HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN VĂN CẢNH

NGHIÊN CỨU PHƯƠNG PHÁP ĐÁNH GIÁ MỨC ĐỘ ƯU TIÊN CỦA THƯ ĐIỆN TỬ

LUẬN VĂN THẠC SĨ KỸ THUẬT

HÀ NỘI – 2020

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN VĂN CẢNH

NGHIÊN CỨU PHƯƠNG PHÁP ĐÁNH GIÁ MỨC ĐỘ ƯU TIÊN CỦA THƯ ĐIỆN TỬ

Chuyên ngành: Hệ thống thông tin

Mã số: 8.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

(*Theo định hướng ứng dụng*)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. ĐỖ XUÂN CHỢ

HÀ NỘI - 2020

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

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

Tác giả

Nguyễn Văn Cảnh

LỜI CẢM ƠN

Tôi xin trân trọng cảm ơn các thầy cô trong Khoa công nghệ thông tin đã tạo điều kiện cho tôi một môi trường học tập tốt, đồng thời truyền đạt cho tôi một vốn kiến thức quý báu, một tư duy khoa học để phục vụ cho quá trình học tập và công tác của tôi.

Tôi xin gửi lời cảm ơn đến các bạn trong lớp Cao học Hệ thống thông tin M18CQIS01-B khóa 2018- 2020 đã giúp đỡ tôi trong suốt thời gian học tập vừa qua. Đặc biệt, tôi xin được bày tỏ lòng biết ơn sâu sắc đến TS. ĐỖ XUÂN CHỢ đã tận tình chỉ bảo cho tôi trong suốt quá trình học tập và nghiên cứu, giúp tôi có nhận thức đúng đắn về kiến thức khoa học, tác phong học tập và làm việc, tạo điều kiện thuận lợi để tôi hoàn thành luận văn này.

Cuối cùng, tôi xin được gửi lời cảm ơn tới gia đình, đồng nghiệp, người thân đã động viên, giúp đỡ tôi trong quá trình hoàn thành luận văn.

Tác giả

Nguyễn Văn Cảnh

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
KNN	K-Nearest Neighbors	K láng giềng gần nhất
IDF	Inverse Document Frequency	Nghịch đảo tần suất của văn bản
TF	Term Frequency	Tần suất xuất hiện của từ

DANH MỤC BẢNG BIỂU

DANH MỤC HÌNH VẼ

MỞ ĐẦU

1. Lý do chọn đề tài

Sự bùng nổ của internet kéo theo hàng loạt các phát minh khoa học công nghệ được ứng dụng trong cuộc sống. Thư điện tử là một ứng dụng được sử dụng rộng rãi trên toàn cầu. Thư điện tử giúp rút ngắn thời gian, khoảng cách giữa việc gửi và nhận thư, tiết kiệm chi phí cho quá trình gửi thư. Trước khi thư điện tử ra đời, các thao tác chuyển thư thật phức tạp từ nhà đến bưu điện, từ bưu điện này đến bưu điện khác, từ bưu điện đến nơi nhận. Quá trình đó làm mất vài ngày hoặc cả tuần, tin tức được truyền đi rất chậm. Giá cả của việc chuyển thư truyền thống thì đắt hơn nhiều so với thư điện tử. Việc chuyển thư điện tử chỉ mất vài giây, mọi thao tác được đơn giản hóa chi phí rất rẻ, miễn là có kết nối Internet. Việc viết thư điện tử cũng nhanh chóng tiện lợi, truyền tải đầy đủ thông điệp mà người dùng muốn gửi đi bao gồm hình ảnh, âm thanh, nội dung văn bản … với dung lượng lớn theo dạng nhập trực tiếp vào khung soạn thảo hoặc đính kèm.

Do hàng ngày người dùng thường nhận được rất nhiều thư điện tử khác nhau nên sẽ khó khăn trong việc xác định và nhận dạng những thư điện tử nào quan trọng cần đọc và trả lời sớm, những thư nào có thể chỉ để theo dõi. Vì vậy ta phải dùng đến khái niệm "Mức độ ưu tiên" với thư điện tử. Theo định nghĩa tiếng Anh "Mức độ ưu tiên" - "Priority" được sử dụng để so sánh hai vật hoặc hai điều kiện, khi mà một vật/điều kiện phải quan tâm nhiều hơn những vật/điều kiện khác và phải được giải quyết trước khi chuyển sang (những) vật/điều kiện tiếp theo. Công cụ hỗ trợ nhận dạng và phân loại mức độ ưu tiên cho thư điện tử là cần thiết.

Từ những lý do trên, học viên với sự giúp đỡ của TS. Đỗ Xuân Chợ lựa chọn đề tài: "**Nghiên cứu phương pháp đánh giá mức độ ưu tiên trong thư điện tử**".

Luận văn bao gồm 3 chương:

Chương 1: Tổng quan về hệ thống thư điện tử

Chương này trình bày tổng quan về hệ thống thư điện tử bao gồm: định nghĩa, thành phần, chức năng, kiến trúc, vai trò và tầm quan trọng… của thư điện tử. Bên cạnh đó, trong chương này, luận văn sẽ trình bày một số công cụ mã nguồn mở để xây dựng hệ thống thư điện tử.

Chương 2 : Đánh giá mức độ ưu tiên của thư điện tử

Luận văn sẽ trình bày một số công nghệ hỗ trơ phân loại mức độ ưu tiên của thư điện tử: Phương pháp phân loại thư rác….Sau đó là phương pháp nhằm đánh giá, phân loại mức độ ưu tiên cho thư điện tử.

Chương 3: Cài đặt và thử nghiệm

Tiến hành thử nghiệm phương pháp đánh giá độ ưu tiên của thư điện tử ở chương hai. Bao gồm: Dữ liệu thực nghiệm ,kịch bản thực nghiệm, kết quả thực nghiệm.

CHƯƠNG 1 - TỔNG QUAN VỀ THƯ ĐIỆN TỬ

Nội dung chương 1 đề cập đến khái niệm hệ thống thư điện tử bao gồm: định nghĩa, thành phần, chức năng, kiến trúc, vai trò và tầm quan trọng và sự cần thiết của việc phân loại độ ưu tiên của thư điện tử.

1.1 Khái niệm thư điện tử

Thư điện còn gọi tắt là E-Mail, là một dịch vụ được triển khai trên các mạng máy tính cho phép người dùng có thể trao đổi thư từ với nhau. Thư điện tử là một thông điệp gửi từ máy tính này đến máy tính khác trên mạng máy tính và mang nội dung cần thiết từ người gửi đến người nhận. Thư điện tử truyền gửi được nội dung chữ và các nội dung đa phương tiện như hình ảnh, âm thanh, video…

1.2 Lịch sử phát triển

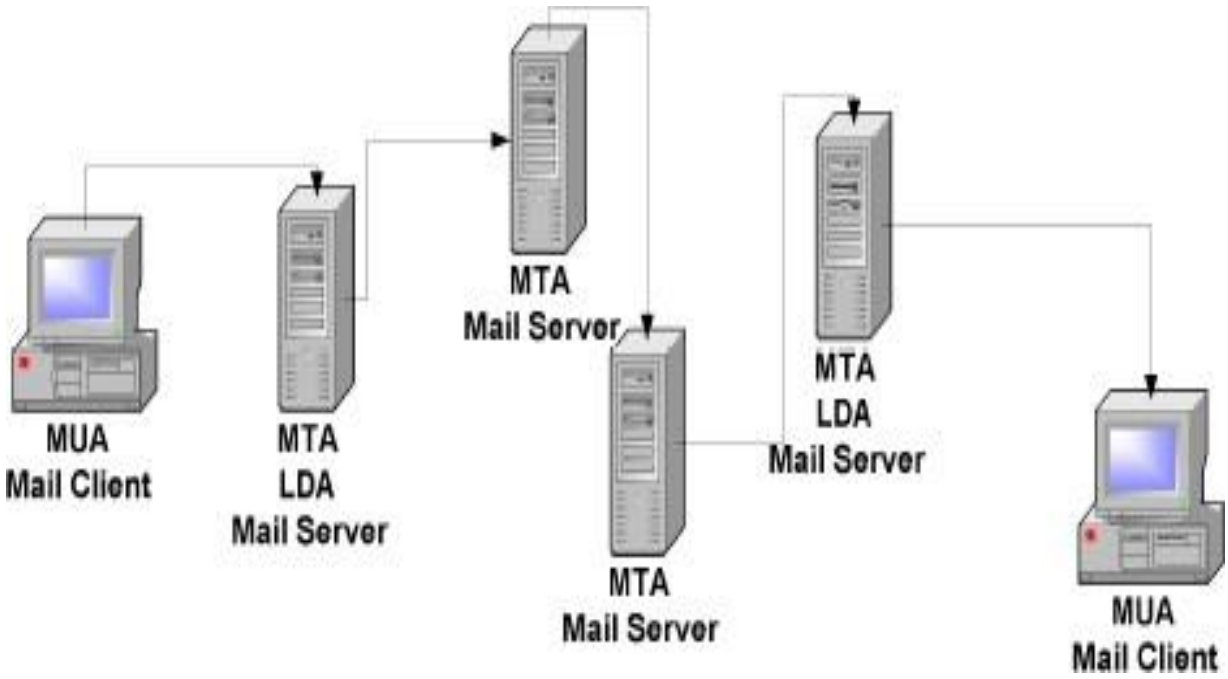
Năm 1971 Ray Tomlinson thực hiện gửi thành công một thông báo thư tín điện tử đầu tiên trong mạng RPANET

Tomlinson đã sửa đổi hệ thống xử lý thông báo để người sử dụng có thể gửi các thông báo cho các đối tượng nhận không chỉ trong một hệ thống mà trên các hệ thống ARPANET khác

Sau đó nhiều công trình nghiên cứu khác đã được tiến hành và thư tín điện tử đã nhanh chóng trở thành một ứng dụng được sử dụng nhiều nhất trên ARPANET trước đây và Internet ngày nay

1.3 Thành phần cấu trúc hệ thống thư điện tử

Hệ thống Mail Server là một hệ thống tổng thể bao gồm nhiều thành phần hoạt động tương tác với nhau. Mỗi thành phần bản thân phục vụ các dịch vụ khác nhau, nhưng đồng thời các kết quả lại được đưa đến các thành phần khác để xử lý tiếp theo. Hình 1.1 dưới đây là mô hình của hệ thống Mail Server và sự tương tác giữa các thành phần:



Hình 1.1 Mô hình hệ thống thư điện tử

Hầu hết hệ thống thư điện tử bao gồm ba thành phần cơ bản là MUA, MTA và MDA.

1.3.1 MTA(Mail Transfer Agent)

- Khi các bức thư được gửi đến từ MUA, MTA có nhiệm vụ nhận diện người gửi và người nhận từ thông tin đóng gói trong phần header của thư và điền các thông tin cần thiết vào header. Sau đó MTA chuyển thư cho MDA để chuyển đến hộp thư ngay tại MTA, hoặc chuyển cho Remote-MTA.
- Việc chuyển giao các bức thư được các MTA quyết định dựa trên địa chỉ người nhận.
- Nếu nó trùng với hộp thư do MTA (Local-MTA) quản lý thì bức thư được chuyển cho MDA để chuyển vào hộp thư.
- Nếu địa chỉ gửi bị lỗi, bức thư có thể được chuyển trở lại người gửi.
- Nếu không bị lỗi nhưng không phải là bức thư của MTA, tên miền được sử dụng

để xác định xem Remote-MTA nào sẽ nhận thư, theo các bản ghi MX trên hệ thống tên miền.

- Khi các bản ghi MX xác định được Remote-MTA quản lý tên miền đó thì không có nghĩa là người nhận thuộc Remote-MTA. Mà Remote- MTA có thể đơn giản chỉ trung chuyển (relay) thư cho một MTA khác, có thể định tuyến bức thư cho địa chỉ khác như vai trò của một dịch vụ domain ảo(domain gateway) hoặc người nhận không tồn tại và Remote-MTA sẽ gửi trả lại cho MUA gửi một cảnh báo.

1.3.2 MDA (Mail Delivery Agent)

Là một chương trình được MTA sử dụng để đẩy thư vào hộp thư của người dùng. Ngoài ra MDA còn có khả năng lọc thư, định hướng thư... Thường là MTA được tích hợp với một MDA hoặc một vài MDA.

1.3.3 MUA (Mail User Agent)

- MUA là chương trình quản lý thư đầu cuối cho phép người dùng có thể đọc, viết và lấy thư về từ MTA.
- MUA có thể lấy thư từ Mail Server về để xử lý (sử dụng giao thức POP) hoặc chuyển thư cho một MUA khác thông qua MTA (sử dụng giao thức SMTP).

Hoặc MUA có thể xử lý trực tiếp thư ngay trên Mail Server (dùng giao thức IMAP).

- Đằng sau những công việc vận chuyển thì chức năng chính của MUA là cung cấp giao diện cho người dùng tương tác với thư, gồm có:
- Soạn thảo, gửi thư.
- Hiển thị thư, gồm cả các tệp đính kèm.
- Gửi trả hay chuyển tiếp thư.

- Gắn các tệp vào các thư gửi đi (Text, HTML, MIME v.v…).
- Thay đổi các tham số(ví dụ như server được sử dụng, kiểu hiển thị thư, kiểu mã hoá thư v.v…).
- Thao tác trên các thư mục thư địa phương và ở đầu xa.
- Cung cấp số địa chỉ thư (danh bạ địa chỉ).
- Lọc thư.

1.4 Các giải pháp thư điện tử mã nguồn mở

Hiện nay trên thế giới đã xuất hiện rất nhiều sản phẩm xây dựng một hệ thống Mail Server. Có nhiều sản phẩm với giá rẻ (thậm chí miễn phí), nhỏ gọn, cài đặt và quản trị đơn giản, như WorkGroupMail, Surge Mail Server, Kerio Mail Server. Cũng có những sản phẩm lớn, giá thành cao, tính năng phong phú, đáp ứng được sự ổn định và an toàn như Mail Exchange của Microsoft, Merak Mail Server. Trong thế giới mã nguồn mở hiện nay, đã có rất nhiều hệ thống truyền tải thư điện tử MTA (Mail Transfer Agent) được phát triển. Nổi tiếng và phổ biến trong số đó gồm có: Zimbra, Sendmail, Qmail, Postfix, Exim, Courier. Mỗi MTA đều có những ưu điểm và nhược điểm riêng.[7]

1.4.1 Zimbra

Zimbra, hệ thống thư điện tử thế hệ mới, được xây dựng bởi cộng đồng phầm mềm tự do nguồn mở và công ty VMWare, đáp ứng các nhu cầu về trao đổi thư tín điện tử và hỗ trợ làm việc cộng tác kỷ nguyên hậu PC. Ứng dụng nguồn mở này có thể áp dụng cho các doanh nghiệp, nhà cung cấp dịch vụ, các tổ chức giáo dục, hay trong môi trường chính phủ..., mang tới cho người dùng rất nhiều lợi ích trong việc quản lý và chia sẻ thư tín, lịch công tác, sổ địa chỉ, tài liệu. Với hiệu năng hoạt động cao, các thao tác gửi, nhận, tải dữ liệu diễn ra hết sức nhanh chóng góp phần tiết kiệm thời gian cho người dùng. Đồng thời, người dùng cũng không cần phải lo lắng về việc quản trị hệ thống bởi mọi thao tác đều hết sức đơn giản và tiện lợi.Một điều rất đáng quan tâm của

hệ thống thư điện tử Zimbra đó là công nghệ trên mã nguồn mở cho phép người dùng tiết kiệm được tối đa chi phí mà vẫn đảm bảo được nguyên tắc tôn trọng bản quyền.

1.4.2 Sendmail

Sendmail (http://www.sendmail.org) là MTA đơn giản và lâu đời nhất trên các dòng Unix thời xưa. Ngày nay, trên các hệ thống Linux, đặc biệt là các sảm phẩm của RedHat, Sendmail vẫn được cài đặt là MTA mặc định cho hệ thống. Ngày nay, Sendmail đa được thương mại hóa bên cạnh sản phẩn miễn phí và vẫn được tiếp tục duy trì, phát triển. Tuy nhiên, vì được thiết kế theo cấu trúc khối và ảnh hưởng từ cấu trúc cũ, nên Sendmail chưa đạt được tính năng ổn định và bảo mật của một MTA như mong muốn.

1.4.3 Qmail

Qmail được viết bởi Bernstein, là một MTA dành cho hệ điều hành tựa Unix, bao gồm Linux, FreeBSD, Sun Solaris. Qmail ra đời như một tất yếu thay thế cho Sendmail và các yếu điểm của nó. Vì vậy, Qmail ngay từ ban đầu đã được thiết kế đơn giản, module hóa với tiêu chí bảo mật được đặt lên rất cao. Đồng thời, Qmail là một MTA hiện đại nên hỗ trợ tốt các kiểu định dạng mới hiện nay như định dạng hòm thư Maildir…Do Qmail được thiết kế module hóa và tối ưu hóa các tính năng ngay từ đầu, nên nó có tốc độ thực thi rất nhanh và ổn định.

1.4.4 Postfix

Weitse Venema, tác giả của các phần mềm miễn phí nổi tiếng như TCP Wrappers, SATAN và Logdaemon, ông không hài lòng khi sử dụng các MTA hiện có (bao gồm cả Qmail), vì vậy, ông đa viết ra Postfix (http://www.postfix.org). Postfix là một MTA mới, có khả năng thực thi cao, thừa kế cấu trúc thiết kế tốt từ Qmail, trong khi đó vẫn giữ được tính tương thích tối đa với Sendmail. So sánh với Qmail, Postfix có kích thước lớn hơn, phức tạp hơn, trong khi đó lại kém bảo mật, kém tin cậy và chạy chậm

hơn. Tuy Postfix cũng được thiết kế theo cấu trúc module, nhưng các module của Postfix chạy dưới quyền của cùng một người dùng hệ thống, vì vậy sự hỏng hóc của một module có thể ảnh hưởng đến toàn bộ hệ thống. Xét về tổng thể, Postfix là một MTA tốt. Nếu vấn đề bảo mật và khả năng thực thi của hệ thống không được đòi hỏi quá cao, người quản trị có thể chọn và sử dụng Postfix.

1.4.5 Exim

Philip Hazel đa phát triển Exim (http://www.exim.org) tại trường đại học Cambridge. Nó được thiết kế theo xu hướng nhỏ và đơn giản nhưng vẫn đảm bảo các tính năng. Tuy nhiên, Exim vẫn được thiết kế theo cấu trúc khối, và hai yếu tố quan trọng với các MTA hiện đại là bảo mật và khả năng thực thi lại không được coi trọng. Hiện nay, Exim là MTA được lựa chọn và cài đặt mặc định trên các phiên bản phân phối Linux dựa theo Debian, ngoài ra nó không được sử dụng rộng rãi.

Như vậy, tùy theo mục đích và nhu cầu sử dụng, người quản trị sẽ lựa chọn một MTA cho hệ thống của mình, ngoài ra, với mỗi điều kiện và môi trường khác nhau, mỗi MTA lại có mức độ phù hợp khác nhau. Với các ưu điểm vượt trội rõ rệt của Zimbra, đây là một lựa chọn phù hợp cho các doanh nghiệp, nhà cung cấp dịch vụ, các tổ chức giáo dục, hay trong môi trường chính phủ.

1.5 Kiến trúc hệ thống thư điện tử mã nguồn mở Zimbra

Zimbra là ứng dụng thư điện tử nguồn mở cung cấp một giải pháp, một hệ thống hoàn chỉnh để triển khai dịch vụ email (cả server và client) và môi trường chia sẻ cộng tác phục vụ cho quản lý và công việc.

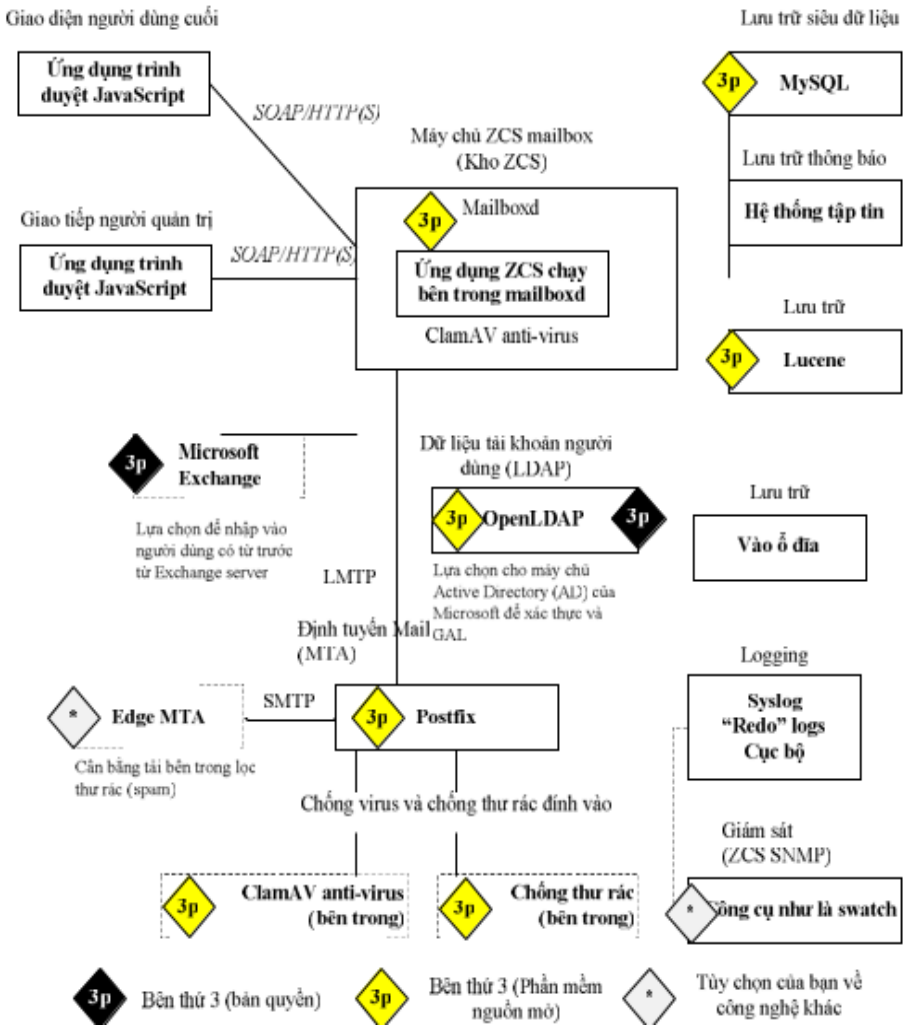
Kiến trúc hệ thống thư điện tử nguồn mở Zimbra bao gồm những lõi sau [8]:

- Các mã nguồn mở tích hợp trong Zimbra: Linux®, Apache Tomcat, Postfix, MySQL®, OpenLDAP®.
- Giao thức chuẩn được sử dụng là: SMTP, LMTP, SOAP, XML, IMAP, POP.
- Công nghệ được sử dụng để thiết kế là: Java, JavaScript thin client, DHTML.

- Trình duyệt dựa trên giao diện giao diện khách hàng, giao diện này cho phép người dùng dễ dàng truy cập vào tất cả các chức năng của Zimbra Collaboration Suite (ZCS).

Các thành phần mã nguồn mở được dùng với zimba [8]:

- Jetty ứng dụng máy chủ web chạy phần mềm zimbra.
- Postfix một nguồn mở chuyển giao các agent.
- OpenLDAP phần mềm nguồn mở xác thực người dùng (LDAP: Lightweight Directory Access Protocol).
- Phần mềm cơ sở dữ liệu MySQL.
- Lucence với đầy đủ tính năng và công cụ tìm kiếm.
- Verity dùng để chuyển đổi các tin đính kèm nhất định.
- Anti-virus các thành phần chống thư rác.
- ClamAV phần mềm quét chống virus để bảo vệ chống các tập tin độc hại.
- SpamAssassinxác định thư rác.
- James/Sieve filtering: Sử dụng để tạo các bộ lọc cho thư điện tử.



Hình 1.2 Kiến trúc hệ thống Zimbra [8]

- **Zimbra Core**: Gói này bao gồm các thư viện, tiện ích, công cụ giám sát và cấu hình cơ bản các tập tin.
- **Zimbra Convertd**: Được cài đặt trên máy chủ Zimbra.
- **Zimbra LDAP**: Xác thực người dùng được cung cấp qua OpenLDAP. Mỗi tài khoản trên máy chủ Zimbra có một ID hộp thư duy nhất để xác định tài khoản.
- **Zimbra MTA** (Mail Transfer Agent): Postfix là nguồn mở để chuyển, nhận thư

thông qua SMTP và định tuyến mỗi tin nhắn đến hộp thư máy chủ.

- Zimbra Store (Zimbra server):

Bao gồm các gói phần mềm lưu trữ trong Zimbra giúp cho việc cài đặt hộp thư máy chủ.

Mỗi tài khoản được cấu hình trên một hộp thư máy chủ, tài khoản này được liên kết với hộp thư có chứa tin nhắn và các tệp đính kèm.

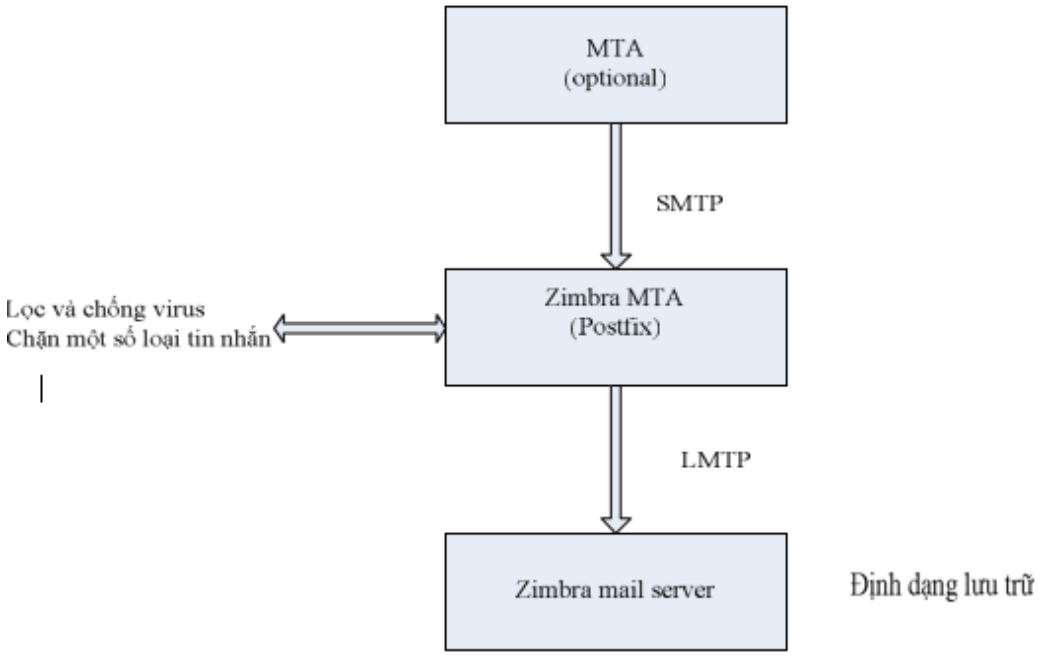
Tệp đính kèm được chuyển sang định dạng HTML khi người dùng click vào chức năng xem dạng HTML trên web Zimbra.

- **Data store**: MySQL được dùng để lưu trữ dữ liệu, các ID hộp thư nội bộ được liên kết với tài khoản người dùng. Cơ sở dữ liệu này chứa các thiết lập của người dùng định nghĩa, các thư mục, lập lịch, địa chỉ liên lạc, tình trạng mỗi email (đọc hay chưa đọc), các thẻ liên quan đến tin nhắn.
- **Message store**: Đây là nơi lưu trữ tất cả các thông báo của hộp thư, và các tệp đính kèm. Tin nhắn được lưu trữ theo định dạng MIME (MIME: Multipurpose Internet Mail Extensions là một chuẩn Internet về định dạng cho thư điện tử). Mỗi tin nhắn đó được gửi tới nhiều người nhận có tài khoản trên một hộp thư máy chủ được lưu trữ trong hệ thống tập tin.
- **Index store**: Chỉ số và công nghệ tìm kiêm được cung cấp qua Lucene (*Lucene*là thư viện Java mã mở nổi tiếng giúp xây dựng một công cụ tìm kiếm). Chỉ số các tập tin được duy trì cho mỗi hộp thư.
- **Zimbra Logger**: Cài đặt gói Zimbra Logger là tùy chọn và được cài đặt trên hộp thư máy chủ. Các Zimbra Logger được cài đặt làm công cụ tập hợp, báo cáo syslog. Nếu hệ thống không cài Logger thì số liệu thông kê log sẽ không hiển thị trên giao diện điều khiển.

- **Zimbra Spell**: Đây là gói cài đặt để kiểm tra chính tả trên web Zimbra.
- **Zimbra Proxy**: Sử dụng IMAP proxy cho phép thu hồi thư cho một miền được phân chia giữa nhiều máy chủ Zimbra cho mỗi người dùng.
- **Zimbra MTA:** Zimbra MTA (Mail Transfer Agent) là một dịch vụ nhận mail thông qua giao thức SMTP sử dụng giao thức LMTP (Local Mail Transfer Protocol) cho phù hợp với mail server Zimbra.

1.6 Triển khai Zimbra MTA

- Tổ chức Zimbra bao gồm một phiên bản dịch sẵn của Postfix.
- Postfix thực hiện việc chuyển tiếp mail Zimbra, nó nhận được tin nhắn thông qua SMTP và các thông điệp từ máy chủ Zimbra.
- Postfix cũng đóng vai trò trong việc chuyển giao các thông điệp gửi đi.

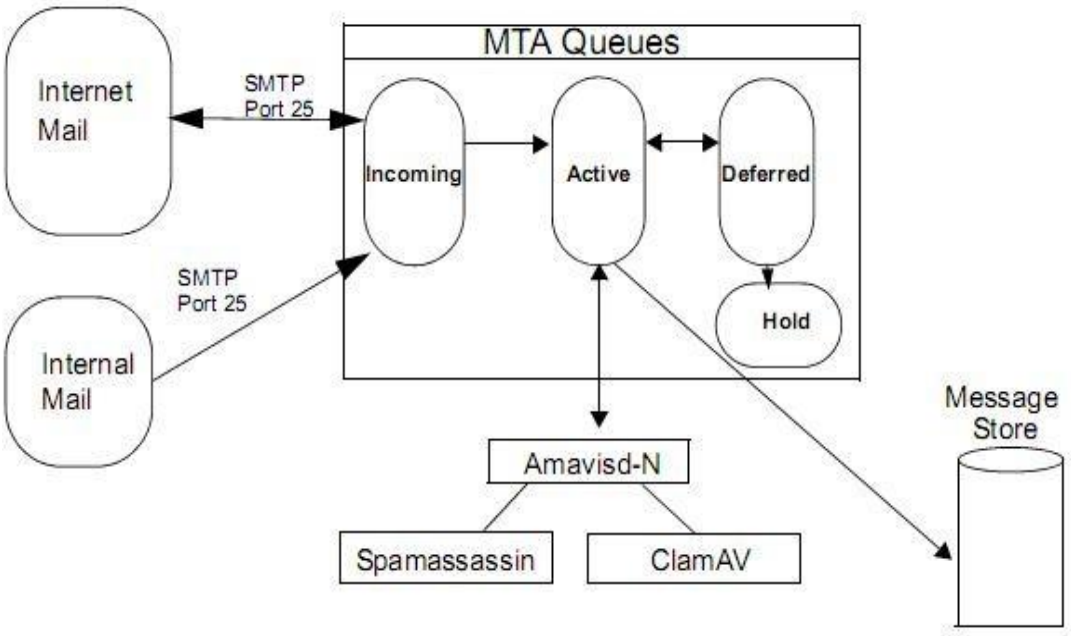


Hình 1.3 Postfix trong môi trường Zimbra [8]

1.6.1 Tiếp nhận và gửi thư thông qua Zimbra MTA

- Các MTA Zimbra cung cấp cả đầu vào và thông điệp gửi đi. Đối với thư gửi đi các MTA Zimbra xác địa chỉ định đích đến của người nhận. Nếu đích đến là một máy chủ mail từ xa thì MTA Zimbra phải thiết lập một phương thức truyền thông để truyền thông điệp đến máy chủ đó. Đối với tin nhắn gửi đến các MTA phải có khả năng chấp nhận kết nối các yêu cầu từ máy chủ ở xa và nhận tin nhắn cho mạng nội bộ sử dụng.
- Để gửi và nhận email các MTA Zimbra phải được cấu hình trong DNS. Đối với thư gửi đi sử dụng DNS để định tuyến email.
- Để nhận được email thì bản ghi MX phải được cấu hình định tuyến đúng để thư đến được máy cài mail server.
- Hàng đợi tin nhắn trong Zimbra MTA:
- Khi nhận được nhiều mail Zimbra MTA có các cơ chế định tuyến để quản lý hàng đợi tin nhắn

Trong Zimbra MTA duy trùy 4 hàng đợi: Incoming, active, deferred, hold



Hình 1.4 Hàng đợi tin nhắn trong Zimbra MTA [8]

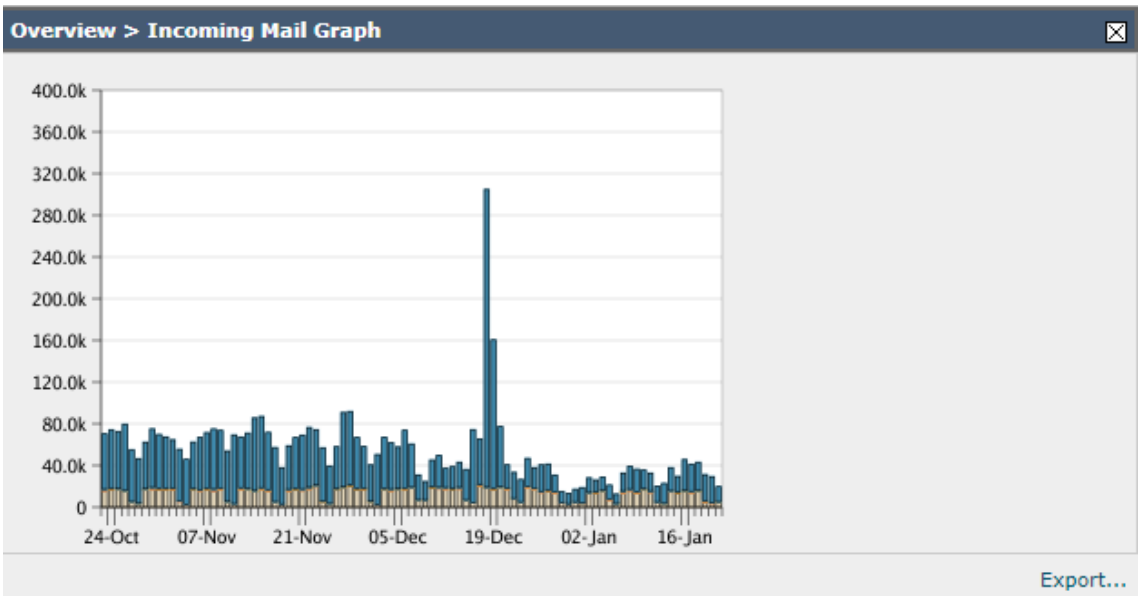
- **Incoming**: Khi các thư, tin nhắn mới được gửi đến sẽ được xếp vào hàng đợi. Mỗi tin nhắn được xác định với một tệp tin duy nhất. Thông điệp trong hàng đợi sẽ được chuyển đến nơi mà các thư đang xếp hàng. Nếu không có vấn đề gì xảy ra thì việc di chuyển tin nhắn thông qua các hàng đợi này sẽ được diễn ra một cách nhanh chóng.
- **Active**: Các hàng đợi thư được kích hoạt để sẵn sàng gửi đi.MTA đưa ra một số lượng giới hạn các tin nhắn được xếp vào hàng đợi tại một thời điểm. Từ đây các tin nhắn sẽ được chuyển đến hệ thống chống virus và lọc thư rác trước khi được giao hoặc chuyển đến hàng đợi khác.
- **Deferred**: Đây là các tin nhắn không được giao đi và được đặt trong hàng đợi chậm (chờ). Hàng đợi này sẽ được quét thường xuyên để gửi lại tin nhắn. Nếu tin nhắn không thể gửi lại sau một số lần nhất định thì tin nhắn đó sẽ được đánh dấu gửi không thành công. Mặc định cho các hàng đợi trả lại là 5 ngày, bạn cũng có thể thay đổi giá trị mặc định này trong MTA.
- **Hold**: Đây là hàng đợi lưu giữ các mail mà có thể không được xử lý.Các mail ở trong hàng đợi này cho đến khi người quản trị xử lý đến.

1.7 Những tiện ích và vai trò của thư điện tử trong cuộc sống ngày nay

Thư điện tử là một ứng dụng được sử dụng rộng rãi trên toàn cầu. Thư điện tử giúp rút ngắn thời gian, khoảng cách giữa việc gửi và nhận thư, tiết kiệm chi phí cho quá trình gửi thư. Trước khi thư điện tử ra đời, các thao tác chuyển thư thật phức tạp từ nhà đến bưu điện, từ bưu điện này đến bưu điện khác, từ bưu điện đến nơi nhận. Quá trình đó làm mất vài ngày hoặc cả tuần, tin tức được truyền đi rất chậm. Giá cả của việc chuyển thư truyền thống thì đắt hơn nhiều so với thư điện tử. Việc chuyển thư điện tử chỉ mất vài giây, mọi thao tác được đơn giản hóa chi phí rất rẻ, miễn là có kết nối

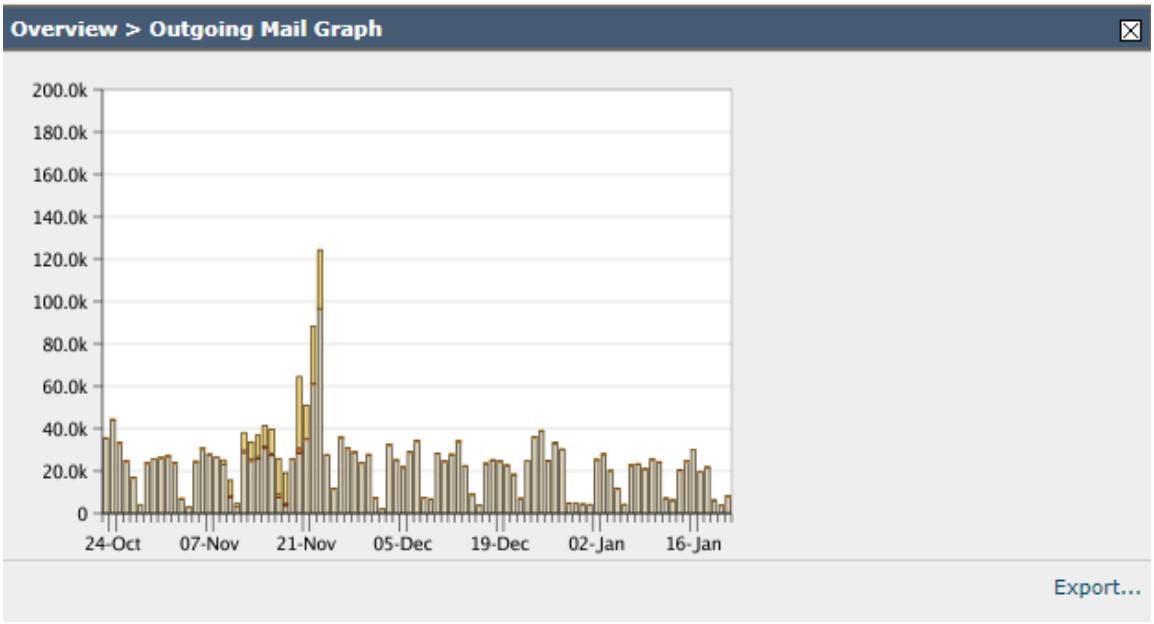
Internet. Việc viết thư điện tử cũng nhanh chóng tiện lợi, truyền tải đầy đủ thông điệp mà người dùng muốn gửi đi bao gồm hình ảnh, âm thanh, nội dung văn bản … với dung lượng lớn theo dạng nhập trực tiếp vào khung soạn thảo hoặc đính kèm.

Sau đây là thống kê số thư điện tử của một ngân hàng tại Việt Nam trong khoảng tháng 10/2018 đến tháng 2/2019.



Hình 1.5 Thống kê số lượng Incoming emails

Lượng mail lớn nhất: 320.000 emails/ngày.Lượng mail trung bình: 100.000 mails/ngày.



Hình 1.6 Thống kê số lượng Outgoing emails

Theo nguồn thống kê trên Internet (Radicati) [4] :

Công nhân Mỹ sẽ nhận được trung bình 126 email mỗi ngày vào cuối năm 2019.

Bảng 1.1 Thống kê lượng email gửi đi hàng ngày trên toàn thế giới [4]

Daily Email Traffic	2015	2016	2017	2018	2019
Total Worldwide Emails Sent/Received Per Day (B)	**205.6**	**215.3**	**225.3**	**235.6**	**246.5**
% Growth		*5%*	*5%*	*5%*	*5%*
Business Emails Sent/Received Per Day (B)	**112.5**	**116.4**	**120.4**	**124.5**	**128.8**
% Growth		*3%*	*3%*	*3%*	*3%*
Consumer Emails Sent/Received Per Day (B)	**93.1**	**98.9**	**104.9**	**111.1**	**117.7**
% Growth		*6%*	*6%*	*6%*	*6%*

Table 2: Worldwide Daily Email Traffic (B), 2015-2019

Có hơn 3,9 tỷ người dùng email trên toàn thế giới. Năm nay, số lượng người dùng email đạt mốc 3,9 tỷ, điều đó có nghĩa là hơn 50% dân số thế giới hiện đang sử dụng email. Năm 2020, số lượng người dùng email sẽ tăng lên 4 tỷ. Theo số liệu thống kê

tiếp thị qua email gần đây, tốc độ tăng trưởng người dùng dự đoán trong bốn năm tới là 3%, tức là khoảng 100 triệu người dùng mỗi năm. Vì vậy, vào năm 2023, số lượng người dùng email trên toàn thế giới sẽ xấp xỉ 4,3 tỷ. Có khoảng 5,59 tỷ tài khoản email đang hoạt động. Nhiều người dùng email có nhiều email, đó là lý do tại sao số lượng tài khoản email đang hoạt động cao hơn gần 1,5 lần so với số lượng người dùng email. Trong khi có 5,59 tỷ tài khoản, chỉ có 3,9 tỷ người dùng email. Một trong những lý do chính khiến hầu hết mọi người có hai tài khoản email là tách email công việc khỏi tài khoản cá nhân. Có 293,6 tỷ email được gửi mỗi ngày. Con số này bao gồm cả email của người tiêu dùng và doanh nghiệp. Năm tới, Tập đoàn Radicati dự đoán, số lượng email được gửi hàng ngày sẽ tăng 4,4%, dẫn đến 306,4 tỷ email mỗi ngày. Đến năm 2023, tốc độ tăng trưởng sẽ giảm 0,2%, khi số lượng email đang lưu hành sẽ ở mức 347,3 tỷ email được gửi hàng ngày.

1.8 Kết luận chương

Qua những thống kê trên, hàng ngày mỗi người dùng thường nhận được rất nhiều thư điện tử khác nhau nên sẽ khó khăn trong việc xác định và nhận dạng những thư điện tử nào quan trọng cần đọc và trả lời sớm, những thư nào có thể chỉ để theo dõi. Vì vậy ta phải dùng đến khái niệm “Mức độ ưu tiên” với thư điện tử. Theo định nghĩa tiếng Anh “Mức độ ưu tiên” - “Priority” được sử dụng để so sánh hai vật hoặc hai điều kiện, khi mà một vật/điều kiện phải quan tâm nhiều hơn những vật/điều kiện khác và phải được giải quyết trước khi chuyển sang (những) vật/điều kiện tiếp theo. Công cụ hỗ trợ nhận dạng và phân loại mức độ ưu tiên cho thư điện tử là cần thiết. Chương tiếp theo của luận văn xin được trình bày phương pháp đánh giá độ ưu tiên cho thư điện tử.

CHƯƠNG 2 – ĐÁNH GIÁ MỨC ĐỘ ƯU TIÊN CỦA THƯ ĐIỆN TỬ

Chương 2 sẽ trình bày phương pháp nhằm đánh giá, phân loại mức độ ưu tiên cho thư điện tử. Trước khi đánh giá mức độ ưu tiên của thư, bước đầu tiên là loại bỏ thư rác. Sau đây luận văn sẽ trình bày về một số phương pháp lọc thư rác để hỗ trợ cho việc phân loại mức độ ưu tiên của thư điện tử.

2.1 Một số công nghệ phân loại thư rác hỗ trợ phân loại mức độ ưu tiên của thư điện tử

2.1.1 Định nghĩa 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 (Unsolicited Commercial Email-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 (Unsolicited Bulk Emai -UBE).

Nội dung thông dụng nhất về định nghĩa thư rác: Thư rác (spam mail) là những bức thư điện tử không yêu cầu, không mong muốn và được gửi hàng loạt tới người nhận.

2.1.2 Các phương pháp loc thư rác

2.1.2.1 Phương pháp dùng danh sách trắng đen

Danh sách đen (Blacklist) Người ta lập ra một danh sách các địa chỉ gửi thư rác. Các nhà cung cấp dịch vụ thư điện tử (ISP) 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. Một số danh sách đen điển hình được lập ra như: 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 đã làm 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 (Whitelist) Danh sách các địa chỉ tin cậy (Safe Sender List), 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: 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.

2.1.2.2 Phương pháp lọc theo từ khóa

Phương pháp lọc thư rác theo từ khóa là một phương pháp truyền thống trong việc lọc thư rác. Người ta dựa vào những từ hay cụm từ có trong đầu đề của thư (subject) và nội dung của thư để lọc.Khi một thư mới được gửi tới hòm thư của bạn, bạn 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. Vì 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. Những cụm từ, câu chữ như *"Silk ties"* (Cà vạt lụa) hoặc *"Eliminate debt"* (Xoá nợ) xuất hiện thường xuyên trên spam và được coi những cụm từ thường xuyên xuất hiện nhất trong các bức thư không mong muốn. Các đặc điểm nội dung khác để nhận diện spam như yêu cầu hành động như *"Fin out how, click here"* hoặc thông báo huỷ như *"If you want to be removed from our mailing lists…"*.

Một vài năm gần đây, 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 những kẻ gửi thư rác này đã thay

đổi cách viết nội dung của thư rác nhằm làm cho thư rác của chúng có thể "xuyên qua" các bộ lọc. Điều này có thể giải thích tại sao bạn nhận nhiều thư với những từ như *"Vi@gra", "Mort.gage", "L|0|a|n|$"* hay những tranh ảnh được nhúng vào trong thư. Phương pháp này có một số ưu điểm và nhược điểm sau.

Ưu điểm:

- Tính thích nghi: Người dùng có thể dễ dàng biến đổi bộ lọc của mình để nó có thể lọc các kiểu thư rác mà người đó đang phải nhận và điều quan trọng là nó không cản trở (thích nghi) các từ và các cụm từ được sử dụng hàng ngày trong kinh doanh thương mại với bạn bè hay những người thân quen.

Nhược điểm:

- Yêu cầu nhiều tiến trình xử lý bằng tay để điều chỉnh và duy trì bộ lọc được hiệu quả. Để có thể đánh lừa các bộ lọc, những kẻ gửi thư rác luôn luôn thay đổi hình thức nội dung của thư rác, do đó những bộ lọc mở rộng phải được tạo ra để chống lại điều đó.

2.1.2.3 Phương pháp lọc dựa trên mạng xã hội

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 email, cung được thêm vào giữa 2 node A và B nếu giữa A và B có sự trao đổi thư qua lại). Người ta đã sử dụng một số tính chất đặc trưng của mạng xã hội để xây dựng một công cụ lọc thư rác. Đầu tiên, người ta phân đồ thị thành các thành phần con rồi tính độ phân cụm cho từng thành phần này. Mỗi thành phần con là một đồ thị mạng xã hội của một node, bao gồm tất cả các node xung quanh là "node hàng xóm" (các node có cung liên kết với node này) và những cung liên kết giữa các node hàng xóm này với nhau. Nếu thành phần nào có độ phân cụm thấp thì node tương ứng với thành phần đó là một địa chỉ gửi thư rác. Trong thành phần mạng xã hội của những node gửi thư rác, những node hàng xóm

của nó thường là những node rất ngẫu nhiên, không có mối quan hệ (không có sự trao đổi email qua lại với nhau) nên độ phân cụm của mạng xã hội của những node này rất thấp. Ngược lại, mạng xã hội ứng với những người dùng bình thường có độ phân cụm cao hơn. Dựa vào độ phân cụm, người ta tạo được danh sách đen (Blacklist) gồm địa chỉ email tương ứng với những node có độ phân cụm rất thấp, danh sách trắng (Whitelist) ứng với node có độ phân cụm cao, số node còn lại sẽ được đưa vào danh sách cần xem xét (Greylist). Phương pháp này có thể phân loại được 53% tổng số email một cách chính xác là ham hay spam. Nhược điểm của phương pháp là những spammer có thể xây dựng mạng xã hội của chính họ nên khó có thể phát hiện ra.

2.1.2.4 Phương pháp lọc thư rác dùng chuỗi hỏi đáp (Challenge/Response filters)

Đặc trưng của phương pháp này là khả năng 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 kiểm tra này được đặt tên là *"Turing Test"* do nhà toán học người anh tên là Alan Turing nghĩ ra.

Một vài dịch vụ Internet tự động xử lý hàm Challenge/Response này cho người dùng. Chương trình yêu cầu người gửi thư phải vào website của họ và trả lời một số câu hỏi đơn giản để xác minh về email mà người này đã gửi.Việc này chỉ được yêu cầu trong lần gửi thư đầu tiên. Đáp ứng hàm Challenge/Response này rất đơn giản và không có gì khó khăn khi một người dùng 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.

Ưu điểm:

- Đối với một số người dùng có lượng thư trao đổi thấp, hệ thống đơn lẻ này có thể chấp nhận được như một phương pháp hoàn hảo để loại trừ hoàn toàn thư rác từ hòm thư của họ.

Nhược điểm:

- Người dùng thường cảm thấy không thuận tiện. Những kẻ gửi thư rác có thể viết những chương trình trả lời tự động những chuỗi hỏi đáp trên.

2.1.2.5 Lọc thư rác dựa trên xác suất thống kê và học máy

Đầu tiên sẽ phân loại các bức thư thành thư rác và thư hợp lệ. Một thuật toán được áp dụng để trích chọn và đánh trọng số cho các đặc trưng của thư rác theo một cách nào đó (thường là áp dụng công thức xác suất). Sau khi trích chọn đặc trưng, hai tập thư rác và thư hợp lệ sẽ được sử dụng để huấn luyện một bộ phân loại tự động. Quá trình huấn luyện dựa trên một phương pháp học máy.

Tỉ lệ chặn thư rác của bộ lọc sử dụng phương pháp này rất cao, khoảng 99%. Chương trình SpamProbe có thể đạt tới tỉ lệ lọc thư rác tới 99.9%. Các phương pháp học máy và xác suất thống kê cho phép phân loại cả những thư rác chưa từng xuất hiện trước đó. Phương pháp này còn có tỉ lệ chặn thư hợp pháp rất thấp, thấp hơn nhiều so với phương pháp heuristic. Nhược điểm của phương pháp này là phải có một tập hợp các thư để huấn luyện.

Hiệu suất của bộ lọc sẽ phụ thuộc nhiều vào tập huấn luyện này. Tập dữ liệu càng lớn càng chứa nhiều dạng khác nhau thì kết quả phân loại về sau sẽ càng chính xác.

2.1.2.6 Lọc bằng phần mềm SpamAssassin

Phương pháp lọc bằng phần mềm SpamAssassin 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, nó dùng đầu đề (header) và nội dung của thư rồi dựa trên tập các luật được xác định trước và những kí hiệu dấu câu đặc biệt (tell-tale), xem thư có vi phạm các luật này không sau đó tính điểm đối với từng thư. Từ kết quả thu được, xác định được một thư là thư rác hay thư thường.

Ưu điểm:

- Tỉ lệ lọc thư rác của phương pháp SpamAssassin rất cao

Nhược điểm:

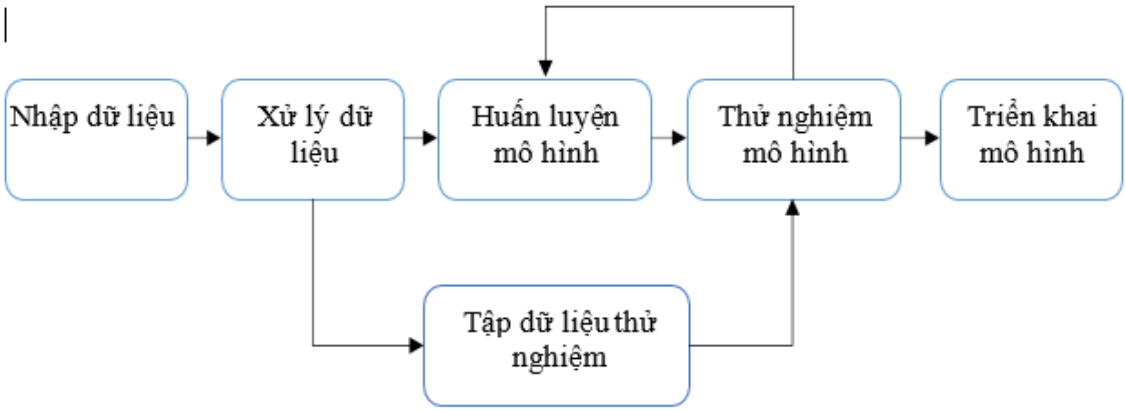
- Phương pháp SpamAssassin tiêu tốn khá nhiều tài nguyên (khối điều khiển trung tâm CPU, bộ nhớ, thời gian xử lý) của máy chủ, đặc biệt khi phải xử lý những email có dung lượng lớn. Cấu hình để SpamAssassin hoạt động tốt, đồng thời giảm nhẹ sự tiêu tốn tài nguyên cho máy chủ là vấn đề quan trọng.

2.2 Tổng quan về học máy

2.2.1 Khái niệm cơ bản

Sự phát triển nhanh chóng của các kỹ thuật khai phá dữ liệu đã đưa Học máy thành một lĩnh vực riêng biệt của Khoa học máy tính. Học máy là một lĩnh vực của Trí tuệ nhân tạo liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể [10]. Nó được đề cập lần đầu bởi Arthur Samuel vào năm 1959, như là “một lĩnh vực nghiên cứu mà ở đó máy tính có khả năng tự học mà không cần phải lập trình”. Tom Mitchell, giáo sư nổi tiếng của Đại học Carnegie Mellon University định nghĩa cụ thể và chuẩn mực hơn như sau: "Một chương trình máy tính CT được xem là học cách thực thi một lớp nhiệm vụ NV thông qua trải nghiệm KN, đối với thang đo năng lực NL nếu như dùng NL ta đo thấy năng lực thực thi của chương trình có tiến bộ sau khi trải qua KN"[6].

Ý tưởng cơ bản của học máy là huấn luyện mô hình, dựa trên một số thuật toán, để thực thi các nhiệm vụ cụ thể như: phân lớp, phân cụm, hồi quy,… Dựa trên tập dữ liệu đầu vào, mô hình sẽ được huấn luyện và sau đó được dùng để dự đoán. Một số ứng dụng thực tế như: cho tập dữ liệu về thuộc tính nhà cửa như số phòng, kích thước, giá tiền, dự đoán giá của một ngôi nhà mới; dựa vào hai tập dữ liệu về hình ảnh của những người khỏe mạnh và hình ảnh của người bị u, phân loại các hình ảnh mới; phân cụm các bức ảnh động vật từ tập dữ liệu chưa được sắp xếp.



Hình 2.1 Quy trình học máy

Hình 2.1 mô tả quy trình chung của một tiến trình học máy. Tiến trình bao gồm 5 bước sau:

- Nhập dữ liệu. Đầu tiên, tập dữ liệu được tải lên từ tệp tin và lưu vào bộ nhớ.
- Xử lý dữ liệu. Tại bước này, dữ liệu được tải lên từ bước 1 sẽ được chuyển đổi, làm sạch và chuẩn hóa để phù hợp với thuật toán. Dữ liệu được chuyển đổi để nằm trong cùng một giới hạn, có cùng định dạng,… Quá trình trích xuất và lựa chọn đặc trưng cũng diễn ra ở bước này. Sau đó, dữ liệu được chia ra thành hai tập – 'tập huấn luyện' và 'tập thử nghiệm'. Dữ liệu từ tập huấn luyện được dùng để xây dựng mô hình, sau đó sẽ được đánh giá thông qua tập thử nghiệm.
- Huấn luyện mô hình. Xây dựng mô hình dựa trên thuật toán đã lựa chọn.
- Thử nghiệm mô hình. Mô hình đã được xây dựng và huấn luyện ở bước 3 sẽ được thử nghiệm thông qua tập dữ liệu thử nghiệm, và kết quả sinh ra được dùng để xây dựng nên một mô hình mới, quá trình diễn ra lặp đi lặp lại này được gọi là "học" từ các mô hình trước đó.
- Triển khai mô hình. Ở bước này, mô hình tốt nhất sẽ được lựa chọn để triển khai (sau một số bước lặp nhất định hoặc khi đạt được kết quả cần thiết).

2.2.2 Trích chọn đặc trưng

Trong các ví dụ đã đưa ra ở trên, cần phải trích xuất các thuộc tính từ dữ liệu đầu vào để đưa vào thuật toán. Ví dụ, với trường hợp tính giá nhà, dữ liệu có thể được biểu diễn dưới dạng ma trận đa chiều, với mỗi cột là một thuộc tính và mỗi dòng là giá trị của thuộc tính đó. Trong trường hợp hình ảnh, dữ liệu có thể được biểu diễn dưới dạng giá trị RGB của mỗi pixel. Các thuộc tính này được gọi là đặc trưng, và ma trận là vector đặc trưng. Quá trình trích xuất dữ liệu từ tệp tin được gọi là trích xuất đặc trưng. Mục đích của quá trình này là thu được một tập dữ liệu chi tiết và không dư thừa. Các đặc trưng phải biểu diễn thông tin quan trọng và liên quan tới tập dữ liệu nếu không kết quả dự đoán sẽ không chính xác. Vì thế, trích xuất đặc trưng là một nhiệm vụ không rõ ràng, cần rất nhiều nghiên cứu và thử nghiệm. Hơn nữa, với mỗi lĩnh vực, đặc trưng của dữ liệu cũng khác nhau, nên không có phương pháp chung cho việc trích xuất đặc trưng. Một yêu cầu quan trọng khác đối với một tập đặc trưng hợp lệ là không dư thừa. Có các đặc trưng dư thừa như các đặc trưng nói về cùng một thông tin, hoặc các thuộc tính dư thừa, có thể gây ảnh hưởng đến thuật toán, từ đó dẫn đến kết quả không chính xác. Ngoài ra, nếu dữ liệu đầu vào quá lớn (quá nhiều đặc trưng), nó có thể được chuyển đổi thành các vector đặc trưng nhỏ hơn (giảm bớt số đặc trưng). Quá trình này gọi là chọn lọc đặc trưng. Kết quả của quá trình này là lựa chọn ra các đặc trưng bao quát nhất từ tập dữ liệu mà không làm giảm độ chính xác.

2.2.3 Phân loại học máy

2.2.3.1 Học có giám sát và học không giám sát

Đối với **học có giám sát**, việc học được dựa trên các dữ liệu được dán nhãn. Trong trường hợp này, chúng ta sẽ dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Ví dụ về định giá nhà ở trên là một trường hợp học có giám sát: tập dữ liệu cho trước bao gồm các ngôi nhà, thuộc tính và mức giá của chúng. Ở đây, nhà và thuộc tính là dữ liệu, mức giá là nhãn; cần phải dự

đoán giá của một ngôi nhà mới.

Một ví dụ khác, trong nhận dạng chữ viết tay, có ảnh của hàng nghìn ví dụ của mỗi chữ số được viết bởi nhiều người khác nhau. Các bức ảnh này được đưa vào trong một thuật toán và chỉ cho nó biết mỗi bức ảnh tương ứng với chữ số nào. Sau khi thuật toán tạo ra một mô hình, tức một hàm số mà đầu vào là một bức ảnh và đầu ra là một chữ số, khi nhận được một bức ảnh mới mà mô hình chưa nhìn thấy bao giờ, nó sẽ dự đoán bức ảnh đó chứa chữ số nào.

Hình 2.2 Bộ cơ sở dữ liệu của chữ số viết tay

(Nguồn: Simple Neural Network implementation in Ruby)

Học có giám sát được chia nhỏ thành hai loại chính:

- **Phân lớp (Classification)**. Dựa vào tập dữ liệu đã được dán nhãn, với mỗi nhãn định nghĩa một lớp, dự đoán xem một dữ liệu mới chưa biết thuộc vào lớp nào. Số lớp thường nhỏ và hữu hạn. Ví dụ: Gmail xác định xem một email có phải là spam hay không; các hãng tín dụng xác định xem một khách hàng có khả năng thanh toán nợ hay không. Ví dụ về chữ viết tay ở trên cũng thuộc loại này.
- **Hồi quy (Regression)**. Nhãn không được chia thành các nhóm mà là một giá trị

thực cụ thể. Ví dụ về dự đoán mức giá của một ngôi nhà thuộc loại này.

Ngược lại với học có giám sát, trong **học không giám sát**, dữ liệu không được dán nhãn. Ở đây, mục tiêu là tìm một số mẫu trong tập dữ liệu chưa được phân loại, thay vì dự đoán một số giá trị. Một bài toán quen thuộc của học không giám sát là phân cụm (clustering). **Phân cụm** là việc tìm kiếm điểm chung giữa các dữ liệu trong tập dữ liệu và chia chúng thành các cụm tương ứng dựa vào điểm chung này. Ví dụ: phân nhóm khách hàng dựa trên hành vi mua hàng.[12]

2.2.3.2 Một số kỹ thuật học máy

K-Nearest Neighbors

K-Nearest Neighbors (KNN) là một trong những thuật toán đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong số các thuật toán của học máy. KNN là một thuật toán phi tham số, tức là nó không đưa ra bất kỳ dự đoán nào về cấu trúc của dữ liệu. Khi huấn luyện, thuật toán này không học một điều gì từ dữ liệu huấn luyện (đây cũng là lý do thuật toán này được xếp vào loại lazy learning).

KNN có thể áp dụng được vào cả hai loại của bài toán học có giám sát là Phân lớp và Hồi quy. Trong cả hai bài toán, kết quả dự đoán của một điểm dữ liệu mới được suy ra trực tiếp từ *k* điểm dữ liệu gần nhất trong tập dữ liệu huấn luyện. Đối với bài toán phân lớp, kết quả đầu ra sẽ là lớp mà dữ liệu thuộc về, dựa trên việc bình chọn (majority vote) của *k* điểm gần nhất. Trong bài toán hồi quy, đầu ra của một điểm dữ liệu sẽ bằng trung bình của đầu ra của *k* điểm gần nhất.

Có nhiều phương pháp đo khoảng cách giữa các điểm để tìm ra điểm gần nhất. Các phương pháp phổ biến nhất bao gồm khoảng cách Hamming, khoảng cách Manhattan, khoảng cách Minkowski:

$$\textit{Khoảng cách Hamming: } d_{ij} = \sum_{k=1}^{p} | x_{ik} - x_{jk} | \qquad (2.1)$$

$$\textit{Khoảng cách Manhattan: } d_1(p,q) = ||p-q||_1 = \sum_{i=1}^{n} | p_i - q_i | \qquad (2.2)$$

$$\textit{Khoảng cách Minkowski} = (\sum_{i=1}^{n} | x_i - y_i|^p)^{1/p} \qquad (2.3)$$

Phương pháp phổ biến nhất đối với các biến liên tục là khoảng cách Euclidean, được

định nghĩa bởi công thức (2.4) dưới đây:

$$d_{Euclidean} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}; \textit{ p và q là các điểm trong không gian n} \quad (2.4)$$

Khoảng cách Euclidean phù hợp với các bài toán có đặc trưng cùng kiểu, với các bài toán có đặc trưng thuộc nhiều kiểu khác nhau, nên sử dụng khoảng cách Manhattan.

Đối với các bài toán phân lớp, đầu ra có thể được biểu diễn dưới dạng tập các xác suất mà mỗi điểm thuộc về lớp nào đó. Ví dụ, với bài toán nhị phân, xác suất có thể được tính theo công thức $(0) = \frac{N_0}{N_0 + N_1}$, với *P(0)* là xác suất một điểm thuộc lớp 0 và N_0, N_1 là số các điểm lân cận thuộc lớp 0 hoặc 1.[9]

Giá trị của *k* cũng đóng vai trò quan trọng trong độ chính xác của thuật toán dự đoán. Tuy nhiên, việc chọn *k* lại là một nhiệm vụ không hề đơn giản. Nếu *k* quá nhỏ thì độ chính xác sẽ giảm, đặc biệt là với các tập dữ liệu có nhiều giá trị nhiễu. Còn *k* quá lớn sẽ giảm hiệu năng của thuật toán. Ngoài ra, nếu giá trị *k* quá lớn sẽ khiến cho mô hình bị quá tải, làm cho ranh giới giữa các lớp trở nên ít khác biệt, dẫn đến độ chính xác cũng bị giảm. Thông thường, *k* thường được chọn theo công thức (2.5) dưới đây:

$$k = \sqrt{n} \quad (2.5)$$

Với các bài toán có số lớp là chẵn, nên chọn *k* lẻ để quá trình bình chọn không có kết quả hòa.

Hạn chế của thuật toán KNN là hiệu suất kém trên các bộ dữ liệu được phân phối không đồng đều. Nếu một lớp có nhiều điểm dữ liệu hơn các lớp khác, một điểm dữ liệu mới sẽ có nhiều điểm lân cận thuộc lớp đó và từ đó dẫn đến kết quả dự đoán không chính xác [5].

Thuật toán Random Forest

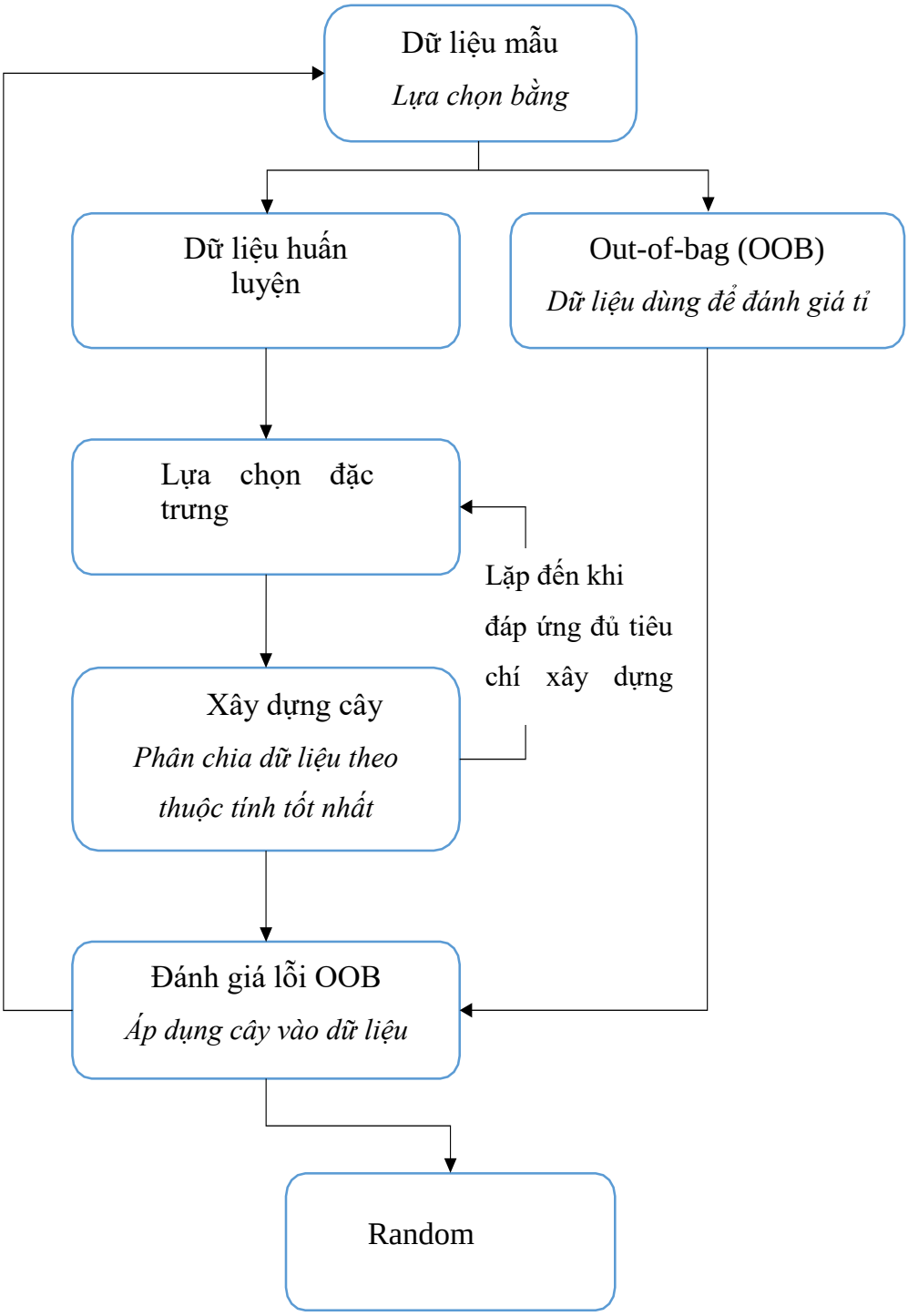
Random Forest là một trong các thuật toán thông dụng nhất trong học máy. Thuật toán này hầu như không yêu cầu tiền xử lý dữ liệu và lập mô hình nhưng thường cho kết quả khá chính xác. Random Forest dựa trên tính ngẫu nhiên (random) và được tạo

nên từ nhiều cây quyết định (forest – "rừng").

Ý tưởng cơ bản của thuật toán là xây dựng các cây quyết định dựa trên các tập con độc lập nhau thuộc tập dữ liệu cho trước. Tại mỗi nút, một số giá trị đặc trưng sẽ được lựa chọn ngẫu nhiên cho tới khi tìm thấy cách phân chia tốt nhất. Tóm lại, thuật toán có thể được mô tả như sau :

- Các cây được xây dựng dựa trên 2/3 dữ liệu của tập dữ liệu huấn luyện (62.3%). Dữ liệu được lựa chọn ngẫu nhiên.
- Một số biến dự đoán được chọn ngẫu nhiên từ tổng số các biến dự đoán. Sau đó, cách phân chia tốt nhất của các biến được lựa chọn sẽ được dùng để phân chia nút. Theo mặc định, số lượng biến được chọn sẽ là căn bậc hai của tổng số các thuộc tính dùng để dự đoán và không đổi đối với các cây.
- Tỉ lệ dự đoán sai được tính toán dựa vào phần dữ liệu còn lại (dữ liệu out-of-bag).
- Mỗi cây huấn luyện sẽ đưa ra một kết quả phân loại, được gọi là "bỏ phiếu". Lớp nhận được nhiều "phiếu" nhất sẽ được chọn là kết quả cuối cùng. Sơ đồ thuật toán được thể hiện ở hình 2.4

Random Forest thừa kế rất nhiều ưu điểm của thuật toán cây quyết định. Random Forest có thể dùng cho cả hai bài toán phân loại và hồi quy, bởi nó đơn giản và dễ thích nghi, kết quả đưa ra cũng chính xác hơn. Tuy nhiên, không như cây quyết định, cấu trúc của Random Forest rất phức tạp nên không thể hiểu được cơ chế hoạt động bên trong của thuật toán. Ngoài ra, Random Forest cũng ổn định hơn so với cây quyết định. Đối với cây quyết định, chỉ cần dữ liệu bị sửa đổi một chút thì cả cây cũng sẽ bị thay đổi, làm giảm độ chính xác. Còn với thuật toán Random Forest, do nó được kết hợp từ rất nhiều cây quyết định nên nó sẽ ổn định hơn .[1]



Hình 2.3 Sơ đồ thuật toán Random Forest

Thuật toán Logistic Regression

Phương pháp hồi quy logistic là một mô hình hồi quy nhằm dự đoán giá trị đầu ra *rời rạc* (*discrete target variable*) *y* ứng với một véc-tơ đầu vào **x**. Việc này tương đương với chuyện phân loại các đầu vào **x** vào các nhóm *y* tương ứng. Ví dụ, xem một bức ảnh có chứa một con mèo hay không. Thì ở đây ta coi đầu ra *y* = 1 nếu bước ảnh có một con mèo và *y* = 0 nếu bức ảnh không có con mèo nào. Đầu vào **x** ở đây sẽ là các pixel một bức ảnh đầu vào. Sử dụng phương pháp thống kê ta có thể coi rằng khả năng đầu vào x nằm trong nhóm y_0 là xác xuất nhóm y_0 khi biết x: p(y_0|x). Ta có hàm **sigmoid** (logistic sigmoid function).[6]

p(y_0|**x**)=$\frac{1}{1+\exp(-a)}$=σ(a) (2.6)

Vận dụng thuyết phân phối chuẩn, ta có thể chỉ ra rằng: a=$\boldsymbol{w^T x+w_0}$

Đặt $\boldsymbol{x_0}$=[1,…..,1] ta có thể viết gọn : a=$\boldsymbol{w^T x}$

Thay vào công thức (2.6) bên trên ta có : p(y_0|**x**)=$\frac{1}{1+\exp(-a)}$= $\sigma(\boldsymbol{w^T x})$ Trong đó x là thuộc tính đầu vào còn w là trọng số tương ứng.

Ta phải tối ưu hàm mất mát

$$J(\mathbf{w}) = -\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} log\sigma^{(i)} + (1-y^{(i)}) log(1-\sigma^{(i)})\right)$$

Theo phương pháp Gradient Descent ta cập nhật tham số sau mỗi vòng lặp [11]:

$$\mathbf{w} = \mathbf{w} - \eta\frac{1}{m}\mathbf{X}^{\intercal}(\sigma - \mathbf{y})$$

2.2.4 Thuật toán khai phá dữ liệu văn bản

Thuật toán TF-IDF

TF-IDF (Term Frequency – Inverse Document Frequency) là một kĩ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị cao thể hiện độ quan trọng cao và nó phụ thuộc

vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu. Một vài biến thể của TF-IDF thường được sử dụng trong các hệ thống tìm kiếm như một công cụ chính để đánh giá và sắp xếp văn bản dựa vào truy vấn của người dùng. TF-IDF cũng được sử dụng để lọc những từ stopwords trong các bài toán như tóm tắt văn bản và phân loại văn bản.

TF: Term Frequency(Tần suất xuất hiện của từ) là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Như vậy, term frequency thường được chia cho độ dài văn bản(tổng số từ trong một văn bản).

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

Trong đó:

tf(t, d): tần suất xuất hiện của từ t trong văn bản d

f(t, d): Số lần xuất hiện của từ t trong văn bản d

max({f(w, d) : w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d.

IDF: Inverse Document Frequency(Nghịch đảo tần suất của văn bản), giúp đánh giá tầm quan trọng của một từ . Khi tính toán TF , tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng một số từ như “is”, “of” và “that” thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế chúng ta cần giảm độ quan trọng của những từ này xuống.

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

Trong đó:

idf(t, D): giá trị idf của từ t trong tập văn bản

|D|: Tổng số văn bản trong tập D

|{d ∈ D : t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.

Cơ số logarit trong công thức này không thay đổi giá trị idf của từ mà chỉ thu hẹp khoảng giá trị của từ đó. Vì thay đổi cơ số sẽ dẫn đến việc giá trị của các từ thay đổi bởi một số nhất định và tỷ lệ giữa các trọng lượng với nhau sẽ không thay đổi. (nói cách khác, thay đổi cơ số sẽ không ảnh hưởng đến tỷ lệ giữa các giá trị IDF). Việc sử dụng logarit nhằm giúp giá trị tf-idf của một từ nhỏ hơn, do chúng ta có công thức tính tf-idf của một từ trong 1 văn bản là tích của tf và idf của từ đó. Cụ thể, chúng ta có công thức tính TF-IDF hoàn chỉnh như sau:

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

2.3 Phương pháp phân loại độ ưu tiên của thư điện tử

2.3.1 Các thành phần của một thư điện tử

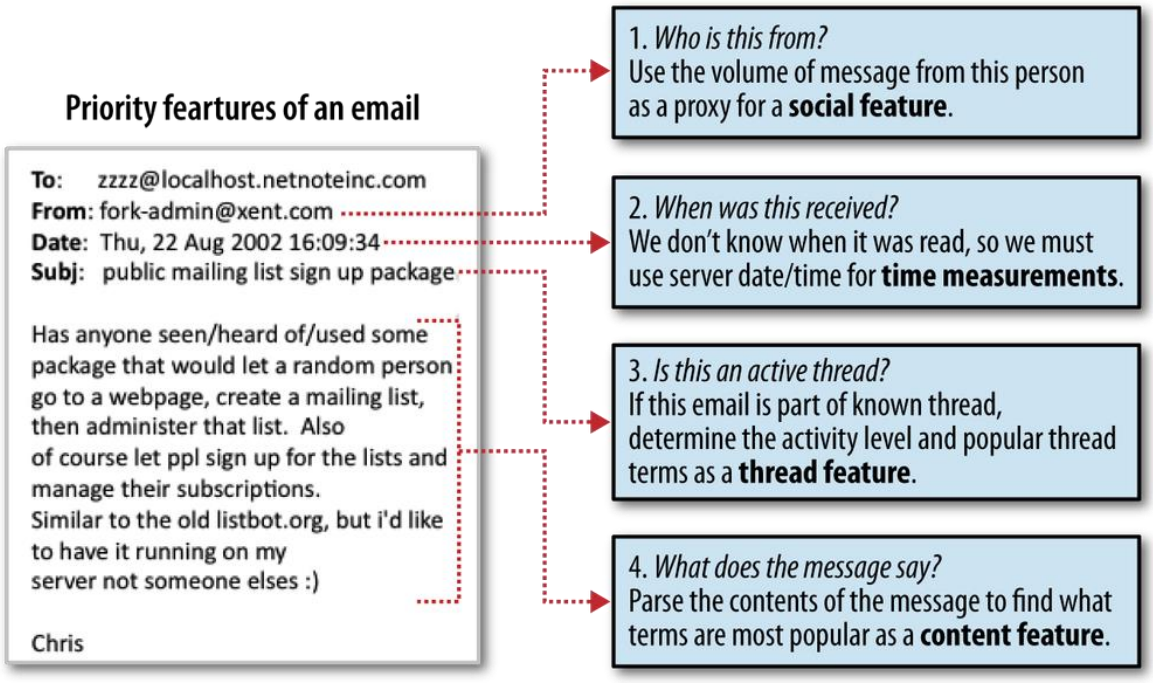
Các thành phần của một thư điện tử thông thường bao gồm người gửi, người nhận, thời gian, tiêu đề, phần nội dung, các tệp tin đính kèm. Trong công việc hằng ngày, ta sẽ nhận được rất nhiều email nên có thể đọc lướt qua tiêu đề để nắm được nội dung sơ lược và quyết định đọc email nào trước. Do đó, tiêu đề thường được viết cụ thể và ngắn gọn điều quan trọng chứa đựng trong nội dung. Nội dung của thư là phần người viết và người đọc trao đổi với nhau, chứa thông tin mà bức thư muốn truyền tải. Thư điện tử khác với thư thường nên người dùng có thể gửi thêm các tập kèm theo phục vụ cho việc truyền tải thông tin được cụ thể rõ ràng hơn. Thời gian gửi theo định dạng ngày tháng năm, các tiêu đề, nội dung ở dạng văn bản. Người gửi người nhận là các địa chỉ

hòm thư của người dùng.

2.3.2 Lựa chọn đặc trưng để xét độ ưu tiên

Thư điện tử là một phương tiện dựa trên sự trao đổi qua lại. Mọi người gửi và nhận thư theo thời gian. Một bức thư điện tử quan trọng hay không phải dựa vào các đặc trưng của cuộc trao đổi giữa người gửi và người nhận chứ không đơn thuần chỉ là dựa vào nội dung của thư đó. Ta dựa vào các đặc trưng của thư để dự đoán xem người dùng sẽ tương tác thế nào với thư nhận được trong thời gian sắp tới. Đó là mục đích của phương pháp phân loại độ ưu tiên thư điện tử. Có tới hàng trăm các đặc trưng của thư có thể được xét tới. Những đặc trưng nào có nhiều giá trị và đáng được quan tâm hơn . Thư điện tử là phương tiện dựa trên giao dịch, các đăc trưng xã hội sẽ là tối quan trọng trong việc đánh giá tầm quan trọng của thư [3]. Nó được gửi đến từ ai. Rõ ràng một người nhận được một khối lượng lớn các tin nhắn email từ một địa chỉ nhất định, thì có thể người dùng có kết nối xã hội mạnh mẽ với người gửi. Nếu người dùng có tần suất phản hồi thường xuyên với địa chỉ email người gửi thì càng chắc chắn kết nối xã hội mạnh mẽ giữa 2 người. Vậy đặc trưng đáng xem xét là địa chỉ người gửi, người nhận, tần suất phản hồi giữa họ. Đặc trưng quan trọng mà ta chú ý là thời gian nhận được email. Tiếp theo xem xét email đó có đang ở trong một luồng email nào đó không. Những email cùng luồng thường cùng chủ đề, và có thể là để trả lời lại một thư khác. Ví dụ như ở Gmail thì nó được đánh dấu là “RE”. Ta trích xuất đặc trưng từ nội dung của thư bằng các kỹ thuật khai thác văn bản. Cụ thể, nếu có các thuật ngữ phổ biến trong các chủ đề và nội dung email mà người dùng nhận được, thì các email trong tương lai có chứa các thuật ngữ này ở trong chủ đề và nội dung có thể quan trọng hơn thuật ngữ không xuất hiện. Đây là một kỹ thuật phổ biến và được đề cập ngắn gọn trong phần mô tả về hộp thư ưu tiên Google Google. Khi xét đến các đặc trưng nội dung dựa cả chủ đề và nội dung email, có một số thuật ngữ ít quan trọng hơn trong một chủ đề email so với khi ở trong nội dung. Do đó, không nên coi tầm quan trọng tương

đối của các thuật ngữ phổ biến trong hai tính năng này như nhau. [2]



Hình 2.4 Các đặc trưng cần quan tâm [2]

2.3.3 Cách tính trọng số dựa vào các đặc trưng

Đặc trưng tần suất thư gửi đến: Đếm số lần xuất hiện của mỗi địa chỉ email trong số email dùng để trainning. Với số lần xuất hiện của một địa chỉ email là x_i. Trọng số thứ nhất: $w_1 = \log_{10} x_i$

Đặc trưng tần suất thư phản hồi: Lọc các email là email phản hồi. Gọi số lần xuất hiện một địa chỉ email trong số các email phản hồi là x_j. Trọng số thứ hai: $w_2 = \log_{10} x_j$

Đặc trưng tỉ lệ số lượng thư trên thời gian của luồng email: Lọc các thread thư, Loại các thread không có reply, tính tổng thời gian của thread đó. Với thread i. Gọi tổng thời gian của thread là t với t tính bằng giây, số lượng thư qua lại của luồng thư i là n. Trọng số thứ ba : $w_3 = \log_{10} \frac{n}{t}$

Sử dụng phương pháp TF-IDF, tính được độ quan trọng của các từ trong nội dung của các email trong tập mẫu. Với m là số lượng từ của nội dung thư, x_j là độ quan trọng của từng từ.

Đặc trưng độ quan trọng của nội dung thư: Trọng số thứ tư là : $w_4 = \log_{10} \sum_{m}^{i=1} x_j$

Đặc trưng độ quan trọng của tiêu đề: Với n là số lượng từ của tiêu đề của mỗi thư, x_i là độ quan trọng của từng từ. Trọng số thứ năm là $w_5 = \log_{10} \sum_{n}^{i=1} x_i$. [2]

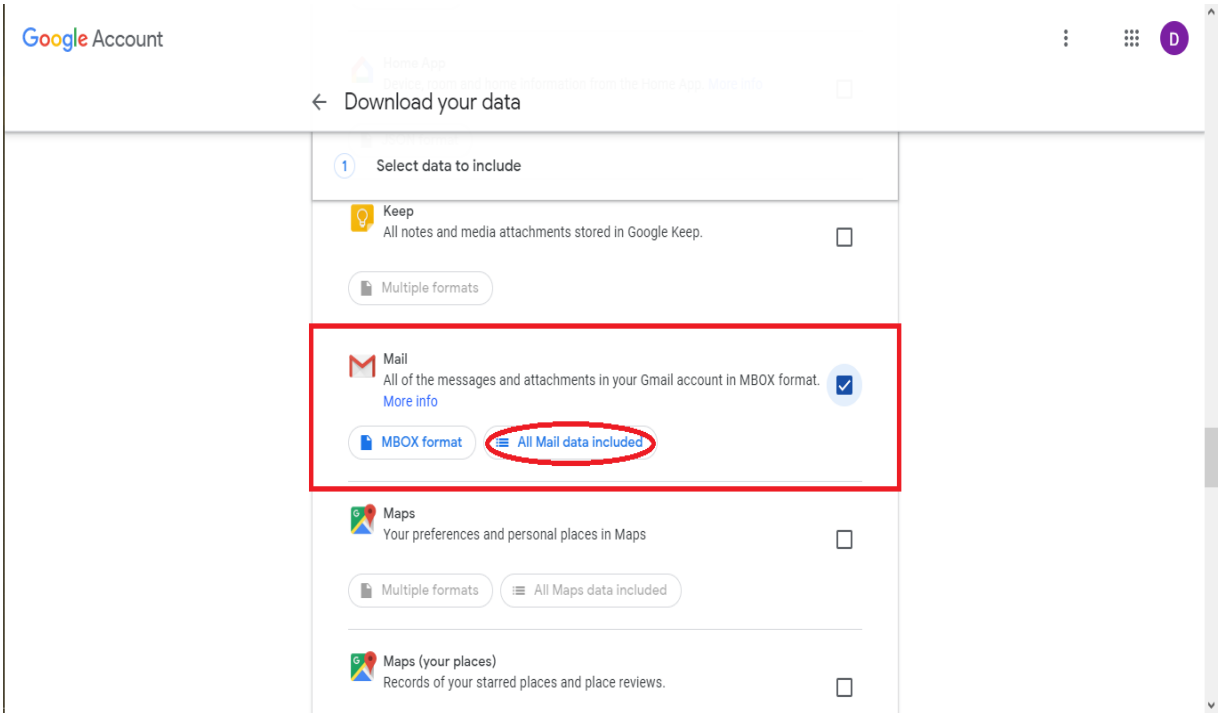
CHƯƠNG 3 - CÀI ĐẶT VÀ THỬ NGHIỆM

Chương 3 sẽ tiến hành áp dụng phương pháp phân loại đã giới thiệu ở chương 2 vào tập dữ liệu mẫu. Sau đó, đưa ra kết quả thu được và kết luận.

3.1 Thu thập và tiền xử lý dữ liệu

3.1.1 Thu thập dữ liệu

Trong phần chương 3, bộ dữ liệu được sử dụng là bộ dữ liệu thu thập trên mạng internet. Sử dụng Goolge takeout để lấy flie Mbox dữ liệu mail của tên miền @fpt.edu.vn



Hình 3.1 : Lấy dữ liệu bằng Google Takeout

Hình 3.2 : Lấy dữ liệu bằng Google Takeout 2

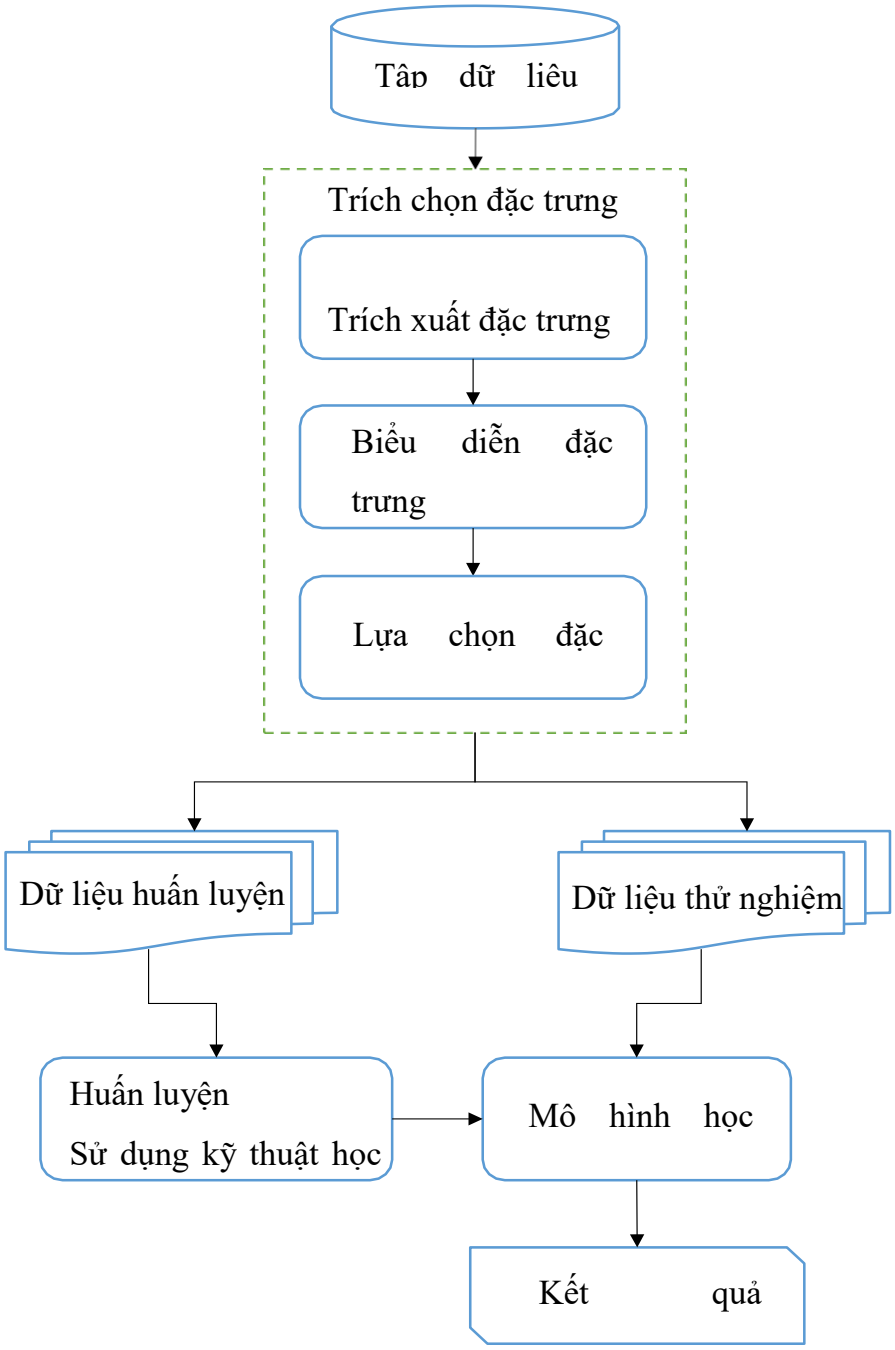
Bộ dữ liệu thực nghiệm gồm 30 user:

Tổng số mail	Số mail quan trọng	Số mail không quan trọng
61733	20054	41679

3.1.2 Tiền xử lý dữ liệu

Với mỗi email có tối đa 12 trường dữ liệu. Các email đều được lấy từ tên miền @fpt.edu.vn. Với mỗi email được lấy ra với các trường dữ liệu {subject',’from', ‘to',’date',’body'} lọc bỏ các email có loại ngôn ngữ khác chỉ để lại các thư là tiếng Việt. Các email được lưu trong tệp định dạng mbox chuyển về định dang csv.

3.2 Thực nghiệm đánh giá



Hình 3.3 : Mô hình quá trình phân loại thư điện tử

Quá trình thực hiện bao gồm 2 giai đoạn:

- Giai đoạn huấn luyện : Đầu vào của giai đoạn này là các dữ liệu đã được tiền xử lý để đưa ra các vector đặc trưng. Trong bước huấn luyện, dữ liệu sẽ được phân loại theo nhãn phân loại tương ứng, sau đó sử dụng các thuật toán học máy để đưa ra bộ phân loại tương ứng phục vụ cho giai đoạn phát hiện.
- Giai đoạn phát hiện: Dữ liệu trong giai đoạn này được xử lý tương tự như dữ liệu trong giai đoạn huấn luyện. Đầu vào của giai đoạn phát hiện là dữ liệu đã được tiền xử lý và model (bộ phân loại – kết quả của giai đoạn huấn luyện).

Áp dụng các tính các trọng số ở chương 2 ta sẽ có điểm số cụ thể cho mỗi e-mail và được tính bằng hàm log của tích các đặc trưng.

Môi trường thử nghiệm: Hệ điều hành window 10, Ngôn ngữ python.

3.3 Kết quả chạy thực nghiệm

Bảng 3.1: Kết quả chạy thử nghiệm

User	Model								
	Random Forest			KNN			Logistic Regression		
	AUC	F1	Recall	AUC	F1	Recall	AUC	F1	Recall
chiennthe141748	0.912	0.892	0.896	0.835	0.862	0.876	0.795	0.84	0.885
dangnhha140192	0.713	0.666	0.670	0.667	0.625	0.632	0.551	0.431	0.571
datntse04909	0.953	0.915	0.916	0.846	0.878	0.885	0.715	0.812	0.853
ducnmhe130666	0.676	0.617	0.618	0.673	0.637	0.637	0.598	0.551	0.563
ducnmse05559	0.834	0.745	0.745	0.675	0.631	0.631	0.495	0.5	0.515
hiepphse04711	0.838	0.767	0.768	0.683	0.643	0.646	0.67	0.635	0.646
hieudtse04712	0.882	0.841	0.843	0.800	0.794	0.802	0.796	0.782	0.797
linhnptsb02246	0.832	0.795	0.802	0.666	0.689	0.705	0.658	0.568	0.694
phucnhse04534	0.849	0.772	0.772	0.722	0.664	0.664	0.65	0.609	0.613
quangnvse05839	0.884	0.795	0.795	0.762	0.702	0.703	0.692	0.644	0.645
quynhthse04640	0.869	0.777	0.776	0.758	0.691	0.692	0.708	0.657	0.659
sanglqse04676	0.949	0.894	0.895	0.862	0.841	0.846	0.778	0.782	0.807
toannbsb02527	0.843	0.775	0.776	0.720	0.673	0.675	0.645	0.606	0.638
tuanntse04733	0.925	0.877	0.879	0.809	0.806	0.815	0.695	0.702	0.767
tuanthsb01889	0.808	0.762	0.769	0.684	0.677	0.689	0.631	0.585	0.682
tungptse04569	0.901	0.819	0.819	0.788	0.718	0.719	0.564	0.431	0.528
tungtmse05324	0.847	0.803	0.809	0.724	0.736	0.754	0.714	0.666	0.753

Bảng 3.2 Độ hiệu quả trung bình của từng thuật toán

Model								
Random Forest			KNN			Logistic Regression		
AUC	F1	Recall	AUC	F1	Recall	AUC	F1	Recall
0.854	0.795	0.797	0.746	0.722	0.728	0.668	0.635	0.683

Accuracy: tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử. **Recall**: là tỷ lệ số điểm true positive trong tổng số những điểm thực sự là positive (TP+FN). Giá trị recall cao đồng nghĩa với việc TPR (true positive Rate) cao, tức là tỷ lệ bỏ sót các điểm thực sự positive là thấp.**F1-score**: là harmonic mean của precision và recall. F1 càng cao, bộ phân loại càng tốt.

3.3 Kết luận chương 3

Từ kết quả trên ta thấy được với thuật toán Random Forest các chỉ số là tốt nhất với các chỉ số : AUC : 0.854,F1 : 0.795,Recall : 0.797 Cho kết quả phân loại tốt nhất trong ba thuật toán.

KẾT LUẬN VÀ KIẾN NGHỊ

1. Kết quả đạt được

- Trình bày sự phổ biến vai trò của thư điện tử trong cuộc sống hiện đại.

- Trình bày kết quả nghiên cứu về thư điện tử: định nghĩa, lịch sử phát triển thư điện tử, các thành phần cấu trúc của hệ thống thư điện tử.

- Các giải pháp hệ thống thư điện tử mã nguồn mở. Chi tiết cài đặt kiến trúc hệ thống, các thành phần của mã nguồn mở Zimba.

- Trình bày các phương pháp hỗ trợ đánh giá giá mức độ ưu tiên thư điện tử.

- Trình bày cơ sở lý thuyết, phương pháp đánh giá mức độ ưu tiên của thư điện tử
- Tiến hành thực nghiệm, đánh giá kết quả. Quá trình thực nghiệm học viên xử lý dữ liệu là các email thu thập được trên internet. Sử dụng thuật toán, phương pháp được trình bày ở chương 2 để tính toán các trọng số từ các đặc trưng của thư, đưa vào các thuật toán học máy để thực hiện phân lớp. Thuật toán hiệu quả nhất khi thực nghiệm là thuật toán Random Forest.

2. Hướng phát triển của luận văn

Một số hướng phát triển tiếp theo của luận văn:

- Nghiên cứu các công nghệ mới được ứng dụng trong phân loại và đánh giá mức độ ưu tiên của thư điện tử.
- Nghiên cứu cái tiến phương pháp đánh giá mức độ ưu tiên của thư điện tử bằng tiếng Việt.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

Tiếng Anh :

[1] Biau, G. (2013). Analysis of a Random Forests Model. *Journal of Machine Learning Research*, 1063-1095.

[2] Drew-Conway-John-Myles-White-Machine-Learning-for-Email_-Spam-Filtering-and-Priority-Inbox-2011-OReilly-Media

[3] Douglas Aberdeen, Ondrej Pacovsky, Andrew Slater Google Inc. Zurich, Switzerland

[4] https://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf

[5] Jorma Laaksonen, Erkki Oja. (1996). Classification with learning k-Nearest Neighbors..

[6] Mitchell, T. (1997). *Machine Learning.*

[7] Milestracy, Wayne Jansen, Scott Bisker, Guidelines on Electronic Mail Securrity U.S Government Printing Office Washington, 2002.

[8] Trang thông tin của Zimbra www.zimbra.com

[9] Thirumuruganathan, S. (2010). *A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm.* Retrieved from https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed- introduction-to-k-nearest-neighbor-knn-algorithm/

Tiếng Việt :

[10] https://vi.wikipedia.org/wiki/H%E1%BB%8Dc_m%C3%A1y#%C4%90%E1%BB%8Bnh_ngh%C4%A9a. Đã truy cập 10 09, 2019

[11] https://machinelearningcoban.com/2017/01/27/logisticregression/#mo-hinh-logistic-regression

[12] https://machinelearningcoban.com/2016/12/27/categories/