Xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 28 tháng 02 năm 2023

Học viên thực hiện luận văn

Trần Huỳnh Tiến

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
DNS	Domain Name System	
RL	Representation learning	Học biểu diễn / học đại diện
DOM	Document Object Model	
TFIDF	Term Frequency Inverse Document Frequency	
SVD	Singular value decomposition	
NMF	Non- negative Matrix Factorization	
RF	Random forest	Rừng ngẫu nhiên
SVM	Support vector machine	Máy vectơ hỗ trợ
DT	Decision forest	Rừng quyết định
PCA	Principal component analysis	Phép phân tích thành phần chính
k-NN	K-nearest neighbor	k hàng xóm gần nhất
CNN	Convolutional neural network	Mạng thần kinh tích chập
LSTM	Long short-term memory	Bộ nhớ dài-ngắn hạn

MỤC LỤC

MỞ ĐẦU

1. Lý do chọn đề tài

Thông tin là một tài sản vô cùng quý giá của chính phủ, các tổ chức, doanh nghiệp hay bất cứ cá nhân nào. Ai có được thông tin là có thể đạt được tỉ lệ thành công cao. Chính vì vậy việc trao đổi và giữ bí mật thông tin là một vấn đề rất quan trọng. Ngày nay công nghệ thông tin bùng nổ với sự phát triển của Internet và mạng máy tính đã giúp cho việc trao đổi thông tin và các giao dịch một các dễ dàng hơn. Tuy nhiên lại phát sinh những vấn đề mới đó là tội phạm mạng máy tính đã phát sinh, tồn tại và nhanh chóng phát triển không ngừng. Thông tin quan trong được nằm trong kho dữ liệu hoặc nằm trên đường truyền có thể bị trộm cắp, có thể làm sai lệch, có thể bị giả mạo. Những bí mật kinh doanh, tài chính là mục tiêu của đối thủ cạnh tranh. Những tin tức về an ninh quốc gia là mục tiêu tình báo trong và ngoài nước. Những vấn đề xoay quanh tội phạm mạng máy tính thường bao gồm các hoạt động bất hợp pháp như: chiếm dụng và sử dụng trái phép tài nguyên máy tính, vi phạm bản quyền, các chương trình giám sát bất hợp pháp, ở những mức độ trầm trọng hơn, các hoạt động tội phạm công nghệ cao còn nhắm đến việc phá hoại các hệ thống máy tính bằng cách phát tán các mã độc, ăn cắp các thông tin về thẻ tín dụng, tài khoản ngân hàng của nạn nhân, lấy cắp các thông tin tình báo, bí mật quốc gia. Điều đó có thể ảnh hưởng các tổ chức, công ty hay cả một quốc gia.

Tấn công lừa đảo (Phishing) [1] là hình thức tấn công phi kỹ thuật được tội phạm mạng sử dụng nhiều nhằm đánh cắp dữ liệu bí mật từ máy tính hay một mạng máy tính của người dùng, sau đó sử dụng dữ liệu cho nhiều mục đích khác nhau, như lấy cắp tiền của nạn nhân hoặc bán lại dữ liệu đã đánh cắp.

Thông thường tin tặc giả mạo thành ngân hàng, trang website giao dịch trực tuyến, ví điện tử, các công ty có tín dụng để lừa đảo người dùng chia sẽ thông tin như: tài khoản và mật khẩu đăng nhập, mật khẩu giao dịch, thẻ tín dụng và các thông tin quan trọng khác. Phương thức tấn công này được tin tặc thực hiện thông qua thư điện tử, tin nhắn văn bản hoặc mạng xã hội, khi người dùng thông qua

đường link giả mạo sẽ được yêu cầu đăng nhập. Nếu người sử dụng truy cập vào thông tin giả mạo đó tin tặc sẽ có được thông tin ngay tức thì.

Hiện nay trên thế giới có nhiều tổ chức, cá nhân đã phát triển các phần mềm phòng chống và tấn công Phishing như: Google hay Microsoft đều có bộ lọc Email spam/Phishing để bảo vệ người dùng. Ngoài ra, còn có Anti-Phishing Domain Advisor: bản chất là một thanh công cụ giúp cảnh báo những trang website lừa đảo, dựa theo dữ liệu của công ty Panda Security. Netcraft Anti-Phishing Extension: Netcraft là một đơn vị uy tín cung cấp các dịch vụ bảo mật bao gồm nhiều dịch vụ. Trong số đó, tiện ích mở rộng chống tấn công Phishing của Netcraft được đánh giá khá cao với nhiều tính năng cảnh báo thông minh.

Tuy nhiên đa số phần mềm này là sản phẩm thương mại, nếu là bản miễn phí thì bị giới hạn tính năng, khó nâng cấp, bảo trì. Vì thế cần phải có một giải pháp để phòng chống tấn công Phishing trên Internet và mạng máy tính.

Phát hiện các tấn công Phishing trở thành bài toán quan trọng trong an toàn thông tin. Phát hiện được tấn công Phishing là việc làm khó khăn, mặc dù có các giải pháp được công bố nhưng vẫn cần độ chính xác cao. Sự phát triển của trí tuệ nhân tạo, máy học trong những năm gần đây rất có tiềm năng áp dụng để phát hiện tấn công Phishing với độ chính xác cao. Trong đó mô hình dựa vào máy học có thể phát huy nhiều ưu điểm cho bài toán này. Xuất phát từ thực tế đó đề cương luận văn tập trung nghiên cứu:

“Ứng dụng representation learning phát hiện tấn công Phishing”

2. Tổng quan về vấn đề nghiên cứu

Trong nghiên cứu này sẽ đề cập về cách tấn công Phishing và phương pháp phòng chống tấn công Phishing bằng nhiều hình thức. Hiện nay nhiều hình thức để thực hiện một vụ tấn công Phishing. Để thực hiện ý tưởng đề ra cần nghiên cứu và tiến hành triển khai các nội dung sau: tìm hiểu mạng máy tính, các phần mềm dùng chung phổ biến như các phần mềm của tỉnh Tây Ninh đề ra giải pháp hợp lý trong việc xây dựng và triển khai ứng dụng. Nghiên cứu các thuật toán, các phương pháp, các công cụ từ đó phân tích đánh giá, triển khai xây dựng ứng dụng. Áp dụng cơ

sở lý thuyết làm nền tảng để xây dựng và hướng phát triển ứng dụng. Trong đó tập trung nghiên cứu vấn đề sau:

- Nghiên cứu cách thức tấn công Phishing thư điện tử [2]: Đây là một kỹ cơ bản trong tấn công Phishing. Tin tặc sẽ gửi thư cho người dùng dưới danh nghĩa là một đơn vị, tổ chức uy tín, dụ người dùng truy cập vào đường link giả mạo và tin tặc có được thông tin mong muốn.
- Nghiên cứu cách thức tấn công Phishing website [2]: việc giả mạo website trong tấn công Phishing thực chất là giả một phần của trang chủ chứ không phải là toàn bộ website. Trang được làm giả thường là trang đăng nhập để lấy thông tin của nạn nhân.

Jian Feng và cộng sự trong một nghiên cứu về phương pháp phát hiện trang web lừa đảo dựa trên tính năng Web2Vec [3] trong nghiên cứu này thành phần chính là automatic RL từ tính năng đặc trưng của mô hình multi-aspects thông qua RL và trích xuất các tính năng bằng phương pháp mạng học sâu. Thứ nhất mô hình xử lý URL, nội dung trang HTML và cấu trúc DOM của trang Web dưới dạng ký tự tương ứng và sử dụng công nghệ RL để tự động học cách biểu diễn của các website sau đó gửi nhiều biểu diễn đến một mạng học sâu bao gồm một mạng nơ-ron phức tạp và mạng hai chiều thông qua các kênh khác nhau để trích xuất mạng cục bộ và mạng toàn cầu và sử dụng các cơ chế chú ý để tăng cường ảnh hưởng của các đối tượng ở các vị trí địa lý quan trọng. Cuối cùng đầu ra của nhiều kênh được hợp nhất để thực hiện dự đón phân loại. Thông qua các thử nghiệm của Jiang Feng và cộng sự kết quả cho thấy hiệu quả phân loại tổng thể của mô hình tốt hơn các phương pháp truyền thống. Qua đó ta thấy rằng các công việc trích xuất các tính năng trang Web từ nhiều khía cạnh thông qua kết hợp giữa representation learning và mạng học sâu có thể cải thiện hiệu quả phát hiện các trang Web lừa đảo.

Một nghiên cứu khác của nhóm Harikrishnan NB, Vinayakumar R, Soman KP [4] Họ sử dụng TFIDF + SVD và TFIDF + NMF representations sử dụng máy học cho phân loại email hợp pháp hay lừa đảo. Các hiệu suất của Decision Tree

and Random Forest là cao nhất trong trường hợp đào tạo chính xác. Kết quả, dữ liệu cho thấy Decision Tree and Random Forest không phụ thuộc vào giới hạn dữ liệu. Với dữ liệu không cân bằng, tỉ lệ cao, chúng có thể đạt được khả năng phát hiện tỉ lệ email lừa đảo cao. Tỉ lệ phát hiện email lừa đảo bằng phương pháp này có thể được tăng cường thêm các nguồn dữ liệu bổ sung một các dễ dàng. Điều này được coi là một hướng đi quan trọng được hướng tới trong tương lai. Giờ hạn của phương pháp này là tác giả không sử dụng phương pháp Deep learning cho các phương pháp trên.

Yasser Yasani [5] và cộng sự trình bày một phương pháp tổ hợp dựa trên thuật toán K-mean Clustering và thuật toán ID3 Decision Tree cho phân loại các bất thường và bình thường trong lưu lượng ARP trong mạng máy tính. Các phương pháp K-mean Clustering được áp dụng cho các trường hợp huấn luyện thông thường để phân vùng nó thành K-clusters bằng cách sử dụng tương tự khoảng cách Euclidean. ID3 Decision Tree được xây dựng trên từng cụm. Điểm bất thường từ thuật toán phân cụm k-Means và quyết định của ID3 Decision Tree được trích xuất. Một thuật toán được sử dụng để kết hợp kết quả của hai thuật toán và thu được giá trị điểm bất thường cuối cùng. Các quy tắc ngưỡng được áp dụng để đưa ra quyết định về tính chính xác của phiên bản thử nghiệm. Thử nghiệm được thực hiện trên lưu lượng ARP mạng đã thu được. Một số tiêu chí bất thường đã được xác định và áp dụng cho lưu lượng ARP thu được để tạo ra quá trình đào tạo bình thường các trường hợp. Hiệu suất của phương pháp đề xuất được đánh giá bằng cách sử dụng năm thước đo đã được xác định và so sánh với hiệu suất của từng cụm k-Means và ID3 Decision Tree và các phương pháp tiếp cận được đề xuất khác dựa trên chuỗi Markovian và tự động học ngẫu nhiên. Kết quả thực nghiệm cho thấy rằng cách tiếp cận được đề xuất có tính cụ thể và giá trị dự đoán chính xác cao.

Một nghiên cứu của nhóm tác giả Manh Thang Nguyen, Alexander Kozachok trình bày mô hình biểu diễn các yêu cầu Web, dựa trên mô hình không gian vectơ và các thuộc tính của các yêu cầu đó sử dụng giao thức HTTP, sử dụng

bộ dữ liệu KDD 99 [6] trong đào tạo cũng như phát hiện tấn công đi kèm với việc biểu diễn truy vấn dựa trên không gian vectơ và phân loại dựa trên mô hình cây quyết định. Nhằm tăng cường độ chính xác phát hiện các cuộc tấn công máy tính vào các ứng dụng Web. Kết quả tập dữ liệu lớn, thời gian và kiểm tra kỹ thuật cần được cải thiện.

Một số nghiên cứu khác tập trung theo hướng áp dụng các thuật toán máy học để phát hiện xâm nhập như trong báo cáo luận văn Máy vector hỗ trợ đa lớp và ứng dụng phát hiện tấn công của Tác giả Nguyễn Đức Hiền [7] tập trung nghiên cứu kỹ thuật M-SVM vào việc phân loại các kết nối mạng trên bộ dữ liệu KDD 99 [9]. Trong nghiên cứu này độ chính xác của thuật toán phụ thuộc vào các tham số δ và C do người sử dụng lựa chọn đồng thời với tập dữ liệu lớn thời gian huấn luyện và kiểm tra của kỹ thuật này vẫn cần được cải thiện.

Một cuộc khảo sát của tác giả Abdul Basit, Maham Zafar và cộng sự [2] cho chỉ ra cho các nhà nghiên cứu hiểu được các phương pháp và xu hướng để phát hiện tấn công Phishing có độ chính xác cao bằng cách phân tích và thực nghiệm các phương pháp Machine learning và phương pháp Deep learning, ngoài ra tác giả cũng đề cập đến các phương pháp phân loại như RF, SVM, Thuật toán C4.5, DT, PCA, k-NN thường được sử dụng hiệu quả trong phát hiện tấn công Phishing.

Phishing [8] là một loại tấn công mạng nổi tiếng với việc đánh cắp thông tin cá nhân của người dùng mà họ không hề hay biết. Mặc dù các nhà nghiên cứu đã đề xuất nhiều phương pháp phát hiện lừa đảo, nhưng hầu hết các phương pháp đều tốn kém về mặt tính toán và khó cập nhật các quy tắc phát hiện của chúng dựa trên những thay đổi trong các mẫu tấn công. Trong bài báo này, các tác giả đề xuất PhishTrim, một phương pháp phát hiện URL lừa đảo dựa trên học đại diện sâu, nhanh và thích ứng. Các tác giả của bài nghiên cứu này nhận được bản trình bày nhúng ban đầu của các URL thông qua mô hình đào tạo trước Skip-gram. Sau đó, bộ nhớ dài hạn hai chiều (Bi-LSTM) được sử dụng để trích xuất sự phụ thuộc vào ngữ cảnh để tìm hiểu thêm về cách trình bày sâu sắc của URL. Các tính năng n-gram cục bộ được trích xuất bằng cách sử dụng Mạng thần kinh tích chập (CNN).

Các thử nghiệm cho thấy PhishTrim hoạt động tốt hơn trên các tập dữ liệu quy mô lớn với độ chính xác 99,797% và chỉ ra rằng phương pháp của họ có khả năng nhất định để phát hiện các cuộc tấn công lừa đảo zero-day. Tập dữ liệu PhishTrim2019 của nhóm tác giả được xuất bản tại https://github.com/DataReleased/PhishTrim.

Lừa đảo (Phishing) là quá trình mô tả các trang web ác tính thay cho các trang web chính hãng để lấy thông tin quan trọng và tế nhị từ người dùng cuối. Ngày nay, lừa đảo trực tuyến được coi là một trong những mối đe dọa nghiêm trọng nhất đối với bảo mật web. Hầu hết các kỹ thuật hiện tại có để phát hiện lừa đảo thông qua việc sử dụng phân loại của Bayes để phân biệt các trang web ác tính với các trang web chính hãng. Các phương pháp này hoạt động tốt nếu một tập dữ liệu chứa ít trang web và chúng cung cấp độ chính xác lên đến 90 phần trăm. Trong những năm gần đây, kích thước của web đang tăng lên rất nhiều và các phương pháp hiện có không còn cung cấp độ chính xác đủ tốt cho các tập dữ liệu lớn. Vì vậy, bài báo [9] đề xuất một cách tiếp cận sáng tạo để xác định các trang web lừa đảo bằng cách sử dụng các siêu liên kết có sẵn trong mã nguồn của trang HTML trong trang web tương ứng. Phương pháp được đề xuất sử dụng một vector đặc trưng với 30 tham số để phát hiện các trang web ác tính. Các tính năng này được sử dụng để đào tạo mô hình Mạng thần kinh sâu được giám sát với trình tối ưu hóa Adam để phân biệt các trang web lừa đảo với các trang web chính hãng. Mô hình học sâu được đề xuất với Adam Optimizer sử dụng cách tiếp cận Listwise để phân loại các trang web lừa đảo và trang web chính hãng. Hiệu suất của phương pháp được đề xuất là khá tốt khi so sánh với các phương pháp học máy truyền thống khác như SVM, Adaboost, AdaRank. Kết quả cho thấy cách tiếp cận được đề xuất cung cấp kết quả chính xác hơn trong việc phát hiện các trang web lừa đảo.

Với việc áp dụng rộng rãi blockchain vào trong lĩnh vực tài chính, bảo mật đã phải đối mặt với những thách thức rất lớn do tội phạm mạng mang lại, đặc biệt là các trò lừa đảo trực tuyến. Nó buộc các tác giả phải khám phá các biện pháp và quan điểm đối phó hiệu quả hơn để có giải pháp tốt hơn. Vì mô hình biểu đồ cung cấp thông tin phong phú cho các tác vụ hạ lưu có thể xảy ra, nhóm tác giả sử dụng

biểu đồ xung quanh để mô hình hóa dữ liệu giao dịch của một địa chỉ đích, nhằm mục đích phân tích danh tính của một địa chỉ bằng cách xác định mẫu giao dịch của nó trên cấu trúc cấp cao. Trong bài báo [10], nhóm tác giả đề xuất một khung phân loại dựa trên đồ thị trên Ethereum. Đầu tiên, các tác giả thu thập hồ sơ giao dịch của một số địa chỉ lừa đảo đã được xác minh và cùng một số địa chỉ bình thường. Thứ hai, họ tạo một tập hợp các đồ thị con, mỗi đồ thị chứa một địa chỉ đích và mạng lưới giao dịch xung quanh của nó để thể hiện địa chỉ ban đầu ở mức đồ thị. Cuối cùng, dựa trên phân tích dòng Ether của chu kỳ lừa đảo lừa đảo, các tác giả đề xuất một Graph2Vec cải tiến và đưa ra dự đoán phân loại trên các đồ thị con mà họ đã xây dựng. Kết quả thử nghiệm cho thấy khung của họ đã đạt được khả năng cạnh tranh lớn trong nhiệm vụ phân loại cuối cùng, điều này cũng chỉ ra giá trị tiềm năng của việc phát hiện lừa đảo trên Ethereum thông qua việc học cách đại diện của mạng giao dịch.

Sau khi nghiên cứu các tài liệu liên quan đến đề tài, học viên nhận thấy độ chính xác và thời gian phát hiện tấn công giả mạo là hai yếu tố quan trong. Trong đề tài này sẽ tập trung vào hai yếu tố trên để tăng hiệu quả khả năng phát hiện xâm nhập với thời gian phù hợp nhất.

3. Mục đích nghiên cứu

Mục tiêu chính: *Xây dựng mô hình máy học sử dụng phương pháp representation learning để phát hiện tấn công phishing nhằm nâng cao độ chính xác của phát hiện*. Từ mục tiêu trên, luận văn sẽ có những mục tiêu cụ thể như sau:

- Nghiên cứu cơ sở lý thuyết về tấn công Phishing, các kỹ thuật phát hiện ra tấn công Phishing.
- Nghiên cứu về thuật toán máy học Representation Learning, các ưu điểm nhược điểm và đặc tính của phương pháp này.
- Nghiên cứu và thu thập bộ dữ liệu liên quan tới tấn công phishing … để nhằm phát hiện ra Phishing. Từ đó xây dựng mô hình dự báo / cảnh báo tấn công Phishing thông qua dữ liệu huấn luyện.

- Nghiên cứu xây dựng ứng dụng phát hiện tấn công Phishing thông qua mô hình dự báo với representation learning.

4. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính là tấn công Phishing [7], [11] và phương pháp representation learning [2], [12], nghiên cứu các mô hình dự báo áp dụng vào phương pháp representation learning.

- Tìm hiểu về tổng quan về phương pháp representation learning
- Tìm hiểu về các thuật toán & kỹ thuật liên quan đến phương pháp representation learning.
- Các dữ liệu đặc trưng của Trung tâm tích hợp dữ liệu tỉnh Tây Ninh có thể áp dụng vào phát hiện tấn công Phishing / hoặc lấy dữ liệu từ các trang cộng đồng mạng như KAGGLE, MENDELEY…
- Nghiên cứu chính là phát hiện tấn công Phishing thông qua máy học sử dụng phương pháp representation learning.
- Nghiên cứu các kỹ thuật máy học phổ biến như R, MatLab, Python… để xây dựng mô hình phát hiện tấn công bằng phương pháp máy học.

5. Phạm vi nghiên cứu

Xây dựng mô hình mô phỏng máy học, sử dụng phương pháp để phát hiện tấn công Phishing: Mô phỏng thực hiện trong mạng LAN nhỏ (từ 3~5 máy) có một số máy chủ ảo có kết nối Internet có tấn công Phishing.

6. Phương pháp nghiên cứu

- *Phương pháp luận:* Dựa trên cơ sở là lý thuyết về phương pháp RL; Dự kiến dùng mô hình RL học viên áp dụng các phương pháp Deep Learning và HTML Analysis [13]. Học viên dự kiến dùng dữ liệu được tải về từ các website cung cấp thông tin các đường link giả mạo miễn phí như: https://www.kaggle.com/datasets/shashwatwork/web-page-phishing-detection-dataset để thử nghiệm và nghiên cứu.
- *Phương pháp đánh giá dựa trên cơ sở toán học:* Trên cơ sở các lý thuyết về phương pháp RL. Đề xuất ra thuật toán để dự báo khả năng xảy ra một cuộc tấn

Phishing có độ chính xác cao dựa trên các thuật toán, chứng minh thuật toán và đánh giá hiệu quả của thuật toán.

- *Phương pháp đánh giá bằng mô hình mô phỏng thực nghiệm:* Xây dựng mô hình mô phỏng và thực nghiệm để hoàn thành đề xuất.

7. Bố cục luận văn

Bên cạnh phần mở đầu, phần kết luận và phần tài liệu tham khảo, phần nội dung chính của bài nghiên cứu được chia thành 3 chương chính như sau:

Chương 1: Tổng quan tấn công phishing và representation learning

Chương 2: Xây dựng mô hình phát hiện tấn công phishing

Chương 3: Thí nghiệm và đánh giá

CHƯƠNG 1. TỔNG QUAN TẤN CÔNG PHISHING VÀ REPRESENTATION LEARNING

1.1. Tổng quan về tấn công Phishing

Phishing [1] là một trong những loại tấn công mạng nguy hiểm do các tội phạm mạng gây ra bằng cách tạo ra các thông tin giả mạo từ các website, cơ sở, doanh nghiệp uy tín nhằm lừa đảo và chiếm đoạt thông tin của người dùng. Phishing kết hợp nhiều kỹ thuật giả mạo tinh vi đến mức người dùng không thể phát hiện ra và tự động cung cấp thông tin quan trọng cho kẻ xấu. Loại tấn công này thường chủ yếu nhắm đến những người thiếu kiến thức về bảo mật trên môi trường mạng, không quan tâm đến quyền riêng tư về thông tin của các loại tài khoản như Facebook, Gmail, tài khoản thẻ tín dụng ngân hàng và các loại tài khoản liên quan đến tài chính khác,…

Các loại tấn công Phishing được tiếp cận rất đa dạng và biến hóa khôn lường. Một số các loại tấn công Phishing có thể kể đến như [14]:

- **Email Phishing:** Ở loại tấn công này, các tội phạm mạng sẽ gửi mail đến người dùng và yêu cầu người dùng xác thực hoặc cập nhật thông tin vào một biểu mẫu hoặc đường link được đính kèm trong mail. Người dùng dễ dàng mắc bẫy nếu không kiểm tra cẩn thận các mail được gửi đến và từ đó vô tình để lộ thông tin cá nhân quan trọng của mình khi click vào các đường link hoặc.
- **Spear Phishing:** Các tội phạm mạng sẽ tấn công vào một tổ chức hoặc một cá nhân cụ thể. Đây là kiểu tấn công chuyên sâu, các tội phạm mạng đã nắm bắt rõ các thông tin liên quan đến cá nhân hoặc tổ chức đó.
- **Whailing:** Kiểu tấn công này sẽ nhắm vào những cá nhân có vai trò quan trọng trong một công ty hoặc tổ chức ví dụ như CEO, CFO,… nhằm chiếm đoạt thông tin từ cá nhân này cũng như những người liên quan khác một cách dễ dàng hơn.

- **Smishing:** Cách tấn công này sẽ tiếp cận người dùng thông qua tin nhắn SMS, kẻ tấn công sẽ gửi tin nhắn kèm theo link lừa đảo với nội dung đa dạng nhằm hấp dẫn người dùng click vào link để chiếm đoạt thông tin.
- **Vishing:** Loại tấn công này còn có tên gọi khác là Voice Vishing, tiếp cận nạn nhân thông qua đoạn tin nhắn hội thoại với danh xưng là nhân viên của một dịch vụ hoặc tổ chức mà nạn nhân đang sử dụng (thường là nhân viên ngân hàng), sau đó yêu cầu nạn nhân gọi vào các số điện thoại miễn phí cước nhằm chiếm đoạt các thông tin liên quan đến ngân hàng của nạn nhân.
- **Pharming:** Tấn công đến máy tính nạn nhân bằng mã độc, thay đổi file host trên máy tính nạn nhân, khai thác các lỗ hổng DNS để dẫn người dùng đến một trang web giả mạo khi người dùng truy cập vào một trang web uy tín.
- **Content-injection Phishing:** Kẻ tấn công sẽ thay đổi ngẫu nhiên một số nội dung của một trang web uy tín và các nội dung này tương tự với nội dung trên trang web uy tín để người dùng dễ dàng tin tưởng và nhập các thông tin cá nhân.
- **Search Engine Phishing:** Ở loại tấn công này, kẻ tấn công sẽ tạo ra một website thu hút người dùng với những khuyến mãi, quà tặng trúng thưởng và đặc biệt là với nội dung website phù hợp với các công cụ tìm kiếm, từ đó người dùng sẽ dễ dàng tìm đến website như thế này và bị lừa để nhập các thông tin cá nhân để nhận thưởng.

Hình 1.1: Các loại tấn công Phishing [14]

1.2. Các phương pháp phòng chống và phát hiện Phishing trên mạng

Tấn công Phishing luôn tiềm ẩn và khó nhận biết vì mức độ tinh vi của nó với bất kì cá nhân hoặc tổ chức nào, vì vậy các cá nhân hoặc tổ chức cần nâng cao cảnh giác đối với các loại tài khoản cũng như thông tin cá nhân của mình. Một số cách phòng chống tấn công Phishing được trang Trung tâm an ninh mạng quốc gia của chính phủ nước Anh đề xuất [15] như sau:

- **Cấu hình tài khoản:** các tổ chức nên cấu hình các loại tài khoản của nhân viên theo nguyên tắc giảm thiểu tối đa các loại đặc quyền, chỉ cấp các quyền cần thiết cho nhân viên. Điều này sẽ giảm thiểu rủi ro đáng kể nếu như tài khoản của nhân viên bị tấn công Phishing. Ngoài ra, để tăng cường thêm tính bảo mật và giảm rủi ro khi bị tấn công bằng các loại mã độc thì các tổ chức cần đảm bảo rằng nhân viên không truy cập vào bất cứ website hay kiểm tra email bằng tài khoản được cấp bởi tổ chức. Thêm vào đó, tính năng

xác thực hai bước (2FA) trên tài khoản (ví dụ như email) cũng sẽ nâng cao tính bảo mật cho tài khoản.

- **Tập huấn cho nhân viên:** Các nhân viên trong một tổ chức cần được tập huấn để hiểu được cách hoạt động bình thường của hệ thống, từ đó có thể tự trang bị cho bản thân các kiến thức cũng như nhận biết được những lúc hệ thống có các hoạt động bất thường.
- **Kiểm tra các dấu hiệu của Phishing:** nâng cao cảnh giác với một số email đến từ nước ngoài, có nội dung không hoàn chỉnh (lỗi chính tả, sai dấu chấm câu,…). Tuy nhiên với thủ đoạn ngày càng tinh vi, các email được gửi được tinh chỉnh nội dung và hình thức một cách chuyên nghiệp, vì vậy chỉ nên click vào các đường dẫn trong mail nếu có sự chỉ đạo từ cấp trên hoặc thật sự tin tưởng. Ngoài ra, các email có nội dung đe dọa như buộc tội, vu khống và yêu cầu phải click vào đường link ngay lập tức cũng là dấu hiệu cho thấy đây là mail giả mạo nhằm mục đích chiếm đoạt thông tin hoặc cài các phần mềm mã độc vào máy tính. Thêm vào đó, cần chú ý đến tên và địa chỉ các email được gửi từ cấp trên phải trùng khớp với tên và địa chỉ của email chính chủ.
- **Báo cáo lại tất cả các cuộc tấn công:** bản thân nhân viên của một tổ chức nếu có phát hiện bất cứ trường hợp tấn công nào hoặc có thể đã trở thành nạn nhân của cuộc tấn công thì cần báo cáo lại với cấp trên để được hỗ trợ kịp thời, tránh những rủi ro đáng tiếc xảy ra.
- **Kiểm tra dấu vết thông tin cá nhân:** kẻ tấn công thường sẽ tìm đến thông tin của tổ chức và thông tin các nhân viên liên quan để tạo ra các cuộc tấn công Phishing với khả năng thành công cao, vì vậy cần ý thức đến việc chia sẻ thông tin nhạy cảm về cơ quan, tổ chức hoặc thông tin cá nhân trên các trang mạng xã hội để tránh các cuộc tấn công có thể xảy đến. Bên cạnh đó, cần phải đảm bảo được các đối tác, nhà cung cấp của cơ quan, tổ chức không chia sẻ thông tin liên quan đến cơ quan hay tổ chức này một cách tùy tiện. Ngoài ra, thay vì kiểm soát chặt chẽ thông tin cá nhân của nhân viên, các tổ

chức nên điều chỉnh và tập huấn cho nhân viên nhận biết được những thông tin nào có thể chia sẻ được và ngược lại.

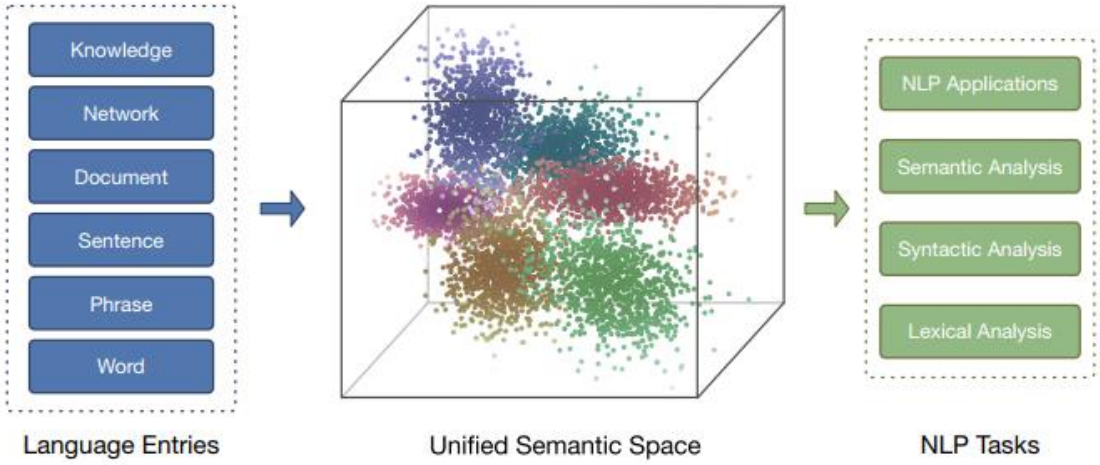
1.3. Tổng quan về representation learning

Representation learning là tập hợp các kỹ thuật cho phép một hệ thống tự động khám phá các biểu diễn cần thiết để phát hiện hoặc phân loại đặc trưng từ bộ dữ liệu thô [16] [17]. Điều này sẽ thay thế kỹ thuật trích xuất đặc trưng và cho phép máy có khả năng vừa học các đặc trưng vừa sử dụng chúng để thực hiện một tác vụ cụ thể. Ở các miền dữ liệu về khoa học như trí tuệ nhân tạo (AI), tin sinh học (Bioinformatics) hay tài chính, việc học các dữ liệu đặc trưng là một bước quan trọng để tạo điều kiện cho quá trình phân lớp, rút trích và đề xuất các tác vụ tiếp sau đó [18]. Mạng nơ-ron sâu có thể được coi là mô hình RL thường mã hóa thông tin được chiếu vào một không gian con khác. Sau đó, những biểu diễn này thường được chuyển cho một bộ phân loại tuyến tính, ví dụ, để huấn luyện một bộ phân loại. RL có thể chia thành:

- Supervised representation learning: học các biểu diễn về nhiệm vụ A bằng cách sử dụng dữ liệu được chú thích và được sử dụng để giải quyết nhiệm vụ B.
- Unsupervised representation learning: học các biểu diễn về một nhiệm vụ theo cách không được giám sát (dữ liệu không có nhãn). Sau đó, chúng được sử dụng để giải quyết các tác vụ xuôi dòng và giảm nhu cầu về dữ liệu có chú thích khi tìm hiểu các tác vụ tin tức. Các mô hình mạnh mẽ như GPT và BERT tận dụng việc học đại diện không giám sát để giải quyết các nhiệm vụ ngôn ngữ.

Các kĩ thuật Representation Learning lần đầu tiên được phát triển để phục vụ cho quá trình xử lí ngôn ngữ tự nhiên, tuy nhiên chúng đã được mở rộng sang kiểu xử lí dữ liệu khác như là hình ảnh, video và hệ thống mạng. Trong lĩnh vực phân tích dữ liệu, RL đóng vai trò quan trọng trong việc dự đoán các tác vụ, phát hiện gian lận trong quá trình giao dịch qua thẻ tín dụng [19]. Thêm vào đó, RL cũng đã trở thành một kĩ thuật không thể thiếu trong các nghiên cứu và ứng dụng

về NLP. RL hỗ trợ việc chuyển giao tri thức qua nhiều mục thông tin về ngôn ngữ, đa tác vụ trong NLP và đa miền ứng dụng, đồng thời cải thiện và tối ưu hiệu quả cũng như hiệu suất của NLP một cách đáng kể [20].



Hình 1.2: Quá trình phân loại đặc trưng nhằm cung cấp không gian ngữ nghĩa thống nhất cho hỗn hợp đa thông tin về ngôn ngữ và đa tác vụ trong NLP [20]

1.4. Một số đặc điểm nổi bật của representation learning

Ưu tiên cho RL trong AI

Smoothness: giả sử hàm được học f là s.t. x ≈ y thường ngụ ý f (x) ≈ f (y).

Nhiều yếu tố giải thích: phân phối tạo dữ liệu được tạo ra bởi các yếu tố cơ bản khác nhau và phần lớn những gì người ta tìm hiểu về một yếu tố sẽ khái quát trong nhiều cấu hình của các yếu tố khác. Mục tiêu để khôi phục hoặc ít nhất là gỡ rối các yếu tố cơ bản của sự biến đổi này.

Một tổ chức có thứ bậc của các yếu tố giải thích: các khái niệm hữu ích để mô tả thế giới xung quanh có thể được định nghĩa theo các khái niệm khác, trong một hệ thống thứ bậc, với các khái niệm trừu tượng hơn trong hệ thống thứ bậc, được định nghĩa theo các khái niệm ít trừu tượng hơn.

Học bán giám sát: với đầu vào X và mục tiêu Y để dự đoán, một tập hợp con của các yếu tố giải thích phân phối của X giải thích phần lớn Y, cho X. Do đó, các biểu diễn hữu ích cho P (X) có xu hướng hữu ích khi học P (Y | X), cho phép

chia sẻ sức mạnh thống kê giữa các nhiệm vụ học tập được giám sát và không giám sát.

Các yếu tố được chia sẻ giữa các nhiệm vụ: với nhiều Y quan tâm hoặc nhiều nhiệm vụ học tập nói chung, các nhiệm vụ (ví dụ: tương ứng với P (Y | X, nhiệm vụ)) được giải thích bằng các yếu tố được chia sẻ với các nhiệm vụ khác, cho phép chia sẻ các điểm mạnh thống kê qua các nhiệm vụ.

Manifolds: khối lượng xác suất tập trung gần các vùng có kích thước nhỏ hơn nhiều so với không gian ban đầu nơi dữ liệu tồn tại.

Phân cụm tự nhiên: các giá trị khác nhau của các biến phân loại như các lớp đối tượng được liên kết với các đa tạp riêng biệt. Chính xác hơn, các biến cục bộ trên đa tạp có xu hướng bảo toàn giá trị của một danh mục và nội suy tuyến tính giữa các ví dụ của các lớp khác nhau nói chung liên quan đến việc đi qua một vùng mật độ thấp, tức là P (X | Y = i) cho các i khác nhau có xu hướng tách biệt rõ ràng và không trùng lặp nhiều. các nhiệm vụ học tập thường liên quan đến việc dự đoán các biến phân loại như vậy.

Tính nhất quán theo thời gian và không gian: các quan sát liên tiếp (từ một trường hợp) hoặc các quan sát gần nhau về mặt không gian có xu hướng được liên kết với cùng một giá trị của các khái niệm phân loại có liên quan, hoặc dẫn đến một chuyển động nhỏ trên bề mặt của đa tạp mật độ cao. Nhìn chung, các yếu tố khác nhau thay đổi ở các quy mô không gian và thời gian khác nhau, và nhiều khái niệm phân loại về sở thích thay đổi chậm. Khi cố gắng nắm bắt các biến phân loại như vậy, điều này có thể được thực thi trước đó bằng cách làm cho các đại diện liên quan thay đổi từ từ, tức là phạt những thay đổi trong giá trị theo thời gian hoặc không gian.

Độ thưa thớt: đối với bất kỳ quan sát x đã cho nào, chỉ một phần nhỏ các yếu tố có thể là có liên quan. Về mặt biểu diễn, điều này có thể được biểu thị bằng các đặc trưng thường bằng 0, hoặc thực tế là hầu hết các đối tượng được trích xuất không nhạy cảm với các biến thể nhỏ của x. Điều này có thể đạt được với một số dạng mồi nhất định trên các biến tiềm ẩn (đạt đỉnh là 0) hoặc bằng cách sử dụng độ

không tuyến tính có giá trị thường bằng phẳng ở 0 (tức là 0 và với đạo hàm 0), hoặc đơn giản bằng cách xử lý độ lớn của ma trận Jacobian (của các đạo hàm) của đầu vào ánh xạ hàm để biểu diễn.

Tính đơn giản của các yếu tố phụ thuộc: trong các biểu diễn cấp cao, các yếu tố có liên quan với nhau thông qua các phụ thuộc tuyến tính, đơn giản. Điều này có thể thấy trong nhiều định luật vật lý và được giả định khi cắm một công cụ dự đoán tuyến tính lên trên một biểu diễn đã học.

Các yếu tố bất đồng của sự thay đổi

Các yếu tố giải thích khác nhau của dữ liệu có xu hướng thay đổi độc lập với nhau trong phân phối đầu vào và chỉ một số yếu tố tại thời điểm có xu hướng thay đổi khi người ta xem xét một chuỗi các đầu vào liên tiếp trong thế giới thực. Dữ liệu phức tạp phát sinh từ sự tương tác phong phú của nhiều nguồn. Các yếu tố này tương tác trong một trang web phức tạp có thể làm phức tạp các nhiệm vụ liên quan đến AI như phân loại đối tượng. Ví dụ, một hình ảnh bao gồm sự tương tác giữa một hoặc nhiều nguồn sáng, các hình dạng vật thể và chất liệu chống đỡ của các bề mặt khác nhau hiện diện trong hình ảnh. Bóng từ các đối tượng trong cảnh có thể đổ lên nhau theo các mẫu phức tạp, tạo ra ảo giác về ranh giới đối tượng ở những nơi không có và ảnh hưởng đáng kể đến hình dạng đối tượng được cảm nhận. Phải tận dụng chính dữ liệu, sử dụng số lượng lớn các ví dụ không được gắn nhãn, để tìm hiểu các biểu diễn tách biệt các nguồn giải thích khác nhau. Làm như vậy sẽ làm tăng khả năng biểu diễn mạnh mẽ hơn đáng kể đối với các biến thể phức tạp và có cấu trúc phong phú hiện có trong các nguồn dữ liệu tự nhiên cho các tác vụ liên quan đến AI.

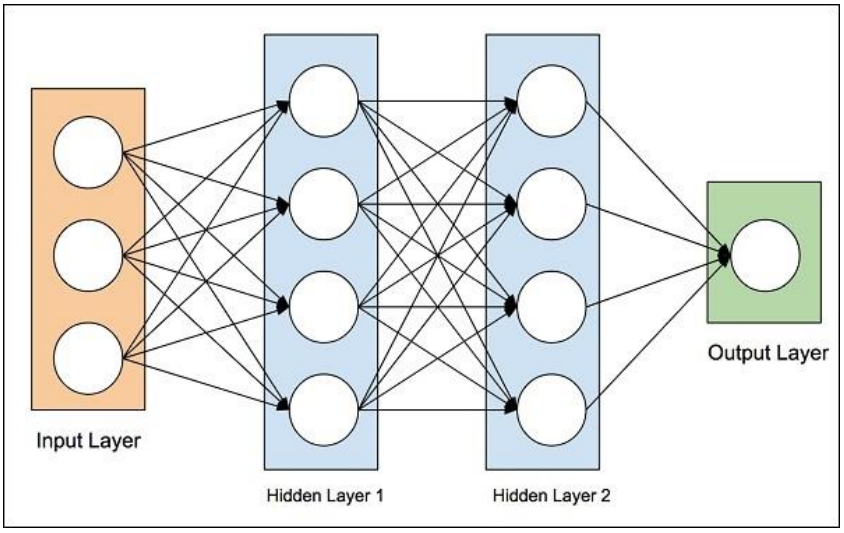
Điều quan trọng là phải phân biệt giữa các mục tiêu liên quan nhưng riêng biệt của việc học các tính năng bất biến và học cách gỡ rối các yếu tố giải thích. Sự khác biệt trung tâm là bảo quản thông tin. Các tính năng bất biến, theo định nghĩa, đã giảm độ nhạy theo hướng bất biến. Đây là mục tiêu của việc xây dựng các tính năng không nhạy cảm với sự thay đổi trong dữ liệu không liên quan đến nhiệm vụ đang thực hiện. Thật không may, thường rất khó xác định trước rằng tập hợp các

tính năng và biến thể nào cuối cùng sẽ liên quan đến nhiệm vụ đang thực hiện. Hơn nữa, như thường lệ trong bối cảnh của các phương pháp học sâu, tập hợp tính năng đang được đào tạo có thể được sử dụng trong nhiều nhiệm vụ có thể có tập hợp con riêng biệt của các tính năng có liên quan. Những cân nhắc như vậy đưa đến kết luận rằng cách tiếp cận mạnh mẽ nhất để học tính năng là loại bỏ càng nhiều yếu tố càng tốt, loại bỏ càng ít thông tin về dữ liệu càng tốt. Nếu một số hình thức giảm kích thước là mong muốn, thì chúng tôi giả thuyết rằng các hướng cục bộ của biến thể ít được thể hiện nhất trong dữ liệu huấn luyện trước tiên phải được loại bỏ (ví dụ như trong PCA, nó thực hiện trên toàn cầu thay vì xung quanh mỗi ví dụ).

1.5. Mạng nơ-ron và deep learning

1.5.1. Mạng nơ-ron

Neural network [33] là một mạng lưới thần kinh được tạo thành từ các nút xử lý được kết nối dày đặc, tương tự như các tế bào thần kinh trong não. Mỗi nút có thể được kết nối với các nút khác nhau trong nhiều lớp bên trên và bên dưới nó. Các nút này di chuyển dữ liệu qua mạng theo kiểu chuyển tiếp, có nghĩa là dữ liệu chỉ di chuyển theo một hướng. Nút "kích hoạt" giống như một nơ-ron khi nó chuyển thông tin đến nút tiếp theo. Một mạng nơ ron đơn giản có một lớp đầu vào, lớp đầu ra và một lớp ẩn giữa chúng. Một mạng có nhiều hơn ba lớp, bao gồm cả đầu vào và đầu ra, được gọi là mạng học sâu. Trong mạng học sâu, mỗi lớp nút đào tạo dựa trên dữ liệu dựa trên kết quả từ lớp trước. Càng nhiều lớp, khả năng nhận ra thông tin phức tạp hơn - dựa trên dữ liệu từ các lớp trước đó càng lớn. (Hình 3.1)



Hình 1.3: Các lớp của một mạng nơ-ron [33]

Mạng đưa ra quyết định bằng cách gán mỗi nút được kết nối với một số được gọi là "trọng số". Trọng số thể hiện giá trị của thông tin được gán cho một nút riêng lẻ (tức là nó hữu ích như thế nào trong việc phân loại thông tin một cách chính xác). Khi một nút nhận được thông tin từ các nút khác, nó sẽ tính toán tổng trọng lượng hoặc giá trị của thông tin. Nếu số lượng vượt quá một ngưỡng nhất định, thông tin sẽ được chuyển sang lớp tiếp theo. Nếu trọng lượng dưới ngưỡng, thông tin sẽ không được chuyển sang. Trong một mạng nơ-ron mới được hình thành, tất cả các trọng số và ngưỡng được đặt thành số ngẫu nhiên. Khi dữ liệu đào tạo được đưa vào lớp đầu vào, trọng số và ngưỡng sẽ tinh chỉnh để luôn mang lại kết quả đầu ra chính xác.

Một số hàm kích hoạt được sử dụng phổ biến:

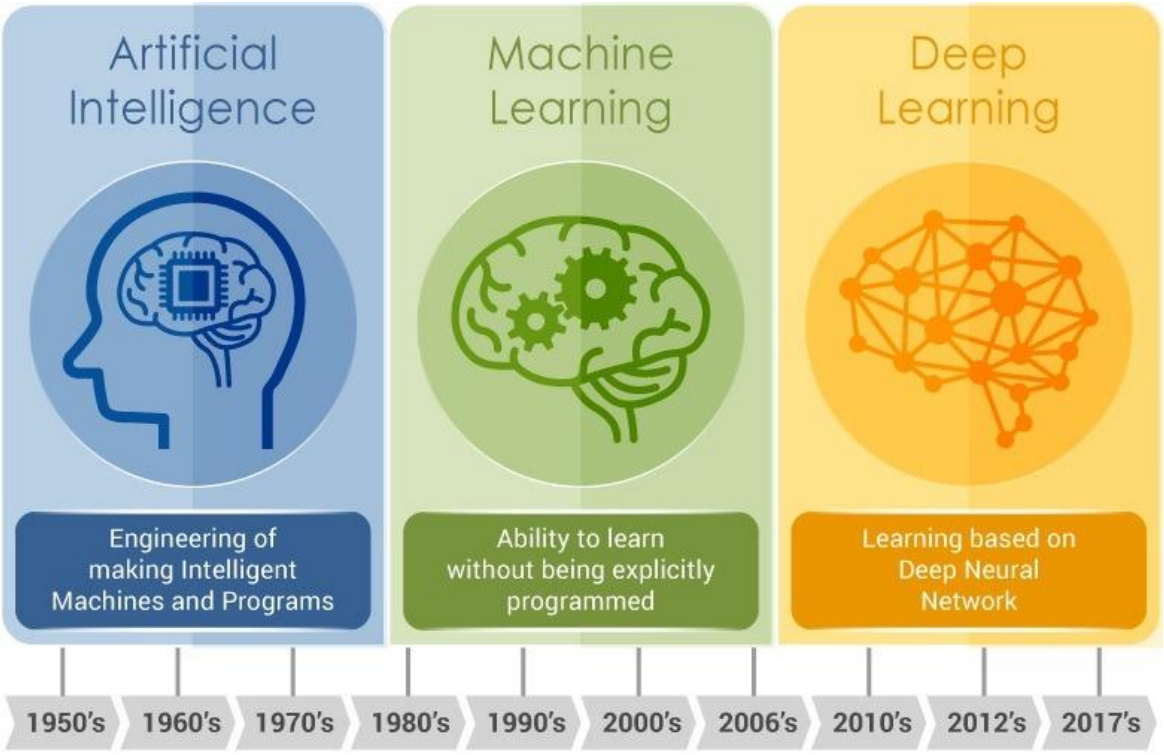
- Binary step: $f(x) = 1, x \geq 0$
- Linear: $f(x) = ax$
- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$
- Tanh: $tanh(x) = \frac{2}{1+e^{-2x}} - 1$
- ReLU: $f(x) = \max(0, x)$
- Leaky ReLU: $f(x) = \begin{cases} x \; với \; x > 0 \\ ax \; ngược \; lại \end{cases}$

– Softmax: $a(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$ $với\ j = 1, \dots, k$

1.5.2. Deep learning

Deep learning (DL) hay học sâu là một tập con của học máy (ML), về cơ bản là một mạng nơ-ron có ba lớp trở lên. Những mạng lưới thần kinh này cố gắng mô phỏng hành vi của não người - mặc dù không phù hợp với khả năng của nó - cho phép nó "học" từ một lượng lớn dữ liệu. Mặc dù mạng nơ-ron với một lớp duy nhất vẫn có thể đưa ra các dự đoán gần đúng, nhưng các lớp ẩn bổ sung có thể giúp tối ưu hóa và tinh chỉnh để có độ chính xác.

DL thúc đẩy nhiều ứng dụng và dịch vụ trí tuệ nhân tạo (AI) nhằm cải thiện tự động hóa, thực hiện các tác vụ phân tích và vật lý mà không cần sự can thiệp của con người. Công nghệ học sâu nằm sau các sản phẩm và dịch vụ hàng ngày (chẳng hạn như trợ lý kỹ thuật số, điều khiển từ xa hỗ trợ giọng nói và phát hiện gian lận thẻ tín dụng) cũng như các công nghệ mới nổi (chẳng hạn như ô tô tự lái). (hình 3.2 mô tả mối quan hệ giữa AI, ML và DL).



Hình 1.4: Mối liên hệ giữa AI, ML và DL [34]

1.6. Các công trình ở trong nước

"Phishing Attacks Detection Using Genetic Programming" [21]

Vào năm 2014, tác giả Phạm Tuấn Anh cùng các cộng sự của mình đã đề xuất giải pháp chống tấn công Phishing bằng Genetic Programming (GP). Sau quá trình thử nghiệm trên tập dữ liệu bao gồm các trang web lừa đảo và hợp pháp được thu thập từ internet, thuật toán học máy GP đã chứng minh tính hiệu quả cao và được nhóm tác giả cho là giải pháp tốt nhất cho việc phát hiện các cuộc tấn công lừa đảo. Các giai đoạn để triển khai giải pháp đề xuất gồm hai phần là trích xuất đặc trưng và mô tả hệ thống. Trích xuất các đặc trưng từ tập dữ liệu là giai đoạn quan trọng và có khả năng ảnh hưởng lớn đến hiệu quả của thuật toán đề xuất. Giai đoạn này sẽ hỗ trợ việc phân biệt các trang web giả mạo và hợp pháp. Ở giai đoạn mô tả hệ thống, tập dữ liệu sẽ được chia làm hai bước để phân tích đó là training và testing.

"Detecting Phishing Web Pages based on DOM-Tree Structure and Graph Matching Algorithm" [22]

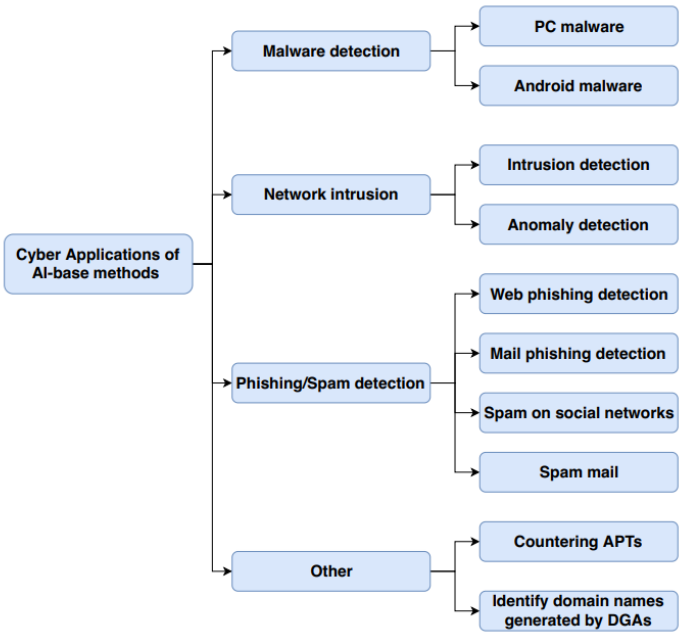
Tác giả Le Dang Nguyen, Đại học Hải Phòng, năm 2014 cùng các cộng sự của mình nghiên cứu và đề xuất các giải pháp để phát hiện các trang web lừa đảo, giả mạo đựa trên cấu trúc của cây DOM (DOM-Tree) và thuật toán Graph Matching. Thuật toán đề xuất gồm 4 bước để phát hiện các trang web lừa đảo lần lượt là trích xuất thông tin từ cây DOM (DOM-Tree Extraction), xác định độ tương đồng giữa web thật và web lừa đảo (Computing Similarity), phát hiện và báo cáo Phishing (Phishing Detect, Phishing Report).

Hình 1.5: Quá trình phát hiện trang web Phishing [22]

"Artificial Intelligence in the Cyber Domain: Offense and Defense" [23]

Một khảo sát về ứng dụng các kĩ thuật AI nhằm nâng cao cơ chế phòng thủ cũng như tấn công trong an toàn bảo mật thông tin được thực hiện bởi Trương Thành Công cùng các cộng sự của mình vào năm 2019. Các đóng góp chính trong bài khảo sát được chia thành 4 phần:

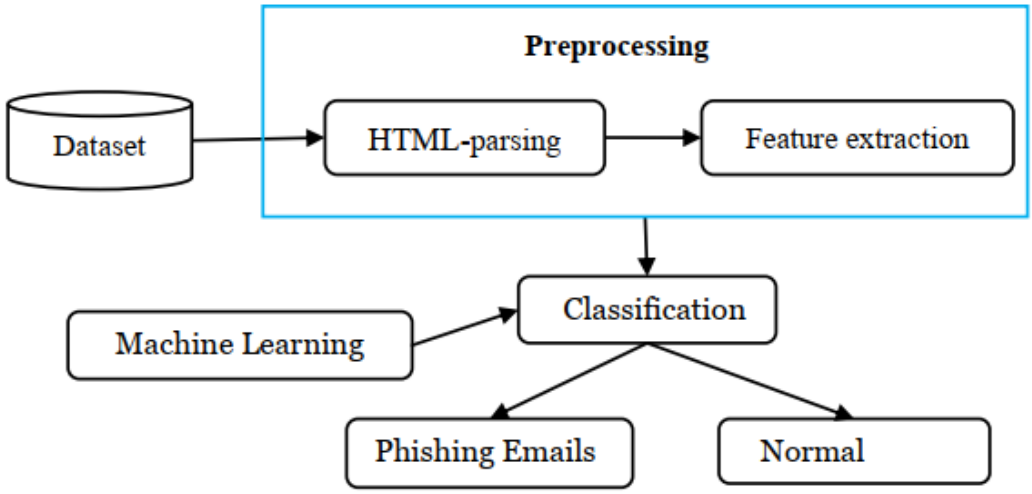
- **Trình bày sự ảnh hưởng của AI đến an toàn bảo mật thông tin (ATBMTT):** cung cấp cái nhìn tổng quan, ngắn gọn về AI.
- **Ứng dụng của AI trong ATBMTT:** thực hiện các khảo sát về các ứng dụng của AI vào ATBMTT, một lĩnh vực được cho là có khả năng bao phủ tất cả các loại tấn công mạng.
- **Thảo luận về tiềm năng của AI trong việc xử lí các mối đe dọa từ đối thủ:** tiến hành nghiên cứu đa dạng các loại tấn công và đe dọa nguy hiểm có khả năng vượt qua hệ thống sử dụng AI.
- **Các thách thức và hướng phát triển:** thảo luận về các thách thức và hướng mở rộng của nghiên cứu về ATBMTT.



Hình 1.6: Một số nhánh chính của các ứng dụng an toàn bảo mật áp dụng các kĩ thuật AI [23]

"A Framework for Vietnamese Email Phishing Detection" [24]

Vào năm 2018, Do Xuan Cho cùng các cộng sự của mình đã thực hiện nghiên cứu về hệ thống phòng chống tấn công Phishing qua email cho người Việt Nam. Nghiên cứu bao gồm các kĩ thuật trích xuất đặc trưng (feature selection) kết hợp với các thuật toán Machine Learing để cải thiện hiệu suất của hệ thống phát hiện tấn công Phishing qua email. Phương pháp đề xuất trong nghiên cứu sử dụng 2 tập dữ liệu bao gồm tập dữ liệu chứa các email Phishing tiếng Việt được thu thập từ các tình nguyện viên người Việt và tập dữ liệu còn lại được sử dụng rộng rãi ở nhiều nghiên cứu khác với nội dung email được trình bày bằng tiếng Anh. Kết quả thí nghiệm của nghiên cứu cho thấy phương pháp đề xuất được ứng dụng thực tiễn cho hệ thống phát hiện các email Phishing cho người Việt.



Hình 1.7: Sơ đồ luồng biểu diễn mô hình ứng dụng Machine Learning [24]

1.7. Các công trình trên thế giới

"lRepresentation Learning: A Review and New Perspectives" [25]

Vào năm 2014, Yoshua Bengio cùng các cộng sự của mình đã thực hiện bài đánh giá và giới thiệu về thuật toán vô cùng mạnh mẽ trong lĩnh vực ML và DL là RL. Học đặc trưng dữ liệu giúp dễ dàng trích xuất thông tin hữu ích khi xây dựng bộ phân loại hoặc các yếu tố dự đoán khác. Trong trường hợp của các mô hình xác suất, một biểu diễn tốt thường là một biểu diễn nắm bắt được sự phân bố sau của các yếu tố giải thích cơ bản cho đầu vào được quan sát. Một đại diện tốt cũng là

một đại diện hữu ích như là đầu vào cho một dự đoán được giám sát. Trong số các cách học biểu diễn khác nhau, bài báo này tập trung vào các phương pháp học sâu: những phương pháp được hình thành bởi sự kết hợp của nhiều phép biến đổi phi tuyến tính, với mục tiêu tạo ra các biểu diễn trừu tượng hơn - và cuối cùng là hữu ích hơn. Ở đây chúng tôi khảo sát khu vực đang phát triển nhanh chóng này với sự chú trọng đặc biệt vào những tiến bộ gần đây. Chúng tôi xem xét một số câu hỏi cơ bản đã thúc đẩy nghiên cứu trong lĩnh vực này. Cụ thể, điều gì làm cho một biểu diễn này tốt hơn một biểu diễn khác? Đưa ra một ví dụ, chúng ta nên tính toán biểu diễn của nó như thế nào, tức là thực hiện trích xuất đối tượng địa lý? Ngoài ra, các mục tiêu thích hợp để học cách đại diện tốt là gì?

Bài đánh giá về RL và DL bao gồm ba cách tiếp cận chính và dường như không kết nối: mô hình xác suất (cả loại có hướng như sparse coding và loại không có hướng như máy Boltzmann), các thuật toán dựa trên tái cấu trúc liên quan đến mã tự động, và phương pháp tiếp cận hình học "đa tạp" (Geometrically motivated manifold-learning). Việc rút ra kết nối giữa các cách tiếp cận này hiện đang là một lĩnh vực nghiên cứu rất tích cực và có khả năng sẽ tiếp tục tạo ra các mô hình và phương pháp tận dụng các điểm mạnh tương đối của mỗi mô hình.

"An overview on data representation learning: From traditional feature learning to recent deep learning" [26]

Bài báo này xem xét nghiên cứu về học biểu diễn dữ liệu, bao gồm học tập tính năng truyền thống và học tập sâu gần đây. Từ sự phát triển của các phương pháp học tập tính năng và mạng nơ-ron nhân tạo, ta có thể thấy rằng học sâu không hoàn toàn mới. Đó là kết quả của sự tiến bộ vượt bậc của nghiên cứu học tập tính năng, sự sẵn có của dữ liệu được gắn nhãn quy mô lớn và phần cứng. Tuy nhiên, bước đột phá của học sâu không chỉ ảnh hưởng đến lĩnh vực trí tuệ nhân tạo mà còn cải thiện đáng kể sự tiến bộ của nhiều lĩnh vực. Đối với các nghiên cứu trong tương lai về học sâu, bài báo đề xuất ba hướng: lý thuyết cơ bản, thuật toán mới và ứng dụng. Một số nhà nghiên cứu đã cố gắng phân tích mạng lưới thần kinh sâu , tuy nhiên, khoảng cách giữa lý thuyết và ứng dụng của học sâu vẫn còn khá lớn.

Mặc dù nhiều thuật toán học sâu đã được đề xuất, nhưng hầu hết chúng đều dựa trên CNN hoặc RNN sâu. Hơn nữa, các thuật toán học sâu đã được khai thác sơ bộ trong nhiều lĩnh vực. Tuy nhiên, để giải quyết một số vấn đề khó khăn trong xử lý ngôn ngữ tự nhiên và tầm nhìn máy tính, các mô hình và thuật toán phức tạp hơn được mong muốn. Cuối cùng, nhấn mạnh rằng học sâu không phải là tất cả mọi thứ của học máy và là cách duy nhất để hiện thực hóa trí tuệ nhân tạo. Để giải quyết các vấn đề trong thế giới thực, nhiều mô hình và thuật toán phân tích dữ liệu thông minh là không thể thiếu

"RLOSD: Representation Learning based Opinion Spam Detection" [17]

Ngày nay, bởi sự gia tăng đáng kể của các bài đánh giá trực tuyến, ảnh hưởng có hại của các bài đánh giá spam đối với việc ra quyết định gây ra những kết quả không thể phục hồi cho cả khách hàng và tổ chức. Các phương pháp hiện tại điều tra để tìm ra cách phân biệt giữa các bài đánh giá spam và không spam. Hầu hết các thuật toán tập trung vào các phương pháp tiếp cận kỹ thuật tính năng để hiển thị chỗ ở của biểu diễn dữ liệu. Bài báo này đề xuất một phương pháp dựa trên cây quyết định để tiết lộ các đánh giá lừa đảo từ những người đáng tin cậy. Sử dụng phương pháp RL không giám sát cùng với các phương pháp lựa chọn đối tượng địa lý truyền thống để trích xuất các đối tượng địa lý thích hợp và đánh giá chúng bằng cây quyết định. mô hình tổ hợp bao gồm các giai đoạn kỹ thuật tính năng và giảm tính năng được xây dựng để giảm kích thước của không gian tính năng và loại bỏ các tính năng dư thừa và không liên quan để cải thiện hiệu suất phát hiện spam xem xét. Trong bước đầu tiên, quá trình tiền xử lý được thực hiện do loại bỏ các từ không hiệu quả khỏi tài liệu và chuẩn bị dữ liệu để phân tích. Quy trình này được thực hiện bằng cách loại bỏ các từ dừng, từ gốc và gắn thẻ POS. Ở cấp độ tiếp theo, các kỹ thuật thiết kế tính năng, ví dụ: TF_IDF và bigram được áp dụng để soạn các đối tượng địa lý. Vì mức này tạo ra nhiều tính năng, PCA được sử dụng để loại bỏ các tính năng không cần thiết và giảm không gian tính năng. Cuối cùng, để phân biệt giữa các bài đánh giá spam và không spam, một bộ phân loại được sử dụng

trong đó sử dụng Information Gain để xếp hạng các tính năng và phát hiện các bài đánh giá lừa đảo. Kết quả RLOSD được so sánh bởi các bộ phân loại khác và nó cho thấy rằng mô hình đạt được hiệu suất cao hơn.

"A Survey on Representation Learning Efforts in Cybersecurity Domain" [12]

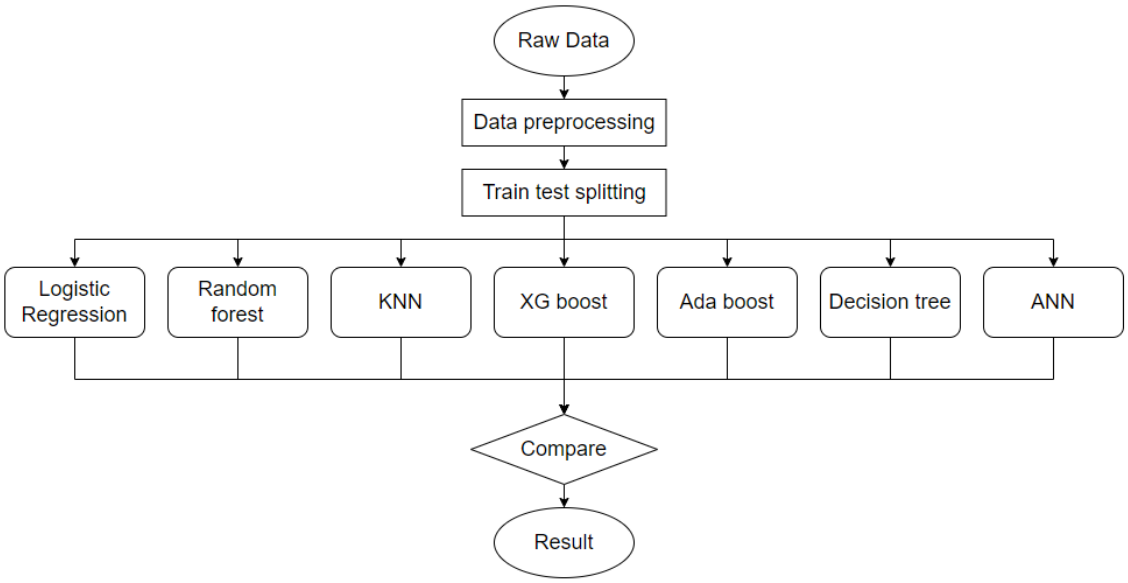
Trong thời đại dựa trên công nghệ này, các hệ thống dựa trên mạng đang phải đối mặt với các cuộc tấn công mạng mới hàng ngày. Các phương pháp tiếp cận an ninh mạng truyền thống dựa trên cơ sở dữ liệu kiến thức về mối đe dọa cũ và cần được cập nhật hàng ngày để chống lại thế hệ mối đe dọa mạng mới và bảo vệ các hệ thống dựa trên mạng cơ bản. Cùng với việc cập nhật cơ sở dữ liệu kiến thức về mối đe dọa, cần có sự quản lý và xử lý thích hợp đối với dữ liệu được tạo ra bởi các ứng dụng thời gian thực nhạy cảm. Trong những năm gần đây, các nền tảng máy tính khác nhau dựa trên các thuật toán RL đã nổi lên như một nguồn tài nguyên hữu ích để quản lý và khai thác dữ liệu được tạo ra để trích xuất thông tin có ý nghĩa. Trong các tác vụ học máy, việc phân loại dữ liệu yêu cầu dữ liệu đầu vào thuận tiện về mặt toán học và tính toán. Tuy nhiên, dữ liệu đa phương tiện và phi đa phương tiện theo thời gian thực, ví dụ: hình ảnh, âm thanh, video và dữ liệu cảm biến, không xác định các tính năng cụ thể. Các kỹ thuật RL cung cấp một giải pháp thay thế bằng cách khám phá các tính năng hoặc cách biểu diễn thông qua việc kiểm tra mà không cần dựa vào các thuật toán rõ ràng. Bài báo đã thảo luận về các cuộc tấn công mạng khác nhau và các sáng kiến được thực hiện bởi các khu tổ chức quốc tế. Trong lĩnh vực an ninh mạng, các ứng dụng thời gian thực xử lý cả dữ liệu đa phương tiện và dữ liệu phi đa phương tiện. Để xử lý dữ liệu được tạo ra bởi các ứng dụng thời gian thực khác nhau, tác giả đã cung cấp tổng quan chuyên sâu về các nền tảng máy tính học biểu diễn khác nhau. Các nền tảng máy tính này được giới thiệu bởi các nhà cung cấp nổi tiếng, ví dụ: IBM, Microsoft, Google, Amazon và Big ML. Bài báo cũng đã thảo luận về các bộ dữ liệu khác nhau có thể được sử dụng bằng các nhịp điệu thuật toán học biểu diễn trong miền an ninh mạng. Sau đó, thảo luận và tóm tắt những nỗ lực gần đây dành cho các hệ thống an ninh

mạng bằng cách sử dụng các thuật toán học đại diện. Những nỗ lực này được phân thành ba loại lớn, tức là kiến trúc có giám sát, không giám sát và sâu. Cuối cùng, nêu bật các hạn chế khác nhau trong các bộ dữ liệu có sẵn và các kỹ thuật dựa trên việc học biểu diễn hiện có. Mục đích chính của việc nêu bật những hạn chế là để nói với các nhà nghiên cứu và chuyên gia phát triển rằng vẫn còn nhiều thách thức nghiên cứu mở cần được giải quyết khi sử dụng các bộ dữ liệu và kỹ thuật có sẵn. Những hạn chế này cũng làm nổi bật nhiều lần đọc lại các nghiên cứu khác nhau trong tương lai. Các hướng nghiên cứu này nêu bật các dữ kiện khác nhau cần được xem xét để cải thiện các tính năng khác nhau của các kỹ thuật học biểu diễn sẵn có để làm cho chúng tương thích với các ứng dụng thời gian thực.

"Phishing website detection using machine learning and deep learning techniques" [27]

Tấn công Phishing dần trở nên phổ biến và lan rộng hơn, đặc biệt là trong bối cảnh số lượng các website ngày càng tăng, dẫn đến việc thông tin của người dùng có thể dễ dàng bị đánh cắp nếu không nâng cao cảnh giác hoặc bảo mật cẩn thận. Nhận thấy những nguy hiểm đến từ Phishing, nhóm các nhà khoa học gồm Selvakumari M và các cộng sự của mình đã thực hiện nghiên cứu nhằm so sánh các thuật toán ML và kĩ thuật DL trong việc phát hiện ra các website được tạo với mục đích Phishing.

Nhóm tác giả của công trình nghiên cứu đã tiến hành lấy bộ dữ liệu được sử dụng rộng rãi trên mạng như Kaggle và kết hợp với bộ dữ liệu mà nhóm tự xây dựng. Có 20% tập dữ liệu Phishing từ Kaggle được sử dụng làm tập Test và 80% tập dữ liệu còn lại sử dụng cho tập Train cho mô hình. Sau đó, bộ dữ liệu sẽ trải qua quá trình tiền xử lí bao gồm nhiều kĩ thuật trích xuất đặc trưng như Feature Extraction, Instance Selection, Normalization,… Đây là quá trình có thể ảnh hưởng đến độ chính xác và kết quả cuối cùng, vì thế quá trình này cần phải xử lý và loại bỏ các đặc trưng cũng như dữ liệu không cần thiết cho quá trình thử nghiệm.



Hình 1.8: Lưu đồ mô tả quy trình

Quá trình thử nghiệm và phân tích sẽ áp dụng các thuật toán ML và DL như KNN, Decision Tree, Random Forest,… để so sánh và kết luận thuật toán tốt nhất cho mô hình nói riêng và công trình nghiên cứu nói chung.

Bảng 1.1: Bảng so sánh các thuật toán

S.No	Algorithm	Traning set accuracy	Testing set accuracy	Precision score
1	Logistic Regression	79.00	79.00	82.30
2	KNN	96.70	93.10	93.84
3	Decision Tree	100	95.50	96.13
4	Random Forest	95.00	94.40	94.81
5	XG Boost	93.80	93.40	93.92
6	Ada Boost	87.00	86.90	84.71

“Learning Representations for Log Data in Cybersecurity” [28]

Vào năm 2017, Ignacio Arnaldo cùng các cộng sự của mình đã thực hiện nghiên cứu về các phương pháp phòng chống tấn công mạng. Trong nghiên cứu này nhóm tác giả giới thiệu một framework để khám phá và học các đặc trưng của

dữ liệu nhật ký được tạo bởi các thiết bị bảo mật cấp doanh nghiệp với mục tiêu phát hiện các mối đe dọa nguy hiểm (Advanced Persistent Threats - APT) kéo dài trong vài tuần. Framework được trình bày sử dụng chiến lược "chia để trị" kết hợp phân tích hành vi, mô hình chuỗi thời gian và thuật toán học đặc trưng để mô hình hóa khối lượng lớn dữ liệu. Ngoài ra, nhóm tác giả có quyền truy cập vào các tính năng do con người thiết kế, điều này hỗ trợ cho việc phân tích khả năng của một loạt các thuật toán học đặc trưng để bổ sung cho các tính năng do con người thiết kế theo nhiều cách tiếp cận phân loại. Nhóm tác giả chứng minh cách tiếp cận với một tập dữ liệu mới được trích xuất từ 3 tỷ dòng nhật ký được tạo ra tại ranh giới mạng doanh nghiệp với báo cáo chỉ huy và điều khiển thông tin liên lạc. Các kết quả được trình bày đã xác nhận phương pháp tiếp cận của nhóm tác giả đạt được diện tích theo đường cong ROC là 0:943 và 95 dương tính thực trong số 100 trường hợp được xếp hạng hàng đầu trên tập dữ liệu thử nghiệm.

"Representation Learning in Graphs for Credit Card Fraud Detection" [19]

Trong một nghiên cứu về kĩ thuật RL vào năm 2020, Rafael Van Belle cùng các cộng sự của mình đã chứng minh được RL có tính hữu ích cho nhiều tác vụ dự đoán. Nhóm tác giả đã đánh giá tính khả thi của việc học đặc trưng trong bối cảnh gian lận thẻ tín dụng. Phân tích dữ liệu đã thành công trong việc dự đoán gian lận trong nghiên cứu trước đây. Tuy nhiên, lĩnh vực nghiên cứu đã tập trung vào các kỹ thuật nhằm tiết chế các tính năng được thực hiện bằng tay một cách nhàm chán và tốn kém. Ngoài ra, các công trình hiện tại thường bỏ qua thông tin liên quan đến mạng lưới giao dịch. Học đặc trưng trong đồ thị giải quyết được cả hai thách thức này.

Đầu tiên, nó cung cấp khả năng khai thác các khía cạnh quan hệ và cấu trúc của mạng giao dịch và tận dụng chúng trong một mô hình dự đoán. Thứ hai, nó làm mịn biểu đồ mà không cần kỹ thuật tính năng thủ công. Công trình này đóng góp cho tài liệu bằng cách là người đầu tiên chỉ ra một cách rõ ràng và rộng rãi cách mô hình phát hiện gian lận có thể có lợi từ việc học đại diện. Nhóm tác giả phân biệt

ba cách tiếp cận khác nhau trong bài báo này: phương pháp phát triển các đặc trưng trên mạng truyền thống, thuật toán học đặc trưng quy nạp và chuyển đổi đặc trưng. Thông qua đánh giá thử nghiệm rộng rãi trên tập dữ liệu trong thế giới thực, nhóm tác giả cho thấy việc học đặc trưng hiện đại trong đồ thị tốt hơn so với phương pháp đặc trưng hóa đồ thị truyền thống (Traditional Graph Featurization).

"PhishTrim: Fast and adaptive phishing detection based on deep representation learning" [8]

Một nghiên cứu về phòng chống tấn công Phishing vào năm 2020 được thực hiện bởi nhóm nghiên cứu đến từ Trung Quốc gồm Lei Zhang và Peng Zhang. Nhóm tác giả đã đề xuất phương pháp PhishTrim - một phương pháp phát hiện URL lừa đảo có mức độ nguy hiểm nhẹ dựa trên học sâu các đặc trưng, phương pháp được nhóm tác giả nhận định là nhanh và thích ứng tốt.

Nhóm tác giả lấy các đặc trưng nhúng ban đầu từ các URL thông qua mô hình được huấn luyện trước đó là Skip-gram. Sau đó, bộ nhớ ngắn hạn hai chiều (Bi-LSTM) được sử dụng để trích xuất sự phụ thuộc vào ngữ cảnh để tìm hiểu thêm về đặc trưng sâu của URL. Các đặc tính n-gram cục bộ được trích xuất bằng cách sử dụng CNN. Các thử nghiệm cho thấy PhishTrim hoạt động tốt hơn trên các bộ dữ liệu quy mô lớn với độ chính xác 99,797% và chỉ ra rằng phương pháp của nhóm tác giả có khả năng nhất định để phát hiện các cuộc tấn công lừa đảo zero-day.

"Intelligent phishing detection scheme using deep learning algorithms" [29]

Nghiên cứu này tập trung vào việc thiết kế và phát triển một giải pháp phát hiện lừa đảo dựa trên học sâu dựa trên công cụ định vị tài nguyên phổ quát và nội dung trang web như hình ảnh, văn bản và khung. Trong nghiên cứu này, CNN và LSTM đã được sử dụng để xây dựng một mô hình phân loại kết hợp có tên là hệ thống phát hiện lừa đảo thông minh (IPDS). Để xây dựng mô hình đề xuất, bộ phân loại CNN và LSTM đã được đào tạo bằng cách sử dụng bộ định vị tài nguyên phổ quát 1m và hơn 10.000 hình ảnh. Sau đó, độ nhạy của mô hình đề xuất được xác định bằng cách xem xét các yếu tố khác nhau như loại đối tượng địa lý, số lượng

phân loại sai và các vấn đề phân chia. Kết quả - Một phân tích thử nghiệm mở rộng đã được thực hiện để đánh giá và so sánh hiệu quả của IPDS trong việc phát hiện các trang web lừa đảo và các cuộc tấn công lừa đảo khi áp dụng cho các tập dữ liệu lớn. Kết quả cho thấy, mô hình đạt tỷ lệ chính xác 93,28% và thời gian phát hiện trung bình là 25 giấy. Phương pháp kết hợp sử dụng thuật toán học sâu của cả phương pháp CNN và LSTM đã được sử dụng trong công trình nghiên cứu này. Một mặt, sự kết hợp của cả CNN và LSTM được sử dụng để giải quyết vấn đề của tập dữ liệu lớn và hiệu suất dự đoán của bộ phân loại cao hơn. Do đó, việc kết hợp hai phương pháp dẫn đến kết quả tốt hơn với ít thời gian đào tạo hơn đối với kiến trúc LSTM và CNN, đồng thời sử dụng các tính năng hình ảnh, khung và văn bản như một kết hợp để phát hiện mô hình của chúng tôi. Các tính năng kết hợp và trình phân loại IPDS để phát hiện lừa đảo là điểm mới của nghiên cứu này theo hiểu biết tốt nhất của các tác giả.

“Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning” [30]

Lừa đảo đã trở thành một trong những mối đe dọa mạng lớn nhất và hiệu quả nhất, gây ra thiệt hại hàng trăm triệu đô la và hàng triệu vụ vi phạm dữ liệu mỗi năm. Hiện tại, các kỹ thuật chống lừa đảo yêu cầu các chuyên gia trích xuất các tính năng của các trang lừa đảo và sử dụng các dịch vụ của bên thứ ba để phát hiện các trang lừa đảo. Các kỹ thuật này có một số hạn chế, một trong số đó là việc trích xuất các tính năng lừa đảo đòi hỏi chuyên môn và tốn nhiều thời gian. Thứ hai, việc sử dụng các dịch vụ của bên thứ ba làm chậm việc phát hiện các trang web lừa đảo. Do đó, bài báo này đề xuất một phương pháp phát hiện trang web lừa đảo tích hợp dựa trên CNN và rừng ngẫu nhiên RF. Phương pháp này có thể dự đoán tính hợp pháp của các URL mà không cần truy cập nội dung web hoặc sử dụng dịch vụ của bên thứ ba. Kỹ thuật được đề xuất sử dụng kỹ thuật nhúng ký tự để chuyển đổi URL thành ma trận kích thước cố định, trích xuất các tính năng ở các cấp độ khác nhau bằng mô hình CNN, phân loại các đối tượng địa lý nhiều cấp độ bằng cách sử dụng nhiều bộ phân loại RF và cuối cùng, xuất ra kết quả dự đoán bằng cách sử

dụng phương pháp lấy tất cả . Trên tập dữ liệu, tỷ lệ chính xác 99,35% đã đạt được bằng cách sử dụng mô hình được đề xuất. Tỷ lệ chính xác đạt được là 99,26% trên dữ liệu điểm chuẩn, cao hơn nhiều so với tỷ lệ của mô hình cực đoan hiện có.

"Applying Machine Learning Techniques to Detect and Analyze Web Phishing Attacks" [31]

Trong bài báo này, nhóm tác giả đề xuất một phương pháp có thể phân biệt giữa các trang web nhằm mục đích thực hiện các cuộc tấn công lừa đảo và các trang web hợp pháp. Gần đây, một cách tiếp cận hiệu quả hơn để chống lại lừa đảo dựa trên các kỹ thuật máy học đã xuất hiện. Trong cách tiếp cận này, các mô hình được trích xuất bằng kỹ thuật ML được sử dụng để phân loại các trang web là hợp pháp hoặc lừa đảo, dựa trên các tính năng nhất định. Đề xuất một phương pháp dựa trên máy học có thể phát hiện xem một trang web có biểu hiện các cuộc tấn công đánh bóng hay không. Phương pháp được đề xuất dựa trên một vector đặc trưng dễ thu thập mà không cần tính toán thêm. Các thuật toán phân loại thu được độ chính xác tốt nhất là J48 và RepTree, nhưng khi xem xét chỉ số thu hồi, thu hồi RepTree thấp hơn nếu so sánh với thuật toán thu được bởi thuật toán phân loại J48: đây là lý do tại sao các tác giả xác nhận thuật toán J48 là cái có được hiệu suất tốt nhất về độ chính xác và khả năng thu hồi để phát hiện các cuộc tấn công lừa đảo trên web. Trên thực tế, các thuật toán còn lại (tức là HoeffdingTree, Random Forest, LMT và DecisionStump) thể hiện hiệu suất thấp hơn J48 và RepTree về độ chính xác và thu hồi. Cụ thể, độ chính xác bằng 0,923 và độ thu hồi bằng 0,916 khi phát hiện tấn công bằng cách sử dụng thuật toán J48.

"A lightweight data representation for phishing URLs detection in IoT environments" [32]

Lừa đảo là một cuộc tấn công mạng khai thác sự thiếu hiểu biết hoặc ngây thơ về kỹ thuật của nạn nhân và thường liên quan đến Bộ định vị tài nguyên thống nhất (URL). Do đó, việc phát hiện một cuộc tấn công lừa đảo bằng cách phân tích các URL trước khi truy cập chúng là một điều thuận lợi. Với sự phát triển của Internet of Things (IoT), các cuộc tấn công lừa đảo đang chuyển sang lĩnh vực này

do số lượng thiết bị IoT và lượng thông tin cá nhân mà chúng xử lý. Mặc dù một số phương pháp tiếp cận đã được đề xuất để phát hiện các cuộc tấn công lừa đảo, nhưng các phương pháp Học máy dựa trên URL thu được kết quả hiệu suất tốt hơn, nhưng tất cả chúng đều phụ thuộc vào bộ tính năng được sử dụng. Ngược lại, chỉ một số công trình về việc chọn bộ tính năng phù hợp nhất để cải thiện quy trình phát hiện lừa đảo đã được công bố. Nghiên cứu hiện tại khám phá cách có được bộ tính năng nâng cao đáng kể tỷ lệ phát hiện lừa đảo trong môi trường IoT. Do đó, một thuật toán lựa chọn tính năng đã được thông qua và mở rộng để có được bộ tính năng tiêu biểu nhất. Khi Random Forest được sử dụng với biểu diễn dữ liệu được đề xuất, tỷ lệ phát hiện ra các cuộc tấn công URL lừa đảo là 99,57%.

CHƯƠNG 2. XÂY DỰNG MÔ HÌNH PHÁT HIỆN TẤN CÔNG PHISHING

2.1. Thiết kế mô hình

Trong luận văn này, với tính chất của các url của cách tấn công phishing, sử dụng tokenization để chuyển thành ma trận số dựa vào xử lý ngôn ngữ tự nhiên các url. Từ đó, chuyển ma trận url này thành ma trận hình ảnh grayscale và áp dụng ResNet18 để training và xây dựng mô hình nhận diện Phishing.

Với ý tưởng này, luận văn đề xuất như xây dựng mô hình như sau:

(1) Url → Tokenization → Text_to_matrix → numpy Matrix

(2) Numpy Matrix → convert to Image Matrix (Gray scale) → Array of images

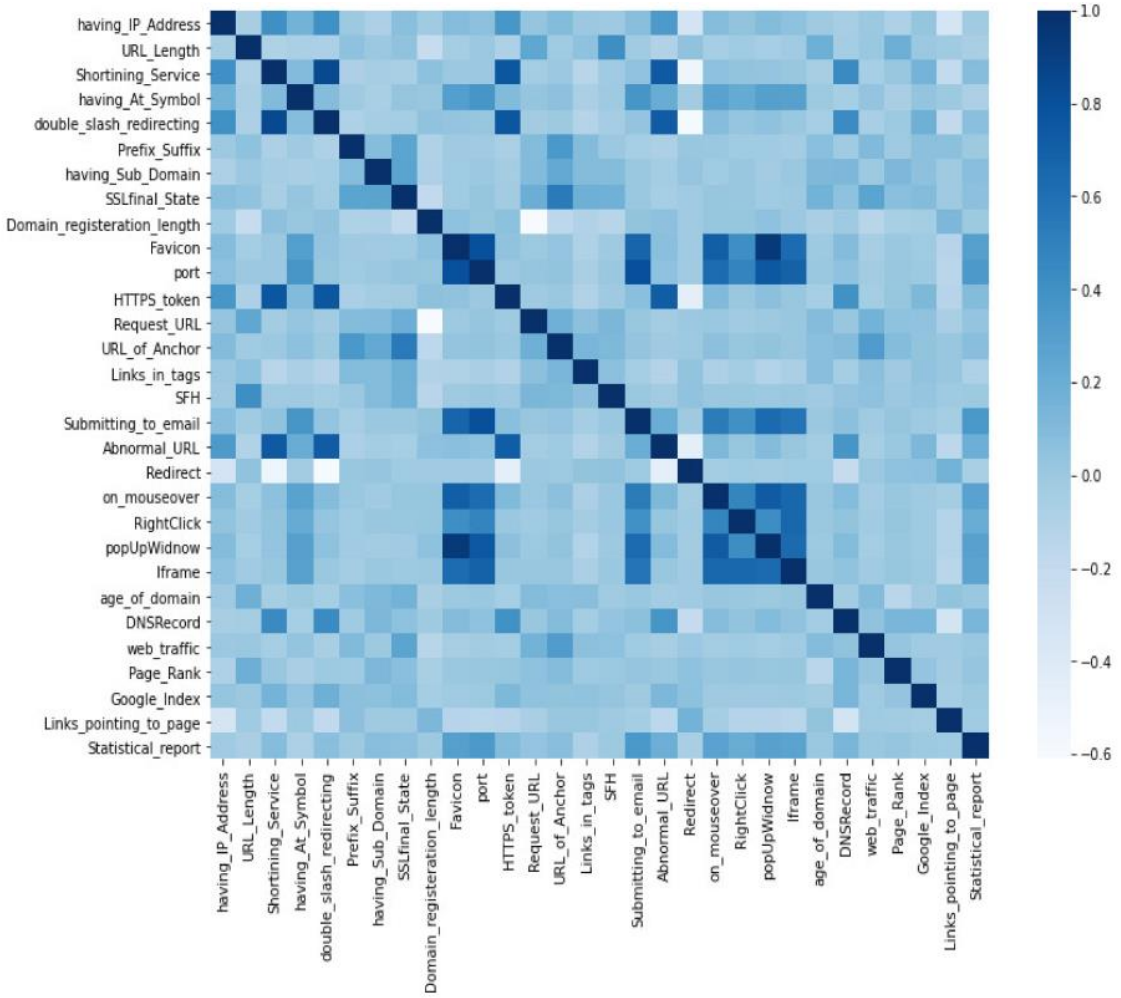
(3) Array of Images → training with ResNet → Model

Sử dụng các website features

Trong thử nghiệm, các tính năng của trang web được chuyển đổi thành các vectơ đặc trưng và được sử dụng làm đầu vào cho các mô hình DL. Bảng 4 cho thấy danh sách 30 tính năng được sử dụng trong nghiên cứu này. Mỗi tính năng có ba giá trị có thể có: −1, 0 và 1 (−1 là lừa đảo, 0 là đáng ngờ và 1 là lành tính). Tính năng cuối cùng, được đặt tên là "lớp", là phân loại của URL.

Biểu đồ Heatmap hay còn gọi là bản đồ nhiệt là cách thể hiện dữ liệu một cách trực quan thông qua màu sắc với 2 loại màu nóng – lạnh. Màu nóng là nơi có giá trị cao nhất, quan trọng nhất, hoặc mạnh nhất (dữ liệu càng tăng khi màu càng đi về phía gam màu nóng), màu lạnh sẽ mang ý nghĩa ngược lại. Hình 3.4 là bản đồ nhiệt hiển thị ma trận tương quan của các tính năng này. Phạm vi tương quan tiêu chuẩn là từ −1 đến +1, trong đó −1 là tương quan âm thấp nhất và +1 là tương quan dương cao nhất. Mối tương quan âm được hiển thị trong dải màu sáng hơn, trong khi mối tương quan dương được hiển thị trong dải màu tối hơn. Đặc biệt trong tập dữ liệu này, ánh xạ của hai đối tượng địa lý khác nhau, có tên Favicon và popUpWindow, cho thấy màu tối nhất, có nghĩa là

chúng có tương quan cao hoặc thuận. Tương quan tích cực có nghĩa là một đối tượng địa lý đánh dấu URL là lừa đảo và đối tượng địa lý khác cũng vậy. Trong khi các mối tương quan phủ định có nghĩa là một tính năng đánh dấu URL là độc hại, trong khi tính năng kia thì không [6].



Hình 2.1: Ma trận hệ số tương quan giữa các features [20]

Ta có thể hình dung cụ thể hơn thông qua Hình 2.3 như sau:

Type	No	Feature	Name	Description	Value
Address bar-based	1	IP address	UsingIP	Having IP address in URL	−1, 1
	2	URL length	LongURL	Long URL to hide the suspicious part	−1, 0, 1
	3	Shortening service	ShortURL	Using URL shortening services "TinyURL"	−1, 1
	4	@ Symbol	Symbol@	URL's having @ symbol	−1, 1
	5	"//" redirecting	Redirecting//	Having "//" within URL path for directing	−1, 1
	6	Prefix suffix	PrefixSuffix	Adding prefix or suffix separated by (-) to the domain	−1, 1
	7	Sub domain	SubDomains	Sub domain and multi sub domain	−1, 0, 1
	8	SSL final state	HTTPS	Existence of HTTPS and validity of the certificate	−1, 0, 1
	9	Domain registration	DomainRegLen	Expiry date of domains/Domain registration length	−1, 1
	10	Favicon	Favicon	Favicon loaded from a domain	−1, 1
	11	Port	NonStdPort	Using non-standard port	−1, 1
	12	HTTPS token	HTTPSDomainURL	The existence of HTTPS token in the domain part of URL	−1, 1
Abnormal-based	13	Request URL	RequestURL	Request URL within a webpage/Abnormal request	−1, 1
	14	URL of anchor	AnchorURL	URL within <a> tag/Abnormal anchor	−1, 0, 1
	15	Links in tags	LinksInScriptTags	Links in <Meta>, <Script> and <Link> tags	−1, 0, 1
	16	SFH	ServerFormHandler	Server Form Handler	−1, 0, 1
	17	Email	InfoEmail	Submitting information to E-mail	−1, 1
	18	Abnormal URL	AbnormalURL	Host name is included in the URL/Whois	−1, 1
HTML and JavaScript-based	19	Redirecting	WebsiteForwarding	Number of times a website has been redirected	0, 1
	20	On mouseover	StatusBarCust	On mouse over changes status bar/Status bar customization	−1, 1
	21	Right click	DisableRightClick	Disabling right click	−1, 1
	22	Pop-up window	UsingPopupWindow	Using Pop-up window	−1, 1
	23	Iframe redirection	IframeRedirection	Using Iframe	−1, 1
Domain-based	24	Age of domain	AgeofDomain	Minimum age of a legitimate domain is 6 months	−1, 1
	25	DNS record	DNSRecording	Existence of DNS record for the domain	−1, 1
	26	Website traffic	WebsiteTraffic	Being among top 100,000 in Alexa rank	−1, 0, 1
	27	Page rank	PageRank	Having a page rank greater than 0.2	−1, 1
	28	Google index	GoogleIndex	Website indexed by Google	−1, 1
	29	Link reference	LinksPoitingToPage	Number of links pointing to a page	−1, 0, 1
	30	Statistical report	StatsReport	Top 10 domain and top 10 Ips from PhishTank	−1, 1
		Result	class	Phishing or legitimate	−1, 1

Hình 2.2: Mô tả mối tương quan giữa các đặc tính trong ma trận

- Với ý tưởng phát triển mô hình làm việc như bộ phân loại. Mô hình nhận dữ liệu chuỗi URL từ đó, chỉ ra chính xác dữ liệu tấn công phishing hay không.

- Ý tưởng ở đây là: Hiện nay các mô hình representation learning sử dụng mạng neural học sâu đã cho kết quả rất khả quan trong lĩnh vực thị giác máy tính, áp dụng thành công trong phân loại ảnh và video. Áp dụng các mô hình này vào lĩnh vực an toàn thông tin mà cụ thể là phát hiện tấn công phishing để cải thiện khả năng phát hiện dạng tấn công này. Tuy nhiên dữ liệu tấn công phishing khác với dữ liệu multimedia trên các mô hình thị giác máy tính. Cần có phương pháp tạo sự tương thích để có thể áp dụng được. Chính vì vậy, từ các công trình nghiên cứu và đã công bố, luận văn này đưa ra ý tưởng biến dữ liệu URL của phishing thành dữ liệu hình ảnh thông qua các tính chất của chuỗi URL. Từ đó, phân lớp bằng representation learning, xây dựng mô hình phù hợp để nhận diện tấn công phishing.

Một trong các mô hình representation learning đơn giản là RestNet18, và cách biến đổi URL thành ma trận thông qua Tokenizer của TensorFlow. Từ 2 kỹ thuật phổ biến này, luận văn tích hợp và xây dựng mô hình phát hiện tấn công Phishing.

2.1.1. Giới thiệu về ResNet

ResNet (Residual Network) được giới thiệu đến công chúng vào năm 2015 và thậm chí đã giành được vị trí thứ 1 trong cuộc thi ILSVRC 2015 với tỉ lệ lỗi top 5 chỉ 3.57%. Không những thế nó còn đứng vị trí đầu tiên trong cuộc thi ILSVRC and COCO 2015 với ImageNet Detection, ImageNet localization, Coco detection và Coco segmentation.Hiện tại thì có rất nhiều biến thể của kiến trúc ResNet với số lớp khác nhau như ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152,...Với tên là ResNet theo sau là một số chỉ kiến trúc ResNet với số lớp nhất định.

Tại sao lại xuất hiện mạng ResNet

Mạng ResNet (R) là một mạng CNN được thiết kế để làm việc với hàng trăm lớp. Một vấn đề xảy ra khi xây dựng mạng CNN với nhiều lớp chập sẽ xảy ra hiện tượng Vanishing Gradient dẫn tới quá trình học tập không tốt.

Vanishing Gradient

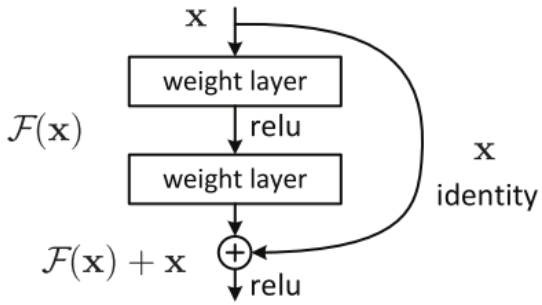
Trước hết thì Backpropagation Algorithm là một kỹ thuật thường được sử dụng trong quá trình tranining. Ý tưởng chung của thuật toán là sẽ đi từ output layer đến input layer và tính toán gradient của cost function tương ứng cho từng parameter (weight) của mạng. Gradient Descent sau đó được sử dụng để cập nhật các parameter đó.

Toàn bộ quá trình trên sẽ được lặp đi lặp lại cho tới khi mà các parameter của network được hội tụ. Thông thường chúng ta sẽ có một hyperparametr (số Epoch - số lần mà traninig set được duyệt qua một lần và weights được cập nhật) định nghĩa cho số lượng vòng lặp để thực hiện quá trình này. Nếu số lượng vòng lặp quá nhỏ thì ta gặp phải trường hợp mạng có thể sẽ không cho ra kết quả tốt và ngược lại thời gian tranining sẽ lâu nếu số lượng vòng lặp quá lớn.

Tuy nhiên, trong thực tế Gradients thường sẽ có giá trị nhỏ dần khi đi xuống các layer thấp hơn. Dẫn đến kết quả là các cập nhật thực hiện bởi Gradients Descent không làm thay đổi nhiều weights của các layer đó và làm chúng không thể hội tụ và mạng sẽ không thu được kết quả tốt. Hiện tượng như vậy gọi là Vanishing Gradients.

Kiến trúc mạng ResNet

Cho nên giải pháp mà ResNet đưa ra là sử dụng kết nối "tắt" đồng nhất để xuyên qua một hay nhiều lớp. Một khối như vậy được gọi là một Residual Block, như trong hình sau :



Hình 2.3: Residual learning: a building block.

ResNet gần như tương tự với các mạng gồm có convolution, pooling, activation và fully-connected layer. Ảnh bên trên hiển thị khối dư được sử dụng trong mạng. Xuất hiện một mũi tên cong xuất phát từ đầu và kết thúc tại cuối khối dư. Hay nói cách khác là sẽ bổ sung Input X vào đầu ra của layer, hay chính là phép cộng mà ta thấy trong hình minh họa, việc này sẽ chống lại việc đạo hàm bằng 0, do vẫn còn cộng thêm X. Với H(x) là giá trị dự đoán, F(x) là giá trị thật (nhãn), chúng ta muốn H(x) bằng hoặc xấp xỉ F(x).

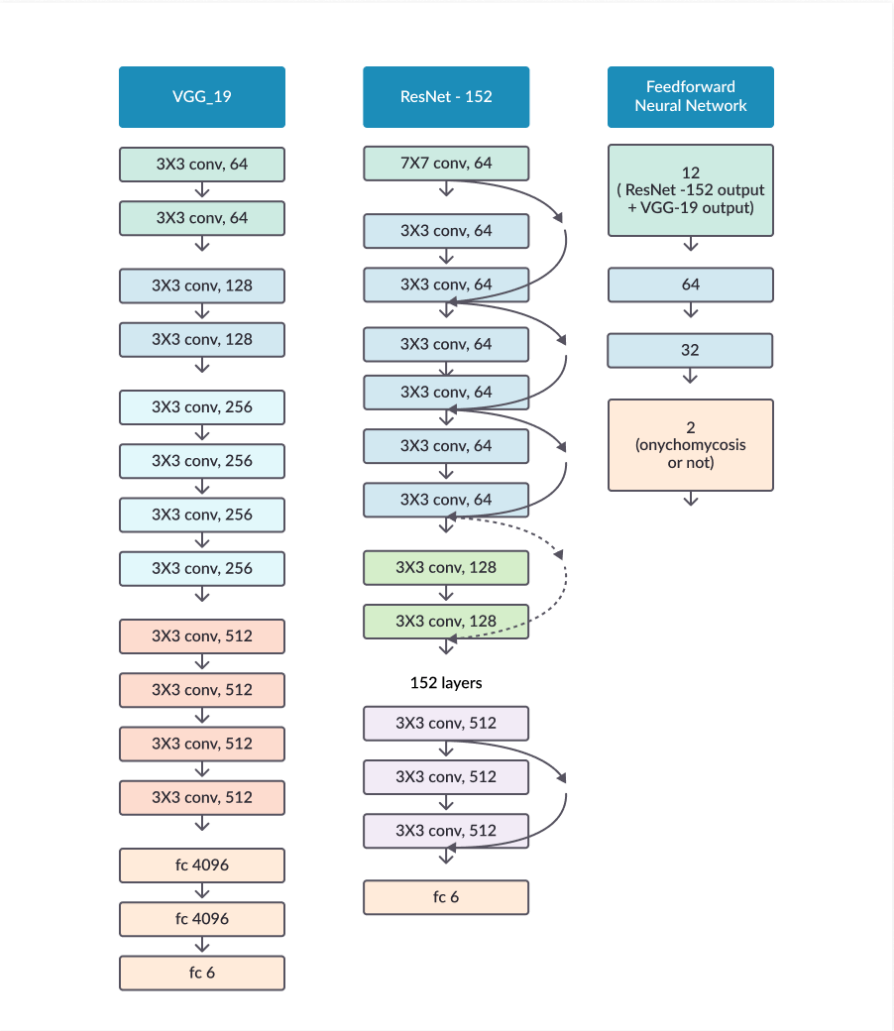
Việc F(x) có được từ x như sau:

X->weight1-> ReLU -> weight2

Giá trị H(x) có được bằng cách:

F(x) + x -> ReLU

Như chúng ta đã biết việc tăng số lượng các lớp trong mạng làm giảm độ chính xác, nhưng muốn có một kiến trúc mạng sâu hơn có thể hoạt động tốt.



Hình 2.4: ResNet sử dụng các kết nối tắt (kết nối trực tiếp đầu vào của lớp (n) với (n+x) được hiển thị dạng mũi tên cong. Qua mô hình nó chứng minh được có thể cải thiện hiệu suất trong quá trình training model khi mô hình có hơn 20 lớp.

2.1.2. Tokenization

Xử lý ngôn ngữ tự nhiên (NLP) là một nhánh của Trí tuệ nhân tạo (AI) cung cấp cho máy tính khả năng hiểu ngôn ngữ viết và nói của con người. Dễ dàng kể đến một số ứng dụng của NLP trong kiểm tra chính tả, tự động điền, phát hiện thư rác, trợ lý ảo trên điện thoại và ô tô. Tuy nhiên, ít ai biết rằng máy móc hoạt động với các con số chứ không phải các chữ cái/từ/câu. Vì vậy, để làm việc với một lượng lớn dữ liệu văn bản có sẵn, tiền xử lý văn bản (text pre-processing) là quá

trình cần thiết giúp làm sạch văn bản. Bản thân tiền xử lý văn bản bao gồm nhiều giai đoạn, và một trong số đó là tách từ (hay còn gọi là Tokenization).

Tokenization (tách từ) là một trong những bước quan trọng nhất trong quá trình tiền xử lý văn bản. Cho dù bạn đang làm việc với các kỹ thuật NLP truyền thống hay sử dụng các kỹ thuật học sâu nâng cao thì vẫn không thể bỏ qua bước này. Nói một cách đơn giản, tokenization là quá trình tách một cụm từ, câu, đoạn văn, một hoặc nhiều tài liệu văn bản thành các đơn vị nhỏ hơn. Mỗi đơn vị nhỏ hơn này được gọi là Tokens.

Có thể coi tokens là các khối xây dựng của NLP và tất cả các mô hình NLP đều xử lý văn bản thô ở cấp độ các Tokens. Chúng được sử dụng để tạo từ vựng trong một kho ngữ liệu (một tập dữ liệu trong NLP). Từ vựng này sau đó được chuyển thành số (ID) và giúp chúng ta lập mô hình. Tokens có thể là bất cứ thứ gì – một từ (word), một từ phụ (sub-word) hoặc thậm chí là một ký tự (character). Các thuật toán khác nhau tuân theo các quy trình khác nhau trong việc thực hiện mã hóa và sự khác biệt giữa ba loại tokens này sẽ được chỉ ra dưới đây.

2.2. Bộ dữ liệu của bài toán

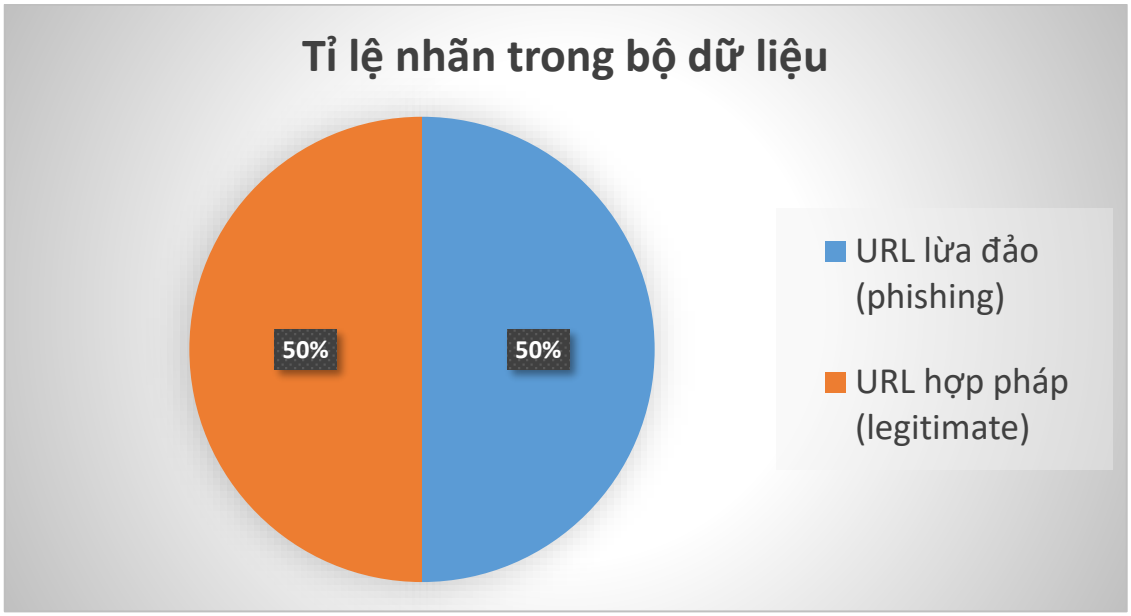
Luận văn sử dụng bộ dữ liệu Web page Phishing Detection [35]. Bộ dữ liệu bao gồm 11,430 dòng và 89 cột cung cấp 11,429 URL với 87 tính năng được trích xuất. Bộ dữ liệu được thiết kế để sử dụng làm chuẩn cho các hệ thống phát hiện lừa đảo dựa trên công nghệ máy học. Trong số 89 dữ liệu có chứa 1 cột là url mang địa chỉ của trang web, 1 cột là nhãn của dữ liệu và 87 tính năng. Các tính năng là từ ba lớp khác nhau: 56 tính năng được trích xuất từ cấu trúc và cú pháp của URL, 24 tính năng được trích xuất từ nội dung của các trang tương ứng của chúng và 7 tính năng được trích xuất từ các dịch vụ bên ngoài. Bộ dữ liệu được cân bằng, nó chứa chính xác 50% URL lừa đảo (phishing) và 50% URL hợp pháp (legitimate). Dựa vào tập dữ liệu, các tác giả đã cung cấp các tập lệnh Python có thể được sử dụng cho việc trích xuất các tính năng với mục đích nhân rộng hoặc mở rộng. Bộ dữ liệu được xây dựng vào tháng 5 năm 2020. Bảng 2.1 dưới đây liệt kê 89 thuộc tính và kiểu dữ liệu của từng thuộc tính trong bộ dữ liệu.

Bảng 2.1: Các thuộc tính của bộ dữ liệu

STT	Tên cột	Kiểu DL	STT	Tên cột	Kiểu DL
1	**url**	object	46	longest_words_raw	int
2	length_url	int	47	longest_word_host	int
3	length_hostname	int	48	longest_word_path	int
4	ip	int	49	avg_words_raw	float
5	nb_dots	int	50	avg_word_host	float
6	nb_hyphens	int	51	avg_word_path	float
7	nb_at	int	52	phish_hints	int
8	nb_qm	int	53	domain_in_brand	int
9	nb_and	int	54	brand_in_subdomain	int
10	nb_or	int	55	brand_in_path	int
11	nb_eq	int	56	suspecious_tld	int
12	nb_underscore	int	57	statistical_report	int
13	nb_tilde	int	58	nb_hyperlinks	int
14	nb_percent	int	59	ratio_intHyperlinks	float
15	nb_slash	int	60	ratio_extHyperlinks	float
16	nb_star	int	61	ratio_nullHyperlinks	int
17	nb_colon	int	62	nb_extCSS	int
18	nb_comma	int	63	ratio_intRedirection	int
19	nb_semicolumn	int	64	ratio_extRedirection	float
20	nb_dollar	int	65	ratio_intErrors	int
21	nb_space	int	66	ratio_extErrors	float
22	nb_www	int	67	login_form	int
23	nb_com	int	68	external_favicon	int
24	nb_dslash	int	69	links_in_tags	float
25	http_in_path	int	70	submit_email	int
26	https_token	int	71	ratio_intMedia	float
27	ratio_digits_url	float	72	ratio_extMedia	float
28	ratio_digits_host	float	73	sfh	int

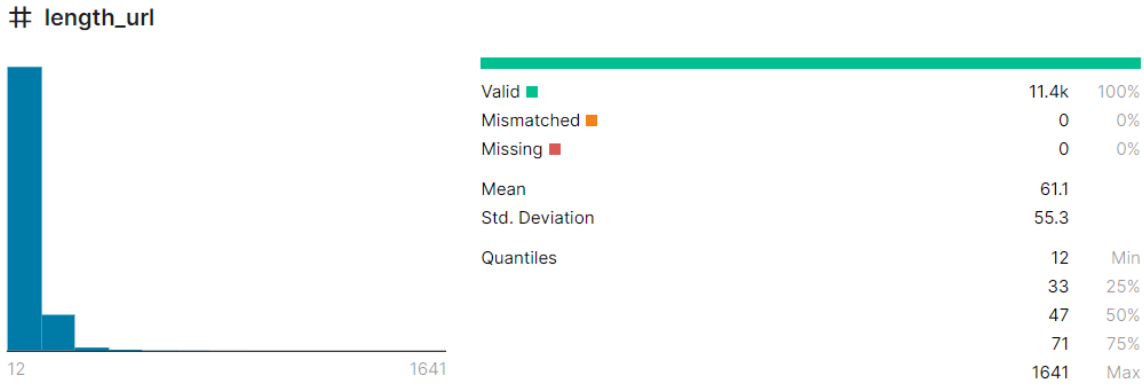
29	punycode	int	74	iframe	int
30	port	int	75	popup_window	int
31	tld_in_path	int	76	safe_anchor	float
32	tld_in_subdomain	int	77	onmouseover	int
33	abnormal_subdomain	int	78	right_clic	int
34	nb_subdomains	int	79	empty_title	int
35	prefix_suffix	int	80	domain_in_title	int
36	random_domain	int	81	domain_with_copyright	int
37	shortening_service	int	82	whois_registered_domain	int
38	path_extension	int	83	domain_registration_length	int
39	nb_redirection	int	84	domain_age	int
40	nb_external_redirection	int	85	web_traffic	int
41	length_words_raw	int	86	dns_record	int
42	char_repeat	int	87	google_index	int
43	shortest_words_raw	int	88	page_rank	int
44	shortest_word_host	int	89	**status**	object
45	shortest_word_path	int			

Trong tổng số 89 trường dữ liệu, trường url và status mang giá trị chuỗi, các trường còn lại mang kiểu dữ liệu số nguyên (chiếm đa số) hoặc số thực. Ngoài ra, qua thống kê cho thấy không có trường nào bị mất mát dữ liệu. Cột status chứa nhãn của bộ dữ liệu, chỉ tồn tại một trong hai giá trị “phishing” hoặc “legitimate”, tỉ lệ giữa số lượng URL lừa đảo và hợp pháp là cân bằng (Hình 2.6).

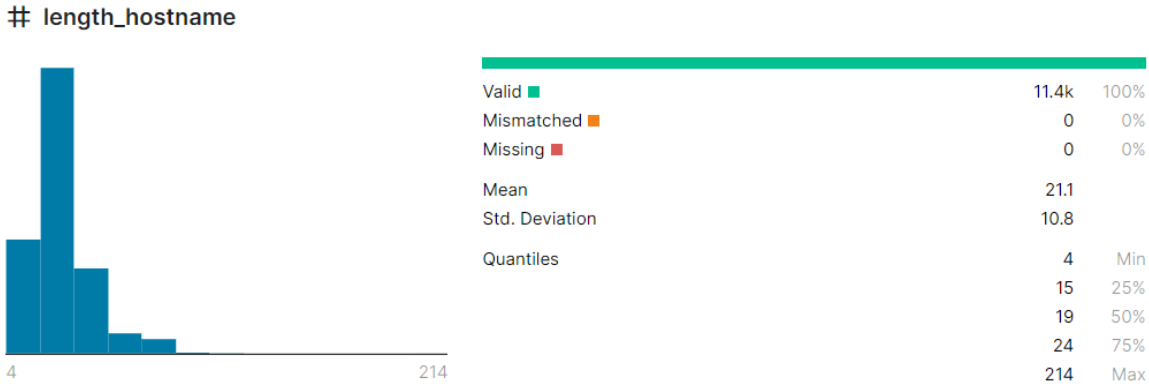


Hình 2.5: Tỉ lệ nhãn trong bộ dữ liệu

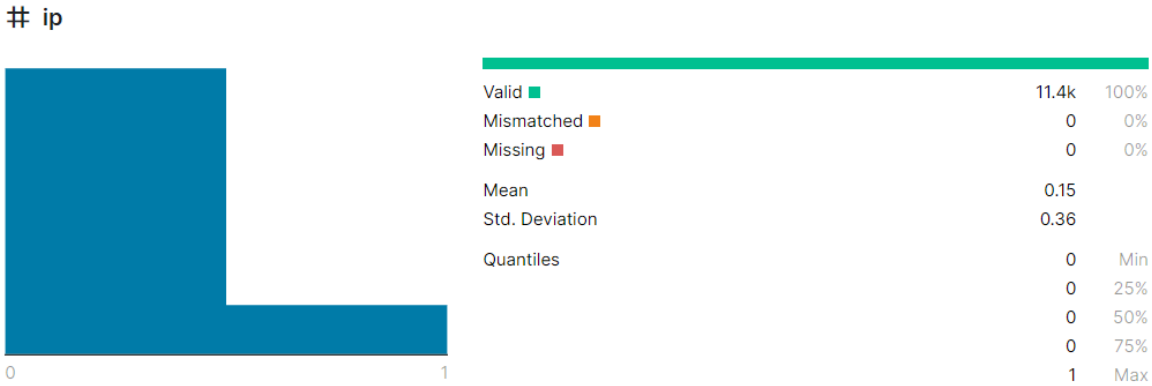
Để kiểm tra đặc tính của bộ dữ liệu, tiến hành quan sát một số thuộc tính như: length_url, leng_hostname, ip, nb_dots, nb_hyphens, nb_at, nb_qm, nb_and, nb_or. Sau khi quan sát thu được kết quả như các hình dưới đây (Hình 4.2 – 4.10):



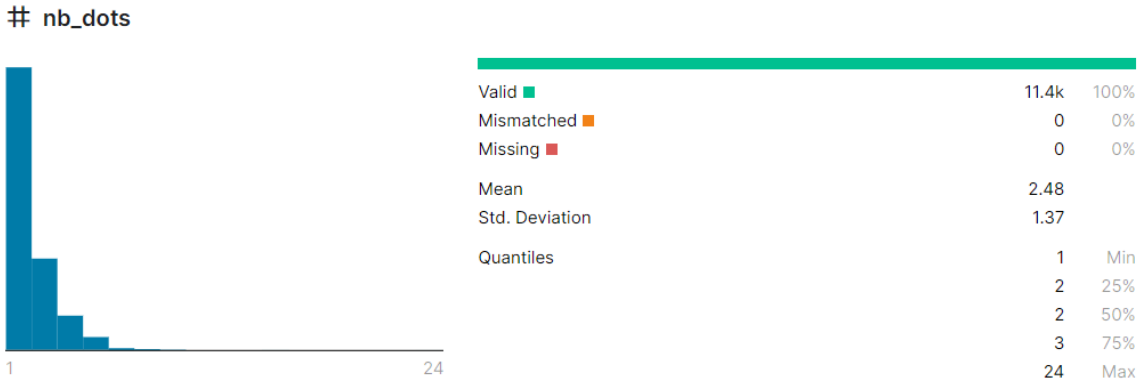
Hình 2.6: Thuộc tính length_url



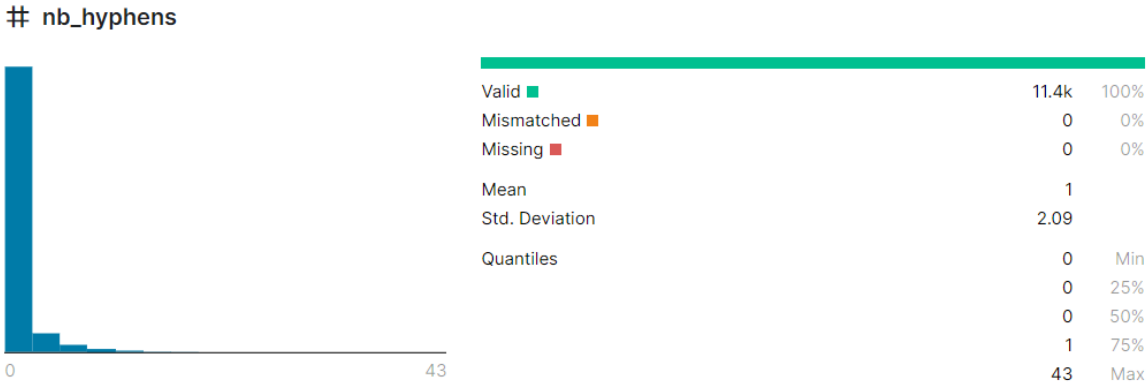
Hình 2.7: Thuộc tính length_hostname



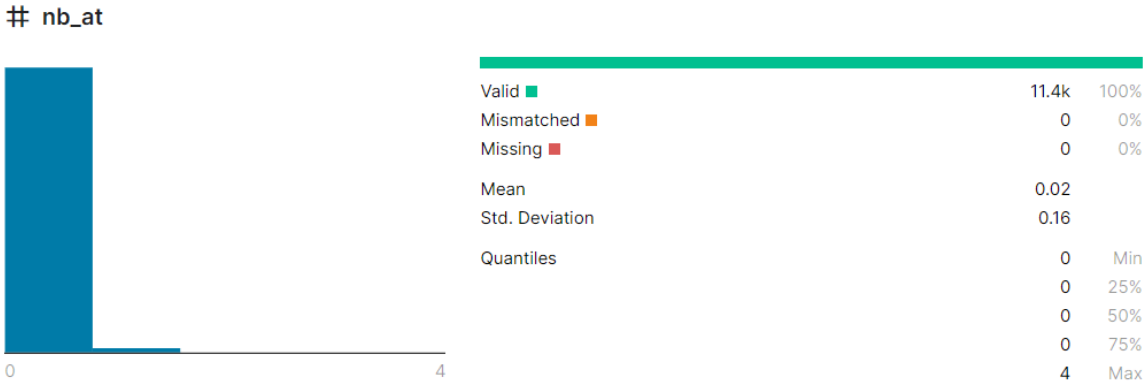
Hình 2.8: Thuộc tính ip



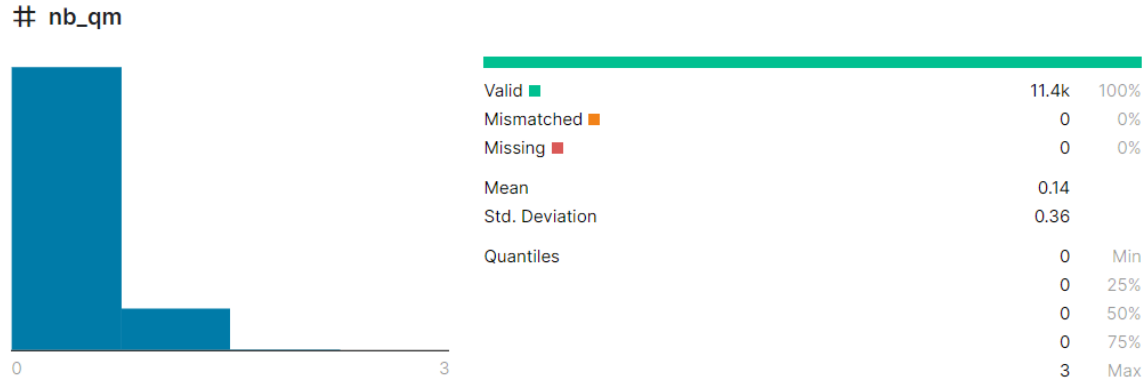
Hình 2.9: Thuộc tính nb_dots



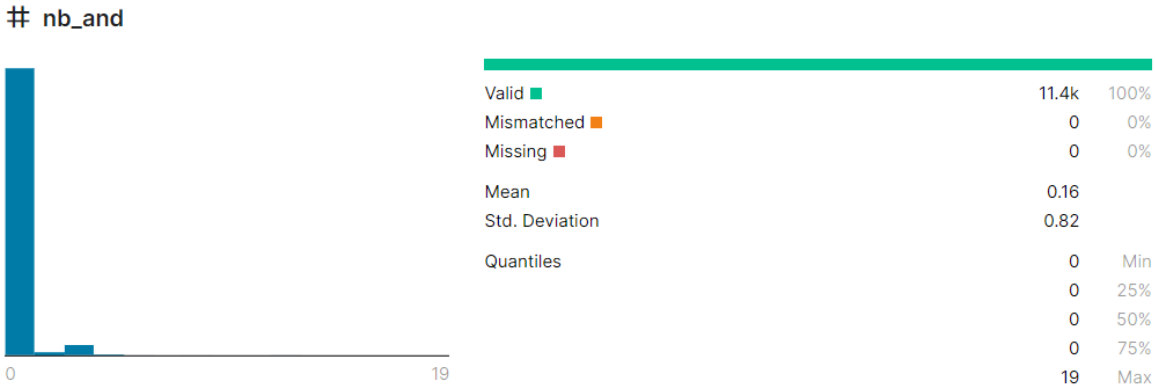
Hình 2.10: Thuộc tính nb_hyphens



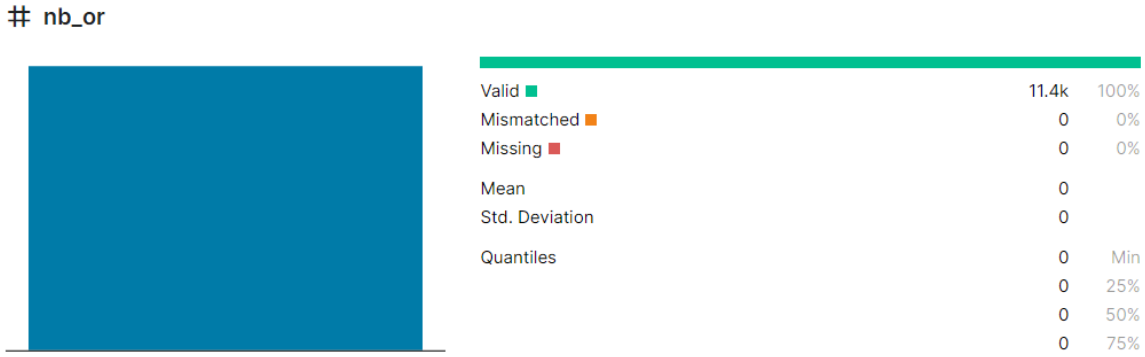
Hình 2.11: Thuộc tính nb_at



Hình 2.12: Thuộc tính nb_qm

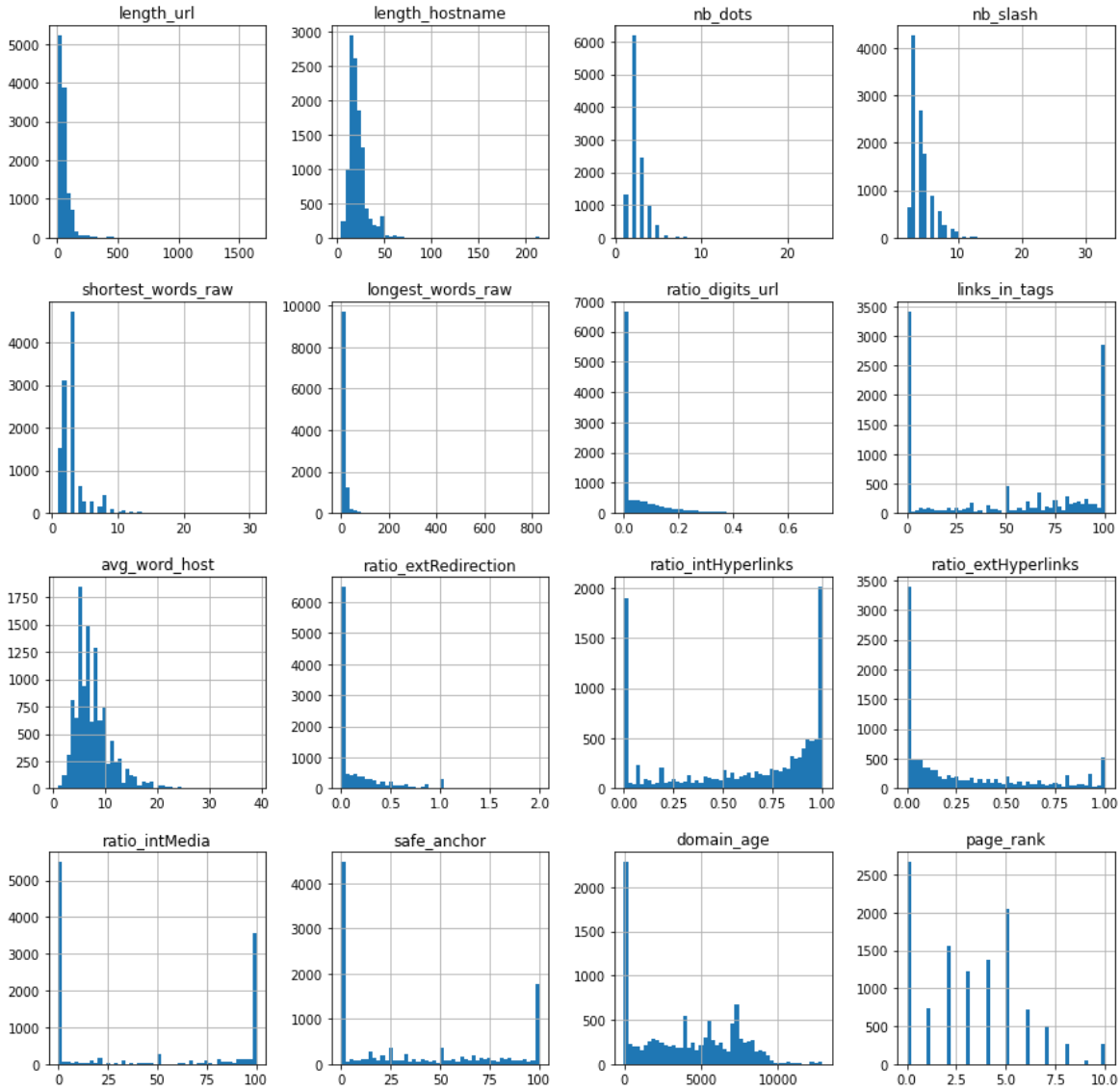


Hình 2.13: Thuộc tính nb_and



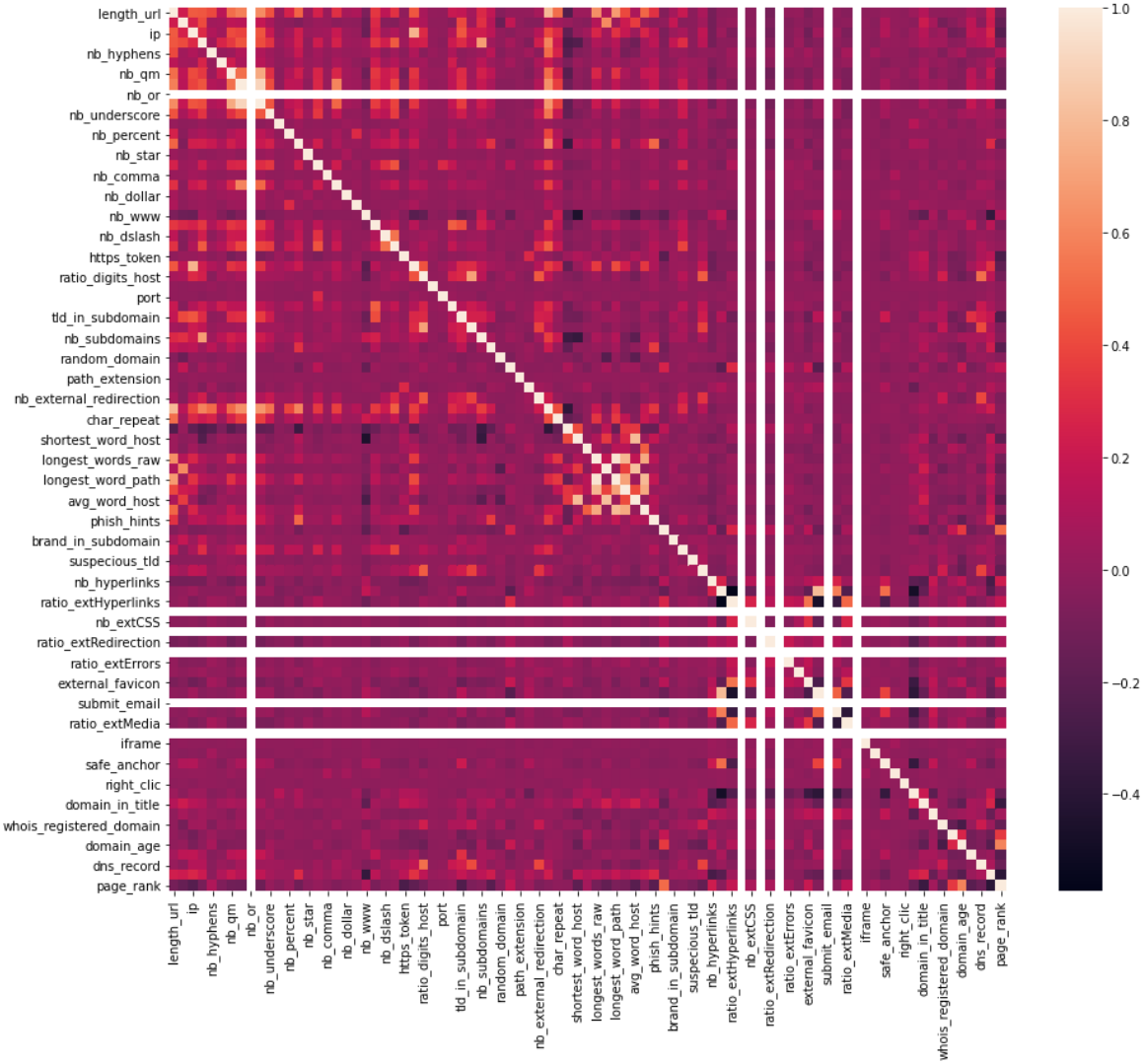
Hình 2.14: Thuộc tính nb_or

Với mỗi thuộc tính, quan sát các đặc điểm như miêu tả về tổng số dữ liệu hợp lệ, giá trị trung bình, độ lệch chuẩn, giá trị nhỏ nhất, từ giá trị của dữ liệu trở xuống chiếm 25%, 50% và 75%, giá trị lớn nhất. Cụ thể, lấy ví dụ trường hợp của thuộc tính length_url (độ dài của url), ta có giá trị valid thể hiện số lượng dữ liệu hợp lệ là 11,400 hay chính xác là 11,439 dòng chiếm tỉ lệ 100%, không có dòng nào bị mất mát hoặc sai lệch dữ liệu. Giá trị trung bình của thuộc tính là 61.1, độ lệch chuẩn là 55.3, giá trị nhỏ nhất là 12 và lớn nhất là 1,641. Độ dài từ 33 trở xuống chiếm 25%, từ 47 trở xuống chiếm 50% và từ 71 trở xuống chiếm 76%. Các thuộc tính khác ta xem xét tương tự. Tiếp theo tiến hành sử dụng Google Colab để quan sát sự phân bổ của dữ liệu.



Hình 2.15: Phân bố dữ liệu của một số thuộc tính

Hình 2.16 thể hiện phân bố dữ liệu của 16 trên tổng số 87 tính năng của bộ dữ liệu. Từ những biểu đồ trên cho thấy dữ liệu ở từng tính năng phân bố rải rác, không đồng đều và có độ lệch lớn. Dữ liệu phân bố theo khu vực (chiều x), cụ thể là khu vực cao tương đối ít, không có dữ liệu nào trong vùng này, ngược lại dữ liệu trong khu vực thấp và trung bình chiếm tỉ lệ tương đối lớn. Ngoài ra giá trị của dữ liệu cũng nằm trong khoảng (chiều y) thấp nhiều hơn giá trị ở các khoảng cao, lấy ví dụ của tính năng avg_word_host, dữ liệu phân bố nhiều ở khoảng 0-250, khoảng từ 250-750 tương đối và khoảng trên 750 tương đối ít.



Hình 2.16: Ma trận hệ số tương quan giữa các đặc tính

Căn cứ vào Hình 2.17 là bản đồ nhiệt hiển thị ma trận tương quan của các tính năng trong bộ dữ liệu. Phạm vi tương quan tiêu chuẩn từ +1 đến -0.4, trong đố -0.4 là tương quan âm thấp nhất và +1 là tương quan dương cao nhất. Mối tương quan âm được hiển thị trong dải màu tối hơn (dần về đen), trong khi mối tương quan dương được hiển thị trong dải màu sáng hơn (dần về cam nhạt). Đặc biệt trong tập dữ liệu này, ánh xạ của đối tượng khác nhau, có tên submit_email, nb_or, length_url và page_rank, cho thấy màu sáng nhất, có nghĩa là chúng có tương quan cao hoặc thuận. Tương quan tích cực có nghĩa là có thể tìm ra được

các vị trí xuất hiện các nguy cơ bị tấn công cao. Trong khi các mối tương quan phủ định thì lại ít nguy cơ nhất.

2.3. Phương pháp đánh giá

Độ chính xác (hay còn gọi là accuracy) sẽ được sử dụng trong trường hợp này. Độ chính xác là một thước đo để đánh giá các mô hình phân loại. Nói chính xác hơn thì độ chính xác là một phần nhỏ của các dự đoán mà mô hình đã đúng. Về mặt hình thức, độ chính xác được định nghĩa là bằng tỉ lệ giữa số lượng dự đoán chính xác và tổng tất cả các dự đoán, công thức như sau:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

Đối với phân loại nhị phân, độ chính xác cũng có thể được tính theo mặt tích cực (Positive) và tiêu cực (Negative) với công thức như sau:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Cụ thể:

- TP (True Positive): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Positive (dự đoán đúng)
- TN (True Negative): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Negative (dự đoán đúng)
- FP (False Positive): đối tượng ở lớp Negative, mô hình phân đối tượng vào lớp Positive (dự đoán sai)
- FN (False Negative): đối tượng ở lớp Positive, mô hình phân đối tượng vào lớp Negative (dự đoán sai)

True/False thể hiện cho tính chính xác khi phân loại của mô hình, True đồng nghĩa với việc mô hình phân loại đúng và ngược lại. Positive/Negative là lớp mà đối tượng được mô hình xếp vào, Positive có nghĩa rằng mô hình phân đối tượng vào lớp Positive và ngược lại.

2.4. Hiện thực mô hình

Để thực nghiệm, luận văn sử dụng bộ data được công bố, và Google Colaboratory, Python và Tensorflow để xây dựng một số mô hình Deep Learning, cụ thể là Representation Learning. Ngôn ngữ lập trình được viết bằng mã Python với sự trợ giúp của gói TensorFlow. Việc sử dụng máy chủ đám mây cho phép tận dụng sức mạnh của phần cứng của Google Colab để thực thi mã và chạy các hoạt động Tensorflow.

2.4.1. Xử lý các URL

Sử dụng Keras.Tokenizer để chuyển bộ dữ liệu url thành tập các ma trận

```
texts = data['url'].values
texts = [s.lower() for s in texts]
n = len(texts)
arr = []
tk = Tokenizer(num_words=None, char_level=True, oov_token='UNK')
alphabet = "abcdefghijklmnopqrstuvwxyz0123456789,;.!?:'\"/\\|_@#$%^&*~`+-=<>()[]{}"
for i in range(n):
  t1 = texts[i]
  tk.fit_on_texts(t1)
    char_dict = {}
  for i, char in enumerate(alphabet):
    char_dict[char] = i + 1
  tk.word_index = char_dict.copy()
  tk.word_index[tk.oov_token] = max(char_dict.values()) + 1

  t1_sequences = tk.texts_to_sequences(t1)
  t1_matrix = tk.sequences_to_matrix(t1_sequences, 'count')

  t1_data = pad_sequences(t1_sequences, maxlen=len(t1), padding='post')

  t1_data = np.array(t1_data, dtype='int')
  arr.append(t1_data)

# ======================Get classes================
text_classes = df1['label'].values
text_class_list = [x - 1 for x in text_classes]

text_classes = to_categorical(text_class_list)
#https://towardsdatascience.com/character-level-cnn-with-keras-50391c3adf33
arr = np.array(arr)
```

Sau khi xử lý, các ma trận có kích thước là (37x37)

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 255))
arr_imgData = []
for i in range(n):
  t1_data =arr[i]
  t1_data = scaler.fit_transform(t1_data)
  t1_data = np.rint(t1_data)
  arr_imgData.append(t1_data)

arr_imgData = np.array(arr_imgData)

print(arr_imgData.shape)
print(arr_imgData[0].shape)
```

```
(11430,)
(37, 37)
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:10: Vis
  # Remove the CWD from sys.path while we load stuff.
```

Sau đó chuyển sang hình ảnh gray-scale

```
from PIL import Image
import matplotlib.pyplot as plt

plt.figure(figsize=(10,10))
for i in range(9):
    plt.subplot(3,3,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    #plt.imshow(train_images[i])
    #plt.imshow(trainX[i], cmap=plt.get_cmap('gray'))
    img1_data = arr_imgData[i]
    pixels = img1_data.astype(np.uint8)
    pixels = np.resize(pixels, (75, 75))
    image = Image.fromarray(pixels)
    plt.imshow(image, cmap=plt.get_cmap('gray'))
    # The CIFAR labels happen to be arrays,
    # which is why you need the extra index
    plt.xlabel(df1['status'][i])
plt.show()
```

2.4.2. Xây dựng mô hình ResNet18

Để xác định tiền xử lý cho dữ liệu ảnh, chúng ta sẽ phải lật ngang ngẫu nhiên, xoay, chuẩn hóa, v.v. Sau đó, thay đổi kích thước hình ảnh phải là (n * n) vì Resnet chấp nhận kích thước hình ảnh đầu vào là (n * n).

```
transforms_train = transforms.Compose([
    transforms.Resize((75, 75)),   #must same as here
    transforms.RandomResizedCrop(75),
    transforms.RandomHorizontalFlip(), # data augmentation
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225]) # normalization
])
transforms_test = transforms.Compose([
    transforms.Resize((75, 75)),   #must same as here
     transforms.CenterCrop((75, 75)),
    transforms.ToTensor(),
    transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])
])
```

Sau đó chúng ta cần thiết lập đường dẫn thư mục ***train*** và ***test*** của mô hình.

```
train_dir = "/content/Datafol/trainfol"
test_dir = "/content/Datafol/testfol"
train_classa_dir = "/content/Datafol/trainfol/leg" # legitimate
train_classb_dir = "/content/Datafol/trainfol/phi" # phishing
test_classa_dir = "/content/Datafol/testfol/leg"
test_classb_dir = "/content/Datafol/testfol/phi"

# Tạo thư mục Images_Data
create_directory("/content/Datafol")
create_directory(train_dir)
create_directory(test_dir)
create_directory(train_classa_dir)
create_directory(train_classb_dir)
create_directory(test_classa_dir)
create_directory(test_classb_dir)
```

Chia tập dữ liệu thành train và test với tỉ lệ 8:2

```
def move_image1(imgs, fdir, tdir):
  for img in imgs:
    os.rename(img, img.replace(fdir, tdir))


dir_leg = '/content/Images_Data/Legitimate'
dir_phi = '/content/Images_Data/Phishing'
list_phishing = []
list_legitimate = []
for img in os.listdir(dir_phi):
  list_phishing.append(os.path.join(dir_phi, img))
for img in os.listdir(dir_leg):
  list_legitimate.append(os.path.join(dir_leg, img))
train_phishing, test_phishing = train_test_split(list_phishing, test_size=0.2, random_state=0)
train_legitimate, test_legitimate = train_test_split(list_legitimate, test_size=0.2, random_state=0)
```

Tạo ra các trọng số được train trước cho mô hình resnet18 và thay đổi các lớp của nó và phân loại các lớp cụ thể, trong khi Resnet-18 được đào tạo trên nhiều lớp. Xây dựng mô hình sử dụng chức năng tối ưu hóa và mất mát: trình tối ưu hóa SGD và mất mất mát Cross-Entropy.

```
import torch
import torch.nn as nn
import torch.optim as optim
import torchvision
from torchvision import datasets, models, transforms
import numpy as np
import matplotlib.pyplot as plt
import cv2
import time

device = 'cuda' if torch.cuda.is_available() else 'cpu'
print(device)

model = models.resnet18(pretrained=True)   #load resnet18 model
num_features = model.fc.in_features     #extract fc layers features
model.fc = nn.Linear(num_features, 2) #(num_of_class == 2)
model = model.to(device)
criterion = nn.CrossEntropyLoss()  #(set loss function)
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)
```

Xây dựng mô hình, huấn luyện với 150 vòng trở lên.

```
num_epochs = 150   #(set no of epochs)
start_time = time.time() #(for showing time)
for epoch in range(num_epochs): #(loop for every epoch)
    print("Epoch {} running".format(epoch)) #(printing message)
    """ Training Phase """
    model.train()    #(training model)
    running_loss = 0.   #(set loss 0)
    running_corrects = 0
    # load a batch data of images
    for i, (inputs, labels) in enumerate(train_dataloader):
        inputs = inputs.to(device)
        labels = labels.to(device)
        # forward inputs and get output
        optimizer.zero_grad()
        outputs = model(inputs)
        _, preds = torch.max(outputs, 1)
        loss = criterion(outputs, labels)
        # get loss value and update the network weights
        loss.backward()
        optimizer.step()
        running_loss += loss.item() * inputs.size(0)
        running_corrects += torch.sum(preds == labels.data)
    epoch_loss = running_loss / len(train_dataset)
    epoch_acc = running_corrects / len(train_dataset) * 100.
    print('[Train #{}] Loss: {:.4f} Acc: {:.4f}% Time: {:.4f}s'.format(epoch, epoch_loss, epoch_acc, time.time() -start_time))

    """ Testing Phase """
    model.eval()
    with torch.no_grad():
        running_loss = 0.
        running_corrects = 0
        for inputs, labels in test_dataloader:
            inputs = inputs.to(device)
            labels = labels.to(device)
            outputs = model(inputs)
            _, preds = torch.max(outputs, 1)
            loss = criterion(outputs, labels)
            running_loss += loss.item() * inputs.size(0)
            running_corrects += torch.sum(preds == labels.data)
        epoch_loss = running_loss / len(test_dataset)
        epoch_acc = running_corrects / len(test_dataset) * 100.
        print('[Test #{}] Loss: {:.4f} Acc: {:.4f}% Time: {:.4f}s'.format(epoch, epoch_loss, epoch_acc, time.time()- start_time))
```

Sau đó, thử nghiệm và đánh giá mô hình:

```
plt.rcParams['figure.figsize'] = [12, 8]
plt.rcParams['figure.dpi'] = 60
plt.rcParams.update({'font.size': 20})
def imshow(input, title):
    # torch.Tensor => numpy
    input = input.numpy().transpose((1, 2, 0))
    # undo image normalization
    mean = np.array([0.485, 0.456, 0.406])
    std = np.array([0.229, 0.224, 0.225])
    input = std * input + mean
    input = np.clip(input, 0, 1)
    # display images
    plt.imshow(input)
    plt.title(title)
    plt.show()##Testing
model.eval()
start_time = time.time()
criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)
with torch.no_grad():
    running_loss = 0.
    running_corrects = 0
    for i, (inputs, labels) in enumerate(test_dataloader):
        inputs = inputs.to(device)
        labels = labels.to(device)
        outputs = model(inputs)
        _, preds = torch.max(outputs, 1)
        loss = criterion(outputs, labels)
        running_loss += loss.item() * inputs.size(0)
        running_corrects += torch.sum(preds == labels.data)
        if i == 0:
            print('======>RESULTS<======')
            images = torchvision.utils.make_grid(inputs[:4])
            imshow(images.cpu(), title=[class_names[x] for x in labels[:4]])
    epoch_loss = running_loss / len(test_dataset)
    epoch_acc = running_corrects / len(test_dataset) * 100.
    print('[Test #{}] Loss: {:.4f} Acc: {:.4f}% Time: {:.4f}s'.
          format(epoch, epoch_loss, epoch_acc, time.time() - start_time))
```

CHƯƠNG 3. THÍ NGHIỆM VÀ ĐÁNH GIÁ

3.1. Các trường hợp thí nghiệm

Việc sử dụng máy chủ đám mây cho phép tận dụng sức mạnh của phần cứng của Google Colab để luyện mô hình.

Bộ dữ liệu bao gồm 11,430 dòng và 89 cột cung cấp 11,429 URL với 87 tính năng được trích xuất.

Trong quá trình thí nghiệm huấn luyện xây dựng mô hình, để tìm ra mô hình phù hợp với bộ dữ liệu, luận văn đề xuất 4 trường hợp chuyển dữ liệu URL dạng text sang dữ liệu URL dạng numpy matrix 4 trường hợp như sau:

(1) ***Trường hợp 1:*** chuyển từ ma trận có kích thước 37x37 sang ma trận 75x75, từ đó convert thành ảnh grayscale. Sau đó chạy huấn luyện với tỷ lệ tập train / tập test là 80 / 20. Số epoch chạy cho trường hợp này là 150 epoches.

(2) ***Trường hợp 2:*** chuyển từ ma trận có kích thước 37x37 sang ma trận 100x100, từ đó convert thành ảnh grayscale. Sau đó chạy huấn luyện với tỷ lệ tập train / tập test là 80 / 20. Số epoch chạy cho trường hợp này là 150 epoches.

(3) ***Trường hợp 3:*** chuyển từ ma trận có kích thước 37x37 sang ma trận 192x192, từ đó convert thành ảnh grayscale. Sau đó chạy huấn luyện với tỷ lệ tập train / tập test là 80 / 20. Số epoch chạy cho trường hợp này là 150 epoches.

(4) ***Trường hợp 4:*** chuyển từ ma trận có kích thước 37x37 sang ma trận 224x224, từ đó convert thành ảnh grayscale. Sau đó chạy huấn luyện với tỷ lệ tập train / tập test là 80 / 20. Số epoch chạy cho trường hợp này là 150 epoches.

Sau khi thí nghiệm với 4 trường hợp kích thước ảnh, so sánh và đánh giá các kết quả thu được cũng như thời gian huấn luyện.

3.2. Luyện và kiểm thử mô hình

Về cài đặt, cả 4 trường hợp đều cài đặt như nhau:

```
#Now, we need to start training, if the above steps work fine, then you can easily start training with the below code
num_epochs = 150   #(set no of epochs)
start_time = time.time() #(for showing time)
for epoch in range(num_epochs): #(loop for every epoch)
    print("Epoch {} running".format(epoch)) #(printing message)
    """ Training Phase """
    model.train()    #(training model)
    running_loss = 0.   #(set loss 0)
    running_corrects = 0
    # load a batch data of images
    for i, (inputs, labels) in enumerate(train_dataloader):
        inputs = inputs.to(device)
        labels = labels.to(device)
        # forward inputs and get output
        optimizer.zero_grad()
        outputs = model(inputs)
        _, preds = torch.max(outputs, 1)
        loss = criterion(outputs, labels)
        # get loss value and update the network weights
        loss.backward()
        optimizer.step()
        running_loss += loss.item() * inputs.size(0)
        running_corrects += torch.sum(preds == labels.data)
    epoch_loss = running_loss / len(train_dataset)
    epoch_acc = running_corrects / len(train_dataset) * 100.
    print('[Train #{}] Loss: {:.4f} Acc: {:.4f}% Time: {:.4f}s'.format(epoch, epoch_loss, epoch_acc, time.time() -start_time))

    """ Testing Phase """
    model.eval()
    with torch.no_grad():
        running_loss = 0.
        running_corrects = 0
        for inputs, labels in test_dataloader:
            inputs = inputs.to(device)
            labels = labels.to(device)
            outputs = model(inputs)
            _, preds = torch.max(outputs, 1)
            loss = criterion(outputs, labels)
            running_loss += loss.item() * inputs.size(0)
            running_corrects += torch.sum(preds == labels.data)
        epoch_loss = running_loss / len(test_dataset)
        epoch_acc = running_corrects / len(test_dataset) * 100.
        print('[Test #{}] Loss: {:.4f} Acc: {:.4f}% Time: {:.4f}s'.format(epoch, epoch_loss, epoch_acc, time.time()- start_time))
```

Kết quả thu được sau khi chạy trường hợp 1

```
Epoch 146 running
[Train #146] Loss: 0.5116 Acc: 73.2065% Time: 2930.2671s
[Test #146] Loss: 0.5900 Acc: 70.7787% Time: 2933.1114s
Epoch 147 running
[Train #147] Loss: 0.5108 Acc: 73.6002% Time: 2949.5793s
[Test #147] Loss: 0.5741 Acc: 71.7848% Time: 2952.4797s
Epoch 148 running
[Train #148] Loss: 0.5110 Acc: 73.3158% Time: 2969.2179s
[Test #148] Loss: 0.6412 Acc: 71.5223% Time: 2972.1125s
Epoch 149 running
[Train #149] Loss: 0.5129 Acc: 73.6439% Time: 2990.0927s
[Test #149] Loss: 0.6498 Acc: 68.8539% Time: 2993.0100s
```

Kết quả thu được sau khi chạy trường hợp 2

```
Epoch 145 running
[Train #145] Loss: 0.1528 Acc: 93.4165% Time: 5267.7632s
[Test #145] Loss: 1.0975 Acc: 67.1479% Time: 5272.7419s
Epoch 146 running
[Train #146] Loss: 0.1621 Acc: 92.8696% Time: 5303.2566s
[Test #146] Loss: 1.1158 Acc: 65.9668% Time: 5308.1906s
Epoch 147 running
[Train #147] Loss: 0.1567 Acc: 93.2415% Time: 5338.7614s
[Test #147] Loss: 1.1218 Acc: 66.0980% Time: 5343.6832s
Epoch 148 running
[Train #148] Loss: 0.1576 Acc: 93.3618% Time: 5374.2988s
[Test #148] Loss: 1.1257 Acc: 67.5853% Time: 5379.2483s
Epoch 149 running
[Train #149] Loss: 0.1575 Acc: 93.2415% Time: 5409.8471s
[Test #149] Loss: 1.0184 Acc: 68.8539% Time: 5414.7810s
```

Kết quả thu được sau khi chạy trường hợp 3

```
Epoch 144 running
[Train #144] Loss: 0.1558 Acc: 93.3399% Time: 5230.3393s
[Test #144] Loss: 1.0467 Acc: 67.9353% Time: 5237.1176s
Epoch 145 running
[Train #145] Loss: 0.1528 Acc: 93.4165% Time: 5267.7632s
[Test #145] Loss: 1.0975 Acc: 67.1479% Time: 5272.7419s
Epoch 146 running
[Train #146] Loss: 0.1621 Acc: 92.8696% Time: 5303.2566s
[Test #146] Loss: 1.1158 Acc: 65.9668% Time: 5308.1906s
Epoch 147 running
[Train #147] Loss: 0.1567 Acc: 93.2415% Time: 5338.7614s
[Test #147] Loss: 1.1218 Acc: 66.0980% Time: 5343.6832s
Epoch 148 running
[Train #148] Loss: 0.1576 Acc: 93.3618% Time: 5374.2988s
[Test #148] Loss: 1.1257 Acc: 67.5853% Time: 5379.2483s
Epoch 149 running
[Train #149] Loss: 0.1575 Acc: 93.2415% Time: 5409.8471s
[Test #149] Loss: 1.0184 Acc: 68.8539% Time: 5414.7810s
```

Kết quả thu được sau khi chạy trường hợp 4

```
Epoch 145 running
[Train #145] Loss: 0.3310 Acc: 84.4926% Time: 6153.5694s
[Test #145] Loss: 0.8597 Acc: 58.7927% Time: 6158.8872s
Epoch 146 running
[Train #146] Loss: 0.3325 Acc: 84.3066% Time: 6195.5201s
[Test #146] Loss: 0.8393 Acc: 61.5048% Time: 6200.9193s
Epoch 147 running
[Train #147] Loss: 0.3255 Acc: 84.8535% Time: 6237.6464s
[Test #147] Loss: 0.8769 Acc: 61.1549% Time: 6243.0402s
Epoch 148 running
[Train #148] Loss: 0.3226 Acc: 85.3565% Time: 6279.5259s
[Test #148] Loss: 0.9210 Acc: 59.7550% Time: 6284.9492s
Epoch 149 running
[Train #149] Loss: 0.3198 Acc: 85.2144% Time: 6321.5848s
[Test #149] Loss: 0.8344 Acc: 59.0989% Time: 6328.7170s
```

3.3. Kết quả và nhận xét

Do kích thước ma trận chuyển từ URL sang là (37x37) khá nhỏ so với các ảnh ResNet18, do đó, để mô hình hiệu quả hơn, các ảnh cần scale-up lên kích thước

lớn hơn để ResNet18 hoạt động hiệu quả hơn. Vì vậy, luận văn này sử dụng 4 trường hợp scale-up: 75x75 pixel, 100x100 pixel, 191x192 pixel và 224x224 pixel.

Kết quả trường hợp 1: Kích thước ảnh 75x75 pixel

Bảng 3.1: Trường hợp 1 với kích thước 75x75 pixel

TH1 (75x75)	**Mất mát (Loss)**	**Độ chính xác (Accuracy)**
Tập Train	0.5129	73.6439%
Tập Test	0.6498	68.8539%

Kết quả trường hợp 2. Kích thước ảnh 100x100 pixel

Bảng 3.2: Trường hợp 2 với kích thước 100x100 pixel

TH2 (100x100)	**Mất mát (Loss)**	**Độ chính xác (Accuracy)**
Tập Train	0.1575	93.2415%
Tập Test	1.0184	68.8539%

Kết quả trường hợp 3. Kích thước ảnh 192x192 pixel

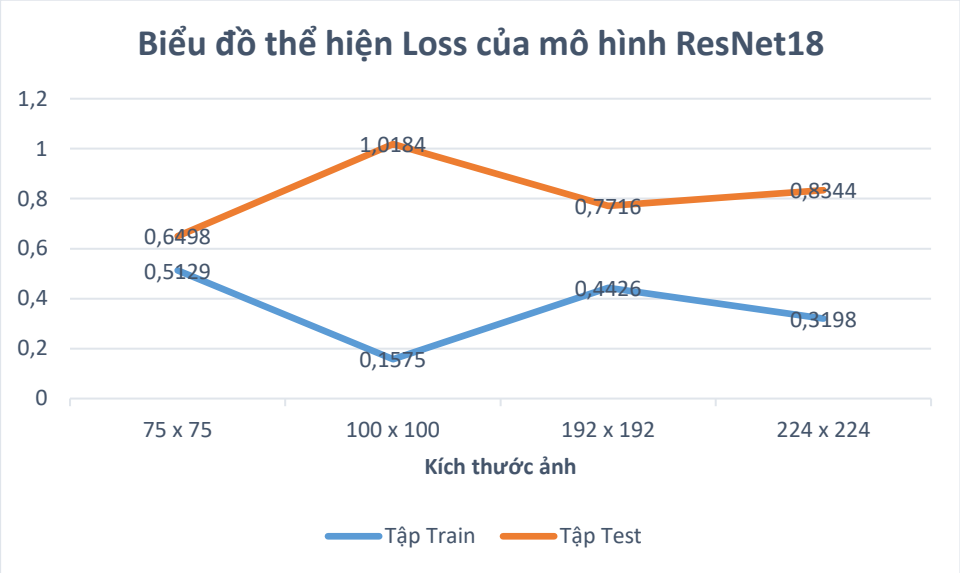
Bảng 3.3. Trường hợp 3 với kích thước 192x192 pixel

TH3 (192x192)	**Mất mát (Loss)**	**Độ chính xác (Accuracy)**
Tập Train	0.4426	77.7887%
Tập Test	0.7716	58.9676%

Kết quả trường hợp 4. Kích thước ảnh 224x224 pixel

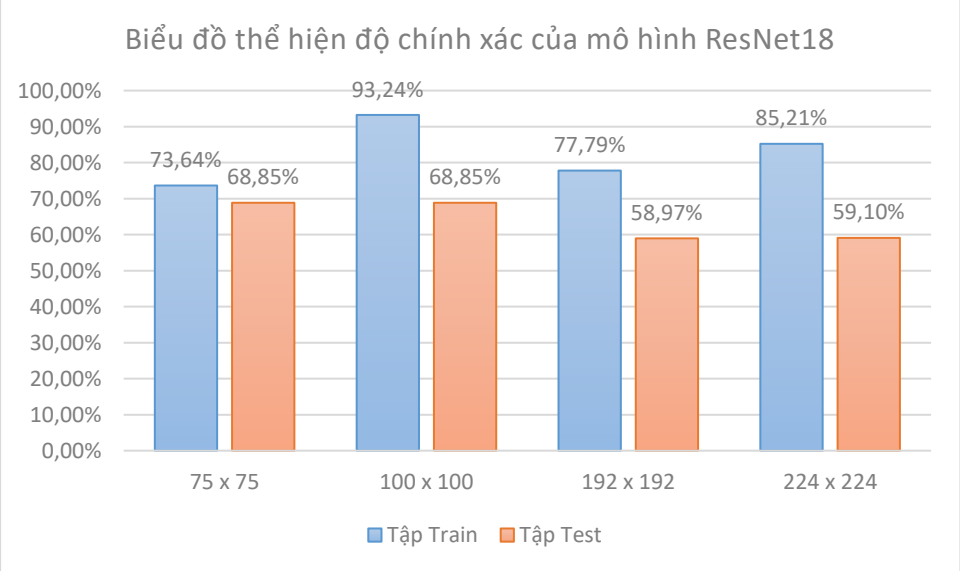
Bảng 3.4: Trường hợp 4 với kích thước 224x224 pixel

TH4 (224x224)	**Mất mát (Loss)**	**Độ chính xác (Accuracy)**
Tập Train	0.3198	85.2144%
Tập Test	0.8344	59.0989%



Hình 3.1: Biểu đồ thể hiện Loss của mô hình ResNet18 với 4 trường hợp

Độ mất mát của tập Train luôn thấp hơn tập test.



Hình 3.2: Biểu đồ thể hiện Accuracy của mô hình ResNet18 với 4 trường hợp

Ở trường hợp kích thước ảnh (100x100) thì tập train đạt độ chính xác cao nhất 93.24%. Tuy nhiên, ở trường hợp kích thước ảnh (75x75), (192x192), (224x224) thì tập test và tập train chênh lệch không cao, độ chính xác dao động từ 58% tới 85%. Vì vậy, kích thước ảnh thay đổi dẫn đến chất lượng mô hình thay đổi theo. Với kết quả như trên rõ ràng với các kích thước ảnh (100x100) đạt độ chính

xác cao vì đây là kích thước ảnh thuận lợi cho việc học và suy diễn trong mạng CNN.

Việc sử dụng máy chủ đám mây cho phép tận dụng sức mạnh của phần cứng của Google Colab để luyện mô hình chỉ phù hợp với bộ dữ liệu có dung lượng nhỏ nên kết quả độ chính xác chưa cao. Bộ dữ liệu bao gồm 11,430 dòng và 89 cột cung cấp 11,429 URL với 87 tính năng được trích xuất.

KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết quả nghiên cứu của đề tài

Tấn công Phishing đã trở thành một trong những dạng tấn công phổ biến mà người sử dụng mạng máy tính và các tổ chức cung cấp dịch vụ phải đối mặt. Phát hiện tấn công Phishing đã trở thành một đề tài nghiên cứu và là một bài toán vô cùng quan trọng trong an toàn thông tin. Sự phát triển của trí tuệ nhân tạo trong thời gian gần đây góp phần ngăn chặn, phát hiện tấn công Phishing với độ chính xác cao. Trong đó mô hình representation learning đã phát huy nhiều ưu điểm cho vấn đề này. Trong đề tài luận văn này, học viên đã tiến hành tìm hiểu những lý thuyết về trí tuệ nhân tạo để nghiên cứu mô hình representation learning từ đó xây dựng mô hình ứng dụng thực tế nhằm áp dụng vào công việc hiện tại.

Hiện nay học viên đang công tác tại Sở Thông tin và Truyền thông Tây Ninh trước những thực trạng đang xảy ra tại nơi học viên làm việc với lĩnh vực đang theo học tập và nghiên cứu. Được sự đồng ý của TS. Nguyễn Hồng Sơn học viên chọn đề tài luận văn: "Ứng dụng representation learning phát hiện tấn công phishing", ứng dụng này góp phần giải quyết các vấn đề rất cần thiết tại nơi học viên làm việc.

2. Hạn chế luận văn

Luận văn sử dụng bộ dữ liệu công bố, nên cần phải đưa bộ dữ liệu thực tế để tiến hành nghiên cứu và đo đạc.

Chưa cài đặt được trên môi trường mạng thực tế mà dừng lại ở mức thực nghiệm phân tích xây dựng mô hình trên dataset.

3. Vấn đề kiến nghị và hướng đi tiếp theo của nghiên cứu

- Xây dựng bộ dữ liệu thực tế với tình hình cyber security ở Việt Nam và thời điểm hiện tại, cập nhật liên tục.

- Cài đặt mô hình và ứng dụng realtime vào mô hình cho việc phát hiện Phishing trên mạng thực

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Subhi Gupta, Abhisshek Singhal, Akansha Kapoor, "A Literature Survey on Social Engineering Attacks: Phishing Attack," in *International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, 2016.

[2] Abdul Basit, Maham Zafar, Xuan Liu, Abdul Rehman Javed, Zunera Jalil , Kashif Kifayat, "A comprehensive survey of AI-enabled phishing attacks detection techniques," pp. 1-16, 2020.

[3] Jian Feng, Lianyang Zou, Ou Ye, Jingzhou Han, "Web2Vec: Phishing Webpage Detection Method Based on Multidimensional Features Driven by Deep Learning," IEEE, 2020, pp. 221214 - 221224.

[4] Harikrishnan NB, Vinayakumar R, Soman KP, "A machine learning approach towards phishing email detection," in *Proceedings of the Anti-Phishing Pilot at ACM International Workshop on Security and Privacy Analytics (IWSPA AP)*, 2018.

[5] Yasser Yasami, Saadat Pour Mozaffari, "A novel unsupervised classification approach for network anomaly detection by k-Means clustering and ID3 decision tree learning methods," *The Journal of Supercomputing*, pp. 231-245, 2009.

[6] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in *Proceedings of the Second IEEE International Conference on Computational Intelligence for Security and Defense Applications*, 2009.

[7] N. Đ. Hiền, Máy Vector hỗ trợ đa lớp và ứng dụng phát hiện tấn công, Hà Nội, 2012.

[8] L. Zhang and P. Zhang, "PhishTrim: Fast and adaptive phishing detection based on deep representation learning," in *IEEE International Conference on Web Services (ICWS)*, 2020.

[9] L. Lakshmi, M. P. Reddy, C. Santhaiah, and U. J. Reddy, "Smart phishing detection in web pages using supervised deep learning classification and optimization technique ADAM," p. 3549–3564, 2021.

[10] Z. Yuan, Q. Yuan, and J. Wu, "Phishing detection on ethereum via learning representation of transaction subgraphs," in *Blockchain and Trustworthy Systems*, 2020, pp. 178-191.

[11] N. Đ. Thuân, Introduction to Data Mining, Học viện Công nghệ Bưu chính viễn thông, 2014.

[12] Muhammad Usman, Mian Ahmad Jan, Xiangjian He, Jinjun Chen, "A Survey on Representation Learning Efforts in Cybersecurity Domain," vol. 52, pp. 1-28, 2019.

[13] Chidimma Opara, Bo Wei, Yingke Chen, "HTMLPhish: Enabling Phishing Web Page Detection by Applying Deep Learning Techniques on HTML Analysis," *International Joint Conference on Neural Networks (IJCNN) 2020*, 2020.

[14] P.Kalaharsha, B. M. Mehtre, "Detecting Phishing Sites - An Overview," Hyderabad, 2021.

[15] "National Cyber Security Center," 10 October 2017. [Online]. Available: https://www.ncsc.gov.uk/collection/small-business-guide/avoiding-phishing-attacks. [Accessed 2021].

[16] M. T. Jones, "Artificial intelligence: A System Approach," pp. 143-176, 2008.

[17] Z. Sedighi, H. E. Komleh, A. Bagheri, "RLOSD: Representation Learning based Opinion Spam Detection," in *3rd Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, Kashan, 2017.

[18] G. Zhong, Li-Na Wang, X. Ling, J. Dong, "An overview on data representation learning: From traditional feature learning to recent deep learning," *The Journal of Finance and Data Science 2*, pp. 265-278, 2017.

[19] R. V. Belle, S. Mitrovic, J. D. Weerdt, "Representation Learning in Graphs for Credit Card Fraud Detection," in *Mining Data for Financial Applications*, Leuven, 2020, pp. 1-15.

[20] Zhiyuan Liu, Yankai Lin, Maosong Sun, "Representation Learning for Natural Language Processing," in *Representation Learning and NLP*, Spinger, 2020, pp. 1-11.

[21] T. A. Pham, Q. U. Nguyen, X. H. Nguyen, "Phishing Attacks Detection Using Genetic Programming," in *Advances in Intelligent Systems and Computing 245*, 2014.

[22] L. D. Nguyen, D. N. Le, L. T. Vinh, "Detecting Phishing Web Pages based on DOM-Tree Structure and Graph Matching Algorithm," in *SoICT '14 Proceedings of the Fifth Symposium on Information and Communication Technology*, 2014.

[23] T. C. Truong, Q. B. Diep, I. Zelinka, "Artificial Intelligence in the Cyber Domain: Offense and Defense," pp. 1-24, 4 March 2020.

[24] C. D. Xuan, H. D. Nguyen, T. V. Nikolaevich, "A Framework for Vietnamese Email Phishing Detection," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 1, pp. 2258-2264, 2019.

[25] Y. Bengio, A. Courville, P. Vincent, "Representation Learning: A Review and New Perspectives," pp. 1-30, 2014.

[26] Guoqiang Zhong, Li-Na Wang, Junyu Dong, "An Overview on Data Representation Learning: From Traditional Feature Learning to Recent Deep Learning," *Journal of Finance and Data Science as an invited paper,* 2016.

[27] Selvakumari M, Sowjanya M, Sneha Das, Padmavathi S, "Phishing website detection using machine learning and deep learning techniques," *Journal of Physics: Conference Series*, pp. 1-7, 2021.

[28] I. Arnaldo, A. Cuesta-Infante, A. Arun, M. Lam, C. Bassias, K. Veeramachaneni, "Learning Representations for Log Data in Cybersecurity," in *International Conference on Cyber Security Cryptography and Machine Learning*, 2017.

[29] Moruf Akin Adebowale, Khin T. Lwin, M. A. Hossain, "Intelligent phishing detection scheme using deep learning algorithms," *Journal of Enterprise Information Management,* 2020.

[30] Yang, R.; Zheng, K.; Wu, B.; Wu, C.; Wang, X., "Phishing Website Detection Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning," *sensors*, 2021.

[31] Alfredo Cuzzocrea, Fabio Martinelli, and Francesco Mercaldo, "Applying Machine Learning Techniques to Detect and Analyze Web Phishing Attacks," in *Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services (iiWAS2018)*, 2018.

[32] Lázaro B. et al, "A lightweight data representation for phishing URLs detection in IoT environments," in *Information Sciences*, 2022.

[33] Wikipedia contributors, "Artificial neural network," Wikipedia, The Free Encyclopedia, [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Accessed 24 May 2021].

[34] E. Kavlakoglu, "AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?," IBM, [Online]. Available: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks.

[35] A. Hannousse and S. Yahiouche, "Web page phishing detection," *Mendeley Data*, V3, doi: 10.17632/c2gw7fy2j4.3, 2021.