HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Thanh Hà

NGHIÊN CỨU PHƯƠNG PHÁP XÁC ĐỊNH THỨ TỰ ƯU TIÊN CỦA THƯ ĐIỆN TỬ

LUẬN ÁN TIẾN SĨ KỸ THUẬT

Hà Nội – Năm 2023

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Thanh Hà

NGHIÊN CỨU PHƯƠNG PHÁP XÁC ĐỊNH THỨ TỰ ƯU TIÊN CỦA THƯ ĐIỆN TỬ

Chuyên ngành : Hệ thống thông tin
Mã số: 9.48.01.04

LUẬN ÁN TIẾN SĨ KỸ THUẬT

NGƯỜI HƯỚNG DẪN KHOA HỌC:
1. PGS. TS. Trần Quang Anh
2. TS. Trần Hùng

Hà Nội - Năm 2023

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi. Nội dung của luận án có tham khảo và sử dụng các tài liệu, thông tin được đăng tải trên những tạp chí và các trang web theo danh mục tài liệu tham khảo. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Hà Nội, ngày tháng năm 2023

Người cam đoan

Nguyễn Thanh Hà

LỜI CẢM ƠN

Lời đầu tiên, tôi xin trân trọng cảm ơn tới Ban Giám đốc Học viện, Khoa Đào tạo Sau Đại học, các Thầy Cô giáo và các Khoa-Phòng liên quan của Học viện đã tạo điều kiện giúp đỡ trong suốt quá trình làm nghiên cứu sinh tại trường.

Tôi xin gửi lời cảm ơn sâu sắc đến PGS.TS. Trần Quang Anh. Thầy là người định hướng và tận tình hướng dẫn, chỉ bảo cho tôi trong suốt quá trình theo đuổi con đường học thuật. Những phương pháp và tầm nhìn của thầy là cơ sở vững chắc cho những thành tựu khoa học mà tôi đạt được.

Tôi xin gửi lời cám ơn chân thành đến TS. Trần Hùng. Thầy là người hướng dẫn, tư vấn quý giá, thầy đã luôn động viên, ủng hộ tôi hoàn thành bản luận án. Thầy đã hướng dẫn phương pháp nghiên cứu khoa học và kịp thời gợi ý nhiều hướng tiếp cận cho nghiên cứu sinh.

Tôi xin dành sự yêu thương và cám ơn tới gia đình, những người thân đã luôn đồng hành cùng tôi vượt qua những khó khăn trên suốt một chặng đường dài.

Cuối cùng, Tôi xin chân thành cảm ơn các lãnh đạo, các bạn đồng nghiệp tại cơ quan đã luôn tạo mọi điều kiện tốt nhất cho tôi thực hiện nghiên cứu của mình.

Xin chân thành cảm ơn!

Hà Nội, ngày tháng năm 2023

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

CLI	Command Line Interface	Giao diện dòng lệnh
DAG	Directed Acyclic Graph	Đồ thị định hướng không tuần hoàn
DAGSVM	Directed Acyclic Graph Support Vector Machine	Đồ thị định hướng không tuần hoàn với máy vector hỗ trợ
ESP	Email Service Provider	Nhà cung cấp dịch vụ thư điện tử
FAR	False Alarm Rate	Tỷ lệ cảnh báo nhầm
GD	Gradient Descent	Thuật toán xuống dốc
DKIM	DomainKeys Identified Mail	Giao thức xác thực người gửi DomainKeys
DMARC	Domain-based Message Authentication, Reporting and Conformance	Giao thức chứng thực, tố cáo và kiểm tra thông điệp dựa trên tên miền
HTML	Hyper Text Markup Language	Ngôn ngữ đánh dấu siêu văn bản
IETF	Internet Engineering Task Force	Tổ chức thiết kế và phát triển Internet quốc tế
ISP	Internet Service Provider	Nhà cung cấp dịch vụ Internet
LMTP	Local Mail Transfer Protocol	Giao thức truyền tải thư cục bộ
MDA	Mail Delivery Agent	Trình chuyển phát thư
MIME	Multipurpose Internet Mail Extensions	Giao thức mở rộng thư điện tử Internet đa mục đích
MLP	Multi-Layer Perceptron	Mạng perceptron nhiều lớp
MTA	Mail Transfer Agent	Trình truyền tải thư
MUA	Mail User Agent	Trình duyệt thư điện tử
OB-MC	Order-Based Most Confident	Bỏ phiếu tự tin nhất có thứ tự
OB-MV	Order-Based Majority Voting	Bỏ phiếu đa số có thứ tự
OVA	One versus All	Một đối với tất cả
OVO	One versus One	Một đối với một
OVR	One versus Rest	Một đối với những cái khác
POP	Post Office Protocol	Giao thức bưu điện
RBL	Realtime Black List	Danh sách đen thời gian thực
RFC	Request For Comments	Yêu cầu bình luận
SGD	Stochastic Gradient Descent	Thuật toán xuống dốc ngẫu nhiên
SMS	Short Message Service	Dịch vụ tin nhắn ngắn
SMTP	Simple Message Transfer Protocol	Giao thức truyền thông điệp đơn giản
SPF	Sender Policy Framework	Bộ quy định dành cho người gửi thư
SVM	Support Vector Machine	Máy vector hỗ trợ
SVOR	Support Vector Ordinal Regression	Hồi quy thứ bậc dựa trên máy vector hỗ trợ
TCP	Transmission Control Protocol	Giao thức điều khiển truyền dẫn
TF	Term Frequency	Tần số từ khóa

TF-IDF	Term Frequency – Inverse Document Frequency	Tần số từ khóa – Tần số tài liệu nghịch đảo
TLS	Transport Layer Security	Giao thức bảo mật tầng giao vận
TREC	Text REtrieval Conference	Hội nghị về khai phá dữ liệu văn bản
UCE	Unsolicited Commercial Email	Thư quảng cáo không mong muốn
UBE	Unsolicited Bulk Email	Thư gửi hàng loạt không mong muốn

DANH MỤC CÁC BẢNG, BIỂU

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC KÝ HIỆU TOÁN HỌC DÙNG TRONG LUẬN ÁN

Ký hiệu	Ý nghĩa
$\{x_1, x_2, \ldots, x_n\}$	Tập hợp gồm n phần tử
$f_{a,b}(c)$	Hàm f với các tham số a, b và đầu vào c
$f(x)$: E → A	Hàm f nhận đầu vào x thuộc tập E và có đầu ra thuộc tập A
$P(A\|B)$	Xác suất của sự kiện A khi có sự kiện B
$\overline{\mathrm{S}}$	Phủ định của sự kiện S
∧	Phép hội (AND)
∨	Phép tuyển (OR)
A ∪ B	Hợp của hai tập A và B
∂	Phép đạo hàm
$\|\mathbf{V}\|$	Độ dài của **V** khi **V** là một vector
$\|x\|$	Giá trị tuyệt đối của x khi x là một số thực
$\mathbb{R}^{\mathrm{N}}$	Không gian số thực N chiều
ln	Hàm logarit tự nhiên
$X \succcurlyeq Y$	Phương án X vượt trội phương án Y

MỞ ĐẦU

1. GIỚI THIỆU

Thư điện tử là một hệ thống chuyển nhận thư từ qua các mạng máy tính. Thư điện tử là một trong những ứng dụng quan trọng nhất mà Internet mang lại. Thư điện tử được sử dụng vào nhiều mục đích khác nhau từ trao đổi thông tin, liên lạc, xác thực danh tính cho đến lưu trữ thông tin, dữ liệu. Thư điện tử có tốc độ truyền thông tin vượt trội so với các phương thức thư tín truyền thống. Trong khoảng từ ba thập kỷ trở lại đây, thư điện tử được sử dụng ngày càng nhiều trên khắp thế giới. Sự phổ biến của nó có nhiều nguyên nhân như chi phí thấp, tính tiện dụng và sự tích hợp với rất nhiều ứng dụng khác trên Internet. Ngày nay, thư điện tử đã và đang được coi là công cụ giao tiếp điện tử chính thống trong công việc và đời sống.

Quá tải thư điện tử là một vấn đề nổi bật mà người dùng gặp phải khi sử dụng dịch vụ này. Đây là tình trạng người dùng nhận được quá nhiều thư, dẫn đến không có đủ thời gian để đọc và xử lý hết lượng thư đó. Tác giả của [57] nhận xét rằng vấn đề quá tải thư điện tử xảy ra khi người dùng nhận được trên 10 bức thư mỗi ngày. Tình trạng này làm ảnh hưởng đến hiệu quả và lợi ích của điện tử đối với người dùng. Các tác hại của vấn đề quá tải thư điện tử [32] bao gồm: giảm năng suất làm việc, ngăn cản những sáng kiến trong công việc, làm mất sự cân bằng giữa công việc và cuộc sống.

Vấn đề quá tải thư điện tử có nguyên nhân đến từ cả thư rác và thư hợp lệ. Những ưu điểm mà thư điện tử mang đến cho người dùng đồng thời cũng được các nhà tiếp thị khai thác như một cách quảng bá sản phẩm, dịch vụ hiệu quả với chi phí thấp. Xuất hiện ngay từ khi thư điện tử ra đời vào giữa thập kỷ 90, những bức thư quảng cáo mà người dùng không mong muốn là ví dụ điển hình của thư rác. Thư rác gây phiền toái khó chịu, tốn thời gian xử lý cho người dùng, giảm tốc độ mạng và tốc độ xử lý của máy chủ. Tuy nhiên, thư rác không phải là yếu tố duy nhất gây ra vấn nạn quá tải thư điện tử. Ngay cả khi các bộ lọc đã loại bỏ được phần lớn thư rác khỏi hòm thư của người dùng, số lượng thư hợp lệ còn lại vẫn làm cho họ không có đủ thời gian để xử lý.

Để giảm thiểu thời gian xử lý thư điện tử cho người dùng, các công cụ hỗ trợ sắp xếp hòm thư là cần thiết. Nền tảng để phát triển các công cụ đó là phương pháp xác định

thứ tự ưu tiên của thư điện tử. Một số ứng dụng dựa trên phương pháp này là các bộ lọc thư rác, công cụ xếp hạng thư điện tử, công cụ gợi ý hành động cần thực hiện đối với thư điện tử.

Luận án này sẽ tập trung nghiên cứu một số phương pháp xác định thứ tự ưu tiên của thư điện tử. Phần tiếp theo sẽ trình bày về tình trạng quá tải thư điện tử trên thế giới, sự cần thiết phải nghiên cứu các phương pháp mới để xác định thứ tự ưu tiên của thư điện tử, cũng như phạm vi và phương pháp nghiên cứu của luận án.

2. TÍNH CẤP THIẾT CỦA LUẬN ÁN

Các báo cáo về thư rác đều khẳng định rằng thư rác chiếm phần lớn trong số những bức thư được truyền tải trên mạng Internet. Theo thống kê của Văn phòng An toàn thông tin – Đại học Texas (Hoa Kỳ), vào tháng 7 năm 2019, hệ thống IronPort đã xử lý hơn 11 triệu bức thư, trong đó 78.0% là thư rác[1]. Theo báo cáo của Symantec[2], tỷ lệ spam trên toàn cầu là 55% trong năm 2017 và 2018. Hãng Trustwave cũng công bố số liệu[3] về tỷ lệ spam là 45.3% trong năm 2018 và 28.5% trong năm 2019. Ngoài ra, thống kê của Kaspersky[4] cho thấy tỷ lệ thư rác là khoảng 55% trong năm 2019 và 2020. Với khối lượng lớn như vậy, thư rác gây ra nhiều thiệt hại lớn về kinh tế, xã hội. Nghiên cứu của Rao và Reiley [50] năm 2012 đã dự đoán thiệt hại mà thư rác gây ra cho nền kinh tế Mỹ là khoảng 20 tỷ đô-la Mỹ mỗi năm. Không chỉ gây thiệt hại về tiền bạc, thư rác còn làm giảm hiệu quả làm việc, gây căng thẳng, tiêu tốn thời gian của người lao động… Những điều này cũng đồng nghĩa với việc năng suất lao động giảm, ảnh hưởng tới hiệu quả kinh doanh. Đôi khi những bức thư chứa mã độc có tiềm năng dẫn đến dữ liệu trong máy tính bị phá hủy. Ngoài ra, tài nguyên của ISP cũng bị chiếm dụng nhiều khi thư rác được gửi.

Theo Radicati[5], trong năm 2019, có khoảng 293.6 tỷ bức thư được gửi và nhận mỗi ngày và khoảng 3.93 tỷ người dùng. Những con số về khối lượng sử dụng thư điện tử cũng được mô tả trong nhiều báo cáo của các tập đoàn về an ninh mạng như Kaspersky[4],

[1] https://www.utep.edu/information-resources/iso/security-awareness/statistics/spam-statistics.html
[2] https://www.statista.com/statistics/270899/global-e-mail-spam-rate/
[3] https://www.statista.com/statistics/420400/spam-email-traffic-share-annual/
[4] https://www.statista.com/statistics/420391/spam-email-traffic-share/
[5] https://www.statista.com/statistics/255080/number-of-e-mail-users-worldwide/; https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/

Trustwave[3] và Symantec[2]. Như vậy, người dùng thư điện tử ngày nay nhận được quá nhiều thư hợp lệ, dẫn đến tình trạng quá tải. Một cuộc khảo sát trên phạm vi toàn quốc ở Mỹ về việc sử dụng thư điện tử cho công việc [31] đã chỉ ra rằng các nhân viên văn phòng nhận được trung bình 41 bức thư hợp lệ mỗi ngày. Số lượng người tham gia khảo sát là 484 người, tất cả đều có việc làm và hoàn thành trọn vẹn phiếu điều tra. Theo một nghiên cứu trên phạm vi nhỏ hơn về vấn đề quá tải thư điện tử [57] vào năm 2014, trong số những bức thư mà 28 người tham gia phỏng vấn nhận được, 29% có nội dung không liên quan đến họ. Mỗi người dành ra trung bình trên 20% tổng thời gian làm việc để đọc và xử lý thư điện tử. Khi bị quá tải thư điện tử, họ không còn đủ thời gian để làm các công việc được giao. 14% trong số họ bị quá tải thư điện tử hằng ngày, 46% bị quá tải từ 1 tới 2 ngày mỗi tuần. Theo thống kê của tập đoàn Radicati [64], vào năm 2015 có 112.5 triệu bức thư được sử dụng hằng ngày cho công việc. Trung bình mỗi nhân viên văn phòng gửi và nhận 122 bức thư mỗi ngày, trong số đó có khoảng 12 bức thư rác (chiếm 9.8% tổng số thư) lọt qua bộ lọc vào tới hòm thư của người sử dụng. Dựa theo một nghiên cứu khác [70] trên tập dữ liệu thư điện tử Yahoo Mail với 2 triệu người dùng và 16 tỷ bức thư, tỷ lệ trả lời thư của những người nhận được dưới 20 thư mỗi ngày là 25%. Với những người dùng nhận được khoảng 100 thư mỗi ngày thì tỷ lệ đó giảm xuống chỉ còn 5%.

Tóm lại, có thể thấy thư rác đã và đang tiếp tục gây ra thiệt hại ngày càng lớn trên phạm vi toàn cầu. Việc nghiên cứu những phương pháp mới để đối phó với vấn nạn thư rác ngày càng tăng về số lượng và độ tinh vi là công việc rất quan trọng cần phải thực hiện. Giải quyết bài toán phát hiện thư rác sẽ mang lại lợi ích to lớn cho kinh tế và đời sống xã hội. Đồng thời với vấn nạn thư rác, tình trạng quá tải mà nguyên nhân là thư hợp lệ cũng hiện hữu đối với rất nhiều người dùng và đã gây ra ảnh hưởng nghiêm trọng đến trải nghiệm sử dụng thư điện tử của họ, đặc biệt là trong công việc.

3. MỤC TIÊU CỦA LUẬN ÁN

Lọc thư rác là hình thức xác định thứ tự ưu tiên của thư điện tử bằng mô hình phân loại hai lớp nhằm giải quyết vấn đề thư rác. Trong phương pháp này, thư điện tử được phân loại thành hai mức độ ưu tiên là thư rác và thư hợp lệ, trong đó thư hợp lệ có thứ tự ưu tiên cao hơn thư rác. Hướng nghiên cứu về lọc thư rác được chia thành các nhóm

phương pháp khác nhau, trong đó có một nhóm các phương pháp lọc thư rác dựa trên nền tảng SpamAssassin. SpamAssassin là nền tảng lọc thư rác dựa trên luật có trọng số được ứng dụng rộng rãi trong thực tế. Đã có nhiều phương pháp xây dựng tập luật được đề xuất dành cho SpamAssassin, nhưng việc lựa chọn luật và gán điểm số cho luật vẫn được thực hiện tách rời nhau, dẫn đến tập luật tìm được chưa thực sự tối ưu. Từ đó, luận án đặt ra câu hỏi nghiên cứu thứ nhất: "*Làm thế nào để đồng thời lựa chọn đặc trưng và gán điểm số cho tập luật SpamAssassin?*".

Dự đoán hành động người dùng là một dạng của bài toán xác định thứ tự ưu tiên của thư điện tử nhằm giải quyết vấn đề quá tải thư điện tử gây ra bởi số lượng thư hợp lệ quá lớn. Trong bài toán này, thư điện tử được phân loại dựa trên hành động mà người dùng có khả năng cao nhất sẽ thực hiện với mỗi bức thư, giúp người dùng nhanh chóng tìm được các bức thư cần xử lý. Số lượng mức độ ưu tiên có thể thay đổi tùy theo từng phương pháp, nhưng thường là từ ba mức độ trở lên. Nhận thấy SpamAssassin đã và đang được sử dụng trong các hệ thống máy chủ thư điện tử để lọc thư rác nhưng nền tảng này chưa có tính năng dự đoán hành động. Nếu có thể bổ sung tính năng dự đoán hành động cho SpamAssassin thì việc triển khai tính năng này trên những hệ thống máy chủ thư điện tử sẽ trở nên dễ dàng hơn. Từ đó, luận án đặt ra câu hỏi nghiên cứu thứ hai: "*Làm thế nào để dự đoán thư điện tử theo hành động người dùng trên nền tảng SpamAssassin?*".

Một dạng khác của bài toán xác định thứ tự ưu tiên của thư điện tử là xếp hạng thư điện tử, nhằm giải quyết vấn đề quá tải thư điện mà nguyên nhân là thư hợp lệ. Trong bài toán này, một bức thư được phân loại dựa trên tầm quan trọng của nó đối với người sử dụng. Nói theo cách khác, các mức độ ưu tiên trong phương pháp này thể hiện tầm quan trọng mang tính cá nhân hóa của thư điện tử. Những nghiên cứu trước đó về xếp hạng thư điện tử đạt được độ chính xác chưa cao. Hơn nữa, vấn đề khan hiếm dữ liệu huấn luyện vẫn còn tồn tại và là một ràng buộc của bài toán. Vì vậy, luận án đặt ra câu hỏi nghiên cứu thứ ba: "*Làm thế nào để xây dựng mô hình xếp hạng thư điện tử với độ chính xác cao hơn những mô hình hiện tại?*".

Mục tiêu chung của luận án là nghiên cứu các phương pháp xác định thứ tự ưu tiên của thư điện tử Tiếng Việt. Mục tiêu này được thể hiện ở những mục tiêu cụ thể sau:

(1) Để tìm câu trả lời cho câu hỏi thứ nhất, luận án tiến hành nghiên cứu và đề xuất phương pháp tự động sinh tập luật lọc thư rác cho nền tảng SpamAssassin. Phương pháp đề xuất sẽ cho phép đồng thời lựa chọn luật và gán điểm số cho luật, từ đó sinh được tập luật tối ưu hơn so với phương pháp cũ.

(2) Để tìm câu trả lời cho câu hỏi thứ hai, luận án tiến hành nghiên cứu và đề xuất phương pháp dự đoán hành động người dùng dựa trên nền tảng SpamAssassin. Phương pháp đề xuất trong luận án được thiết kế để dự đoán ba hành động là "trả lời", "đọc" và "xóa". Phương pháp này cho phép SpamAssassin thực hiện tính năng dự đoán hành động bằng cách kết hợp nhiều tập luật lọc thư rác. Kết quả dự đoán của mô hình phụ thuộc vào cách lựa chọn của người dùng về hành động cần thực hiện đối với thư điện tử.

(3) Để tìm câu trả lời cho câu hỏi thứ ba, luận án tiến hành nghiên cứu và đề xuất phương pháp xếp hạng thư điện tử với năm mức độ ưu tiên, ứng dụng các kỹ thuật phân loại tiên tiến và tập đặc trưng phong phú nhằm đạt được độ chính xác dự đoán cao hơn so với các phương pháp cũ. Nghiên cứu này cũng sẽ được thực hiện dưới sự ràng buộc về số lượng dữ liệu huấn luyện hạn chế.

Phạm vi nghiên cứu của luận án là sử dụng các phương pháp phân loại để giải quyết ba dạng nói trên của bài toán xác định thứ tự ưu tiên của thư điện tử. Đối với bài toán lọc thư rác và dự đoán hành động người dùng, phạm vi nghiên cứu là các phương pháp có thể ứng dụng trên nền tảng SpamAssassin. Tuy nghiên cứu về xác định thứ tự ưu tiên của thư điện tử trên thế giới đã được thực hiện nhiều đối với những ngôn ngữ phổ biến như tiếng Anh, tiếng Trung, nghiên cứu dành cho tiếng Việt còn hạn chế về số lượng. Trong khi đó, các hệ thống xác định thứ tự ưu tiên của thư điện tử sẽ đem lại lợi ích thiết thực cho người sử dụng thư điện tử tại Việt Nam. Vì vậy, luận án xác định đối tượng nghiên cứu là thư điện tử tiếng Việt.

4. PHƯƠNG PHÁP NGHIÊN CỨU

Để đạt được những mục tiêu đã đề ra, luận án vận dụng các phương pháp nghiên cứu cơ sở lý thuyết, kế thừa kết quả nghiên cứu, phân tích thực nghiệm và so sánh, đối chứng kết quả thí nghiệm. Trước tiên, luận án tham khảo và trình bày các kiến thức nền tảng có liên quan đến đối tượng nghiên cứu là thư điện tử tiếng Việt để hỗ trợ cho nghiên

cứu của luận án. Các tài liệu tham khảo tập trung chủ yếu vào các bài toán và phương pháp phân loại và xác định thứ tự ưu tiên của thư điện tử đã công bố. Từ đó rút ra các kết quả nghiên cứu có giá trị và các vấn đề còn tồn đọng. Tiếp đó, luận án kế thừa kết quả của các nghiên cứu được tham khảo đồng thời đề xuất các phương pháp mới để giải quyết các vấn đề còn tồn đọng. Các thí nghiệm được thực hiện đối với các phương pháp đề xuất và kết quả thực nghiệm được phân tích để rút ra được các kết luận. Kết quả thí nghiệm trên phương pháp đề xuất sẽ được đánh giá, so sánh về mặt định lượng cũng như về mặt định tính với những nghiên cứu đã công bố có liên quan.

5. CÁC ĐÓNG GÓP CỦA LUẬN ÁN

Đóng góp thứ nhất của luận án là đề xuất phương pháp tự động sinh tập luật cho SpamAssassin dựa trên mạng nơ-ron để tăng độ chính xác cho bộ lọc lọc thư rác dựa trên SpamAssassin. Phương pháp đề xuất bao gồm các bước: tiền xử lý dữ liệu, biểu diễn dữ liệu, thiết kế mô hình mạng nơ-ron, huấn luyện mạng nơ-ron và tạo tập luật SpamAssassin. Tập đặc trưng được lựa chọn, cập nhật và gán điểm số một cách đồng thời trong quá trình huấn luyện mạng nơ-ron nói trên, thay vì thực hiện tách rời nhau trong các phương pháp cũ [28, 62]. Mục tiêu của phương pháp là tìm ra tập đặc trưng có hiệu quả phân loại tốt nhất và gán điểm số tối ưu cho tập đặc trưng đó. Cách làm này giải quyết hạn chế của các phương pháp cũ đó là chỉ lựa chọn một tập đặc trưng duy nhất và không so sánh với các tập đặc trưng khác, dẫn đến chưa kiểm chứng được hiệu quả của tập đặc trưng được chọn trên dữ liệu.

Đóng góp thứ hai của luận án là đề xuất phương pháp dự đoán hành động người dùng dựa trên nền tảng SpamAssassin. Trong phương pháp đề xuất, các mô hình phân loại đa lớp OVA, OVO, DAG đã được sử dụng để kết hợp nhiều tập luật SpamAssassin thành các máy phân loại đa lớp, cho phép SpamAssassin gợi ý cho người dùng hành động cần được thực hiện trên một bức thư. Phương pháp này khắc phục hạn chế của các hệ thống thư điện tử sử dụng nền tảng SpamAssassin là chưa có tính năng dự đoán hành động cần thực hiện trên thư điện tử cho người dùng.

Đóng góp thứ ba của luận án là đề xuất phương pháp xếp hạng thư điện tử với năm mức độ ưu tiên dựa trên phương pháp học sâu nhằm giải quyết vấn đề quá tải thư điện tử. Phương pháp đề xuất khai thác đồng thời nhóm đặc trưng nội dung và đặc trưng xã

hội từ dữ liệu của người dùng. Nhóm nội dung đặc trưng được biểu diễn bằng phương pháp word embedding nhằm biểu diễn ngữ nghĩa của văn bản tốt hơn so với các phương pháp cũ. Một mô hình học sâu kết hợp các cấu trúc mạng nơ-ron hồi quy và mạng nơ-ron truyền thẳng, cùng kỹ thuật Dropout trong huấn luyện đã được đề xuất. Với các cải tiến nói trên, phương pháp đề xuất có độ chính xác dự đoán cao hơn so với phương pháp cũ và có thể được áp dụng để xây dựng ứng dụng xếp hạng thư điện tử độc lập, không phụ thuộc vào nền tảng SpamAssassin.

6. BỐ CỤC CỦA LUẬN ÁN

Với các mục tiêu nêu trên, luận án được cấu trúc gồm bốn chương:

- Chương 1 – Tổng quan về thư điện tử và xác định thứ tự ưu tiên của thư điện tử.
- Chương 2 – Phát hiện thư rác.
- Chương 3 – Dự đoán hành động người dùng thư điện tử.
- Chương 4 – Xếp hạng thư điện tử.

Chương 1 bao gồm những kiến thức nền tảng về thư điện tử, cung cấp cái nhìn tổng quan về thư điện tử. Trong Chương 1, hệ thống thư điện tử, các đặc điểm của thư điện tử, thư rác, các bài toán xác định thứ tự ưu tiên của thư điện tử sẽ được giới thiệu. Các tập dữ liệu về thư điện tử được mô tả. Một số nghiên cứu liên quan đến các bài toán về thư điện tử được chọn lọc và tóm tắt.

Chương 2 tập trung vào bài toán phát hiện thư rác. Trong Chương 2, luận án đề xuất một phương pháp mới để sinh tập luật dành cho SpamAssassin và thực hiện thí nghiệm trên một tập dữ liệu lọc thư rác tiếng Việt.

Chương 3 thảo luận bài toán dự đoán hành động người dùng đối với thư điện tử. Trong chương này, luận án đề xuất phương pháp ứng dụng SpamAssassin để dự đoán hành động người dùng. Tập dữ liệu dự đoán hành động người dùng được phát triển trên nền tảng tập dữ liệu lọc thư rác ở Chương 2.

Chương 4 tìm hiểu bài toán xếp hạng thư điện tử với năm mức độ ưu tiên nhằm mang lại cho người dùng kết quả dự đoán tầm quan trọng của thư điện tử chính xác, cụ thể hơn. Chương 4 đề xuất áp dụng các kỹ thuật học sâu cho bài toán này và thực hiện thí nghiệm so sánh phương pháp đề xuất với một số phương pháp trước đó.

CHƯƠNG 1 – TỔNG QUAN VỀ THƯ ĐIỆN TỬ VÀ XÁC ĐỊNH THỨ TỰ ƯU TIÊN CỦA THƯ ĐIỆN TỬ

Chương này trình bày những vấn đề tổng quan về xác định thứ tự ưu tiên của thư điện tử, sự cấp thiết của vấn đề nghiên cứu, các phương pháp và tập dữ liệu đã được sử dụng. Trước tiên, những khái niệm cơ bản về thư điện tử được đề cập. Tiếp theo, các bài toán về xác định thứ tự ưu tiên của thư điện tử được định nghĩa cụ thể. Sau đó, luận án tổng hợp các nghiên cứu về các bài toán nói trên, những thành tựu đã đạt được cùng với những vấn đề còn tồn tại. Cuối cùng, một số vấn đề quan trọng mà luận án sẽ tập trung giải quyết sẽ được trình bày trong phần kết luận chương.

1.1. HỆ THỐNG THƯ ĐIỆN TỬ

1.1.1. Sơ lược về thư điện tử

Thư điện tử là phương tiện liên lạc được ra đời sớm nhất trên mạng máy tính và đã được sử dụng từ trước khi mạng Internet xuất hiện cho đến ngày nay. Không có một tác giả cụ thể nào phát minh ra thư điện tử [16] mà chuẩn thư điện tử đồ sộ hiện giờ đã được phát triển dần từ những thông điệp có cấu trúc rất đơn giản. Những bức thư điện tử đầu tiên có dạng tệp văn bản và được gửi đi giữa những người dùng trên cùng máy tính. Hình thức này được áp dụng từ năm 1965 tại học viện MIT và được đặt tên là MAILBOX. Khi mạng ARPANET, tiền thân của Internet, ra đời thì nhu cầu gửi thư điện tử qua mạng nhanh chóng xuất hiện. Cần có hệ thống thư điện tử phức tạp hơn có thể làm điều đó. Ray Tomlinson là người đã xây dựng chuẩn thư điện tử đầu tiên vào năm 1972. Ông nổi tiếng với quy tắc sử dụng cấu trúc *ten_nguoi_dung@ten_may_tinh* để thể hiện địa chỉ hòm thư của người gửi và người nhận. Thư điện tử là ứng dụng chủ yếu duy trì sự tồn tại của ARPANET với khoảng vài trăm người dùng trong quân đội Hoa Kỳ vào năm 1974. Trong cùng khoảng thời gian từ 1974 tới 1975, Larry Roberts áp dụng việc chia hòm thư thành các thư mục. Sau đó, các tính năng của thư điện tử được phát triển thêm bởi nhiều cá nhân, hình thành một hệ thống tiêu chuẩn phức tạp. Hệ thống thư điện tử được sử dụng hiện nay là sự kết hợp giữa các giao thức SMTP, POP3 và IMAP. Những giao thức này được phát minh từ những năm 80 của thế kỷ 20 bởi nhiều tác giả và được liên tục duy trì, cập nhật cho đến ngày nay.

Thư điện tử phát triển nhanh chóng, thúc đẩy sự ra đời của mạng Internet toàn cầu. Một trong những phần mềm thương mại ra đời đầu tiên là Eudora (1988). Không lâu sau, hệ thống Pegasus Mail xuất hiện (1990). Sự phổ biến của mạng Internet toàn cầu đã dẫn đến sự ra đời của các dịch vụ cung cấp thư điện tử lớn và miễn phí như AOL Mail (1993), Hotmail (1996) và Yahoo (1997). Dịch vụ thư điện tử miễn phí lớn nhất ngày nay, Gmail, xuất hiện khá lâu về sau, vào năm 2004.

Mặc dù mạng toàn cầu phát triển mạnh và có khả năng phục vụ việc trao đổi thông tin thông qua giao thức HTTP, thư điện tử vẫn là ứng dụng quan trọng và được sử dụng nhiều nhất của Internet. Năm 2004 có hơn 600 triệu người sử dụng thư điện tử trên toàn thế giới [16]. Cho đến năm 2019, số người dùng đã tăng lên con số khổng lồ 3,93 tỷ người và dự tính vẫn tiếp tục tăng[6] trung bình 2.7% mỗi năm cho tới năm 2024. Theo thống kê của Radicati[7], lượng thư được gửi đi mỗi ngày vào năm 2019 là 293,6 tỷ và dự đoán tới năm 2024 sẽ lên tới 361,6 tỷ, tốc độ tăng trung bình 4,3% mỗi năm.

1.1.2. Cấu trúc của một bức thư điện tử

Tiêu chuẩn mới nhất về cấu trúc của thư điện tử được định nghĩa trong RFC 5322 [38]. Thư điện tử là một tập tin văn bản thuần túy. Một bức thư bao gồm các trường header (gộp chung thành "phần header" của bức thư). Theo sau phần header là phần nội dung thư, phần này có thể có nội dung hoặc để trống. Các trường header là những dòng bắt đầu bằng tên trường, theo sau bởi một dấu hai chấm (":"), tiếp đến là giá trị của header.

Sau đây là một số trường header phổ biến:

- *Message-ID*: chuỗi định danh duy nhất của bức thư.
- *From*: địa chỉ hòm thư của (những) người soạn ra nội dung thông điệp.
- *Sender*: địa chỉ hòm thư của người thực hiện việc gửi thư (nếu người gửi thư không phải là người soạn thư).
- *Reply-To*: địa chỉ (những) hòm thư mà bức thư cần được phản hồi tới.
- *In-Reply-To*: Message-ID của bức được trả lời.

[6] https://www.statista.com/statistics/255080/number-of-e-mail-users-worldwide/
[7] https://www.statista.com/statistics/456500/daily-number-of-e-mails-worldwide/

- *References*: một tập hợp Message-ID của những bức thư liên quan, thường là những bức thư trong cùng chuỗi thư trao đổi qua lại (thread).
- *To*: địa chỉ hòm thư của (những) người mà nội dung bức thư hướng tới.
- *Cc*: địa chỉ hòm thư của (những) người nhận bản sao của bức thư.
- *Bcc*: địa chỉ hòm thư của (những) người nhận bản sao của bức thư nhưng danh tính của họ không được công bố cho những người cùng nhận thư.
- *Subject*: Tiêu đề bức thư.
- *Date*: thời gian bức thư được hoàn thành và sẵn sàng để gửi đi.

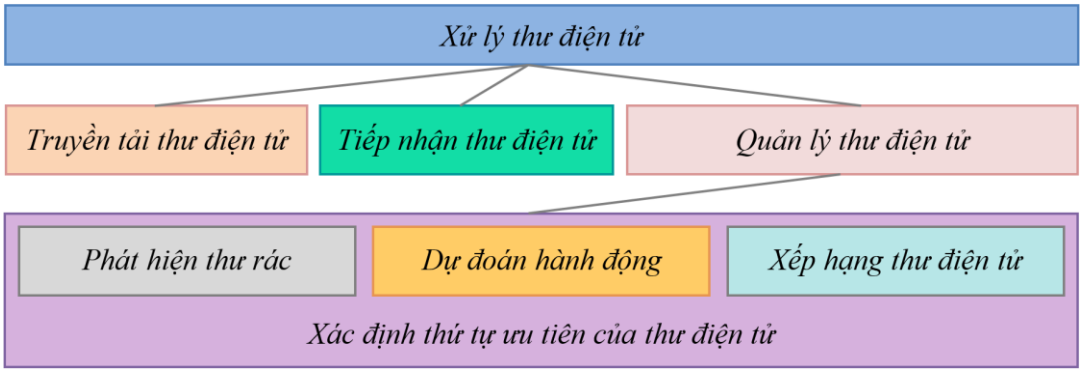
Các trường *Message-ID*, *In-Reply-To* và *References* được sử dụng để xác định chuỗi thư trao đổi (trả lời, chuyển tiếp). Phần nội dung thư là tập hợp của nhiều dòng ký tự, với những quy định sau:

- Ký tự CR (giá trị 13) và ký tự LF (giá trị 10) phải xuất hiện cùng nhau để tạo thành ký tự xuống dòng (CRLF), không được xuất hiện riêng lẻ.
- Một dòng trong phần nội dung không được dài quá 998 ký tự và nên được hạn chế trong vòng 78 ký tự, không tính CRLF.

Nội dung thư ngày nay được chia thành nhiều phần (multipart) trong đó thường có một phần là nội dung thư ở dạng văn bản thuần túy (Content-Type: text/plain) và một phần là nội dung thư định dạng HTML (Content-Type: text/html). Các phần khác của bức thư thường là các tệp đính kèm với kiểu dữ liệu MIME cụ thể (ví dụ: *image/jpeg*, *application/zip*). Tuy nhiên, sự linh hoạt trong tiêu chuẩn về thư điện tử cũng cho phép một bức thư có cấu tạo đơn giản gồm một phần (singlepart).

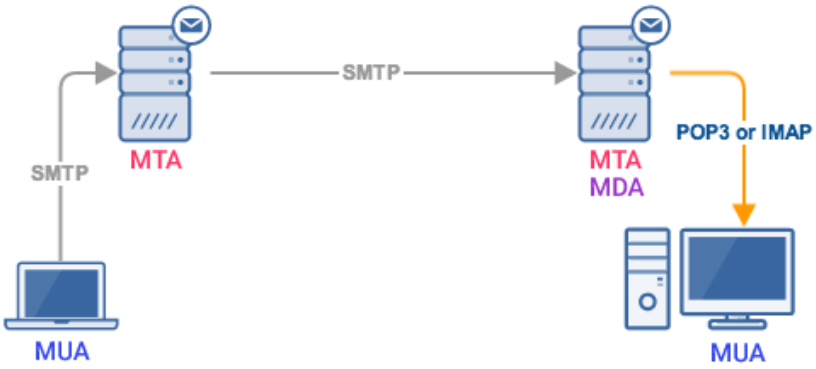
Có thể thấy rõ, thư điện tử có những đặc trưng không tồn tại trong văn bản thông thường như các trường header, người gửi, người nhận… Thư điện tử cũng khác biệt với văn bản thông thường vì nội dung thư thường bao gồm những ký tự trang trí, ký hiệu, ký tự đặc biệt… Ngoài ra, ngôn ngữ sử dụng trong thư điện tử cũng có thể không tuân thủ nghiêm ngặt các quy tắc về đánh vần và ngữ pháp. Đối với trường hợp thư rác, kẻ phát tán thường cố tình soạn nội dung bức thư nhằm mục đích đánh lừa các bộ lọc, điều này không xảy ra đối với việc soạn thảo các văn bản thông thường.

1.1.3. Mô hình xử lý thư điện tử



Hình 1.1: Mô hình xử lý thư điện tử tổng quát

Có 3 khâu chính trong hệ thống xử lý thư điện tử (Hình 1.1) là truyền tải, tiếp nhận và quản lý. Trong mỗi khâu lại có nhiều giao thức được xây dựng để quy định việc định dạng văn bản và giao tiếp qua mạng… Tất cả các vấn đề về thư điện tử đều xảy ra ở một hoặc một số bước của cả tiến trình này. Tác vụ lọc thư rác thường được thực hiện ở bước truyền tải và tiếp nhận bởi số lượng thư rác rất lớn, cần phải được loại bỏ trước khi thư rác được truyền đến hòm thư của người sử dụng. Việc lọc thư rác thực hiện càng sớm thì càng tiết kiệm được nhiều tài nguyên tính toán và tài nguyên mạng. Tác vụ dự đoán hành động và xếp hạng thư điện tử thường được thực hiện ở bước quản lý thư điện tử vì mục tiêu của hai bài toán này là sắp xếp, bố trí lại hòm thư của người dùng. Cả ba bài toán lọc thư rác, dự đoán hành động người dùng và xếp hạng thư điện tử đều nằm trong bài toán tổng quát là bài toán xác định thứ tự ưu tiên của thư điện tử.

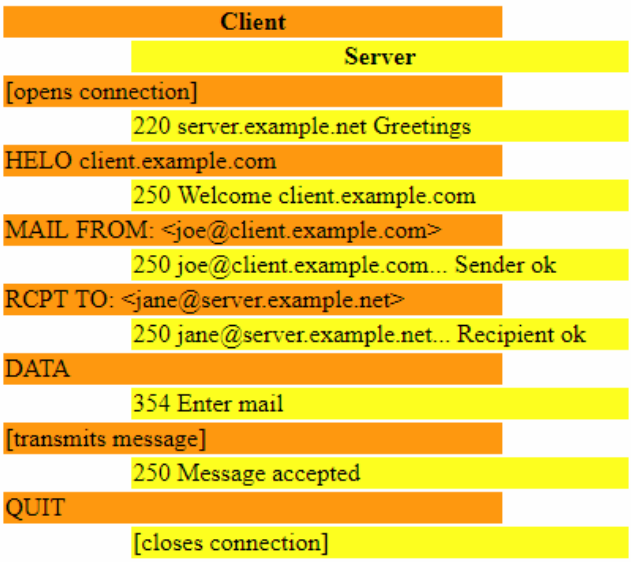


Hình 1.2: Mô hình gửi và nhận thư phổ biến (nguồn: *jscape.com*)

Giao thức truyền tải thư điện tử qua mạng là SMTP. Giao thức này được sử dụng bởi đơn vị truyền tải thư điện tử (MTA), một hệ thống có nhiệm vụ chuyển các bức thư từ máy chủ của người gửi đến máy chủ của người nhận. Giao thức SMTP có hai thành

phần là phần *server* (bên gửi) và phần *client* (bên nhận). Hệ thống MTA tích hợp phần *server* của giao thức SMTP. MUA là phần mềm giúp người dùng gửi và nhận thư điện tử. Trong MUA có tích hợp phần *client* của giao thức SMTP dùng để gửi thư từ máy tính cá nhân lên máy chủ gửi thư MTA.

SMTP là giao thức duy trì kết nối và là giao thức dựa trên văn bản, khác với các giao thức dùng dữ liệu nhị phân. Trong quá trình trao đổi thông điệp, MTA và MUA giao tiếp với nhau bằng chuỗi lệnh SMTP. Ví dụ về các thông điệp khi gửi một bức thư được mô tả trong Hình 1.3. Ở bất kỳ bước nào trong giao dịch đó, *server* có thể gửi trả *client* một thông báo từ chối và giao dịch sẽ kết thúc. Điều này có ứng dụng thực tế để chặn thư rác sớm từ ngay khi vừa phát hiện, tránh làm hao tốn tài nguyên của máy chủ. Các phương pháp lọc thư rác dựa vào địa chỉ IP có thể chặn ngay từ bước bắt đầu kết nối bởi vì ở thông điệp này ta đã biết địa chỉ hòm thư của người gửi. Một phương pháp lọc thư rác dựa theo địa chỉ của người gửi phải chờ đến thông điệp MAIL FROM để có thể quyết định có chặn bức thư hay không.



Hình 1.3: Các thông điệp khi sử dụng giao thức SMTP để gửi một bức thư (nguồn: *smtp2go.com*)

Trong khi SMTP có thể được hiểu là chiếc xe đưa thư đảm nhận việc vận chuyển thư từ bưu cục này đến bưu cục khác thì MDA có thể được hiểu là người đưa thư từ bưu điện địa phương tới hòm thư của người nhận. Nhiệm vụ chính của MDA là nhận thư từ MTA và lưu trữ nó vào đúng hòm thư của người nhận. Vai trò của MDA được miêu tả trong Hình 1.2 và MDA giao tiếp với MUA thông qua một trong hai giao thức là POP

và IMAP. Một phương pháp lọc thư rác có thể được tích hợp vào MDA để chặn những bức thư không mong muốn trước khi chúng kịp vào tới hòm thư của người dùng.

Công đoạn gần với người sử dụng nhất là quản lý thư điện tử, bao gồm bố trí, lưu trữ, hiển thị thư cho người dùng. Một hòm thư điện tử thường được chia thành nhiều thư mục. Những thư mục phổ biến nhất là: inbox (hòm thư nhận), sent (thư đã gửi), trash (thư đã xóa) và spam (thư rác). Giao thức IMAP hỗ trợ gắn nhãn để theo dõi trạng thái của thư (đã đọc, đã trả lời, đã xóa…). Chức năng được hỗ trợ cụ thể phụ thuộc vào cách triển khai chi tiết của từng phần mềm MUA.

1.1.4. Sơ lược về thư rác

Bởi vì thư điện tử có chi phí thấp, độ tin cậy cao và tốc độ truyền tải nhanh nên mức độ sử dụng thư điện tử ngày một tăng. Số lượng thư trao đổi trên mạng Internet ngày càng tăng kéo theo việc một số người sử dụng thư điện tử đã khai thác nó trên quy mô lớn nhằm phục vụ cho mục đích thương mại, quảng cáo, và một số mục đích xấu. Hành vi nói trên được gọi là gửi hoặc phát tán thư rác. Hiện nay vẫn chưa có một định nghĩa hoàn chỉnh, chặt chẽ về thư rác. Có quan điểm coi thư rác là những thư quảng cáo không được yêu cầu (UCE), có quan điểm rộng hơn cho rằng thư rác bao gồm thư quảng cáo, thư quấy rối, và những thư có nội dung không lành mạnh không được người dùng mong muốn và được gửi với số lượng lớn (UBE). Định nghĩa thông dụng nhất về thư rác là những bức thư điện tử không yêu cầu, không mong muốn và được gửi hàng loạt tới người nhận. Một số định nghĩa còn nhấn mạnh rằng thư rác là thư được gửi đi một cách bừa bãi và người gửi thư không có quan hệ với người nhận [23].

1.2. CÁC BÀI TOÁN XÁC ĐỊNH THỨ TỰ ƯU TIÊN CỦA THƯ ĐIỆN TỬ

Một nghiên cứu tổng quan về các bài toán phân loại thư điện tử [72] đã đưa ra danh sách 15 ứng dụng của các phương pháp phân loại thư điện tử, trong đó có phân loại thư rác, phân loại thư theo mục, phát hiện thư lừa đảo, thư thú vị / không thú vị, thư riêng tư / thư công việc… Danh sách này không bao gồm đầy đủ các bài toán về xác định thứ tự ưu tiên của thư điện tử mà luận án sẽ thảo luận. Nghiên cứu [72] chỉ khảo sát các bài báo khoa học gắn với các từ khóa: “Email Classification”, “Email Classification”, “E-mail Classification”, “Email Categorization”, “E-mail Categorization”, “Spam Email Detection” and “Phishing Email Detection”. Một số từ khóa quan trọng để tìm kiếm các

bài toán về thư điện tử mà nghiên cứu [72] không sử dụng có thể kể đến: “anti-spam”, “spam filtering”, “email prioritization”, “email triage”, “email action prediction”, “email priority”, “email sorting”, “email ranking”, “email reply prediction”. Điều này dẫn đến danh sách các nghiên cứu được tổng hợp trong [72] chưa đầy đủ. Do đó, một hướng nghiên cứu quan trọng là xác định thứ tự ưu tiên của thư điện tử, trong tiếng Anh thường được đặt tên là “email prioritization” đã không được đưa vào thành một hướng nghiên cứu về thư điện tử. Luận án không đặt ra mục tiêu khảo sát các hướng nghiên cứu về thư điện tử. Phạm vi nghiên cứu của luận án là các phương pháp xác định thứ tự ưu tiên của thư điện tử. Vì vậy, trong phần này, luận án sẽ đưa ra định nghĩa các dạng bài toán xác định thứ tự ưu tiên của thư điện tử để làm rõ thêm cho phạm vi nghiên cứu.

Trước hết, xác định thứ tự ưu tiên không được coi là tập con của bài toán phân loại thư điện tử. Ngoài phương pháp phân loại, bài toán này còn được giải quyết bằng phương pháp hồi quy. Giữa bài toán phân loại thư điện tử và bài toán xác định thứ tự ưu tiên của thư điện tử tồn tại một bài toán con tương đồng là bài toán lọc thư rác.

Xác định thứ tự ưu tiên của thư điện tử là tên gọi chung để chỉ tất cả các bài toán có mục tiêu sắp xếp thư điện tử theo mức độ ưu tiên. Ở dạng tiêu biểu của bài toán, tập kết quả dự đoán bao gồm các tên gọi thể hiện mức độ ưu tiên tương đối với nhau, ví dụ như “không quan trọng”, “rất quan trọng”, “xử lý gấp”, “không cần xử lý gấp” hoặc “mức độ 1”, “mức độ 2”… *Lọc thư rác* là một bài toán con của bài toán này, với hai mức độ ưu tiên là thư hợp lệ (được ưu tiên) và thư rác (không được ưu tiên). Bài toán dự đoán thư cần trả lời [44, 74, 82] là một dạng khác của bài toán xác định thứ tự ưu tiên của thư điện tử với hai mức độ ưu tiên. Khi mức độ ưu tiên được định nghĩa là các hành động của người dùng đối với bức thư, ta có bài toán *dự đoán hành động của người dùng thư điện tử,* gọi vắn tắt là bài toán *dự đoán hành động*. Khi có nhiều hơn hai mức độ ưu tiên và các mức độ ưu tiên thể hiện tầm quan trọng của bức thư, bài toán được gọi là *xếp hạng thư điện tử*.

1.2.1. Lọc thư rác

Lọc thư rác là một trong những bài toán xác định thứ tự ưu tiên của thư điện tử với hai mức độ ưu tiên. Một hệ thống lọc thư rác được xây dựng để giải quyết vấn đề quá tải thư điện tử có nguyên nhân chính đến từ thư rác. Đầu vào của bài toán lọc thư rác

trong luận án là một tập tin thư điện tử có định dạng được mô tả trong mục 1.1.2. Trong tập tin này, hai thông tin dạng văn bản được trích xuất là tiêu đề thư và nội dung thư. Đầu ra của bài toán là một giá trị nhị phân thể hiện bức thư là thư rác hay thư hợp lệ, với quy ước giá trị 1 là thư rác và giá trị 0 là thư hợp lệ.

1.2.2. Dự đoán hành động của người dùng thư điện tử

Dự đoán hành động người dùng là một dạng của bài toán xác định thứ tự ưu tiên của thư điện tử, được phân biệt với các bài toán khác ở đặc điểm kết quả dự đoán là một hành động của người dùng. Mục tiêu bài toán này là gợi ý một trong một số hữu hạn hành động được định nghĩa sẵn mà người dùng cần thực hiện đối với một bức thư nhận. Bài toán này nằm trong nhóm các bài toán phân loại, không đặt ra ràng buộc về quan hệ tương đối giữa các hành động. Bài toán được định nghĩa cụ thể như sau.

Gọi tập hợp tất cả các bức thư mà người dùng nhận được là tập E. Gọi tập hợp các hành động là A = $\{a_1, a_2, a_n\}$, ($n \geq 2$). Ta cần tìm một hàm dự đoán hành động sao cho đầu vào là một bức thư và đầu ra là hành động cần làm với bức thư đó:

$$f(m): \mathrm{E} \rightarrow \mathrm{A}$$

Phương pháp chung để giải bài toán này bằng kỹ thuật học máy là định nghĩa một tập dữ liệu huấn luyện M = $\{m_1, m_2, \ldots, m_n\}$. M là tập con của E. Mỗi bức thư trong tập M được người dùng gán cho một hành động phù hợp nhất với nó từ tập A. Đó là quá trình gán nhãn cho tập dữ liệu. Tập M đã được gán nhãn được dùng để huấn luyện một mô hình học máy. Kết quả huấn luyện là một mô hình có chức năng gần giống với chức năng của hàm $f(m)$ nói trên. Một số dạng của bài toán dự đoán hành động đã được nghiên cứu đó là gợi ý trả lời thư [44], dự đoán một trong ba hành động phổ biến (*trả lời, đọc, xóa*) [51], và phát hiện hành động (*lưu trữ, trả lời*) [25]. Với cách tiếp cận của nghiên cứu [25], cả hai hành động có thể đồng thời xảy ra, nghĩa là người dùng có thể thực hiện các hành động trả lời và lưu trữ trên cùng một bức thư.

1.2.3. Xếp hạng thư điện tử

Bài toán xếp hạng thư điện tử có mục tiêu chính là đánh giá tầm quan trọng của thư điện tử, nhằm sắp xếp các bức thư theo thứ tự tầm quan trọng. Việc này giúp cải thiện hiệu quả sử dụng thư điện tử của người dùng, từ đó giải quyết vấn đề quá tải thư. Để

làm được điều này, bài toán cần phải đánh giá tầm quan trọng của từng bức thư. Có hai hướng tổng quát để dự đoán tầm quan trọng của một bức là phân loại và hồi quy. Ta cần tìm một hàm dự đoán có dạng:

$$g(m)\text{: E} \rightarrow \text{P}$$

Phương pháp phân loại giả thiết rằng các mức độ quan trọng là rời rạc, có thể có hoặc không có quan hệ tương đối (lớn hơn, nhỏ hơn) với nhau. Trong trường hợp này, P là một tập hợp hữu hạn các giá trị rời rạc. Kích thước của tập P là 2 đối với bài toán phát hiện thư rác. Đối với bài toán dự đoán hành động người dùng, tập P có thể chứa từ hai hành động [44] hoặc nhiều hơn hai hành động [51]. Ngược lại, phương pháp hồi quy giả thiết rằng các mức độ quan trọng là liên tục và có quan hệ tương đối với nhau. Khi đó, P là một tập số thực liên tục. Một nghiên cứu về xếp hạng thư điện tử vào năm 2005 [11] là ví dụ về giải quyết bài toán theo hướng hồi quy.

Khác với bài toán phân loại thư điện tử, bài toán xếp hạng thư điện tử tập trung vào mô phỏng mức độ quan trọng của các bức thư đối với người dùng. Trong khi đó, tầm quan trọng của bức thư không làm ảnh hưởng tới việc lựa chọn nhãn trong bài toán phân loại theo thư mục.

Hiện tại, một số giải pháp đã được đưa ra để xếp hạng thư điện tử với các thuật toán và tiêu chí đánh giá kết quả khác nhau [11, 40, 49]. Tuy vậy, nhìn chung bài toán xếp hạng thư điện tử vẫn chưa được giải quyết triệt để. Thí nghiệm của bài báo [11] cho thấy trung bình sai số của phương pháp xếp hạng đối với 236 bức thư dùng để thử nghiệm là 33.1 với độ lệch chuẩn là 29.1. Những con số này cho thấy mức độ sai số trong xếp hạng còn lớn và giá trị độ lệch chuẩn cao cho thấy có những bức thư được xếp hạng sai lệch xa so với thứ tự thực tế của chúng. Từ kết quả thí nghiệm của nghiên cứu [49], sai số trung bình thấp nhất đạt được là khoảng 0.8. Với 5 mức độ ưu tiên được sử dụng nghiên cứu này, tuy chỉ số accuracy không được công bố nhưng ta có thể tính được khoảng giá trị của accuracy dựa theo sai số đó. Trường hợp có chỉ số accuracy tệ nhất: có khoảng 80% bức thư được dự đoán sai với sai số là 1. Trường hợp có chỉ số accuracy tốt nhất: có 20% bức thư được dự đoán sai với sai số là 4. Vậy, chỉ số accuracy trong trường hợp tốt nhất tương ứng với trung bình sai số 0.8 là 80% và trong trường hợp xấu nhất là 20%. Với việc sử dụng thêm nhiều đặc trưng xã hội, nghiên cứu [46]

cho thấy hiệu quả cao hơn, với trung bình sai số đạt được khoảng 0.67, trên tập dữ liệu nhỏ hơn so với tập dữ liệu trong [49]. Giá trị trung bình sai số này tương ứng với accuracy tối đa là 83.25%. Các kết quả nói trên cho thấy bài toán xếp hạng cần được tiếp tục nghiên cứu.

1.3. TỔNG QUAN NGHIÊN CỨU VỀ XÁC ĐỊNH THỨ TỰ ƯU TIÊN CỦA THƯ ĐIỆN TỬ

1.3.1. Nghiên cứu về lọc thư rác

Bộ lọc thư rác có chức năng tự động phát hiện thư rác với mục đích ngăn ngừa sự truyền tải thư rác. Tính đến nay, nhiều phương pháp khác nhau đã được đề xuất dành cho việc lọc thư rác. Có nhiều cách để phân loại các phương pháp lọc thư rác. Cách thứ nhất là phân loại dựa trên đặc trưng được sử dụng để phát hiện thư rác. Theo cách này, các phương pháp lọc thư rác thành 3 nhóm: (a) dựa vào đặc trưng nội dung, (b) dựa vào đặc trưng xã hội và (c) dựa vào siêu dữ liệu (được sử dụng bởi các giao thức) của bức thư. Cách thứ hai là phân ra theo hướng tiếp cận: (a) lọc theo danh sách (danh sách đen, trắng, xám, danh sách đen thời gian thực), (b) lọc theo luật (có và không có trọng số), (c) học máy, (d) phương pháp hợp tác, (e) xác thực người gửi. Trong nhóm các kỹ thuật lọc thư rác dựa trên học máy lại chia ra thành các phương pháp học máy có giám sát và không giám sát, trong đó phương pháp học máy có giám sát chiếm đa số. Phương pháp học máy có giám sát được dùng để huấn luyện trực tiếp ra một mô hình có khả năng phân loại thư rác và thư hợp lệ. Cách làm thường thấy trong phương pháp học máy không giám sát là áp dụng thuật toán phân cụm trên tập dữ liệu huấn luyện, sau đó dự đoán nhãn cho một bức thư mới dựa trên khoảng cách của bức thư đó với các trung tâm cụm. Một bộ lọc thư rác có thể thuộc về các phân nhóm khác nhau tùy thuộc vào cách phân loại được sử dụng. Ví dụ, bài báo [28] đề xuất một phương pháp sinh tập luật phát hiện thư rác dành cho SpamAssassin. Theo cách phân loại thứ nhất, phương pháp này được phân vào nhóm bộ lọc dựa vào đặc trưng nội dung. Mặc dù bộ lọc thư rác được xây dựng bởi phương pháp [28] có cơ chế lọc theo luật có trọng số của SpamAssassin, nhưng tập luật của nó được sinh từ dữ liệu với phương pháp học máy có giám sát. Vì vậy, theo cách phân loại thứ hai thì phương pháp này thuộc về nhóm các phương pháp lọc thư rác dựa trên học máy có giám sát.

Bởi vì có nhiều cách để phân loại các phương pháp lọc thư rác, một phương pháp thường không thuộc về một nhóm duy nhất. Do đó, ở phần sau đây, luận án không phân nhóm các phương pháp dựa trên một tiêu chí duy nhất mà chỉ nhóm các phương pháp liên quan một cách linh hoạt nhằm giúp cho người đọc tiện theo dõi.

1.3.1.1. Lọc thư rác trên nền tảng SpamAssassin

SpamAssassin là dự án phần mềm lọc thư rác mã nguồn mở của tổ chức Apache từ năm 2002 cho đến nay. SpamAssassin tương thích với rất nhiều hệ thống thư điện tử, trong đó có những hệ thống phổ biến như procmail, sendmail, Postfix, qmail… SpamAssassin đang được tiếp tục cập nhật với các kế hoạch cho những phiên bản mới. Ưu điểm của SpamAssassin không chỉ nằm ở sự ổn định và hiệu năng hoạt động, mà còn ở sự sẵn có về tài liệu cũng như sự dễ dàng tùy biến với định dạng luật đơn giản, dễ hiểu. So với Sendria, SpamAssassin có mức độ phổ biến cao hơn và cộng đồng lớn hơn. Nhiều tập luật dành cho các ngôn ngữ khác nhau như tiếng Đức, tiếng Hy Lạp, tiếng Ý, tiếng Trung, tiếng Thái, tiếng Thổ Nhĩ Kỳ, tiếng Việt… được đóng góp và duy trì bởi cộng đồng trên toàn thế giới. Vì các lý do nêu trên, luận án đã đặt mục tiêu nghiên cứu các phương pháp xác định thứ tự ưu tiên của thư điện tử trên nền tảng SpamAssassin.

Những bộ lọc thư rác phổ biến nhất trên thế giới có thể được chia thành hai nhóm là các bộ lọc mã nguồn đóng và các bộ lọc mã nguồn mở. Những bộ lọc trong nhóm mã nguồn đóng có thể kể đến SpamTitan, Cisco IronPort, ZEROSPAM, GlockApps. Những bộ lọc này yêu cầu người dùng trả phí bản quyền để có thể sử dụng. Một số bộ lọc như ZEROSPAM hoạt động dưới dạng một dịch vụ điện toán đám mây. Nhóm còn lại bao gồm những bộ lọc mà mã nguồn được công bố, có thể kể đến SpamAssassin, MailScanner và Sendria. Những bộ lọc này là hoàn toàn miễn phí và cũng cho phép người dùng được tùy chỉnh mã nguồn. Tuy nhiên, hệ thống Sendria (trước đây là MailTrap) có lượng tài liệu hạn chế nên việc tùy biến bộ lọc gặp nhiều khó khăn. Để hiểu nguyên lý hoạt động của Sendria, người dùng sẽ phải dành nhiều thời gian để đọc mã nguồn của hệ thống. Hơn nữa, bản chất Sendria là phần mềm máy chủ SMTP nên mục tiêu chính của Sendria là gửi thư chứ không phải là xử lý thư nhận được. MailScanner là một bộ lọc thư rác mã nguồn mở miễn phí khác. Hệ thống này cung cấp

chủ yếu hai nhóm tính năng là lọc thư rác và phát hiện virus. Trong đó, hệ thống sử dụng SpamAssassin để đáp ứng chức năng lọc thư rác. Trên trang chủ của MailScanner (*https://www.mailscanner.info*) đã khẳng định SpamAssassin là nền tảng lọc thư rác "phổ biến" và "chuẩn mực" nhất.

Nền tảng SpamAssassin nằm trong nhóm phương pháp lọc thư rác theo luật có trọng số [13]. SpamAssassin là một nền tảng lọc thư rác mã nguồn mở bao gồm một tập các chương trình lọc và các luật để xác định và đánh dấu thư rác. Để xác định một thư mới đến có phải là thư rác hay không, SpamAssassin so sánh header và nội dung của thư với một tập luật được thiết kế sẵn. Điểm số của những luật mà bức thư "vi phạm" sẽ được cộng vào tổng điểm của bức thư. Từ điểm số thu được, xác định được một thư là thư rác hay thư hợp lệ. Việc gắn điểm số cho các luật từ khóa khiến cho SpamAssassin đạt hiệu quả cao hơn bộ lọc từ khóa đơn giản. Mỗi luật được biểu diễn với 3 thông tin chính: kiểu của luật, từ khóa và điểm số (trọng số) của luật. Hình 1.4 là ví dụ về một luật SpamAssassin.

```
body        MONEY_BACK        /money back guarantee/i
describe    MONEY_BACK        Money back guarantee
score       MONEY_BACK        1.887
```

Hình 1.4: Một luật từ khóa của SpamAssassin áp dụng với phần `body` của bức thư.

Luật này được kích hoạt khi phần nội dung của bức thư có chứa cụm từ "`money back guarantee`". Luật SpamAssassin có thể được biểu diễn dưới dạng biểu thức chính quy để phát hiện được biến thể của các từ khóa, cũng như nhận diện được những dấu hiệu một cách tổng quát hơn so với phương pháp so khớp. Lựa chọn "`/i`" cho phép bỏ qua sự phân biệt ký tự viết hoa và viết thường khi SpamAssassin áp dụng luật đối với nội dung thư. Giá trị điểm số của luật trong ví dụ là 1.887. Ta có thể mô tả việc áp dụng luật r có trọng số s và từ khóa k đối với bức thư m bằng công thức (1.1).

$$f_r(m) = \begin{cases} s, & k \in m \\ 0, & k \notin m \end{cases} \tag{1.1}$$

Mỗi tập luật SpamAssassin có chứa nhiều luật như vậy. Ta biểu diễn một tập gồm n luật dưới dạng $S = \{r_1, r_2, \ldots, r_n\}$. Khi đó, việc áp dụng tập luật S với bức thư m được thực hiện theo công thức (1.2).

$$y = G_S(m) = \sum_{i=1}^{n} f_{r_i}(m) \quad (1.2)$$

Độ lớn của số thực *y* phụ thuộc vào hai yếu tố là số lượng từ khóa của luật được tìm thấy trong bức thư và độ lớn trọng số của những luật đó. Các trọng số *s* của mỗi luật có thể là số âm nên *y* có thể nhận giá trị âm. Để có thể kết luận một bức thư là thư rác hay thư hợp lệ, *y* được so sánh với một số thực *T*, gọi là ngưỡng của tập luật, với giá trị mặc định là $T = 5.0$, nếu $y \geq T$ thì bức thư *m* được coi là thư rác.

```
This mail is probably spam. The original message has been attached
along with this report, so you can recognize or block similar unwanted
mail in the future. See http://spamassassin.org/tag/ for more details.

Content analysis details: (5.03 points, 5 required)
FREE                (1.872 points)
DEAR_FRIEND         (0.732 points)
MONEY_BACK          (1.887 points)
BODY_BEST           (0.539 points)
```

Hình 1.5: Nội dung bức thư bị SpamAssassin đánh dấu là thư rác, bao gồm báo cáo về các luật được áp dụng và bức thư gốc dưới dạng tệp đính kèm.

Cách làm chung để xây dựng tập luật SpamAssassin cho mục đích phát hiện thư rác bao gồm hai tác vụ chính: *xác định tập luật* và *tối ưu tập luật*. Toàn bộ quá trình sinh tập luật có thể được thực hiện một cách thủ công. Tác vụ thứ nhất, xác định tập luật, có mục tiêu là tìm một tập hợp các từ khóa hữu ích để làm cơ sở cho việc dự đoán một bức thư có phải là thư rác hay không. Phương pháp được sử dụng là suy đoán dựa theo kinh nghiệm của chuyên gia và sự quan sát các bức thư rác. Đây là cách làm đã được sử dụng trên thực tế khi các phương pháp sinh tập luật tự động chưa ra đời. Có rất nhiều tập luật được đóng góp bởi cộng đồng người sử dụng SpamAssassin đến từ các quốc gia khác nhau. Trọng số của luật cũng được điều chỉnh một cách thủ công dựa theo sự quan sát của người dùng. Việc này trở nên không quá khó thực hiện đối với người dùng bởi vì SpamAssassin tạo ra một báo cáo (Hình 1.5) về những luật mà một bức thư "vi phạm" mỗi khi đánh dấu một bức thư là thư rác. Những trường hợp cảnh báo nhầm sẽ được xem xét và những luật đóng góp vào kết quả một cách không phù hợp sẽ được giảm trọng số sao cho điểm số *y* bức thư đó giảm xuống nhỏ hơn ngưỡng *T*. Quá trình thủ công này chính là việc tối ưu hóa tập luật, được lặp lại nhiều lần, bởi nhiều người dùng.

Hiệu quả của tập luật phụ thuộc vào chất lượng của những luật từ khóa mà người dùng xác định được, cũng như độ chính xác của việc điều chỉnh trọng số luật thủ công.

Một cách sinh tập luật khác đó là kết hợp giữa xác định tập luật thủ công với tối ưu tập luật bằng phương pháp học máy. Thay vì điều chỉnh trọng số luật sau khi quan sát các trường hợp cảnh báo nhầm thì người dùng SpamAssassin có thể thu thập và gán nhãn một số lượng thư rác và thư hợp lệ để xây dựng một tập dữ liệu. Bộ lọc SpamAssassin đã áp dụng thuật toán tiến hóa GA và sau đó là thuật toán xuống dốc SGD [17] để điều chỉnh trọng số cho tập luật dựa vào dữ liệu. Với cách làm này, kết quả tối ưu trọng số cho tập luật phụ thuộc vào thuật toán tối ưu và tập dữ liệu huấn luyện.

Tác vụ xác định tập luật cũng có thể được thực hiện tự động bằng cách phân tích những bức thư rác [28] hoặc toàn bộ tập dữ liệu, bao gồm cả thư rác và thư hợp lệ [62] để tìm ra những từ khóa hữu ích. Phương án đơn giản nhất để thực hiện việc này là lựa chọn những từ khóa có tần số xuất hiện nhiều nhất. Những từ này phần lớn là từ chức năng không mang nhiều ý nghĩa khi đứng độc lập, được gọi là *stop words*. Có nghiên cứu đề xuất loại bỏ chúng [14] trong khi một nghiên cứu khác lại cho rằng không nên loại bỏ stop words [9]. Phương án thứ hai để xác định tập từ khóa là đánh giá mức độ hữu ích của các từ khóa trong tập dữ liệu. Nghiên cứu [62] đề xuất đánh giá các từ khóa bằng tỷ lệ $R_t = V_{ts} / V_{th}$ trong đó t là một từ khóa, V_{ts} là mức độ liên quan giữa t và thư rác, V_{th} là mức độ liên quan giữa t và thư hợp lệ. Các giá trị V_{ts} và V_{th} được tính toán bằng những cách khác nhau dựa theo lý thuyết xác suất của Bayes. Document Frequency là tỷ lệ thư có chứa từ khóa t trong số những bức thư rác, ký hiệu là DF và được tính theo công thức (1.3). Trong công thức (1.3), DF_t là Document Frequency của từ khóa t, A là sự kiện t xuất hiện trong một bức thư, S là sự kiện bức thư đó là thư rác.

$$DF_t = P(A|S) = \frac{P(A \cap S)}{P(S)} \tag{1.3}$$

$$CP_t = P(S|A) = \frac{P(S \cap A)}{P(A)} \tag{1.4}$$

V_{ts} và V_{th} cũng có thể được tính theo công thức (1.4), trong đó CP_t là xác suất có điều kiện của từ khóa *t*. $P(S|A)$ được hiểu là xác suất mà một bức thư là thư rác khi nó có chứa từ khóa *t*. Khi đó, ý nghĩa của biến V_{ts} là xác suất mà một bức thư là thư rác khi nó có chứa từ khóa *t*. Tương tự, V_{th} có ý nghĩa là xác suất mà bức thư là thư hợp lệ khi nó có chứa từ khóa *t*. Các giá trị khác như thông tin tương hỗ (MI), độ lợi thông tin (IG), phân kỳ Kullback-Leibler cũng được sử dụng trong [28] để lựa chọn đặc trưng cho bộ lọc thư rác.

Tối ưu tập luật có mục tiêu xác định trọng số cho các đặc trưng trong tập luật để đạt được hiệu quả phát hiện thư rác cao nhất trên thực tế. Ở đây, một tập luật được hiểu là tương đương với một bộ lọc thư rác bởi vì nó chứa đựng toàn bộ thông tin cần thiết để xác định một bức thư là thư rác hay thư hợp lệ. Trong phương pháp này, xác định tập luật và tối ưu điểm số cho tập luật là hai tác vụ được thực hiện riêng rẽ. Vì vậy, bản chất của việc tối ưu tập luật trong trường hợp này là đi tìm tập trọng số tối ưu cho các đặc trưng được lựa chọn ở khâu trước.

SpamAssassin có cơ chế phát hiện thư rác tương tự như mạng nơ-ron một lớp (thể hiện trong công thức 1.2), hay còn gọi là mạng perceptron, với hai điểm khác biệt. Thứ nhất, cơ chế của SpamAssassin không áp dụng hàm kích hoạt đối với đầu ra của lớp mạng. Thứ hai, tập luật của SpamAssassin không có biến số *bias* (độ lệch) như trong mô hình perceptron. Bởi vì sự giống nhau nói trên, việc tối ưu tập luật trong SpamAssassin được thực hiện thông qua việc tối ưu mạng perceptron bằng thuật toán SGD [17], một biến thể của thuật toán tìm kiếm GD. Thuật toán này được sử dụng phổ biến để huấn luyện mạng nơ-ron.

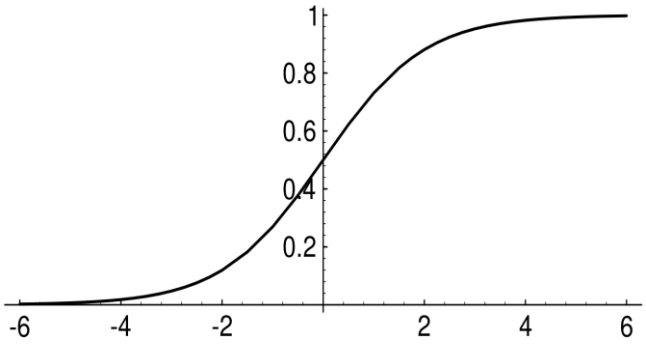
Bản chất của tối ưu mạng perceptron chính là việc tìm kiếm trên không gian $\mathbb{R}^n$ với *n* là số lượng biến số của mạng nơ-ron, hay ở trường hợp này, là số lượng trọng số cần tìm của tập luật. Quá trình tính toán trong mạng perceptron bao gồm một hàm vận chuyển và một hàm kích hoạt. Hàm vận chuyển (1.5) là tổng của tích các trọng số và các phần tử đầu vào tương ứng, cộng thêm độ lệch *bias*.

$$f(x) = \sum_{i=1}^{n} w_i * x_i + bias \tag{1.5}$$

Hàm kích hoạt là một hàm sigmoid (1.6) lấy kết quả của hàm vận chuyển làm đầu vào. Đồ thị của hàm sigmoid được minh họa trong Hình 1.6.

$$y = S(x) = \frac{1}{1 + e^{-x}} \tag{1.6}$$

Với tác dụng của hàm kích hoạt sigmoid, kết quả đầu ra *y* của mạng perceptron được giới hạn trong khoảng (0, 1). Phương pháp sinh tập luật của SpamAssassin quy ước kết quả dự đoán hướng tới giá trị 0 cho những bức thư rác và hướng tới giá trị 1 cho những bức thư hợp lệ. Khi không có đặc trưng nào được tìm thấy trong nội dung thư hoặc tổng trọng số của các đặc trưng được tìm thấy là 0, tức là $f(x) = 0$, thì *y* có giá trị 0.5, giá trị nằm ở vị trí chính giữa của đồ thị hàm sigmoid. Giá trị này được dùng làm ngưỡng để đưa ra kết quả dự đoán một bức thư có phải là thư rác hay không.



Hình 1.6: Đồ thị của hàm kích hoạt sigmoid (1.6) của mạng perceptron [52]

Một tập dữ liệu được dùng để huấn luyện mạng perceptron, trong đó những bức thư rác được gán nhãn $\hat{y} = 0$ và thư hợp lệ được gán nhãn $\hat{y} = 1$. Sự khác biệt giữa kết quả dự đoán *y* của mô hình và kết quả mục tiêu $\hat{y}$ của bức thư được gọi là sai số *E* của mô hình. Ở đây, hàm tổn thất MSE (1.7) được sử dụng để tính sai số *E*.

$$E = MSE = \frac{1}{t}\sum_{i=1}^{t}(\hat{y}_i - y_i)^2 \tag{1.7}$$

Mục tiêu của quá trình huấn luyện là giảm thiểu sai số *E*, từ đó cải thiện hiệu quả dự đoán của mô hình. Đạo hàm riêng (đạo hàm thành phần) của hàm tổn thất đối với từng trọng số được tính bằng công thức (1.8). Khi đã tính được giá trị đạo hàm riêng, các trọng số trong mạng perceptron được cập nhật theo công thức (1.9).

$$\frac{\partial E}{\partial w_i} = -\frac{2}{t}\sum_{j=1}^{t}(\hat{y}_j - y_j) * [y_j * (1 - y_j)] * x_{ji} \quad (1.8)$$

$$w_i = w_i + \frac{\partial E}{\partial w_i} * learning\_rate \quad (1.9)$$

Điểm khác biệt cơ bản giữa thuật toán SGD với thuật toán GD là ở chỗ thuật toán SGD cập nhật các trọng số dựa trên việc tính hàm tổn thất trên một mẫu huấn luyện thay vì trên toàn bộ tập dữ liệu như trong thuật toán GD. Chính vì vậy, với phương pháp SGD thì biến số *t* trong công thức tính hàm tổn thất (1.7) luôn có giá trị là 1 để thể hiện việc sử dụng một mẫu huấn luyện. Nhằm đơn giản hóa việc tính toán đạo hàm, công thức của hàm MSE đã được thay đổi để tính một nửa giá trị của bình phương sai số.

Quá trình huấn luyện mô hình với thuật toán SGD được thực hiện như sau. Với tập dữ liệu $D_{train} = \{(s_1, l_1), (s_2, l_2), \ldots, (s_m, l_m)\}$, trong đó l_i là nhãn của mẫu s_i, mô hình được huấn luyện theo nhiều vòng lặp (iteration). Với thuật toán GD, giá trị hàm tổn thất, ví dụ như hàm tổn thất MSE, sẽ được tính toán trên toàn bộ tập dữ liệu sau mỗi vòng lặp. Dựa trên giá trị tổn thất đó, đạo hàm riêng của tất cả các trọng số mới có thể được tính toán và sau đó các trọng số được cập nhật. Như vậy, tập trọng số của mạng nơ-ron sẽ được cập nhật một lần sau mỗi vòng lặp. Để giảm thiểu khối lượng tính toán và tăng tốc độ huấn luyện, thuật toán SGD lấy ngẫu nhiên một hoặc một số mẫu trong D_{train} để tính giá trị hàm tổn thất cho mỗi lần cập nhật trọng số. Nói một cách khác, thuật toán SGD giảm thiểu số lượng mẫu tham gia vào mỗi vòng lặp để tăng tốc độ huấn luyện. Mỗi vòng lặp có thể chỉ sử dụng một mẫu để tính giá trị hàm tổn thất. Nếu có nhiều hơn một mẫu được lấy ở mỗi vòng lặp, ta gọi đó là việc huấn luyện với *mini-batch*.

Sau khi hoàn thành huấn luyện mạng perceptron với thuật toán SGD, một bước nhỏ cần được thực hiện để có được điểm số cho những luật trong tập luật. Bởi vì sự khác nhau giữa mô hình perceptron và cơ chế phát hiện thư rác của SpamAssassin, trọng số trong mạng perceptron cần được chuyển đổi thành điểm số bằng công thức (1.10) để bù độ lệch của mạng perceptron và để phù hợp với ngưỡng *T* đã được đặt sẵn của SpamAssassin.

$$score(w_i) = \frac{T}{-bias} * w_i \quad (1.10)$$

Việc phát hiện thư rác chỉ có ý nghĩa thực tiễn khi bộ lọc được sử dụng để dự đoán những bức thư mới, chưa từng xuất hiện trong dữ liệu huấn luyện. Trong khi đó, dữ liệu có sẵn để huấn luyện bộ lọc chỉ là một tập nhỏ trong toàn bộ dữ liệu thư điện tử thực tế. Do đó, mô hình thu được sau khi huấn luyện với tập dữ liệu D_{train} là một sự ước lượng so với mô hình lọc thư rác lý tưởng. Giống như phương pháp lấy mẫu để ước lượng kết quả khái quát trong khoa học thống kê, bộ lọc thư rác cũng được huấn luyện với một số lượng mẫu giới hạn để khái quát hóa cho bài toán thực tế. Giả thiết đặt ra khi thực hiện huấn luyện mô hình là tập dữ liệu D_{train} có khả năng đại diện cho dữ liệu thực tế. Một trong những mục tiêu khi thực hiện các nghiên cứu với phương pháp học máy thống kê đó là cải thiện tính đại diện của tập dữ liệu.

Sau khi đã huấn luyện xong, ta cần thử nghiệm để đánh giá chất lượng của tập luật trước khi đưa vào áp dụng thực tế. Ở bước này, tập luật được sử dụng để phát hiện thư rác trên tập dữ liệu $D_{test} = \{(s_1, l_1), (s_2, l_2), \ldots, (s_k, l_k)\}$ với điều kiện $D_{test} \cap D_{train} = \emptyset$, mục đích là để đo lường tính khái quát của tập luật. Nói theo cách khác, nếu hiệu năng trên tập D_{train} và D_{test} tương tự nhau thì mức độ tự tin về tính khái quát của tập luật được tăng thêm, tập luật có nhiều khả năng sẽ phát hiện thư rác tốt trên dữ liệu thực tế. Ngược lại, sự khác biệt đáng kể giữa hiệu năng trên hai tập dữ liệu có thể là biểu hiện của tình trạng overfitting, tập luật bị mất tính khái quát. Quá trình huấn luyện tập luật có chứa các yếu tố ngẫu nhiên bao gồm trọng số ban đầu của luật và thứ tự mẫu tham gia vào quá trình huấn luyện đối với thuật toán tối ưu lựa chọn mẫu một cách ngẫu nhiên như SGD. Vì vậy, tập trọng số được sinh ra sau mỗi lần huấn luyện không giống nhau. Tập trọng số này có thể đại diện cho một điểm tối ưu cục bộ nào đó trong không gian tìm kiếm. Dựa vào kết quả đánh giá, người thực hiện sinh tập luật có thể xem xét việc huấn luyện lại để sinh ra tập trọng số khác phù hợp hơn hoặc xây dựng tập luật mới theo phương án khác.

Việc đánh giá tập luật có nội dung cơ bản là thử nghiệm tập luật với những bức thư chưa từng có mặt trong dữ liệu huấn luyện. Các đánh giá mô hình học máy đơn giản nhất là chia tập dữ liệu thành hai phần D_{train} và D_{test} như đã đề cập ở trên. Điểm hạn chế

của cách làm này là mô hình chỉ được đánh giá trên một phần của tập dữ liệu. Phương pháp kiểm chứng chéo *k* lần (*k*-fold cross-validation) có mục tiêu khắc phục hạn chế này, nó cho phép toàn bộ tập dữ liệu được tham gia thử nghiệm mô hình. Dữ liệu được chia đều thành *k* phần và thử nghiệm được thực hiện *k* lần, các mỗi phần dữ liệu sẽ lần lượt được chọn làm tập D_{test} trong khi các phần còn lại được gộp vào làm tập D_{train}. Giá trị *k* thường dùng là 3, 5 hoặc 10, trong một số trường hợp *k* có thể được tăng cao hơn và điều đó sẽ dẫn đến thời gian huấn luyện tăng tỷ lệ thuận với độ lớn của *k*.

1.3.1.2. Lọc thư rác bằng phương pháp học máy thống kê

Học máy thống kê luôn được coi là giải pháp chung cho các bài toán dự đoán trong khoa học máy tính, trong đó bài toán lọc thư rác là một ví dụ điển hình.

Nghiên cứu của Sahami [5] là một trong những nghiên cứu đầu tiên áp dụng máy phân loại Bayes để lọc thư rác tiếng Anh. Phương pháp của Sahami đạt độ chính xác 92% đo bằng chỉ số accuracy với tỷ lệ cảnh báo nhầm 1.16%. Nghiên cứu của Graham [14] cũng đề xuất máy phân loại Bayes để lọc thư rác nhưng với một số cải tiến trong phương pháp trích chọn đặc trưng và tập dữ liệu huấn luyện lớn hơn. Nghiên cứu này đạt kết quả tốt hơn đáng kể so với phương pháp trước đó của Sahami, với độ chính xác 99.5% và tỷ lệ cảnh báo nhầm 0.03%.

Một giả thiết quan trọng mà các bộ lọc Bayes dựa vào đó là các đặc trưng trong vector đầu vào phải là các biến cố độc lập về mặt thống kê, điều thường không xảy ra trong các bài toán về xử lý văn bản. Ngoài ra, những kẻ phát tán có thể chèn những nội dung hợp lệ vào thư rác để qua mặt các bộ lọc Bayes. Vì những lý do trên, Yerazunis đã so sánh bộ lọc Bayes với bộ lọc Markov [18] và đã kết luận rằng bộ lọc Markov có độ chính xác cao hơn. Để giải thích cho kết quả, Yerazunis cho rằng bộ lọc Bayes chỉ là một mô hình tuyến tính, trong khi mô hình Markov có yếu tố phi tuyến tính.

Các kỹ thuật học máy thường đạt hiệu quả cao với sự can thiệp tối thiểu của con người và có khả năng cập nhật nhanh chóng khi có dữ liệu huấn luyện mới. Công việc quan trọng nhất khi áp dụng học máy để lọc thư rác là lựa chọn được bộ thuộc tính có chất lượng cao. Ngoài ra, các bước tiền xử lý văn bản như phân tích cú pháp, đưa từ ngữ về dạng nguyên thể (stemming), dọn dẹp văn bản (cleaning), tính toán trọng số cho thuộc tính (term-weighting) và tiêu chuẩn hóa vector đầu vào (normalization)… cũng

rất quan trọng và góp phần tăng hiệu quả cho bộ lọc. Các phương pháp tìm trọng số cho thuộc tính nổi tiếng là TF, TF-IDF và biểu diễn nhị phân (còn gọi là one-hot encoding). Nhìn chung, giảm số lượng đặc trưng làm tăng tính tổng quát và đồng thời làm giảm khả năng tách biệt giữa các lớp phân loại của mô hình [35]. Ngoài ra, một điểm đáng chú ý là cả hai phương pháp biểu diễn thuộc tính truyền thống vector nhị phân và vector trọng số, ví dụ như TF-IDF, đều không giữ được thứ tự xuất hiện của từ trong văn bản.

Các kỹ thuật phân loại Bayes và Markov tỏ ra rất hiệu quả với dữ liệu thư rác tiếng Anh. Bộ lọc Bayes [5, 14] cho độ chính xác lên tới 99.9% khi đo bằng chỉ số accuracy. Trong khi đó, Yerazunis [18] áp dụng thuật toán Markov để lọc thư rác báo cáo tỷ lệ lỗi (*fp* + *fn*) thấp hơn phương pháp Bayes tới 40%. Bộ lọc Bayes thường được sử dụng làm cơ sở để so sánh với các phương pháp khác. Các kỹ thuật học máy như SVM, mạng nơ-ron, logistic regression, kNN, hệ miễn dịch nhân tạo (artificial immune systems), boosting và nhiều phương pháp khác đều được áp dụng để lọc thư rác và cho kết quả ngang bằng và tốt hơn bộ lọc Bayes [43]. Phần lớn các phương pháp được nhắc đến trong [18] và [43] đều sử dụng thuộc tính nội dung của thư điện tử.

Bộ lọc Bayes hoạt động dựa trên lý thuyết xác suất của Bayes với giả thiết đơn giản rằng các thuộc tính là các biến độc lập với nhau. Nó phân loại một bức thư bằng cách tính xác suất bức thư đó là thư rác và thư hợp lệ rồi so sánh xem xác suất nào cao hơn. Với giả thiết về sự độc lập của các biến, xác suất để một bức thư là thư rác là tích xác suất bức thư là thư rác khi xuất hiện từng từ trong bức thư. Công thức (1.11) thể hiện cách tính xác suất đó.

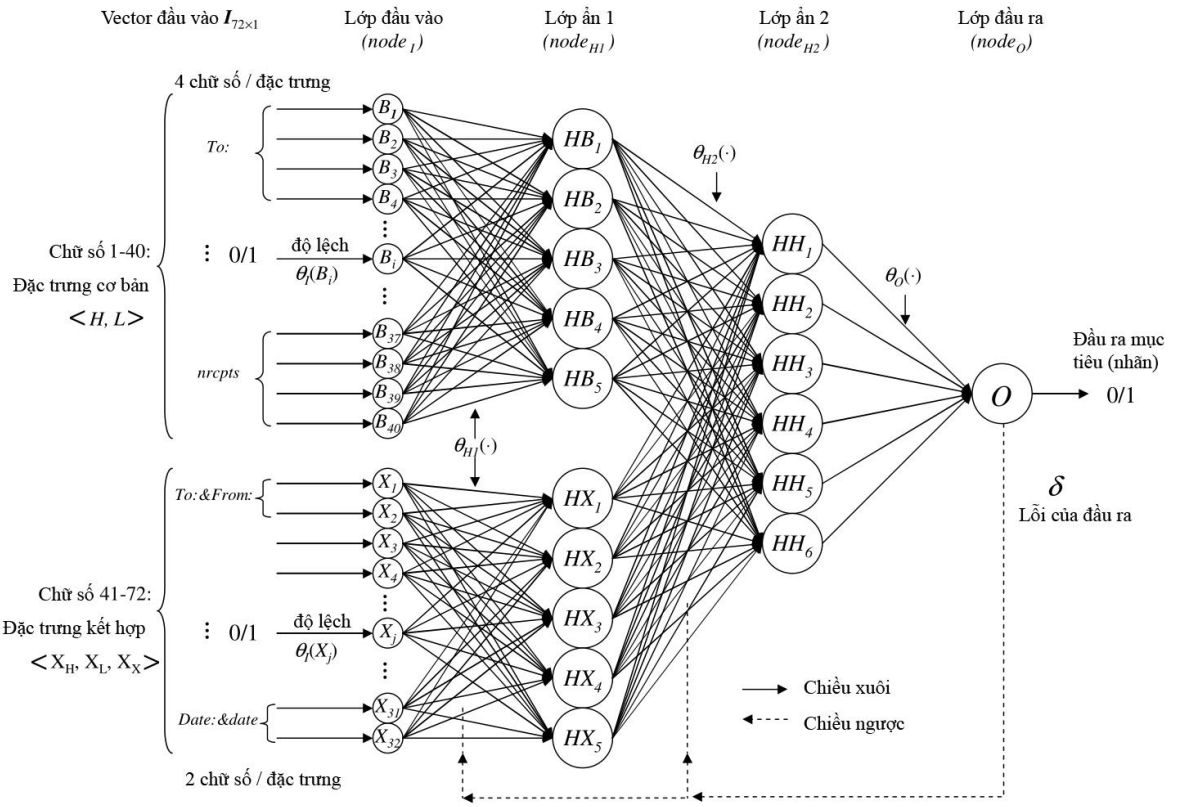
$$\mathrm{P}(\mathrm{SPAM} \mid m) = \prod_{w \in m} \mathrm{P}(w \mid \mathrm{SPAM}) \tag{1.11}$$

Hiệu quả của bộ lọc Bayes rất cao, nhưng nó cũng có những điểm yếu đã bị phát hiện. Một yếu điểm lớn của bộ lọc Bayes và cũng là của những bộ lọc dựa trên thống kê khác đó là nó dễ bị đánh bại bởi kiểu tấn công statistical poisoning. Spammer tìm cách chèn nội dung thường thấy trong thư hợp lệ vào thư rác, khiến cho "phần hợp lệ" trong những bức thư đó trở thành chủ yếu và thư rác lọt qua được bộ lọc, trở thành *False Negative*. Theo Graham-Cumming [21], ta có thể dùng một bộ lọc Bayes để đánh bại chính bộ lọc Bayes đó, cụ thể là bộ lọc Bayes có thể được dùng để tìm ra những từ hoặc

cụm từ làm xác suất một bức thư là thư hợp lệ tăng lên. Một kỹ thuật phổ biến của những kẻ phát tán thư rác là trộn những từ ngữ ngẫu nhiên vào thư rác để tỷ lệ những từ khóa có trong thư rác giảm đi, khiến cho những bộ lọc dựa trên xác suất từ khóa (ví dụ như bộ lọc Naïve Bayes) không còn hiệu quả [21].

Một lợi thế của SVM so với Bayes đó là SVM vẫn cho hiệu quả cao với lượng thuộc tính lớn [43] nên SVM không đòi hỏi quá trình lựa chọn thuộc tính. Ngoài ra, máy phân loại SVM có khả năng phân loại phi tuyến tính trong khi Bayes chỉ là một bộ lọc tuyến tính [18]. Vấn đề quan trọng nhất khi sử dụng SVM làm bộ lọc thư rác đó là lựa chọn dữ liệu huấn luyện có tính đại diện cao. Khi sử dụng một tập dữ liệu lớn nhưng không đại diện cho thư điện tử của một lượng lớn người sử dụng, ví dụ như tập Enron, để huấn luyện SVM, hiệu quả đạt được bị hạn chế [43].

Mạng nơ-ron có kết quả tốt hơn bộ lọc Bayes ở một số thí nghiệm được công bố [43] nhưng hiệu quả đó chưa được chứng minh là ổn định. Ngoài thuộc tính nội dung thì trong [39], tác giả sử dụng các thuộc tính về hành vi của người gửi thư, dùng mô hình mạng nơ-ron nhiều lớp (MLP) và công bố kết quả tốt hơn đáng kể so với việc sử dụng thuộc tính nội dung. Hình 1.7 minh họa cấu trúc mạng nơ-ron nhiều lớp truyền thẳng được sử dụng trong [39] dùng để phát hiện thư rác. Trong mô hình này, tác giả đã kết hợp nhiều loại đặc trưng của thư điện tử, trong đó không có đặc trưng nội dung. Các đặc trưng được trích xuất từ các trường header phổ biến nhất của bức thư và các trường phổ biến nhất trong ghi chép hệ thống (syslog) của MTA. Các đặc trưng được chia thành hai loại: đặc trưng cơ bản (từ B1 đến B40) và đặc trưng kết hợp (từ X1 đến X32). Các đặc trưng này được trích xuất bằng nhiều quy tắc heuristic. Mô hình MLP trong [39] có sự hợp thành của hai đầu vào độc lập. Hai vector đầu vào này được biến đổi thành hai nhóm đặc trưng HB và HX. Sau đó HB và HX được hợp nhất với nhau và tiếp tục trở thành đầu vào của lớp mạng phía sau.



Hình 1.7: Lọc thư rác bằng mạng nơ-ron 2 lớp ẩn dựa trên hành vi người gửi [39]

Phương pháp tự động sinh tập luật SpamAssassin được mô tả trong [28] cũng được phân loại và nhóm các phương pháp lọc thư rác dựa trên học máy thống kê. Đó là bởi vì một tập luật SpamAssassin có cách hoạt động giống với một mô hình mạng nơ-ron một lớp (perceptron). Việc lựa chọn từ khóa dùng để sinh tập luật mà các tác giả đã trình bày chính là bước lựa chọn đặc trưng trong học máy thống kê. Vấn đề chọn điểm số cho các luật SpamAssassin có thể được giải quyết bằng cách huấn luyện mạng perceptron nói trên, cụ thể là bằng thuật toán Stochastic Gradient Descent. Phương pháp [62] có cách tiếp cận tương tự nhưng đối tượng áp dụng là thư điện tử tiếng Việt và đề xuất sử dụng thêm một tập đặc trưng được gọi là các *luật ham*. Trong [62], một giải thuật tiến hóa mang tên HPSOWM [37] đã được sử dụng và đạt hiệu quả tốt hơn so với thuật toán SGD được sử dụng trước đó.

Các phương pháp sinh tập luật SpamAssassin nói trên đều gặp phải vấn đề khi lựa chọn giá trị của ngưỡng *T* cho tập luật. Nguyên nhân của tình trạng này nằm ở chỗ các thuật toán huấn luyện điểm số cho tập luật đều có cơ chế tối ưu hóa đơn mục tiêu, trong khi đó lại có nhiều tiêu chí để đánh giá một bộ lọc thư rác. Điều đó dẫn đến hai tiêu chí *recall* (1.14) và *precision* (1.15) không thể được tối ưu đồng thời. Để khắc phục hạn

chế này, giải thuật tiến hóa đa mục tiêu NSGA-II và DMEA-II đã được áp dụng để sinh tập luật cho SpamAssassin trong [76]. Dựa trên kết quả thí nghiệm, thuật toán DMEA-II cho thấy khả năng tìm được những tập tham số tốt nhất và vượt trội so với phương pháp tối ưu đơn mục tiêu truyền thống và NSGA-II. Một hạn chế mà tác giả của [76] đã chỉ ra trong bài báo là kích thước của tập dữ liệu thử nghiệm chưa đủ lớn nên thời gian huấn luyện của phương pháp chưa được đánh giá một cách chính xác.

Độ khó của bài toán lọc thư rác đến từ thực tế rằng việc lọc nhầm thư hợp lệ không được chấp nhận bởi người dùng. Vì vậy, cần phải kết hợp nhiều đặc điểm của thư điện tử để lọc thư rác một cách chính xác nhất. Các phương pháp lọc thư rác đã khai thác các thuộc tính về nội dung [36, 43, 62, 65, 75], về hành vi người gửi [39], chuỗi lệnh SMTP [42]... Ngoài những đặc trưng nói trên, các thuộc tính về người gửi thư là những đặc trưng có khả năng ảnh hưởng đáng kể đến việc bức thư có là thư rác hay không, cũng như tới tầm quan trọng của bức thư. Hầu hết các phương pháp lọc thư rác dựa trên đặc trưng xã hội của người gửi thư đều sử dụng mạng xã hội thư điện tử để trích xuất đặc trưng. Lý thuyết đồ thị được áp dụng, trong đó các địa chỉ thư điện tử là các đỉnh đồ thị và hành động gửi thư giữa những người dùng là các cạnh có hướng. Các phương pháp trong nhóm này tính toán các hệ số để đại diện cho người dùng thư điện tử với nhiều công thức khác nhau [26, 29, 54]. Nghiên cứu [54] đã đưa ra kết quả so sánh giữa các phương pháp mạng thư điện tử khác nhau. Đây là một cách tiếp cận có nhiều tiềm năng giúp cải thiện các bộ lọc thư rác.

Một trong những kỹ thuật qua mặt bộ lọc thư rác của những kẻ phát tán đó là nhúng nội dung cần truyền tải vào một bức ảnh và gửi kèm với bức thư. Kỹ thuật này có thể khiến các bức thư rác vượt qua được các bộ lọc dựa hoàn toàn trên nội dung ở dạng văn bản. Tuy nhiên, hạn chế của những bộ lọc dựa trên văn bản nói trên có thể được giải quyết bằng cách nhận dạng chữ viết trong hình ảnh. Tính năng nhận dạng chữ viết là một sự bổ sung [66, 75] giúp tăng hiệu quả cho bộ lọc thư rác.

1.3.1.3. Lọc thư rác bằng phương pháp mạng thư điện tử

Các nghiên cứu gần đây đã bắt đầu khai thác thông tin từ mạng xã hội cho việc xác định thư rác bằng cách xây dựng một đồ thị (các đỉnh là địa chỉ thư điện tử, cung được thêm vào giữa 2 đỉnh A và B nếu giữa A và B có sự trao đổi thư qua lại). Một công cụ

lọc thư rác có thể được xây dựng dựa trên các đặc trưng mạng xã hội từ tập dữ liệu thư điện tử. Các đặc trưng mạng xã hội được áp dụng để phát hiện thư rác lần đầu tiên bởi Boykin và Roychowdhury [26] khi họ đề xuất phương pháp phát hiện thư rác dựa trên hệ số phân cụm. Nói một cách khác, bài báo [26] đã đặt giả thiết rằng tầm quan trọng của một bức thư phụ thuộc vào người gửi bức thư đó. Theo đó, hai tác giả đã thu thập thư điện tử trong hộp thư cá nhân của mình để xây dựng nên mạng thư điện tử, trong đó các nút mạng là các địa chỉ thư điện tử và các cạnh là sự trao đổi thư điện tử giữa các nút mạng đó. Các tác giả mô hình hóa sự trao đổi thư điện tử của tập người dùng, một mạng thư điện tử, như một mạng xã hội. Dựa theo ý nghĩa của hai độ đo đặc trưng trên đây, công thức (1.12) được dùng để tính độ phân cụm của đỉnh thứ *i* trong mạng thư điện tử được đề xuất.

$$C_i = \frac{2 * E_i}{k_i(k_i - 1)} \tag{1.12}$$

Trong đó, C_i là độ phân cụm của đỉnh *i*, k_i là số đỉnh kết nối với đỉnh *i*, E_i là số lượng cung nối giữa các đỉnh láng giềng của *i*. Tác giả đã nhận định rằng đỉnh có độ phân cụm càng cao thì khả năng địa chỉ thư điện tử tương ứng với đỉnh đó gửi thư rác càng thấp, hay nói cách khác, đó là người dùng bình thường.

Nghiên cứu này có một số hạn chế. Thứ nhất, nó đã bỏ qua tất cả các đỉnh có $k = 1$. Thứ hai và là quan trọng hơn, kết quả tính toán không cho phép phân biệt được các đỉnh tuy có cùng giá trị $E = 0$ nhưng có giá trị *k* khác nhau ($C = 0$ khi $E = 0$). Ngoài ra quá trình thực nghiệm của phương pháp này sử dụng hòm thư cá nhân, như vậy các thông tin mạng thư điện tử sẽ không toàn diện.

Để khắc phục những nhược điểm của phương pháp trên trên, nghiên cứu [29] đã thay đổi cách tính độ phân cụm *C* trong công thức (1.13).

$$C_i = \frac{2 * (E_i + 1)}{k_i(k_i - 1) + 1} + 0.2 * R_i \tag{1.13}$$

Trong đó, E_i là số cung nối giữa các node xung quanh node *i*, S_i là số node có thư gửi từ node *i*, R_i là số node gửi thư tới node *i*. Công thức đảm bảo nếu một người gửi thư cho nhiều node lân cận mà những node lân cận này có mối quan hệ với nhau thì độ

phân cụm cao và nếu người này được nhận thư từ các node lân cận khác nữa thì độ phân cụm của người đó càng lớn. Những địa chỉ thư điện tử dùng để gửi thư rác thường không nhận thư nên R_i bằng 0.

Vào năm 2005, một phương pháp đánh giá tầm quan trọng của người gửi thư gần giống với phương pháp xếp hạng PageRank [10] có tên MailRank [27] đã được công bố. Từ những thông số về số lượng thư rác và thư hợp lệ được nhận và gửi bởi mỗi người dùng, phương pháp MailRank tính toán điểm số cho mỗi người dùng và kết luận một bức thư gửi bởi người A tới người B có phải là thư rác hay không.

Nghiên cứu [54] cũng có mục tiêu là giải quyết vấn đề xếp hạng người gửi dựa vào mạng xã hội thư điện tử. Bốn phương pháp phân tích mạng thư điện tử là clustering coefficient, extended clustering coefficient, PageRank và Weighted PageRank đã được thử nghiệm trên 03 tập dữ liệu. Tập dữ liệu thứ nhất là tập dữ liệu thực tế gồm 14,320 bức thư, 101 người dùng, trong đó có 31 người dùng quan trọng. Bộ dữ liệu thứ hai bao gồm bộ dữ liệu thứ nhất cộng thêm một số bức thư được thu thập từ một nguồn khác (thêm 7,634 thư quan trọng và 32,769 thư rác). Bộ dữ liệu thứ ba là bộ dữ liệu chuẩn TREC07p từ hội thảo TREC, gồm 24,984 thư quan trọng và 49,240 thư rác. Nghiên cứu đưa ra kết luận rằng phương pháp xếp hạng dựa vào mạng thư điện tử phát huy được hiệu quả tốt khi xây dựng được mạng thư điện tử lớn, hoàn thiện. Với tập dữ liệu mà trong đó thư điện tử chỉ được gửi từ mạng nội bộ ra bên ngoài, thiếu đi tương tác trong mạng nội bộ và từ bên ngoài vào, phương pháp xếp hạng người dùng dựa trên mạng thư điện tử tỏ ra không hiệu quả.

1.3.1.4. Lọc thư rác dựa trên dấu hiệu

Phương pháp lọc thư rác theo từ khóa là phương pháp truyền thống trong việc lọc thư rác. Người ta dựa vào những từ hay cụm từ có trong tiêu đề và nội dung của thư để lọc. Khi một thư mới được gửi tới hòm thư, người dùng phải tạo một bộ lọc mới đơn giản bằng cách chọn một số từ hoặc cụm từ trong nội dung thư. Các từ hay cụm từ này sẽ xác định đó là thư rác hay không. Khi spam mới xuất hiện, mục đích của tất cả spam cơ bản là giống nhau (bán hoặc quảng cáo một sản phẩm hay một dịch vụ) và nội dung của hầu hết spam đều mang các đặc điểm chung với những từ, cụm từ đặc trưng. Khi đó, những bộ lọc thủ công vẫn còn tỏ ra hữu dụng. Sau đó, những kẻ gửi thư rác đã bắt

đầu nhận ra rằng thư rác của chúng đã bị chặn bởi bộ lọc theo từ khóa này. Do vậy, cách viết nội dung của thư rác cũng được thay đổi nhằm làm cho thư rác có thể lọt qua các bộ lọc đơn giản. Những bức thư rác với những từ được viết cách điệu như "`Vi@gra`", "`Mort.gage`", "`L|0|a|n|$`" hay những hình ảnh được nhúng vào trong thư xuất hiện ngày càng nhiều. Nội dung văn bản được hiển thị dưới dạng một bức ảnh và được nhúng vào bức thư. Đây là cách mà kẻ phát tán dùng để qua mặt các hệ thống lọc thư rác. Khi đó, những hệ thống lọc dựa vào nội dung thư dạng văn bản thuần túy sẽ bị vượt qua một cách dễ dàng [66]. Hiệu quả của bộ lọc theo từ khóa cần được duy trì bằng cách điều chỉnh danh sách từ khóa một cách thủ công nên nó không theo kịp được tốc độ phát triển của thư rác.

1.3.1.5. Lọc thư rác dựa trên cơ sở hạ tầng gửi thư

Giao thức SMTP là nền tảng cơ sở hạ tầng phục vụ cho sự hoạt động của thư điện tử. Tuy SMTP là cốt lõi của hệ thống thư điện tử nhưng thiết kế của nó chứa đựng nhiều điểm yếu mà kẻ phát tán thư rác có thể lợi dụng. Nhóm phương pháp lọc thư rác dựa trên giao thức SMTP nhắm tới việc khắc phục những đặc điểm cố hữu của giao thức SMTP, từ sự đơn giản, thuận tiện và nguyên tắc giữ gìn thông tin của nó [35].

Đầu tiên, một nghiên cứu đề xuất thay đổi từ giao thức SMTP với nguyên lý người gửi đẩy thư tới người nhận thành DMTP với nguyên lý người nhận chủ động tải thư về từ máy chủ của người gửi [33]. Như vậy, máy chủ của kẻ phát tán thư rác sẽ là vị trí dễ bị quá tải nhất, khiến cho việc gửi thư hàng loạt gặp khó khăn. Để thực hiện theo hướng giải quyết này, người ta phải thực hiện những thay đổi cốt lõi trên giao thức SMTP. Điều đó sẽ gây ra nhiều vấn đề với hàng loạt những dịch vụ có liên quan, những xung đột phần mềm hoặc những hậu quả từ việc không tương thích giữa các hệ thống. Tuy DMTP có một số tiềm năng nhưng chi phí để thay đổi toàn bộ cơ sở hạ tầng phần mềm của các hệ thống thư điện tử đang sử dụng SMTP sang DMTP là quá lớn.

Một nghiên cứu khác [42] phát hiện thư rác dựa vào các chuỗi lệnh SMTP bất thường trong dữ liệu mạng và báo cáo kết quả rằng lượng thư rác có thể được giảm thiểu tới 11%. SMTP có một tập các lệnh cơ bản: HELO, MAIL FROM, RCPT TO, DATA, . , QUIT, và RSET. Các tác giả của phương pháp này cho rằng những chuỗi lệnh SMTP bất thường (lấy dữ liệu của người dùng thông thường làm mẫu để đối chiếu) chính là

các chuỗi lệnh dùng để gửi thư rác. Phương pháp này có thể áp dụng dưới hình thức một lớp bảo vệ ngoài cùng của hệ thống nhận thư. Tuy nhiên, điểm yếu của nó là kẻ phát tán thư rác có thể theo dõi những lần gửi thư thất bại và thay đổi các chuỗi lệnh gửi thư để qua mặt bộ lọc.

Khi nhắc đến những kỹ thuật lọc thư rác dựa trên SMTP, không thể không kể tới phương pháp lọc dựa trên danh sách, cụ thể là dựa trên địa chỉ IP của người gửi. Ta có ba hướng đó là danh sách đen (*blacklist*), danh sách trắng (*whitelist*) và danh sách xám (*greylisting*). Phương pháp danh sách đen duy trì một danh sách các địa chỉ IP bị chặn. Người ta lập ra một danh sách các địa chỉ gửi thư rác. Các nhà cung cấp dịch vụ thư điện tử (ESP) sẽ dựa trên danh sách này để loại bỏ những thư nằm trong danh sách này. Danh sách này thường xuyên được cập nhật và được chia sẻ giữa các nhà cung cấp dịch vụ. Có một loại danh sách đen có tên "danh sách đen thời gian thực" (RBL) là danh sách đen được cập nhật liên tục và hoàn toàn tự động. Một số danh sách đen phổ biến là SpamCop Blocking List và Composite Block List. Ưu điểm của phương pháp này là các ISP sẽ ngăn chặn được khá nhiều địa chỉ gửi thư rác. Mặc dù danh sách đen này luôn được cập nhật nhưng với sự thay đổi liên tục địa chỉ, sự giả mạo địa chỉ hoặc lợi dụng một mail server hợp pháp để gửi thư rác, số lượng thư rác gửi đi vẫn ngày càng tăng cao. Do đó phương pháp này chỉ ngăn chặn được một nửa số thư rác gửi đi và sẽ mất rất nhiều thư hợp pháp nếu ngăn chặn nhầm.

Danh sách trắng còn được gọi là danh sách các địa chỉ IP tin cậy. Danh sách này có thể do một nhà cung cấp dịch vụ nào đó cung cấp. Những địa chỉ thuộc danh sách sẽ được cho qua bộ lọc. Người dùng phải đăng ký với nhà cung cấp danh sách để được nằm trong danh sách. Ưu điểm của phương pháp này là số lượng địa chỉ trong danh sách trắng sẽ ít hơn trong danh sách đen vì thế sẽ dễ cập nhật hơn danh sách đen và giải quyết được tình trạng chặn nhầm thư. Tuy nhiên, cả hai phương pháp trên đều có nhược điểm là khó cập nhật, nhất là khi ai đó thay đổi địa chỉ IP. Ngoài ra người gửi cũng có thể lợi dụng server mail có trong danh sách trắng để gửi thư rác, khi đó rất khó kiểm soát.

Phương pháp *danh sách xám* hoạt động như sau: mỗi khi có thư từ một địa chỉ mới, máy chủ trả lại người gửi thư một thông điệp yêu cầu người gửi trả lời lại nó để xác thực người gửi và đưa người đó vào danh sách trắng. Từ lần sau trở đi, máy chủ sẽ tự

động chấp nhận thư đến từ người gửi đó. Phương pháp danh sách xám dựa trên ý tưởng rằng kẻ phát tán thư rác thường không gửi lại sau khi gửi đi thất bại. Phương pháp đơn giản này tỏ ra khá hiệu quả trên thực tế [35] nhưng nếu kẻ phát tán thư rác nắm được nó, họ dễ dàng sửa đổi công cụ gửi thư rác để qua mặt bộ lọc.

Phương pháp lọc thư rác sử dụng chuỗi hỏi đáp (Challenge/Response) có cơ chế tự động gửi thư hồi đáp cho người gửi để yêu cầu một số hành động kiểm tra chắc chắn về việc gửi thư của họ. Chương trình yêu cầu người gửi thư phải trả lời một số câu hỏi đơn giản (thử thách) để xác minh về bức thư mà người này đã gửi.Việc này chỉ được yêu cầu trong lần gửi thư đầu tiên. Việc đáp ứng thử thách này thường không có gì khó khăn đối với người gửi thư khi họ muốn gửi thư cho một người khác nhưng nó không mấy dễ dàng cho những kẻ gửi thư rác muốn phát tán một lượng lớn thư rác đi. Tuy nhiên, cách làm này vẫn gây ra một chút phiền phức cho người gửi thư khi họ thường xuyên phải trả lời các câu hỏi khi gửi thư đến các địa chỉ thư điện tử mới.

Giả mạo địa chỉ thư điện tử cũng là một kỹ thuật gửi thư của những kẻ phát tán thư rác, bởi vì giao thức SMTP có một hạn chế là không có cơ chế xác thực địa chỉ người gửi. Thật khó để biết chắc một bức thư có bị giả mạo địa chỉ người gửi hay không. Do đó việc xác nhận danh tính của người gửi là rất cần thiết. Một phương pháp để xác nhận danh tính của người gửi là DKIM [48] hay còn gọi là DomainKeys. Phương pháp DomainKeys có thể giúp phân định thư rác và thư thường bằng cách thêm khóa công khai vào bản ghi DNS của tên miền gửi thư. Đồng thời, mỗi bức thư gửi đi được đính kèm một chữ ký điện tử được mã hóa bằng khóa bí mật tương ứng của khóa công khai nói trên. Kẻ phát tán thư rác không thể biết được khóa bí mật của máy chủ gửi thư, từ đó không thể tạo ra chữ ký điện tử hợp lệ cho mỗi bức thư. Cách xác thực người gửi này tương tự với cơ chế của phương pháp chữ ký điện tử thông dụng. SPF [60] là một phương pháp khác để xác thực người gửi. Với SPF, chỉ những địa chỉ IP được cho phép trong bản ghi DNS của một tên miền mới được phép gửi thư từ tên miền đó. Ví dụ, khi một bức thư có địa chỉ người gửi là *user@example.com* được gửi từ máy chủ có địa chỉ IP là 1.2.3.4, nếu địa chỉ 1.2.3.4 không tồn tại trong bản ghi SPF của tên miền *example.com* thì bức thư sẽ bị chặn. SPF cho phép xác định máy chủ nào có thể gửi thư từ một tên miền nhất định. Kỹ thuật này cho phép loại bỏ thư rác trước khi chúng được

gửi đi, nếu địa chỉ IP dùng để gửi thư không được cho phép trên bản ghi DNS của tên miền. Bởi vì tính đơn giản và hiệu quả của nó, ngày nay, SPF vẫn đang được sử dụng rộng rãi. Tuy nhiên, SPF không thể ngăn cản được những bức thư rác được gửi từ địa chỉ IP được tên miền cho phép, điều mà những kẻ phát tán thư rác hoàn toàn có thể thực hiện được với tên miền riêng của họ. DMARC [68] là sự kết hợp của hai phương pháp SPF và DomainKeys, nhằm hướng tới cơ chế xác thực người gửi chặt chẽ hơn so với việc sử dụng riêng lẻ SPF hoặc DomainKeys.

Trên thực tế, thư rác có thể được gửi thông qua các máy chủ gửi thư lớn như Gmail, Outlook, SendGrid, MailGun, AmazonSES, Elastic Email… Có thể nói hầu hết các nhà cung cấp dịch vụ thư điện tử đều cho phép các nhà phát triển gửi thư thông qua API mà họ cung cấp với mức chi phí thấp. Đây là một cách phổ biến mà kẻ phát tán thư rác sử dụng ngày nay. Thư được gửi từ các nhà cung cấp nói trên bao gồm cả thư hợp lệ và thư rác nên không thể bị chặn hoàn toàn. Vì vậy, các phương pháp như DomainKeys, SPF, DMARC không có hiệu quả trong tình huống này.

1.3.2. Nghiên cứu về dự đoán hành động người dùng

Bài toán dự đoán hành động người dùng là một dạng của bài toán xác định thứ tự ưu tiên của thư điện tử, có mục tiêu giải quyết vấn đề quá tải thư điện tử. Một trong những nghiên cứu đầu tiên về dự đoán hành động người dùng thư điện tử [25] tuy không xây dựng một mô hình dự đoán tự động nhưng đã đưa ra được những yếu tố trong nội dung các bức thư có ảnh hưởng đến hành động của người dùng đối với chúng. Phương pháp sử dụng trong bài báo này là phương pháp khảo sát và phân tích. Nghiên cứu này chỉ ra rằng tầm quan trọng của bức thư bị ảnh hưởng lớn nhất từ những đặc điểm sau: người nhận có quan hệ công việc với người gửi (tăng 23% độ quan trọng), trong nội dung có chứa lịch làm việc (tăng 24%) hoặc bức thư có nội dung xã giao (giảm 32%). Trong số các thuộc tính được khảo sát, xác xuất trả lời thư tăng nhiều nhất khi bức thư có chứa nội dung xã giao (tăng 23%) hoặc nếu đó là thư để hỏi thông tin (tăng 22%). Kết quả của nghiên cứu này có ý nghĩa to lớn hỗ trợ cho những phương pháp xếp hạng thư điện tử trong tương lai.

Phương pháp phân loại Bayes đã được áp dụng bởi Minh và cộng sự để dự đoán hành động người dùng, với chỉ số accuracy được báo cáo là 83.3% trên tập dữ liệu tự thu thập

[51]. Phương pháp này chọn các hành động phổ biến nhất là: *trả lời, đọc* và *xóa* để gợi ý cho người dùng. Thuộc tính được lựa chọn bằng kỹ thuật ước lượng xác suất (m-estimate of probability). Chỉ có những từ có xác suất đối với một lớp (hành động) cao hơn hẳn với hai lớp kia mới được chọn làm đặc trưng. Tác giả của [51] đưa ra nhận xét đối với kết quả thí nghiệm rằng máy phân loại Naïve Bayes gặp khó khăn trong việc phân biệt giữa hành động *đọc* và hành động *trả lời* bởi vì những bức thư của hai hành động này có nội dung khá giống nhau. Một hạn chế trong thí nghiệm của bài báo [51] là việc thu thập dữ liệu từ một số danh sách gửi thư chứ không phải dữ liệu thư điện tử thực tế của người dùng.

Một nghiên cứu khác của Ayodele và Zhou [44] vào năm 2009 đề xuất phương pháp dự đoán thư cần trả lời, một cách hiệu quả để giảm thiểu thời gian xử lý thư. Nội dung chính của phương pháp là một thuật toán dự đoán trong đó kết hợp một số luật dựa trên kinh nghiệm (heuristics) với các thuộc tính có được từ quá trình học máy không giám sát trên dữ liệu thu được trong quá trình sử dụng của người dùng. Các thuộc tính thống kê được từ phương pháp học máy không giám sát gồm: tên miền của địa chỉ người gửi, những cuộc hội thoại trước đó với người gửi, các cụm từ nghi vấn hoặc yêu cầu hành động trong tiêu đề thư, tệp đính kèm và các từ ngữ yêu thích của người dùng.

Di Castro và cộng sự đã nghiên cứu về 04 hành động của người dùng đối với thư điện tử là đọc, trả lời, xóa sau khi đã đọc, và xóa khi chưa đọc trong một nghiên cứu [71] vào năm 2016. Nghiên cứu này được thực hiện tại Yahoo! Labs với lượng dữ liệu nghiên cứu khổng lồ gồm dữ liệu của hơn 100.000 người dùng thư điện tử Yahoo. Nghiên cứu chỉ ra những hành động phổ biến nhất là đọc, trả lời, xóa và tập con của hành động xóa là xóa mà không đọc thư. Trong bài báo này, các tác giả cho rằng hành động và tầm quan trọng không thể đánh đồng với nhau. Một bức thư có hành động "đọc" có thể là rất quan trọng. Ví dụ, một bức thư có nội dung xác nhận công việc, tuy không cần người dùng thực hiện thêm hành động tiếp theo nhưng việc người dùng đọc được bức thư đó là rất quan trọng.

Trong phương pháp này, hành động gợi ý cho một bức thư đến được dự đoán bằng một mô hình kết hợp giữa phương pháp học máy theo chiều dọc (vertical learning) và theo chiều ngang (horizontal learning). Học máy theo chiều dọc là chỉ việc xây dựng

các mô hình phân loại cho mỗi người dùng cá nhân, dự đoán hành động dựa theo dữ liệu trong quá khứ của chính người dùng đó. Học máy theo chiều ngang là chỉ việc dự đoán hành động cho một người dùng dựa trên lịch sử sử dụng của những người dùng khác, khi một bộ phận người dùng có quá ít tương tác với hòm thư của họ.

Các đặc trưng mà nghiên cứu của Di Castro sử dụng bao gồm đặc trưng cục bộ, nghĩa là các đặc trưng được tính toán từ hòm thư của người dùng, và thuộc tính toàn cục, nghĩa là các đặc trưng được tính toán trên toàn bộ tập dữ liệu. Các thuộc tính cục bộ bao gồm nội dung, header thư, hành vi sử dụng của người dùng trong quá khứ đối với những bức thư tương tự bức thư cần dự đoán hành động. Các thuộc tính toàn cục bao gồm tổng số thư đã được gửi bởi người gửi, số lượng thư được áp dụng hành động trong số những thư đã gửi của người gửi.

Nghiên cứu của Yang và cộng sự [74] vào năm 2017 có một cách tiếp cận khác biệt về bài toán dự đoán hành động người dùng, đó là bỏ qua tính cá nhân hóa của bài toán này. Đối với một bức thư được gửi cho nhiều người, bài báo này giải quyết bài toán dự đoán xem bức thư đó có được trả lời (bởi bất kỳ ai trong số những người nhận) hay không. Bài báo đã đề xuất một số phương pháp trích chọn đặc trưng (HistIndiv, HistPair, Temporal…) và so sánh với phương pháp túi từ truyền thống cùng một vài phương pháp trích chọn đặc trưng đơn giản khác.

Vào năm 2019, Mukherjee và Jiang đã đề xuất một phương pháp [82] để giải quyết bài toán dự đoán hành động trả lời thư. Nghiên cứu này có mục tiêu dự đoán hành động mang tính cá nhân hóa và được tiếp cận dưới hình thức xây dựng hệ gợi ý. Nói theo cách khác, với cùng một bức thư khi được gửi cho nhiều người, hệ gợi ý sẽ dự đoán hành động trả lời thư cho từng người nhận. Bài toán có độ khó cao hơn so với toán đặt ra bởi Yang và cộng sự [74]. Trong nghiên cứu của Mukherjee và Jiang, mỗi người nhận được biểu diễn dựa trên những bức thư mà người đó đã nhận trong quá khứ. Vector đại diện người dùng được so sánh với vector biểu diễn bức thư cần dự đoán để tính ra mức độ tương tự mà trong bài báo được gọi là đặc trưng `similarity`. Đặc trưng `similarity` được bổ sung vào đặc trưng văn bản của bức thư để làm đầu vào cho máy phân loại. Nghiên cứu rút ra nhận xét rằng, với các máy phân loại truyền thống như *logistic regression* hay cây quyết định tăng cường độ dốc (*gradient- boosted decision*

trees), kỹ thuật biểu diễn đặc trưng TF-IDF có hiệu quả tốt hơn so với kỹ thuật biểu diễn từ ngữ fasttext [77]. Tuy nhiên, đặc trưng fasttext tỏ ra hiệu quả hơn khi kết hợp với mạng nơ-ron sâu CNN-MLP. Máy phân loại CNN-MLP còn thể hiện một ưu điểm nữa đó là với các cách biểu diễn đặc trưng khác nhau thì mô hình này vẫn cho kết quả gần với kết quả tốt nhất.

1.3.3. Nghiên cứu về xếp hạng thư điện tử

Ý tưởng sắp xếp hòm thư để giải quyết vấn đề quá tải thư điện tử đã được đề xuất từ lâu, trong nghiên cứu của Neustaedter và cộng sự [24]. Trong nghiên cứu nói trên, công cụ sắp xếp hòm thư theo một số đặc trưng xã hội của thư điện tử đã được đề xuất nhằm hỗ trợ người dùng trong việc tìm kiếm thư quan trọng. Nghiên cứu của nhóm Neustaedter đề xuất 11 đặc trưng xã hội được trích xuất bằng cách thống kê số lượng thư được gửi và nhận giữa những người dùng. Xét một bức thư được gửi từ *người gửi* đến *người nhận*, một trong số các đặc trưng là số lượng thư mà người *người gửi* đã từng gửi cho *người nhận*. Một đặc trưng khác là số lượng thư *đã được đọc* trong số các bức thư nói trên. Với công cụ này, người dùng phải chọn tiêu chí sắp xếp và phát hiện các bức thư quan trọng một cách thủ công. Công cụ không có chức năng tự động chỉ ra bức thư nào là quan trọng, cần phải được ưu tiên xử lý.

Từ việc nhận thấy các hệ thống lọc thư rác và dự đoán hành động chưa triệt để giải quyết vấn đề quá tải thư điện tử, trong những năm gần đây, các nhà khoa học đã đề xuất phương pháp xếp hạng thư điện tử với mục tiêu sắp xếp hòm thư của người dùng một cách tối ưu hóa hơn nữa. Với nhiều mức độ phân loại hơn so với hai bài toán trước đó, cộng thêm các khó khăn đến từ quyền riêng tư cũng như tính cá nhân hóa, đây là bài toán có độ khó cao trong số những bài toán về xử lý thư điện tử. Trong số những nghiên cứu về xếp hạng thư điện tử, các phương pháp mang tính cá nhân hóa [40, 49, 51] chiếm đa số bởi ứng dụng thực tế của nó. Tuy chưa có nhiều phương pháp được công bố, hướng nghiên cứu về xếp hạng thư điện tử là một hướng nghiên cứu có tính cấp thiết, có tiềm năng giúp tăng năng suất làm việc cho người sử dụng.

Trước khi các phương pháp xếp hạng thư điện tử ra đời, phân loại thư điện tử theo chủ đề và thư mục cũng là một hướng nghiên cứu đáng chú ý. Nghiên cứu của Alsmadi và Alhami vào năm 2015 [67] là một nghiên cứu về phân cụm thư điện tử và đánh giá

hiệu quả phân cụm bằng phân loại. Nghiên cứu có mục tiêu phân loại thư điện tử thành 3 mục: *personal*, *progessional* và *other*. Việc phân loại này có mục tiêu làm giảm thiểu thời gian xử lý thư của người dùng vì nó cho phép người dùng tập trung vào một chủ đề mà họ quan tâm. Nghiên cứu này đề xuất việc sử dụng những bức thư được gán nhãn bằng thuật toán phân cụm k-Means để làm dữ liệu huấn luyện và thử nghiệm cho mô hình phân loại bằng một thuật toán học máy có giám sát SVM. Thông qua thí nghiệm, Alsmadi và Alhami đã so sánh các kỹ thuật biểu diễn thuộc tính khác nhau và tìm ra phương án tốt nhất cho tập dữ liệu của họ là phương pháp N-Grams.

Tuy nhiên, phương án tự động gán nhãn bằng thuật toán phân cụm và phân loại bằng thuật toán học máy có giám sát là một giả định thiếu thực tế. Nó đòi hỏi thuật toán phân cụm phải có độ chính xác tuyệt đối thì kết quả thí nghiệm của nghiên cứu này mới có nghĩa. Tuy thí nghiệm cho thấy hiệu quả khá cao của phương pháp với $precision_{\mu}$ từ 0.807 đến 0.976 và $recall_{\mu}$ từ 0.802 lên tới 0.975 nhưng kết quả này không có ý nghĩa bởi vì không có cơ sở để khẳng định tính đúng đắn của tập dữ liệu huấn luyện được gán nhãn bằng thuật toán phân cụm.

Trong nghiên cứu của Klimt và Yang [19], ngoài việc giới thiệu tập dữ liệu Enron thì các tác giả cũng thực hiện phân loại thư điện tử trên tập dữ liệu này theo từng đặc trưng riêng biệt để phân tích sơ lược về tập dữ liệu. Phương pháp TF-IDF theo dạng *ltc* được sử dụng để biểu diễn các bức thư dưới dạng vector đặc trưng. Thuật toán phân loại SVM được dùng để tiến hành phân tích tập dữ liệu Enron bằng nhiều phương án. Ở phương án thứ nhất, các thuộc tính "From", "Subject", "Body", "To, CC" được sử dụng riêng biệt để phân loại thư theo thư mục. Với phương án thứ hai, tất cả các thuộc tính được gộp lại và biểu diễn dưới dạng một vector túi từ duy nhất dùng làm đầu vào phân loại. Cuối cùng, các máy phân loại của các thuộc tính riêng biệt được kết hợp với nhau một cách tuyến tính với trọng số cho mỗi máy phân loại. Các trọng số được học bằng thuật toán ridge regression.

Nghiên cứu của Bekkerman [20] dùng 4 thuật toán Maximum Entropy, Naïve Bayes, SVM và Wide-Margin Winnow để phân loại thư điện tử theo thư mục trên tập dữ liệu Enron và SRI. Chỉ những thư mục gắn với chủ đề mới được giữ lại để làm thí nghiệm, các thư mục chung như Inbox, Sent đều bị loại bỏ. Vector đặc trưng là vector dạng bag-

of-words với mỗi phần tử là số lần xuất hiện của một từ trong tập từ vựng. Nghiên cứu này loại bỏ 100 từ phổ biến nhất và những từ chỉ xuất hiện một lần trong toàn tập dữ liệu. Tiêu chí accuracy được chọn để báo cáo kết quả thí nghiệm. Thí nghiệm cho thấy, SVM có hiệu quả trung bình cao hơn và chiếm ưu thế so với các thuật toán khác đối với 10 trên tổng số 14 hòm thư của người dùng. Bekkerman đưa ra nhận định rằng độ chính xác thu được từ các thí nghiệm là "tương đối thấp" khi có tới 9 trong số 14 trường hợp mà tiêu chí accuracy nhỏ hơn 70%.

Khác với bài toán phân loại nói trên, bài toán xếp hạng thư điện tử nhấn mạnh vào mức độ quan trọng của các bức thư đối với người dùng. Nghiên cứu đầu tiên đề xuất một mô hình xếp hạng thư điện tử được công bố vào năm 2009 bởi Yoo và cộng sự [40]. Nghiên cứu này sử dụng một số đặc trưng mạng xã hội kết hợp với nội dung thư để giải bài toán xếp hạng thư điện tử. Bài toán được đặt ra là xếp hạng thư điện tử với 5 mức độ ưu tiên được đánh số từ 1 tới 5. Những đặc trưng mạng xã hội này được trích xuất từ dữ liệu không gán nhãn bằng phương pháp phân cụm. Thuật toán Newman Clustering được sử dụng để phân cụm người dùng trên mạng thư điện tử cá nhân. Từ kết quả phân cụm mạng thư điện tử cá nhân của người dùng, 07 chỉ số của người gửi thư dựa trên đồ thị được tính toán để sử dụng làm đặc trưng mạng xã hội của một bức thư. Cụm của người gửi trong mạng thư điện tử cũng được trích xuất làm một đặc trưng cho bức thư. Những đặc trưng xã hội nói trên được kết hợp cùng đặc trưng nội dung thư dưới dạng vector nhị phân. Các máy phân loại nhị phân SVM được xây dựng để kết hợp với nhau thành máy phân loại 5 lớp. Tuy Yoo và các cộng sự không nêu tên phương án kết hợp các máy phân loại nhị phân thành máy phân loại đa lớp nhưng theo như mô tả của bài báo thì phương án OVA đã được áp dụng trong thí nghiệm.

Aberdeen và cộng sự là nhóm nghiên cứu đứng sau hệ thống Gmail. Nhóm đã công bố một phương pháp xếp hạng thư điện tử [45] vào năm 2010. Phương pháp này định nghĩa mức độ quan trọng của một bức thư dựa trên xác suất mà người nhận sẽ thực hiện một hành động trên bức thư đó. Cách tiếp cận này khác với bài toán dự đoán hành động ở chỗ nó không phân biệt giữa các hành động khác nhau (ví dụ: giữa hành động đọc và trả lời thư). Trong phương pháp này, một mô hình phân loại logistic regression được xây dựng riêng biệt dành cho mỗi người sử dụng. Dữ liệu huấn luyện là những bức thư

nhận được trong quá khứ kèm theo hành động mà người dùng đã thực hiện trên các bức thư đó, bao gồm cả hành động "chưa đọc" thư. Mô hình thực hiện dự đoán và đánh dấu quan trọng đối với các bức thư có xác suất cao. Tính cá nhân hóa được thể hiện rõ rệt trong phương pháp này. Tuy nhiên, còn một điểm tồn tại đó là nếu người dùng quá bận rộn và bỏ sót một số thư quan trọng, những bức thư tương tự trong tương lai cũng sẽ không được đánh dấu quan trọng.

Vào năm 2011, nhóm Yoo và cộng sự tiếp tục công bố một nghiên cứu [49] đề xuất cách giải quyết bài toán xếp hạng thư điện tử với 5 nhãn, tương ứng với 5 cấp độ quan trọng của một bức thư, dựa trên dữ liệu thư điện tử cá nhân. Nghiên cứu này đã so sánh hiệu quả phân loại đa lớp giữa phương pháp hồi quy thứ bậc với sự kết hợp của nhiều máy phân loại nhị phân. Nghiên cứu này chỉ sử dụng các đặc tính về nội dung của các bức thư. Nhiều máy phân loại nhị phân có thể được kết hợp theo một số phương án để trở thành máy phân loại đa lớp. Trong mỗi phương án đó, khi dự đoán lớp của một mẫu, mỗi máy phân loại "bỏ phiếu" cho một lớp, rồi kết quả cuối cùng được tổng hợp theo một quy tắc nhất định.

Trong nghiên cứu này, Yoo và cộng sự áp dụng hai phương pháp học máy là hồi quy thứ tự dựa trên SVM (SVOR) và phân loại đa lớp SVM, kết hợp nhiều máy phân loại SVM theo các cách: OVA, OVO, DAGSVM [8], OB-MV và OB-MC. OB-MV là phương án kết hợp các máy phân loại nhị phân giống như phương án DAG nhưng các máy phân loại được sắp xếp theo thứ tự mức độ quan trọng và kết quả được tính dựa trên biểu quyết số đông. OB-MC có cách sắp xếp các máy phân loại nhị phân giống như phương án OB-MV nhưng kết quả được tính dựa trên máy phân loại có điểm số cao nhất. Vector đặc trưng được các tác giả trích xuất từ các trường người gửi, người nhận, CC, tiêu đề và nội dung thư. Kết quả cho thấy, với bộ dữ liệu thư điện tử cá nhân do các tác giả tự thu thập, OB-MV là phương pháp hiệu quả nhất và SVOR có kết quả thấp nhất. Như vậy, sự kết hợp của nhiều máy phân loại nhị phân cho hiệu quả dự đoán cao hơn so với phương pháp hồi quy thứ bậc trên tập dữ liệu thư điện tử cá nhân. Kết quả này một phần cho thấy các mức độ ưu tiên của thư điện tử không có quan hệ tương đối với nhau.

Trái lại, đối với 7 tập dữ liệu chuẩn lấy từ thư viện UCI, SVOR lại cho kết quả tốt nhất, OB-MV và OB-MC cho kết quả trung bình và OVA cho kết quả thấp nhất. Để lý giải cho việc SVOR cho kết quả khác nhau trên tập dữ liệu thư điện tử cá nhân và các tập dữ liệu chuẩn, các tác giả tiến hành phân tích các tập dữ liệu nói trên. Từ đó rút ra được kết luận rằng SVOR cho kết quả tốt với các tập dữ liệu chuẩn có phân bổ tuyến tính nhưng không đạt kết quả tốt với dữ liệu thư điện tử cá nhân. Giả thiết về sự tuyến tính của dữ liệu là rất hạn chế với bài toán xếp hạng thư điện tử cá nhân vì quan điểm về sự ưu tiên của mỗi người là khác nhau. Vì vậy, các phương pháp phân loại tỏ ra hiệu quả hơn các phương pháp hồi quy. Thí nghiệm của Yoo và cộng sự [49] đã trả lời một số câu hỏi về đặc điểm của dữ liệu thư điện tử và đã so sánh hiệu quả xếp hạng giữa mô hình phân loại và hồi quy. Tuy nhiên, mô hình phân loại được đưa ra chưa đạt được hiệu quả cao trong dự đoán mức độ ưu tiên.

Nghiên cứu của Long và cộng sự vào năm 2020 [85] đề xuất phương pháp phân loại thư điện tử thành 02 mức độ: *quan trọng* và *không quan trọng*. Đặc trưng của thư điện tử được sử dụng trong nghiên cứu này là tiêu đề, người gửi, nội dung và được biểu diễn bằng phương pháp TF-IDF. Long và cộng sự đã thực hiện thu thập và xây dựng tập dữ liệu thư điện tử tiếng Việt bằng cách tải dữ liệu hòm thư của 17 người dùng từ hệ thống Gmail. Nghiên cứu đã sử dụng nhãn *quan trọng* trong trường header ‘google_label’ của các bức thư để làm nhãn cho tập dữ liệu nói trên. Hạn chế của cách làm này là nhãn của dữ liệu phụ thuộc vào độ chính xác của thuật toán phân loại thư quan trọng mà Gmail sử dụng. Trong nghiên cứu, Long và cộng sự đã tiến hành so sánh các thuật toán Random Forest, KNN và Logistic Regression trên tập dữ liệu nói trên. Nghiên cứu đã kết luận thuật toán Random Forest có độ chính xác cao hơn hai thuật toán còn lại, thể hiện bởi các tiêu chí F_1 (1.16), Recall (1.14) và AUC.

1.3.4. Các tiêu chí đánh giá

1.3.4.1. Đánh giá mô hình phân loại nhị phân

Mô hình phân loại nhị phân được ứng dụng trong bài toán lọc thư rác. Có nhiều tiêu chí được dùng để đo hiệu quả của mô hình phân loại nhị phân. Mỗi tiêu chí lại có tác dụng đánh giá một khía cạnh riêng của mô hình. Tuy sở hữu những đặc trưng riêng biệt,

các tiêu chí đánh giá này đều được tính toán dựa trên các định lượng cơ bản. Các khái niệm định lượng cơ bản nhất thể hiện kết quả của bộ lọc thư rác đó là:

– *True positive (tp)*: Số lượng thư rác được dự đoán chính xác.
– *False positive (fp)*: Số lượng thư hợp lệ mà bộ lọc cảnh báo nhầm thành thư rác.
– *True negative (tn)*: Số thư hợp lệ mà bộ lọc nhận diện chính xác.
– *False negative (fn)*: Số lượng thư rác được bộ lọc coi là thư hợp lệ.

Đối với bài toán về thư điện tử, trường hợp *fp* được coi là gây hậu quả lớn hơn so với *fn* [36], cần phải ưu tiên tránh lỗi này. Sai số *fp* của một bộ lọc thư rác làm cho người dùng bỏ lỡ một bức thư hợp lệ, gây tổn thất thông tin hoặc gián đoạn liên lạc. Trong nhiều trường hợp, khi bức thư đó có nội dung rất quan trọng, sai số *fp* sẽ gây hậu quả nặng nề cho người dùng và không thể được chấp nhận. Trong khi đó, *fn* là lỗi thường có thể chấp nhận được nếu không xảy ra quá nhiều. Việc bộ lọc để lọt một vài bức thư rác vào hòm thư có thể gây mất thời gian và phiền phức cho người dùng nhưng nhìn chung là không có tác hại lớn. Các công cụ lọc thư rác thường được tính toán sao cho độ đo *fp* bằng 0 hoặc càng gần 0 càng tốt. Đối với một bộ lọc thư rác lý tưởng thì cả hai độ đo *fp* và *fn* đều đạt được mức 0 nhưng các bộ lọc trên thực tế luôn luôn có sai số lớn hơn 0. Vì vậy, để tránh thiệt hại cho người sử dụng, ta có thể hy sinh độ đo *fn* để đổi lại sự tối ưu cho *fp* bằng cách điều chỉnh độ nhạy của bộ lọc.

Kết hợp các chỉ số nói trên, ta được những chỉ số có ý nghĩa hơn, thể hiện được tính chất của bộ lọc. Đầu tiên đó là chỉ số recall, còn được gọi là độ đo triệu hồi, được tính bằng công thức (1.14). Recall thể hiện độ nhạy của bộ lọc [41] cũng như độ hoàn thiện của kết quả. Giá trị recall bằng 1.0 khi đánh giá một bộ lọc thư rác được hiểu là "tất cả các bức thư rác đều được phát hiện". Tuy nhiên, kết quả này có thể đạt được một cách dễ dàng bằng việc thiết kế một bộ lọc thư rác để chặn mọi bức thư đến mà không cần quan tâm đó là thư rác hay thư hợp lệ.

$$recall = \frac{tp}{tp + fn} \tag{1.14}$$

Một tiêu chí có thể được coi là đối nghịch với recall đó là precision. Precision được tính bằng công thức (1.15). Precision thể hiện độ tin cậy của kết quả dự đoán. Khi precision bằng 1.0, tất cả các dự đoán đều chính xác hay nói cách khác, *fp* bằng 0. Có

thể nói hai tiêu chí này đối nghịch với nhau bởi vì với một bộ lọc không hoàn hảo, khi tăng precision sẽ kéo theo giảm recall và ngược lại. Trên thực tế, để đạt được giá trị precision lý tưởng thì chỉ số recall sẽ bị giảm xuống thấp. Bởi vậy cho nên khi thiết kế bộ lọc người ta thường đưa ra một tiêu chí mục tiêu để tối ưu các tiêu chí khác theo đó.

$$precision = \frac{tp}{tp + fp} \tag{1.15}$$

Một tiêu chí kết hợp cả hai tiêu chí vừa đề cập là chỉ số F_1, được tính bằng công thức (1.16). Chỉ số F_1 cao cho thấy cả recall và precision đều cao, nghĩa là kết quả dự đoán đồng thời đạt được tính đầy đủ và đáng tin cậy.

$$F_1 = \frac{2 \times precision * recall}{precision + recall} \tag{1.16}$$

Ngoài ra thì nhiều nghiên cứu cũng sử dụng chỉ số accuracy như một thước đo hiệu quả. Chỉ số này được tính bằng công thức (1.17). Accuracy chính là tỷ lệ các dự đoán đúng (*tp* + *tn*) trên tổng số lần dự đoán.

$$accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{1.17}$$

1.3.4.2. Đánh giá mô hình phân loại đa lớp

Mô hình phân loại đa lớp được ứng dụng cho bài toán dự đoán hành động người dùng và bài toán xếp hạng thư điện tử. Bởi vì số lượng lớp phân loại lớn hơn 2 nên các tiêu chí recall, precision và F_1 không áp dụng được cho những mô hình này. Một bài báo công bố năm 2009 [41] đã đưa ra 8 tiêu chí đánh giá cho bài toán phân loại đa lớp. Những tiêu chí đánh giá đó là: average accuracy (1.18), error rate (1.19), $precision_{\mu}$ (1.20), $recall_{\mu}$ (1.21), $Fscore_{\mu}$ (1.22), $precision_{m}$ (1.23), $recall_{m}$ (1.24) và $Fscore_{m}$ (1.25). Giả sử bài toán phân loại có l lớp, các tiêu chí nói trên được biểu diễn bởi những công thức sau đây.

$average\ accuracy = \frac{\sum_{i=1}^{l} \frac{tp_i + tn_i}{tp_i + fn_i + fp_i + tn_i}}{l}$	(1.18)

$$error\ rate = \frac{\sum_{i=1}^{l} \frac{fp_i + fn_i}{tp_i + fn_i + fp_i + tn_i}}{l}$$	(1.19)
$$precision_{\mu} = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fp_i)}$$	(1.20)
$$recall_{\mu} = \frac{\sum_{i=1}^{l} tp_i}{\sum_{i=1}^{l} (tp_i + fn_i)}$$	(1.21)
$$Fscore_{\mu} = \frac{(\beta^2 + 1) \times Precision_{\mu} * Recall_{\mu}}{\beta^2 * Precision_{\mu} + Recall_{\mu}}$$	(1.22)
$$precision_m = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l}$$	(1.23)
$$recall_m = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l}$$	(1.24)
$$Fscore_m = \frac{(\beta^2 + 1) * Precision_m * Recall_m}{\beta^2 * Precision_m + Recall_m}$$	(1.25)

Tiêu chí average accuracy ở công thức (1.18) là trung bình cộng tiêu chí accuracy riêng của tất cả các lớp. Tiêu chí error rate ở công thức (1.19) có giá trị tương đương với ($1.0 - average\ accuracy$), là trung bình cộng tỷ lệ dự đoán sai của các lớp.

Ta có hai cách để tính giá trị trung bình của các tiêu chí là micro (μ) và macro (m). Cách tính macro là tính các chỉ số đối với từng lớp (trong số l lớp) sau đó lấy trung bình cộng l kết quả đó. Cách tính micro là lấy tổng các thông số (*tp*, *fp*, *tn*, *fn*) của tất cả các lớp lại rồi mới tính chỉ số. Cách này có đặc điểm là kết quả trung bình sẽ nghiêng nhiều hơn về tiêu chí của lớp có đông cá thể nhất trong khi cách macro thì đối xử với các lớp như nhau kể cả khi có những lớp có lượng cá thể nhiều hơn hẳn.

1.4. TẬP DỮ LIỆU THƯ ĐIỆN TỬ

1.4.1. Tập dữ liệu Enron

Enron là một tập dữ liệu thư điện tử lớn với phiên bản gốc được đưa công khai lên Internet vào năm 2003, sau một cuộc điều tra đối với tập đoàn Enron, Hoa Kỳ. Trong phiên bản gốc, Enron có 619,446 bức thư thuộc về 158 người dùng. Tập dữ liệu được

dọn dẹp và công bố lại vào năm 2004 [19] với số lượng mẫu giảm xuống còn 200,399 bức thư, còn số người dùng thì được giữ nguyên. Hòm thư của mỗi người dùng được chia thành nhiều thư mục và số lượng thư mục của mỗi tài khoản là khác nhau, từ một vài thư mục cho tới hơn 100 thư mục. Số lượng thư của mỗi tài khoản cũng đa dạng, từ những tài khoản ít hoạt động với chỉ một vài bức thư, tới hàng chục, hàng trăm và hàng chục nghìn bức thư. Tuy vậy, tập dữ liệu Enron không được gán nhãn. Vì lý do này, ngoài tác vụ phân loại thư điện tử theo thư mục thì nó không thể được sử dụng trực tiếp vào những bài toán khác như bài toán lọc thư rác, xếp hạng thư điện tử hoặc dự đoán hành động người dùng. Thêm nữa, tập dữ liệu bị hạn chế về ngôn ngữ, chỉ có thư tiếng Anh mà không phải là một ngôn ngữ khác như tiếng Việt.

1.4.2. Tập dữ liệu TREC

Hội nghị TREC hỗ trợ hạ tầng cơ sở vật chất và công nghệ cho các nghiên cứu trong lĩnh vực khai phá dữ liệu từ năm 1992. Một trong những tài nguyên quan trọng của TREC là cơ sở dữ liệu mẫu trong nhiều lĩnh vực, trong đó có lĩnh vực lọc thư rác. Hội nghị TREC có nhiều nhánh, trong đó có một nhánh về lọc thư rác là Spam Track [23]. Trong các năm 2005, 2006 và 2007, TREC đã cung cấp 04 tập dữ liệu mẫu về thư rác trong đó có 03 tập dữ liệu mẫu về tiếng Anh và 01 tập dữ liệu mẫu về tiếng Trung. Các tập dữ liệu này đều được thu thập từ các máy chủ thư điện tử thực tế trong một thời gian khoảng 3 tháng. Sau đó do các chuyên gia về thư rác xử lý, phân loại là thư rác hay là thư bình thường, sau đó tập hợp lại thành các tập dữ liệu chuẩn và cung cấp cho các nhà nghiên cứu trong lĩnh vực thư rác.

Bảng 1.1: Các tập dữ liệu công khai về thư điện tử

Tập dữ liệu	Ngôn ngữ	Tổng số thư	Năm công bố	Nhãn
Enron	Tiếng Anh	200,399	2003	Không
SRI	Tiếng Anh	22,000	-	Thư mục
UC Berkeley	Tiếng Anh	1,700	2006	Thể loại
LingSpam	Tiếng Anh	2,893	2003	Thư rác
TREC 2005	Tiếng Anh	92,189	2005	Thư rác
TREC 2006	Tiếng Anh	37,822	2006	Thư rác
TREC 2006	Tiếng Trung	64,620	2006	Thư rác
TREC 2007	Tiếng Anh	50,199	2007	Thư rác

Dữ liệu thư điện tử của TREC lưu trữ toàn bộ nội dung thư và phần header của thư, vì vậy hoàn toàn có thể sử dụng để tiến hành các nghiên cứu về lọc thư rác dựa trên nội dung, như dựa trên từ khóa, dựa trên phương pháp phân loại văn bản. Đồng thời, dữ liệu thư điện tử TREC cũng có thể dùng để nghiên cứu các nghiên cứu về lọc thư rác dựa trên xác thực địa chỉ người gửi như SPF.

Một điểm cần lưu ý là tập dữ liệu thư điện tử TREC chỉ bao gồm những thư điện tử gửi đến một máy chủ thư điện tử nào đó chứ không bao gồm những thư điện tử gửi từ máy chủ đó đi, vì vậy sử dụng các dữ liệu TREC để nghiên cứu về mạng xã hội, mạng thư điện tử còn nhiều hạn chế. Ngoài ra, dữ liệu thư điện tử của TREC chỉ phân loại thư rác và thư bình thường, không có thông tin về người dùng (như chức vụ), vì vậy, dữ liệu này ít khi dùng để phân tích những phương pháp xếp hạng người dùng. Các tập dữ liệu về lọc thư rác của hội thảo TREC có thể được tải về từ địa chỉ: *https://trec.nist.gov/data/spam.html*.

1.4.3. Các tập dữ liệu khác

Một dự án nghiên cứu thuộc trường Đại học Berkeley (California, USA) cung cấp một tập dữ liệu thư điện tử dựa trên tập dữ liệu Enron. Các bức thư trong tập dữ liệu của UC Berkeley được gán nhãn theo thể loại (categories) và có thể được tải về miễn phí từ địa chỉ: *http://bailando.sims.berkeley.edu/enron_email.html*. Tập dữ liệu này gồm có 1,700 bức thư, mỗi bức thư được gán với một số thể loại (trong số 53 thể loại) liên quan đến nội dung của nó, cùng với trọng số của thể loại đó đối với bức thư.

Tập dữ liệu SRI cũng được nhắc tới trong một số nghiên cứu. Tập dữ liệu này được mô tả là có chứa 22,000 bức thư của 196 người dùng [20]. Mỗi tài khoản cũng được chia thành các thư mục giống như trong tập dữ liệu Enron. Tuy nhiên, ở thời điểm của báo cáo này, tập dữ liệu SRI đã không còn được công bố hoặc nhắc đến trên đường dẫn mà bài báo [20] cung cấp.

LingSpam [15] là tập dữ liệu bao gồm gồm 2,893 bức thư, trong đó có 481 bức thư rác và 2,412 bức thư hợp lệ được thu thập từ một danh sách gửi thư có tên Linguist. Linguist là một danh sách gửi thư được kiểm duyệt có chủ đề về học thuật và nghề nghiệp trong lĩnh vực ngôn ngữ. Tập dữ liệu đã được giới thiệu vào năm 2003 và đã

được sử dụng để thử nghiệm một phương pháp phát hiện thư rác dựa trên phân loại văn bản [15].

1.4.4. Tập dữ liệu thư điện tử tiếng Việt

Ở phần trên, luận án đã liệt kê và so sánh một số tập dữ liệu công khai trên thế giới về thư điện tử. Hầu hết các tập dữ liệu công khai về thư điện tử có ngôn ngữ là tiếng Anh. Tập dữ liệu thuộc về các ngôn ngữ khác chiếm số lượng nhỏ, trong đó chưa có tập dữ liệu thư điện tử tiếng Việt được công bố. Tình trạng khan hiếm tập dữ liệu nói trên gây khó khăn cho các nghiên cứu có phạm vi là thư điện tử tiếng Việt. Để có thể thực hiện thí nghiệm đánh giá các mô hình đề xuất trong chương này cũng như trong các chương tiếp theo, luận án đã tiến hành tự xây dựng một tập dữ liệu thư điện tử có nội dung tiếng Việt. Phần này sẽ miêu tả chi tiết quá trình xây dựng tập dữ liệu. Dữ liệu thư điện tử được thu thập từ một số tình nguyện viên. Tập dữ liệu thô bao gồm 37,003 bức thư đến từ 7 tình nguyện viên. Các tình nguyện viên thực hiện sao lưu toàn bộ dữ liệu từ hệ thống Gmail bằng tính năng Google Takeout[8]. Dữ liệu thư điện tử tải về bằng tính năng Google Takeout được chia theo cấu trúc thư mục của người dùng trên Gmail, mỗi thư mục tương ứng với một tập tin có định dạng *mbox*. Các bức thư được tách ra từ tập tin mbox bằng thư viện Mail::Mbox::MessageParser[9] và được phân tích bằng thư viện MIME::Parser[10] của ngôn ngữ lập trình Perl để trích xuất những phần thông tin như tiêu đề, nội dung, các trường header… Các bức thư được phân biệt với nhau bằng trường *Message-ID* trong header của bức thư. Toàn bộ các thông tin trích xuất được từ các bức thư được lưu trên một cơ sở dữ liệu MySQL để tiến hành xử lý tiếp.

1.4.4.1. Loại bỏ thư có nội dung trùng lặp

Loại bỏ thư trùng lặp là một tác vụ quan trọng khi xây dựng tập dữ liệu thư điện tử. Thư trùng lặp trong dữ liệu thô có thể được sinh ra bởi phần mềm quản lý thư để cung cấp một số tính năng cho người sử dụng và cần được loại bỏ để đảm bảo việc huấn luyện mô hình phân loại thư đạt được mục tiêu đề ra là mô phỏng được cách mà con người phân loại thư điện tử [19]. Ngoài ra, các trường hợp thư bị trùng lặp phổ biến gồm có thư được các tình nguyện viên gửi cho nhau, thư gửi tự động từ các nền tảng

[8] https://takeout.google.com/settings/takeout
[9] https://metacpan.org/pod/Mail::Mbox::MessageParser
[10] https://metacpan.org/pod/MIME::Parser

web, các ứng dụng di động. Các tập dữ liệu được trình bày ở trên như Enron, TREC… đều sử dụng phương pháp thủ công để loại bỏ thư trùng lặp. Cách làm này tuy có độ tin cậy cao nhưng tốn nhiều thời gian và công sức. Luận án đề xuất một phương pháp bán tự động để loại bỏ những bức thư trùng lặp nhằm tiết kiệm thời gian trong khi vẫn đảm bảo mức độ tin cậy cao. Để tự động tìm những bức thư có nội dung trùng lặp để loại bỏ, luận án sử dụng một công cụ phát hiện thư trùng lặp dựa trên phương pháp tính khoảng cách Euclidean giữa những bức thư. Kết quả phát hiện của công cụ nói trên được kiểm duyệt một cách thủ công trước khi xác nhận loại bỏ các bức thư trùng lặp. Dưới đây là phương pháp cụ thể nhằm xác định thư trùng lặp đã được áp dụng.

Mỗi bức thư được vector hóa dựa trên nội dung thư thành một vector túi từ nhị phân. Cụ thể, nội dung của một bức thư được tách ra thành các từ bằng cách phân chia văn bản dựa vào ký tự khoảng trống. Khi đó, có thể biểu diễn mỗi bức thư thành một vector có dạng $m = \{w_i \mid i = 1, 2 \dots n\}$. Trong đó, w_i, là những từ ngữ được lấy từ tiêu đề và nội dung thư ghép với nhau. Cách tính khoảng cách Euclidean được thể hiện trong công thức (2.1).

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)} \tag{2.1}$$

Trong công thức trên, p và q là vector biểu diễn hai bức thư phân biệt trong tập dữ liệu. Trong luận án, công cụ sẽ tự động tìm ra những bức thư có mức độ tương tự từ 75% trở lên so với từng bức thư trong tập dữ liệu. Công cụ này đo lường mức độ giống nhau của tất cả các cặp thư điện tử trong tập dữ liệu dựa trên khoảng cách Euclidean. Danh sách những bức thư tương tự được phát hiện bởi công cụ sẽ được kiểm tra và xác nhận một cách thủ công. Cách làm này cho phép tìm và loại bỏ các bức thư tương tự một cách nhanh chóng và chính xác.

1.4.4.2. Loại bỏ thư tiếng nước ngoài

Chỉ những bức thư có nội dung tiếng Việt được giữ lại và đưa vào tập dữ liệu. Ngôn ngữ mà những tình nguyện viên sử dụng là tiếng Việt. Tuy nhiên, trên thực tế, họ nhận được một số lượng nhất định những bức thư được viết bằng các ngôn ngữ khác như tiếng Anh, tiếng Trung Quốc… Để tập dữ liệu đáp ứng được mục tiêu nghiên cứu của

luận án, cần phát hiện và loại bỏ những bức thư nói trên. Sau đây, phương pháp phát hiện những bức thư không phải là tiếng Việt sẽ được trình bày.

Danh sách 10,000 từ tiếng Anh phổ biến nhất[11] trích xuất từ tập dữ liệu văn bản của Google đã được sử dụng. Một danh sách gồm 5,847 từ đơn tiếng Việt đã được tác giả xây dựng bằng cách tổng hợp từ một số nguồn[12]. Danh sách này bao gồm những từ tiếng Việt có dấu và những từ tiếng Việt không dấu có từ 2 ký tự trở lên và không tồn tại trong danh sách 10,000 từ tiếng Anh nói trên. Để xác định một văn bản là tiếng Việt hay tiếng nước ngoài, có hai chỉ số được tính toán. Thứ nhất là tỷ lệ từ có trong danh sách từ đơn tiếng Việt trên tổng số từ của văn bản, gọi là V. Thứ hai là tỷ lệ từ có trong danh sách 10,000 từ tiếng Anh, ký hiệu là E. Nếu văn bản có tỷ lệ $E > 50\%$ thì công cụ kết luận bức thư là tiếng Anh. Nếu $E < 50\%$ và $V \geq 35\%$ thì công cụ kết luận bức thư là tiếng Việt. Những bức thư mà công cụ không đưa ra được kết luận (có $E < 50\%$ và $V < 35\%$) sẽ được gán nhãn ngôn ngữ thủ công.

1.4.4.3. Các bước tiền xử lý khác

Ngoài ra, những bức thư có nội dung quá ngắn, những bức thư có mục đích chủ yếu để gửi tệp đính kèm hoặc để thông báo người nhận đã nhận được thư… cũng được loại bỏ. Những bức thư này được xem là không có đủ thông tin để đánh giá về mức độ quan trọng. Hơn nữa, bởi vì nội dung ngắn, những bức thư này không làm mất nhiều thời gian của người dùng. Một trường hợp ngoại lệ là khi toàn bộ những nội dung không mong muốn đều được kẻ phát tán thư rác đưa vào tệp đính kèm hoặc trong hình ảnh. Một phương pháp lọc thư rác dựa trên nội dung thông thường sẽ không phát hiện được những bức thư như vậy. Khi đó, bộ lọc thư rác cần được tích hợp thêm những kỹ thuật nhận dạng văn bản từ hình ảnh và kỹ thuật đọc nội dung từ các loại tập tin văn bản khác nhau. Trong tập dữ liệu, không có bức thư rác nào chứa nội dung văn bản dưới dạng hình ảnh được tìm thấy.

1.4.4.4. Gán nhãn cho tập dữ liệu

Sau các bước xử lý ở trên, tập dữ liệu còn lại 12,118 bức thư tiếng Việt. Toàn bộ thư trong tập dữ liệu là thư có nội dung chính bằng tiếng Việt và được người dùng xác nhận

[11] https://github.com/first20hours/google-10000-english
[12] https://github.com/thanhphu/tudien/blob/master/tudien.txt; https://github.com/undertheseanlp/dictionary

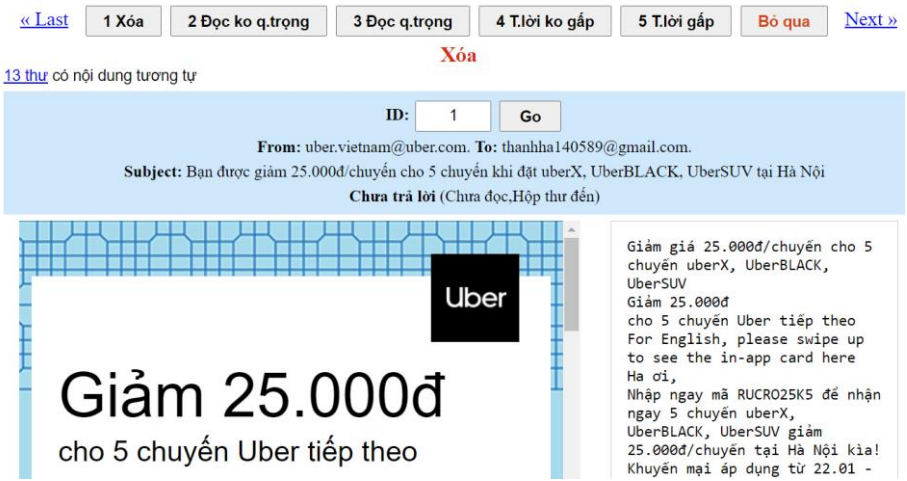
là có đủ thông tin trong tiêu đề và nội dung để đánh giá mức độ quan trọng của bức thư. Tập dữ liệu được tiến hành gán nhãn thủ công bởi 07 tình nguyện viên đã cung cấp dữ liệu thư điện tử cho nghiên cứu. Trước tiên, tập dữ liệu được gán nhãn cho bài toán lọc thư rác với 02 lớp: *thư rác*, *thư hợp lệ*. Những tình nguyện viên được hướng dẫn gán nhãn *thư rác* cho một bức thư nếu người dùng cho rằng (1) bức thư không có giá trị đối với người dùng trong quá khứ, hiện tại và tương lai hoặc (2) người dùng bị làm phiền bởi bức thư đó. Kết thúc quá trình gán nhãn cho bài toán lọc thư rác, có 2,527 bức thư được gán nhãn *thư rác* và 9,591 bức thư được gán nhãn *thư hợp lệ*.

Sau khi gán nhãn tập dữ liệu dành cho bài toán lọc thư rác, luận án tiếp tục phát triển tập dữ liệu để sử dụng cho bài toán dự đoán hành động người dùng. Có thể dễ dàng nhận thấy thư rác chính là những bức thư mà người dùng nên xóa mà không cần đọc. Vì vậy, những bức thư được gán nhãn *thư rác* ở tập dữ liệu lọc thư rác sẽ được tự động gán nhãn *xóa* để sử dụng trong bài toán dự đoán hành động người dùng. Tiếp theo, những bức thư được gán nhãn *thư hợp lệ* được tiến hành gán một trong hai nhãn *đọc* hoặc *trả lời*. Những tình nguyện viên được hướng dẫn gán nhãn dữ liệu của họ theo quyết định cá nhân về hành động mà họ mong muốn thực hiện đối với mỗi bức thư.

Những bức thư hợp lệ được tiến hành gán nhãn để thu được 7,966 bức thư cần *đọc* và 1,625 bức thư cần *trả lời*. Những bức thư thuộc hành động *đọc* chiếm tỷ lệ lớn nhất trong tập dữ liệu. Điều đó cho thấy phân bố của dữ liệu thư điện tử tập trung nhiều ở các bức thư có mức độ quan trọng trung bình và trên thực tế những bức thư quan trọng, cần sự quan tâm xử lý của người dùng, có số lượng nhỏ. Tỷ lệ phân bổ này làm rõ thêm cho lợi ích của phương pháp xếp hạng thư điện tử. Nếu hòm thư không được sắp xếp một cách tự động, người dùng sẽ cần phải bỏ ra nhiều thời gian để đọc những bức thư không quan trọng trước khi tìm thấy những bức thư cần được xử lý.

Tiếp theo, tập dữ liệu được phát triển để có thể sử dụng cho bài toán xếp hạng thư điện tử với 05 mức độ ưu tiên. Theo đó, 03 nhãn dữ liệu có sẵn sẽ được tiếp tục chia thành các mức độ ưu tiên nhỏ hơn. Mức độ *xóa* được coi là mức độ 1. Mức độ *đọc* được chia thành hai mức độ nhỏ là mức độ 2, *đọc không quan trọng*, và mức độ 3, *đọc quan trọng*. Mức độ 2 được định nghĩa là những bức thư có chứa thông tin không quan trọng mà người dùng có thể đọc khi rảnh rỗi hoặc có thể bỏ qua nếu không có thời gian đọc.

Mức độ 3 được định nghĩa là những bức thư có chứa thông tin mà người dùng cần đọc nhưng không cần trả lời. Nếu bỏ qua và không đọc các bức thư thuộc mức độ 3, người dùng có thể bị gián đoạn thông tin trong công việc hoặc các lĩnh vực khác của cuộc sống. Mức độ *trả lời* được chia thành hai mức độ nhỏ là mức độ 4, *trả lời không gấp* và mức độ 5, *trả lời gấp*.



Hình 1.8: Công cụ gán nhãn thư với chức năng phát hiện thư tương tự.

Như đã đề cập về tính quan trọng và tính khẩn cấp của thông tin, trong nghiên cứu này, ưu tiên được dành cho tính cấp thiết. Vì vậy, định nghĩa về tầm quan trọng của các bức thư cần trả lời phần lớn được quyết định bởi tính khẩn cấp. Những bức thư thuộc mức độ 4 được định nghĩa là những bức thư quan trọng cần đọc và cần được trả lời nhưng người gửi không nhấn mạnh về tính khẩn cấp. Những bức thư thuộc mức độ 5 được định nghĩa là những bức thư quan trọng cần đọc và cần được trả lời trong thời gian ngắn. Điều đó có nghĩa rằng người dùng cần phải đọc và xử lý những bức thư này trước những bức thư thuộc các mức độ khác.

1.4.4.5. Bảo mật thông tin cá nhân

Một trong những khó khăn trong việc xây dựng tập dữ liệu về thư điện tử đến từ vấn đề quyền riêng tư của người sử dụng. Bởi vì nội dung thư điện tử chứa thông tin quan trọng về công việc cũng như thông tin cá nhân nhạy cảm nên người dùng không mong muốn công khai dữ liệu của họ. Tập dữ liệu Enron [19] được công khai sau khi tập đoàn Enron giải thể và bị nhà chức trách tiến hành điều tra. Đây là một trong số ít những trường hợp đặc biệt mà dữ liệu thư điện tử liên quan đến công việc được công bố. Tập

dữ liệu TREC được đóng góp bởi các tình nguyện viên và thông qua quy trình xử lý thủ công để loại bỏ những thông tin nhạy cảm [23] trước khi được công bố.

Các tình nguyện viên cung cấp dữ liệu thư điện tử không mong muốn nội dung thư của họ bị tiết lộ cho một bên thứ ba. Đây cũng là một trong những lý do để giải thích cho việc không có tập dữ liệu thư điện tử tiếng Việt được công bố trên mạng Internet. Trong quá trình xây dựng tập dữ liệu để sử dụng cho các nghiên cứu trong luận án, tổng số thư thu được từ các hòm thư lên tới 37,003 bức thư. Với số lượng không nhỏ, việc xác định và loại bỏ những thông tin cá nhân nhạy cảm tốn nhiều thời gian và công sức. Vì vậy, thay vào đó, tác giả đã thực hiện nạc danh hóa dữ liệu nhằm bảo vệ thông tin cá nhân của các tình nguyện viên. Mục tiêu của công đoạn này được xác định là biến đổi tập dữ liệu thành một dạng khác để lược bỏ ngữ nghĩa trong khi vẫn giữ nguyên những thông tin đầu vào cần thiết cho các mô hình học máy.

Bản thân các mô hình học máy được áp dụng trong luận án không có khả năng hiểu ngữ nghĩa của các từ ngữ đơn lẻ trong nội dung thư điện tử. Khi biểu diễn văn bản bằng phương pháp *word2vec* [56] hoặc phân loại văn bản bằng mạng nơ-ron hồi quy như LSTM [4], các mô hình học máy đã dựa vào vị trí tương đối của các từ ngữ trong văn bản để mô phỏng khả năng hiểu ý nghĩa văn bản. Do đó, kết quả biểu diễn văn bản bằng *word2vec* và hoạt động của các mô hình học máy không bị ảnh hưởng khi ta thay thế các từ trong văn bản đầu vào thành các chuỗi thay thế. Gọi $A = \{t_i \mid i = 1, 2 \dots k\}$ là tập hợp các từ ngữ không trùng lặp trong tập dữ liệu gốc và $B = \{v_i \mid i = 1, 2 \dots k\}$ là tập các chuỗi không trùng lặp. Tập B được sinh ra để với mỗi từ t_i trong tập A, ta có một chuỗi v_i duy nhất để thay thế cho nó ($v_i \neq t_i$). Việc nạc danh hóa tập dữ liệu được hiểu là biến đổi tập dữ liệu gốc thành một tập dữ liệu nạc danh, cụ thể các từ t_i sẽ được thay thế bằng các chuỗi v_i. Phương pháp biến đổi dữ liệu cần thỏa mãn hai điều kiện: (1) mỗi chuỗi v_i là chuỗi duy nhất trong tập hợp B; (2) vị trí tương đối giữa các từ ngữ trong văn bản được bảo toàn. Ví dụ, văn bản "`ngày mai họp lúc 4h`" có thể thay thế bằng văn bản "`a b c d e`" trong tập dữ liệu nạc danh. Một văn bản khác, "`cuộc họp sẽ bắt đầu vào 4h`", sẽ được thay thế bằng "`f c g h i k e`". Trong ví dụ trên, từ "`họp`" trong tập A được thay thế bởi chuỗi "`c`" trong tập B và từ "`4h`" trong tập A được thay thế bởi chuỗi "`e`" trong tập B. Sự biến đổi này xảy ra nhất quán trên tất cả các bức thư. Trong phương

pháp này, các chuỗi "a", "b", "c"… là các chuỗi được sinh ra ngẫu nhiên và đảm bảo điều kiện các chuỗi trong tập B không bị trùng lặp.

Phương pháp nạc danh hóa tập dữ liệu này có thể đáp ứng được yêu cầu của các thí nghiệm ứng dụng mô hình học máy. Tuy nhiên, tập dữ liệu nạc danh sẽ không thể được sử dụng bởi các nghiên cứu về xử lý ngôn ngữ tự nhiên, chẳng hạn như trong bài toán xác định từ loại, phát hiện ngôn ngữ văn bản.

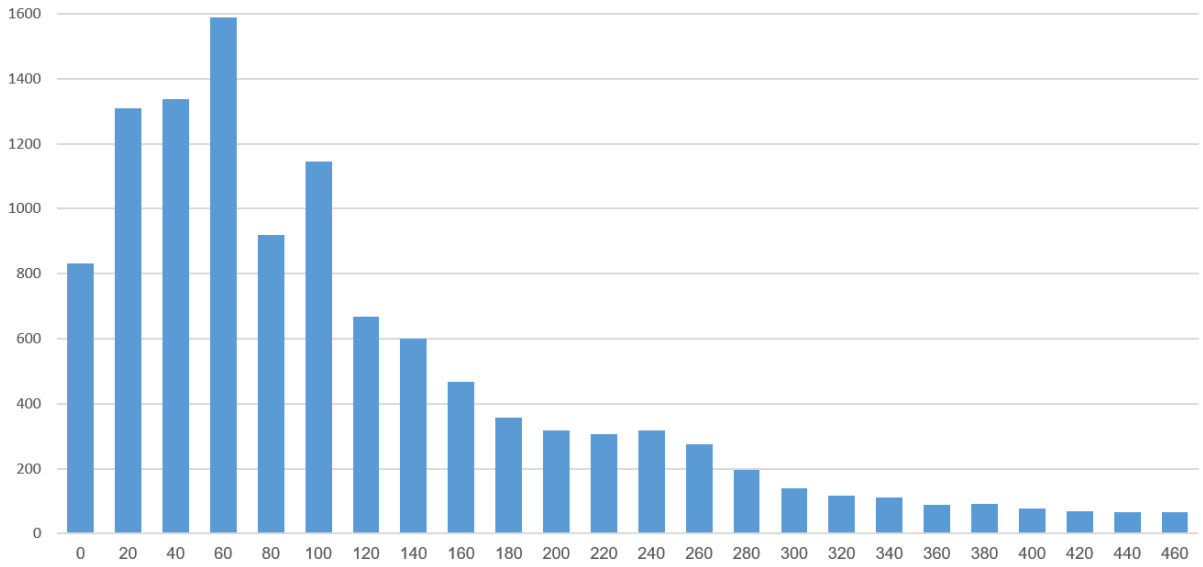
1.4.4.6. Các thông số của tập dữ liệu

Tập dữ liệu bao gồm 12,118 bức thư, trong đó có 2,527 bức thư cần *xóa*, 7,966 bức thư cần *đọc* và 1,625 bức thư cần *trả lời*. Trong quá trình xây dựng tập dữ liệu, các tác vụ (i) loại bỏ các bức thư có nội dung quá ngắn và có ngôn ngữ không phải là tiếng Việt, (ii) loại bỏ các bức thư có nội dung trùng lặp và (iii) gán nhãn được thực hiện song song với nhau để tiết kiệm thời gian cho các tình nguyện viên. Hình 1.8 là giao diện của công cụ gán nhãn đã được sử dụng. Phương pháp hỗ trợ tìm kiếm những bức thư có nội dung trùng lặp đã được trình bày trong Chương 2 của luận án. Sau khi tiến hành gán lại nhãn theo phương án đã đề cập, số thư thuộc các mức độ ưu tiên được mô tả trong Bảng 1.4. Những bức thư thuộc mức độ 3 chiếm số lượng lớn nhất trong tập dữ liệu. Điều đó cho thấy phân bố của dữ liệu tập trung nhiều nhất ở các bức thư có mức độ quan trọng trung bình và trên thực tế những bức thư quan trọng và khẩn cấp chiếm tỷ lệ nhỏ.

Luận án nhận thấy sự phân bổ thư rác và thư hợp lệ trong tập dữ liệu chưa phản ánh đúng tỷ lệ thư rác trên thực tế bởi vì hai lý do sau đây. Thứ nhất, tập dữ liệu được thu thập từ hòm thư của người dùng trên hệ thống Gmail. Một số lượng thư rác đã bị loại bỏ trên đường truyền tải thư trước khi đến được hòm thư của người dùng bởi những bộ lọc thư rác dựa trên danh sách, xác thực người gửi, bộ lọc thư rác của Gmail… Thứ hai, một phần thư rác đến được hòm thư của người dùng có thể đã bị xóa đi trong suốt quãng thời gian sử dụng thư điện tử của họ.

Bảng 1.2 thống kê độ dài thư của tập dữ liệu. Độ dài thư là tổng số từ của cả tiêu đề và nội dung thư. Trong đó, một từ tiếng Việt có thể là từ đơn hoặt từ ghép có chứa nhiều âm tiết. Từ dữ liệu thống kê này có thể nhận thấy độ dài thư không thể hiện một xu hướng cụ thể nào giữa các mức độ ưu tiên. 5% các bức thư ngắn nhất có độ dài nhỏ hơn

hoặc bằng 16 từ. 5% các bức thư dài nhất có độ dài lớn hơn hoặc bằng 478 từ. Như vậy, 90% thư có độ dài từ 17 tới 477 từ.



Hình 1.9: Phân bổ độ dài thư của tập dữ liệu thư điện tử tiếng Việt.

Hình 1.9 thể hiện phân bổ về độ dài của các bức thư trong tập dữ liệu. Mặc dù độ dài tối đa của một bức thư là 2,160 từ, những bức thư ngắn chiếm số lượng lớn nhất, trong đó độ dài từ 60 đến 80 từ là phổ biến nhất.

Bảng 1.2: Thống kê độ dài thư của tập dữ liệu thư điện tử tiếng Việt.

Mức độ ưu tiên	**Độ dài bức thư**		
	Nhỏ nhất	*Trung bình*	*Lớn nhất*
1	6	152.9	1212
2	2	120.4	1113
3	2	163.1	2160
4	3	222.6	2153
5	7	173.1	2049
Tất cả	2	158.4	2160

Bảng 1.3 thống kê về số lượng người gửi thư trong tập dữ liệu, tính cả những bức thư mà địa chỉ người nhận là các danh sách gửi thư mà các tình nguyện viên tham gia. Có tổng số 1,345 địa chỉ gửi thư trong tập dữ liệu. Bảng 1.3 cũng thống kê số lượng người gửi theo từng mức độ ưu tiên. Trong đó có thể thấy rõ mức độ ưu tiên trung bình (mức độ 3) có số lượng địa chỉ gửi thư nhiều nhất. Theo cả hai chiều tăng hoặc giảm mức độ quan trọng, số lượng người gửi đều có xu hướng giảm dần. Một địa chỉ có thể gửi nhiều

thư thuộc về các mức độ ưu tiên khác nhau nên tổng số lượng địa chỉ gửi thư của các mức độ ưu tiên không bằng với tổng số lượng địa chỉ gửi thư.

Bảng 1.3: Thống kê về người gửi thư của tập dữ liệu thư điện tử tiếng Việt.

Mức độ ưu tiên	**Số lượng địa chỉ gửi thư**	**Số lượng thư / người gửi**
1	167	15.13
2	308	7.00
3	865	5.81
4	213	3.09
5	178	2.78
Tất cả	1345	8.07

Số lượng thư trung bình đến từ một người gửi là 8.07 bức thư. Số lượng thư trung bình trên người gửi thư có xu hướng giảm dần khi mức độ ưu tiên tăng lên. Những địa chỉ gửi thư tới nhiều nhất là thư tự động của các nền tảng, ứng dụng ví dụ như Grab, Twitter, Lazada, Vietcombank…

Bảng 1.4: Phân bổ thư theo nhãn của tập dữ liệu thư điện tử tiếng Việt.

Mức độ ưu tiên	**Số lượng thư**
1 – Thư cần xóa	2,527
2 – Đọc không quan trọng	2,179
3 – Đọc quan trọng	5,787
4 – Trả lời không gấp	970
5 – Trả lời gấp	655
Tổng số	12,118

Tập dữ liệu được mô tả trong phần này được sử dụng trong các thí nghiệm đánh giá và so sánh trong các đề xuất của luận án. Tập dữ liệu đã được công bố công khai[13] trên mạng Internet.

1.5. KẾT LUẬN CHƯƠNG 1

Chương này đã trình bày giới thiệu tổng quan về thư điện tử và bài toán xác định thứ tự ưu tiên của thư điện tử. Luận án đã giới thiệu các khái niệm chung về thư điện tử, vấn đề quá tải thư điện tử, ý nghĩa của bài toán xác định thứ tự ưu tiên của thư điện tử và các dạng của bài toán là lọc thư rác, dự đoán hành động người dùng và xếp hạng thư

[13] https://github.com/VnEmailPrioritization/dataset2021

điện tử. Các cách tiếp cận để giải quyết các bài toán nói trên cũng được khảo sát và phân tích để chỉ ra những thành tựu đã đạt được cùng các vấn đề chưa được giải quyết.

Thông qua tổng quan tài liệu, nhiều điểm hạn chế của các phương pháp xác định thứ tự ưu tiên của thư điện tử đã được nêu ra. Tuy không thể giải quyết tất cả các vấn đề đã nêu ra, luận án đã giới hạn phạm vi nghiên cứu là các mô hình xác định thứ tự ưu tiên của thư điện tử dựa trên phương pháp phân loại. Cụ thể là:

- Thu thập và xây dựng tập dữ liệu thư điện tử tiếng Việt. Tập dữ liệu được xây dựng với mục tiêu để sử dụng cho cả ba dạng của bài toán xác định thứ tự ưu tiên của thư điện tử. Quy trình xây dựng tập dữ liệu đã được trình bày cụ thể trong mục 1.4.4 thuộc Chương 1. Tập dữ liệu này sẽ được sử dụng cho thí nghiệm trong các đề xuất của Chương 2, Chương 3 và Chương 4.
- Nghiên cứu phương pháp lọc thư rác hiệu quả dành cho thư điện tử tiếng Việt. Với bài toán này, mục tiêu đặt ra cụ thể là tăng độ chính xác của kết quả phát hiện thư rác dựa trên hệ thống SpamAssassin để phát huy được những ưu điểm của hệ thống này. Phương pháp đề xuất được trình bày chi tiết trong Chương 2 của luận án.
- Nghiên cứu phương pháp dự đoán hành động người dùng hiệu quả dành cho thư điện tử tiếng Việt. Chương 3 của luận án sẽ thảo luận chi tiết về các hướng giải quyết bài toán này.
- Nghiên cứu phương pháp xếp hạng thư điện tử với 5 mức độ ưu tiên có độ chính xác cao dành cho thư điện tử tiếng Việt. Phương pháp cụ thể sẽ được trình bày trong Chương 4 của luận án.

CHƯƠNG 2: PHÁT HIỆN THƯ RÁC

2.1. MỞ ĐẦU

2.1.1. Đặc điểm của thư rác

Để đưa ra hướng tiếp cận phù hợp cho bài toán lọc thư rác, ta cần tìm hiểu kỹ những đặc điểm của thư rác. Thư rác được gửi nhằm thực hiện một số mục đích của kẻ phát tán thư rác, ví dụ như quảng cáo, đính kèm virus, thậm chí là lừa đảo. Thông thường người gửi không biết người nhận là ai và ngược lại. Đứng trên quan điểm của người gửi, đó là một hình thức gửi thư theo số lượng lớn (nên gọi là "bulk email"). Danh sách địa chỉ người nhận thường được thu thập từ các diễn đàn, các danh sách thư, mạng xã hội… Cũng có nhiều công ty mà công việc kinh doanh chính của họ là gửi thư rác theo đơn đặt hàng của những công ty khác. Chính vì vậy, hầu hết thư rác hiện nay thường có nội dung quảng cáo thương mại và dịch vụ. Tuy nhiên có nhiều thư trong số đó có chứa những đường dẫn ngụy trang, tức là link có vẻ ngoài là đường dẫn tới một trang web quen thuộc với người dùng, nhưng thực chất là dẫn đến một trang web giả mạo, trang web có chứa mã độc, nhằm lừa hoặc ăn cắp thông tin của người dùng. Những bức thư rác như vậy được gộp vào nhóm thư rác lừa đảo. Thư rác cũng có thể chứa tệp đính kèm là mã độc hoặc virus máy tính, nhằm phát tán virus hoặc đánh cắp thông tin cá nhân. Ngoài ra còn một lượng thư rác có tính chất quấy nhiễu, chứa những nội dung không lành mạnh (khiêu dâm, chống phá chính trị…).

Thư rác được gửi đi một cách tự động, bất chấp mọi cấm đoán hoặc ngăn cản [36]. Mục đích của những kẻ gửi thư rác là có thể phát tán lượng thư rác tới người dùng càng nhiều càng tốt. Mục đích của những kẻ phát tán thư rác có thể là để quảng bá thông tin, cũng có thể là để phá vỡ và gây cản trở công việc của người nhận với số lượng thư rác lớn và những thông tin sai lệch. Một số trường hợp, mục tiêu gây thiệt hại của thư rác không phải là người dùng mà là các nhà cung cấp thư điện tử. Do vậy, những kẻ phát tán thường viết ra các phần mềm tự động gửi một lượng lớn thư rác trong một khoảng thời gian ngắn.

Địa chỉ người gửi trên thư rác thường là những địa chỉ giả mạo. Phiên bản SMTP năm 2008 được sử dụng hiện nay vẫn giữ bản chất là một giao thức đơn giản. Nó không xác minh người gửi nên việc giả mạo địa chỉ người gửi là khá dễ dàng. Để tránh sự nghi

ngờ của người nhận, kẻ phát tán thường lấy địa chỉ thư điện tử của một người dùng bình thường trong một máy chủ thư điện tử nào đó. Việc sử dụng một hoặc nhiều địa chỉ thư điện tử ảo để gửi thư rác cũng rất phổ biến.

Sự trao đổi thư giữa người gửi và người nhận thư rác thường chỉ diễn ra một chiều. Trong hầu hết mọi trường hợp, người dùng thư điện tử sẽ không trả lời một bức thư rác quảng cáo bởi vì theo định nghĩa, thư rác là thư không mong muốn. Trường hợp ngoại lệ khi người dùng trả lời thư rác đó là đối với những bức thư rác dạng lừa đảo.

Thư rác được gửi đến những địa chỉ ngẫu nhiên trên một diện rộng. Từ đó, nếu nhìn vào đồ thị mạng xã hội của người gửi và người nhận, những địa chỉ gửi thư rác là những địa chỉ có nhiều mối liên kết với những địa chỉ thư điện tử khác, trong đó thư gửi đi chiếm tỷ lệ lớn, thư nhận về có tỷ lệ nhỏ. Tuy nhiên, những địa chỉ thư điện tử dùng để phát tán thư một cách hợp lệ đến các danh sách gửi thư (cơ quan, tổ chức, khách hàng đăng ký...) cũng có mối liên kết trên đồ thị mạng xã hội tương tự. Bởi vậy, khi sử dụng đồ thị mạng xã hội làm cơ sở để phát hiện địa chỉ gửi thư rác, ta cần phải có cơ chế danh sách trắng cho những trường hợp ngoại lệ.

Hình 2.1: Một bức thư rác lừa đảo
có nội dung hứa hẹn một số tiền lớn cho người nhận.

Nội dung của thư rác là những nội dung không mong muốn và gây phiền hà cho người nhận. Phần lớn nội dung của thư rác là những thông tin mời chào về thương mại,

quảng cáo sản phẩm. Bên cạnh đó, phải kể đến những thư rác có nội dung xấu (như khiêu dâm, chống phá chính trị, lừa đảo...) gây ảnh hưởng không tốt đến xã hội và đời sống.

Lượng thư rác phát tán virus/mã độc cũng không nhỏ. Một số thư rác thường được gắn kèm những chương trình virus nguy hiểm có khả năng ăn cắp thông tin cá nhân hoặc làm hỏng dữ liệu, phần mềm lưu trên máy tính của người dùng, gián tiếp tổn hại đến kinh tế của các cá nhân, tổ chức và doanh nghiệp. Hiện nay, những bức thư rác với nội dung hứa hẹn mang đến một khoản tiền lớn cho người đọc thư rác đã tăng nhanh (Hình 2.1). Những người dùng kém hiểu biết, cả tin thường bị lừa bởi hình thức này. Có một loại hình thư rác cũng rất nguy hại đó là mạo danh ngân hàng, hay công ty nhằm lừa gạt người nhận để lấy mật khẩu, tài khoản, số thẻ tín dụng… Một kiểu nội dung nữa thường thấy ở thư rác đó là nội dung văn bản được hiển thị dưới dạng một bức ảnh và được nhúng vào bức thư. Đây là cách mà kẻ phát tán dùng để qua mặt các hệ thống lọc thư rác.

2.1.2. Những vấn đề còn tồn tại

Bộ lọc thư rác theo luật SpamAssassin đang được sử dụng phổ biến trên các máy chủ thư điện tử, có tốc độ xử lý nhanh, đáp ứng tốt yêu cầu xử lý một lượng thư lớn trong thời gian thực. Tuy vậy, bộ lọc SpamAssassin chưa có sẵn một hệ thống tự động sinh tập luật, dẫn đến hạn chế về hiệu quả phát hiện bởi vì nội dung thư rác luôn luôn biến đổi. Để khắc phục hạn chế nói trên, các phương pháp tự động sinh tập dựa trên mô hình học máy [17, 28, 62] dành cho SpamAssassin cũng đã được đề xuất và áp dụng trên thực tế. Mục tiêu chính trong các nghiên cứu về lọc thư rác trên SpamAssassin là xây dựng được tập luật có hiệu quả cao. Trong chương này, luận án sẽ trình bày hai vấn đề tồn tại của các phương pháp đã đề xuất, từ đó đưa ra các đề xuất để nâng cao hiệu quả của tập luật được sinh ra.

Vấn đề thứ nhất còn tồn tại trong bài toán lọc thư rác trên nền tảng SpamAssassin nằm ở khâu lựa chọn đặc trưng trong quy trình tự động xây dựng tập luật lọc thư rác. Việc nâng cao hiệu quả phát hiện của bộ lọc thư rác luôn là một yêu cầu cấp thiết khi nội dung thư rác ngày càng đa dạng hơn và các kỹ thuật phát tán thư rác cũng ngày càng được cải thiện. Học máy là phương pháp được áp dụng phổ biến nhất để xây dựng bộ

lọc thư rác nói chung và tập luật lọc thư rác trên SpamAssassin nói riêng. Tập đặc trưng quyết định không gian tìm kiếm của quá trình huấn luyện mô hình học máy và là yếu tố quan trọng ảnh hưởng tới hiệu quả của một mô hình phân loại. Vì vậy, một cách để nâng cao hiệu quả phát hiện thư rác của những mô hình được xây dựng bằng phương pháp học máy đó là nâng cao chất lượng của tập đặc trưng. Các phương pháp xây dựng tập luật SpamAssassin được công bố đến thời điểm hiện tại đều thực hiện quy trình xây dựng mô hình học máy bao gồm tiền xử lý dữ liệu, trích chọn đặc trưng, huấn luyện mô hình và thử nghiệm mô hình. Trong đó, khâu lựa chọn đặc trưng thường được thực hiện dựa trên phân tích dữ liệu kết hợp với kinh nghiệm của chuyên gia. Cách làm này được gọi chung là gia công đặc trưng thủ công. Khi làm theo cách nói trên, khâu lựa chọn đặc trưng cần phải được điều chỉnh khi đặc tính của dữ liệu thay đổi. Thêm vào đó, không gian tìm kiếm của bước huấn luyện mô hình bị giới hạn bởi tập đặc trưng đã được chọn. Vì vậy, kết quả tốt nhất của quá trình huấn luyện là một phương án tối ưu cục bộ, tức là tập luật tối ưu trong số các tập luật có cùng tập đặc trưng, chứ không phải là phương án tối ưu toàn cục, tức là tập luật tối ưu trong số mọi tập luật có thể được tìm thấy.

Thứ hai, một vấn đề khi điều chỉnh mô hình trước khi đưa vào sử dụng trong thực tế đó là khi mục tiêu tối ưu là để tăng chỉ số *recall* (1.14) thì kéo theo tỷ lệ chặn nhầm thư hợp lệ *FAR* (2.2) cũng tăng cao. Ngược lại, để giảm tỷ lệ *FAR* thì chỉ số *recall* cũng đồng thời bị giảm. Bởi vì một bộ lọc không thể đồng thời đạt được giá trị tối ưu đối với cả hai tiêu chí đánh giá, bài toán đặt ra là cần phải điều chỉnh bộ lọc sao cho cân bằng được giữa *recall* và *FAR* để đem lại lợi ích lớn đối với mục đích sử dụng cụ thể.

$$FAR = \frac{fp}{fp + tn} = 1 - precision \tag{2.2}$$

Một điểm cần được nhấn mạnh đó là mỗi người dùng có sự đánh giá khác nhau về độ nghiêm trọng của trường hợp lọc nhầm thư rác và bỏ sót thư rác. Vì vậy, các bộ lọc thư rác cần được điều chỉnh thường xuyên theo các tiêu chí khác nhau để phục vụ cho mục đích sử dụng của từng người dùng cụ thể.

Hai tiêu chí phổ biến để đánh giá bộ lọc thư rác là *recall* (1.14) và *FAR* (2.2). Tiêu chí *recall* phụ thuộc theo tỷ lệ nghịch với chỉ số *fn*, nhưng không bị ảnh hưởng bởi chỉ

số *fp*. Trong khi đó, tiêu chí *FAR* là tỷ lệ thuận với chỉ số *fp* và không bị ảnh hưởng bởi chỉ số *fn*. Khi hai mô hình có cùng giá trị *accuracy* (1.17), mô hình với giá trị *FAR* thấp hơn là mô hình có ít dự đoán dương tính nhưng tỷ lệ cảnh báo nhầm trong số các dự đoán thấp hơn. Mô hình có giá trị *recall* cao hơn sẽ có nhiều dự đoán dương tính và đồng thời tỷ lệ cảnh báo nhầm cũng cao hơn. Từ đó, để đạt được lợi ích tối đa từ một bộ lọc thư rác, hai tiêu chí nói trên cần được điều chỉnh ở mức cân bằng.

Gọi $recall_0$ và FAR_0 là các giá trị mong muốn của hai tham số *recall* và *FAR* của bộ lọc thư rác cần thiết kế (với ý nghĩa bộ lọc đạt yêu cầu là bộ lọc có $recall \geq recall_0$ và $FAR \leq FAR_0$). Trong quy trình xây dựng tập luật SpamAssassin thông thường, sau khi xác định ngưỡng *T*, điểm số của mỗi luật sẽ được tính toán để cho tham số *recall* của bộ lọc thu được là lớn nhất. Tuy nhiên các khả năng sau có thể xảy ra sau khi thực thi xong thuật toán tính điểm số: (a) Giá trị *recall* của bộ lọc thu được không đạt yêu cầu; (b) Giá trị *FAR* của bộ lọc thu được không đạt yêu cầu; (c) Giá trị *recall* và *FAR* đạt yêu cầu nhưng chưa phải là tối ưu. Để giải quyết vấn đề trên, người dùng phải thử chọn các giá trị ngưỡng *T* khác, thực hiện lại thuật toán tính điểm và tiếp tục kiểm tra xem các tham số *recall* và *FAR* của bộ lọc đã đạt yêu cầu chưa. Quy trình này không chỉ gây tốn thời gian, tốn tài nguyên hệ thống mà còn chưa giải quyết triệt để vấn đề (c) do giá trị *FAR* không tham gia vào quá trình tính điểm cho tập luật và chỉ có thể biết được sau khi huấn luyện hoàn tất. Phương pháp tối ưu hóa đa mục tiêu NSGA-II và DMEA-II đã được áp dụng để xây dựng tập luật SpamAssassin [50], tuy nhiên thí nghiệm cần được thực hiện trên những thuật toán khác, với những tập dữ liệu lớn hơn để thu được những kết quả phong phú hơn.

Đề xuất dành cho hai vấn đề nêu trên sẽ được trình bày và thảo luận trong chương này. Phần 2.2 mô tả quá trình xây dựng tập dữ liệu thư điện tử tiếng Việt để phục vụ nghiên cứu về lọc thư rác. Phần 2.3 đề xuất một phương pháp tự động sinh tập luật lọc thư rác cho SpamAssassin dựa trên mạng nơ-ron. Phần 2.4 trình bày ứng dụng của phương pháp tối ưu hóa đa mục tiêu để cân bằng độ chính xác và tỷ lệ lọc nhầm cho tập luật SpamAssassin. Phần 2.5 trình bày các thí nghiệm cùng kết quả thí nghiệm của các phương pháp được đề xuất.

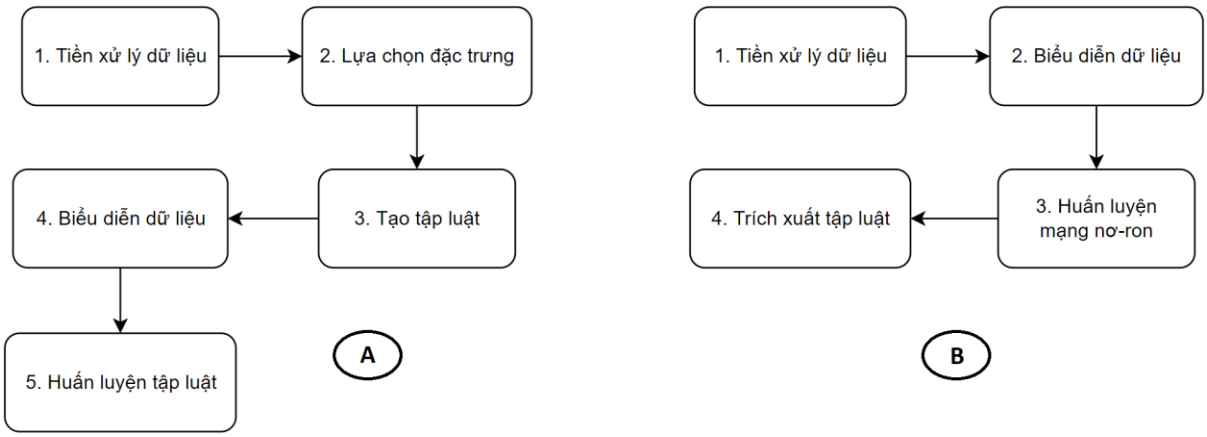
2.2. ỨNG DỤNG MẠNG NƠ-RON ĐỂ TỰ ĐỘNG LỰA CHỌN ĐẶC TRƯNG CHO BÀI TOÁN SINH TẬP LUẬT SPAMASSASSIN

Mục tiêu sinh tập luật lọc thư rác dành cho SpamAssassin được hiểu cụ thể là tìm một tập đặc trưng hữu hạn và gán điểm số cho tập đặc trưng đó để thu được một tập luật có trọng số. Để đánh giá hiệu quả của tập luật được xây dựng, ta thường dùng tập luật để phát hiện thư rác trên một tập dữ liệu thử nghiệm và tính kết quả dựa trên một tiêu chí đánh giá dành cho máy phân loại nhị phân. Quá trình huấn luyện tập luật là liên tục điều chỉnh để tìm ra tập luật có kết quả đánh giá tốt nhất. Giả sử tập luật cần tìm có tập đặc trưng là $R = \{r_1, r_2, \ldots, r_\beta\}$ có tập trọng số tương ứng là $W = \{w_1, w_2, \ldots, w_\beta\}$. Mục tiêu của bài toán là tìm ra tập R và W sao cho tập luật có hiệu quả dự đoán tốt nhất.

Trong mục này, luận án đề xuất một phương pháp nhằm giải quyết vấn đề tồn đọng thứ nhất trong bài toán lọc thư rác đã trình bày ở phần trước đó là tập đặc trưng chưa thực sự tốt do khâu lựa chọn đặc trưng và tối ưu trọng số được thực hiện tách rời. Cách làm này có mục tiêu cải thiện tập đặc trưng cho mô hình lọc thư rác dựa trên nền tảng SpamAssassin. Trong phạm vi của luận án, phương pháp được mô tả sau đây được đặt tên là phương pháp SD_1 để thuận tiện cho việc theo dõi.

2.2.1. Quy trình xây dựng tập luật SpamAssassin với mạng nơ-ron

Trong quy trình thông thường để sinh tập luật lọc thư rác cho SpamAssassin dựa trên phương pháp học máy (Hình 2.2a), hai bước lựa chọn đặc trưng và huấn luyện tập luật được thực hiện tách rời với nhau. Có nghĩa là, tập đặc trưng R được lựa chọn dựa trên một tiêu chí nào đó, sau đó tập trọng số W sẽ được điều chỉnh tối ưu dựa trên một tiêu chí khác. Một khi tập đặc trưng đã được chọn, nó không được cập nhật trong quá trình huấn luyện. Như vậy, hiệu quả dự đoán đạt tốt nhất được chỉ là phương án tối ưu cục bộ bị giới hạn bởi tập đặc trưng R. Cách làm này chưa khai thác được sự điều chỉnh từ việc huấn luyện với dữ liệu để củng cố chất lượng của tập đặc trưng. Thêm nữa, số lượng luật (đặc trưng) cũng là một yếu tố có thể ảnh hưởng tới hiệu quả của bộ lọc. Thông thường, việc tăng số lượng luật sẽ dẫn đến kết quả thử nghiệm tốt hơn. Trong khi đó, bộ lọc có số lượng luật nhỏ thường có tính khái quát cao hơn [35].



Hình 2.2: (a) Quy trình tự động sinh tập luật SpamAssassin truyền thống;
(b) Quy trình tự động sinh tập luật SpamAssassin dựa trên mạng nơ-ron

Trong những năm gần đây, việc huấn luyện mạng nơ-ron đã trở nên dễ dàng hơn nhờ vào các kỹ thuật tối ưu hóa và hàm kích hoạt mới. Kỹ thuật lan truyền ngược (back-propagation) dùng để tính đạo hàm thành phần cho phép huấn luyện mạng nơ-ron có nhiều lớp ẩn. Bài toán sinh tập luật SpamAssassin như đã trình bày ở các phần trước vốn là bài toán xây dựng mô hình mạng nơ-ron một lớp *perceptron*. Trong phần này, tác giả luận án trình bày mô hình mạng nơ-ron có nhiều lớp ẩn nhằm mục tiêu kết hợp việc trích chọn đặc trưng và tối ưu tập luật trên cùng một cấu trúc mạng. Cách tiếp cận này hướng tới giải quyết đồng thời các vấn đề chủ yếu sau:

– Lựa chọn tập đặc trưng tốt nhất.
– Gán tập điểm số tốt nhất cho các luật trong tập luật.
– Giới hạn số lượng luật nhằm đảm bảo hiệu năng xử lý cho tập luật cần xây dựng.

Trong quy trình tự động sinh tập luật SpamAssassin của phương pháp đề xuất (Hình 2.2b), việc lựa chọn đặc trưng không còn là một khâu tách rời mà đã được tích hợp vào trong quá trình huấn luyện mô hình mạng nơ-ron. Như vậy, mục tiêu khi huấn luyện mô hình này được mở rộng so với mục tiêu khi gán điểm số cho tập luật SpamAssassin theo cách làm truyền thống. Thay vì đi tìm tập trọng số cho kết quả tốt nhất dành cho một tập đặc trưng duy nhất, ta đi tìm tập đặc trưng và các trọng số tương ứng có kết quả tốt nhất trong số mọi tập đặc trưng thỏa mãn điều kiện về số lượng luật. Nếu ta coi việc gán điểm số truyền thống là đi tìm phương án tối ưu cục bộ trên một tập đặc trưng cụ thể thì phương án đề xuất là việc đi tìm phương án tối ưu toàn cục trên mọi

tập đặc trưng tiềm năng. Để thực hiện được mục tiêu này, ta cần tạo ra cơ chế để thay đổi không gian tìm kiếm từ tập đặc trưng này sang tập đặc trưng khác. Việc thêm, bớt đặc trưng hoặc thay thế đặc trưng đều có tác dụng biến đổi tập đặc trưng, dẫn tới không gian tìm kiếm không còn bị giới hạn trong chỉ một tập đặc trưng.

Trong phương pháp này, chất lượng của các đặc trưng được thể hiện bằng độ lớn tuyệt đối của trọng số của chúng. Giá trị này có ý nghĩa giúp tìm ra những đặc trưng có tính đại diện cao cho một loại thư (thư rác, thư hợp lệ). Những đặc trưng có trọng số lớn là đặc trưng có sức ảnh hưởng lớn đến kết quả dự đoán. Trong một kích bản huấn luyện lý tưởng, những đặc trưng không có ý nghĩa sẽ có trọng số tiến dần về giá trị 0. Một ngưỡng có thể tự động điều chỉnh ε được dùng để giới hạn số lượng luật nhằm đảm bảo có α luật được sinh ra.

2.2.2. Tiền xử lý và biểu diễn dữ liệu

Từ tập dữ liệu huấn luyện bao gồm thư rác và thư hợp lệ, công cụ tách từ tiếng Việt vnTokenizer [34] được sử dụng để tách nội dung và tiêu đề các bức thư thành những từ có nghĩa. Sau khi loại bỏ các từ trùng lặp, ta có được tập từ ngữ $\mathbf{V_s}$ từ tiêu đề thư và tập từ ngữ $\mathbf{V_b}$ từ nội dung thư. Việc loại bỏ các từ chức năng (stop words) không được thực hiện ở bước này bởi vì vai trò loại bỏ các đặc trưng không có ý nghĩa thuộc về khâu huấn luyện mô hình. Những từ ngữ không có giá trị phân loại giữa thư rác và thư hợp lệ sẽ được loại bỏ trong quá trình đó.

Mỗi bức thư được biểu diễn theo phương pháp túi từ dưới dạng một vector nhị phân để mô phỏng cơ chế phát hiện thư rác của SpamAssasssin. Mỗi phần tử của vector đại diện cho một từ trong tập từ vựng, với giá trị 1 thể hiện sự có mặt của từ trong bức thư và giá trị 0 nếu ngược lại. Tần số xuất hiện của từ không được giữ lại trong cách biểu diễn văn bản này nên những đặc trưng xuất hiện nhiều hơn một lần trong văn bản cũng sẽ có giá trị là 1. Một đặc trưng được đặc tả bởi hai thuộc tính là bản thân từ ngữ và nơi mà từ ngữ đó xuất hiện. Cho nên, một từ ngữ nằm trong tiêu đề thư và chính từ đó nằm trong nội dung thư là hai đặc trưng phân biệt. Chính vì vậy, số lượng đặc trưng cần thiết để biểu diễn toàn bộ thư trong tập dữ liệu huấn luyện, cũng chính là độ dài vector đầu vào $\boldsymbol{x}$ của mô hình, là $|\boldsymbol{x}| = |\mathbf{V_s}| + |\mathbf{V_b}|$.

2.2.3. Mô hình mạng nơ-ron

Có hai mục tiêu cần đạt được khi thiết kế mô hình mạng nơ-ron trong phương pháp đề xuất. Thứ nhất, sau khi mạng nơ-ron hoàn thành huấn luyện thì ta phải có được những tham số cần thiết để sinh ra được tập luật SpamAssassin. Vì vậy, mạng nơ-ron phải có một bộ phận mô phỏng được cơ chế phát hiện thư rác của SpamAssassin. Mục tiêu thứ hai là phải lựa chọn được những đặc trưng giúp cho kết quả phân loại thư rác chính xác nhất. Để làm được điều này, mạng nơ-ron cần phải chứa các tham số thể hiện chất lượng của mỗi đặc trưng và phải điều chỉnh các tham số này dựa trên kết quả dự đoán của mô hình đối với mỗi bức thư. Để thực hiện các mục tiêu nói trên, luận án đề xuất mô hình mạng nơ-ron theo mô tả dưới đây.

Mạng nơ-ron được đề xuất để sinh tập luật SpamAssassin gồm hai thành phần chính. Thành phần thứ nhất là một lớp mạng có nhiệm vụ lựa chọn đặc trưng, gọi là lớp *FS*. Thành phần thứ hai là một mạng perceptron có nhiệm vụ phân loại thư rác, gồm một lớp ẩn được đặt tên là lớp *P*. Trong phần mô tả sau đây, thuật ngữ *thuộc tính* được dùng để chỉ một từ ngữ bất kỳ được trích xuất từ tập dữ liệu; thuật ngữ *đặc trưng* được dùng để chỉ một từ ngữ được lựa chọn để đưa vào tập luật SpamAssassin.

Bức thư đầu vào được biểu diễn dưới dạng vector nhị phân $\boldsymbol{x}$ theo như mô tả ở phần trước. Lớp *FS* là một lớp mạng truyền thẳng có số lượng trọng số bằng với kích thước của vector đầu vào $\boldsymbol{x}$. Ta gọi tập trọng số của lớp *FS* là ω, khi đó $|\omega| = |\boldsymbol{x}|$. Mỗi trọng số trong tập ω có vai trò thể hiện mức độ quan trọng của một thuộc tính đầu vào tương ứng với nó và quyết định thuộc tính đó có được lựa chọn làm đặc trưng hay không. Như vậy, tập ω được hiểu là tập trọng số của các thuộc tính. Hàm kích hoạt mà lớp mạng *FS* sử dụng được miêu tả trong công thức (2.3), trong đó, ε là tham số thích nghi có vai trò là một ngưỡng để giới hạn số lượng đặc trưng được lựa chọn trong giới hạn α đặc trưng.

$$f(x) = \begin{cases} 1, & |x| > \varepsilon \\ 0, & |x| \leq \varepsilon \end{cases} \tag{2.3}$$

Thành phần thứ hai của mạng nơ-ron là cấu trúc mạng perceptron mô phỏng bộ lọc thư rác SpamAssassin, gồm một lớp ẩn *P*. Đầu vào của mạng perceptron này là vector gồm những đặc trưng được chọn ở lớp trước, tức là các nơ-ron thuộc lớp *FS* có đầu ra

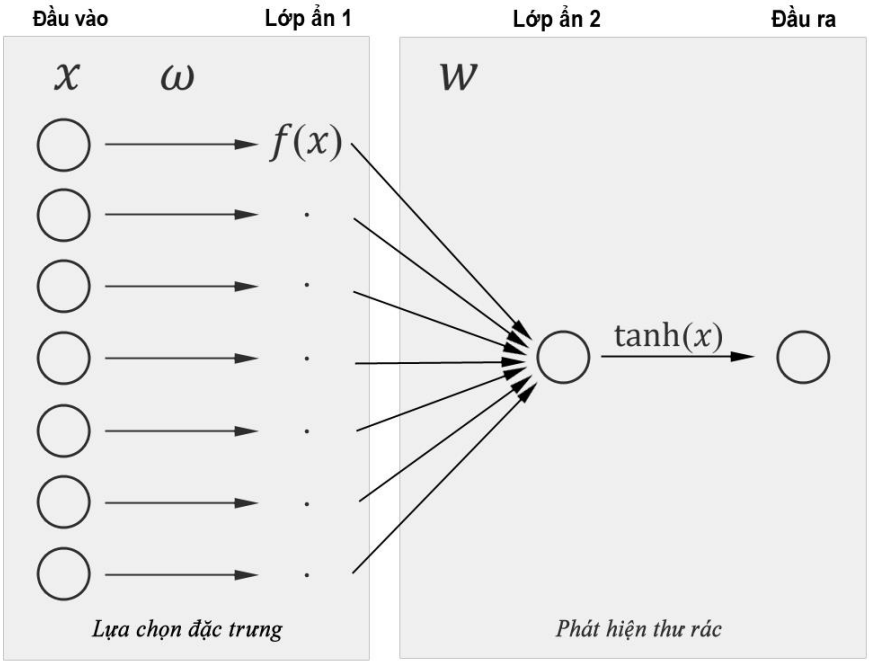
$f(x) = 1$ theo công thức (2.3). Trong thiết kế của mô hình huấn luyện, lớp P có số nơ-ron bằng $|\omega|$. Tuy nhiên, số lượng nơ-ron thực tế hoạt động là số đặc trưng được chọn, ký hiệu là β, thỏa mãn ràng buộc $\beta \leq \alpha$. Ta gọi tập trọng số thực tế trong lớp P là tập $w = (\omega_i \mid f(\omega_i) = 1) = (w_1, w_2, \ldots, w_\beta)$. Tập w được hiểu là tập trọng số của các đặc trưng. Đầu ra của lớp P được tính bằng công thức (2.4) và sau đó được áp dụng hàm kích hoạt *sigmoid* (1.6).

$$P_{out} = \sum_{i=1}^{n} x_i * f(\omega_i) * \omega_i \tag{2.4}$$

Mạng nơ-ron nói trên được huấn luyện dựa trên một biến thể của phương pháp SGD với *mini-batch* (loạt nhỏ). Phương pháp SGD gồm có 3 ưu điểm chính: (i) tốc độ nhanh, (ii) yếu tố tìm kiếm ngẫu nhiên của SGD thường cho kết quả huấn luyện tốt và (iii) thuật toán SGD giúp mô hình thích nghi với dữ liệu thư điện tử, một loại dữ liệu thường xuyên thay đổi. Theo [52], không gian tìm kiếm của mạng nơ-ron thường bao gồm nhiều điểm tối ưu cục bộ. Thuật toán SGD sử dụng một tập mẫu ngẫu nhiên của tập dữ liệu để ước lượng các đạo hàm của trọng số, dẫn đến giá trị đạo hàm liên tục dao động sau mỗi mẫu huấn luyện, khiến cho phương hướng tìm kiếm không ổn định. Chính vì vậy, vị trí tìm kiếm có khả năng di chuyển "nhảy cóc" từ khu vực của một điểm tối ưu cục bộ khu vực của điểm tối ưu cục bộ tốt hơn khi sử dụng thuật toán SGD. Lợi thế này đồng thời cũng là một điểm bất lợi vì sự dao động của các đạo hàm làm cho mô hình khó hội tụ tại một điểm tối ưu. Sử dụng *mini-batch* là phương án để cân bằng ưu và nhược điểm này của thuật toán SGD. Ngoài ra, nội dung thư rác thường xuyên thay đổi bởi vì người phát tán thư rác luôn luôn cải tiến phương pháp gửi thư rác để qua mặt các bộ lọc. Một ưu điểm nữa của thuật toán SGD là có thể đáp ứng việc cập nhật mô hình để thích nghi với tính chất của dữ liệu mới bằng cách tăng tỷ lệ các mẫu mới khi lựa chọn mẫu cho việc huấn luyện.

Ở mỗi vòng lặp của quá trình huấn luyện, trước hết, sai số được tính theo hàm tổn thất MSE (1.7) với kích thước *mini-batch* là $t = 15$. Kích thước *mini-batch* được lựa chọn để đáp ứng số lượng mẫu tham gia mỗi vòng lặp không quá ít và thời gian huấn luyện không quá lâu. Để thực hiện một lần cập nhật trọng số, t mẫu được lấy ngẫu nhiên từ tập dữ liệu huấn luyện để tính sai số trên các mẫu đó. Tập ω được cập nhật dựa vào

đạo hàm riêng của mỗi trọng số (1.9). Cuối mỗi vòng lặp, ngưỡng ε được cập nhật bằng cách sắp xếp tập ω theo thứ tự giảm dần giá trị tuyệt đối và lấy giá trị ở vị trí α làm giá trị của ε. Nhờ đó, số lượng đặc trưng được lựa chọn sẽ không vượt quá con số α đặc trưng. Một đặc trưng có thể bị thay thế bằng đặc trưng khác khi trọng số của nó giảm xuống dưới ngưỡng ε và trọng số của một đặc trưng khác tăng lên vượt ngưỡng ε. Giá trị ngưỡng ε có thể thay đổi, có thể tăng hoặc giảm, tùy thuộc vào tập trọng số ω ở mỗi thời điểm.

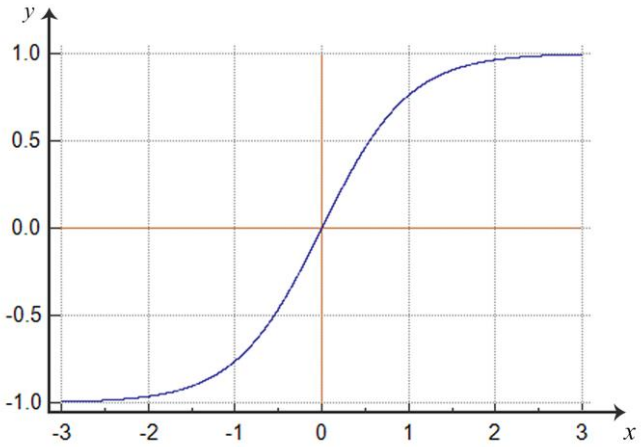


Hình 2.3: Cấu trúc mạng nơ-ron với hai phần lựa chọn đặc trưng (*FS*) và phát hiện thư rác (*P*)

Ở thời điểm bắt đầu huấn luyện, không có thuộc tính nào được lựa chọn làm đặc trưng. Kết quả đầu ra của mô hình được so sánh với giá trị nhãn của mẫu huấn luyện để tính ra sai số, từ đó tính được đạo hàm thành phần của các trọng số trong tập ω. Trong quá trình tập trọng số được cập nhật, những thuộc tính (từ khóa) nổi bật sẽ có trọng số đủ lớn để được lựa chọn làm đặc trưng. Chỉ những đặc trưng được chọn mới tham gia vào việc tính toán đầu ra của mô hình, nên việc có thêm đặc trưng sẽ ảnh hưởng đến giá trị sai số trong những lần huấn luyện tiếp theo. Những đặc trưng đã được lựa chọn cũng có thể bị loại bỏ khi trọng số của nó suy giảm xuống dưới ngưỡng ε trong hai trường hợp: (i) ngưỡng ε tăng lên để đảm bảo số lượng luật tối đa; (ii) trọng số của đặc trưng giảm xuống dưới ngưỡng ε sau khi được cập nhật bởi các mẫu huấn luyện. Các đặc

trưng sẽ được lựa chọn và loại bỏ liên tục. Mạng nơ-ron sẽ đạt được sự hội tụ khi việc thêm/bớt, hoán đổi đặc trưng không còn đem lại cải thiện về độ chính xác của mô hình.

Trong quá trình huấn luyện bằng thuật toán SGD, tuy đầu ra cuối cùng của mạng chỉ phụ thuộc vào tập trọng số của đặc trưng là tập *w*, nhưng đạo hàm được tính cho toàn bộ tập ω. Cách làm này tương tự với việc đồng thời đặt ra hai câu hỏi: (i) "trong số những thuộc tính chưa được chọn làm đặc trưng, những thuộc tính nào nên được chọn để giảm sai số của đầu ra cuối cùng?" và (ii) "trong những đặc trưng đã được chọn thì đặc trưng nào không thực sự quan trọng, cần bị loại bỏ?".



Hình 2.4: Đồ thị của hàm kích hoạt tanh.

Khi bắt đầu huấn luyện, tập trọng số ω cần được khởi tạo một cách ngẫu nhiên với những trọng số ban đầu có giá trị nhỏ. Trong phương pháp này, trọng số được khởi tạo với phân phối chuẩn Gauss có độ lệch chuẩn (scale) là 0.01. So với hầu hết điểm số của luật SpamAssassin thì giá trị 0.01 là đủ nhỏ để khởi tạo các trọng số. Tham số *learning_rate* trong thuật toán SGD được sử dụng để điều chỉnh mức độ tăng hoặc giảm của các trọng số trong công thức (1.9), từ đó kiểm soát tốc độ di chuyển trong không gian tìm kiếm. Tham số *learning_rate* thường được điều chỉnh dựa trên việc quan sát quá trình huấn luyện. Giá trị phổ biến nhất được khuyến nghị cho tham số này là 0.001. Thông qua quan sát quá trình huấn luyện mô hình mạng nơ-ron trong phương pháp đề xuất, giá trị tham số *learning_rate* có thể thay đổi trong khoảng [0.001, 0.01] mà vẫn cho phép mô hình hội tụ với kết quả nhất quán.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.5}$$

Khác với mô hình perceptron mặc định mà SpamAssassin sử dụng để huấn luyện tập luật, thay vì hàm kích hoạt *sigmoid* (1.6) thì mô hình đề xuất sử dụng hàm kích hoạt *tanh* (2.5). Hàm tanh có lợi thế so với hàm sigmoid vì nó cho phép huấn luyện nhanh hơn. Theo [52], hàm kích hoạt đối xứng tại giá trị 0 (Hình 2.4) giúp tăng tốc độ hội tụ cho mạng nơ-ron. Bởi vì *tanh* được chọn là hàm kích hoạt của mô hình, những bức thư rác sẽ có giá trị nhãn là 1.0 và những bức thư hợp lệ sẽ có giá trị nhãn là -1.0, tương ứng với giới hạn trên và giới hạn dưới của hàm kích hoạt.

2.2.4. Tạo tập luật SpamAssassin

Sau khi hoàn thành huấn luyện mô hình, những đặc trưng được chọn và trọng số tương ứng sẽ được sử dụng để sinh tập luật SpamAssassin. Mỗi đặc trưng tương đương với một từ khóa. Những từ khóa được chọn sẽ trở thành một luật trong tập luật. Nếu đặc trưng thuộc về tập $\mathbf{V_s}$, luật được tạo ra sẽ là kiểu luật HEADER. Nếu đặc trưng thuộc về tập $\mathbf{V_b}$, luật tạo ra sẽ có kiểu là BODY. Trọng số của đặc trưng sẽ được dùng để làm điểm số của luật. Trong cơ chế phân loại của mạng *perceptron*, đầu ra là tổng có trọng số của đầu vào được cộng thêm giá trị độ lệch *bias* (1.5). Đầu ra này được so sánh với ngưỡng 0 để đưa ra kết quả dự đoán. Điều đó tương đương với việc lấy tổng có trọng số của đầu vào và so sánh với ngưỡng -*bias*. Cơ chế phát hiện thư rác của SpamAssassin là so sánh tổng có trọng số của đầu vào (1.2) với ngưỡng T để đưa ra kết quả dự đoán. Nếu trực tiếp sử dụng trọng số của các đặc trưng để làm điểm số cho luật, ta có thể lấy giá trị -*bias* để làm ngưỡng cho SpamAssassin. Một cách khác để gán điểm số cho tập luật trong khi vẫn giữ nguyên giá trị ngưỡng $T = 5.0$ là sử dụng công thức (1.10) để tính điểm số của luật từ trọng số.

2.3. ỨNG DỤNG TỐI ƯU HÓA ĐA MỤC TIÊU ĐỂ XÁC ĐỊNH ĐIỂM SỐ CHO TẬP LUẬT SPAMASSASSIN

Trong phần này, tác giả đề xuất phương án ứng dụng tối ưu hóa đa mục tiêu để giải quyết vấn đề tồn tại thứ hai của bài toán lọc thư rác như đã trình bày ở trên. Mục tiêu đặt ra là tối ưu hóa bộ lọc thư rác dựa trên nền tảng SpamAssassin, hướng tới cân bằng các chỉ số quan trọng của tập luật, nhằm đạt được lợi ích cao nhất cho người sử dụng. Phương pháp được đặt tên là phương pháp SD_2 trong suốt luận án. Các thí nghiệm được

thực hiện để chứng minh phương pháp tối ưu hóa đa mục tiêu có nhiều ưu điểm hơn so với phương pháp cũ.

2.3.1. Ứng dụng tối ưu hóa đa mục tiêu để sinh tập luật SpamAssassin

Từ vấn đề trong quy trình sinh tập luật SpamAssassin đã nêu ở trên, có thể coi bài toán xác định điểm số cho các luật lọc thư rác là một bài toán tối ưu hóa đa mục tiêu trong đó ta cần tìm giá trị ngưỡng *T* và các giá trị điểm số của mỗi luật sao cho giá trị tham số *recall* và *FAR* của bộ lọc thu được là tối ưu. Do giữa *recall* và *FAR* có sự phụ thuộc lẫn nhau, trên thực tế không thể tìm được một bộ giá trị của *T* và các điểm số sao cho *recall* và *FAR* cùng đạt tối ưu (*recall* = 100% và *FAR* = 0%). Thay vào đó, ta có thể tối ưu hóa lợi ích mà tập luật mang lại bằng cách đi tìm tập các phương án thỏa hiệp, còn được gọi là các phương án tối ưu Pareto [22].

Giả sử tập mẫu ban đầu bao gồm tập thư rác $SP = (s_1, s_2, \ldots, s_K)$ và tập thư hợp lệ $H = (h_1, h_2, \ldots, h_L)$. Giả sử bộ lọc thư rác cần xây dựng bao gồm tập luật *S*, mỗi luật cần được xác định tương ứng là một phần tử trong tập điểm $X' = (x_1, x_2, \ldots, x_N)$. Một bộ lọc thư rác sử dụng luật SpamAssassin đưa ra kết quả dự đoán theo công thức:

$$Spam(m) = \begin{cases} 1, & G_S(m) \geq T \\ 0, & G_S(m) < T \end{cases} \tag{2.6}$$

Trong công thức nói trên, $G_S(m)$ là hàm tính điểm số của bức thư *m* khi áp dụng tập luật *S*. Hàm $G_S(m)$ được mô tả và giải thích trong công thức (1.2). Do bản thân giá trị ngưỡng *T* cũng là một biến số nên ta sử dụng x_0 để ký hiệu thay cho *T*. Khi bổ sung điểm x_0 vào tập *X'*, ta được tập $X = (x_0, x_1, \ldots, x_N)$. Các chỉ số *recall* và *FAR* của bộ lọc thư rác được tính theo các công thức:

$$\begin{aligned} recall(X) &= \frac{1}{K}\sum_{i=1}^{K} Spam(s_i) \\ FAR(X) &= \frac{1}{L}\sum_{i=1}^{L} Spam(h_i) \end{aligned} \tag{2.7}$$

recall(*X*) và *FAR*(*X*) là giá chị các chỉ số *recall* và *FAR* đạt được với tập luật *S* và ngưỡng $T = x_0$. Cuối cùng bài toán tối ưu hóa đa mục tiêu được phát biểu như sau:

$$z_1 = recall(X) \rightarrow Max$$
$$z_2 = FAR(X) \rightarrow Min \quad (2.8)$$

với X là tập các tham số tối ưu $X \in \mathbb{R}^{N+1}$ và các ràng buộc $x_{imin} \leq x_i \leq x_{imax}$; $i = 0...N$. Các giá trị $recall(X)$ và $FAR(X)$ được tính theo các công thức (2.7). Các giá trị x_{imin} và x_{imax} thể hiện khoảng giá trị cho phép của biến x_i.

2.3.2. Ứng dụng phương pháp tối ưu hóa Pareto

Trên thực tế, hai mục tiêu z_1 và z_2 của bài toán tối ưu hóa (2.8) không thể đạt được đồng thời, do đó phương pháp tối ưu hóa Pareto [22] được áp dụng để giải bài toán. Ta xem xét bài toán tối ưu hóa đa mục tiêu tổng quát với yêu cầu phải đồng thời tối thiểu hóa P hàm mục tiêu – các mục tiêu loại tối đa hóa có thể được chuyển thành loại tối thiểu hóa bằng cách nhân với -1:

$z_i = f_i(X) \rightarrow$ Min, $X = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^N$, $i = 1, 2, \ldots P$ ($P \geq 2$) với các ràng buộc: $g_j(X) = b_j$; $j = 1...m$.

Một phương án khả thi X được gọi là vượt trội so với phương án khả thi Y (ký hiệu $X \succcurlyeq Y$), nếu và chỉ nếu, $z_i(X) \leq z_i(Y)$ ($i = 1, ..., P$) và $z_j(X) < z_j(Y)$ ở ít nhất một mục tiêu j. Một phương án được gọi là phương án tối ưu Pareto nếu nó không bị vượt trội bởi bất cứ phương án nào khác trong không gian phương án $\{X\}$. Các giá trị hàm mục tiêu tương ứng của các phần tử trong tập các phương án tối ưu Pareto nói trên tạo thành *đường biên Pareto* trong không gian mục tiêu.

Các giải thuật tối ưu hóa đa mục tiêu lý tưởng sẽ tìm ra tất cả các phương án trong tập tối ưu Pareto. Tuy nhiên việc chứng minh một tập hợp các phương án tìm được là tập tối ưu Pareto thường không khả thi. Do đó một cách tiếp cận thực tế thường được chọn là tìm kiếm tập các phương án là thể hiện tốt nhất có thể của tập tối ưu Pareto, một tập các phương án như vậy được gọi là *tập Pareto được biết tốt nhất*.

Ba tiêu chí sau đây thường được dùng để đánh giá một tập Pareto được biết tốt nhất:

- Là một tập con của tập tối ưu Pareto.
- Các giá trị của hàm mục tiêu tương ứng của các phương án phải phân bố đều và đa dạng trên đường biên Pareto trong không gian mục tiêu.

– Các giá trị của hàm mục tiêu tương ứng phải biểu thị toàn cảnh của đường biên Pareto.

2.3.3. Các giải thuật tiến hóa đa mục tiêu

Với cách tiếp cận nói trên, việc giải bài toán tối ưu hóa đa mục tiêu được thực hiện thông qua quá trình tìm kiếm tập Pareto được biết tốt nhất. Do đó các giải thuật tìm kiếm dựa trên tiến hóa sẽ là các công cụ đặc biệt phù hợp để giải quyết lớp bài toán này. Thực tế các giải thuật tiến hóa đa mục tiêu như NSGA hay SPEA có thể thực hiện tìm kiếm tập Pareto được biết tốt nhất chỉ trong một lượt chạy. Theo thống kê trong [30], các giải thuật tiến hóa chiếm 70% trong tổng số các phương pháp tối ưu hóa đa mục tiêu dựa trên siêu kinh nghiệm.

Đã có nhiều giải thuật tiến hóa đa mục tiêu (MOEA) được công bố [61]. Điểm khác biệt chủ yếu giữa các giải thuật này nằm ở cách *tính độ thích nghi* cho mỗi cá thể, cách *duy trì quần thể ưu tú* và phương pháp để *đa dạng hóa quần thể*. Xếp hạng Pareto là một phương pháp thường dùng để tính độ thích nghi của cá thể bằng cách gán thứ hạng 1 (độ thích nghi cao nhất) cho các cá thể không bị vượt trội trong quần thể và loại chúng ra khỏi danh sách xếp hạng, rồi tìm các cá thể không bị vượt trội mới để gán thứ hạng 2 và tiếp tục như vậy cho đến khi toàn bộ quần thể được xếp hạng.

Duy trì quần thể ưu tú là một vấn đề quan trọng trong tối ưu hóa đa mục tiêu sử dụng MOEA. Trong ngữ cảnh của giải thuật MOEA, tất cả những cá thể không bị vượt trội được phát hiện bởi MOEA được coi như là những thành viên của quần thể ưu tú. Có hai chiến lược thường dùng để hiện thực việc duy trì quần thể ưu tú: (i) lưu trữ các cá thể ưu tú trong chính quần thể và (ii) lưu trữ các cá thể ưu tú trong một danh sách thứ cấp bên ngoài quần thể và đưa chúng trở lại quần thể.

Phương pháp *chia sẻ độ thích nghi* [7] được dùng để đa dạng hóa quần thể. Phương pháp này khuyến khích tìm kiếm trên những vùng chưa biết của đường biên Pareto bằng cách giảm bớt độ thích nghi của các cá thể ở những vùng có mật độ cao. Các kỹ thuật khác nhau thường được dùng để ước lượng mật độ các cá thể xung quanh một cá thể đang xét như kỹ thuật đếm số vùng lân cận (niche count) hay kỹ thuật tính khoảng cách mật độ trong đó ước tính giá trị khoảng cách Euclidean trung bình trong không gian mục tiêu của cá thể đang xét tới các láng giềng gần nhất thứ *k* (*k*-th nearest neighbor)

của nó. Khoảng cách mật độ cũng được dùng trong cơ chế chọn cha mẹ như sau: lấy ngẫu nhiên hai cá thể *x* và *y*; nếu chúng có cùng thứ tự (non-domination rank) thì cá thể nào có khoảng cách mật độ cao hơn sẽ được chọn; ngược lại cá thể có mức thứ tự thấp hơn sẽ được chọn.

2.3.4. Ứng dụng SPEA-II để giải quyết bài toán

Trong phần lớn các nghiên cứu hiện tại, việc tính điểm cho các luật dùng trong bộ lọc SpamAssassin được thực hiện thông qua việc giải bài toán tối ưu hóa đơn mục tiêu sử dụng giải thuật di truyền [28, 62] hoặc mạng nơ-ron [17]. Các giải thuật tiến hóa đa mục tiêu cũng được đã sử dụng hiệu quả trong vấn đề lọc thư rác tiêu biểu là các nghiên cứu ứng dụng MOEA để xác định các đặc trưng của mỗi luật [59] hoặc tạo ra các luật phức hợp từ các luật cơ bản [34]. Mô hình áp dụng các giải thuật tiến hóa đa mục tiêu như NSGA-II và DMEA-II để xây dựng tập luật SpamAssassin đã được giới thiệu [63] trong một nghiên cứu. Trong phương pháp đề xuất, SPEA-II [12] được lựa chọn vì đây là một trong những giải thuật tiến hóa đa mục tiêu được ứng dụng rộng rãi nhất và chưa từng được khảo sát trên cùng mô hình. Sau đây là một số điểm chính trong quá trình sử dụng SPEA-II để giải bài toán.

Biểu diễn nhiễm sắc thể: Bài toán yêu cầu tìm kiếm giá trị ngưỡng *T* và điểm cho từng luật có trong bộ lọc thư rác SpamAssassin sao cho các tham số *recall* và *FAR* của bộ lọc thu được là tốt nhất. Do đó mỗi nhiễm sắc thể sẽ biểu diễn một phương án khả thi để gán giá trị cho ngưỡng *T* và các luật có trong bộ lọc. Cụ thể mỗi nhiễm sắc thể sẽ là một vector chứa $N + 1$ số thực (các gen) tương ứng với một phương án $X = \{x_0, x_1, \ldots, x_N\} \in \mathbb{R}^{N+1}$ trong không gian phương án. Giá trị của mỗi x_i phải nằm trong ngưỡng cho phép $x_{imin} \leq x_i \leq x_{imax}$ đã xác định trước. Phương pháp mã hóa số thực (real-coded method) [61] được sử dụng để biểu diễn mỗi nhiễm sắc thể.

Tính toán giá trị hàm mục tiêu: Giá trị hàm mục tiêu của mỗi nhiễm sắc thể được tính toán thông qua phần mềm SpamAssassin. Bộ lọc SpamAssassin tương ứng với nhiễm sắc thể sẽ được sử dụng để kiểm tra các thư có trong tập mẫu bao gồm tập thư rác *SP* và tập thư hợp lệ *H*. Từ kết quả kiểm tra ta có thể tính được các tham số *recall* và FAR của bộ lọc và từ đó xác định được các giá trị hàm mục tiêu của nhiễm sắc thể.

Lưu ý để cho đơn giản ta chọn giá trị hàm mục tiêu (1 – *recall*) thay vì *recall*, như thế mục tiêu của bài toán sẽ là tối tiểu hóa hai hàm mục tiêu *FAR* và (1 – *recall*).

Cơ chế chọn lọc: Được được dùng để chọn các nhiễm sắc thể cha mẹ cho việc sinh ra thế hệ tiếp theo. Chúng tôi sử dụng cơ chế chọn lọc dựa trên đấu loại trực tiếp (binary tournament selection) trong đó hai nhiễm sắc thể được chọn ngẫu nhiên từ quần thể để tham gia đấu loại, nhiễm sắc thể nào có giá trị hàm thích nghi tốt hơn sẽ là người chiến thắng.

Phép toán lai tạo: Hai nhiễm sắc thể cha mẹ được chọn sẽ tạo ra hai nhiễm sắc thể con mới cho quần thể. Chúng tôi sử dụng phép toán lai tạo giả nhị phân (Simulated Binary Crossover) để thực hiện quá trình này.

Phép toán đột biến: Chúng tôi chọn phép đột biến đa thức (polynomial mutation operator) để biến đổi nhiễm sắc thể nhằm tăng tính đa dạng của quần thể.

Gán độ thích nghi: Phương pháp xếp hạng Pareto được dùng để gán độ thích nghi cho mỗi nhiễm sắc thể có trong quần thể.

Duy trì quần thể ưu tú: SPEA-II sử dụng một danh sách thứ cấp để lưu trữ các nhiễm sắc thể ưu tú (là các phương án không bị vượt trội như mô tả trong phương pháp tối ưu Pareto) của quần thể. Danh sách này sẽ được đưa lại vào quần thể trong quá trình chọn lọc.

Chia sẻ độ thích nghi: Sử dụng kỹ thuật tính khoảng cách mật độ đã trình bày ở trên.

2.4. THỰC NGHIỆM

2.4.1. Thí nghiệm ứng dụng mạng nơ-ron để sinh tập luật SpamAssassin

Thí nghiệm này so sánh 3 phương pháp sinh tập luật SpamAssassin sử dụng tập dữ liệu lọc thư rác tiếng Việt như đã mô tả ở phần trước. Phương pháp cơ sở (baseline) là một phương án sinh tập luật SpamAssassin đơn giản bao gồm việc lựa chọn các từ khóa phổ biến nhất trong tập dữ liệu để xây dựng tập luật và huấn luyện tập luật bằng phương pháp SGD được sử dụng mặc định bởi SpamAssasssin như được mô tả trong [17]. Trong phương án cơ sở, công cụ tách từ vnTokenizer [34] được sử dụng để tách từ tiếng Việt từ nội dung và tiêu đề thư. Phương pháp thứ hai tham gia vào thí nghiệm so sánh là một phương pháp sinh tập luật SpamAssassin được áp dụng cho tiếng Việt [62].

Bảng 2.1: Kết quả thí nghiệm so sánh một số
phương pháp sinh tập luật SpamAssassin dành cho tiếng Việt

Phương pháp	**Precision**			**F_1**		
	250 luật	500 luật	750 luật	250 luật	500 luật	750 luật
Cơ sở	0.8156	0.8716	0.8581	0.8578	0.8616	0.8807
Phương pháp [62]	0.9206	0.9372	0.9224	0.9235	0.9498	0.9517
Phương pháp đề xuất	0.9516	0.9621	0.9633	0.9699	0.9535	0.9693

Thí nghiệm được tiến hành theo phương án kiểm chứng chéo k lần truyền thống, với $k = 10$. Tập dữ liệu được trộn ngẫu nhiên và được chia thành 10 phần đều nhau với xấp xỉ cùng tỷ lệ thư rác và thư hợp lệ trong tất cả các phần. Trong mỗi lần huấn luyện và đánh giá mô hình, một phần dữ liệu được sử dụng làm tập thử nghiệm D_{test} trong khi những phần còn lại được dùng làm tập huấn luyện D_{train}. Để tìm hiểu về tác dụng của số lượng luật (tham số α), các thí nghiệm khác nhau được thực hiện với số lượng luật lần lượt là 250, 500 và 750 luật.

Đối với bài toán phát hiện thư rác, số lượng cảnh báo nhầm thường nhận được sự chú ý nhiều nhất bởi vì trường hợp phát hiện nhầm một bức thư hợp lệ là thư rác gây ra thiệt hại lớn hơn rất nhiều so với trường hợp ngược lại. Vì lý do này, *precision* (1.15) là một chỉ số hữu ích khi đánh giá hiệu năng của một bộ lọc thư rác khi áp dụng trên thực tế. Ngoài ra, chỉ số F_1 (1.16) cũng được sử dụng trong kết quả thí nghiệm để đưa ra một đánh giá trung lập giữa hai chỉ số *recall* và *precision*. Kết quả của thí nghiệm được tổng hợp trong Bảng 2.1. Trong phương pháp cơ sở và phương pháp [62], hiệu quả của tập luật có xu hướng tăng đáng kể khi số lượng luật mục tiêu tăng từ 250 lên 500 và 750. Trong khi đó, hiệu quả của phương pháp mới không tăng nhiều khi số lượng luật mục tiêu tăng lên. Điều này cho thấy phương pháp cũ chưa sắp xếp đặc trưng theo thứ tự giảm dần độ tốt một cách chính xác. Nói theo cách khác, phương pháp mới có thể đánh giá chính xác hơn chất lượng của các đặc trưng.

2.4.2. Thí nghiệm ứng dụng SPEA-II để sinh tập luật

Thử nghiệm xây dựng bộ lọc thư rác SpamAssassin dựa trên tối ưu hóa đa mục tiêu được thực hiện với hai kịch bản sử dụng hai tập dữ liệu thư điện tử khác nhau chứa 300 thư và 750 thư. Mục tiêu chính của thí nghiệm là so sánh hiệu quả của thuật toán tối ưu đa mục tiêu SPEA-II và thuật toán huấn luyện đơn mục tiêu. Do đặc điểm của thuật

toán tối ưu đa mục tiêu là tốn nhiều thời gian huấn luyện, một tập dữ liệu nhỏ đã được sử dụng để giảm thiểu thời gian thực hiện thí nghiệm. Trong mỗi kịch bản, tập thư điện tử ban đầu được chia thành hai tập con gọi là tập mẫu và tập kiểm tra. Tập mẫu được dùng trong quá trình tìm kiếm bộ lọc có các tham số *recall* và *FAR* tối ưu, còn tập kiểm tra được dùng để đánh giá bộ lọc khi hoạt động thực tế. Cả tập mẫu và tập kiểm tra đều chứa các thư rác và các thư hợp lệ. Bảng 2.2 mô tả số lượng thư cụ thể dùng trong mỗi kịch bản.

Bảng 2.2: Số lượng thư điện tử dùng trong các kịch bản.

	Kịch bản 1 (300 thư)		**Kịch bản 2 (750 thư)**	
	Huấn luyện	**Thử nghiệm**	**Huấn luyện**	**Thử nghiệm**
Thư rác	120	60	300	150
Thư hợp lệ	80	40	200	100

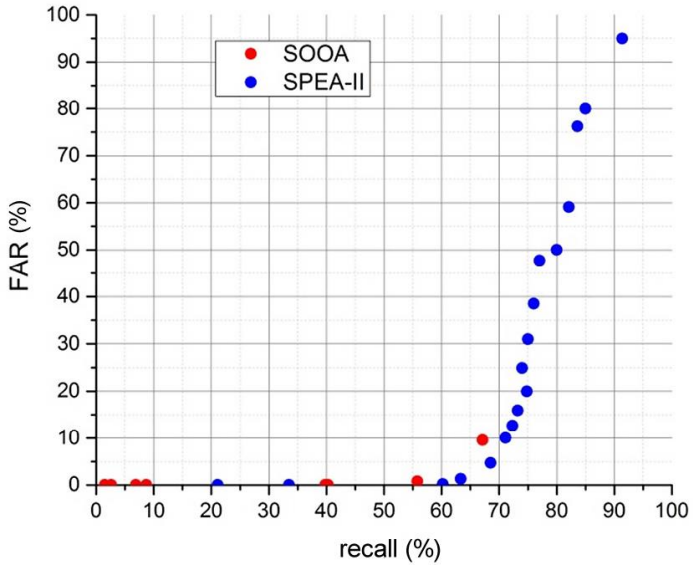
Trong mỗi kịch bản thử nghiệm, bộ lọc được thiết kế gồm 30 luật và 100 luật để đảm bảo các thực nghiệm được thực hiện với số lượng thư và số lượng luật ở quy mô nhỏ và quy mô lớn. Dải giá trị hợp lệ được chọn cho ngưỡng T là [0,5]; cho điểm của mỗi luật là [0,2]. Thuật toán SPEA-II được cài đặt để tính điểm cho mỗi luật có trong bộ lọc, các tham số của SPEA-II được mô tả trong Bảng 2.3 (N có giá trị lần lượt là 30 và 100).

Bảng 2.3: Các tham số của thuật toán SPEA-II.

Tham số	**Giá trị**	**Tham số**	**Giá trị**
Kích thước quần thể	100	Cận dưới của biến N+1	0
Số lượng thế hệ	1000	Cận trên của biến N+1	2
Số mục tiêu	2	Xác suất lai tạo	0.9
Số biến thực	N+1	Xác suất đột biến	1/(N+1)
Cận dưới của biến 1	0		
Cận trên của biến 1	5		

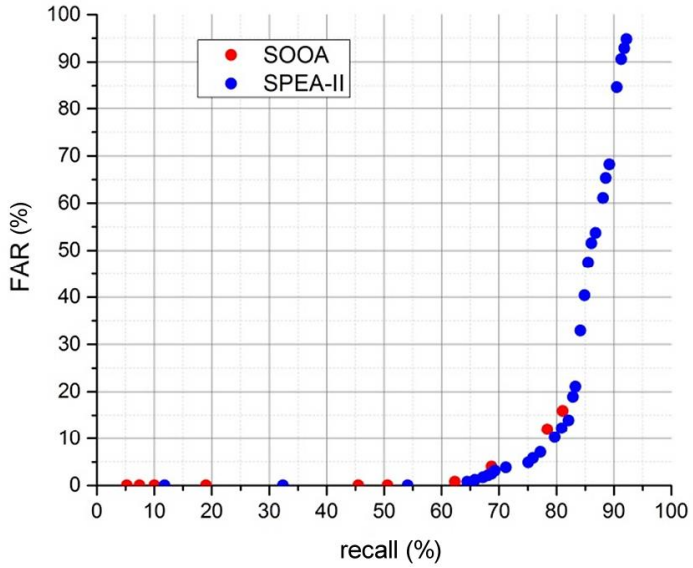
Để đảm bảo tính khách quan mỗi thử nghiệm được chạy 20 lần với các nhân ngẫu nhiên khác nhau, các số liệu trình bày trong luận án là giá trị trung bình của kết quả thu được sau mỗi lần chạy. Kết quả thử nghiệm được so sánh với phương pháp tính điểm tối ưu hóa đơn mục tiêu (SOOA) [28] của SpamAssassin trên cùng mẫu dữ liệu thư điện tử. Để thực hiện so sánh, 10 giá trị ngưỡng T phân bố đều trong khoảng [0,5] đã được chọn, với mỗi giá trị ngưỡng phương pháp tính điểm hiện tại sẽ tính toán điểm của các

luật để tối ưu hóa tham số *recall*, giá trị của tham số *FAR* của bộ lọc cũng được tính toán và so sánh với phương pháp đề xuất.



Hình 2.5: Kết quả kịch bản thí nghiệm 1 với bộ lọc 30 luật

Các kết quả thử nghiệm theo kịch bản thứ nhất (với tập chứa 300 thư) được trình bày trong Hình 2.5 (bộ lọc có 30 luật) và Hình 2.6 (bộ lọc có 100 luật). Các kết quả thu được khi thiết kế bộ lọc bằng phương pháp SOOA cũng được trình bày trong các hình vẽ này để tiện so sánh. Các số liệu cho thấy bộ lọc thiết kế sử dụng SPEA-II có các tham số *recall* và *FAR* tốt hơn so với bộ lọc thiết kế bằng phương pháp SOOA. Cụ thể đối với bộ lọc có 30 luật, giả sử ta muốn thiết kế bộ lọc có *FAR* = 0%, thì kết quả tốt nhất mà phương pháp SOOA tìm được là (*recall* = 40,8%, *FAR* = 0%). Trong khi đó sử dụng SPEA-II ta thu được bộ lọc có (*recall* = 60%, *FAR* = 0%). Tương tự nếu ta chỉ quan tâm đến những bộ lọc có *FAR* ≤ 10% thì các kết quả tốt nhất mà SOOA tìm được là (67.7%, 10.0%) và (55.8%, 1.25%) trong khi SPEA-II tìm ra những kết quả tốt hơn như (60%, 0.0%), (64.2%, 1.3%) và (68.3%, 5.0%).



Hình 2.6: Kết quả kịch bản thí nghiệm 1 với bộ lọc 100 luật

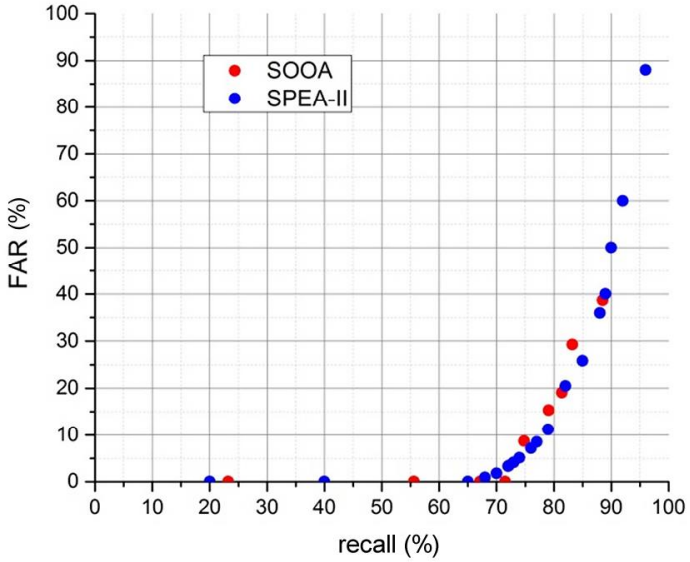
Khi tăng số luật của bộ lọc lên thành 100 luật ta cũng thu được các kết quả tương tự. Phương pháp SPEA-II cho kết quả tốt hơn SOOA trên cả hai phương diện tối ưu hóa riêng *FAR* hoặc *recall*. Hơn nữa bằng việc khảo sát đường biên Pareto, người dùng có thể cân nhắc việc đánh đổi giữa *recall* và *FAR* để từ đó tìm được giải pháp phù hợp với yêu cầu.

Bảng 2.4: So sánh kết quả thu được khi sử dụng hai phương pháp SSOA và SPEA-II trong kịch bản thí nghiệm 1

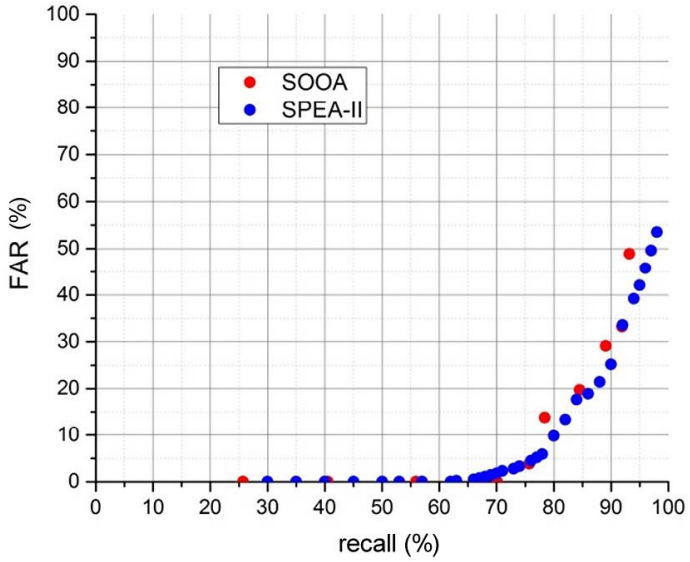
Phương pháp SOOA								
	Bộ lọc 30 luật				**Bộ lọc 100 luật**			
	Thiết kế		**Thực tế**		**Thiết kế**		**Thực tế**	
	rec.	*FAR*	*rec.*	*FAR*	*rec.*	*FAR*	*rec.*	*FAR*
1	67.7	10.0	65.0	12.5	81.3	15.0	81.7	17.5
2	55.8	1.25	56.7	2.5	78.3	12.5	78.3	12.5
3	40.8	0.0	45.0	2.5	68.3	3.8	66.7	5.0
Phương pháp SPEA-II								
	Bộ lọc 30 luật				**Bộ lọc 100 luật**			
	Thiết kế		**Thực tế**		**Thiết kế**		**Thực tế**	
	rec.	*FAR*	*rec.*	*FAR*	*rec.*	*FAR*	*rec.*	*FAR*
1	72.5	12.5	71.7	12.5	82.5	13.8	83.3	15.0
2	71.0	10.0	71.7	10.0	80.8	12.5	80.0	12.5
3	73.3	16.3	73.3	17.5	80.0	11.3	80.0	10.0

Trên mặt phẳng tọa độ tạo bởi hai trục *recall* và *FAR*, dễ thấy bộ lọc lý tưởng tương ứng với điểm có tọa độ I(100,0). Gọi D là khoảng cách Euclidean từ một điểm trên đường biên Pareto đến điểm I, khi đó giá trị của D sẽ cho ta thông tin ước lượng tương đối về chất lượng của bộ lọc thu được (D càng nhỏ thì chất lượng bộ lọc càng cao).

Bảng 2.4 trình bày 03 bộ lọc tốt nhất do SPEA-II tìm được và so sánh chúng với phương pháp SOOA. Các số liệu về *recall* và *FAR* khi thiết kế (sử dụng tập thư mẫu) và khi hoạt động thực tế (sử dụng tập thư kiểm tra) cũng được trình bày trong Bảng 2.4. Phân tích kỹ hơn số liệu trình bày trong bảng 2.4 ta thấy khi bộ lọc chứa nhiều luật hơn thì SPEA-II cũng tìm được những kết quả tốt hơn. Điều này thể hiện ở số lượng điểm tìm được trong tập đường biên Pareto (18 điểm và 31 điểm ứng với các trường hợp sử dụng 30 luật và 100 luật) và giá trị trung bình của khoảng cách D (51.3 và 47 ứng với các trường hợp sử dụng 30 luật và 100 luật).



Hình 2.7: Kết quả kịch bản thí nghiệm 2 với bộ lọc 30 luật



Hình 2.8: Kết quả kịch bản thí nghiệm 2 với bộ lọc 100 luật

Các kết quả thu được khi thử nghiệm theo kịch bản thứ hai với số lượng thư mẫu lớn hơn cũng cho thấy các kết luận tương tự như trong kịch bản thứ nhất:

- SPEA-II cho các kết quả tốt hơn so với SOOA trên cả hai phương diện tối ưu hóa *FAR* hay *recall*.
- SPEA-II cho phép người dùng lựa chọn các thiết kế phù hợp nhất căn cứ vào đường biên Pareto tìm được.
- Với bộ lọc sử dụng nhiều luật hơn thì SPEA-II tìm được các kết quả tốt hơn.

Bảng 2.5: So sánh kết quả thu được khi sử dụng hai phương pháp SSOA và SPEA-II trong kịch bản 2

Phương pháp SOOA								
	Bộ lọc 30 luật				**Bộ lọc 100 luật**			
	Thiết kế		**Thực tế**		**Thiết kế**		**Thực tế**	
	rec.	*FAR*	*rec.*	*FAR*	*rec.*	*FAR*	*rec.*	*FAR*
1	67.7	10.0	65.0	12.5	81.3	15.0	81.7	17.5
2	55.8	1.25	56.7	2.5	78.3	12.5	78.3	12.5
3	40.8	0.0	45.0	2.5	68.3	3.8	66.7	5.0
Phương pháp SPEA-II								
	Bộ lọc 30 luật				**Bộ lọc 100 luật**			
	Thiết kế		**Thực tế**		**Thiết kế**		**Thực tế**	
	rec.	*FAR*	*rec.*	*FAR*	*rec.*	*FAR*	*rec.*	*FAR*
1	72.5	12.5	71.7	12.5	82.5	13.8	83.3	15.0
2	71.0	10.0	71.7	10.0	80.8	12.5	80.0	12.5
3	73.3	16.3	73.3	17.5	80.0	11.3	80.0	10.0

Hình 2.7 và 2.8 mô tả kết quả thử ngiệm theo kịch bản thứ hai khi sử dụng bộ lọc có 30 luật và 100 luật. Bảng 2.5 tóm tắt các kết quả tốt nhất do SOOA và SPEA-II tìm ra trong thực nghiệm.

2.5. KẾT LUẬN CHƯƠNG 2

Chương này đã trình bày chi tiết về các phương pháp giải quyết bài toán lọc thư rác, một trong những dạng của bài toán xác định thứ tự ưu tiên của thư điện tử. Cả hai phương pháp được trình bày đều có mục tiêu xây dựng tập luật lọc thư rác cho SpamAssassin. Tập luật được sinh ra từ các phương pháp này có thể được sử dụng trực tiếp trên hệ thống SpamAssassin đã triển khai bằng cách thay thế tập luật cũ đang được sử dụng.

Phương pháp thứ nhất tập trung cải thiện chất lượng của tập đặc trưng bằng cách đồng bộ hóa khâu lựa chọn đặc trưng với khâu điều chỉnh trọng số luật, từ đó nâng cao

hiệu quả phát hiện của tập luật. Kết quả thí nghiệm so sánh cho thấy phương pháp đề xuất có hiệu quả cải thiện so với một phương pháp cơ sở đơn giản và phương pháp sinh tập luật SpamAssassin [62] dựa trên hai tiêu chí đánh giá là *precision* và điểm số F_1. Phương pháp đề xuất thứ hai tập trung cải thiện hiệu quả của tập luật SpamAssassin bằng phương pháp tối ưu hóa đa mục tiêu. Các phương án thỏa hiệp Pareto giữa hai tiêu chí đối nghịch *recall* và *FAR* được tìm ra bằng giải thuật tiến hóa đa mục tiêu SPEA-II. Dựa trên kết quả thí nghiệm phương pháp đề xuất với một phương pháp sinh tập luật SpamAssassin theo quy trình học máy truyền thống [28], phương pháp đề xuất không chỉ cho kết quả *recall* tốt hơn với cùng giá trị *FAR* mà còn cho phép lựa chọn phương án thỏa hiệp dựa theo giá trị cho trước của một tiêu chí.

Nội dung trình bày trong chương này là kết quả các công trình nghiên cứu số 2 và số 4 của tác giả. Công trình nghiên cứu số 2 đề xuất phương pháp sinh tập luật SpamAssassin dựa trên mạng nơ-ron. Công trình nghiên cứu số 4 đề xuất phương pháp sinh tập luật SpamAssassin dựa trên tối ưu hóa đa mục tiêu. Hai đề xuất nói trên thể hiện hai hướng tiếp cận khác nhau của tác giả đối với bài toán lọc thư rác. Đối với phương pháp từ công trình nghiên cứu số 2, các thí nghiệm đã được bổ sung và được thực hiện lại trên tập dữ liệu được giới thiệu trong chương này.

CHƯƠNG 3: DỰ ĐOÁN HÀNH ĐỘNG NGƯỜI DÙNG THƯ ĐIỆN TỬ

3.1. MỞ ĐẦU

Với số lượng thư hợp lệ nhận được ngày càng tăng, những mô hình xác định thứ tự ưu tiên phức tạp hơn so với bộ lọc thư rác đã trở nên cần thiết. Phương pháp dự đoán hành động người dùng đem lại lợi ích bằng cách gợi ý cho người dùng thư điện tử một hành động phù hợp để thực hiện đối với mỗi bức thư mà họ nhận được. Khi đó, người dùng không phải tốn thời gian đọc những bức thư mà đáng lẽ cần phải xóa đi, cũng như có thể sắp xếp thời gian để xử lý sớm những bức thư cần được phản hồi. Phương pháp này giúp cho người sử dụng thư điện tử tiết kiệm thời gian và không bị bỏ lỡ những bức thư quan trọng. Đã có nhiều phương pháp được đề xuất để xây dựng tập luật lọc thư rác cho SpamAssassin, nhưng chưa có nghiên cứu nào đề xuất ứng dụng SpamAssassin để dự đoán hành động của người dùng thư điện tử. Trong chương này, luận án sẽ phân tích những khó khăn, tồn tại của bài toán dự đoán hành động người dùng, từ đó đề xuất các phương pháp mới để giải quyết vấn đề quá tải thư điện tử trên nền tảng SpamAssassin một cách hiệu quả hơn.

3.1.1. Những khó khăn, tồn tại

Tự động dự đoán hành động người dùng là một bài toán được quan tâm nghiên cứu và đã được giới thiệu trong một số nghiên cứu [44, 51, 71, 74, 82]. Tuy vậy, hiệu quả dự đoán được công bố bởi các nghiên cứu theo hướng này vẫn còn hạn chế. Tiêu chí đánh giá của các phương pháp dự đoán hành động cũng không thống nhất. Phần lớn các phương pháp đều sử dụng tiêu chí đánh giá accuracy, một tiêu chí thông dụng cho bài toán phân loại đa lớp [51]. Trong khi đó, một số phương pháp khác sử dụng tiêu chí *recall* và *precision* từ bài toán phân loại nhị phân [44]. Trong nghiên cứu [51], chỉ số *accuracy* cao nhất đạt được là 87.88%. Hầu hết các nghiên cứu đều sử dụng tập dữ liệu tự thu thập [51], một số ít sử dụng tập dữ liệu công khai, ví dụ như tập dữ liệu Enron [44]. Một tập dữ liệu công khai như Enron có thể được sử dụng cho bài toán dự đoán thư trả lời [44]. Tuy nhiên, với bài toán với số lượng hành động lớn hơn 2, nhãn của dữ liệu cần phải được bổ sung.

Một khía cạnh quan trọng trong bài toán dự đoán hành động người dùng là cách mà các hành động được định nghĩa. Mỗi người dùng có cách lựa chọn riêng đối với hành động cần thực hiện trên một bức thư. Một mô hình dự đoán phù hợp với người dùng A sẽ không áp dụng hiệu quả cho người dùng B bởi vì hai người có thói quen sử dụng và quan điểm khác nhau. Chính vì lý do này, không giống như lọc thư rác, dự đoán hành động người dùng đối với thư điện tử là một bài toán có tính cá nhân hóa. Hầu hết các nghiên cứu về dự đoán hành động người dùng đều sử dụng dữ liệu thư điện tử cá nhân thay vì những tập dữ liệu công khai. Dữ liệu huấn luyện cũng cần được gán nhãn bởi chính người sở hữu hòm thư. Cách gán nhãn khác nhau của mỗi cá nhân là một trong những yếu tố cản trở việc xây dựng các tập dữ liệu công khai cho bài toán này. Các nghiên cứu về dự đoán hành động người dùng gặp nhiều khó khăn đến từ việc có ít dữ liệu huấn luyện, mặc dù các bài toán này có ý nghĩa to lớn trên thực tế. Một nguyên nhân nữa của tình trạng khan hiếm dữ liệu nghiên cứu cho bài toán xác định thứ tự ưu tiên của thư điện tử nói chung là quyền riêng tư của người sử dụng. Người dùng thường ngại chia sẻ những thông tin liên quan đến công việc và thông tin cá nhân nhạy cảm.

3.1.2. Hướng tiếp cận giải quyết bài toán

Một trong những hướng giải quyết vấn đề quá tải thư điện tử nêu trên là xây dựng hệ thống tự động gợi ý hành động mà người dùng cần làm đối với các bức thư. Tuy các mô hình dự đoán khó đạt được độ chính xác tuyệt đối nhưng vẫn có thể phần nào giúp người dùng nhanh chóng tìm được những bức thư cần xử lý, tiết kiệm thời gian dành để đọc nội dung của các bức thư.

Với cùng mục tiêu dự đoán hành động người dùng, những hướng tiếp cận khác nhau đã được đề xuất. Các yếu tố ảnh hưởng tới quyết định về hành động đối với thư điện tử của người dùng đã được tìm hiểu bằng một nghiên cứu dựa trên kết quả khảo sát [25] vào năm 2005. Kết quả của nghiên cứu đó cho thấy nội dung thư và mối quan hệ giữa người gửi và người nhận có ảnh hưởng đến hành động của người dùng đối với bức thư. Nghiên cứu [44] đặt mục tiêu phát hiện những bức thư cần được trả lời với phương pháp nhận dạng theo dấu hiệu (luật). Thuật toán mà [44] đề xuất bao gồm việc kiểm tra và gán điểm số khi một số đặc trưng xuất hiện trong nội dung thư và các trường CC, BCC của bức thư. Những đặc trưng này được thiết kế một cách thủ công, không đòi hỏi việc

sử dụng một tập dữ liệu. Trong nghiên cứu [51], Minh và cộng sự đề xuất sử dụng mô hình Naïve Bayes để xây dựng hệ gợi ý hành động cho người dùng thư điện tử. Nghiên cứu [51] cho thấy độ chính xác cao của mô hình Naïve Bayes khi phân biệt giữa các bức thư thuộc hành động "xóa" và hành động "trả lời". Trong khi đó, mô hình có độ chính xác thấp khi phân biệt giữa hành động "trả lời" và "đọc" bởi vì nội dung thư của hai hành động này có mức độ tương đồng cao.

Bộ lọc thư rác theo luật SpamAssassin có tốc độ xử lý thư điện tử nhanh, đáp ứng tốt yêu cầu xử lý một lượng thư lớn trong thời gian thực. Đã có nhiều phương pháp được đề xuất để xây dựng tập luật lọc thư rác cho SpamAssassin [17, 28, 62], nhưng chưa có nghiên cứu nào đề xuất ứng dụng SpamAssassin để dự đoán hành động người dùng cho thư điện tử với nhiều hơn hai cấp độ dự đoán.

Trong chương này, luận án sẽ lần lượt trình bày phương pháp xây dựng hệ thống dự đoán hành động người dùng dựa trên nền tảng SpamAssassin và những cải tiến dành cho phương pháp này. Đồng thời, phương án giải quyết các khó khăn, tồn tại về dữ liệu nghiên cứu cũng như về tính cá nhân hóa của bài toán dự đoán hành động người dùng cũng được trình bày.

3.2. DỰ ĐOÁN HÀNH ĐỘNG NGƯỜI DÙNG VỚI TẬP LUẬT SPAMASSASSIN

Trong phần này, bài toán dự đoán hành động người dùng sẽ được giải quyết bằng phương pháp phân loại. Phương pháp đề xuất được đặt tên là UAP_1. Gọi tập hợp những bức thư được gửi tới người dùng là $\boldsymbol{M} = \{m_1, m_2, \ldots, m_\infty\}$ và tập hợp các hành động của người dùng là $\boldsymbol{C} = \{c_1, c_2, \ldots, c_k\}$. Bài toán có mục tiêu là đi tìm một máy phân loại có dạng $\boldsymbol{f : M \rightarrow C}$. Bởi vì việc sở hữu tập $\boldsymbol{M}$ trên thực tế là không khả thi, thay vào đó, ta thu thập một tập $\boldsymbol{D} = \{m_1, m_2, \ldots, m_n\}$ là tập con của $\boldsymbol{M}$ và ước lượng máy phân loại $\boldsymbol{f}$ bằng cách đi tìm máy phân loại $\boldsymbol{g : D \rightarrow C}$. Trong phương pháp đề xuất, tập $\boldsymbol{C}$ bao gồm 3 hành động là *trả lời*, *đọc* và *xóa*, tương ứng với các gợi ý "*cần đọc và trả lời thư*", "*cần đọc nhưng không cần trả lời*" và "*xóa thư mà không cần đọc*" cho người dùng.

Một tập luật SpamAssassin hoạt động tương tự một máy phân loại nhị phân. Một tập luật đơn lẻ không đáp ứng được yêu cầu phân loại 3 lớp của bài toán dự đoán hành động người dùng nói trên. Với mỗi mẫu đầu vào, cụ thể là một bức thư điện tử, tập luật có

thể đưa ra một trong hai kết quả dự đoán. Tuy nhiên, một số kỹ thuật có thể được sử dụng để kết hợp nhiều máy phân loại nhị phân trở thành máy phân loại đa lớp. Một số kỹ thuật phổ biến có tên gọi OVA, OVO và DAG.

3.2.1. Xây dựng máy phân loại nhị phân

Tập luật SpamAssassin được dùng trong phương pháp này được xây dựng dựa trên phương pháp được đề xuất trong [28]. Đặc điểm của phương pháp này là chỉ có các đặc trưng trích xuất từ thư rác được sử dụng để xây dựng tập luật. Quy trình xây dựng tập luật có thể được tóm tắt theo các bước như sau:

Bước 1 – Chia tập dữ liệu huấn luyện thành hai phần: tập D_1 bao gồm thư rác và tập D_2 bao gồm các bức thư hợp lệ.

Bước 2 – Tách từ tiếng Việt từ tiêu đề và nội dung thư với công cụ vnTokenizer [34]. Các từ ngữ tách ra từ tiêu đề thư trong tập D_1 được lưu vào tập từ vựng WS_1 . Tương tự, các từ trong nội dung thư của tập D_1 được lưu vào tập từ vựng WB_1.

Bước 3 – Lựa chọn từ khóa: những từ phổ biến nhất được tách ra từ tập WS_1 và WB_1 để tạo thành hai tập mới, lần lượt là WS_2 và WB_2. Khi đó, $WS_2 = (\forall\, w \in WS_1, freq(w) > a)$ và $WB_2 = (\forall\, w \in WB_1, freq(w) > b)$, trong đó $freq(w)$ là số lần xuất hiện của từ w trong tập từ vựng tương ứng, a và b là hai tham số nhằm loại bỏ những từ khóa ít xuất hiện và có thể điều chỉnh được. Bởi vì mỗi từ khóa được lựa chọn sẽ tạo thành một luật trong tập luật đầu ra, hai tham số này cần được điều chỉnh tùy theo kích thước của tập dữ liệu để đáp ứng số lượng luật mục tiêu. Với tập dữ liệu được sử dụng trong thí nghiệm của chương này, giá trị của tham số a được chọn là 2 và tham số b được chọn là 6.

Công cụ MassCheck của SpamAssassin được sử dụng kiểm tra số lượng thư mà một từ khóa có mặt trong tiêu đề hoặc nội dung. Để thuận tiện, các từ khóa trong tập WS_2 được chuyển thành các luật HEADER và các từ khóa trong tập WB_2 được chuyển thành các luật BODY, tất cả được đưa vào một tập luật gọi là tập R_1. Đối với mỗi luật, số lượng thư có tồn tại luật đó trong tập D_1 và D_2 được đếm. Một tỷ lệ $R_t = V_{ts} / V_{th}$ được tính đối với mỗi luật, trong V_{ts} là mức độ liên quan giữa luật và thư rác, V_{th} là mức độ liên quan giữa luật và thư hợp lệ. Các giá trị V_{ts} và V_{th} được tính toán bằng công thức

conditional probability (1.4) theo lý thuyết xác suất của Bayes. Cuối cùng, *n* luật có tỷ lệ Rt cao nhất được giữ lại, tạo thành tập luật R_2 bao gồm những luật được lựa chọn. Hình 3.1 minh họa một luật HEADER được tạo ra từ một từ khóa trong tiêu đề thư.

```
header   ReplySubj_i Subject ~= /\b<word>\b/i
describe ReplySubj_i Subject contains “word”
score    ReplySubj_i 0.1
```

Hình 3.1: Cấu trúc của một luật HEADER trước khi được gán điểm số.

Bước 4 – Gán điểm số cho tập luật: Tập luật R_2 được huấn luyện với dữ liệu bằng phương pháp tối ưu điểm số của SpamAssassin như đã mô tả trong [17]. Phương pháp huấn luyện này có thể được tóm tắt như sau. Tập luật được coi là một mô hình mạng nơ-ron một lớp, còn gọi là *perceptron*, với hàm vận chuyển tuyến tính và hàm kích hoạt *sigmoid* (1.6). Mô hình này sau đó được huấn luyện bằng thuật toán SGD để tối thiểu hóa giá trị hàm tổn thất MSE (1.7). Điểm số dành cho tập luật được chuyển hóa từ trọng số của mạng perceptron bằng công thức (1.10). Cuối cùng, tập luật R_3 được tạo ra bằng cách thay thế điểm số đã huấn luyện vào các luật của tập luật R_2.

3.2.2. Xây dựng máy phân loại đa lớp

3.2.2.1. OVA (One vs. All)

OVA là một trong những phương án để kết hợp nhiều máy phân loại nhị phân thành máy phân loại đa lớp. Giả sử một máy phân loại N lớp cần được xây dựng, trong đó các lớp được gọi là X_i (i = 1, 2…N, N > 2). Khi đó, N máy phân loại nhị phân C_i (i = 1, 2…N) cần được xây dựng để hợp thành máy phân loại đa lớp theo phương án OVA. Mỗi máy phân loại C_i có khả năng phân biệt dữ liệu thuộc về một lớp X_i với dữ liệu thuộc về (N – 1) lớp còn lại. Trong một số tài liệu, phương pháp OVA còn có tên gọi là OVR. Vì một tập luật SpamAssassin là tương đương với một máy phân loại nhị phân, N tập luật cần được xây dựng để thỏa mãn phương án này. Hình 3.2 thể hiện thuật toán kết hợp các máy phân loại để đưa ra dự đoán đa lớp của phương án OVA. Khi xây dựng tập luật cho lớp X_i với quy trình xây dựng máy phân loại nhị phân được mô tả ở phần trước, tập dữ liệu D_1 sẽ bao gồm những bức thư thuộc về lớp X_i và tập D_2 sẽ gồm có những bức thư thuộc về hai lớp còn lại. Sau mỗi quy trình sinh tập luật, một tập luật RS_i sẽ được tạo ra và có N tập luật như vậy (RS_1, RS_2…RS_N).

Đầu vào: Bức thư m
N tập luật RS_i ($i = 1, 2 \ldots N$)
N ngưỡng T_i của tập luật RS_i ($i = 1, 2 \ldots N$)
Hành động dự đoán mặc định
Đầu ra: Hành động được dự đoán cho bức thư m

1. **Set** $S = new\ Array(),\ max = 0, class = defaultClass$
2. **For** $i = 1 \rightarrow N$
3. **Set** $S_i = Score_{RS_i}(m) \div T_i - 1$
4. **If** $(S_i > max)$ **then** { Set $class = i,\ max = S_i$ }
5. **Return** $(class)$

Hình 3.2: Thuật toán dự đoán theo phương án phân loại đa lớp OVA.

3.2.2.2. OVO (One vs. One)

Trong phương án OVO, ta cần xây dựng một máy phân loại $C_{i,j}$ để phân loại dữ liệu thuộc về hai lớp khác nhau bất kỳ X_i và X_j ($i \neq j$). Một bức thư đầu vào được xử lý bởi tất cả các máy phân loại, sau đó các kết quả phân loại được tổng hợp để đưa ra dự đoán cuối cùng. Có nhiều phương án tổng hợp kết quả dành cho OVO. Trong phương pháp này, một số phương án phổ biến là *max sum* (còn gọi là phương án "bỏ phiếu có trọng số") [6], *majority voting* (phương án "bỏ phiếu đa số") [3] và *most confident* (phương án bỏ phiếu "tự tin nhất") [49] sẽ được áp dụng. Với số lượng lớp là N, số lượng máy phân loại nhị phân cần được xây dựng là $N_R = N \times (N - 1) \div 2$. Ví dụ, với N = 3, 4, 5 thì N_R = 3, 6, 10. Gọi $RS_{i,j}$ là những tập luật cần xây dựng. Khi xây dựng tập luật $RS_{i,j}$ theo quy trình được mô tả ở phần trên, tập dữ liệu D_1 sẽ bao gồm những bức thư từ lớp X_i và tập D_2 sẽ là những bức thư thuộc lớp X_j. Ngưỡng của tập luật $R_{i,j}$ là $T_{i,j}$. Thuật toán trong Hình 3.3 được sử dụng để dự đoán kết quả theo phương án tổng hợp "bỏ phiếu có trọng số" OVO-MS.

Ở phương án OVO-MS, mỗi tập luật $RS_{i,j}$ sẽ đưa ra một điểm số dự đoán dưới dạng một giá trị số thực, tạm gọi là $S_{i,j}$. Nếu giá trị đó lớn hơn ngưỡng $T_{i,j}$, tức là tập luật dự đoán lớp C_i, thì ta cộng điểm số chênh lệnh giữa ngưỡng $T_{i,j}$ và điểm số $S_{i,j}$ vào quỹ điểm số của lớp C_i. Ngược lại, ta cộng điểm số chênh lệnh vào quỹ điểm của lớp C_j nếu tập luật dự đoán cho lớp C_j. Cuối cùng, lớp nào có quỹ điểm lớn nhất sẽ là lớp được dự đoán.

Đầu vào: Bức thư m
N_R tập luật $R_{i,j}$ ($1 \le i \le N$, $1 \le j \le N$, $i < j$)
N_R ngưỡng $T_{i,j}$ tương ứng với tập luật $R_{i,j}$
Hành động dự đoán mặc định *defaultClass*
Đầu ra: Hành động được dự đoán cho bức thư m

1. **Set** $S = new\ Array()$
2. **For** $i = 1 \rightarrow N$ { **Set** $S_i = 0$ }
3. **For** $i = 1 \rightarrow N - 1$
4. **For** $j = i + 1 \rightarrow N$
5. **Set** $tmp = Score_{R_{i,j}}(m) \div T_{i,j} - 1$
6. **Set** $S_i = S_i + tmp,\ S_j = S_j - tmp$
7. **Set** $equalCheck = \text{true},\ class = 1,\ max = S_1$
8. **For** $i = 2 \rightarrow N$
9. **If** $S_i \neq S_{i-1}$ **then** { Set $equalCheck = \text{false}$ }
10. **If** $S_i > max$ **then** { Set $class = i,\ max = S_i$ }
11. **Return** $(equalCheck\ ?\ defaultClass : class)$

Hình 3.3: Thuật toán tổng hợp kết quả dự đoán theo phương án OVO-MS.

Trong phương án thứ hai, "bỏ phiếu đa số" hay gọi tắt là OVO-MV, mỗi lớp có một quỹ phiếu bầu thay vì quỹ điểm số như trong phương án OVO-MS. Điểm số dự đoán $S_{i,j}$ đưa ra bởi mỗi tập luật $RS_{i,j}$ sẽ được so sánh với ngưỡng $T_{i,j}$ để xác định tập luật dự đoán cho lớp nào. Nếu tập luật $RS_{i,j}$ dự đoán cho lớp C_i thì quỹ phiếu bầu của lớp C_i được cộng thêm 1. Ngược lại, 1 sẽ được cộng thêm vào quỹ phiếu bầu của lớp C_j. Cuối cùng, lớp nào nhận được nhiều phiếu bầu lớn nhất sẽ là lớp được dự đoán. Thuật toán dự đoán OVO-MV được mô tả trong Hình 3.4.

Đầu vào: Bức thư m
N_R tập luật $R_{i,j}$ ($1 \le i \le N$, $1 \le j \le N$, $i < j$)
N_R ngưỡng $T_{i,j}$ tương ứng với tập luật $R_{i,j}$
Hành động dự đoán mặc định *defaultClass*
Đầu ra: Hành động được dự đoán cho bức thư m

1. **Set** $S = new\ Array(),\ max = 1,\ class = defaultClass$
2. **For** $i = 1 \rightarrow N$ { **Set** $S_i = 0$ }
3. **For** $i = 1 \rightarrow N - 1$
4. **For** $j = i + 1 \rightarrow N$
5. **If** $Score_{R_{i,j}}(m) \div T_{i,j} \ge 1$ **then**
6. **Set** $S_i = S_i + 1$
7. **Else**
8. **Set** $S_j = S_j + 1$
9. **For** $i = 1 \rightarrow N$
10. **If** $S_i > max$ **then** { Set $class = i,\ max = S_i$ }
11. **Return** $(class)$

Hình 3.4: Thuật toán tổng hợp kết quả dự đoán theo phương án OVO-MV.

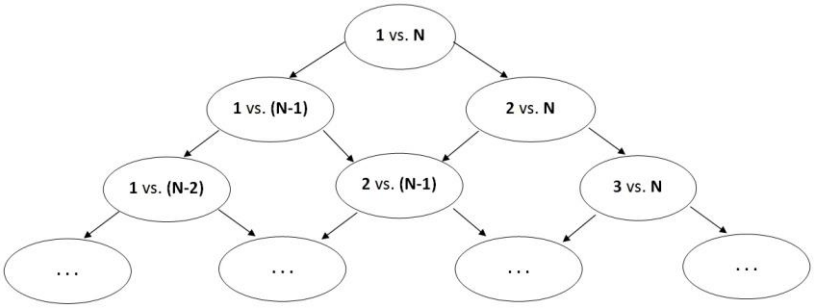
Đầu vào: Bức thư m
N_R tập luật $R_{i,j}$ ($1 \le i \le N$, $1 \le j \le N$, $i < j$)
N_R ngưỡng $T_{i,j}$ tương ứng với tập luật $R_{i,j}$
Hành động dự đoán mặc định *defaultClass*
Đầu ra: Hành động được dự đoán cho bức thư m

1. **Set** $S = new\ Array()$
2. **For** $i = 1 \rightarrow N$ { **Set** $S_i = 0$ }
3. **For** $i = 1 \rightarrow N - 1$
4. **For** $j = i + 1 \rightarrow N$
5. **Set** $tmp = Score_{R_{i,j}}(m) \div T_{i,j}$
6. **If** $tmp - 1 > S_i$ **then**
7. **Set** $S_i = tmp - 1$
8. **If** $1 - tmp > S_j$ **then**
9. **Set** $S_j = 1 - tmp$
10. **Set** $equalCheck = \text{true},\ class = 1,\ max = S_1$
11. **For** $i = 2 \rightarrow N$
12. **If** $S_i \neq S_{i-1}$ **then** { **Set** $equalCheck = \text{false}$ }
13. **If** $S_i > max$ **then** { **Set** $class = i,\ max = S_i$ }
14. **Return** $(equalCheck\ ?\ defaultClass : class)$

Hình 3.5: Thuật toán của phương án tổng hợp kết quả dự đoán OVO-MC.

Phương án cuối cùng được sử dụng là phương án phiếu bầu "tự tin nhất" OVO-MC, với thuật toán được mô tả trong Hình 3.5. Sau khi toàn bộ các máy phân loại nhị phân đã đưa ra điểm số dự đoán, điểm số $S_{i,j}$ nào cách biệt lớn nhất với ngưỡng $T_{i,j}$ sẽ được lựa chọn làm kết quả dự đoán cuối cùng.

3.2.2.3. DAG (Directed Acyclic Graph)



Hình 3.6: Mô hình dự đoán dựa trên cây nhị phân của phương án DAG.

Giống như OVO, phương án DAG yêu cầu một máy phân loại nhị phân cho mỗi cặp hai lớp X_i và X_j khác nhau. Như vậy, số lượng tập luật cần xây dựng cho phương án DAG tương tự là $N_R = N \times (N - 1) \div 2$. Tuy nhiên, DAG giảm thiểu số lần dự đoán của các máy phân loại nhị phân xuống còn $(N - 1)$ lần dự đoán bằng cách sử dụng một cây quyết định nhị phân (Hình 3.6). Các máy phân loại được sắp xếp theo thứ tự trong cây

nhị phân và một trong hai máy phân loại (nhánh trái hoặc nhánh phải) sẽ được chọn làm máy phân loại kế tiếp phụ thuộc vào kết quả của mỗi lần dự đoán. Để áp dụng phương án DAG cho bài toán dự đoán hành động người dùng, quy trình xây dựng các tập luật của phương án OVO được lặp lại và thuật toán trong Hình 3.7 được thực hiện để đưa ra kết quả dự đoán cuối cùng.

Đầu vào: Bức thư m
N_R tập luật $R_{i,j}$ ($1 \le i \le N,\ 1 \le j \le N,\ i < j$)
N_R ngưỡng $T_{i,j}$ tương ứng với tập luật $R_{i,j}$
Đầu ra: Hành động được dự đoán cho bức thư m

1. **Set** $i = 1,\ j = N,\ class = 0$
2. **While** $i < j$ **do**
3. **If** $Score_{R_{i,j}}(m) > T_{i,j}$ **then**
4. **Set** $j = j - 1,\ class = i$
5. **Else**
6. **Set** $i = i + 1,\ class = j$
7. **Return** $(class)$

Hình 3.7: Thuật toán dự đoán dành cho phương án DAG.

3.3. ÁP DỤNG LUẬT HAM ĐỂ CẢI THIỆN TẬP LUẬT SPAMASSASSIN TRONG BÀI TOÁN DỰ ĐOÁN HÀNH ĐỘNG NGƯỜI DÙNG

Trong phương pháp dự đoán hành động người dùng UAP_1 được giới thiệu ở trên, các đặc trưng trong dữ liệu thư hợp lệ, tức là các bức thư trong tập D_2, chưa được sử dụng trong quy trình xây dựng các tập luật SpamAssassin. Ngoài ra, tập dữ liệu dùng để xây dựng các tập luật trong phương pháp UAP_1 được gãn nhãn một cách thủ công, một phương án gây tốn thời gian và không tiện dùng cho người dùng phổ thông. Vì vậy, sau đây luận án sẽ mô tả phương pháp dự đoán hành động người dùng có tên UAP_2 để giải quyết hai điểm hạn chế nói trên.

3.3.1. Tự động gán nhãn cho dữ liệu

Thông thường, để xây dựng được một hệ thống xếp hạng thư điện tử dựa trên phương pháp học máy có giám sát, người dùng phải gán nhãn một cách thủ công cho hàng ngàn bức thư trong tập dữ liệu huấn luyện. Đây là một công việc tốn nhiều thời gian cho cả người sử dụng và những người làm nghiên cứu. Thêm vào đó, các nghiên cứu thường gặp khó khăn khi tìm dữ liệu thực tế để thực hiện thí nghiệm. Không chỉ e ngại việc

chia sẻ dữ liệu có chứa thông tin nhạy cảm, người dùng thư điện tử cũng hiếm khi đồng ý bỏ thời gian để gán nhãn những dữ liệu đó.

Dữ liệu thư điện tử có thể được tự động gán nhãn dựa trên thông tin về các hành động sử dụng thực tế của người dùng. Vì vậy, luận án đề xuất một kịch bản xây dựng mô hình dự đoán hành động người dùng có thể áp dụng trên thực tế để mang lại lợi ích cho cả người sử dụng và người làm nghiên cứu. Một công cụ gán nhãn dữ liệu tự động và bộ công cụ xây dựng và huấn luyện mô hình được gửi đến người sử dụng thư điện tử. Người dùng thực hiện thí nghiệm trên dữ liệu cá nhân với bộ công cụ đó và gửi các kết quả đầu ra (tập luật, các chỉ số, tham số…) về cho chuyên gia. Chuyên gia dựa vào kết quả thử nghiệm và những thông tin được ghi lại trong quá trình xây dựng mô hình để điều chỉnh các tham số cho phù hợp, rồi gửi lại bộ cấu hình mới cho người dùng để họ lặp lại việc xây dựng mô hình. Về phía người sử dụng, họ có được mô hình dự đoán hành động cá nhân để cải thiện năng suất làm việc trong khi không phải bỏ nhiều thời gian để gán nhãn dữ liệu của chính mình. Về phía nhà nghiên cứu, họ thu thập được kết quả thí nghiệm trên dữ liệu thực tế để hiểu rõ hơn về vấn đề nghiên cứu, về ảnh hưởng của các phương pháp khác nhau, những giá trị tham số khác nhau đối với hiệu quả của mô hình. Trong toàn bộ quy trình nói trên, dữ liệu thư điện tử cá nhân của người dùng được bảo mật.

Các trường *header* của một bức thư cho ta biết hành động nào đã được thực hiện trên bức thư đó. Từ đó, phần lớn những bức thư của người dùng, trừ các bức thư chưa được đọc, có thể được gán nhãn một cách tự động. Sau đây là các quy tắc được đề xuất để tự động gán nhãn cho dữ liệu thư điện tử.

- Những bức thư có nhãn “unread” là những bức thư chưa được đọc và sẽ được bỏ qua. Trong trường hợp này, ta không có đủ cơ sở để xác định nhãn của bức thư.
- Để tìm những bức thư đã được trả lời, ta phân tích các bức thư đã được người dùng gửi đi được lưu trong thư mục “Sent” và “Outbox”. Những bức thư được gửi đi có thể rơi vào hai trường hợp: (a) thư mà người dùng chủ động soạn để gửi đi và (b) thư mà người dùng gửi đi bằng cách trả lời một bức thư được nhận trước đó. Trong tất cả những bức thư ở thư mục “Sent” và “Outbox”, những bức thư có trường “In-Reply-To” là những bức thư thuộc trường hợp (b) nói trên. Trường “In-Reply-To” có chứa “Message-ID” của bức thư đã

được trả lời. Như vậy, ta cần gán nhãn "trả lời" cho bức thư có giá trị "Message-ID" đó.

- Những bức thư trong các thư mục "Trash", "Junk" hoặc "Spam" được gán nhãn "xóa".
- Những bức thư đã được đọc và chưa được gán nhãn còn lại trong thư mục "Inbox" sẽ được gán nhãn "đọc".

Một công cụ tự động gán nhãn dựa theo bộ quy tắc nói trên và hoạt động với định dạng hòm thư MBOX đã được xây dựng để tiến hành thí nghiệm.

3.3.2. Sinh tập luật SpamAssassin với luật Ham

SpamAssassin sử dụng phương pháp luật có trọng số với mục tiêu chính là để phát hiện thư rác. Với một bài toán có bản chất là phân loại nhị phân như vậy, các đặc trưng từ một trong hai lớp dữ liệu có thể là đủ để làm cơ sở dự đoán cho mô hình. Trong khi cách làm này có thể đưa ra kết luận một bức thư có là thư rác hay không, điểm số mà mô hình đưa ra không thể hiện chính xác mức độ tự tin trong dự đoán của nó. Gọi điểm số mà mô hình đưa ra là S và ngưỡng dùng để so sánh là T. T được coi là ranh giới giữa thư hợp lệ và thư rác. Nếu chỉ có các đặc trưng từ thư rác (luật *spam*) được dùng, điểm số S sẽ có xu hướng nghiêng nhiều hơn về phía có giá trị lớn hơn T. Trong trường hợp này, một bức thư có chứa cả luật *ham* và luật *spam* sẽ có điểm số S là tổng của các trọng số của các luật *spam* mà không tính đến các luật *ham*. Giá trị tuyệt đối $|S - T|$ càng lớn thể hiện mức độ tự tin càng cao đối với kết quả dự đoán. Ngay cả khi điểm số S nằm trong khoảng của thư hợp lệ ($S < T$) thì mức độ tự tin cũng thấp hơn so với mức độ tự tin nên có. Đó là bởi vì mức độ tự tin này đến từ việc mô hình xét các đặc trưng thuộc về thư rác, mà chưa xét các đặc trưng thuộc về thư hợp lệ. Nói theo cách khác, lượng thông tin mà mô hình dựa vào là chưa đầy đủ. Khi kết hợp các máy phân loại nhị phân để tạo thành máy phân loại đa lớp, đặc biệt là khi các phương án OVO và DAG được áp dụng, thì độ tự tin trong dự đoán của mỗi máy phân loại cần được đảm bảo. Độ chính xác của từng điểm số dự đoán sẽ đóng góp vào độ chính xác của kết quả dự đoán tổng hợp cuối cùng. Điều này được thể hiện rõ trong mô tả chi tiết các phương án OVO, OVA, và DAG ở phần trước.

Vì các lý do trên, luận án đề xuất cải tiến phương pháp xây dựng tập luật SpamAssassin như sau. Trong trường hợp tổng quát , một máy phân loại nhị phân được

xây dựng để phân loại giữa hai lớp C^+ và C^-. Gọi dữ liệu huấn luyện thuộc về lớp C^+ là tập D^+ và dữ liệu huấn luyện có gán nhãn C^- là tập D^-. Coi toàn tập từ vựng được trích xuất ra từ tiêu đề và nội dung thư trong tập D^+ là tập SC^+ và BC^+. Tương tự, ta có tập SC^- và BC^- là các tập từ vựng của tiêu đề và nội dung thư trong D^-. Tần số xuất hiện của các từ khóa trong các tập SC^+, BC^+, SC^- và BC^- cũng được đếm ghi lại. Những tập từ vựng này sẽ được sắp xếp theo thứ tự giảm dần tần số của từ khóa và những từ khóa ít xuất hiện sẽ được loại bỏ. Tương tự như các sinh tập trong phương pháp SA đã được miêu tả ở phần trước, ta có các tham số *a, b, c, d* để chỉ tần số tối thiểu của các từ khóa được giữ lại cho các tập vừ vựng SC^+, BC^+, SC^- và BC^-. Sau khi loại bỏ các từ khóa ít xuất hiện, ta thu được các tập từ vựng tương ứng là RSC^+, RBC^+, RSC^- và RBC^-.

Ta gọi A là sự kiện một bức thư bất kỳ trong tập D^+ hoặc D^- có chứa từ khóa *w* và S là sự kiện một bức thư thuộc về tập D^+. Như vậy, $\overline{S}$ là sự kiện một bức thư thuộc về tập D^-. Với mỗi từ khóa *w* thuộc tập RSC^+ và RBC^+, ta tính tỷ lệ $R_w = V_{w+} / V_{w-}$ trong đó $V_{w+} = P(S|A)$ và $V_{w-} = P(\overline{S}|A)$. Ngược lại, với mỗi từ khóa *w* thuộc tập RSC^- và RBC^-, ta tính tỷ lệ $R_w = V_{w-} / V_{w+}$. Các tập từ vựng được sắp xếp theo thứ tự giảm dần giá trị R_w. Sau đó, ta giữ lại *p* từ khóa có giá trị R_w cao nhất trong tập $RSC^+ \cup RBC^+$ và *q* từ khóa giá trị R_w cao nhất trong tập $RSC^- \cup RBC^-$. Các từ khóa được giữ lại sẽ hình thành tập luật. Theo kết quả của nghiên cứu [62], tập luật có hiệu quả tốt nhất khi tỷ lệ $q / (p + q)$ là từ 0.25 đến 0.5. Trong phương pháp này, *q* và *p* được lựa chọn với giá trị là 500. Như vậy, tổng số luật trong tập luật là 1000 luật với tỷ lệ 50% luật ham.

Cuối cùng, tập luật được huấn luyện với tập dữ liệu $D^+ \cup D^-$ bằng phương pháp tối ưu điểm số của SpamAssassin như đã mô tả trong [17]. Phương pháp huấn luyện này coi tập luật là một mô hình mạng *perceptron*. Mô hình này được huấn luyện bằng thuật toán SGD và các trọng số sau khi huấn luyện được chuyển hóa thành điểm số của luật.

3.4. ỨNG DỤNG PHƯƠNG PHÁP SD_1 TRONG MÔ HÌNH DỰ ĐOÁN HÀNH ĐỘNG NGƯỜI DÙNG

3.4.1. Cải tiến máy phân loại nhị phân trong mô hình phân loại đa lớp

Ở Chương 2, một phương pháp xây dựng tập luật SpamAssassin dựa trên mạng nơ-ron, tên gọi SD_1, với khả năng tự động lựa chọn đặc trưng đã được trình bày. Hướng tiếp cận của phương pháp dự đoán hành động người dùng UAP_1 và UAP_2 là kết hợp

máy phân loại nhị phân để giải quyết bài toán dự đoán hành động người dùng. Hiệu quả của mô hình dự đoán đa lớp phụ thuộc vào hiệu quả của các máy phân loại thành phần. Nhận thấy lợi ích tiềm năng của việc sử dụng những tập luật tốt hơn trong mô hình phân loại đa lớp, luận án đề xuất thay thế các máy phân loại nhị phân trong phương pháp UAP_1 và UAP_2 bằng các tập luật được sinh ra bằng phương pháp SD_1. Ta gọi phương pháp dự đoán hành động người dùng với kỹ thuật sinh tập luật SD_1 là phương pháp UAP_3.

3.4.2. Cải thiện trong khâu tiền xử lý dữ liệu

Với tập dữ liệu dự đoán người dùng đã được gán nhãn theo 3 hành động "trả lời", "đọc" và "xóa", một số bước tiền xử lý cần được thực hiện. Để xây dựng tập luật SpamAssassin, phần tiêu đề thư và nội dung thư được sử dụng. Trong tiếng Việt, có nhiều trường hợp một từ được bỏ dấu theo nhiều cách, dẫn đến tình trạng cùng một từ khóa được trích xuất thành nhiều đặc trưng khác nhau. Một số ví dụ minh họa cho trường hợp này là các từ "`thúy`" và "`thuý`", "`hoài`" và "`hòai`", "`tòa`" và "`toà`"… Hiện tượng văn bản tiếng Việt sử dụng các bảng mã tiếng Việt khác nhau cũng gây mất tính nhất quán trong dữ liệu. Bảng mã VIQR (được mô tả trong tiêu chuẩn RFC 1456 [2]) là cách biểu diễn tiếng Việt bằng bảng chữ cái 7-bit. Các ví dụ nói trên khi sử dụng bảng mã VIQR sẽ trở thành các từ "`thu'y`", "`hoa`i`" và "`to`a`". Ngoài ra, Unicode tổ hợp và Unicode dựng sẵn là hai bảng mã được sử dụng đồng thời bởi các nhóm người sử dụng khác nhau. Đối với người dùng, văn bản tiếng Việt sử dụng hai bảng mã này nhìn giống nhau nhưng khác nhau đối với máy tính. Cụ thể hơn, cùng là một ký tự "`ạ`", bảng mã Unicode dựng sẵn biểu diễn bằng chuỗi nhị phân "`E1 BA A1`" trong khi bảng mã Unicode tổ hợp biểu diễn bằng chuỗi "`61 CC A3`". Vì vậy, các bức thư cần được chuẩn hóa về cùng một chuẩn bỏ dấu và chuẩn mã hóa để đảm bảo tính nhất quán. Tiếp theo, một phương pháp tách từ tiếng Việt [34] được sử dụng để chia tiêu đề và nội dung thư thành các đơn vị ngôn ngữ. Những đơn vị ngôn ngữ được tách ra bởi công cụ tách từ bao gồm cả những chuỗi ký tự không đóng góp vào ý nghĩa của văn bản. Phương pháp đề xuất chỉ giữ lại các từ có nghĩa và các dấu chấm câu. Những đơn vị văn bản có tính chất cụ thể cũng bị loại bỏ, ví dụ như số điện thoại, địa chỉ thư điện tử, đường dẫn trang web và các giá trị số.

3.4.3. Sinh tập luật SpamAssassin dựa trên mạng nơ-ron

Mô hình mạng nơ-ron trong phương pháp SD_1 được áp dụng để xây dựng các máy phân loại nhị phân trong phương pháp này. Mô hình gồm 2 thành phần chính, lớp mạng *FS* có tác dụng lựa chọn đặc trưng và lớp mạng *P* có tác dụng phân loại bức thư đầu vào sử dụng các đặc trưng đã được chọn. Phương pháp sinh tập luật này lựa chọn đặc trưng bằng hàm kích hoạt (2.3), trong đó tham số ε tự động thích nghi trong khi huấn luyện để kiểm soát số lượng đặc trưng tối đa được lựa chọn. Lớp mạng *P* mô phỏng cơ chế phát hiện thư rác sử dụng luật có trọng số của SpamAssassin. Hàm kích hoạt *tanh* được sử dụng cho lớp *P* thay vì hàm *sigmoid* trong mô hình *perceptron* mặc định của SpamAssassin [17].

Mô hình được huấn luyện bằng thuật toán SGD với *mini-batch*. Thuật toán SGD thông thường sử dụng một mẫu ngẫu nhiên từ tập huấn luyện cho mỗi lần cập nhật trọng số của mạng. Cách làm này khác biệt với thuật toán GD khi toàn bộ tập huấn luyện được sử dụng cho mỗi lần cập nhật trọng số. Cách làm của thuật toán GD có ưu điểm là tiến trình huấn luyện đi theo một hướng ổn định và khả năng hội tụ tốt. Điểm hạn chế của thuật toán GD là tốc độ huấn luyện chậm và dễ bị hội tụ tại một điểm tối ưu cục bộ. Thuật toán SGD có lợi thế là thời gian huấn luyện ngắn và có khả năng thoát khỏi vùng tối ưu cục bộ. Tuy nhiên, SGD cũng có một vài hạn chế là hướng di chuyển không ổn định của quá trình huấn luyện và khả năng hội tụ thấp. *Mini-batch* là kỹ thuật được sử dụng để cân bằng các ưu và nhược điểm của hai thuật toán SGD và GD.

Để đánh giá hiệu quả khi thay thế các máy phân loại nhị phân cho bài toán dự đoán hành động người dùng, các thí nghiệm so sánh giữa phương pháp UAP_1, UAP_2 và UAP_3 sẽ được thực hiện ở phần tiếp theo.

3.5. THỰC NGHIỆM

3.5.1. Tiêu chí đánh giá

Ba tiêu chí đánh giá là *accuracy*, *precision vĩ mô* P_m (1.23) và FPR_{del} (3.1) được sử dụng để đánh giá hiệu quả của ba phương pháp trong thí nghiệm. Tiêu chí FPR_{del} được tính bằng cách lấy tổng số thư cần đọc và thư cần trả lời bị dự đoán nhầm là thư cần xóa, ký hiệu là fp_{del}, chia cho tổng số lượng thư cần đọc (n_{read}) và thư cần xóa (n_{reply}). Tiêu chí này là một tiêu chí quan trọng vì nó thể hiện tỷ lệ lỗi mang lại thiệt hại lớn nhất

cho người dùng. Một bức thư quan trọng bị dự đoán nhầm là thư cần xóa có thể khiến cho người dùng hoàn toàn bỏ qua bức thư đó. Ngay cả khi người dùng đọc những bức thư cần xóa, họ cũng thường đọc những bức thư này sau khi đã xử lý những bức thư được dự đoán là cần đọc và cần trả lời. Tiêu chí FPR_{del} được biểu diễn bằng đơn vị phần trăm. Chẳng hạn, giá trị FPR_{del} = 1.0 có nghĩa cứ 100 thư cần đọc và cần trả lời thì có 1 bức thư bị dự đoán nhầm là thư cần xóa. Tiêu chí precision vĩ mô (P_m) được tính bằng cách tính riêng tiêu chí precision đối với từng hành động và cuối cùng lấy trung bình cộng của ba giá trị. Tiêu chí này đặt trọng số lớn hơn cho những lớp có số lượng mẫu nhỏ. Trong trường hợp này, hai hành động với số lượng thư nhỏ là *xóa* và *trả lời* sẽ có sức ảnh hưởng ngang bằng với hành động *đọc*.

$$FPR_{del} = \frac{fp_{del}}{n_{read} + n_{reply}} \tag{3.1}$$

3.5.2. Thí nghiệm

Các thí nghiệm được thực hiện theo cấu hình kiểm chứng chéo *k* lần, *k* = 10 tăng độ tin cậy của kết quả. Bảng 3.1 tổng hợp kết quả của ba phương pháp được so sánh với kết quả chi tiết đối với từng phương án phân loại đa lớp. Các phương án phân loại đa lớp được sử dụng là OVA, OVO-MS, OVO-MV, OVO-MC và DAG.

Thí nghiệm so sánh cho thấy phương pháp UAP_2 giúp giảm tỷ lệ gợi ý nhầm đối với hành động xóa thư trong khi phương pháp UAP_3 giúp tăng độ chính xác chung của các gợi ý so với phương pháp UAP_1. Có thể nhận thấy trong phương án OVA của mô hình UAP_1, tỷ lệ dự đoán nhầm đối với hành động xóa (FPR_{del}) có trị số cao nhất. UAP_1 là mô hình trong đó các tập luật không sử dụng luật *ham*, dẫn đến tỷ lệ phát hiện nhầm của các máy phân loại thành phần đều cao hơn so với trường hợp có sử dụng luật *ham*. Chỉ số FPR_{del} cao cho thấy có nhiều bức thư thuộc hành động *đọc* và *trả lời* bị phát hiện nhầm thành thư có hành động *xóa*. Trên thực tế, thư cần *đọc* và *trả lời* có sự tương đồng tương đối lớn trong nội dung và khác nhiều so với nội dung thư cần *xóa*. Vì vậy, máy phân loại giữa hành động *xóa* và hai hành động còn lại (*đọc*, *trả lời*) thường có điểm số dự đoán tự tin hơn so với những máy phân loại khác, ví dụ như giữa hành động *trả lời* và hai hành động còn lại (*đọc*, *xóa*).

Bảng 3.1: Kết quả thí nghiệm so sánh các phương pháp UAP_1, UAP_2 và UAP_3 theo ba tiêu chí *accuracy*, P_m và FPR_{del} (%).

	Phương pháp	**UAP_1**			**UAP_2**			**UAP_3**		
	Tiêu chí	*Acc.*	P_m	FPR_{del}	*Acc.*	P_m	FPR_{del}	*Acc.*	P_m	FPR_{del}
Phương án	OVA	0.822	0.766	6.245	0.812	0.760	**0.855**	0.854	0.817	1.585
	OVO-MS	0.795	0.740	3.128	0.805	0.791	1.658	**0.865**	**0.839**	3.045
	OVO-MV	0.769	0.737	2.148	0.767	0.734	1.001	0.821	0.789	2.169
	OVO-MC	0.755	0.735	1.731	0.819	0.793	1.877	0.814	0.796	3.107
	DAG	0.804	0.806	1.345	0.761	0.713	0.675	0.802	0.839	2.940

Kết quả thí nghiệm cho thấy phương pháp UAP_2 có tiêu chí accuracy không cao hơn rõ rệt so với phương pháp UAP_1. Tuy nhiên, phương pháp UAP_2 có tỷ lệ dự đoán nhầm thư cần xóa, thể hiện ở tiêu chí FPR_{del}, thấp hơn đáng kể so với phương pháp UAP_1. Việc thêm vào các luật *ham* dẫn đến sự cải thiện này là phù hợp với lập luận trước đó về tác dụng giảm thiểu tỷ lệ cảnh báo nhầm. Luật *ham* tăng độ chính xác của điểm số dự đoán của các tập luật thành phần, giúp giảm thiểu những trường hợp dự đoán nhầm trong đó một bức thư cần *đọc* hoặc cần *trả lời* có chứa một số đặc trưng có trong thư cần *xóa*. Những trường hợp gợi ý nhầm thư cần *xóa* cho người dùng có thiệt hại lớn đã được giảm thiểu. So với hai phương pháp còn lại, phương pháp UAP_3 có tỷ lệ dự đoán đúng cao hơn, với phương án OVO-MS có độ chính xác thể hiện bởi tiêu chí *accuracy* lên tới 86.5%. Tuy nhiên, các dự đoán nhầm hành động xóa cũng có tỷ lệ cao hơn đáng kể cho với phương pháp UAP_2. Phương pháp UAP_3 sử dụng tất cả các đặc trưng, bao gồm các các đặc trưng từ thư hợp lệ (luật *ham*) nhưng lại có tỷ lệ dự đoán nhầm cao hơn so với UAP_2 bởi vì khi tỷ lệ phát hiện tăng thì tỷ lệ lọc nhầm cũng tăng. Khi chỉ số *accuracy* đạt được là cao nhất với phương áp OVO-MS của mô hình UAP_3 thì đồng thời chỉ số FPR_{del} cũng ở mức cao. Các cấu hình có sự cân bằng tốt nhất giữa độ chính xác dự đoán và tỷ lệ dự đoán nhầm của ba phương pháp lần lượt là: phương án DAG của phương pháp UAP_1, phương án OVA của phương pháp UAP_2 và phương án OVA của phương pháp UAP_3.

3.6. KẾT LUẬN CHƯƠNG 3

Thực trạng quá tải thư điện tử gây giảm năng suất làm việc của người dùng đã dẫn tới sự cần thiết xây dựng các hệ thống phân loại thư điện tử thông minh hơn, cụ thể là hệ thống dự đoán hành động đối với các bức thư nhận. Để tăng sự thuận tiện khi áp

dụng vào thực tế và không làm tốn thời gian cho người sử dụng, phương pháp dự đoán hành động người dùng trong chương này đặt mục tiêu triển khai trên hệ thống lọc thư rác phổ biến SpamAssassin và chú trọng vào tính năng tự động gán nhãn cho dữ liệu huấn luyện.

Chương này đã đề xuất dự đoán hành động bằng phương pháp phân loại. Các mô hình phân loại đa lớp đã được xây dựng bằng cách kết hợp nhiều tập luật SpamAssassin. Đối với bài toán dự đoán hành động người dùng, việc gán nhãn tự động cho dữ liệu thư điện tử là khả thi và cũng đã được đề xuất. Tác giả đã kết hợp các quy trình sinh tập luật SpamAssassin cũng như nhiều phương án phân loại đa lớp khác nhau trong thí nghiệm so sánh. Mục đích là để kiểm nghiệm hiệu quả của nhiều cách làm khác nhau, nhằm tìm ra cách làm hiệu quả thông qua thực nghiệm. Để phục vụ thí nghiệm, tập dữ liệu lọc thư rác tiếng Việt được mô tả trong Chương 2 tiếp tục được gán bổ sung các nhãn về hành động người dùng.

Để áp dụng kết quả của các đề xuất trong chương này vào một hệ thống thư điện tử có sử dụng bộ lọc thư rác SpamAssassin, một số tùy chỉnh cần được thực hiện. Một hệ thống thư điện tử thường bao gồm MTA (postfix, Sendmail, qmail…), MDA (procmail, maildrop…), hệ thống hòm thư (mbox hoặc Maildir) và một hệ thống phần mềm khách. Các phần mềm khách có thể được xây dựng trên nền tảng web (gọi là webmail), máy tính để bàn hoặc di động. Ví dụ về phần mềm webmail mã nguồn mở là Rainloop và Roundcube. Giả sử phần mềm khách trên nền tảng web được chọn cho hệ thống thư điện tử. Trong kịch bản ứng dụng phương pháp dự đoán hành động người dùng, ngoài việc nhận thư để đưa về hòm thư của người dùng, MDA cần đảm nhiệm thêm nhiệm vụ là thực hiện phân loại thư, sau đó lưu kết quả huấn luyện vào header của bức thư. Ta cần chỉnh sửa mã nguồn của MDA để thực hiện thuật toán phân loại đa lớp, ví dụ như OVA (Hình 3.2), trong đó có bao gồm việc thực thi bộ lọc SpamAssassin bằng các câu lệnh thông qua CLI trên máy chủ. So với việc tùy biến hệ thống webmail để thực hiện việc phân loại thì cách làm này giúp cải thiện trải nghiệm của người dùng. Thư được xử lý phân loại trước bởi MDA ngay khi máy chủ nhận được nên người dùng sẽ không phải đợi mô hình xử lý khi sử dụng hòm thư. Tiếp theo, ta cần tùy chỉnh hệ thống

webmail để đọc thông tin về kết quả phân loại từ header của mỗi bức thư để sắp xếp và hiển thị các bức thư cho người dùng.

Nội dung được trình bày trong chương này là tổng hợp kết quả từ công trình nghiên cứu số 1, số 2 và số 5 của tác giả. Nghiên cứu số 1 của tác giả đã giới thiệu mô hình dự đoán hành động với các máy phân loại thành phần là các tập luật SpamAssassin. Nghiên cứu đã tổng hợp các kỹ thuật phân loại đa lớp OVA, OVO-MS, OVO-MC, OVO-MV và DAG. Tiếp nối nghiên cứu số 1, nghiên cứu số 5 của tác giả đã đề xuất phương pháp tự động gán nhãn cho dữ liệu và thay đổi quy trình xây dựng các máy phân loại thành phần dựa trên đề xuất sử dụng luật ham cho tập luật SpamAssassin từ nghiên cứu [62]. Trong luận án, tác giả tiếp tục đề xuất áp dụng phương pháp xây dựng các máy phân loại thành phần dựa trên kết quả nghiên cứu về lọc thư rác từ đóng góp số 2.

CHƯƠNG 4: XẾP HẠNG THƯ ĐIỆN TỬ

4.1. MỞ ĐẦU

Việc xử lý toàn bộ thư nhận được không phải lúc nào cũng khả thi với người dùng khi họ nhận được quá nhiều thư. Số lượng thư điện tử trung bình mà một người nhận được mỗi ngày không ngừng tăng [57, 64], không thể tránh khỏi tình huống số lượng thư gửi đến luôn luôn vượt quá tốc độ đọc và xử lý của người sử dụng. Với những người sử dụng thư điện tử cho mục đích công việc, những bức thư không quan trọng làm mất thời gian và ảnh hưởng đến hiệu quả công việc.

Dựa trên thống kê tập dữ liệu thư điện tử tiếng Việt (Bảng 1.4), mức độ quan trọng của các bức thư trong hòm thư không giống nhau, phần lớn những bức thư hợp lệ mà người dùng nhận được là thư không quan trọng. Điều này dẫn đến việc người dùng gặp khó khăn khi cần xác định được những bức thư quan trọng. Khi đó, cần có các công cụ hỗ trợ người dùng sắp xếp các bức thư hợp lệ theo thứ tự ưu tiên từ cao xuống thấp. Nghiên cứu về bài toán xếp hạng thư điện tử có mục tiêu để phát triển công cụ nói trên.

Xếp hạng thư điện tử là một trong ba bài toán con của bài toán xác định thứ tự ưu tiên của thư điện tử. Phương án chung để ứng dụng mô hình xếp hạng thư điện tử vào thực tế là tự động sắp xếp hòm thư của người dùng dựa theo kết quả dự đoán của mô hình. Cụ thể, mô hình xếp hạng thư điện tử có nhiệm vụ dự đoán tầm quan trọng của các bức thư trong hòm thư của người dùng, sau đó phần mềm quản lý thư điện tử sẽ dựa vào đó và sắp xếp hòm thư sao cho các bức thư quan trọng hơn sẽ nhận được sự chú ý của người dùng sớm hơn. Nhờ đó, người dùng có thể tối ưu hóa lợi ích từ khoảng thời gian giới hạn mà họ dành để xử lý thư. Xếp hạng thư điện tử dựa trên phân loại đa lớp là cách tiếp cận phổ biến nhất để giải quyết bài toán này [40, 49].

Bài toán xếp hạng thư điện tử dựa trên phân loại được trình bày trong chương này được giải quyết theo hướng phân loại và được phát biểu như sau. Gọi tập hợp những bức thư được gửi tới người dùng là $\boldsymbol{M} = \{m_1, m_2, \ldots, m_\infty\}$ và tập hợp các mức độ ưu tiên đầu ra của mô hình xếp hạng thư điện tử là $\boldsymbol{P} = \{p_1, p_2, \ldots, p_k\}$. Bài toán có mục tiêu là đi tìm một máy phân loại có dạng $\boldsymbol{f : M \rightarrow P}$. Bởi vì việc sở hữu tập $\boldsymbol{M}$ trên thực tế là không khả thi, thay vào đó, ta thu thập một tập $\boldsymbol{D} = \{m_1, m_2, \ldots, m_n\}$ là tập con của $\boldsymbol{M}$ và ước lượng máy phân loại $\boldsymbol{f}$ bằng cách đi tìm máy phân loại $\boldsymbol{g : D \rightarrow P}$. Trong

phương pháp đề xuất, tập ***P*** bao gồm 5 mức độ quan trọng là *xóa*, *đọc không quan trọng*, đọc *quan trọng*, *trả lời không gấp* và *trả lời gấp*.

4.1.1. Những khó khăn và tồn tại

Trước khi xây dựng một mô hình để dự đoán mức độ ưu tiên của những bức thư, ta cần đưa ra định nghĩa rõ ràng về mức độ ưu tiên của thư điện tử. Một bức thư với mức độ ưu tiên cao sẽ được đặt trên ví trí đầu trong hòm thư của người dùng và sẽ được người dùng ưu tiên xử lý trước những bức thư có mức độ ưu tiên thấp hơn nó. Một số bức thư có tính khẩn cấp nhưng không có tầm quan trọng lớn, trong khi một bức thư khác lại quan trọng nhưng không khẩn cấp. Một vấn đề nảy sinh đó là trong hai bức thư nói trên, bức thư nào nên có mức độ ưu tiên cao hơn. Theo ma trận quản lý thời gian của Eisenhower [55], với một công việc khẩn cấp nhưng không quan trọng, ta nên ủy thác cho người khác làm, trong khi một công việc quan trọng nhưng không khẩn cấp nên được lên kế hoạch để thực hiện sau. Để ủy thác một công việc khẩn cấp hoặc để trả lời một bức thư khẩn cấp có thể không làm tốn nhiều thời gian, nhưng nếu không xử lý bức thư đó kịp thời thì người dùng có thể nhận thiệt hại đáng kể. Vì vậy, trong việc gán nhãn dữ liệu cho thí nghiệm trong nghiên cứu này, tính khẩn cấp được ưu tiên nhiều hơn so với tính quan trọng của bức thư. Tuy vậy, trên thực tế, dù là dựa vào tính quan trọng hay tính khẩn cấp, mỗi người dùng sẽ tự quyết định các tiêu chí để gán nhãn mức độ ưu tiên cho các bức thư của mình. Douglas và cộng sự [45] đã nêu quan điểm rằng tầm quan trọng của một bức thư là xác suất mà người dùng thực hiện một hành động đối với bức thư đó.

Vấn đề tiếp theo là độ khó phân loại đến từ số lượng mức độ ưu tiên. Nhìn chung, sử dụng nhiều nhãn hơn sẽ khiến cho người dùng khó khăn hơn trong việc gán nhãn dữ liệu huấn luyện, đồng thời làm tăng độ khó của mô hình dự đoán. Một số nghiên cứu sử dụng 3 nhãn để biểu diễn mức độ ưu tiên của thư điện tử [51] trong khi một số nghiên cứu khác [40, 49] sử dụng 5 nhãn.

Ngoài ra, trải nghiệm sử dụng thư điện tử của mỗi người là khác nhau và để cải thiện trải nghiệm này thì cần phải dựa trên dữ liệu thực tế của chính người dùng đó. Tuy nhiên, dữ liệu thư điện tử của một người thường nhỏ về số lượng và rất khó để người dùng có thể tự thực hiện tiền xử lý dữ liệu (dọn dẹp, chuẩn hóa dữ liệu, loại bỏ phần dư

thừa…). Điều này càng khiến cho bài toán xếp hạng thư điện tử trở nên khó giải quyết hơn.

4.1.2. Hướng tiếp cận của bài toán

Các nghiên cứu đã công bố về bài toán xếp hạng thư điện tử được chia thành hai cách tiếp cận: phân loại và hồi quy. Trong đó, hướng tiếp cận phổ biến nhất cho bài toán này là phương pháp phân loại [45, 49, 85]. Kết quả của nghiên cứu [49] đã cho thấy phương pháp phân loại có hiệu quả hơn so với phương pháp hồi quy đối với bài toán xếp hạng thư điện tử dựa trên dữ liệu thư điện tử cá nhân. Một cách tiếp cận dễ nhận thấy là ứng dụng phương pháp phân loại đa lớp dựa trên SpamAssassin mà luận án đã đề xuất trong Chương 3 dành cho bài toán gợi ý hành động người dùng. Luận án sẽ áp dụng và đánh giá hiệu quả của cách tiếp cận này thông qua thí nghiệm. Ngoài ra, các nghiên cứu trước đó về xếp hạng thư điện tử chỉ dừng lại ở các kỹ thuật học máy truyền thống. Chưa có nghiên cứu ứng dụng phương pháp học sâu để giải quyết bài toán xếp hạng thư điện tử.

Thông qua tham khảo tài liệu, luận án nhận thấy các thuật toán chủ yếu được sử dụng trong các nghiên cứu về phân loại thư điện tử và xếp hạng thư điện tử bao gồm máy phân loại Bayes [26], máy phân loại SVM [46], máy vector hỗ trợ hồi quy thứ bậc [49], mô hình logistic regression [45] và các mô hình khác như Random Forest, kNN [85], cho đến các mô hình học sâu như mạng MLP [80]. Những nghiên cứu đã công bố chia thư điện tử thành 3 hoặc 5 mức độ quan trọng, sử dụng kỹ thuật phân loại hoặc hồi quy và đã đạt được những thành tựu nhất định.

Trong những năm gần đây, các kỹ thuật học sâu [69] đã đạt được nhiều thành tựu khi được áp dụng vào bài toán phân loại văn bản và xử lý ngôn ngữ tự nhiên. Những mô hình mạng nơ-ron truyền thẳng như CNN và mạng nơ-ron hồi quy như LSTM đều đã đạt được kết quả tốt trong bài toán phân loại văn bản. Có một vài lý do có thể giải thích cho hiệu quả cao của các mô hình học sâu. Hiệu quả của những kỹ thuật học máy truyền thống phần lớn phụ thuộc vào quá trình trích chọn đặc trưng được thực hiện thủ công, tách rời với mô hình. Các mô hình mạng nơ-ron sâu có thể dựa vào cấu trúc phức tạp bên trong để tự động "học" đặc trưng từ dữ liệu. Mạng nơ-ron hồi quy còn có khả năng mô phỏng những chuỗi giá trị có thứ tự, giống như cách mà một đoạn văn trong ngôn ngữ tự nhiên được hình thành từ chuỗi các từ ngữ và ký hiệu được sắp xếp theo

thứ tự. Đặc biệt, mạng LSTM đã cho thấy khả năng học được sự phụ thuộc giữa các giá trị trong những chuỗi rất dài. Tuy chưa được áp dụng cho bài toán xếp hạng thư điện tử, các thuật toán auto-encoders xếp chồng [65], mạng nơ-ron tích chập theo thời gian [75] và mạng bộ nhớ ngắn dài hạn (LSTM) [84] đã được áp dụng cho bài toán phát hiện thư rác. Vì vậy, luận án đưa ra đề xuất áp dụng một mô hình học sâu để giải quyết bài toán xếp hạng thư điện tử.

Về mặt đặc trưng được sử dụng cho bài toán xếp hạng thư điện tử, có nhiều dạng đặc trưng trích xuất từ nội dung thư [49], đặc trưng mạng xã hội [26] hoặc kết hợp cả hai loại đặc trưng [40]. Đặc trưng nội dung là các đơn vị văn bản (từ, cụm từ) được trích xuất từ tiêu đề và nội dung thư, còn đặc trưng mạng xã hội được tính toán dựa trên đồ thị [26] mối quan hệ giữa địa chỉ thư điện tử của người gửi và người nhận. Word embedding là một kỹ thuật mới dành cho việc biểu diễn đặc trưng nội dung văn bản. Mỗi từ ngữ được biểu diễn bằng một vector số thực có kích thước cố định, gọi là *vector từ ngữ*. Trong số các cách để sinh ra vector từ ngữ, *word2vec* [56] là thuật toán được sử dụng rộng rãi nhất và cũng là thuật toán có kết quả tốt nhất nhất trong nhiều nghiên cứu đã được công bố. Thuật toán *word2vec* có khả năng tìm ra cách biểu diễn ý nghĩa của từ ngữ trong ngôn ngữ tự nhiên. Nói theo cách khác, các đặc trưng *word2vec* có thể đem đến khả năng hiểu ý nghĩa văn bản tới một mức độ nhất định cho mô hình học máy. Từ đó có thể thấy, tính chất này của word embedding sẽ mang lại lợi ích cho những tác vụ liên quan đến nội dung thư như bài toán xếp hạng thư điện tử.

Thêm vào đó, xu hướng nổi bật trong các nghiên cứu về thư điện tử đó là việc sử dụng các đặc trưng xã hội để biểu diễn các bức thư. Các đặc trưng mạng xã hội được áp dụng lần đầu tiên trong xử lý thư điện tử, cụ thể là để phát hiện thư rác, trong một nghiên cứu [26] vào năm 2005. Trong nghiên cứu nói trên, các tác giả đã đặt giả thiết rằng tầm quan trọng của một bức thư phụ thuộc vào người gửi bức thư đó và đưa ra kết quả thí nghiệm để hỗ trợ giả thiết này. Ngoài ra, nghiên cứu [40] cũng kết hợp nội dung thư và một số đặc trưng mạng xã hội để giải quyết bài toán xếp hạng thư điện tử. Những kết quả nghiên cứu đã cho thấy các đặc trưng mạng xã hội có thể giúp cải thiện hiệu quả của phương pháp xếp hạng thư điện tử. một mô hình học sâu kết hợp sử dụng các đặc trưng xã hội để xếp hạng thư điện tử cũng sẽ được luận án đề xuất và thử nghiệm.

Dựa trên các khảo sát nói trên, trong chương này, luận án đề xuất phương pháp xếp hạng thư điện tử theo hướng phân loại dựa trên mô hình học sâu. Phương pháp được đề xuất có ứng dụng các kỹ thuật biểu diễn đặc trưng hiện đại, cụ thể là phương pháp word2vec để biểu diễn từ ngữ, kết hợp giữa đặc trưng nội dung và đặc trưng xã hội của thư điện tử.

4.2. XẾP HẠNG THƯ ĐIỆN TỬ BẰNG PHƯƠNG PHÁP HỌC SÂU

Trong phần này, luận án trình bày một mô hình xếp hạng thư điện tử theo 5 mức độ ưu tiên, sử dụng phương pháp học sâu. Điểm mới trong phương pháp là sự kết hợp giữa đặc trưng văn bản biểu diễn dưới dạng vector từ ngữ *word2vec* và *đặc trưng mạng xã hội* được trích xuất từ dữ liệu. Một cấu trúc mạng nơ-ron sâu được trình bày để đáp ứng đầu vào kết hợp hai loại đặc trưng nói trên. Mạng nơ-ron này sử dụng các đơn vị LSTM kết hợp với một số kỹ thuật thiết kế và huấn luyện của mạng nơ-ron.

Trong những bài toán liên quan đến thư điện tử có một xu hướng về cá nhân hóa khá rõ rệt. Lý do đầu tiên cho sự xuất hiện của xu hướng này đó là dữ liệu nhạy cảm thường có trong nội dung thư, như những thông tin cá nhân hoặc thông tin công việc quan trọng. Lý do thứ hai đó là rất khó để đưa ra một giải pháp chung cho tất cả các nhóm người dùng bởi vì nội dung thư của mỗi người dùng đều mang đặc điểm riêng biệt. Chưa có nghiên cứu nào chỉ ra những đặc tính, đặc trưng chung trong dữ liệu thư điện tử của mọi người dùng. Vì vậy, dữ liệu thư đã nhận và gửi của người dùng là nguồn thông tin đầu vào đáng tin cậy nhất mà nghiên cứu này có thể dùng để xây dựng mô hình dự đoán để giải quyết các bài toán về thư điện tử.

4.2.1. Phương pháp học sâu trong xử lý thư điện tử

Vào đầu những năm 2010, khoảng thời gian với hàng loạt kỹ thuật học sâu xuất hiện, một nghiên cứu [65] đã khai thác khả năng học đặc trưng của mạng nơ-ron để giải quyết bài toán phát hiện thư rác. Mô hình phân loại được sử dụng là mô hình nhiều *auto-encoder* xếp chồng. Auto-encoder có tác dụng nén vector đầu vào *x* thành một vector ở không gian ít chiều hơn *x*’ và giải nén vector đó trở lại thành *x*. Từ đó cho thấy, *x*’ tuy ngắn hơn nhưng có chứa đầy đủ thông tin của *x*, hay nói cách khác, *auto-encoder* có tác dụng giảm số chiều không gian của dữ liệu trong khi vẫn giữ lại hầu hết những thông

tin hữu ích. Mô hình mạng nơ-ron trong [65] được xây dựng từ nhiều lớp truyền thẳng từ to tới nhỏ dần, cuối cùng là một lớp softmax. Những lớp truyền thẳng này lấy tập trọng số từ lớp ẩn của các auto-encoder. Dữ liệu đầu vào được thể hiện dưới dạng vector nhị phân Chuỗi các lớp truyền thẳng như vậy có tác dụng tạo ra các vector có kích thước giảm dần nhưng vẫn chứa hầu hết thông tin hữu ích từ vector đầu vào. Cuối cùng, kết quả phân loại được thể hiện ở lớp softmax. Trong phương pháp này, mô hình phân loại là một mạng nơ-ron truyền thẳng nhiều lớp MLP, nhưng điểm đặc biệt là các trọng số không được khởi tạo ngẫu nhiên như cách làm thông thường mà được huấn luyện sẵn bằng các auto-encoder. Các tác giả cho rằng việc sử dụng trọng số được huấn luyện từ trước sẽ cho phép mô hình phân loại đạt được trạng thái gần tối ưu ngay cả khi nó chưa được huấn luyện.

Những cấu trúc mạng nơ-ron mới tiếp tục được công bố. Một mô hình mạng nơ-ron đa thể (multimodal) được công bố bởi [75] để phân loại thư rác, sử dụng cả đầu vào văn bản và hình ảnh. Trong mô hình này, văn bản và hình ảnh được xử lý riêng biệt trước khi được kết hợp lại, và được tiếp tục xử lý bằng một lớp truyền thẳng và cuối cùng là một lớp đầu ra softmax. Đầu vào hình ảnh được xử lý bằng các lớp tích chập, còn đầu vào văn bản được biểu diễn bằng word embedding và được xử lý bằng các lớp tích chập và *max pooling* theo thời gian.

Một mô hình mạng MLP với 2 lớp ẩn được sử dụng trong [80] để phát hiện thư rác với tập dữ liệu SpamBase. Tập dữ liệu này cung cấp 57 đặc trưng dạng số, đó là tần số xuất hiện của 48 từ khóa, 6 ký tự và 3 chỉ số về các chuỗi những ký tự chữ in hoa liên tiếp trong nội dung bức thư. Báo cáo cho thấy mạng MLP có hiệu quả dự đoán cao hơn so với máy phân loại Naïve Bayes.

Mạng bộ nhớ ngắn dài hạn LSTM được biết đến với khả năng học những mối quan hệ phụ thuộc giữa các đặc trưng xuất hiện cách xa nhau trong chuỗi đầu vào. Mạng LSTM đã được áp dụng thành công trong các tác vụ xử lý ngôn ngữ tự nhiên như phân loại văn bản và dịch thuật [69]. Tác giả của nghiên cứu [84] định nghĩa *semantic LSTM* là mạng LSTM có đầu vào là word embedding. Phương pháp trong [84] là kết hợp giữa bộ word embedding được công bố bởi Google (dùng thuật toán word2vec) với các bộ cơ sở dữ liệu về từ gần nghĩa là WordNet và ConceptNet để tối đa hóa số lượng từ được

chuyển hóa thành word embedding. WordNet và ConceptNet có thể được dùng để tìm các từ gần nghĩa của một từ. Nếu embedding của một từ không tồn tại trong bộ embeddings của Google thì những từ gần nghĩa sẽ được tìm và embeddings của từ gần nghĩa nhất sẽ được sử dụng thay thế.

4.2.2. Tiền xử lý dữ liệu

Dữ liệu thư điện tử thô có chứa headers, nội dung thư và các tệp đính kèm. Thông tin hữu ích được trích xuất từ những thành phần dữ liệu nói trên. Trong nghiên cứu này, đặc trưng được trích xuất từ địa chỉ người gửi, địa chỉ người nhận, tiêu đề và nội dung văn bản của thư điện tử. Những bức thư có định dạng HTML được xử lý loại bỏ thẻ trong khi vẫn giữ lại cấu trúc đoạn văn bản. Khi thực hiện các thí nghiệm, thuật toán word2vec yêu cầu đầu vào huấn luyện là các câu riêng rẽ thay vì toàn bộ văn bản. Một số bước sau đã được thực hiện để tách câu từ nội dung văn bản của các bức thư:

- Tất cả các ký tự khoảng trống, bao gồm ký tự xuống dòng, trong mã nguồn HTML đều được hiển thị là khoảng trống trên trình duyệt. Vì lý do này, khi loại bỏ thẻ từ nội dung có định dạng HTML, các thẻ có xuống dòng và thẻ không xuống dòng được phân biệt. Nội dung trong một thẻ có xuống dòng sẽ được chuyển đổi thành một dòng trong văn bản đầu ra.
- Người dùng thư điện tử thường không tuân thủ đúng ngữ pháp, đặc biệt là về dấu chấm câu khi soạn nội dung thư. Trong văn bản đã loại bỏ thẻ HTML, mỗi dòng không được kết thúc bởi dấu chấm câu sẽ được sửa đổi. Một dấu chấm câu là một trong 3 ký tự sau: dấu chấm (“.”), dấu chấm hỏi (“?”) và dấu chấm than (“!”). Nếu một dòng kết thúc với một ký hiệu không thuộc về ba dấu chấm câu, ký hiệu đó sẽ được thay thế bởi dấu chấm(“.”). Nếu một dòng kết thúc bằng một chữ cái hoặc chữ số, một dấu chấm (“.”) sẽ được thêm vào cuối dòng.

Tiêu đề và nội dung của bức thư cần được tách ra thành các từ có nghĩa riêng biệt. Trong tiếng Anh, một từ đơn âm tiết hoặc đa âm tiết đều được biểu diễn dưới dạng một chuỗi ký tự liền mạch không chứa khoảng trống. Không giống như tiếng Anh, trong những từ đa âm tiết của tiếng Việt, các âm tiết được tách rời bởi khoảng trống. Tùy theo ngữ cảnh, hai âm tiết kế tiếp nhau có thể được hiểu là một từ đa âm tiết (từ ghép) hoặc hai từ đơn. Trong bài báo này, các tác giả sử dụng bộ công cụ VNCoreNLP [78] để xử

lý văn bản đầu ra đã được loại bỏ thẻ HTML. Ngoài chức năng tách từ, bộ công cụ VNCoreNLP còn có chức năng POS Tagging dùng để xác định từ loại của các từ. Chức năng này được dùng để xác định và loại bỏ những thuộc tính không hữu dụng để có thể chọn làm đặc trưng, ví dụ như các ký tự trang trí, các chuỗi ký tự không có nghĩa, các giá trị dạng số cụ thể… VNCoreNLP cho đầu ra là tập hợp các câu được tách từ văn bản, mỗi câu bao gồm các từ được tách riêng cùng thông tin về từ loại kèm theo mỗi từ.

4.2.3. Biểu diễn đặc trưng mạng xã hội

Địa chỉ thư điện tử của người gửi thư là thông tin có khả năng hỗ trợ xác định tầm quan trọng của bức thư. Việc biểu diễn thuộc tính của người gửi dưới dạng vector được dựa trên giả thiết rằng một người sẽ tiếp tục gửi những bức thư có tầm quan trọng giống với những bức thư mà người đó đã gửi trong quá khứ. Mỗi bức thư trong tập dữ liệu được gán một trong 5 nhãn tương ứng với 5 mức độ quan trọng. Ta có thể đếm số lượng thư thuộc về mỗi mức độ quan trọng của một người gửi trong dữ liệu huấn luyện. Như vậy, mỗi người gửi có thể được biểu diễn bởi một tập hợp 5 số nguyên. Một hàm *sigmoid* (4.1) được sử dụng để chuẩn hóa những giá trị số nguyên đó thành các giá trị số thực trong khoảng [0, 1).

$$\frac{x}{1+|x|} \ (x \geq 0) \tag{4.1}$$

Số lượng thư mà người gửi đã nhận cũng được đếm để đưa vào vector người gửi dựa trên giả thiết rằng một kẻ phát tán thư rác thường nhận rất ít hoặc hoàn toàn không nhận thư. Ngược lại, một người dùng quan trọng thường nhận rất nhiều thư gửi về. Nói theo cách khác, ta đặt giả thiết rằng một người dùng thư điện tử tích cực sẽ không phát tán thư rác mà sẽ gửi những bức thư có mức độ quan trọng cao. Số lượng thư đã nhận cũng được bình thường hóa bằng hàm sigmoid như đã đề cập ở trên và được thêm vào vector đặc trưng người gửi, khiến cho vector bao gồm tổng cộng 6 phần tử.

4.2.4. Biểu diễn đặc trưng nội dung

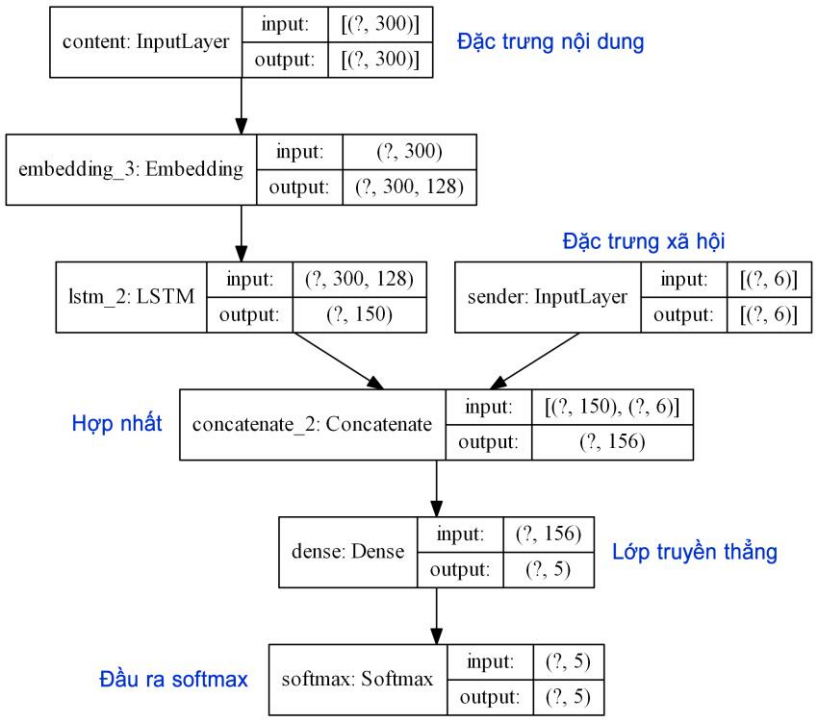
Đặc trưng nội dung trong phương pháp đề xuất được vector hóa bằng phương pháp word embedding với thuật toán *word2vec*. Có hai cách để sử dụng word embedding: word embedding huấn luyện sẵn (pre-trained) và word embedding trực tuyến (online). Word embedding huấn luyện sẵn là việc sử dụng một tập dữ liệu văn bản để tạo ra một

bộ vector từ ngữ của tất cả các từ xuất hiện trong tập dữ liệu đó, dùng phương pháp học máy không giám sát. Sau đó, bộ vector từ ngữ đó được dùng làm trọng số của lớp Embedding trong mạng nơ-ron, các trọng số của lớp này không đổi khi huấn luyện mạng. Lớp này lấy đầu vào là văn bản được vector hóa dưới dạng chuỗi chỉ mục từ ngữ (là một chuỗi các số nguyên) trong tập từ vựng và chuyển hóa văn bản thành một ma trận $n \times m$ với n là độ dài văn bản và m là độ dài của mỗi vector từ ngữ. Word embedding trực tuyến là việc khởi tạo trọng số ngẫu nhiên cho lớp Embedding và huấn luyện lớp này cùng với mạng nơ-ron. Trong phương án sử dụng word embedding huấn luyện sẵn, luận án sử dụng thuật toán *word2vec* có trong bộ công cụ Gensim để tạo bộ vector từ ngữ từ dữ liệu huấn luyện. Thuật toán word2vec thường yêu cầu một tập dữ liệu lớn để sinh ra bộ vector từ ngữ có chất lượng tốt. Các bộ vector từ ngữ huấn luyện sẵn và được công bố bởi các hãng hoặc phòng thí nghiệm lớn (giống như bộ vector từ ngữ word2vec của Google) thường được khuyên dùng. Tuy nhiên, những bộ vector từ ngữ như vậy dành cho tiếng Việt không có sẵn. Vì vậy, các thí nghiệm trong luận án được thực hiện trong phạm vi tập dữ liệu thư điện tử tiếng Việt được mô tả trong mục 1.4.4. Đối với phương án word embedding trực tuyến, lớp Embedding của bộ công cụ học sâu Keras được sử dụng. Lớp này được đặt ở ngay sau đầu vào của mạng và các trọng số của lớp được huấn luyện cùng với toàn bộ mạng nơ-ron. Việc sử dụng word embedding huấn luyện sẵn và word embedding trực tuyến là hoàn toàn khác nhau về nguyên lý. Vector từ ngữ huấn luyện sẵn được sinh ra bằng cách học máy không giám sát (sử dụng cơ chế của auto-encoder), còn vector từ ngữ trực tuyến được huấn luyện có giám sát dựa theo nhãn của tập huấn luyện. Bài báo này so sánh hiệu quả của hai cách làm nói trên.

Kích thước (số chiều không gian) của vector từ ngữ cũng được nhiều học giả quan tâm nghiên cứu. Google chọn con số 300 làm kích thước vector trong bộ vector từ ngữ word2vec được công bố của họ. Lý do cho lựa chọn này không được công khai. Giá trị này cũng được lựa chọn bởi nhiều nghiên cứu khác [81]. Trong chương này, tác giả luận án không đề xuất một phương pháp để lựa chọn kích thước vector từ ngữ mà thay vào đó khảo sát hiệu quả khi sử dụng số chiều không gian nhỏ hơn 300 dựa trên giả thiết rằng một tập dữ liệu nhỏ sẽ cần các vector từ ngữ nhỏ hơn. Các thí nghiệm trong chương này sẽ so sánh các phương án kích thước vector từ ngữ lần lượt là 128 và 300 để xác minh giả thiết nói trên.

4.2.5. Cấu trúc mạng nơ-ron

Cấu trúc mạng nơ-ron trong Hình 4.1 được đề xuất để kết hợp đặc trưng người gửi một cách hiệu quả với các đặc trưng nội dung. Nội dung của một bức thư được biểu diễn bằng một tập hợp vector từ ngữ. Biểu diễn của phần nội dung là một ma trận kích thước $n \times m$ với n là độ dài nội dung thư và m là số chiều không gian vector từ ngữ. Độ dài văn bản n được cố định là 300 để đơn giản hóa việc thực hiện thí nghiệm trên bộ công cụ Keras. Các văn bản dài hơn 300 từ được cắt ngắn, các văn bản ngắn hơn 300 từ được tăng độ dài thành 300 bằng kỹ thuật đệm số 0 (*zero padding*). Các đặc trưng người gửi được biểu diễn trong vector kích thước 6 phần tử.



Hình 4.1: Cấu trúc mạng nơ-ron đề xuất dành cho đầu vào kết hợp đặc trưng nội dung (dưới dạng word embedding) và đặc trưng xã hội (các đặc trưng người gửi).

Hai đầu vào có hình thù khác nhau, đầu vào nội dung là một ma trận còn đầu vào đặc trưng người dùng là một vector, nên không thể trực tiếp kết hợp chúng với nhau. Tuy nhiên, ta có thể dùng một hoặc một số lớp mạng nơ-ron để biến đổi đầu vào nội dung thành vector. Đối với nội dung văn bản, các lớp mạng nơ-ron hồi quy (recurrent) là lựa chọn phù hợp. Ở đây, cấu trúc LSTM được sử dụng. Giống như các lớp mạng hồi quy khác, lớp LSTM nhận một vector từ ngữ làm đầu vào và sản sinh ra một giá trị số thực cho mỗi đơn vị (unit) ở đầu ra của lớp. Như vậy, đầu ra của lớp LSTM có dạng một

vector số thực với độ dài bằng với số lượng đơn vị LSTM bên trong nó. Mỗi vector từ ngữ sẽ kích hoạt sự biến đổi bên trong của các đơn vị LSTM. Sau khi đọc hết các vector từ ngữ của nội dung đầu vào, trạng thái của vector đầu ra chính là kết quả dự đoán của lớp LSTM. Thông thường, vector đầu ra của lớp LSTM được dùng làm đầu vào cho một mạng nơ-ron truyền thẳng ví dụ như MLP, CNN hoặc perceptron. Trong cấu trúc mạng được đề xuất ở đây, đầu ra của lớp LSTM được ghép nối với vector đặc trưng người gửi, tạo thành một vector chứa đựng cả thông tin từ nội dung thư và đặc tính của người gửi thư. Đầu ra của lớp mạng hồi quy là một tập hợp các giá trị số thực trong một thể thống nhất hoàn chỉnh, đại diện cho nội dung của bức thư, chứ không phải là một kiểu dữ liệu dạng chuỗi thay đổi theo thời gian như đầu vào của nó. Từ lý do này, cấu trúc mạng truyền thẳng là phù hợp để xử lý vector kết hợp nói trên.

4.2.6. Huấn luyện mạng nơ-ron

Thuật toán tối ưu đóng vai trò quan trọng cho sự thành công của việc huấn luyện mạng nơ-ron. Mỗi thuật toán và biến thể của chúng đều dành sự ưu tiên khác nhau cho những kiểu đặc trưng khác nhau, cũng như có những cơ chế di chuyển khác nhau trong không gian tìm kiếm để phục vụ mục đích cuối cùng là tìm ra vị trí tối ưu, hay nói cách khác là tìm ra tập trọng số tối ưu cho mạng nơ-ron. Mặc dù cách tiếp cận cơ bản trong việc huấn luyện mạng nơ-ron là xuống dốc theo phương pháp lan truyền ngược sai số, nhiều biến thể khác nhau của phương pháp xuống dốc đã được đưa ra để tương thích với các cấu trúc mạng và loại dữ liệu khác nhau.

Thuật toán Adagrad [47] tính toán và ghi nhớ tốc độ học riêng biệt cho mỗi nơ-ron trong mạng. Thuật toán này giảm dần tốc độ học của những nơ-ron phổ biến (được cập nhật thường xuyên) và tăng dần tốc độ học của những nơ-ron hiếm gặp (ít khi được cập nhật). Điều này giúp tăng sức ảnh hưởng cho các đặc trưng hiếm, bởi vì trong nhiều trường hợp, các đặc trưng ít xuất hiện thường có sức ảnh hưởng lớn đến kết quả dự đoán.

Thuật toán RProp [1] là một biến thể khác của thuật toán xuống dốc, trong đó, các trọng số được cập nhật dựa vào dấu của đạo hàm mà không dựa vào độ lớn của đạo hàm. Độ lớn của thay đổi được quyết định bởi một giá trị bước nhảy dành riêng cho từng trọng số và được biến đổi thích ứng với trọng số đó. Thuật toán RMSProp [53] là

một phiên bản được cải tiến từ RProp để phù hợp với việc huấn luyện theo loạt (mini-batch). Thuật toán này giải quyết một vấn đề của RProp khi đạo hàm của hai loạt dữ liệu huấn luyện liên tiếp có cách biệt lớn về giá trị. Điều này dẫn đến sự thay đổi trọng số đột ngột bởi vì giá trị bước nhảy được nhân lên rất lớn.

Thuật toán Adam [73] cũng hướng tới việc ghi nhớ tốc độ học riêng biệt và thích nghi cho từng trọng số (tương ứng với từng nơ-ron), nhưng khác với Adagrad, việc này được thực hiện dựa trên cơ chế quán tính. Với cơ chế này, tốc độ học của một vòng huấn luyện được tổng hợp từ các giá trị tốc độ học trong một vòng huấn luyện gần nhất trong quá khứ.

Cross-entropy (hay còn gọi là *log loss* hoặc *logistic loss*) là sự lựa chọn mặc định về hàm tổn thất dành cho các bài toán phân loại đa lớp. Hàm tổn thất này đánh giá mức độ tự tin của kết quả dự đoán đưa ra bởi một mô hình. Một mô hình phân loại 5 lớp với đầu ra softmax sẽ có đầu ra là 5 giá trị số thực, mỗi giá trị là kết quả dự đoán của mô hình dành cho một lớp trong tổng số 5 lớp. Kết quả dự đoán cuối cùng là lớp có giá trị dự đoán cao nhất trong 5 giá trị đó. Tuy nhiên, sự cách biệt giữa giá trị dự đoán cho lớp chính xác và 4 lớp còn lại càng lớn sẽ cho thấy độ tự tin càng cao của mô hình. Giá trị cross-entropy là thấp nhất khi kết quả dự đoán cho lớp chính xác là giá trị tối đa và kết quả dự đoán cho các lớp không chính xác là tối thiểu. Hay nói cách khác, nếu kết quả dự đoán hoàn toàn trùng khớp với nhãn được gán thì cross-entropy có giá trị là 0 dành cho lần dự đoán đó. Hàm tổn thất này không đặt giả thiết về mối quan hệ tương đối giữa các lớp (nhãn).

Ngoài ra, những hàm tổn thất thường dùng cho các bài toán hồi quy, ví dụ như *trung bình của bình phương lỗi* (MSE) hay trung bình giá trị tuyệt đối của lỗi (MAE) cũng có thể được áp dụng cho bài toán phân loại. Khi đó, giả thiết rằng các lớp có mối quan hệ tương đối được áp dụng. Điều này có nghĩa là khác biệt giữa lớp 1 và 2 được coi là nhỏ hơn khác biệt giữa lớp 1 và 5. Trong ngữ cảnh của nghiên cứu này, sự khác biệt giữa một bức thư có nhãn ‘xóa’ với bức thư có nhãn ‘đọc không quan trọng’ được coi là nhỏ hơn sự khác biệt giữa bức thư có nhãn ‘xóa’ với bức thư có nhãn ‘trả lời gấp’.

Thuật ngữ *over-fitting* nói về một hiện tượng không mong muốn khi mà một mô hình học máy mô phỏng chính xác tập dữ liệu huấn luyện nhưng không mô phỏng chính xác

tập dữ liệu cần dự đoán trên thực tế. Vì tập huấn luyện chỉ là một phần nhỏ trong toàn bộ dữ liệu thực tế nên ta thường gọi tập huấn luyện là *tập mẫu*. Giả thiết cơ bản nhất của học máy là tập huấn luyện lý tưởng có khả năng *đại diện* cho toàn bộ dữ liệu thực tế vì nó mang đầy đủ đặc trưng và tính chất của toàn bộ dữ liệu. Tuy nhiên, trên thực tế, sự khác biệt giữa tập mẫu và toàn bộ dữ liệu (tạm gọi là phần *nhiễu*) là không nhỏ. Khi over-fitting xảy ra, mô hình học máy đã dung nạp cả phần nhiễu vào nó, khiến cho kết quả dự đoán trên dữ liệu thực tế trở nên kém chính xác. Ta gọi một mô hình bị over-fitting là thiếu tính khái quát bởi vì nó quá phù hợp với một phần dữ liệu thiểu số mà lại sai khác với đa số dữ liệu còn lại. Một số cách để tránh hiện tượng over-fitting đó là cải thiện tính đại diện của tập huấn luyện hoặc dừng việc huấn luyện sớm (còn gọi là kỹ thuật *early stopping*). Trong mạng nơ-ron, kỹ thuật *dropout* [58] cũng là một cách hiệu quả để tránh hiện tượng over-fitting. Kỹ thuật này loại bỏ một phần nơ-ron một cách ngẫu nhiên trong mỗi vòng huấn luyện. Mục đích của dropout là để tránh trường hợp một tập hợp nhỏ nơ-ron có sức ảnh hưởng áp đảo so với các nơ-ron khác trên cùng một lớp mạng.

4.3. XẾP HẠNG THƯ ĐIỆN TỬ DỰA TRÊN SPAMASSASSIN

Trong Chương 3, các phương pháp phân loại đa lớp bằng cách kết hợp nhiều máy phân loại nhị phân dưới dạng các tập luật SpamAssassin đã được giới thiệu và áp dụng vào bài toán dự đoán hành động người dùng với 3 hành động. Phương pháp phân loại đa lớp nói trên có thể được mở rộng để giải quyết bài toán xếp hạng thư điện tử với 5 mức độ ưu tiên. Trong số các phương pháp dự đoán hành động người dùng được đề xuất trong luận án, phương pháp UAP_3 có kết quả cao khi thử nghiệm trên tập dữ liệu tiếng Việt. Phương pháp UAP_3 là phương pháp dự đoán hành động người dùng dựa trên việc kết hợp nhiều máy phân loại nhị phân, trong đó mỗi máy phân loại nhị phân là một tập luật SpamAssassin được xây dựng theo phương pháp SD_1.

Phương pháp UAP_3 sẽ được chỉnh sửa để đưa vào thí nghiệm so sánh trong chương này để tìm hiểu hiệu quả khi mở rộng phương pháp này đối với bài toán xếp hạng thư điện tử có 5 mức độ ưu tiên. Sau đây, quy trình xây dựng mô hình xếp hạng thư điện tử dựa trên phương pháp UAP_3, ký hiệu là phương pháp EP_1, sẽ được trình bày tóm tắt.

4.3.1. Xây dựng máy phân loại nhị phân

Dữ liệu thư điện tử được biểu diễn dưới dạng vector nhị phân. Hai tập từ vựng là tập V_s được trích xuất từ tiêu đề thư và tập V_b được trích xuất từ nội dung thư. Mỗi bức thư được biểu diễn bằng một vector *x* với độ dài bằng tổng độ dài của hai tập từ vựng nói trên. Mỗi từ ngữ xuất hiện trong tiêu đề hoặc nội dung bức thư là một thuộc tính và được biểu diễn trong vector *x* tại một vị trí xác định. Các thuộc tính từ tiêu đề thư được biểu diễn ở $|V_s|$ vị trí đầu của vector *x* và các thuộc tính từ nội dung thư được biểu diễn ở $|V_b|$ vị trí tiếp theo. Giá trị 1 thể hiện sự có mặt của một thuộc tính và giá trị 0 thể hiện thuộc tính không xuất hiện trong bức thư.

Mô hình mạng nơ-ron với hai lớp ẩn trong hình 2.3 được huấn luyện trên một tập dữ liệu bao gồm các bức thư thuộc hai lớp khác nhau. Khi áp dụng mô hình này với bài toán phát hiện thư rác, tập dữ liệu sẽ bao gồm thư rác và thư hợp lệ. Khi áp dụng mô hình với các phương án như OVA, OVO… thì hai lớp dữ liệu trong tập huấn luyện phụ thuộc vào phương án phân loại đa lớp được lựa chọn. Mô hình mạng nơ-ron được thiết kế sao cho chỉ có α đặc trưng được lựa chọn từ toàn bộ các thuộc tính trong vector đầu vào x và trọng số của các đặc trưng tương ứng trong lớp ẩn thứ hai, có tập trọng số là w, có thể được sử dụng để tính toán ra điểm số của luật SpamAssassin tương ứng với đặc trưng đó. Như đã đề cập ở trên, một đặc trưng được thể hiện bởi một từ ngữ và vị trí xuất hiện của từ ngữ đó (trong tiêu đề hoặc trong nội dung thư). Các đặc trưng có vị trí xuất hiện ở tiêu đề sẽ được chuyển thành luật *header* và các đặc trưng xuất hiện trong nội dung sẽ được chuyển thành luật *body*.

Mô hình mạng nơ-ron nói trên được huấn luyện bằng một biến thể của thuật toán SGD đã được trình trong phần 2.2 của luận án. Sau khi hoàn thành huấn luyện, để áp dụng tập luật được sinh ra trên nền tảng SpamAssassin, trọng số của các đặc trưng được chuyển hóa thành điểm số của luật bằng công thức (1.10). Các đặc trưng và tập trọng số cũng có thể được trích xuất để xây dựng một mô hình *perceptron* như được mô tả trong [17], có khả năng phân loại nhị phân tương đương với tập luật SpamAssassin nói trên, mà không cần thay đổi các trọng số. Phương án này được chọn để thực hiện thí nghiệm trong chương này bởi vì sự tiện dụng khi không cần tích hợp hệ thống SpamAssassin vào chương trình thí nghiệm.

4.3.2. Các phương án phân loại đa lớp

Có ba cách phổ biến để xây dựng máy phân loại đa lớp từ nhiều máy phân loại nhị phân là OVA, OVO và DAG. Trong đó, phương án OVO có ba biến thể là OVO-MS, OVO-MV và OVO-MC. Các phương án này đã được trình bày chi tiết trong Chương 3 của luận án. Phần này xin được tóm tắt lại các điểm chính.

Với bài toán phân loại 5 lớp, phương án OVA cần xây dựng là 5 máy phân loại nhị phân. Số lượng cần xây dựng cho phương án OVO và DAG là 10 máy phân loại, tương đương với tổ hợp chập 2 của 5 phần tử bởi vì ta cần một máy phân loại cho mỗi hai lớp dữ liệu. Ở phương án OVA, mỗi máy phân loại M_i được gắn với lớp C_i. Khi dự đoán của máy phân loại M_i (một giá trị số thực) đối với bức thư *m* là cao nhất, ta khẳng định bức thư m thuộc về lớp C_i. Ở phương án OVO, các máy phân loại được ký hiệu là $M_{i,j}$ để thể hiện đó là máy phân loại để phân biệt một bức thư thuộc về lớp C_i hay C_j. Kết quả của tất cả 10 máy phân loại được tổng hợp lại. Trong trường hợp cụ thể của bài toán phân loại 5 lớp, luôn có 4 máy phân loại có kết quả dự đoán liên quan đến lớp C_i. Ví dụ, lớp C_1 sẽ có các máy phân loại liên quan là $M_{1,2}$, $M_{1,3}$, $M_{1,4}$ và $M_{1,5}$. Giả sử các máy phân loại này lần lượt dự đoán các kết quả 0.1, 0.2, -0.3, -0.1 thì điểm số cuối cùng cho lớp C_1 theo phương án OVO-MS sẽ là tổng của các giá trị nói trên và là -0.1. Cuối cùng, bức thư *m* sẽ được kết luận là thuộc về lớp C_i có tổng điểm cao nhất. Chi tiết về các phương án tổng hợp kết quả được trình bày trong phần 3.3.2. Phương án DAG tìm ra kết quả dự đoán theo một cây quyết định nhị phân. Trước tiên, máy phân loại $M_{1,n}$ được sử dụng trên bức thư *m* với C_1 và C_n là hai lớp được xem là có sự khác biệt lớn nhất. Chẳng hạn, khi lớp C_1 được chọn, máy phân loại $M_{1,n-1}$ sẽ được sử dụng trên bức thư *m*. Khoảng cách giữa hai lớp tham gia phân loại dần được thu hẹp cho đến khi đó là hai lớp liền kề nhau trong tập hợp các lớp cần phân loại. Kết quả phân loại được quyết định bởi máy phân loại cuối cùng được thực thi trên bức thư *m*. Thuật toán trong hình 3.7 mô tả phương pháp dự đoán DAG một cách cụ thể. Cách phân loại này có ưu điểm tăng tốc xử lý cho bài toán phân loại đa lớp bởi vì số lượng máy phân loại cần được thực thi nhỏ hơn so với hai phương án OVA và OVO.

4.4. THỰC NGHIỆM

4.4.1. Tiêu chí đánh giá

Đối với mỗi lần thí nghiệm mô hình, tập dữ liệu được chia thành hai tập *huấn luyện* và *thử nghiệm* với tỷ lệ 90% cho huấn luyện và 10% cho thử nghiệm. Để tiến hành kiểm chứng chéo, tập dữ liệu được trộn ngẫu nhiên rồi chia ra thành 10 phần sao cho các bức thư thuộc về 5 nhãn được phân chia đồng đều cho các phần dữ liệu, nhằm tăng tính khách quan cho kết quả thí nghiệm. Mỗi thí nghiệm được thực hiện 10 lần. Từng phần dữ liệu được lần lượt sử dụng làm dữ liệu thử nghiệm, còn 9 phần dữ liệu còn lại được sử dụng để huấn luyện. Như vậy, sau 10 lần lặp lại thí nghiệm, toàn bộ tập dữ liệu đều đã được sử dụng vào việc thử nghiệm mô hình. Kết quả thí nghiệm cuối cùng được tính là trung bình cộng kết quả của 10 lần thí nghiệm.

Để đánh giá kết quả thí nghiệm, ba tiêu chí đánh giá là accuracy, cross-entropy và điểm số F_1 vĩ mô (macro F_1 score) [41] được lựa chọn sử dụng. Những tiêu chí này phù hợp với bài toán phân loại đa lớp không đặt giả thiết về quan hệ tương đối giữa các lớp. Tiêu chí accuracy là tổng số dự đoán đúng trên toàn bộ số lần dự đoán. Tiêu chí *cross-entropy* (4.2), gọi tắt là CCE, dùng để đo sự sai khác giữa kết quả dự đoán $\hat{y}$ và kết quả mục tiêu y. Khi đánh giá mô hình, kết quả dự đoán càng gần với nhãn đã gán thì giá trị cross-entropy càng thấp, cho thấy mô hình càng tự tin với kết quả dự đoán mà nó đưa ra.

$$H(y, \hat{y}) = -\sum y_i * \ln(\hat{y}_i) \tag{4.2}$$

$$P_m = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fp_i}}{l} \tag{4.3}$$

$$R_m = \frac{\sum_{i=1}^{l} \frac{tp_i}{tp_i + fn_i}}{l} \tag{4.4}$$

Tiêu chí F_1 vĩ mô (4.5) được tính toán từ hai tiêu chí *precision vĩ mô* (4.3) và *recall vĩ mô* (4.4). Vĩ mô (macro) ở đây là chỉ cách tính tiêu chí cho từng nhãn và lấy trung bình cộng để ra kết quả. Tiêu chí F_1 vi mô (micro F_1 score) tính tiêu chí cho từng mẫu và lấy trung bình cộng để ra kết quả. Tiêu chí F_1 vi mô có cùng giá trị với điểm số *accuracy*. Giống như điểm số F_1 dành cho bài toán phân loại nhị phân, tiêu chí F_1 vĩ mô

là sự cân bằng giữa *recall* – độ hoàn chỉnh của kết quả dự đoán – và *precision* – độ tin cậy của kết quả dự đoán.

$$\text{Macro } F_1 = \frac{2 * P_m * R_m}{P_m + R_m} \tag{4.5}$$

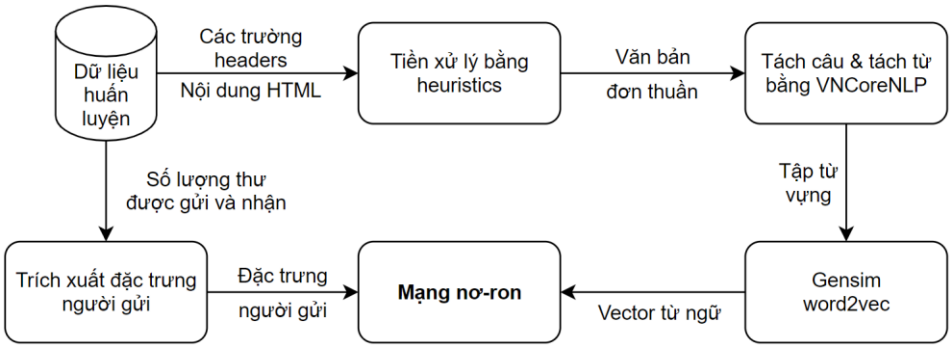
4.4.2. So sánh các thuật toán tối ưu mạng nơ-ron (thí nghiệm 1)

Mục tiêu của thí nghiệm này là so sánh những thuật toán huấn luyện mạng nơ-ron. Thí nghiệm được thực hiện đối với ba thuật toán phổ biến: Adam, RMSProp và Adagrad. Những thuật toán này là các biến thể của thuật toán cổ điển xuống dốc ngẫu nhiên (SGD). Mỗi thuật toán có những tham số điều chỉnh khác nhau. Trong thí nghiệm này, những giá trị tham số gợi ý bởi bài báo công bố của mỗi thuật toán đã được sử dụng. Tham số α dùng để đặt tốc độ học khởi tạo. Tham số ε là một số thực dương có giá trị nhỏ, thường được đặt trong khoảng từ 10^{-8} đến 10^{-7}, dùng để tránh các phép chia cho 0. Giá trị gợi ý cho tham số α của cả ba thuật toán là 0.001. Mỗi thuật toán còn có những tham số điều chỉnh riêng biệt. Thuật toán Adam có tham số β_1 được đặt giá trị 0.9 và β_2 được đặt giá trị 0.999. Với thuật toán Adagrad, giá trị tích lũy ban đầu (initial accumulator) được gợi ý là 0.1. Thuật toán RMSProp được đặt giá trị 0.9 cho *rho* – hệ số chiết khấu (discounting factor) – và giá trị 0 cho quán tính ban đầu (intial momentum).

Với cùng cấu trúc mạng nơ-ron (Hình 4.1), các thí nghiệm riêng biệt được thực hiện với word embedding huấn luyện sẵn và word embedding trực tuyến, với số chiều không gian cùng là 128. Kết quả của thí nghiệm 1 được trình bày trong Bảng 4.1 với ba tiêu chí đánh giá đã đề cập. Quy trình tiền xử lý dữ liệu dành cho phương pháp đề xuất được minh họa trong Hình 4.2.

Sự khác nhau trong quá trình tiền xử lý dữ liệu giữa hai cấu hình thí nghiệm với đặc trưng nội dung dạng word embedding trực tuyến và word embedding huấn luyện sẵn cần được làm rõ. Trước tiên, lớp mạng Embedding của mạng nơ-ron sâu (Hình 4.1) có bộ trọng số được hình thành từ *n* vector từ ngữ có độ dài *m*, hợp lại thành một ma trận có kích thước $n \times m$, trong đó *n* là kích thước của tập từ vựng và *m* là độ dài vector từ ngữ (*m* có giá trị 128 hoặc 300 trong thí nghiệm của chương này). Đầu vào của lớp Embedding là một văn bản được biểu diễn dưới dạng một chuỗi các chỉ mục (vị trí) của

các từ ngữ trong tập từ vựng. Văn bản đầu vào này có độ dài cố định được chọn là l = 300. Dựa trên chuỗi đầu vào là chỉ mục của l từ ngữ, lớp Embedding lấy các vector từ ngữ tương ứng từ bộ trọng số của nó để tạo ra một ma trận có kích thước $l \times m$, là biểu diễn word embedding của văn bản đầu vào.



Hình 4.2: Quy trình tiền xử lý dữ liệu trong phương pháp xếp hạng thư điện tử dựa trên học sâu.

Đầu ra của thuật toán word2vec trong bộ công cụ Gensim bao gồm một tập từ vựng và bộ vector từ ngữ tương ứng. Bộ vector từ ngữ mà thuật toán word2vec sinh ra là một ma trận số thực có kích thước và chức năng trùng khớp với bộ trọng số của lớp Embedding. Vì vậy, bộ vector từ ngữ word2vec được sử dụng làm trọng số của lớp Embedding trong phương án word embedding huấn luyện sẵn. Thuật toán huấn luyện cũng được cấu hình để không cập nhật trọng số của lớp Embedding. Trái lại, với cấu hình word embedding trực tuyến, lớp mạng Embedding sẽ có bộ trọng số được khởi tạo ngẫu nhiên và được cập nhật trong quá trình huấn luyện mạng. Trong trường hợp này, lớp Embedding không sử dụng các vector từ ngữ word2vec mà chỉ sử dụng chỉ mục của các từ ngữ trong tập từ vựng để lớp mạng này tương thích với đầu vào của mô hình.

Bảng 4.1: Kết quả so sánh ba thuật toán huấn luyện mạng nơ-ron

Thuật toán huấn luyện	**Accuracy**		**Macro F_1**		**CCE**	
	(a)	(b)	(a)	(b)	(a)	(b)
Adam	0.6641	0.9115	0.3769	**0.8641**	6.6992	**0.6650**
Adagrad	0.5209	0.6448	0.1374	0.5090	**1.7875**	1.0729
RMSProp	**0.7134**	**0.9126**	**0.5014**	0.8632	5.9510	0.7260
(a) Word embedding trực tuyến, m = 128 (b) Word embedding huấn luyện sẵn (word2vec), m = 128						

4.4.3. So sánh các phương án word embedding (thí nghiệm 2)

Thuật toán tối ưu RMSProp đã thể hiện kết quả tốt nhất trong thí nghiệm 1. Tuy nhiên, tác dụng của kích thước vector từ ngữ đối với tập dữ liệu trong bài báo này vẫn chưa được xác định. Chính vì vậy trong thí nghiệm này, mô hình trong Hình 4.1 được huấn luyện với hai kích thước vector từ ngữ là 128 và 300, so sánh giữa phương án word embedding huấn luyện sẵn và word embedding trực tuyến. RMSProp được chọn làm thuật toán huấn luyện chung cho thí nghiệm 2. Bộ vector từ ngữ huấn luyện sẵn được sinh ra từ dữ liệu với thuật toán *word2vec* từ bộ công cụ xử lý ngôn ngữ tự nhiên Gensim. Trọng số của bộ vector từ ngữ này được dùng làm bộ trọng số của lớp Embedding và lớp này được cài đặt để các trọng số không thay đổi khi huấn luyện, hay nói cách khác là không tham gia vào quá trình huấn luyện mạng nơ-ron.

Bảng 4.2: Kết quả thí nghiệm so sánh các cấu hình word embedding khác nhau.

Cấu hình embedding	**Accuracy**	**Macro F1**	**CCE**
Word2vec, m = 128	0.9126	0.8632	0.7260
Word2vec, m = 300	0.9185	0.8764	0.7146
Trực tuyến, m = 128	0.7134	0.5014	5.9510
Trực tuyến, m = 300	0.7900	0.5918	4.2800

4.4.4. So sánh một số phương pháp xếp hạng thư điện tử (thí nghiệm 3)

Thí nghiệm 3 so sánh hiệu quả của ba phương pháp xếp hạng thư điện tử sau:

- Thứ nhất là phương pháp xếp hạng thư điện tử bằng mô hình phân loại đa lớp với máy phân loại nhị phân là tập luật SpamAssassin được sinh bằng phương pháp SD_1. Phương pháp này được đặt tên là EP_1. Phương pháp EP_1 đã được trình bày ở phần 4.3 của chương này. Phương pháp SD_1 đã được trình bày trong Chương 2 của luận án.
- Thứ hai là phương pháp xếp hạng thư điện tử dựa trên mô hình học sâu được trình bày trong phần 4.2 của Chương 4. Phương pháp này sẽ được ký hiệu là EP_2.
- Thứ ba là phương pháp xếp hạng thư điện tử được giới thiệu trong [49]. Luận án tạm đặt tên phương pháp này là YooEP dựa theo tên của tác giả đề xuất phương pháp. Kết quả của phương pháp YooEP được tái tạo với tập dữ liệu xếp hạng thư điện tử tiếng Việt.

Để thực hiện thí nghiệm này, các bức thư được vector hóa với định dạng TF-IDF. Để tính toán các vector TF-IDF, nội dung thư được tách từ theo cùng một cách như đã mô tả ở phần tiền xử lý dữ liệu. Để sự so sánh được nhất quán, phương án phân loại đa lớp được chọn là OVA bởi vì OVA có cùng nguyên lý hoạt động với lớp đầu ra softmax của mô hình trong phương pháp EP_2. Phương pháp YooEP đề xuất mô hình phân loại đa lớp dựa trên máy phân loại SVM và đặc trưng TF-IDF. Phương pháp EP_2 trong thí nghiệm so sánh sử dụng cấu hình tốt nhất dựa theo kết quả của thí nghiệm 1 và thí nghiệm 2, đó là sử dụng thuật toán huấn luyện RMSProp và đặc trưng vector từ ngữ word2vec độ dài 300.

Bảng 4.3: So sánh phương pháp EP_2 với phương pháp EP_1 và YooEP [49]

Phương pháp	**Accuracy**	**Macro F_1**	**CCE**
OVA-EP_1	0.8219	0.7757	1.0036
EP_2*, word2vec, m = 300	**0.9185**	**0.8764**	0.7146
YooEP-OVA, epoch=50	0.7137	0.4529	0.7893
YooEP-OVA, epoch=100	0.7847	0.5550	0.6161
YooEP-OVA, epoch=150	0.8225	0.6360	**0.5207**
* EP_2 dùng thuật toán huấn luyện RMSProp, số lượng epoch = 15			

Bảng 4.3 tổng hợp kết quả thí nghiệm so sánh 3 phương pháp xếp hạng thư điện tử khác nhau, trong đó phương án YooEP-OVA được báo cáo kết quả với ba cấu hình huấn luyện. Các phương pháp trong thí nghiệm đều có mục tiêu phân loại thư thành 5 mức độ ưu tiên và được thử nghiệm trên cùng tập dữ liệu xếp hạng thư điện tử tiếng Việt đã được miêu tả trong bảng 1.4. Với sự bổ sung các thuộc tính xã hội, phương pháp EP_2 có hiệu quả cao hơn đáng kể so với hai phương án còn lại. Điều này thể hiện qua giá trị của các tiêu chí đánh giá được sử dụng. Phương pháp EP_2 đạt được chỉ số *accuracy* và *macro* F_1 cao nhất trong các phương pháp được so sánh. Điểm số F_1 được suy ra từ các tiêu chí *recall* và *precision*. Tiêu chí *macro* F_1 là trung bình cộng của điểm số F_1 của các lớp riêng biệt. Như vậy, các lớp có ít mẫu trong tập dữ liệu, chẳng hạn như lớp *trả lời không gấp* (970 bức thư) và lớp *trả lời gấp* (655 bức thư) trong bày toán được xét, sẽ có ảnh hưởng cao hơn tới giá trị *macro* F_1. Giá trị *macro* F_1 thấp hơn đáng kể so với tiêu chí *accuracy* cho thấy hiệu quả phân loại đối với các lớp có ít dữ liệu thấp hơn so

với các lớp có số lượng dữ liệu lớn, chẳng hạn như lớp thư đọc quan trọng với 5,787 bức thư.

Từ kết quả thí nghiệm, phương pháp phân loại đa lớp dựa trên nền tảng SpamAssassin (phương pháp OVA-EP_1) không cho hiệu quả cao như khi áp dụng cho bài toán dự đoán hành động người dùng. Để thu được kết quả này, có thể kể đến một số lý do. Tuy nền tảng SpamAssassin cho phép triển khai dễ dàng mô hình phân loại thư điện tử dựa trên nội dung, nguyên lý phân loại của SpamAssassin còn đơn giản vì tập luật SpamAssassin có bản chất là máy phân loại tuyến tính. Các đặc trưng nội dung trong SpamAssassin được biểu diễn theo phương pháp túi từ truyền thống nên lượng thông tin hữu ích cho mô hình phân loại được giữ lại không nhiều. Hơn nữa, cơ chế hoạt động của SpamAssassin không cho phép bổ sung các đặc trưng xã hội vào bức thư. Những đặc trưng xã hội giúp tăng lượng thông tin đầu vào cho các mô hình phân loại và đã cho thấy hiệu quả trong nhiều nghiên cứu về thư điện tử [26, 40, 54].

4.5. KẾT LUẬN CHƯƠNG 4

Chương này đã trình bày các đề xuất để giải quyết bài toán xếp hạng thư điện tử, một hướng nghiên cứu mới và có ý nghĩa to lớn đối với người dùng thư điện tử trong kỷ nguyên bùng nổ thông tin. Luận án xin trình bày phương án ứng dụng phương pháp được đề xuất trong chương này trên hệ thống thư điện tử thực tế. Như đã đề cập trong phần 3.6, một hệ thống xử lý thư điện tử thường bao gồm ba thành phần chính là MTA, MDA và phần mềm khách. Ở đây phần mềm khách trên nền tảng web, gọi là webmail, được lựa chọn để thuận tiện cho việc trình bày bởi vì phần mềm khách trên mọi nền tảng đều có vai trò và nhiệm vụ tương tự nhau. Trước hết, cần phải xây dựng mới một phần mềm chạy trên máy chủ thư điện tử dựa trên mô hình học sâu đã trình bày và bộ trọng số đã được huấn luyện. Trong phương án áp dụng mô hình xếp hạng thư điện tử, MDA sẽ phải đảm nhiệm thêm nhiệm vụ thực thi phần mềm xếp hạng thư điện tử, lấy kết quả trả về và ghi kết quả vào bức thư dưới dạng header. Phần mềm webmail cần phải được tùy biến để đọc kết quả phân loại từ header của các bức thư nhằm sắp xếp và hiển thị các bức thư theo thứ tự từ quan trọng đến không quan trọng trên giao diện web cho người dùng.

Nội dung trình bày trong chương này được tổng hợp từ kết quả đã được công bố trong các công trình nghiên cứu số 1, số 2 và số 3 của tác giả. Phương pháp xếp hạng thư điện tử dựa trên phân loại đa lớp EP_1 là sự kết hợp giữa mô hình dự đoán hành động người dùng trình bày trong nghiên cứu số 1 và phương pháp sinh tập luật SpamAssassin dựa trên mạng nơ-ron từ nghiên cứu số 2. Phương pháp xếp hạng thư điện tử dựa trên mô hình học sâu EP_2 đã được công bố trong nghiên cứu số 3. Hai phương pháp trên cùng một phương pháp xếp hạng thư điện tử dựa trên phân loại từ nghiên cứu [49] đã được so sánh trong thí nghiệm. Kết quả thí nghiệm cho thấy phương pháp EP_2 đạt hiệu quả tốt trên tập dữ liệu xếp hạng thư điện tử tiếng Việt khi đánh giá bằng tiêu chí *accuracy* và *macro* F_1. Kết quả nói trên xác nhận hiệu quả của các lựa chọn khi thiết kế và huấn luyện mô hình học sâu. Sự kết hợp đặc trưng nội dung với kỹ thuật word2vec và các đặc trưng xã hội khi biểu diễn các bức thư đã cung cấp tập đặc trưng giàu thông tin làm đầu vào cho mô hình phân loại. Cấu trúc mạng nơ-ron hồi quy LSTM với sở trường ghi nhớ những phụ thuộc cách xa nhau trong chuỗi đầu vào đã thể hiện hiệu quả tốt đối với nội dung thư điện tử trong đó ý nghĩa của văn bản không chỉ thể hiện ở các từ ngữ và số lượng của chúng, mà còn ở vị trí tương đối của các từ ngữ.

KẾT LUẬN

Xác định thứ tự ưu tiên của thư điện tử là một hướng giải quyết tình trạng quá tải thư điện tử, một vấn đề đang ngày càng trở nên cấp thiết. Luận án tập trung nghiên cứu phương pháp xác định thứ tự ưu tiên của thư điện tử theo 03 hướng tiếp cận chính là lọc thư rác, dự đoán hành động người dùng và xếp hạng thư điện tử. Luận án thể hiện 03 đóng góp chính, một là đề xuất phương pháp tự động sinh tập luật mới cho SpamAssassin, trong đó bước lựa chọn luật và bước xác định trọng số của luật được tiến hành đồng thời. Hai là đề xuất phương pháp sử dụng nền tảng SpamAssassin kết hợp với các mô hình phân loại đa lớp để gợi ý hành động người dùng. Ba là đề xuất phương pháp học sâu để xếp hạng thư điện tử theo 5 mức độ ưu tiên khác nhau, sử dụng word embedding để biểu diễn nội dung thư kết hợp với đặc trưng mạng xã hội. Ngoài ra, để thực hiện thí nghiệm cho các đề xuất trong 03 hướng tiếp cận nói trên, luận án đã thu thập và xây dựng tập dữ liệu thư điện tửu tiếng Việt.

Đóng góp thứ nhất cho bài toán lọc thư rác của luận án là đề xuất phương pháp xây dựng tập luật SpamAssassin dựa trên mạng nơ-ron. Phương pháp có hiệu quả dự đoán cải thiện hơn so với những phương pháp cũ. Thông qua tổng quan tài liệu, luận án nhận thấy các phương pháp xây dựng tập luật SpamAssassin dựa trên học máy đều thực hiện tách rời hai khâu lựa chọn đặc trưng và huấn luyện trọng số. Cách làm này dẫn đến một hạn chế đó là chưa kiểm chứng được hiệu quả của tập đặc trưng được chọn trên dữ liệu bởi vì chỉ có một tập đặc trưng duy nhất được lựa chọn và không được so sánh với các tập đặc trưng tiềm năng khác. Mô hình mạng nơ-ron được đề xuất trong đóng góp thứ nhất có mục tiêu giải quyết vấn đề nói trên. Mô hình gồm hai lớp ẩn, một lớp có chức năng lựa chọn đặc trưng và lớp còn lại có chức năng điều chỉnh trọng số của đặc trưng, từ đó hợp nhất hai khâu lựa chọn luật và gán điểm số vốn tách rời trong các phương pháp sinh tập luật SpamAssassin trước đó, giúp nâng cao chất lượng của tập luật được xây dựng.

Đóng góp thứ hai cho bài toán lọc thư rác của luận án là một phương pháp khác để sinh tập luật lọc thư rác cho SpamAssassin, hướng tới mở rộng tác vụ sinh tập luật từ bài toán tối ưu đơn mục tiêu thành bài toán tối ưu đa mục tiêu, chú trọng cải thiện khâu

gán điểm số cho tập luật. Phương pháp này giải quyết vấn đề quan trọng của bài toán sinh tập luật SpamAssassin đó là sự cân bằng giữa hai tiêu chí đối nghịch *recall* và *FAR*.

Với bài toán dự đoán hành động người dùng, luận án đã đề xuất phương pháp giải quyết với mô hình phân loại đa lớp trên nền tảng SpamAssassin. Kết quả từ các nghiên cứu đã công bố số 1, số 2 và số 5 đã được tổng hợp để đề xuất và cải tiến phương pháp dự đoán hành động. Luận án đã ứng dụng cách kỹ thuật khác nhau để kết hợp nhiều tập luật SpamAssassin thành máy phân loại đa lớp có tác dụng dự đoán hành động cho người dùng thư điện tử. Phương pháp này có tính ứng dụng cao trên thực tế bởi vì sự phổ biến của hệ thống SpamAssassin và tốc độ xử lý nhanh của cơ chế luật có trọng số. Luận án cũng trình bày hai phương án nhằm cải thiện hiệu quả của mô hình dự đoán hành động nói trên dựa trên cải thiện hiệu quả của các máy phân loại nhị phân thành phần, từ đó nâng cao hiệu quả của máy phân loại đa lớp. Cách thứ nhất là ứng dụng thêm luật ham cho tập luật SpamAssassin. Cách thứ hai là ứng dụng phương pháp sinh tập luật SpamAssassin dựa trên mạng nơ-ron từ nghiên cứu đã công bố số 2. Thí nghiệm so sánh cho thấy phương án thứ nhất giúp giảm tỷ lệ gợi ý nhầm đối với hành động xóa thư trong khi phương án thứ hai giúp tăng độ chính xác chung của các gợi ý.

Về bài toán xếp hạng thư điện tử, đóng góp của luận án là một mô hình phân loại dựa trên học sâu để giải quyết bài toán xếp hạng thư điện tử, là kết quả đã được công bố trong công trình nghiên cứu đã công bố số 3. Mô hình được đề xuất không chỉ tích hợp các kỹ thuật học sâu, trong đó nổi bật là cấu trúc mạng LSTM, mà còn sử dụng bộ đặc trưng nội dung kết hợp với đặc trưng xã hội. Thuật toán word2vec đã được sử dụng để biểu diễn thông tin ngữ nghĩa trong nội dung thư điện tử. Các chỉ số khác nhau liên quan đến người gửi thư đã được trích xuất thành vector đặc trưng xã hội đại diện cho người gửi thư. Các thí nghiệm đã được thực hiện trên tập dữ liệu xếp hạng thư điện tử cá nhân do tác giả thu thập và xử lý. Phương pháp đề xuất đã thể hiện hiệu quả tốt hơn đáng kể so với phương pháp học máy truyền thống dựa trên máy phân loại SVM và bộ đặc trưng TF-IDF. Ngoài ra, so sánh đã được đưa ra giữa các cấu hình khác nhau của mô hình mạng nơ-ron, cụ thể là về kích thước vector từ ngữ và lựa chọn về thuật toán huấn luyện.

Trong khuôn khổ thời gian thực hiện nghiên cứu hạn chế, còn nhiều khía cạnh mà luận án chưa nghiên cứu một cách đầy đủ. Những vấn đề mà luận án chưa giải quyết được dưới đây sẽ là định hướng cho các nghiên cứu tiếp theo.

Các đề xuất trong luận án đã được thử nghiệm và so sánh với một số phương pháp khác nhưng số lượng phương pháp được thử nghiệm, so sánh còn hạn chế. Trong các nghiên cứu tiếp theo, những phương pháp đề xuất cần được thử nghiệm trên các tập dữ liệu khác. Đồng thời, cần thử nghiệm thêm nhiều phương pháp liên quan trên bộ dữ liệu mà luận án đã xây dựng. Những thí nghiệm nói trên có mục tiêu làm rõ hiệu quả của các đề xuất so với các phương pháp liên quan và tiếp tục kiểm chứng chất lượng của tập dữ liệu đã xây dựng.

Các đề xuất trong luận án chủ yếu sử dụng đặc trưng nội dung của thư điện tử và đặc trưng liên quan đến người gửi thư. Trong tương lai, nghiên cứu sẽ khai thác thêm các đặc trưng khác của thư điện tử như thời gian gửi/nhận thư, địa chỉ mạng của người gửi thư, và các thông tin từ những trường header khác. Ngoài ra, xác định thứ tự ưu tiên cho những bức thư có nội dung được mã hóa dưới dạng hình ảnh cũng là một trong những nội dung cần được khảo cứu trong các nghiên cứu tiếp theo.

Ngoài phương pháp xác định thứ tự ưu tiên của thư điện tử, còn những hướng khác để giải quyết tình trạng quá tải thư điện tử, ví dụ như tóm tắt nội dung thư hoặc trích xuất nội dung chính của thư điện tử. Đây cũng là những hướng nghiên cứu cần được xem xét trong tương lai.

Để cải thiện đóng góp cho bài toán xếp hạng thư điện tử của luận án, hướng nghiên cứu sau này sẽ áp dụng thêm những kỹ thuật biểu diễn nội dung mới hơn so với word2vec. Thử nghiệm thêm với các phương pháp biểu diễn từ ngữ phụ thuộc vào ngữ cảnh như ELMo [79] và BERT [83]. Về mặt thuật toán, các nghiên cứu tiếp theo có thể ứng dụng thêm những mô hình học sâu dành cho văn bản mà luận án chưa thử nghiệm. Về mặt đặc trưng, nghiên cứu tiếp theo có thể bổ sung thêm các đặc trưng xã hội khác nhau vào vector biểu diễn thư điện tử.

DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ

TẠP CHÍ KHOA HỌC

[1] Thanh, H. N., Dinh, Q. D., & Anh-Tran, Q. (2017). Personalized Email User Action Prediction Based on SpamAssassin. In *Cong Vinh P., Tuan Anh L., Loan N., Vongdoiwang Siricharoen W. (eds) Context-Aware Systems and Applications. ICCASA 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* (Vol. 193). Springer, Cham. https://doi.org/10.1007/978-3-319-56357-2_17

[2] Nguyễn, H. T., Đặng, Q. Đ., & Trần, A. Q. (2020). A neural network method for spamassasin rules generation. *Journal of Science and Technology on Information and Communications, 1*(4A), 4-11.

[3] Ha, N. T., Quan, D. D., & Anh, T. Q. (2021). Combining content and social features in a deep learning approach to Vietnamese email prioritization. *REV Journal on Electronics and Communications, 11*(3–4).

HỘI NGHỊ KHOA HỌC

[4] Nguyễn X. T., Trần Q. A., Trịnh B. N., & Nguyễn T. H. (2015). Ứng dụng tối ưu hóa đa mục tiêu trong bài toán tự động phân loại thư rác. *Hội thảo Quốc gia 2015 về Điện tử, Truyền thông và Công nghệ thông tin (REV-ECIT 2015)*, 30–35.

[5] Thanh, H. N., Dinh, Q. D., & Tran, Q. A. (2018). Predicting user's action on emails: Improvement with ham rules and real-world dataset. *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*. https://doi.org/10.1109/KSE.2018.8573330

TÀI LIỆU THAM KHẢO

[1] Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *IEEE International Conference on Neural Networks*, *1*, 586–591. https://doi.org/10.1109/ICNN.1993.298623

[2] *Conventions for Encoding the Vietnamese Language VISCII: VIetnamese Standard Code for Information Interchange VIQR: VIetnamese Quoted-Readable Specification* (Request for Comments RFC 1456). (1993). Internet Engineering Task Force. https://doi.org/10.17487/RFC1456

[3] Friedman, J. H. (1996). Another approach to polychotomous classification. *Technical Report, Statistics Department, Stanford University*.

[4] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

[5] Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. *Learning for Text Categorization: Papers from the 1998 Workshop*, *62*, 98–105.

[6] Hastie, T., & Tibshirani, R. (1998). Classification by pairwise coupling. *The Annals of Statistics*, *26*(2), 451–471. https://doi.org/10.1214/aos/1028144844

[7] Sareni, B., & Krahenbuhl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, *2*(3), 97–106. https://doi.org/10.1109/4235.735432

[8] Platt, J. C., Cristianini, N., & Shawe-Taylor, J. (1999). Large margin DAGs for multiclass classification. *Advances in Neural Information Processing Systems*, *12*, 547–553.

[9] Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, *10*(5), 1048–1054. https://doi.org/10.1109/72.788645

[10] Page, L., Brin, S., Motwani, R., & Winograd, T. (1999). *The PageRank Citation Ranking: Bringing Order to the Web* (Technical Report No. 1999–66). Stanford InfoLab.

[11] Hasegawa, T., & Ohara, H. (2000). Automatic Priority Assignment to E-mail Messages Based on Information Extraction and User's Action History. *Intelligent Problem Solving. Methodologies and Approaches*, 573–582. https://doi.org/10.1007/3-540-45049-1_69

[12] Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Report*, *103*. https://doi.org/10.3929/ETHZ-A-004284029

[13] Mason, J. (2002). Filtering spam with spamassassin. *HEANet Annual Conference*, *103*.

[14] Graham, P. (2003). Better bayesian filtering. *Proceedings of the 2003 Spam Conference*, *11*, 15–17.

[15] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., & Stamatopoulos, P. (2003). A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists. *Information Retrieval*, *6*(1), 49–73. https://doi.org/10.1023/A:1022948414856

[16] Peter, I. (2004). *The History of email*. Internet History Project. http://www.nethistory.info/History of the Internet/email.html

[17] Stern, H. (2004). *Fast SpamAssassin score learning tool*. https://svn.apache.org/repos/asf/spamassassin/trunk/masses/README.perceptron

[18] Yerazunis, W. S. (2004). The spam-filtering accuracy plateau at 99.9% accuracy and how to get past it. *Proceedings of the 2004 MIT Spam Conference*.

[19] Klimt, B., & Yang, Y. (2004). The Enron Corpus: A New Dataset for Email Classification Research. *Machine Learning: ECML 2004*, 217–226. https://doi.org/10.1007/978-3-540-30115-8_22

[20] Bekkerman, R. (2004). Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora. *Computer Science Department Faculty Publication Series*, *218*. https://scholarworks.umass.edu/cs_faculty_pubs/218

[21] Graham-Cumming, J. (2004, January 21). *How to Beat a Bayesian Spam Filter*. The MIT 2004 Spam Conference. https://lwn.net/Articles/67242/

[22] Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, *26*(6), 369–395. https://doi.org/10.1007/s00158-003-0368-6

[23] Cormack, G. V., & Lynam, T. R. (2005). TREC 2005 Spam Track Overview. *The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings*.

[24] Neustaedter, C., Brush, A., Smith, M., & Fisher, D. (2005, January 1). *The Social Network and Relationship Finder: Social Sorting for Email Triage.* Proceedings of the 2005 Conference on Email and Anti-Spam (CEAS).

[25] Dabbish, L. A., Kraut, R. E., Fussell, S., & Kiesler, S. (2005). Understanding email use: Predicting action on a message. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 691–700. https://doi.org/10.1145/1054972.1055068

[26] Boykin, P. O., & Roychowdhury, V. P. (2005). Leveraging social networks to fight spam. *Computer*, *38*(4), 61–68. https://doi.org/10.1109/MC.2005.132

[27] Chirita, P. A., Diederich, J., & Nejdl, W. (2005). MailRank: Using ranking for spam detection. *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 373–380. https://doi.org/10.1145/1099554.1099671

[28] Tran, Q. A., Duan, H., & Li, X. (2006). Real-time statistical rules for spam detection. *IJCSNS International Journal of Computer Science and Network Security*, *6*(2B), 178–184.

[29] Bui, N. L., Tran, Q. A., & Ha, Q. T. (2006). *User's authentic rating based on email networks*. The First International Conference on Mobile.

[30] Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, *91*(9), 992–1007. https://doi.org/10.1016/j.ress.2005.11.018

[31] Dabbish, L. A., & Kraut, R. E. (2006). Email Overload at Work: An Analysis of Factors Associated with Email Strain. *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, 431–440. https://doi.org/10.1145/1180875.1180941

[32] Spira, J. B., & Goldes, D. M. (2007). *Information overload: We have met the enemy and he is us*. Basex Inc.

[33] Duan, Z., Dong, Y., & Gopalan, K. (2007). DMTP: Controlling spam through message delivery differentiation. *Computer Networks*, *51*(10), 2616–2630. https://doi.org/10.1016/j.comnet.2006.11.015

[34] Le, H. P., Nguyen, T. M. H., Roussanaly, A., & Ho, T. V. (2008). A Hybrid Approach to Word Segmentation of Vietnamese Texts. *Language and Automata Theory and Applications*, 240–249. https://doi.org/10.1007/978-3-540-88282-4_23

[35] Caruana, G., & Li, M. (2008). A survey of emerging approaches to spam filtering. *ACM Computing Surveys*, *44*(2), 9:1-9:27. https://doi.org/10.1145/2089125.2089129

[36] Cormack, G. V. (2008). Email Spam Filtering: A Systematic Review. *Foundations and Trends in Information Retrieval*, *1*(4), 335–455. https://doi.org/10.1561/1500000006

[37] Ling, S. H., Iu, H. H. C., Chan, K. Y., Lam, H. K., Yeung, B. C. W., & Leung, F. H. (2008). Hybrid particle swarm optimization with wavelet mutation and its industrial applications. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics: A Publication of the IEEE Systems, Man, and Cybernetics Society*, *38*(3), 743–763. https://doi.org/10.1109/TSMCB.2008.921005

[38] Resnick, P. (2008). *Internet Message Format* (Request for Comments RFC 5322). Internet Engineering Task Force. https://doi.org/10.17487/RFC5322

[39] Wu, C. H. (2009). Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. *Expert Systems with Applications*, *36*(3), 4321–4330. https://doi.org/10.1016/j.eswa.2008.03.002

[40] Yoo, S., Yang, Y., Lin, F., & Moon, I. C. (2009). Mining social networks for personalized email prioritization. *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 967–976. https://doi.org/10.1145/1557019.1557124

[41] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

[42] Li, Q., & Mu, B. (2009). A Novel Method to Detect Junk Mail Traffic. *2009 Ninth International Conference on Hybrid Intelligent Systems*, *3*, 129–133. https://doi.org/10.1109/HIS.2009.239

[43] Guzella, T. S., & Caminhas, W. M. (2009). A review of machine learning approaches to Spam filtering. *Expert Systems with Applications*, *36*(7), 10206–10222. https://doi.org/10.1016/j.eswa.2009.02.037

[44] Ayodele, T., & Zhou, S. (2009). Applying machine learning techniques for e-mail management: Solution with intelligent e-mail reply prediction. *Journal of Engineering and Technology Research*, *1*(7), 143–151.

[45] Aberdeen, D., Pacovsky, O., & Slater, A. (2010). *The Learning Behind Gmail Priority Inbox*. LCCC : NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds.

[46] Yang, Y., Yoo, S., Lin, F., & Moon, I.-C. (2010). Personalized Email Prioritization Based on Content and Social Network Analysis. *IEEE Intelligent Systems*, *25*(4), 12–18. https://doi.org/10.1109/MIS.2010.56

[47] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, *12*(61), 2121–2159.

[48] Kucherawy, M., Crocker, D., & Hansen, T. (2011). *DomainKeys Identified Mail (DKIM) Signatures* (Request for Comments RFC 6376). Internet Engineering Task Force. https://doi.org/10.17487/RFC6376

[49] Yoo, S., Yang, Y., & Carbonell, J. (2011). Modeling personalized email prioritization: Classification-based and regression-based approaches. *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, 729–738. https://doi.org/10.1145/2063576.2063683

[50] Rao, J. M., & Reiley, D. H. (2012). The economics of spam. *Journal of Economic Perspectives*, *26*(3), 87–110. https://doi.org/10.1257/jep.26.3.87

[51] Minh, H. Q., Anh, T. Q., & Trang, L. T. (2012). Personalized Email Recommender System Based on User Actions. *Simulated Evolution and Learning*, 280–289. https://doi.org/10.1007/978-3-642-34859-4_28

[52] LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K.-R. (2012). Efficient BackProp. In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural Networks: Tricks of the Trade: Second Edition* (pp. 9–48). Springer. https://doi.org/10.1007/978-3-642-35289-8_3

[53] Hinton, G., Srivastava, N., & Swersky, K. (2012). *Neural networks for machine learning lecture 6a overview of mini-batch gradient descent*. http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf

[54] Tran, Q. A., Vu, M. T., Frater, M., & Jiang, F. (2012). Email user ranking based on email networks. *AIP Conference Proceedings*, *1479*(1), 1512–1517. https://doi.org/10.1063/1.4756451

[55] Covey, S. R. (2013). *The 7 habits of highly effective people: Powerful lessons in personal change*. Simon and Schuster.

[56] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *ArXiv:1301.3781 [Cs]*. http://arxiv.org/abs/1301.3781

[57] Vacek, M. (2014). Email overload: Causes, consequences and the future. *International Journal of Computer Theory and Engineering*, *6*(2), 170–176. https://doi.org/10.7763/IJCTE.2014.V6.857

[58] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958.

[59] Basto-Fernandes, V., Yevseyeva, I., Frantz, R. Z., Grilo, C., Díaz, N. P., & Emmerich, M. (2014). An Automatic Generation of Textual Pattern Rules for Digital Content Filters Proposal, Using Grammatical Evolution Genetic Programming. *Procedia Technology*, *16*, 806–812. https://doi.org/10.1016/j.protcy.2014.10.030

[60] Kitterman, S. (2014). *Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1* (Request for Comments RFC 7208). Internet Engineering Task Force. https://doi.org/10.17487/RFC7208

[61] von Lücken, C., Barán, B., & Brizuela, C. (2014). A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, *58*(3), 707–756. https://doi.org/10.1007/s10589-014-9644-1

[62] Dinh, Q. D., Tran, Q. A., & Jiang, F. (2014). Automated generation of ham rules for Vietnamese spam filtering. *The 2014 Seventh IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 1–5. https://doi.org/10.1109/CISDA.2014.7035628

[63] Nguyen, L., Tran, A. Q., & Bui, L. T. (2014). DMEA-II and its application on spam email detection problems. *The 2014 Seventh IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 1–6. https://doi.org/10.1109/CISDA.2014.7035634

[64] The Radicati Group. (2015). *Email Statistics Report, 2015-2019* (p. 4). https://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf

[65] Mi, G., Gao, Y., & Tan, Y. (2015). Apply Stacked Auto-Encoder to Spam Detection. In Y. Tan, Y. Shi, F. Buarque, A. Gelbukh, S. Das, & A. Engelbrecht (Eds.), *Advances in Swarm and Computational Intelligence* (pp. 3–15). Springer International Publishing. https://doi.org/10.1007/978-3-319-20472-7_1

[66] Youn, S., & Cho, H. C. (2015). Improved Spam Filter via Handling of Text Embedded Image E-mail. *Journal of Electrical Engineering & Technology*, *10*(1), 401–407. https://doi.org/10.5370/JEET.2015.10.1.401

[67] Alsmadi, I., & Alhami, I. (2015). Clustering and classification of email contents. *Journal of King Saud University - Computer and Information Sciences*, *27*(1), 46–57. https://doi.org/10.1016/j.jksuci.2014.03.014

[68] Kucherawy, M., & Zwicky, E. (2015). *Domain-based Message Authentication, Reporting, and Conformance (DMARC)* (Request for Comments RFC 7489). Internet Engineering Task Force. https://doi.org/10.17487/RFC7489

[69] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

[70] Kooti, F., Aiello, L. M., Grbovic, M., Lerman, K., & Mantrach, A. (2015). Evolution of Conversations in the Age of Email Overload. *Proceedings of the 24th International Conference on World Wide Web*, 603–613. https://doi.org/10.1145/2736277.2741130

[71] Di Castro, D., Karnin, Z., Lewin-Eytan, L., & Maarek, Y. (2016). You've got Mail, and Here is What you Could do With It! Analyzing and Predicting Actions on Email Messages. *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, 307–316. https://doi.org/10.1145/2835776.2835811

[72] Mujtaba, G., Shuib, L., Raj, R. G., Majeed, N., & Al-Garadi, M. A. (2017). Email Classification Research Trends: Review and Open Issues. *IEEE Access*, *5*, 9044–9064. https://doi.org/10.1109/ACCESS.2017.2702187

[73] Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. *ArXiv:1412.6980 [Cs]*. http://arxiv.org/abs/1412.6980

[74] Yang, L., Dumais, S. T., Bennett, P. N., & Awadallah, A. H. (2017). Characterizing and Predicting Enterprise Email Reply Behavior. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 235–244. https://doi.org/10.1145/3077136.3080782

[75] Seth, S., & Biswas, S. (2017). Multimodal Spam Classification Using Deep Learning Techniques. *2017 13th International Conference on Signal-Image Technology Internet-Based Systems (SITIS)*, 346–349. https://doi.org/10.1109/SITIS.2017.91

[76] Nguyen, L., Nguyen, D., Điệp, L., Tuan, V., Tran, Q. A., & Lâm, B. (2017). DETECTING VIETNAMESE SPAMS USING A MULTI-OBJECTIVE EVOLUTIONARY APPROACH. *Journal of Military Science and Technology*, *2017*(12).

[77] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

[78] Vu, T., Nguyen, D. Q., Dras, M., & Johnson, M. (2018). VnCoreNLP: A Vietnamese Natural Language Processing Toolkit. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, 56–60.

[79] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations* (arXiv:1802.05365). arXiv. https://doi.org/10.48550/arXiv.1802.05365

[80] Yawen, W., Fan, Y., & Yanxi, W. (2018). *Research of Email Classification based on Deep Neural Network*. 73–77. https://doi.org/10.2991/icsnce-18.2018.16

[81] Yin, Z., & Shen, Y. (2018). On the dimensionality of word embedding. *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 895–906.

[82] Mukherjee, S., & Jiang, K. (2019). A Content-Based Approach to Email Triage Action Prediction: Exploration and Evaluation. *ArXiv:1905.01991 [Cs, Stat]*. http://arxiv.org/abs/1905.01991

[83] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (arXiv:1810.04805). arXiv. https://doi.org/10.48550/arXiv.1810.04805

[84] Jain, G., Sharma, M., & Agarwal, B. (2019). Optimizing semantic LSTM for spam detection. *International Journal of Information Technology*, *11*(2), 239–250. https://doi.org/10.1007/s41870-018-0157-5

[85] Long, D. H., Lam, N. T., Thuong, P. T., Dam, N. Q., & Nikolaevich, T. V. (2020). Evaluating the priority of email using machine learning. *International Journal of Emerging Trends in Engineering Research*, *8*(9). https://doi.org/10.30534/ijeter/2020/233892020