HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

LÂM BẢO TUẤN

PHÁT HIỆN CẢNH BÁO BẤT THƯỜNG TRÊN HỆ THỐNG MẠNG VÀ TRUYỀN THÔNG DỰA TRÊN PHÂN TÍCH DỮ LIỆU LOG

LUẬN VĂN THẠC SỸ KỸ THUẬT

(Theo định hướng ứng dụng)

TP. HCM - 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

LÂM BẢO TUẤN

PHÁT HIỆN CẢNH BÁO BẤT THƯỜNG TRÊN HỆ THỐNG MẠNG VÀ TRUYỀN THÔNG DỰA TRÊN PHÂN TÍCH DỮ LIỆU LOG

CHUYÊN NGÀNH:HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

LUẬN VĂN THẠC SỸ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC:

PGS.TS. TRẦN MẠNH HÀ

TP. HCM – 2022

LỜI CAM ĐOAN

Tôi cam đoan rằng luận văn: *"***Phát hiện cảnh báo bất thường trên hệ thống mạng và truyền thông dựa trên phân tích dữ liệu log**" là công trình nghiên cứu của chính tôi.

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

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

TP. Hồ Chí Minh, ngày 25 tháng 01 năm 2022

Học viên thực hiện luận văn

Lâm Bảo Tuấn

LỜI CẢM ƠN

Trong quá trình học tập và thực hiện luận văn , tôi đã nhận được sự quan tâm quý báu và hướng dẫn nhiệt tình của quý Thầy Cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp.

Với lòng kính trọng và biết ơn, tôi xin gửi lời cảm ơn chân thành tới: Ban Giám Đốc, Phòng đào tạo sau đại học của Học viện Công Nghệ Bưu Chính Viễn thông cơ sở TP. Hồ Chí Minh và quý Thầy Cô đã tạo mọi điều kiện thuận lợi giúp tôi hoàn thành luận văn.

Tôi xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất tới người thầy kính yêu **Thầy PGS.TS Trần Mạnh Hà** đã hết lòng giúp đỡ, trực tiếp hướng dẫn tận tình, động viên khích lệ, tạo điều kiện cho tôi trong suốt quá trình thực hiện luận văn.

Từ đáy lòng mình tôi xin bày tỏ sự biết ơn vô hạn đến gia đình thân yêu của tôi và xin chân thành cảm ơn bạn bè thân thiết, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành luận văn.

Mặc dù đã có nhiều cố gắng, nỗ lực tìm tòi nghiên cứu, nhưng do thời gian có hạn và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý thiết thực của quý Thầy Cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn.

Xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 25 tháng 01 năm 2022

Học viên thực hiện luận văn

Lâm Bảo Tuấn

MỤC LỤC

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
IP	Internet Protocol	Giao thức Internet
OID	Object Identifier	Bộ nhận dạng đối tượng
MIB	Management Information Base	Cơ sở thông tin quản lý
HTTP	Hypertext Transfer Protocol	Giao thức truyền siêu văn bản
DHCP	Dynamic Host Configuration Protocol	Giao thức cấp phát địa chỉ IP động
CPU	Central Processing Unit	Bộ xử lý trung tâm
UDP	User Datagram Protocol	Giao thức dữ liệu người dùng
WAN	Wide Area Network	Mạng diện rộng
AI	Artificial Intelligence	Trí tuệ nhân tạo
HDFS	Hadoop Distributed File System	Hệ thống tập tin phân tán
ML	Machine Learning	Học máy
SNMP	Simple Network Monitoring Protocol	Giao thức giám sát mạng đơn giản
LAN	Local Area Network	Mạng máy tính cục bộ
HTML	HyperText Markup Language	Ngôn ngữ Đánh dấu Siêu văn bản
SMTP	Simple Mail Transfer Protocol	Giao thức truyền tải thư điện tử
FTP	File Transfer Protocol	Giao thức truyền tải tập tin
TCP	Transmission Control Protocol	Giao thức điều khiển truyền nhận
CLI	Command Line Interface	Giao diện dòng lệnh
GUI	Graphical User Interface	Giao diện đồ họa người dùng
DOS	Disk Operating System	Hệ điều hành chạy đĩa

RFC	Request for Comments	Tiêu chuẩn về viễn thông, công nghệ thông tin
WCSS	Within-Cluster Sums of Squares	Tổng biến thiên bình phương khoảng cách trong cụm
IETF	Internet Engineering Task Force	Tổ chức đặc trách kỹ thuật Internet
OTT	Over-The-Top	Giải pháp cung cấp nội dung số
IP	Internet Protocol	Giao thức Internet
OSI	Open Systems Interconnection	Mô hình kết nối các hệ thống mở
BSD	Berkeley Software Distribution	Hệ điều hành dẫn xuất từ UNIX
NTP	Network Time Protocol	Giao thức đồng bộ thời gian mạng

DANH SÁCH BẢNG

DANH SÁCH HÌNH VẼ

PHẦN MỞ ĐẦU

1. Tính cấp thiết của đề tài

Hệ thống giám sát mạng (Network montoring) [1] là hệ thống giám sát toàn bộ các cảnh báo, hiệu năng, trạng thái của tất cả thiết bị và máy tính trong một hệ thống mạng. Network montoring bao gồm một phần mềm được kết nối vào hệ thống mạng, nó sẽ thu thập, ghi nhận mọi thông tin được xuất ra từ các thiết bị và giúp người quản trị hệ thống có thể giám sát, theo dõi các thông tin đó theo thời gian thực thông qua giao diện đồ họa đồ thị, biểu đồ hay bảng tính, danh sách chi tiết của phần mềm. Phần mềm này còn có khả năng gửi các thông báo, các cảnh báo cho người quản trị hệ thống biết khi có nguy cơ xảy ra sự cố hoặc có sự cố đang diễn ra thông qua hệ thống tin nhắn SMS, email, các ứng dụng nhắn tin OTT. Hệ thống giám sát mạng đóng vai trò rất quan trọng, không thể thiếu trong mọi hệ thống mạng của các cơ quan, đơn vị, tổ chức.

Thời đại công nghiệp 4.0 đã thúc đẩy đột phá trong nhiều lĩnh vực như Trí tuệ nhân tạo (AI), Máy học (Machine Learning) [2] cùng với đó là sự phát triển bùng nổ của viễn thông, internet dẫn đến hạ tầng mạng viễn thông, công nghệ thông tin càng lớn, càng nhiều thiết bị thì số lượng cảnh báo, lỗi trên toàn mạng là rất lớn đòi hỏi một hệ thống giám sát hệ thống mạng không chỉ đơn thuần là đưa ra thông tin cảnh báo của hệ thống và thiết bị mà còn có thể phát hiện ra những lỗi hệ thống mới, những cảnh báo chưa từng được ghi nhận trước đây hoặc những cảnh báo, lỗi thiết bị về lâu dài có thể ảnh hưởng đến an toàn và hiệu năng của toàn bộ hệ thống mạng. Đó là lý do tôi chọn đề tài nghiên cứu phương pháp giúp xác định chính xác lỗi, cung cấp thông tin về loại sự cố hoặc có thể phát triển đến khả năng dự báo hoặc cảnh báo sớm sự cố mạng (cảnh báo trước khi sự cố xảy ra) dựa trên phân tích dữ liệu sử dụng mạng (lưu lượng, log...) sử dụng các kĩ thuật học máy.

2. Mục đích nghiên cứu

Mục tiêu chính: Dựa vào dữ liệu log lọc ra được những log nào bình thường và phân tích được những log nào là bất thường, tiềm ẩn nguy cơ gây ra những lỗi lớn hơn sau này.

Từ mục tiêu chính trên, luận văn sẽ dự kiến các kết quả đạt được như sau:

- Tìm hiểu tổng quan về các giao thức giám sát lỗi mạng: SNMP, IPFIX, SYSLOG, CLI.
- Tìm hiểu về log và các đặc điểm thuộc tính của log.
- Tìm hiểu tập dữ liệu log giám sát hệ thống (log data, monitoring data).
- Tìm hiểu về một số phương pháp lọc dữ liệu lớn
- Tìm hiểu về một số thuật toán học máy về phân loại và phân cụm.
- Tìm hiểu thuật toán K-means clustering trong việc phân cụm dữ liệu lớn.
- Mối tương quan giữa log và các vấn đề nghiêm trọng.
- Cách đo đạc để phân loại bình thường hay bất thường. Khai thác những thuộc tính quan trọng nào của log, thuộc trường nào của log từ đó hình thành giải thuật và đề xuất giải thuật.

3. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu

- Đối tượng nghiên cứu chính dữ liệu log trong hệ thống HDFS .
- Nghiên cứu thuật toán phân cụm K-means clustering để áp dụng phân cụm dữ liệu log.

Phạm vi nghiên cứu

Phạm vi nghiên cứu trong log giám sát hệ thống HDFS:

- Xây dựng mô hình dữ liệu: lược đồ dữ liệu và mô tả dữ liệu

- Cách xử lý dữ liệu dạng số, nhị phân, liệt kê, dữ liệu văn bản...

4. Phương pháp nghiên cứu

Phương pháp luận: Dựa trên cơ sở là các lý thuyết về giao thức giám sát mạng, các thuật toán phân cụm trong các kỹ thuật học máy.

Phương pháp đánh giá dựa trên cơ sở toán học: Trên cơ sở các lý thuyết về giao thức giám sát mạng, các thuật toán phân cụm trong các kỹ thuật học máy. Đề xuất ra thuật toán để lọc dữ liệu log và phân loại được những dữ liệu log đang cảnh báo những nguy cơ tiềm tàng trong hệ thống. Chứng minh thuật toán và đánh giá hiệu quả của thuật toán.

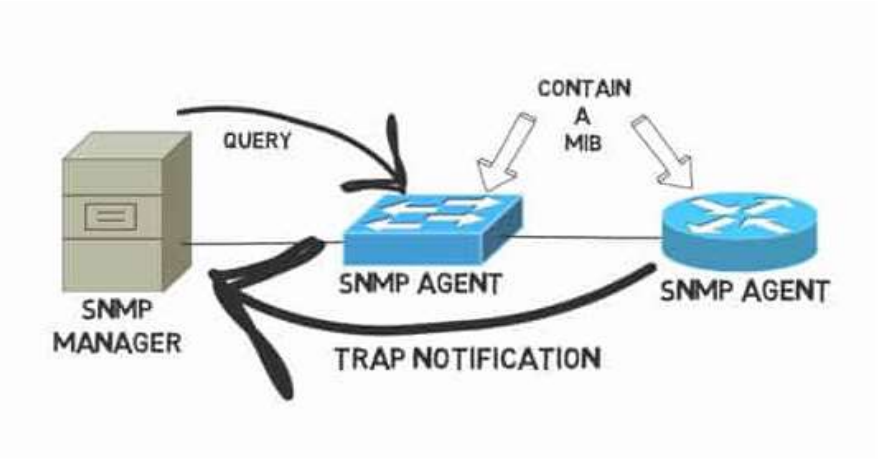
Phương pháp đánh giá bằng mô phỏng thực nghiệm: Xây dựng mô hình mô phỏng và thực nghiệm thuật toán đã đề xuất.

CHƯƠNG 1 - GIỚI THIỆU TỔNG QUAN VỀ CÁC GIAO THỨC GIÁM SÁT LỖI MẠNG VÀ TỔNG QUAN VỀ CÁC KỸ THUẬT HỌC MÁY

1.1 Tổng quan về các giao thức giám sát lỗi mạng

1.1.1 Tổng quan về SNMP

Giao thức quản lý mạng (SNMP) là một giao thức mạng được sử dụng để quản lý và giám sát các thiết bị kết nối mạng trong Giao thức mạng Internet. Giao thức SNMP [3] được nhúng trong nhiều thiết bị cục bộ như bộ định tuyến, bộ chuyển mạch, máy chủ, tường lửa và điểm truy cập không dây bằng cách truy cập qua địa chỉ IP của thiết bị. SNMP cung cấp một cơ chế chung cho các thiết bị mạng để chuyển tiếp thông tin quản lý trong môi trường LAN hoặc WAN của một nhà cung cấp và nhiều nhà cung cấp. Giao thức quản lý mạng đơn giản (SNMP) là một cách để các thiết bị khác nhau trên mạng chia sẻ thông tin với nhau. Nó cho phép các thiết bị giao tiếp ngay cả khi các thiết bị là phần cứng khác nhau và chạy phần mềm khác nhau. Nếu không có giao thức như SNMP, sẽ không có cách nào để các công cụ quản lý mạng xác định thiết bị, giám sát hiệu suất mạng, theo dõi các thay đổi đối với mạng hoặc xác định trạng thái của thiết bị mạng trong thời gian thực. SNMP là một giao thức thuộc lớp ứng dụng trong mô hình OSI.



Hình 1.1: Mô hình kiến trúc SNMP

SNMP có kiến trúc đơn giản dựa trên mô hình máy khách-máy chủ. Các máy chủ, được gọi là người quản lý manager, thu thập và xử lý thông tin về các thiết bị trên mạng. Máy khách, được gọi là agent, là bất kỳ loại thiết bị hoặc thành phần thiết bị nào được kết nối với mạng. Chúng có thể không chỉ bao gồm máy tính mà còn bao gồm cả thiết bị chuyển mạch mạng, điện thoại, máy in, v.v. Một số thiết bị có thể có nhiều thành phần thiết bị. Ví dụ, một máy tính xách tay thường có giao diện mạng có dây cũng như không dây.

Hệ thống phân cấp dữ liệu SNMP có vẻ phức tạp mặc dù kiến trúc SNMP rất đơn giản. Để cung cấp tính linh hoạt và khả năng mở rộng, SNMP không yêu cầu các thiết bị mạng trao đổi dữ liệu ở định dạng có kích thước cố định. Thay vào đó, nó sử dụng định dạng giống cây, theo đó dữ liệu luôn có sẵn để Manager thu thập. Cây dữ liệu bao gồm nhiều bảng (hoặc nhánh, nếu muốn gắn với phép ẩn dụ cây), được gọi là Cơ sở thông tin quản lý hoặc gọi tắt là MIB. MIB nhóm các loại thiết bị hoặc thành phần thiết bị cụ thể lại với nhau. Mỗi MIB có một số nhận dạng duy nhất, cũng như một chuỗi nhận dạng. Số và chuỗi có thể được sử dụng thay thế cho nhau (giống như địa chỉ IP và tên miền).

Mỗi MIB bao gồm một hoặc nhiều nút, đại diện cho các thiết bị riêng lẻ hoặc các thành phần thiết bị trên mạng. Đổi lại, mỗi nút có một mã định danh đối tượng duy nhất OID. OID cho một nút nhất định được xác định bởi số nhận dạng của MIB mà nó tồn tại kết hợp với số nhận dạng của nút trong MIB [4].

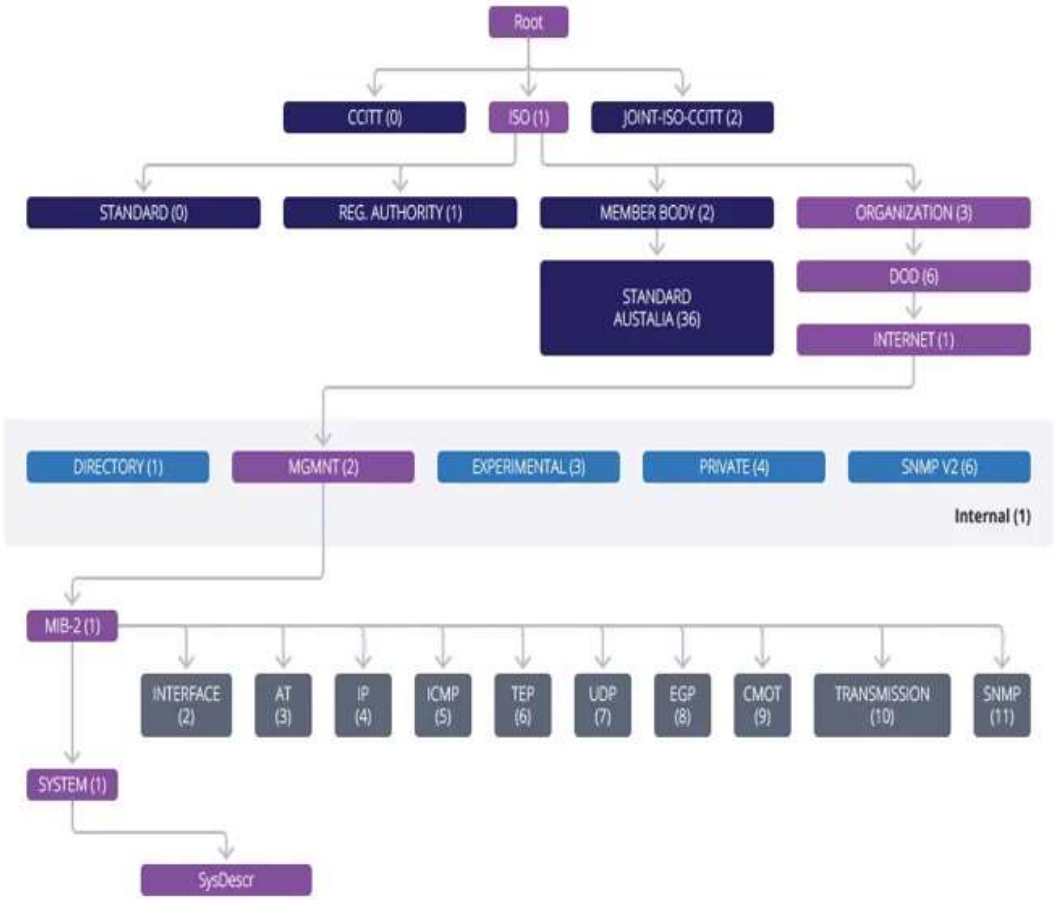
OID có dạng một tập hợp số hoặc chuỗi có thể sử dụng chúng thay thế cho nhau.

Ví dụ OID: 1.3.6.1.4.868.2.4.1.2.1.1.1.3.3562.3.

Được viết bằng chuỗi, OID đó sẽ dịch thành:

iso.org.dod.internet.private.transition.products.chassis.card.slotCps-.cpsSlotSummary.cpsModuleTable.cpsModuleEntry.cpsModuleModel.3562.3.

Sử dụng OID, manager có thể truy vấn agent để tìm thông tin về thiết bị trên mạng. Ví dụ: nếu manager muốn biết liệu một giao diện có hoạt động hay không, trước tiên manager sẽ truy vấn giao diện MIB (được gọi là IF-MIB), sau đó kiểm tra giá trị OID phản ánh trạng thái hoạt động để xác định xem giao diện đó có hoạt động hay không.



Hình 1.2: Mô hình phân cấp MIB

Hệ thống phân cấp dữ liệu MIB và OID có vẻ phức tạp nhưng có một số lợi thế quan trọng đối với một hệ thống. Một là thông tin có thể được lấy bởi nanager mà không cần phải gửi một yêu cầu rõ ràng cho agent để thu thập thông tin đó. Điều đó làm giảm chi phí và đảm bảo thông tin về trạng thái của mạng luôn sẵn sàng.

Hệ thống cũng cung cấp một cách dễ dàng, linh hoạt để tổ chức nhiều thiết bị trên một mạng. Nó hoạt động bất kể mạng lớn hay nhỏ, hoặc loại thiết bị nào trên đó.

SNMP cũng làm cho nó có thể thu thập một lượng lớn thông tin một cách nhanh chóng mà không làm nghẽn mạng về lưu lượng. Vì thông tin về trạng thái thiết bị luôn có sẵn ở định dạng đơn giản và được cập nhật theo thời gian thực, manager có thể lấy thông tin mà không cần đợi dữ liệu được thu thập hoặc yêu cầu truyền dữ liệu lớn.

Cần lưu ý rằng một số giá trị OID dành riêng cho nhà cung cấp thiết bị, điều này giúp dễ dàng có được một số thông tin về một thiết bị chỉ dựa trên OID của nó. Ví dụ: nếu OID bắt đầu bằng 1.3.6.1.4.1.9, thì nó sẽ áp dụng cho thiết bị Cisco. Các nhà cung cấp khác có thông số kỹ thuật OID của riêng họ [5]. Wireshark, phần mềm quét mạng mã nguồn mở, cung cấp một công cụ tra cứu OID tiện dụng. Tiền tố OID tiêu chuẩn, có thể được sử dụng cho hầu hết mọi thiết bị hỗ trợ SNMP là: 1.3.6.1.2.

Các phiên bản SNMP khác nhau thì tính năng sẽ có nhiều khác biệt đặc biệt là vấn đề bảo mật.

Phiên bản đầu tiên của SNMP lầ SNMPv1 cung cấp các tính năng bảo mật yếu. Theo SNMPv1, manager có thể xác thực cho các client mà không cần mã hóa khi yêu cầu thông tin. Điều đó có nghĩa là bất kỳ ai có quyền truy cập vào mạng đều có thể chạy phần mềm "nghe lén" để lấy thông tin về mạng. Điều đó cũng có nghĩa là một thiết bị trái phép có thể dễ dàng giả mạo là một manager hợp pháp khi kiểm soát mạng.

Ngoài ra, SNMPv1 sử dụng một số thông tin đăng nhập mặc định nhất định mà quản trị viên không phải lúc nào cũng cập nhật, giúp các bên trái phép dễ dàng truy cập vào thông tin nhạy cảm về mạng. SNMPv1 ngày nay vẫn được sử dụng tương đối rộng rãi vì một số mạng chưa cập nhật.

SNMPv2, xuất hiện vào năm 1993, cung cấp một số cải tiến bảo mật, nhưng nó đã được thay thế vào năm 1998 bởi SNMPv3, phiên bản này vẫn là phiên bản mới nhất của giao thức và an toàn nhất.

SNMPv3 giúp mã hóa dữ liệu có thể thực hiện được. Nó cũng cho phép quản trị viên chỉ định các yêu cầu xác thực khác nhau trên cơ sở chi tiết cho manager và

agent. Điều này ngăn chặn xác thực trái phép và có thể được tùy chọn sử dụng để yêu cầu mã hóa cho việc truyền dữ liệu.

Điểm mấu chốt là, trong khi các vấn đề bảo mật trong SNMPv1 khiến SNMP bị coi là xấu trong một số vòng kết nối, SNMPv2 và đặc biệt là SNMPv3 đã giải quyết được những vấn đề đó. Các phiên bản SNMP mới hơn cung cấp một cách thức cập nhật và an toàn để giám sát mạng.

SNMP thường không được bật theo mặc định trên các thiết bị. Điều đó có nghĩa là, trong hầu hết các trường hợp, quản trị viên phải đăng nhập và bật tính năng này để cung cấp dữ liệu SNMP. Yêu cầu này giúp giảm nguy cơ chạy phiên bản SNMP không an toàn như vấn đề bảo mật kém trong SNMPv1 mà không không được người quản trị nhận ra.

1.1.2 Giới thiệu về Log

Log ghi lại liên tục thành các bản tin thông báo về trạng thái và hoạt động của hệ thống. Log file thường là các file văn bản thông thường dưới dạng văn bản thuần túy tức là bạn có thể dễ dàng đọc được để theo dõi các sự kiện của hệ thống, vì thế có thể sử dụng các trình soạn thảo văn bản hoặc các trình xem văn bản thông thường như Notepad, wordpad là có thể xem được file log.

Các thuộc tính của log được có trong những thông báo log của hệ thống xuất ra về những thay đổi, quá trình hoạt động của hệ thống từ đăng nhập, đăng xuất, cảnh báo nhiệt độ cao, port down, port up, mất kết nối, cảnh báo bộ nhớ đầy… đến những lỗi phát sinh trong hệ thống. Log được ghi lại liên tục theo thời gian, số lượng log thì vô cùng lớn, mỗi một bản tin log sẽ có rất nhiều thuộc tính có thể lên đến hàng trăm thuộc tính để chỉ ra trạng thái hiện tại của hệ thống.

Log sẽ có các thuộc tính cơ bản như sau:

- <date/time><host><message source><message>

Date/time: Giờ hệ thống của thiết bị khi ghi nhận log

Host: Có thể là tên miền, tên máy, IP của thiết bị

Message source: Nguồn có thể là một phần mềm hệ thống hoặc là một bộ phận mà sinh ra thông báo log

Log message: Thông báo log có thể có nhiều định dạng khác nhau, thông thường bao gồm tên ứng dụng, các biến tình trạng đa dạng, địa chỉ IP nguồn, giao thức, chuỗi ký tự miêu tả thông điệp cảnh báo vấn đề gì.

1.1.3 Tổng quan về Syslog

Syslog là một giao thức tiêu chuẩn để gửi và nhận các thông báo nhật ký ở định dạng văn bản cụ thể, rõ ràng dạng văn bản thuần túy từ các thiết bị mạng khác nhau nhờ đó có thể dễ dàng mở xem và phân tích log. Syslog được thiết kế để giám sát các thiết bị mạng và hệ thống để gửi tin nhắn thông báo nếu có bất kỳ vấn đề nào về chức năng, nó cũng gửi cảnh báo cho các sự kiện được thông báo trước và giám sát hoạt động đáng ngờ thông qua nhật ký thay đổi, nhật ký sự kiện của các thiết bị mạng trong hệ thống. Các cảnh báo bao gồm mốc thời gian, thông báo sự kiện, mức độ nghiêm trọng, địa chỉ IP máy chủ, chẩn đoán. Mỗi thông báo được gắn nhãn cho biết loại hệ thống tạo ra thông báo và được ấn định mức độ nghiêm trọng. Về mức độ nghiêm trọng được tích hợp sẵn, nó trong phạm vi từ cấp 0 cao nhất khẩn cấp nhất tới cấp 7 thấp nhất ít nguy cơ nhất [6].

Bảng 1.1: Các cấp độ cảnh báo xuất ra của log

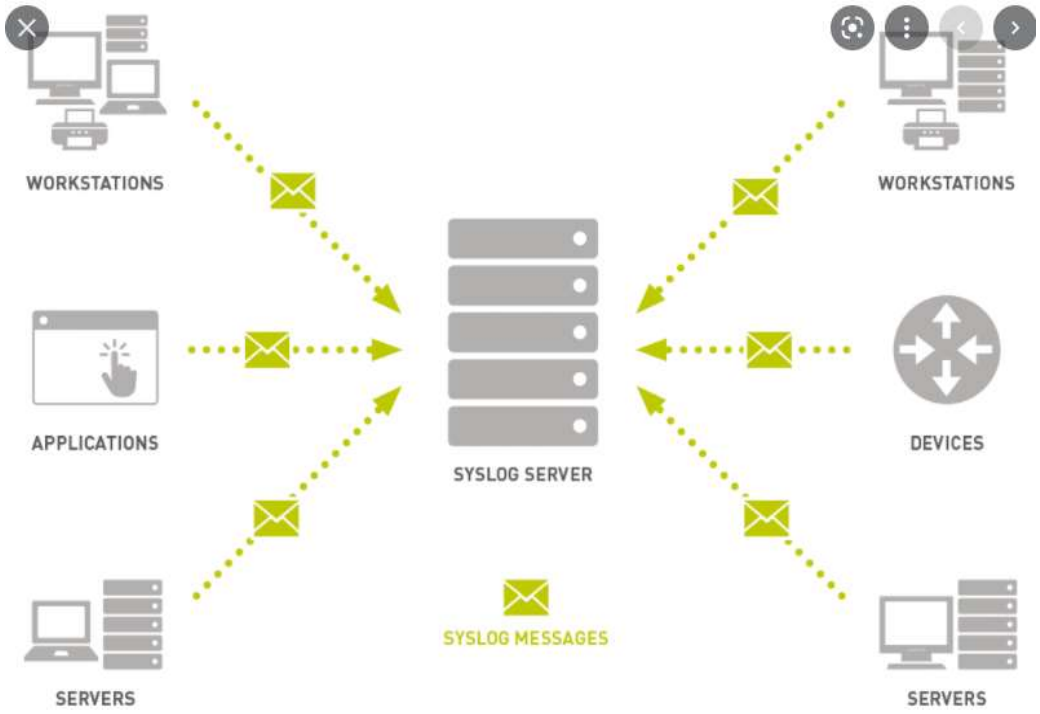
Giá trị	Mức độ cảnh báo	Định nghĩa
0	Emergency	Khẩn cấp
1	Alert	Báo động
2	Critical	Nguy hiểm
3	Error	Lỗi hệ thống
4	Warning	Cảnh báo

5	Notice	Cần chú ý
6	Informational	Thông tin
7	Debug	Gỡ rối

Các kỹ sư thiết kế hệ thống máy tính có thể sử dụng Syslog để quản lý hệ thống và kiểm tra bảo mật cũng như các thông báo thông tin chung, phân tích và gỡ lỗi. Nhiều loại thiết bị, chẳng hạn như máy in, bộ định tuyến và bộ nhận tin nhắn trên nhiều nền tảng sử dụng chung tiêu chuẩn Syslog. Điều này cho phép hợp nhất dữ liệu ghi nhật ký từ các loại hệ thống khác nhau trong một kho lưu trữ trung tâm. Việc triển khai nhật ký hệ thống được thực hiện cho nhiều hệ điều hành. Khi hoạt động trên mạng, Syslog sử dụng kiến trúc máy chủ-máy client nơi máy chủ log hệ thống lắng nghe và ghi nhật ký các thông báo đến từ các máy client.

Giao thức Syslog ban đầu được phát triển bởi Eric Allman và được định nghĩa trong RFC 3164. Các thông báo được gửi qua các mạng IP đến bộ thu thập thông điệp sự kiện hoặc máy chủ nhật ký hệ thống. Syslog sử dụng Giao thức UDP, cổng 514 để giao tiếp. Kể từ năm 2009, Syslog đã được IETF chuẩn hóa trong RFC 5424.

Để tăng độ tin cậy cho Syslog thì IETF đã ban hành tiếp RFC 3195 Reliable Delivery for syslog và RFC 6587 Transmission of Syslog Messages over TCP nên Syslog hiện tại có thể sử dụng UDP hoặc TCP để tăng độ an toàn trong quá trình truyền tin. Ngày nay, nó đã được hỗ trợ rộng rãi trên nhiều hệ điều hành bao gồm hầu hết các phiên bản Linux, Unix và MacOS. Đối với Microsoft Windows, Syslog được hỗ trợ thông qua các nguồn mở và thư viện thương mại của bên thứ ba.



Hình 1.3: Mô hình Syslog Server

Thiết bị mạng qua giao thức syslog sẽ gửi log đến máy chủ lưu trữ log, giao thức syslog cho phép thu thập ghi lại thông tin với các chức năng chính như sau

- Ghi lại thông tin tài khoản đăng nhập để theo dõi các tác động của tài khoản đó
- Cho phép chọn các loại thông tin đăng nhập cần ghi lại
- Xác định nơi lưu thông điệp syslog được ghi lại

Máy chủ Syslog được sử dụng để gửi dữ liệu chẩn đoán và giám sát. Dữ liệu sau đó có thể được phân tích để giám sát hệ thống, bảo trì mạng. Vì giao thức Syslog được hỗ trợ bởi nhiều thiết bị, chúng có thể đăng nhập thông tin vào máy chủ Syslog một cách thuận tiện. Dữ liệu SNMP có thể được sử dụng để đánh giá bất kỳ điểm hỏng hóc nào một cách nhanh chóng. Máy chủ Syslog cũng có thể có các sự kiện tự động để kích hoạt cảnh báo giúp ngăn chặn thời gian ngừng hoạt động.

Mã cơ sở được sử dụng để chỉ ra loại hệ thống đang ghi thông báo. Các tin nhắn với các phương tiện khác nhau có thể được xử lý khác nhau. Danh sách các mã cơ sở có sẵn được xác định theo tiêu chuẩn [7].

Bảng 1.2: Các nguồn sinh ra log

Mã Facility	Từ khóa	Diễn giải
0	Kern	Log từ nhân Kern
1	User	Log từ người dùng
2	mail	Log hệ thống mail
3	Daemon	Log từ tiến trình nền
4	Auth	Log từ xác thực
5	Syslog	Log từ syslod
6	lpr	Log từ quá trình in ấn
7	news	Thông tin hệ thống
8	uucp	Tập hợp chương trình cấp thấp
9	cron	Tiện ích thực hiện tác vụ theo định kỳ
10	authpriv	Truy cập và bảo mật
11	ftp	Log của FTP Daemon
12	ntp	Hệ thống NTP
13	Security	Kiểm tra đăng nhập
14	Console	Cảnh báo hệ thống
15	Solaris-cron	Log lập trình
16-23	Local0 to local7	Log dự trữ sử dụng nội bộ

Facility được sử dụng để xác định chương trình hoặc một phần của hệ thống tạo ra các bản ghi.

Theo mặc định, một số phần trong hệ thống được cung cấp các mức facility như kernel sử dụng kern facility hoặc hệ thống mail bằng cách sử dụng mail facility. Nếu một bên thứ ba muốn phát hành log, có thể đó sẽ là một tập hợp các cấp độ facility được bảo lưu từ 16 đến 23 được gọi là "local use" facility levels.

Ngoài ra, có thể sử dụng tiện ích cấp độ người dùng ("user-level" facility), nghĩa là sẽ đưa ra các log liên quan đến người dùng.

1.1.4 Các ứng dụng dùng để ghi log

Dưới đây là danh sách một số phần mềm Syslog trên Windows:

Kiwi Syslog

Máy chủ này cài đặt và tạo báo cáo ở dạng văn bản thuần túy hoặc HTML. Phần mềm xử lý Syslog và SNMP, ngay cả từ các máy chủ Linux và UNIX. Nó tương thích với Windows XP 32/64, Windows 2003 32/64, Windows Vista 32/64, Win7 32/64, Windows 2008 R2 32/64, Windows 8, Windows 10, Windows Server 2012 & 2012 R2…

Kiwi Syslog nhận bản tin syslog gửi về từ các thiết bị mạng và xuất ra theo thời gian thực

- Bản tin Syslog có thể được xử lý bằng các tác vụ như:
- Hiển thị bản tin trong các cửa sổ Windows
- Ghi log bản tin vào một tập tin văn bản
- Chuyển tiếp bản tin đến server syslog khác
- Gửi e-mail đến người quản trị qua giao thức mail SMTP
- Kích hoạt cảnh báo bằng âm thanh
- Gửi bản tin SNMP Trap
- Truy nhập Kiwi Syslog Web Access

Khi các bản tin nhận được các tác vụ có thể được thực hiện. Bản tin có thể được lọc theo tên server, địa chỉ IP server, độ ưu tiên, nội dung bản tin hoặc thời gian nhận bản tin. Kiwi Syslog nhận bản tin syslog gửi về từ các thiết bị mạng và xuất ra theo thời gian thực

Rsyslog

Rsyslog là một tiện ích phần mềm mã nguồn mở được sử dụng trên các hệ thống máy tính Unix để chuyển tiếp các thông báo nhật ký trong mạng IP. Nó triển khai giao thức nhật ký hệ thống cơ bản, mở rộng nó với tính năng lọc dựa trên nội dung, khả năng lọc phong phú, các hoạt động được xếp hàng để xử lý đầu ra ngoại tuyến, hỗ trợ cho các đầu ra mô-đun khác nhau, tùy chọn cấu hình linh hoạt và thêm các tính năng như sử dụng TCP để truyền tải.

Rsyslog sử dụng giao thức nhật ký hệ thống BSD tiêu chuẩn, được ban hành trong RFC 3164. Rsyslog là hệ thống được phát triển nhằm để quá trình ghi Log được thực hiện một cách nhanh chóng. Rsyslog mang đến hiệu quả và khả năng xử lý Log một các thuyết phục, tính năng bảo mật tốt và xây dựng thêm module cho các sửa đổi tùy chỉnh. Rsyslog đã xây dựng một hệ thống dữ liệu đơn lẻ để phân tích và sắp xếp các bản ghi từ một loạt các nguồn mở rộng, sau đó chuyển đổi và đưa qua một đầu ra để sử dụng trong các chương trình phân tích log hệ thống chuyên nghiệp.

Splunk

Splunk là một phần mềm giám sát mạng dựa trên việc phân tích Log. Splunk thực hiện các công việc tìm kiếm, giám sát và phân tích các dữ liệu lớn được sinh ra từ các ứng dụng, các hệ thống và các thiết bị hạ tầng mạng. Nó có thể thao tác tốt với nhiều loại dịnh dạng dữ liệu khác nhau (Syslog, csv, apache-log, access_combined…).

Các tính năng của Splunk

Hỗ trợ hầu như tất cả các loại log của hệ thống, thiết bị hạ tầng mạng, phần mềm

Splunk có thể thực hiện việc thu thập log từ rất nhiều nguồn khác nhau. Từ một file hoặc thư mục (kể cả file nén) trên server, qua các kết nối UDP, TCP từ các Splunk Server khác trong mô hình Splunk phân tán, từ các Event Log, Registry của Windows …Splunk kết hợp rất tốt với các công cụ thu thập log khác.

Splunk cập nhật dữ liệu liên tục khi có thay đổi trong thời gian thực. Giúp cho việc phát hiện và cảnh báo trong thời gian thực.

Splunk có thể đánh chỉ mục dữ liệu với một khối lượng dữ liệu rất lớn trong một khoảng thời gian ngắn. Giúp việc tìm kiếm diễn ra nhanh chóng và thuận tiện.

Splunk cung cấp cho người dùng một cơ chế cảnh báo dựa trên việc tìm kiếm các thông tin do chính người sử dụng đặt ra. Khi có vấn đề liên quan tới hệ thống phù hợp với các tiêu chí mà người dùng đã đặt ra thì hệ thống sẽ cảnh báo ngay tới người dùng (cảnh bảo trực tiếp qua giao diện, gửi Email).

Splunk cung cấp một cơ chế hiển thị rất trực quan giúp người sử dụng có thể dễ dàng hình dung về tình trạng của hệ thống, đưa ra các đánh giá về hệ thống. Splunk còn từ động kết xuất ra các báo cáo với nhiều loại định dạng một cách rất chuyên nghiệp.

Nagios

Nagios được phát triển bởi Galstad vào năm 1999, Lúc đầu Nagios được biết đến với cái tên là NetSaint. Dần sau đó, Nagios được phát triển như một phần mềm mã nguồn mở dành cho người quản trị mạng trong việc giám sát các Host, Services (DHCP, HTTP, …) hay một số tài nguyên hệ thống như dung lượng trên các ổ đĩa, hoạt động của CPU trong hệ thống mạng.

Hệ thống Nagios được bao gồm 2 phần chính đó là Nagios Plugins và Nagios Core.

Nagios Plugins: là phần mở rộng độc lập để Nagios Core cung cấp ở mức độ thấp về cách theo dõi bất cứ điều gì và tất cả mọi thứ với Nagios Core. Plugins xử lý đối số dòng lệnh, đi về các doanh nghiệp thực hiện kiểm tra, và sau đó trả lại kết quả cho Nagios Core để xử lý tiếp. Plugin có thể được biên dịch nhị phân (viết bằng C, C++, …) hoặc các bản thực thi (Perl, PHP).

Nagios core: Đây được hiểu là công cụ giám sát, đảm nhiệm quản lý những lịch trình sự kiện cơ bản, xử lý sự kiện và quản lý thông báo cho các phần tử được

theo dõi. Nó bổ sung giao diện lập trình ứng dụng. Được sử dụng để mở rộng khả năng để thực hiện nhiệm vụ bổ sung.

Bảng 1.3: So sánh các phần mềm ghi log

Phần mềm	So sánh
Kiwi Syslog	Lưu trữ các loại log từ nhiều thiết bị. Cung cấp một giao diện đơn giản, dễ cài đặt và sử dụng. Tối giản giao diện, không có phân tích log, chỉ hỗ trợ Windows. Không thể cấu hình một số tính năng quản lý thông qua giao diện web
Splunk	Giám sát theo thời gian thực. Cảnh báo theo lịch trình, thiết lập cảnh báo đáng chú ý vào mục riêng. Thời gian phản hồi kết quả tìm kiếm khá tốt. Splunk giúp truy vấn dữ liệu nhanh chóng lập chỉ mục tất cả dữ liệu và cung cấp các khóa để tìm kiếm, cung cấp thông tin chi tiết về dữ liệu lịch sử. Một số truy vấn có thể chạy chậm nếu các chỉ mục không nằm trên một phần của truy vấn sử dụng.
Nagios Log Server	Tính năng kiểm tra log. Giám sát máy chủ tốt. Phần mềm khó cài đặt và cấu hình. Giá thành cao.

1.1.5 Tổng quan về IPFIX

Là một giao thức do IETF tạo ra, IPFIX là viết tắt của IP Flow Information Export. Nó được tạo ra dựa trên nhu cầu về một tiêu chuẩn xuất luồng thông tin chung, phổ biến cho thông tin luồng Giao thức Internet từ bộ định tuyến, đầu dò và các thiết bị khác được sử dụng bởi hệ thống sắp xếp, hệ thống kế toán /thanh toán và hệ thống quản lý mạng để hỗ trợ các dịch vụ như đo lường, kế toán và thanh toán. Tiêu chuẩn IPFIX xác định cách thông tin luồng IP được định dạng và chuyển từ trình xuất sang trình thu thập. Trước đây, nhiều nhà khai thác mạng dữ liệu đang dựa vào công nghệ NetFlow độc quyền của Cisco Systems để xuất thông tin luồng lưu lượng.

Các yêu cầu về tiêu chuẩn IPFIX đã được nêu trong RFC 3917 ban đầu. Cisco NetFlow Phiên bản 9 là cơ sở cho IPFIX. Các thông số kỹ thuật cơ bản cho IPFIX được ghi lại trong RFC 7011 đến RFC 7015 và RFC 5103.

IPFIX rất giống với Netflow, nó cho phép các kỹ sư mạng và quản trị viên thu thập luồng thông tin từ Thiết bị chuyển mạch, Bộ định tuyến và bất kỳ thiết bị mạng nào khác hỗ trợ giao thức và phân tích luồng thông tin, lưu lượng đang được gửi bằng cách xử lý nó qua trình phân tích mạng hoặc luồng mạng.

Giao thức IPFix được tạo ra để trở thành một giao thức chung và phổ biến để xuất luồng thông tin bằng IP từ các thiết bị mạng, bao gồm Thiết bị chuyển mạch, Bộ định tuyến, tường lửa và những thứ đó đến Bộ thu thập hoặc Hệ thống quản lý mạng.

Bắt nguồn từ Netflow Phiên bản 9, nó sử dụng nhiều thủ tục giống nhau để xuất một "luồng" tới nơi thu thập, hoạt động trong mối quan hệ nhiều-nhiều - có nghĩa là thiết bị mạng có thể gửi đến nhiều nơi thu thập và nhiều nơi thu thập có thể thu thập thông tin từ bất kỳ thiết bị nào.

Luồng bao gồm tất cả lưu lượng thuộc cùng một ngữ cảnh giao tiếp, về cơ bản có nghĩa là tất cả các gói dữ liệu IP thuộc về cùng một kết nối.

Thông tin luồng được đẩy đến người thu thập mà không cần yêu cầu bất kỳ điều gì và có thể được tùy chỉnh để bao gồm bất kỳ số lượng, kiểu thông tin dữ liệu được xác định trước hoặc do người dùng xác định. Tính linh hoạt này là một trong

những giao thức phù hợp mạnh mẽ, vì các nhà cung cấp có thể tạo các mẫu tùy chỉnh với thông tin tùy chỉnh mà họ muốn thu thập và phân tích.

Sự khác biệt chính giữa Netflow so với IPFix là trước hết, IPFIX có khả năng tích hợp thông tin thường được gửi đến thông tin Syslog hoặc SNMP trực tiếp trong gói IPFIX, do đó loại bỏ nhu cầu về các dịch vụ bổ sung này thu thập dữ liệu từ mỗi thiết bị mạng. Điều này về cơ bản cho phép các nhà cung cấp phần cứng chỉ định ID nhà cung cấp và đưa bất kỳ thông tin độc quyền nào vào Luồng và xuất nó ra khỏi bộ thu thập / phân tích để phân tích và giám sát thêm. IPFIX cũng cho phép các trường có độ dài "Thay đổi", có nghĩa là không có độ dài cố định mà ID phải tuân theo. Netflow không cho phép loại trường có độ dài thay đổi này. Sau đó, các trường độ dài thay đổi cho phép bạn lưu thông tin như URL (khác nhau giữa các trang web), Tin nhắn, máy chủ HTTP, v.v.

IPFIX có thể tương thích ngược với chuẩn NetFlow v9 do Cisco phát triển và nhiều nhà cung cấp thiết bị khác nhau. NetFlow v9 có một tập hợp 79 loại trường được xác định, trong khi IPFIX có cùng 79 loại trường tương tự để tương thích ngược, nhưng số lượng trường lên tới 238, IPFIX có trường ID của hãng cung cấp thiết bị. IPFLIX cho phép thu thập hầu hết mọi loại dữ liệu.

1.1.6 Tổng quan về CLI

Giao diện dòng lệnh (CLI) xử lý các lệnh tới một chương trình máy tính dưới dạng các dòng văn bản. Chương trình xử lý giao diện được gọi là trình thông dịch dòng lệnh hoặc bộ xử lý dòng lệnh. Hệ điều hành thực hiện một giao diện dòng lệnh trong một trình bao để truy cập tương tác vào các chức năng hoặc dịch vụ của hệ điều hành. Quyền truy cập như vậy chủ yếu được cung cấp cho người dùng bởi các thiết bị đầu cuối máy tính bắt đầu từ giữa những năm 1960 và tiếp tục được sử dụng trong suốt những năm 1970 và 1980 trên các hệ thống VAX/VMS [8], Unix và các hệ thống máy tính cá nhân bao gồm DOS và Apple DOS.

Ngày nay, nhiều người dùng dựa vào giao diện người dùng đồ họa và các tương tác theo hướng menu. Tuy nhiên, một số tác vụ lập trình và bảo trì có thể không

có giao diện người dùng đồ họa và vẫn có thể sử dụng dòng lệnh. Các lựa chọn thay thế cho giao diện dòng lệnh bao gồm các menu giao diện người dùng dựa trên văn bản (ví dụ: IBM AIX SMIT), phím tắt và các phép ẩn dụ trên màn hình khác nhau tập trung vào con trỏ (thường được điều khiển bằng chuột). Ví dụ về điều này bao gồm Microsoft Windows, DOS Shell và Mouse Systems PowerPanel. Giao diện dòng lệnh thường được thực hiện trong các thiết bị đầu cuối cũng có khả năng giao diện người dùng dựa trên văn bản hướng màn hình sử dụng địa chỉ con trỏ để đặt các ký hiệu trên màn hình hiển thị. Các chương trình có giao diện dòng lệnh thường dễ tự động hóa hơn thông qua tập lệnh.

So với giao diện người dùng đồ họa, giao diện dòng lệnh yêu cầu ít tài nguyên hệ thống hơn để triển khai. Vì các tùy chọn cho các lệnh được đưa ra trong một vài ký tự trong mỗi dòng lệnh, người dùng có kinh nghiệm thường thấy các tùy chọn này dễ truy cập hơn. Tự động hóa các tác vụ lặp đi lặp lại được đơn giản hóa bằng các cơ chế chỉnh sửa dòng và lịch sử để lưu trữ các chuỗi được sử dụng thường xuyên; điều này có thể mở rộng sang một ngôn ngữ kịch bản có thể nhận các tham số và các tùy chọn thay đổi. Lịch sử dòng lệnh có thể được lưu giữ, cho phép xem lại hoặc lặp lại các lệnh.

Hệ thống dòng lệnh có thể yêu cầu hướng dẫn sử dụng trên giấy hoặc trực tuyến để người dùng tham khảo, mặc dù thường tùy chọn "trợ giúp" cung cấp đánh giá ngắn gọn về các tùy chọn của lệnh. Môi trường dòng lệnh có thể không cung cấp các cải tiến về đồ họa như các phông chữ khác nhau hoặc các cửa sổ chỉnh sửa mở rộng được tìm thấy trong GUI. Người dùng mới có thể khó làm quen với tất cả các lệnh và tùy chọn có sẵn, so với các biểu tượng và menu thả xuống của giao diện người dùng đồ họa mà không cần tham khảo nhiều lần đến sách hướng dẫn.

Các chương trình ứng dụng (trái ngược với hệ điều hành) cũng có thể có giao diện dòng lệnh.

Một chương trình ứng dụng có thể không hỗ trợ bất kỳ, bất kỳ hoặc tất cả ba loại cơ chế giao diện dòng lệnh chính sau:

Thông số: Hầu hết các hệ điều hành đều hỗ trợ một phương tiện để truyền thông tin bổ sung cho một chương trình khi nó được khởi chạy. Khi một chương trình được khởi chạy từ trình bao dòng lệnh của hệ điều hành, văn bản bổ sung được cung cấp cùng với tên chương trình sẽ được chuyển đến chương trình đã khởi chạy.

Phiên dòng lệnh tương tác: Sau khi khởi chạy, một chương trình có thể cung cấp cho người vận hành một phương tiện độc lập để nhập lệnh dưới dạng văn bản.

Giao tiếp giữa các quá trình: Hầu hết các hệ điều hành đều hỗ trợ các phương tiện giao tiếp giữa các quá trình (ví dụ: các luồng tiêu chuẩn hoặc các đường ống được đặt tên). Các dòng lệnh từ các quy trình khách hàng có thể được chuyển hướng đến chương trình CLI bằng một trong các phương pháp này.

Một số ứng dụng chỉ hỗ trợ CLI, hiển thị lời nhắc CLI cho người dùng và hoạt động theo các dòng lệnh khi chúng được nhập. Các chương trình khác hỗ trợ cả CLI và GUI. Trong một số trường hợp, GUI chỉ đơn giản là một trình bao bọc xung quanh một tệp thực thi CLI riêng biệt. Trong các trường hợp khác, một chương trình có thể cung cấp CLI như một sự thay thế tùy chọn cho GUI của nó. CLI và GUI thường hỗ trợ các chức năng khác nhau. Ví dụ: tất cả các tính năng của MATLAB, một chương trình máy tính phân tích số, có sẵn thông qua CLI, trong khi MATLAB GUI chỉ hiển thị một tập hợp con các tính năng.

1.2 Một số thuật toán học máy

1.2.1 Mạng Nơ-ron nhân tạo (Neural Network)

Một mạng lưới thần kinh sinh học bao gồm một nhóm các tế bào thần kinh liên kết về mặt hóa học hoặc chức năng. Một nơ-ron duy nhất có thể được kết nối với nhiều nơ-ron khác và tổng số nơ-ron và kết nối trong một mạng có thể lớn. Các kết nối, được gọi là khớp thần kinh, thường được hình thành từ sợi trục đến đuôi gai, mặc dù có thể có các khớp thần kinh đuôi gai và các kết nối khác [9]. Ngoài tín hiệu điện, có những hình thức tín hiệu khác phát sinh từ sự khuếch tán chất dẫn truyền thần kinh. Trí tuệ nhân tạo, mô hình nhận thức và mạng nơ-ron là những mô hình xử lý thông tin được lấy cảm hứng từ cách hệ thống thần kinh sinh học xử lý dữ liệu. Mạng

nơ-ron là một chuỗi các thuật toán cố gắng nhận ra các mối quan hệ cơ bản trong một tập hợp dữ liệu thông qua một quá trình bắt chước cách bộ não con người hoạt động. Theo nghĩa này, mạng nơ-ron đề cập đến hệ thống nơ-ron, có thể là hữu cơ hoặc nhân tạo trong tự nhiên. Mạng nơron có thể thích ứng với việc thay đổi đầu vào; để mạng tạo ra kết quả tốt nhất có thể mà không cần thiết kế lại các tiêu chí đầu ra. Khái niệm về mạng nơ-ron, có nguồn gốc từ trí tuệ nhân tạo, đang nhanh chóng trở nên phổ biến trong sự phát triển của các hệ thống giao dịch.

Trí tuệ nhân tạo và mô hình nhận thức cố gắng mô phỏng một số đặc tính của mạng nơ-ron sinh học. Trong lĩnh vực trí tuệ nhân tạo, mạng nơ-ron nhân tạo đã được ứng dụng thành công để nhận dạng giọng nói, phân tích hình ảnh và điều khiển thích ứng, nhằm tạo ra các tác nhân phần mềm (trong máy tính và trò chơi điện tử) hoặc robot tự động. Trong lịch sử, máy tính kỹ thuật số phát triển từ mô hình von Neumann và hoạt động thông qua việc thực hiện các lệnh rõ ràng thông qua quyền truy cập vào bộ nhớ của một số bộ xử lý. Mặt khác, nguồn gốc của mạng nơ-ron dựa trên những nỗ lực lập mô hình xử lý thông tin trong các hệ thống sinh học. Không giống như mô hình von Neumann, tính toán mạng nơ-ron không tách biệt bộ nhớ và xử lý. Lý thuyết mạng lưới thần kinh vừa giúp xác định rõ hơn cách thức hoạt động của các tế bào thần kinh trong não vừa cung cấp cơ sở cho những nỗ lực tạo ra trí thông minh nhân tạo.

Mạng nơron (NN), trong trường hợp các nơron nhân tạo được gọi là mạng nơron nhân tạo (ANN) hoặc mạng nơron mô phỏng (SNN), là một nhóm các nơron tự nhiên hoặc nhân tạo được kết nối với nhau sử dụng mô hình toán học hoặc tính toán để xử lý thông tin dựa trên cách tiếp cận liên kết để tính toán. Trong hầu hết các trường hợp, ANN là một hệ thống thích ứng thay đổi cấu trúc của nó dựa trên thông tin bên ngoài hoặc nội bộ truyền qua mạng.

Nói một cách thực tế hơn, mạng nơ-ron là công cụ ra quyết định hoặc mô hình dữ liệu thống kê phi tuyến tính. Chúng có thể được sử dụng để mô hình hóa các mối quan hệ phức tạp giữa đầu vào và đầu ra hoặc để tìm các mẫu trong dữ liệu. Mạng nơ-ron nhân tạo bao gồm một mạng lưới các phần tử xử lý đơn giản (nơ-ron nhân tạo)

có thể thể hiện hành vi toàn cục phức tạp, được xác định bởi các kết nối giữa các phần tử xử lý và các tham số phần tử.

Tế bào thần kinh nhân tạo lần đầu tiên được đề xuất vào năm 1943 bởi Warren McCulloch, một nhà sinh lý học thần kinh và Walter Pitts, một nhà logic học, người lần đầu tiên cộng tác tại Đại học Chicago. Một loại mạng nơ-ron nhân tạo cổ điển là mạng Hopfield tái diễn. Khái niệm mạng nơ-ron dường như lần đầu tiên được Alan Turing đề xuất trong bài báo Intelligent Machinery năm 1948.

Tiện ích của các mô hình mạng nơron nhân tạo nằm ở chỗ chúng có thể được sử dụng để suy ra một hàm từ các quan sát và cũng có thể sử dụng nó. Mạng nơ-ron không giám sát cũng có thể được sử dụng để học các biểu diễn của đầu vào nắm bắt các đặc điểm nổi bật của phân phối đầu vào.

Mạng nơ-ron, trong thế giới tài chính, hỗ trợ phát triển các quy trình như dự báo chuỗi thời gian, giao dịch theo thuật toán, phân loại chứng khoán, mô hình hóa rủi ro tín dụng và xây dựng các chỉ số độc quyền và các công cụ phái sinh giá cả. Mạng thần kinh hoạt động tương tự như mạng thần kinh của não người. Một "nơ-ron" trong mạng nơ-ron là một hàm toán học thu thập và phân loại thông tin theo một kiến trúc cụ thể. Mạng này có sự tương đồng mạnh mẽ với các phương pháp thống kê như phân tích đường cong và phân tích hồi quy.

Mạng nơ-ron được sử dụng rộng rãi, với các ứng dụng cho hoạt động tài chính, lập kế hoạch doanh nghiệp, giao dịch, phân tích kinh doanh và bảo trì sản phẩm. Mạng nơ-ron cũng đã được áp dụng rộng rãi trong các ứng dụng kinh doanh như các giải pháp nghiên cứu tiếp thị và dự báo, phát hiện gian lận và đánh giá rủi ro. Mạng nơ-ron đánh giá dữ liệu giá cả và tìm ra cơ hội để đưa ra quyết định thương mại dựa trên phân tích dữ liệu. Các mạng có thể phân biệt sự phụ thuộc lẫn nhau phi tuyến tinh vi và các mẫu mà các phương pháp phân tích kỹ thuật khác không làm được. Theo nghiên cứu, độ chính xác của mạng nơ-ron trong việc đưa ra dự đoán giá cổ phiếu là khác nhau. Một số mô hình dự đoán giá cổ phiếu chính xác từ 50 đến 60 phần trăm trong khi những mô hình khác dự đoán chính xác 70 phần trăm trong tất cả các trường hợp. Một số người đã cho rằng cải thiện 10% hiệu quả là tất cả những

gì nhà đầu tư có thể yêu cầu từ mạng nơ-ron. Sẽ luôn có các tập dữ liệu và các lớp nhiệm vụ được phân tích tốt hơn bằng cách sử dụng các thuật toán đã phát triển trước đó. Thuật toán không quá quan trọng; chính dữ liệu đầu vào được chuẩn bị kỹ lưỡng về chỉ số được nhắm mục tiêu sẽ quyết định cuối cùng mức độ thành công của mạng nơ-ron.

1.2.2 Cây quyết định (Decision Tree)

Cây quyết định là một công cụ hỗ trợ quyết định sử dụng mô hình quyết định dạng cây và các hệ quả có thể xảy ra của chúng, bao gồm cả kết quả sự kiện may rủi, chi phí tài nguyên và tiện ích. Đó là một cách để hiển thị một thuật toán chỉ chứa các câu lệnh điều khiển có điều kiện. Cây quyết định thường được sử dụng trong nghiên cứu hoạt động, đặc biệt là trong phân tích quyết định, để giúp xác định chiến lược có nhiều khả năng đạt được mục tiêu nhất, nhưng cũng là một công cụ phổ biến trong học máy. [10]

Cây quyết định là một cấu trúc giống như lưu đồ, trong đó mỗi nút bên trong đại diện cho một "thử nghiệm" trên một thuộc tính (ví dụ: lật xu xảy ra trước), mỗi nhánh biểu thị kết quả. kết quả của bài kiểm tra và mỗi lá đại diện cho một lớp nhãn (quyết định được đưa ra sau khi tính toán tất cả các thuộc tính). Các đường dẫn từ gốc để biểu diễn kiểu luật phân loại.

Trong phân tích quyết định, cây quyết định và sơ đồ ảnh hưởng có liên quan chặt chẽ được sử dụng như một công cụ hỗ trợ ra quyết định trực quan và phân tích, nơi các giá trị kỳ vọng (hoặc tiện ích kỳ vọng) của các lựa chọn thay thế cạnh tranh được tính toán. Một cây quyết định bao gồm ba loại nút

- Các nút quyết định - thường được biểu diễn bằng hình vuông
- Các nút cơ hội - thường được biểu thị bằng các vòng tròn
- Các nút kết thúc - thường được biểu diễn bằng hình tam giác

Một cây có thể được “học” bằng cách tách tập nguồn thành các tập con dựa trên kiểm tra giá trị thuộc tính. Quá trình này được lặp lại trên mỗi tập con dẫn xuất theo cách đệ quy được gọi là phân vùng đệ quy. Quá trình đệ quy được hoàn thành khi tất cả các tập con tại một nút đều có cùng giá trị của biến mục tiêu hoặc khi việc

tách không còn thêm giá trị vào các dự đoán. Việc xây dựng bộ phân loại cây quyết định không yêu cầu bất kỳ kiến thức miền hoặc thiết lập tham số nào, và do đó thích hợp cho việc khám phá kiến thức khám phá. Cây quyết định có thể xử lý dữ liệu chiều cao. Nhìn chung bộ phân loại cây quyết định có độ chính xác tốt. Quy nạp cây quyết định là một cách tiếp cận quy nạp điển hình để tìm hiểu kiến thức về phân loại.

Cây quyết định phân loại các cá thể bằng cách sắp xếp chúng theo cây từ gốc đến một số nút lá, điều này cung cấp sự phân loại của cá thể. Một thể hiện được phân loại bằng cách bắt đầu từ nút gốc của cây, kiểm tra thuộc tính được chỉ định bởi nút này, sau đó di chuyển xuống nhánh cây tương ứng với giá trị của thuộc tính như trong hình trên. Quá trình này sau đó được lặp lại đối với cây con bắt nguồn từ nút mới.

Điểm mạnh của phương pháp cây quyết định là:

- Cây quyết định có thể tạo ra các quy tắc dễ hiểu.
- Cây quyết định thực hiện phân loại mà không cần tính toán nhiều.
- Cây quyết định có thể xử lý cả biến liên tục và biến phân loại.
- Cây quyết định cung cấp một dấu hiệu rõ ràng về các trường nào là quan trọng nhất để dự đoán hoặc phân loại.

Điểm yếu của phương pháp cây quyết định:

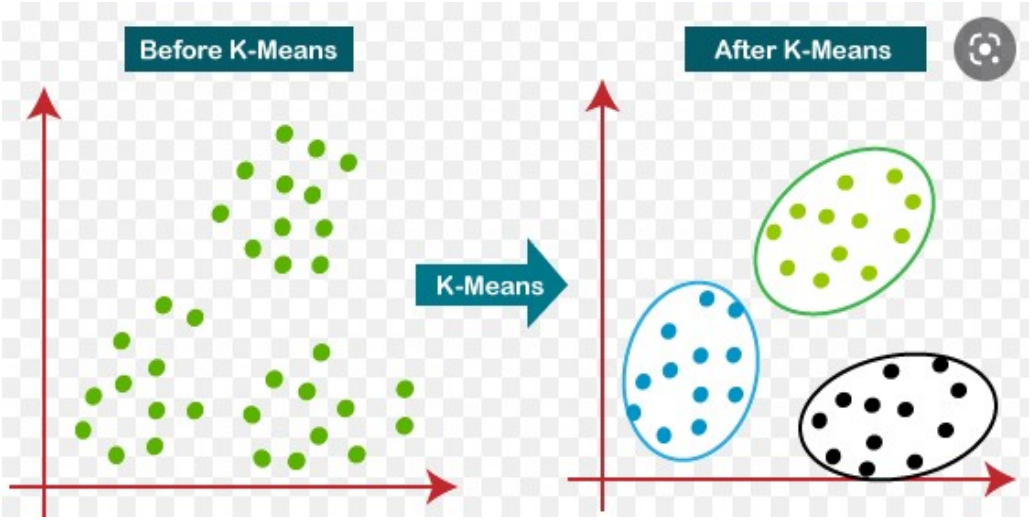
- Cây quyết định ít thích hợp hơn cho các nhiệm vụ ước tính trong đó mục tiêu là dự đoán giá trị của một thuộc tính liên tục.
- Cây quyết định dễ mắc lỗi trong các bài toán phân loại với nhiều lớp và số lượng ví dụ huấn luyện tương đối nhỏ
- Cây quyết định có thể tốn kém về mặt tính toán để đào tạo. Quá trình trồng cây quyết định rất tốn kém về mặt tính toán. Tại mỗi nút, mỗi trường phân tách ứng cử viên phải được sắp xếp trước khi có thể tìm thấy trường phân tách tốt nhất của nó. Trong một số thuật toán, kết hợp các trường được sử dụng và phải thực hiện tìm kiếm để có trọng số kết hợp tối ưu. Các thuật toán cắt tỉa cũng có thể tốn kém vì nhiều cây con ứng cử viên phải được hình thành và so sánh.[11]

1.2.3 K-means clustering

K-Means Clustering là một thuật toán học không giám sát đơn giản và phổ biến được sử dụng để giải quyết các vấn đề phân cụm trong học máy hoặc khoa học dữ liệu. Thông thường, các thuật toán không giám sát đưa ra các suy luận từ tập dữ liệu chỉ sử dụng các vectơ đầu vào mà không đề cập đến các kết quả đã biết hoặc được gắn nhãn.

Mục tiêu của K-means rất đơn giản: nhóm các điểm dữ liệu tương tự lại với nhau. Để đạt được mục tiêu này, K-mean tìm kiếm một số lượng cố định (k) các cụm trong một tập dữ liệu. nhóm các tập dữ liệu không được gắn nhãn thành các cụm khác nhau. Ở đây K là số lượng cụm được xác định trước cần được tạo trong quá trình này, như nếu K = 2, sẽ có hai cụm, và đối với K = 3, sẽ có ba cụm.

Cụm được đề cập đến một tập hợp các điểm dữ liệu được tổng hợp lại với nhau vì có những điểm tương đồng nhất định. Centroid là vị trí đại diện cho trung tâm của cụm. Mọi điểm dữ liệu được phân bổ cho từng cụm với yêu cầu là tổng khoảng cách giữa điểm dữ liệu và các cụm tương ứng của chúng là nhỏ nhất. Nói cách khác, thuật toán K-mean xác định k số centroid, và sau đó phân bổ mọi điểm dữ liệu cho cụm gần nhất, đồng thời giữ các centroid càng nhỏ càng tốt. Ý nghĩa trong K-means đề cập đến giá trị trung bình của dữ liệu; tức là tìm ra điểm trung tâm.



Hình 1.4: Phân cụm bằng K-means

Để sử dụng dữ liệu huấn luyện, quá trình K-means trong Khai phá dữ liệu bắt đầu với nhóm đầu tiên bao gồm các ngẫu nhiên trung tâm được chọn, được sử dụng làm điểm bắt đầu cho tất cả các cụm và sau đó thực hiện các phép tính lặp đi lặp lại để tối ưu hóa vị trí của các trung tâm.

Thuật toán tạm dừng tạo và tối ưu hóa các cụm khi:

- Các centroid đã ổn định không có thay đổi về giá trị của chúng vì việc phân nhóm đã thành công.
- Đã đạt được số lần lặp xác định.

Hoạt động của thuật toán K-Means được giải thích theo các bước dưới đây

- Bước 1: Chọn số K để quyết định số lượng cụm.
- Bước 2: Chọn K điểm hoặc trọng tâm ngẫu nhiên. (Nó có thể khác với tập dữ liệu đầu vào).
- Bước 3: Gán mỗi điểm dữ liệu cho trung tâm gần nhất của chúng, sẽ tạo thành các cụm K được xác định trước.
- Bước 4: Tính toán phương sai và đặt một trung tâm mới của mỗi cụm.
- Bước 5: Lặp lại các bước thứ ba, có nghĩa là chỉ định lại mỗi điểm dữ liệu cho trung tâm gần nhất mới của mỗi cụm.
- Bước 6: Nếu có bất kỳ sự phân công lại nào xảy ra, hãy chuyển sang bước 4, sau đó chuyển đến hoàn tất.
- Bước 7: Mô hình đã sẵn sàng.

Hiệu suất của thuật toán phân cụm K-mean phụ thuộc vào các cụm hiệu quả cao mà nó tạo thành. Nhưng chọn số lượng cụm tối ưu là một nhiệm vụ lớn. Có một số cách khác nhau để tìm số lượng cụm tối ưu, phương pháp thích hợp nhất để tìm số lượng cụm hoặc giá trị của K. Phương pháp được đưa ra dưới đây:

Elbow Method

Phương pháp Elbow là một trong những cách phổ biến nhất để tìm ra số lượng cụm tối ưu. Phương pháp này sử dụng khái niệm giá trị WCSS. WCSS là viết tắt của

Within Cluster Sum of Squares, xác định tổng các biến trong một cụm. Công thức tính giá trị của WCSS (cho 3 cụm) được đưa ra dưới đây:

$WCSS=\Sigma_{Pi\ in\ Cluster1}\ distance(P_iC1)^2 + \Sigma_{Pi\ in\ Cluster2}\ distance(P_iC_2)^2 + \Sigma_{Pi\ in\ Cluster3}\ distance(P_iC_3)^2$ (1.1)

C: là tâm điểm của mỗi cụm

P: là mỗi điểm dữ liệu trong cụm

Silhouette Method

Phương pháp hình bóng cũng là một phương pháp để tìm số tối ưu của các cụm và giải thích và xác nhận tính nhất quán trong các cụm dữ liệu. Phương pháp hình bóng tính toán các hệ số hình bóng của mỗi điểm đo bao nhiêu điểm tương tự với cụm của chính nó so với các cụm khác. bằng cách cung cấp một biểu diễn đồ họa ngắn gọn về mức độ phân loại của từng đối tượng. Tính toán hệ số hình bóng cho từng điểm và lấy trung bình cộng cho tất cả các mẫu để có được điểm hình bóng .

Giá trị hình bóng là thước đo mức độ tương tự của một đối tượng với cụm của chính nó (sự gắn kết) so với các cụm khác (sự tách biệt). Giá trị của hình bóng nằm trong khoảng [1, -1], trong đó giá trị cao chỉ ra rằng đối tượng được đối sánh tốt với cụm của chính nó và đối sánh kém với các cụm lân cận. Nếu hầu hết các đối tượng có giá trị cao, thì cấu hình phân cụm là thích hợp. Nếu nhiều điểm có giá trị thấp hoặc âm, thì cấu hình phân cụm có thể có quá nhiều hoặc quá ít cụm.

1.3 Các công trình nghiên cứu có liên quan

Hiểu biết được những khái niệm về log, kiến thức tổng quan về các giao thức giám sát lỗi mạng, các phần mềm, ứng dụng dùng để ghi log, tìm hiểu các kỹ thuật học máy để giải quyết các bài toán phân loại, phân cụm dữ liệu.

Năm 2020, Shilin He, Jieming Zhu, Pinjia He, Michael R. Lyu công bố các nghiên cứu về bộ sưu tập lớn về tập dữ liệu log hệ thống tiến tới phân tích log tự động. Log đã được áp dụng rộng rãi trong hệ thống phần mềm tiên tiến vì thông tin, thời gian hoạt động hệ thống phong phú của log. Trong những năm gần đây, sự gia tăng của kích thước phần mềm và dẫn đến sự tăng kích thước nhanh chóng của log. Đến xử lý khối lượng lớn log này một cách hiệu quả, một dòng nghiên cứu tập trung

vào phân tích log thông minh được hỗ trợ bởi các kỹ thuật AI (trí tuệ nhân tạo). Tuy nhiên, chỉ một phần nhỏ trong số những kỹ thuật này đã được triển khai thành công trong ngành vì thiếu bộ dữ liệu log công khai và điểm chuẩn cần thiết dựa trên chúng. Để lấp đầy khoảng cách đáng kể này giữa học thuật và ngành công nghiệp và cũng tạo điều kiện cho nghiên cứu nhiều hơn về phân tích log được hỗ trợ bởi AI, nghiên cứu đã thu thập và tổ chức loghub, một bộ sưu tập lớn tập dữ liệu. Đặc biệt, loghub cung cấp 17 bộ dữ liệu log trong thế giới thực được thu thập từ một loạt các hệ thống, bao gồm hệ thống phân tán, siêu máy tính, hệ điều hành, hệ thống di động, máy chủ ứng dụng và phần mềm độc lập. Nghiên cúu đã tóm tắt số liệu thống kê của các bộ dữ liệu này, giới thiệu một số log thực tế các tình huống sử dụng và trình bày một nghiên cứu điển hình về phát hiện bất thường để chứng minh cách loghub tạo điều kiện cho việc nghiên cứu và thực hành trong vùng này. Bộ dữ liệu loghub đã được tải xuống hơn 15.000 lần bởi hơn 380 tổ chức, doanh nghiệp và học thuật.

Loghub duy trì một bộ sưu tập các bản ghi hệ thống, được miễn phí có thể truy cập cho các mục đích nghiên cứu. Một số log được phát hành từ các nghiên cứu trước đây, trong khi một số dữ liệu khác được thu thập từ các hệ thống thực trong môi trường phòng thí nghiệm của tác giả. Tổng số bản ghi lên tới hơn 77 GB. Do quy mô lớn, tác giả lưu trữ một mẫu nhỏ cho mỗi tập dữ liệu trong trang web dự án, trong khi phiên bản đầy đủ có thể được yêu cầu thông qua Zenodo, một trang web chia sẻ tập dữ liệu.

Năm 2021, một nghiên cứu của PGS.TS Trần Mạnh Hà và TS Nguyễn Văn Sinh công bố bài báo “An automated fault detection system for communication networks and distributed systems”. Tự động phát hiện lỗi trong mạng truyền thông và hệ thống phân tán là một quá trình đầy thách thức thường đòi hỏi sự tham gia của các công cụ hỗ trợ và chuyên môn của người vận hành hệ thống. Giám sát sự kiện tự động và tương quan hệ thống tạo ra dữ liệu sự kiện được chuyển tiếp đến người vận hành hệ thống để phân tích các sự kiện lỗi và tạo báo cáo lỗi. Các phương pháp học máy không chỉ giúp phân tích dữ liệu sự kiện chính xác hơn mà còn dự báo các sự kiện lỗi có thể xảy ra bằng cách học hỏi từ những lỗi hiện có. Nghiên cứu này giới

thiệu một hệ thống phát hiện lỗi tự động hỗ trợ người vận hành hệ thống phát hiện và dự báo lỗi. Hệ thống này được đặc trưng bởi khả năng khai thác tài nguyên kiến thức lỗi tại các kho lưu trữ trực tuyến khác nhau, các sự kiện log và các thông số trạng thái từ hệ thống được giám sát; và áp dụng phân tích lỗi và sự kiện các phương pháp lọc để đánh giá sự kiện và dự báo lỗi. Hệ thống chứa một mô hình dữ liệu lỗi để thu thập các báo cáo lỗi, một tính năng và phương pháp lọc ngữ nghĩa để tương quan các sự kiện nhật ký và phương pháp học máy để đánh giá mức độ nghiêm trọng, mức độ ưu tiên và mối quan hệ của các sự kiện nhật ký và dự báo các lỗi nghiêm trọng sắp tới của hệ thống được giám sát. Nghiên cứu đã đánh giá thực hiện tạo mẫu của hệ thống được đề xuất trên hệ thống cụm máy tính hiệu suất cao và cung cấp phân tích chuyên sâu.

Phát hiện lỗi trong mạng truyền thông và phân tán hệ thống thường yêu cầu sự tham gia của các công cụ hỗ trợ và chuyên môn của người vận hành hệ thống. Giám sát và sự kiện hệ thống tương quan tạo ra các sự kiện lỗi được chuyển tiếp cho người vận hành hệ thống để phân tích và tạo báo cáo lỗi. Tự động hóa các chức năng phát hiện lỗi là một thách thức vì thiếu một cách tiếp cận hiệu quả để thay thế kiến thức và cơ chế suy luận của người vận hành hệ thống một số vấn đề liên quan đến tính khả dụng, khả năng chịu lỗi và khả năng dự đoán hiệu suất [9] rất khó phát hiện trên diện rộng mạng truyền thông và hệ thống phân tán với độ phức tạp, khả năng mở rộng và tầm quan trọng. Sự phát triển của phát hiện lỗi đại khái có thể được chia thành ba giai đoạn: phát hiện, phòng ngừa và dự đoán. Các mạng doanh nghiệp quy mô nhỏ thụ động chờ đợi sự cố xảy ra và phụ thuộc nhiều vào người vận hành hệ thống để tìm và khắc phục sự cố; công cụ giám sát sử dụng các giao thức quản lý tiêu chuẩn bao gồm SNMP để nhận các sự kiện nhật ký. Giai đoạn phòng ngừa được đặc trưng bằng bán tự động hóa: các mạng kinh doanh quy mô vừa áp dụng cấu hình tự động bao gồm netconf, cfengine để ngăn sự cố xảy ra, sử dụng các công cụ giám sát để lấy log sự kiện và dữ liệu hệ thống; các công cụ hỗ trợ cung cấp phân tích để người vận hành hệ thống tìm và khắc phục sự cố một cách nhanh chóng. Các giai đoạn dự đoán được phân biệt bởi tính tự động hóa cao: mạng doanh nghiệp chủ động dự báo các vấn đề

cần tránh, sử dụng các công cụ giám sát để thu thập các sự kiện nhật ký, hệ thống và dữ liệu ngữ cảnh; các công cụ hỗ trợ liên quan đến kỹ thuật học máy cung cấp đánh giá và dự đoán cho hệ thống các nhà khai thác để tránh sự cố. Ý tưởng xây dựng hệ thống phát hiện lỗi tự động xuất phát từ hệ thống miễn dịch của con người. Một người đàn ông sau bị nhiễm trùng, nhưng trước khi trở thành bệnh nặng thường phát ra một loạt các triệu chứng bao gồm ho, chảy nước mũi, mệt mỏi, v.v., và đối mặt với nhiều yếu tố bối cảnh bao gồm thay đổi thời tiết, áp lực công việc, v.v. các triệu chứng và yếu tố như vậy rất thường theo sau bởi bệnh tật. Tương tự, một hệ thống trước khi gặp lỗi thường tạo ra các sự kiện bao gồm thông báo, cảnh báo, lỗi, v.v. và sở hữu rất nhiều yếu tố ngữ cảnh bao gồm số lượng quy trình, sử dụng bộ nhớ, thời gian xử lý, v.v. Phân tích những sự kiện và yếu tố này giúp dự báo thất bại với khả năng. Với sự phát triển của các kỹ thuật máy học để khai thác dữ liệu, các phương pháp tiếp cận gần đây có lợi thế trong số các kỹ thuật này để khai thác các lỗi khác nhau và dữ liệu hệ thống để đánh giá và xếp loại. Chúng tôi đã đề xuất một phương pháp phát hiện lỗi tự động cho giao tiếp lớn mạng và hệ thống phân tán trong nghiên cứu này. Chúng ta có trước đây đã thực hiện một số hoạt động nghiên cứu liên quan đến cách tiếp cận này. Lược đồ lỗi trước đó trong nghiên cứu bao gồm một tập hợp các tính năng để đại diện cho các báo cáo lỗi từ lỗi hệ thống theo dõi. Nghiên cứu về phương pháp lọc ngữ nghĩa thích ứng tương quan các sự kiện nhật ký để loại bỏ dư thừa dữ liệu sự kiện. Các phương pháp phân loại trong nghiên cứu đánh giá các báo cáo lỗi dựa trên mức độ nghiêm trọng và các tính năng ưu tiên để phát hiện lỗi. Cách tiếp cận trong nghiên cứu này kế thừa những hoạt động nghiên cứu này và là hình ảnh thu nhỏ nỗ lực của chúng tôi để giải quyết các vấn đề sau: khai thác dữ liệu ngữ cảnh hoặc các thông số trạng thái hệ thống cùng với dữ liệu sự kiện lỗi và nhật ký, và áp dụng các phương pháp phân loại trích xuất các đối tượng đặc trưng để tự động dự đoán. Sự đóng góp với nghiên cứu này thể hiện qua ba việc.

Mở rộng lược đồ lỗi trước đó để bao gồm ngữ cảnh dữ liệu và khai thác các tính năng được trích xuất từ giản đồ để cải thiện các phương pháp lọc và phân loại.

Đề xuất một phương pháp phát hiện lỗi tự động cho phát hiện lỗi dựa trên mức độ nghiêm trọng và mức độ ưu tiên các tính năng và dự đoán các lỗi có thể xảy ra trong hệ thống được giám sát dựa trên tính năng quan hệ.

Phát triển một nguyên mẫu của hệ thống phát hiện lỗi áp dụng giản đồ lỗi mở rộng, phương pháp lọc ngữ nghĩa và phương pháp rừng ngẫu nhiên để khai thác các báo cáo lỗi phần mềm mã nguồn mở, nhật ký sự kiện dữ liệu và dữ liệu ngữ cảnh.

1.4 Kết luận chương

Hiểu biết được những khái niệm về log, kiến thức tổng quan về các giao thức giám sát lỗi mạng, các phần mềm, ứng dụng dùng để ghi log, tìm hiểu các kỹ thuật học máy để giải quyết các bài toán phân loại, phân cụm dữ liệu.

CHƯƠNG 2 - GIẢI PHÁP PHÂN LOẠI VÀ MÔ HÌNH DỮ LIỆU CẢNH BÁO

2.1 Giới thiệu chương

Trong chương này xin giới thiệu các giải pháp phân loại, phân cụm dữ liệu log và mô hình dữ liệu cảnh báo.

2.2 Mô hình dữ liệu

2.2.1 Mô tả dữ liệu đầu vào

Luận văn này đề xuất sử dụng dữ liệu log được lấy từ nguồn dự án nghiên cứu Loghub, LogPAI [12], nghiên cứu dựa này dựa vào nền tảng trí tuệ nhân tạo mã nguồn mở cung cấp một bộ sưu tập lớn dữ liệu log của nhiều hệ thống khác nhau và được dùng để phân tích log tự động. Nhiều hoạt động nghiên cứu đã thực hiện thành công và hiệu quả khi áp dụng phương pháp học máy trên nền tảng và dữ liệu log của dự án này cho các mục đích khác nhau bao gồm phát hiện bất thường hoặc xác định vấn đề lỗi. Nghiên cứu của luận văn cũng sử dụng dữ liệu log từ hệ thống HDFS trong Loghub để thử nghiệm. Dự án nghiên cứu Loghub chia sẻ một bộ sưu tập các bản ghi log hệ thống được đăng tải miễn phí [13]. Dữ liệu log HDFS này chứa các file log thu được từ hệ thống HDFS tại 33 điểm ở một trường đại học.

Bảng 2.1: Báo cáo thống kê về dữ liệu log file

Số lượng log file	33
Kích thước log file (GB)	16.05
Số lượng bản tin log	58095163
Số lượng bản tin INFO	57570609
Số lượng bản tin log WARN	500971
Số lượng bản tin log ERROR	24030
Số lượng bản tin log FATAL	8019

Các bản tin log trong một hệ thống sẽ xuất liên tục, số lượng log là rất lớn. Vì thế để dễ dàng cho việc phân tích mức độ nghiêm trọng của bản tin log, mỗi một log đều có thuộc tính mức độ cảnh báo để nhận biết mức độ quan trọng của dòng log đó.

Mức độ nghiêm trọng của bản tin log của hệ thống HDFS có các giá trị như sau:

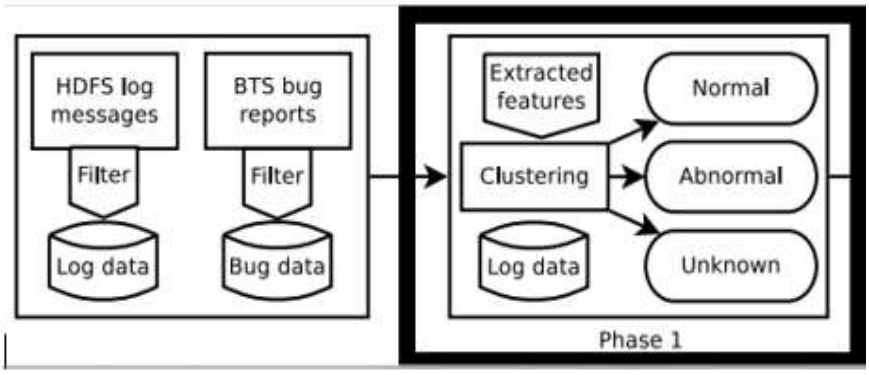
FATAL: Lỗi được hiển thị trên bảng điều khiển trạng thái và có thể gây dừng ứng dụng hoặc hệ thống.

WARN: Cảnh báo tình trạng không mong muốn được hiển thị trên bảng điều khiển trạng thái và đưa ra khả năng có những nguy cơ gây nguy hiểm hệ thống.

INFO: Thông điệp thông báo thông thường trong ứng dụng hoặc tiến trình hệ thống được hiển thị trên bảng điều khiển trạng thái. •

DEBUG: Thông tin chi tiết của một sự kiện để gỡ lỗi ứng dụng hoặc hệ thống được ghi duy nhất vào log.

TRACE: Thông tin chi tiết hơn DEBUG để giúp gỡ lỗi ứng dụng hoặc hệ thống được ghi duy nhất vào log



Hình 2.1: Mô tả thiết kế phát hiện log bất thường

Dữ liệu đầu vào trong luận văn này bao gồm các bản tin log khác nhau của hệ thống HDFS với các mức độ nghiêm trọng theo các cấp độ là INFO, WARN, ERROR và FATAL.

Vì các bản tin log INFO có số lượng rất lớn trong hệ thống và hầu hết là không có nhiều giá trị về mặt bất thường của hệ thống, mang tính chất thông tin về hệ thống hơn các bản tin log khác là cảnh báo nguy cơ nên luận văn đề xuất cách tiếp cận là lọc bản tin log INFO ra, song song đó là loại bỏ các bản tin log bị lặp lại và xử lý các bản tin còn lại để đưa vào thuật toán phân cụm. Khi đưa vào mô hình thì đầu vào sẽ là dữ liệu log đã xử lý và đầu ra là các dữ liệu bản tin log bất thường. [14].

Dữ liệu log sẽ được phân loại dựa vào phương pháp phân cụm để chia các dữ liệu log thành 3 loại chính:

- Log bình thường
- Log bất thường
- Log chưa xác định

2.3 Giải pháp phân loại

Để thuận tiện cho việc phân tích thì dữ liệu đầu vào đưa mô hình sau bước lọc dữ liệu thô không cần thiết ban đầu như đã nói ở chương trên, bước tiếp theo là phải phân loại và trích xuất các tính chất của bản tin log dựa vào đặc trưng của các trường thuộc tính của log.

```
2017-01-26 20:01:44 WARN org.apache.
hadoop.hdfs.server.datanode.DataNode:
Slow BlockReceiver write data to disk
cost:892ms (threshold=300ms)
2017-01-26 20:01:47 WARN org.apache.
hadoop.hdfs.server.datanode.DataNode:
Slow manageWriterOsCache took 822ms
(threshold=300ms)
2017-01-26 20:01:50 WARN org.apache.
hadoop.hdfs.server.datanode.DataNode:
Slow BlockReceiver write data to disk
cost:1653ms (threshold=300ms)
```

Hình 2.2: Cấu trúc của 1 bản tin log WARN trong hệ thống HDFS

Các đặc điểm cơ bản có trong bản tin log WARN ở trên bao gồm

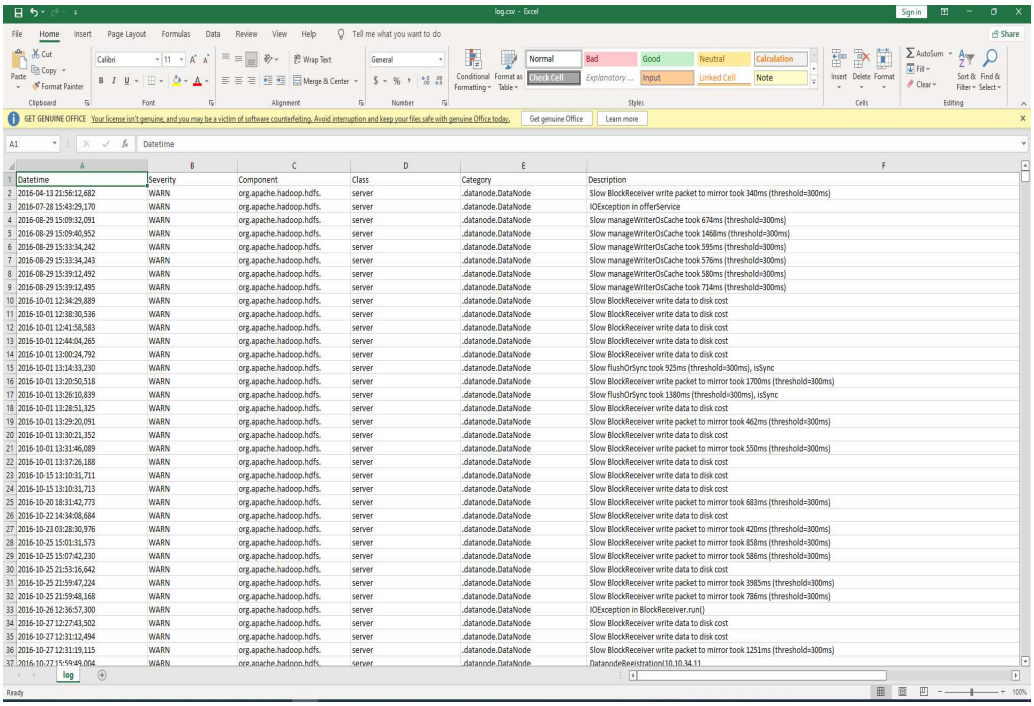
- Ngày tháng năm và giờ xuất log: 2017-01-26 20:01:44

- Mức độ cảnh báo: WARN
- Nơi xuất log: org.apache.hadoop.hdfs.server
- Diễn tả vấn đề lỗi: Slow BlockReceiver write data to disk cost.

Mỗi một thuộc tính của log sẽ được phân biệt bởi khoảng trắng hoặc dấu : tất cả các log đều sẽ bao gồm các thông tin rõ ràng thời gian, loại cảnh báo, nơi xuất cảnh báo và diễn giải vấn đề cảnh báo đang tồn tại trong hệ thống.

Dựa vào các đặc điểm chính, thuộc tính của bản tin log ta sẽ phân loại dữ liệu log theo các đặc trưng, định nghĩa các thuộc tính, đồng bộ các trường dữ liệu đó thành một nội dung hoàn chỉnh để đưa vào thuật toán.

Dữ liệu log sau khi thi thập từ các hệ thống, lọc các dữ liệu dư thừa không cần thiết và phân loại được lưu dưới dạng log.csv như hình dưới đây



Datetime	Severity	Component	Class	Category	Description
2016-04-13 21:56:12,682	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 340ms (threshold=300ms)
2016-07-28 15:43:29,170	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	IOException in offerService
2016-08-29 15:09:32,091	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow manageWriterOsCache took 674ms (threshold=300ms)
2016-08-29 15:09:40,952	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow manageWriterOsCache took 1468ms (threshold=300ms)
2016-08-29 15:33:34,242	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow manageWriterOsCache took 595ms (threshold=300ms)
2016-08-29 15:33:34,243	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow manageWriterOsCache took 576ms (threshold=300ms)
2016-08-29 15:39:12,492	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow manageWriterOsCache took 580ms (threshold=300ms)
2016-08-29 15:39:12,495	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow manageWriterOsCache took 714ms (threshold=300ms)
2016-10-01 12:34:25,889	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-01 12:38:30,536	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-01 12:41:58,583	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-01 12:44:04,265	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-01 13:00:24,792	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-01 13:14:33,230	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow flushOrSync took 925ms (threshold=300ms), isSync
2016-10-01 13:20:50,518	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 1700ms (threshold=300ms)
2016-10-01 13:26:10,839	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow flushOrSync took 1380ms (threshold=300ms), isSync
2016-10-01 13:28:51,325	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-01 13:29:20,091	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 462ms (threshold=300ms)
2016-10-01 13:30:21,352	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-01 13:31:46,089	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 550ms (threshold=300ms)
2016-10-01 13:37:26,188	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-15 13:10:31,711	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-15 13:10:31,713	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-20 18:31:42,773	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 683ms (threshold=300ms)
2016-10-22 14:34:08,684	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-23 03:28:30,976	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 420ms (threshold=300ms)
2016-10-25 15:01:31,573	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 858ms (threshold=300ms)
2016-10-25 15:07:42,230	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 586ms (threshold=300ms)
2016-10-25 21:53:16,642	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-25 21:59:47,224	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 3985ms (threshold=300ms)
2016-10-25 21:59:48,168	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 786ms (threshold=300ms)
2016-10-26 12:36:57,300	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	IOException in BlockReceiver.run()
2016-10-27 12:27:43,302	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-27 12:31:12,494	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write data to disk cost
2016-10-27 12:31:19,115	WARN	org.apache.hadoop.hdfs.	server	.datanode.DataNode	Slow BlockReceiver write packet to mirror took 1251ms (threshold=300ms)

Hình 2.3: Dữ liệu log

Bảng dưới đây trình bày danh sách trích xuất các đặc trưng của log được sử dụng để phân cụm các bản tin log

Bảng 2.2: Danh sách trích xuất các thuộc tính của log

Feature	Description	Type
datetime	Ngày giờ xuất ra log	Ngày giờ
severity	Mức độ ảnh hưởng	Liệt kê
component	Thành phần nơi xảy ra	Liệt kê
class	Cấp độ nơi xảy ra	Liệt kê
keyword	Các cụm từ khác nhau	Chuỗi
category	Danh mục log	Liệt kê
repetition	Dữ liệu log lặp lại	Liệt kê

Các thuộc tính ngày và giờ ở định dạng yy/MM/dd HH:mm:ss thì được gộp lại thành một và nó là một thuộc tính được thêm vào để giảm bản tin log lặp lại. Từ các bản tin log lặp lại sẽ tính ra số lần lặp lại của cùng bản tin trong cùng một khoảng thời gian. Thuộc tính lặp lại có thể dựa theo các giá trị: không lặp, không lặp liên tục và lặp lại cao.

Mức độ nghiêm trọng ảnh hưởng hệ thống (SEVERITY) tập trung vào ba giá trị chính đó là: FATAL, ERROR và WARN. Đây là ba loại log có tiềm tàng nguy cơ trở thành cảnh báo những bất thường trong hệ thống mạng. [15].

Tên thành phần (COMPONENT) và loại (CLASS) nơi xuất ra bản tin log được phân tách thành hai thuộc tính

Ví dụ: org.apache.hadoop.ipc.Server sẽ bao gồm

- org.apache.hadoop.ipc: là tên thành phần (COMPONENT)
- Server: là tên loại (CLASS)

Thuộc tính từ khóa (KEYWORD) chứa các từ quan trọng hoặc cụm từ quan trọng từ nội dung được trình bày chi tiết của bản tin log.

Trong quá trình xử lý và đánh giá từ khóa theo kỹ thuật TF-IDF, danh mục thuộc tính được xác định bằng đặc điểm từ khóa, ví dụ: Bộ nhớ, Đĩa, Bộ nhớ đệm, IO, Quy trình, v.v.

Các thuộc tính dạng dữ liệu liệt kê là khả thi để đào tạo và đánh giá phân loại. Các tính năng văn bản cần lọc ra từ khóa thì yêu cầu các bước xử lý tiếp theo để chuyển đổi dữ liệu thô sang dữ liệu khả thi. Các bước này bao gồm loại bỏ các mục dữ liệu thừa, cung cấp dữ liệu bị thiếu các mục, định dạng lại các mục dữ liệu từ các kiểu dữ liệu khác nhau để liệt kê kiểu dữ liệu. Tập dữ liệu được tải đầu tiên vào dữ liệu khung cho phép các mục dữ liệu được thao tác dễ dàng. Sau đó, tập dữ liệu được trích xuất bởi các đặc trưng quan trọng. Nội dung các từ khóa đặc trưng thường được chứa các chuỗi dài để mô tả chi tiết lỗi.

2.4 Kỹ thuật TFx IDF

TF-IDF (Term Frequency – Inverse Document Frequency) là kĩ thuật sử dụng trong khai phá dữ liệu văn bản để có được các từ khóa quan trọng. Các từ khóa riêng biệt có ít liên quan và từ khóa có trọng số cao tức là có ý nghĩa giá trị cao. Trọng số được dùng để đánh giá sự 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.[16]

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).

IDF – inverse document frequency. Tần số nghịch của 1 từ trong tập văn bản (corpus). Tính IDF để giảm giá trị của những từ phổ biến. Mỗi từ chỉ có 1 giá trị IDF duy nhất trong tập văn bản.

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).

2.5 Tổng kết chương

Trong chương này thông qua việc nghiên cứu tìm hiểu được một số thuật toán và những công trình liên quan tới phân tích dữ liệu, từ đó giúp luận văn này hiểu rõ hơn về phân tích dữ liệu log, từ đó hiểu được những ưu nhược điểm của các thuật toán, và các cách xử lý phân loại, tạo tiền đề và cơ sở vững chắc cho nghiên cứu của đề tài luận văn này.

CHƯƠNG 3 - ĐỀ XUẤT THUẬT TOÁN PHÂN TÍCH DỮ LIỆU LOG ĐỂ PHÁT HIỆN CẢNH BÁO BẤT THƯỜNG TRONG HỆ THỐNG MẠNG

3.1 Giới thiệu chương

Chương 3 sẽ trình bày về các kỹ thuật học máy sử dụng để phân tích dữ liệu log, cách filter các dữ liệu log đầu vào, quá trình xử lý trích xuất các đặc trưng của dữ liệu log.

3.2 Thuật toán đề xuất

Luận văn này đề xuất sử dụng phương pháp phân cụm để phát hiện các bất thường trên hệ thống mạng và truyền thông. Dữ liệu khai thác từ bản tin log của hệ thống. Sẽ áp dụng phương pháp phân cụm K-means để chia dữ liệu bản tin log thành ba cụm:

- Cụm bình thường chứa các thông báo log không thể liên kết đến các lỗi
- Cụm bất thường chứa các thông báo log có thể liên kết đến các lỗi
- Cụm không xác định chứa thông báo log cần điều tra thêm.

Hầu hết các hoạt động nghiên cứu trước đây đều có ứng dụng máy phương pháp học tập để khai thác các tập dữ liệu đơn lẻ bao gồm dữ liệu nhật ký, dữ liệu lỗi hoặc dữ liệu cụ thể khác. Cái này phương pháp tiếp cận tập trung nhiều hơn vào phát hiện lỗi hai giai đoạn tuân theo quy trình quản lý lỗi tiến trình: lọc và phân vùng thông báo nhật ký thành các cụm bình thường, bất thường hoặc không xác định định

Thuật toán 1: Xây dựng các cụm cho dữ liệu log

Đầu vào: Tập dữ liệu X và số lượng cụm K

Đầu ra: Danh sách các tâm điểm M và các điểm dữ liệu Y thuộc chúng.

Các bước:

Bước 1: Chọn ngẫu nhiên K điểm làm tâm điểm ban đầu

Bước 2: Gán mỗi điểm dữ liệu cho một cụm có tâm gần nó nhất

Bước 3: Dừng thuật toán nếu không còn có thay đổi

Bước 4: Tính giá trị trung bình của tất cả các điểm dữ liệu trong các cụm

Bước 5: Cập nhật các tâm điểm cho các cụm K

Bước 6: Lặp lại bước 2

Trả về kết quả: Xác định số tâm điểm M và các điểm dữ liệu Y

Thuật toán 1 trình bày các bước để xây dựng các cụm cho các dữ liệu log. Thuật toán bắt đầu với K centroid, trong đó mỗi centroid là một vectơ gồm d phần tử giá trị ban đầu ngẫu nhiên (Bước 1). Sử dụng Euclide distance, mỗi bản tin log có d đặc trưng dưới dạng vectơ được gán cho một cụm có khoảng cách gần nhất với tâm của cụm (Bước 2). Thuật toán dừng nếu việc gán các bản tin log thành các cụm không còn thay đổi (Bước 3). Nếu không, thuật toán tiếp tục cập nhật các tâm điểm mới cho các cụm bằng cách tính toán các giá trị trung bình của tất cả các các bản tin log trong các cụm (Bước 4 & Bước 5) và sau đó lặp lại Bước 2. Cuối cùng là ra kết quả là danh sách các tâm điểm M và tập hợp các bản tin log Y của từng cụm.

Phương pháp phân cụm K-mean nhằm mục đích phân vùng điểm dữ liệu thành các cụm sao cho điểm dữ liệu trong cùng một cụm chia sẻ các đặc trưng giống nhau. Phương pháp học không giám sát này không có biết về nhãn của các điểm dữ liệu. Giả sử tập dữ liệu $X = [x_1, ..., x_N]$ của N số lượng log; mỗi bản tin log biểu diễn một vectơ $xi = [x_{i1}, ..., x_{id}]$, trong đó d biểu thị số trích xuất các tính năng của một thông báo nhật ký; $K<N$ biểu thị số lượng các cụm. Phương pháp này tìm kiếm tâm cụm $M = [m_1, ..., m_K]$ và thông báo kiểu dữ liệu của chúng, ví dụ: nhãn bình thường, bất thường hoặc không xác định.

Luận văn đã sử dụng python và một số thư viện sklearn, pandas, numpy, v.v. để lọc và xử lý dữ liệu log và thực hiện phân cụm K-mean. Điều cần thiết là trích xuất các thuộc tính, đặc trưng của log vì các phương pháp sử dụng chỉ áp dụng cho kiểu

dữ liệu phân loại theo kiểu phân loại thứ tự, phân loại danh nghĩa, hoặc có đặc điểm liên tục. Tuy nhiên, dữ liệu log thường chứa các đặc điểm văn bản như tiêu đề, mô tả hoặc đoạn văn, v.v. chứa những thông tin quan trọng để khai thác.

Thuật toán 2: Chọn các từ khóa riêng biệt cho dữ liệu log

Đầu vào: Bộ từ khóa thô (tiêu đề, mô tả, đoạn văn, v.v.)

Đầu ra: Bộ từ khóa riêng biệt có trọng số.

Các bước:

Bước 1: Tải bộ từ khóa gốc

Bước 2: Loại bỏ các từ lặp thường xuyên hoặc thừa, gây nhiễu bằng stop-word

Bước 3: Giảm bớt các từ bị nhầm lẫn với cách viết gốc và bổ sung

Bước 4: Xóa các từ vô nghĩa bằng biểu thức chính quy

Bước 5: Xử lý tf × idf trên bộ từ khóa đã lọc

Bước 6: Chọn các từ khóa riêng biệt có trọng số cao

Trả về kết quả: Bộ từ khóa riêng biệt có trọng số

Áp dụng các phương pháp xử lý văn bản cho các tính năng văn bản. Thuật toán 2 chọn các từ khóa riêng biệt với trọng số từ các dữ liệu log. Các thuật toán bắt đầu với việc tải bộ từ khóa gốc (Bước 1), áp dụng một số bước để loại bỏ các từ không quan trọng và sửa các từ được chọn lọc (Bước 2, 3, 4) và tạo bộ từ khóa đã lọc. Thuật toán này sau đó sử dụng kỹ thuật tf×idf để đánh giá trọng số bộ từ khóa đã lọc (Bước 5) và trả về các từ khóa riêng biệt với trọng số cao (Bước 6).

3.3 Các bước thực hiện

3.3.1 Import các thư viện cần thiết

Trong luận văn sử dụng ngôn ngữ lập trình Python với các bộ thư viện để xử lý dữ liệu như:

Scikit-learn là một thư viện ngôn ngữ lập trình Python về trí tuệ nhân tạo giúp áp dụng các thuật toán máy học tiện lợi nhanh chóng hơn [17]. Nó có các thư viện thuật toán có sẵn để phân loại đối tượng, xây dựng hồi quy, nhóm các đối tượng tương tự thành tập hợp như phân cụm, giảm số lượng biến ngẫu nhiên, xử lý trước dữ liệu và có thể so sánh, chọn mô hình.

Numpy: Xử lý mảng đa chiều, ma trận

Pandas: Xử lý và trực quan dữ liệu có cấu trúc

Matplotlib: Thư viện vẽ đồ thị

Seaborn: Đồ thị hóa dữ liệu

Ngoài ra còn có các thư viện khác về toán học như math, scipy.sparse và các thuật toán học máy để phân cụm dữ liệu là K-means [18], tìm số lượng cụm tối ưu cho thuật toán K-means như hệ số Silhouette hoặc phương pháp Elbow

3.3.2 Import dữ liệu log và rút trích thuộc tính quan trọng bằng TF-IDF

Dữ liệu được xử lý trên Colaboratory hay còn được biết đến với tên gọi là Google Colab, là một dịch vụ đám mây từ Google Research, nó cho phép thực thi các đoạn code ngôn ngữ lập trình python thông qua trình duyệt web, rất phù hợp với phân tích dữ liệu, học máy và cho giáo dục. Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, có thể sử dụng, chia sẻ tài nguyên máy tính của Google từ CPU tốc độ cao và cả GPU và cả TPU.

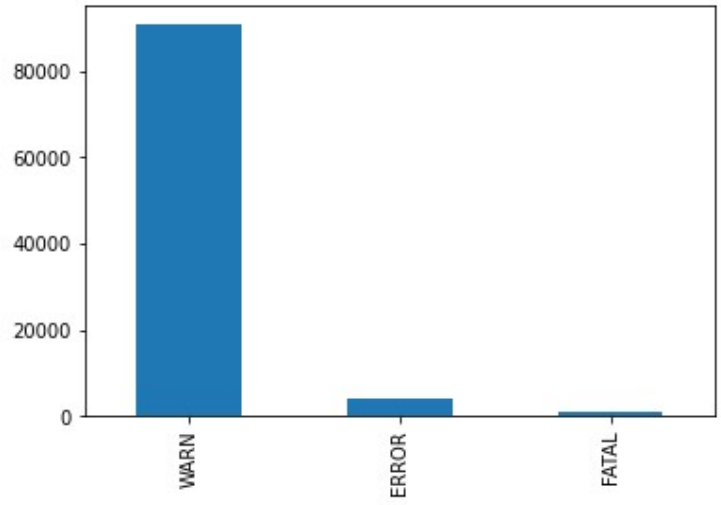
	Datetime	Severity	Component	Class	Category	Description
0	2015-09-19 11:20:12,984	ERROR	org.apache.hadoop.hdfs	server	datanode.DataNode	RECEIVED SIGNAL 15: SIGTERM
1	2015-08-26 15:04:30,360	ERROR	org.apache.hadoop.hdfs	server	datanode.VolumeScanner	nner: VolumeScanner(/opt/hdfs/data, DS-e4e85a3...
2	2015-08-25 19:42:05,762	ERROR	org.apache.hadoop.hdfs	server	datanode.DataNode	RECEIVED SIGNAL 15: SIGTERM
3	2015-08-25 19:04:08,209	ERROR	org.apache.hadoop.hdfs	server	datanode.DataNode	RECEIVED SIGNAL 15: SIGTERM
4	2015-08-21 11:16:44,183	ERROR	org.apache.hadoop.hdfs	server	datanode.DataNode	RECEIVED SIGNAL 15: SIGTERM

Hình 3.1: Dữ liệu log đã Import

Dữ liệu log được lấy từ dự án Loghub được chia sẻ ở trang

https://zenodo.org/record/3227177#.Yachr3tBwh0

Sau khi lọc bỏ các dữ liệu log có thuộc tính Severity là loại INFO, đây là những bản tin thông báo của hệ thống không chứa những nguy cơ nguy hiểm. Sẽ còn lại 3 dạng thuộc tính Severity là: WARN, ERROR và FATAL. Đây là các đặc trưng chính để phân cụm và xác định tính chất của cụm dữ liệu.



Hình 3.2: Thống kê thuộc tính Severity

Dữ liệu bao gồm các thuộc tính có số lượng log như sau:

- WARN = 90746 log
- ERROR=3804 log
- FATAL=888 log

Từ dữ liệu dataset đâu vào, ta sẽ có 6 loại thuộc tính quán trọng với quá trình phân cụm như sau: Severity, Component, Class, Category, Description

Không tính thuộc tính Datetime chỉ ngày tháng năm xuất ra log thì các thuộc tính còn lại có kiểu dữ liệu là liệt kê ngoài trừ thuộc tính Description là dạng chuỗi. [19]

Thuộc tính Severity: Mức độ cảnh báo của bản tin log

Thuộc tính Component: Chứa thành phần nơi xuất ra các bản tin log

Thuộc tính Class: Chứa thông tin hệ thống nơi xuất ra các bản tin log

Thuộc tính Description: Đoạn mô tả, báo cáo lỗi của mỗi log

Thuộc tính Description này sẽ được lọc ra nhờ phương pháp TF x IDFđể trích xuất ra từ quan trọng nhất trong phần mô tả lỗi của một log. Từ khóa này thường là từ xuất hiện nhiều trong mô tả của log này nhưng lại ít xuất hiện trong mô tả của các log khác. Các từ khóa này là các từ có giá trị cao trong phần mô tả của một lỗi và đã được lọc bỏ các từ thông thường trong các đoạn mô tả. Sau khi dùng kỹ thuật TF x IDFdữ liệu log sẽ có thêm một thuộc tính mới là Keyword biểu diễn từ khóa quan trọng của mỗi một log trong dataset.

95438 rows

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	2
error	0.437623	0.455408	0.455408	0.425867	0.455408	0.455408	0.455408	0.449555	0.18624	0.449555	0.121278	0.426602	0.425867	0.226493	0.425867	0.226493	0.425867	0.425867	0.18624	0.18624	0.425867	0.455408	0.455408	0.42586
operationsrc	0.431340	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.00000	0.000000	0.000000	0.000000	0.00000
read_block	0.431340	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.00000	0.000000	0.000000	0.000000	0.00000
dataxceiver	0.302220	0.314502	0.314502	0.000000	0.314502	0.314502	0.314502	0.310460	0.00000	0.310460	0.000000	0.294608	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.00000	0.000000	0.314502	0.314502	0.00000
dst	0.302220	0.314502	0.314502	0.000000	0.314502	0.314502	0.314502	0.310460	0.00000	0.310460	0.000000	0.294608	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.00000	0.000000	0.314502	0.314502	0.00000

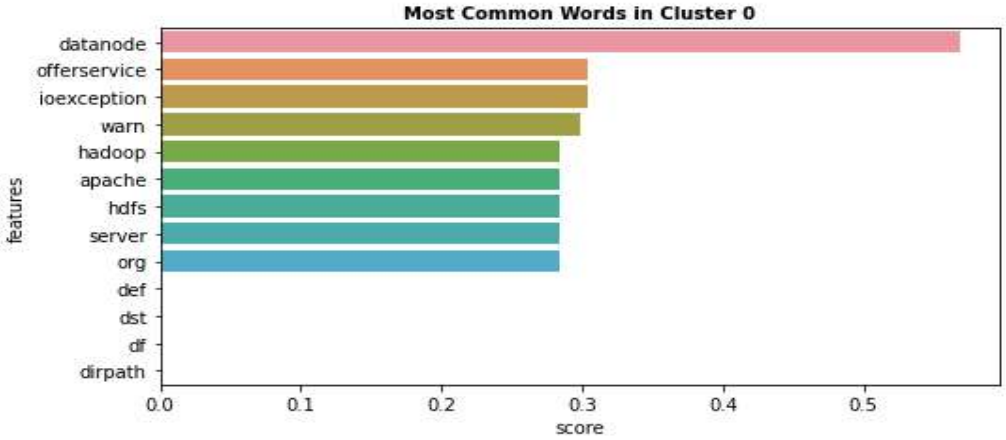
5 rows × 95438 columns

Hình 3.3: Giá trị TF x IDF sau khi tính toán

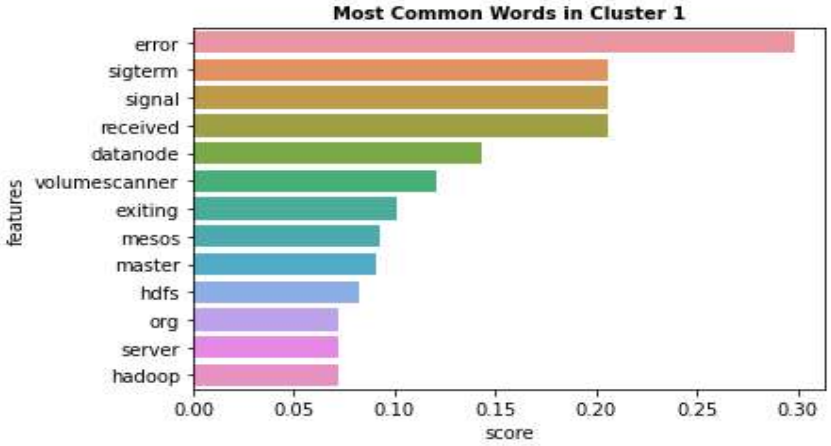
3.3.3 Áp dụng thuật toán K-means phân cụm dữ liệu log

Dựa theo dữ liệu log đã rút trích đặc trưng ta tiến hành phân cụm dữ liệu [20], ta sẽ có 6 loại thuộc tính quán trọng với quá trình phân cụm như sau: Severity, Component, Class, Category, Keyword

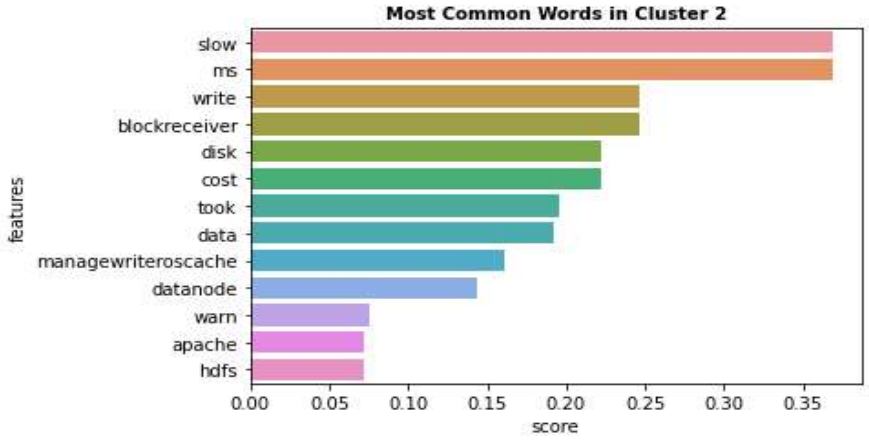
Tiến hành lượng tử hóa các thuộc tính biểu diễn thành dạng vector để đưa vào K-means thực hiện quá trình xử lý, luận văn sẽ thực thi kỹ thuật K-means chia dữ liệu thành 3 cụm.



Hình 3.4: Kết quả phân cụm thứ 1

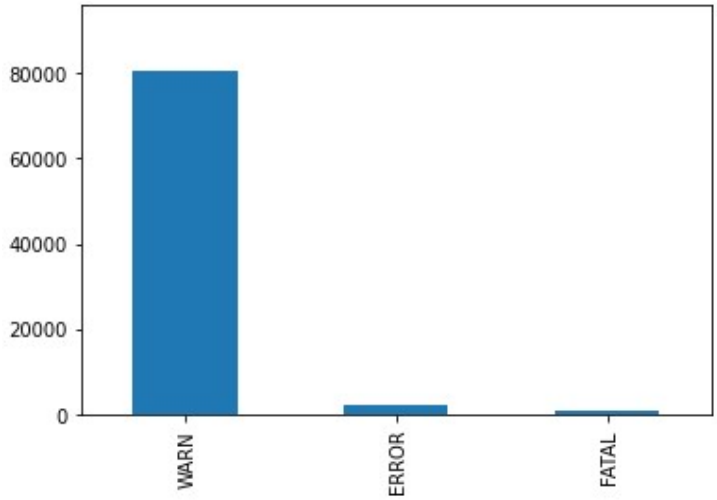


Hình 3.5: Kết quả phân cụm thứ 2

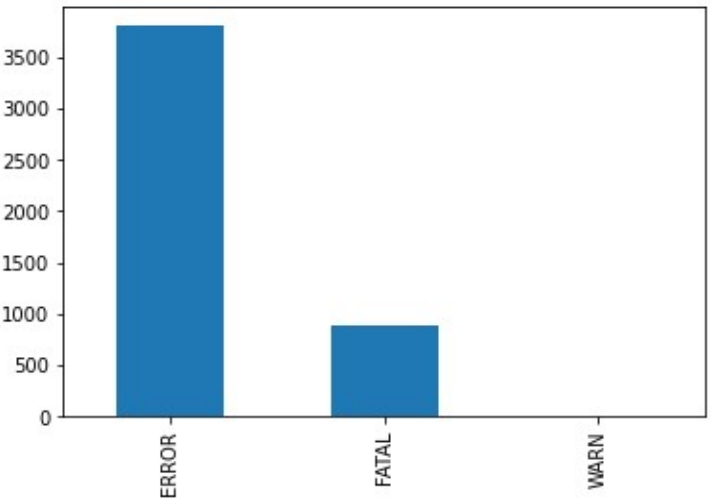


Hình 3.6: Kết quả phân cụm thứ 3

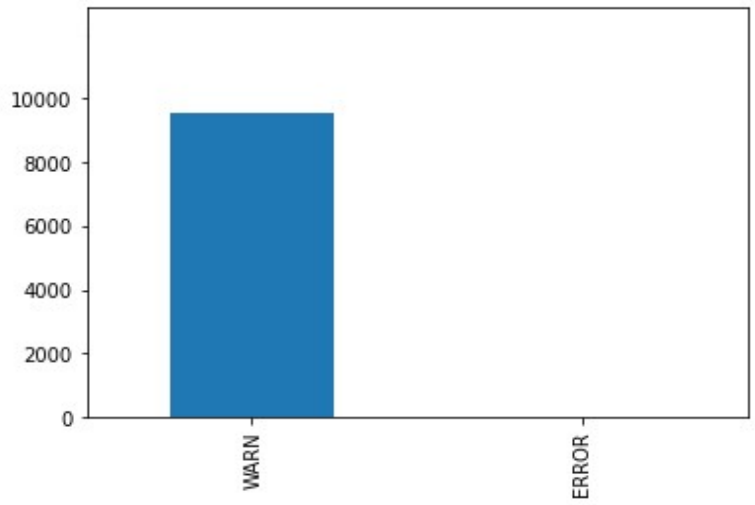
Dựa vào các mô hình hóa trong mỗi cụm đã phân ra được, có thể phân tích ra được các cụm nào là dữ liệu log bất thường, dữ liệu log bình thường và log chưa xác định [21] dựa vào các từ khóa và số lượng log thuộc severity nào.



Hình 3.7: Số lượng log của kết quả phân cụm 1



Hình 3.8: Số lượng log của kết quả phân cụm 2



Hình 3.9: Số lượng log của kết quả phân cụm 2

Theo kết quả phân cụm dựa và số lượng log ở mỗi cụm, ta thấy:

- Số lượng log ở cụm thứ 1 phần lớn là WARN, một số ít là ERROR và FATAL
- Số lượng log ở cụm thứ 2 phần lớn là ERROR, tiếp theo là FATAL và số ít là WARN
- Số lượng log ở cụm thứ 2 phần lớn là WARN

Từ đó kết quả phân cụm như sau

- Cụm 1: Cụm dữ liệu log chưa xác định
- Cụm 2: Dữ liệu log bất thường
- Cụm 3: Dữ liệu log bình thường

3.4 Kết luận chương

Việc ứng dụng các kỹ thuật học máy Machine Learning vào trong phân tích dữ liệu log để phát hiện các cảnh báo bất thường là một trong những xu thế hiện đang thu hút nhiều học giả. Việc kết hợp các trí tuệ nhân tạo, các kỹ thuật học máy với các công cụ giám sát mạng hiện có cũng là một trong những hướng phát triển tốt và giúp nhanh chóng phát hiện và xử lý lỗi. Thực nghiệm trong luận văn này chỉ thể hiện

được một phần nào đó việc phân tích log có thể đạt được. Tuy không thể phân tích và đánh giá hoàn hảo được nhưng kết quả có co sở và có khả quan trong việc kết hợp xử lý Log và các thuật toán thông minh là những bước đi đầu tiên cho những nghiên cứu mở rộng tiếp theo về phân tích log để có thể tiến xa hơn là dự báo trước sự cố.

CHƯƠNG 4 - KẾT LUẬN

4.1 Giới thiệu chương

Đánh giá kết quả đã thực hiện được của luận văn và đưa ra kết luận

4.2 Mô tả môi trường thực nghiệm thuật toán

Dựa vào mô hình dữ liệu được đề xuất tiến hành xây dựng thuật toán phân tích dữ liệu log để phát hiện cảnh báo bất thường, sử dụng ngôn ngữ lập trình Python với bộ thư viện Scikit-learn (Sklearn) là thư viện chuyên sâu nhất dành cho các thuật toán học máy để lọc dữ liệu và tối ưu dữ liệu đầu vào

Bước 1: Tiếp nhận giá trị dữ liệu input

Bước 2: Phân tích các input để lọc các đặc trưng dư thừa, lặp lại, các đặc trưng phổ biến không có nhiều thông tin hay giá trị cần thiết, sau đó rút trích các đặc trưng của các input,

Bước 3: Dựa vào các thuộc tính trên chúng ta sử dụng machine learning, kỹ thuật học máy K-means để phân cụm các dữ liệu đầu vào.

Bước 4: Dựa vào kết quả phân cụm ta sẽ có được dữ liệu cảnh báo bất thường tiềm tàng có nguy cơ gây nguy hiểm.

4.3 Kết quả thực nghiệm của thuật toán

- Kết quả quá trình lọc dữ liệu
- Kết quả quá trình phân cụm
- Đánh giá hiệu quả của quá trình phân cụm dữ liệu log

4.4 Kết quả về mặt lý thuyết

Tìm hiểu và nắm được các nguyên lý cơ bản của các kỹ thuật học máy, lý thuyết về trí tuệ nhân tạo và định nghĩa các phương pháp khai phá dữ liệu

Tìm hiểu về kỹ thuật phân cụm K-means và ứng dụng vào để phân tích dữ liệu log

Hiểu được mô hình dữ liệu log, biết cách lọc dữ liệu đầu vào và trích xuất thuộc tính

4.5 Kết quả về mặt thực tiễn

Luận văn đã đưa ra giải pháp phân loại log, phân tích log dựa các kỹ thuật IF-IDF khai phá văn bản, giúp trích xuất được các thông tin quan trọng từ miêu tả lỗi của log

Luận văn đã đã đề xuất được thuật toán giúp phát hiện cảnh báo bất thường dựa vào phân tích dữ liệu log, thuật toán phân cụm K-means xác định được các loại dữ liệu log bất thường cần chú ý kiểm tra, các dữ liệu log bình thường có thể bỏ qua, các dữ liệu log chưa rõ cần xem xét tiếp.

Xây dựng được mô hình dữ liệu lỗi log giúp phát hiện cảnh báo bất thường bao gồm các thuộc tính cần thiết, phân tích và đánh giá hiệu quả của mô hình

Mô hình trên có thể hỗ trợ người dùng trên các hệ thống giám sát cảnh báo, khi người dùng phân vùng được bản tin log thuôc nhóm nguy cơ gây nguy hiểm cho hệ thống sẽ có thể nhận định sớm từ ban đầu mức độ ảnh hưởng của lỗi đó đến hệ thống mạng.

4.6 Hạn chế

Kết quả có được vẫn còn phải cải tiến thêm, dữ liệu phân tích log cần được phân loại với số lượng lớn hơn và mở rộng đối tượng hệ thống mạng áp dụng.

Dữ liệu log cần phải cải thiện hơn về độ chính xác, loại bỏ các thông tin không cần thiết, gây sai lệch trong quá trình đánh giá hiệu quả mô hình

Các trường hợp lỗi nghiêm trọng bị phân loại sai thành không nghiêm trọng còn nhiều, gây nhầm lẫn cho người sử dụng nếu áp dụng với thực tế.

Thuật toán phân cụm K-means trong luận văn chưa phải là tối ưu nhất, chưa phân cụm hết được tất cả các bản tin log và sẽ có những bản tin log không thể xác định được bằng phương pháp này

4.7 Hướng phát triển

Cải thiện dữ liệu đầu vào, lọc ra và xây dựng được mô hình dữ liệu log có độ tin cậy cao và chính xác hơn.

Tiếp tục cải tiến rút trích các đặc trưng để phù hợp hơn cho quá trình phân tích dữ liệu, cải thiện độ chính xác trong việc phân cụm các dữ liệu log

Nghiên cứu sâu các thuộc tính của log, trích xuất ra được các thuộc tính mới để xây dựng được mô hình dữ liệu log hiệu quả hơn

. Nghiên cứu và phát triển hơn các thuật toán để cải thiện hiệu quả phân tích dữ liệu log. Phân cụm các vùng dữ liệu log chính xác hơn nữa.

Tiến hành áp dụng cho cho hệ thống mạng lưới mạng băng rộng của Viễn thông Tây Ninh. Phân tích ra các log có thể gây lỗi nghiêm trọng dựa trên cơ sở dữ liệu là các log hệ thống xuất ra và mức độ nghiêm trọng của một lỗi thực tế đã từng xảy ra trước đây. Từ phân tích log đó phát hiện ra tín hiệu bất thường trên hệ thống mạng băng rộng thuộc Viễn thông Tây Ninh đưa ra cảnh báo sớm các sự cố có thể ảnh hưởng nghiêm trọng đến hệ thống để có biện pháp ngăn chặn kịp thời. Góp phần giảm thiểu rủi ro cho hệ thống mạng và truyền thông.

.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] https://licensesoft.vn/prtg-network-monitor.htm, truy cập ngày 24/06/2021

[2] https://aicenter-itp.edu.vn/tin-tuc/buoc-chuyen-lon-cua-the-gioi-trong-cuoc-cach-mang-4-0-86.html, truy cập ngày 25/06/2021

[3] https://www.thousandeyes.com/learning/techtorials/SNMP-simple-network-management-protocol, truy cập ngày 26/06/2021

[4] https://docs.oracle.com/cd/E13203_01/tuxedo/tux81/SNMPmref/1tmib.htm truy cập ngày 27/06/2021

[5] https://oidref.com/, truy cập ngày 28/06/2021

[6] https://www.paessler.com/it-explained/syslog, truy cập ngày 28/06/2021

[7] RFC 3164 - The BSD syslog Protocol https://www.hjp.at/(de)/doc/rfc/rfc3164.html, 2001

[8] Buckley MF, Siewiorek DP (1995) VAX/VMS Event monitoring and analysis. In: Proceedings 25th international symposium on fault-tolerant computing (FTCS'95). IEEE computer society, pp 414–423

[9] Bishop CM (1995) Neural networks for pattern recognition.Oxford university press, New York

[10] Sonali. B. Maind, Priyanka Wankar (2014),"Research Paper on Basic of Artificial Neural Network"

[11] L. Breiman (2001), "Random Forests", Machine Learning, vol. 45, no. 1, pp. 5–32

[12] M. R. Lyu, J. Zhu, P. He, S. He, and J. Liu"The logpai project," http://www.logpai.com/, 2019.

[13] Shilin He, Jieming Zhu, Pinjia He, Michael R. Lyu(2020),"Loghub: A Large Collection of System Log Datasets towards Automated Log Analytics

[14] Sinh Van Nguyen, Ha Manh Tran (2020), "An automated fault detection system for communication networks and distributed systems", Springer Science+Business Media, LLC, part of Springer Nature, Available:

https://doi.org/10.1007/s10489-020-02026-2 Communications:Article scheduled for publication in Vol. 11, No. 3–4

[15] Ha Manh Tran, Tuan Anh Nguyen, Son Thanh Le, Giang Vu Truong Huynh, Tuan Bao Lam (2021), "Two-Phase Defect Detection Using Clustering and Classification Methods", REV Journal on Electronics and Communications: Article scheduled for publication in Vol. 11, No. 3–4

[16] Juan Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries,"https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.1424&rep=rep1&type=pdf.

[17] https://scikit-learn.org/, truy cập ngày 19/09/2021

[18] https://machinelearningcoban.com/, truy cập ngày 02/11/2021

[19] W. Xu, L. Huang, A. Fox, D. Patterson, and M. I. Jordan,"Detecting large-scale system problems by mining console log," in Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles, ser. SOSP '09. New York, NY, USA: ACM, 2009, pp. 117–132. [Online]. Available: https://doi.org/10.1145/1629575.1629587

[20] S. He, J. Zhu, P. He, and M. R. Lyu, "Experience report: System log analysis for anomaly detection," in Proceedings of the 27th IEEE International Symposium on Software Reliability Engineering (ISSRE'16). IEEE, 2016, pp. 207–218. [Online]. Available: https://doi.org/10.1109/ISSRE.2016.21.

[21] S. He, Q. Lin, J. G. Lou, H. Zhang, M. R. Lyu, and D. Zhang, "Identifying impactful service system problems via log analysis," in Proceedings of the 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ser. ESEC/FSE'18. New York, NY, USA: ACM, 2018, pp. 60–70. [Online]. Available: https://doi.org/10.1145/3236024.3236083