HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

PHẠM VĂN HUẤN

NGHIÊN CỨU PHƯƠNG PHÁP PHÁT HIỆN BẤT THƯỜNG DỰA TRÊN DỮ LIỆU LOG HỆ THỐNG

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

TÓM TẮT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – 2024

Đề án tốt nghiệp được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: TS. NGUYỄN HUY TRUNG

Phản biện 1:

…….

Phản biện 2:

…….

Đề án tốt nghiệp sẽ được bảo vệ trước Hội đồng chấm đề án tốt nghiệp thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: ..... giờ ..... ngày ..... tháng .... năm .....

Có thể tìm hiểu đề án tốt nghiệp tại:

- Thư viện của Học viện Công nghệ Bưu chính Viễn thông.

LỜI MỞ ĐẦU

1. Lý do chọn của đề tài

Trong bối cảnh ngày nay, cùng với sự phát triển mạnh mẽ của công nghệ thông tin, quy mô và độ phức tạp của hệ thống thông tin ngày càng gia tăng, kéo theo những thách thức lớn về an toàn và bảo mật. Việc đảm bảo an toàn và bảo mật cho hệ thống thông tin là một yêu cầu cấp bách nhằm bảo vệ dữ liệu, ngăn chặn các hành vi xâm nhập và tấn công mạng. Đặc biệt, việc phát hiện sớm các sự kiện bất thường trong hệ thống thông tin đóng vai trò then chốt trong việc ngăn chặn và ứng phó hiệu quả với các mối đe dọa an ninh.

Trong nghiên cứu này, học viên tập trung vào việc nghiên cứu về phương pháp phát hiện bất thường dựa trên dữ liệu log hệ thống. Dữ liệu log hệ thống là nguồn thông tin hữu ích để giám sát và phát hiện bất thường, nó ghi lại trạng thái hệ thống và các sự kiện quan trọng tại các điểm then chốt khác nhau để giúp gỡ lỗi các vấn đề về hiệu suất và sự cố, đồng thời thực hiện phân tích nguyên nhân gốc rễ. Bên cạnh đó, log hệ thống cũng sẽ giúp nhận biết những biểu hiện tiềm ẩn của sự tấn công hoặc lỗi hệ thống.

Tuy nhiên, với sự phát triển mạnh mẽ của hệ thống thông tin hiện nay, đặc biệt trong các hệ thống thông tin quan trọng và cỡ lớn, lượng dữ liệu log hệ thống được tạo ra ngày càng nhiều và đa dạng. Do đó, việc phân tích thủ công là điều không khả thi, vì vậy cần phải có các phương pháp phân tích tự động, linh hoạt và chính xác, có khả năng phát hiện những biểu hiện bất bình thường trong dữ liệu log và đưa ra cảnh báo sớm, giúp quản trị hệ thống nhanh chóng ứng phó và ngăn chặn các rủi ro an ninh.

Dựa vào các phân tích trên, học viên tin tưởng đề án "***Nghiên cứu phương pháp phát hiện bất thường dựa trên dữ liệu log hệ thống"*** sẽ có ý nghĩa khoa học

và tính cấp thiết cao, từ đó góp phần nâng cao hiệu quả và an toàn cho các hệ thống thông tin.

2. Tổng quan về vấn đề nghiên cứu:

Có nhiều phương pháp khác nhau để phát hiện bất thường, một số phương pháp phổ biến bao gồm:

- Phân tích dữ liệu log: Phương pháp này phân tích dữ liệu log để tìm ra các mẫu bất thường. Các mẫu bất thường có thể bao gồm các giá trị dữ liệu nằm ngoài phạm vi bình thường, các xu hướng dữ liệu bất thường hoặc các sự kiện không khớp với mô hình hành vi bình thường.
- Phân tích hành vi: Phương pháp này phân tích hành vi của người dùng hoặc hệ thống để tìm ra các hành vi bất thường. Các hành vi bất thường có thể bao gồm các truy cập hệ thống bất thường, việc sử dụng tài nguyên quá mức hoặc các thay đổi cấu hình bất thường.
- Phân tích mạng: Phương pháp này phân tích lưu lượng mạng để tìm ra các hoạt động bất thường. Các hoạt động bất thường có thể bao gồm các cuộc tấn công mạng, các cuộc truy cập từ xa không xác định hoặc các hoạt động sử dụng mạng bất thường.

Các nghiên cứu về phương pháp phát hiện bất thường dựa trên dữ liệu log đang tập trung vào các hướng sau:

- Sử dụng các kỹ thuật học máy: Các kỹ thuật học máy đang được sử dụng để phát triển các phương pháp phát hiện bất thường hiệu quả hơn. Các kỹ thuật học máy có thể học cách phân biệt dữ liệu bình thường và dữ liệu bất thường ngay cả khi dữ liệu log thay đổi theo thời gian.
- Sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên: Các kỹ thuật xử lý ngôn ngữ tự nhiên đang được sử dụng để phân tích dữ liệu log. Các kỹ thuật xử lý ngôn ngữ tự nhiên có thể giúp phát hiện các mối đe dọa bảo mật mới dựa trên các thông tin chi tiết trong dữ liệu log.

Tóm lại, phát hiện bất thường là một vấn đề nghiên cứu quan trọng trong lĩnh vực bảo mật thông tin. Nghiên cứu phương pháp phát hiện bất thường dựa trên dữ liệu log hệ thống đang được phát triển tích cực với nhiều hướng tiếp cận mới.

3. Mục đích nghiên cứu

Mục tiêu chính của đề án là đưa ra mô hình về phát hiện bất thường dựa trên dữ liệu log hệ thống, một số mục tiêu cụ thể như sau:

- Tiền xử lý dữ liệu
- Xây dựng mô hình phát hiện bất thường dựa trên dữ liệu log hệ thống (system log)
- Thực nghiệm và đánh giá kết quả

4. Đối tượng và phạm vi nghiên cứu

- Đối tượng: Dữ liệu log hệ thống (system log)

- Phạm vi nghiên cứu: Tập trung vào bài toán phát hiện bất thường dựa trên dữ liệu log hệ thống.

- Thời gian thực hiện: 15/12/2023 – 29/04/2024

5. Phương pháp nghiên cứu

- Nghiên cứu lý thuyết

Tiến hành nghiên cứu, khảo sát, tổng hợp, đánh giá các công trình nghiên cứu liên quan ở trong và ngoài nước để phân tích những vấn đề chưa giải quyết, những vấn đề cần tiếp tục nghiên cứu theo hướng của đề tài. Các công trình nghiên cứu được tìm kiếm tại các kho dữ liệu trực tuyến như:

- Google Scholar (https://scholar.google.com/)
- IEEE Xplore (https://ieeexplore.ieee.org/)

- Nghiên cứu thực nghiệm

Thu thập dữ liệu từ các kho lưu trữ công khai. Chia thành dữ liệu các tập huấn luyện và kiểm thử, sử dụng kỹ thuật kiểm thử chéo (cross-validation),… huấn luyện và thử nghiệm.

CHƯƠNG 1: TỔNG QUAN PHÁT HIỆN BẤT THƯỜNG VÀ DỮ LIỆU LOG HỆ THỐNG

1.1 Tổng quan về phát hiện bất thường

1.1.1 Định nghĩa về phát hiện bất thường?

Phát hiện bất thường là kỹ thuật xác định các hoạt động hoặc sự kiện khác biệt so với hành vi bình thường trong hệ thống. Mục tiêu của phát hiện bất thường là phát hiện sớm các mối đe dọa tiềm ẩn và ngăn chặn các vi phạm an ninh trước khi chúng gây ra thiệt hại.

1.1.2 Vai trò và ý nghĩa trong bảo mật hệ thống

1.1.3 Thách thức và rủi ro

1.2 Dữ liệu log hệ thống

1.2.1 Định nghĩa về dữ liệu log hệ thống

Dữ liệu log hệ thống (hay còn gọi là log system, log file) là các tập tin văn bản ghi lại các hoạt động, sự kiện và trạng thái của hệ thống thông tin trong một khoảng thời gian nhất định. Dữ liệu log được tạo ra bởi các thành phần khác nhau của hệ thống, bao gồm hệ điều hành, ứng dụng, dịch vụ và phần mềm bảo mật. Dữ liệu log thường được lưu trữ trong các tệp văn bản, nhưng cũng có thể được lưu trữ trong cơ sở dữ liệu hoặc các định dạng khác. Dữ liệu log hệ thống là loại dữ liệu quan trọng trong bảo mật và giám sát, cung cấp lịch sử đầy đủ của các sự kiện theo thời gian. Ngoài dữ liệu log của hệ điều hành, log còn được sử dụng trong các ứng dụng, trình duyệt web, phần cứng và thậm chí cả email.

1.2.2 Cấu trúc và đặc điểm

1.3 Liên kết phát hiện bất thường và dữ liệu log hệ thống

Các hệ thống và phần mềm ngày nay ngày càng trở nên lớn mạnh và lượng thông tin được lưu trữ bởi các hệ thống này là rất lớn. Điều này dẫn đến nhu cầu đảm bảo tính bảo mật, tính sẵn sàng và độ tin cậy của chúng. Một trong những thách thức của việc này là nhanh chóng xác định được lỗi trong hệ thống. Do đó, với việc theo

dõi log hiệu quả, hệ thống có thể được phòng ngừa hoặc nhanh chóng phát hiện và ứng phó, sửa chữa các lỗi trong hệ thống. Có phương pháp theo dõi log tối ưu giúp giảm thời gian chết và giảm thiểu nguy cơ mất dữ liệu. Chính vì vậy đối với những hệ thống lớn, log thường được gửi đến một máy chủ an toàn hoạt động như một điểm thu thập chung trước khi được quản trị viên hệ thống xử lý thêm.

Dữ liệu log hầu hết là không có cấu trúc, điều này làm cho việc phát hiện bất thường dựa trên log trở nên khó khăn. Yêu cầu cơ bản để phát hiện bất thường chính xác là có thể cấu trúc hiệu quả các log template và phân loại chúng thành các nhóm.

Tuy nhiên, với sự phát triển của ứng dụng AI trong các bài toán thực tế, một số phương pháp sử dụng các thuật toán để phục vụ mục đích phát hiện các bất thường trên hệ thống. Các sự kiện nhật ký này sẽ được ánh xạ tới một không gian vectơ, từ đó các sự kiện nhật ký liên tiếp được chuyển đổi thành vectơ đầu vào cho thuật toán ML. Tuy nhiên, các phương pháp này thường không ổn định trên các tập dữ liệu khác nhau. Bởi vì các sự kiện nhật ký này được tạo bởi các nhà phát triển nên mỗi dòng nhật ký đều có ý nghĩa ngữ nghĩa riêng. Việc ánh xạ như trên gây mất thông tin quan trọng.

Gần đây, các phương pháp học sâu sử dụng các mô hình nổi tiếng như CNN, LSTM, máy biến áp đã được áp dụng để phát hiện các điểm bất thường trong thiết bị. Các mô hình này đã đạt được kết quả tốt trên các tập dữ liệu khác nhau. Các phương pháp này lấy các sự kiện nhật ký tuần tự làm đầu vào để xác định các điểm bất thường bằng cách phát hiện các hành vi vi phạm các mẫu tuần tự. Tuy nhiên, các mô hình này có nhược điểm là chưa khai thác được thông tin về mối liên kết giữa các sự kiện log, điều này rất quan trọng. Ngoài ra, những mô hình này không thể bao quát hết tất cả các sự kiện trong một khoảng thời gian dài. Nếu các cuộc tấn công vào thiết bị xảy ra với tần suất thấp, thì các hoạt động tấn công sẽ cách xa nhau, khiến các phương pháp này không thể nhận ra chúng.

1.4 Kết chương

Chương 1 trình bày tổng quan về phát hiện bất thường trong hệ thống thông tin, cấu trúc và đặc điểm dữ liệu log hệ thống. Từ đó, nội dung chương chỉ ra những vấn đề còn tồn tại, thách thức và đề xuất phương án giải quyết các vấn đề này.

Chương 2 sẽ trình bày tổng quan về học máy, học sâu và một số phương pháp phát hiện bất thường hiện nay đồng thời mô tả chi tiết phương án giải quyết các vấn đề này nhằm đưa ra mô hình đề xuất phù hợp.

CHƯƠNG 2: CÁC PHƯƠNG PHÁP PHÁT HIỆN BẤT THƯỜNG

2.1 Tổng quan về học máy và học sâu

2.1.1 Tổng quan về học máy

2.1.2 Tổng quan về học sâu

2.1.3 Một số phương pháp học sâu

- **Mạng nơ ron sâu (Deep Neural Network-DNN)**
- **Mạng nơ ron tích chập (Convolutional Neural Network)**
- **Mạng nơ ron đồ thị (Graph Neural Network - GNN)**

2.2 Các nghiên cứu về phát hiện bất thường.

Trong những năm qua, đã có nhiều nghiên cứu đề xuất phát hiện bất thường từ dữ liệu log hệ thống. Các mô hình dựa trên học máy dựa vào thống kê log để phát hiện các bất thường của hệ thống. Đã có nhiều phương pháp học máy phổ biến áp dụng vào phát hiện bất thường dựa trên log. Tuy nhiên, các mô hình dựa trên học máy này đều có nhiều hạn chế, các tính năng không linh hoạt, kém hiệu quả và khả năng thích ứng yếu.

Để giải quyết những vấn đề này, học sâu đã được áp dụng và đạt được những kết quả đầy hứa hẹn.

Bảng 2.1: Mô tả về ưu và nhược điểm của các phương pháp

	Deeplog	**LogRobust**	**NeuralLog**	**LogGD**
Ưu điểm	Hoạt động tốt với hệ thống có dữ liệu log ổn định	Hoạt động tốt với hệ thống có dữ liệu log ổn định và không ổn định	Hoạt động tốt với hệ thống có dữ liệu log ổn định và không ổn định	Hoạt động tốt với hệ thống có dữ liệu log ổn định và không ổn định
Nhược điểm	Hoạt động không tốt với hệ thống có dữ liệu log không ổn định - Cần dữ liệu lớn log để huấn luyện	Có độ chính xác chưa được cao so với phương pháp khác vì bỏ qua thông tin về mối liên kết giữa các sự kiện log	Có độ chính xác chưa được cao so với phương pháp khác vì bỏ qua thông tin về mối liên kết giữa các sự kiện log	Có độ chính xác chưa được cao so với phương pháp khác vì việc tính toán chuyển đổi từ chuỗi log sang biểu đồ vẫn còn hạn chế khi bỏ qua chuỗi sự kiện log

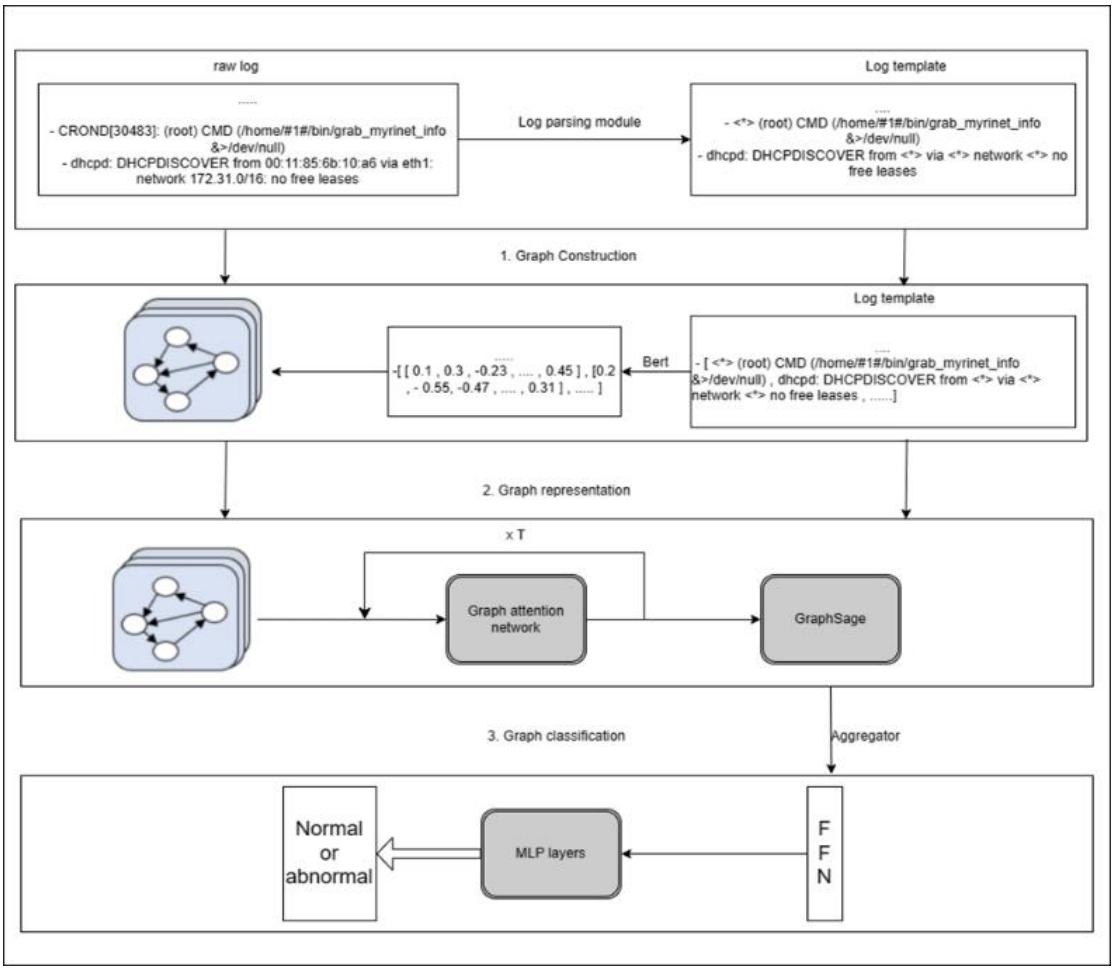
2.3 Đề xuất mô hình phát hiện bất thường

2.3.1 Giới thiệu mô hình

Để giải quyết vấn đề nêu trên, học viên đề xuất phương pháp phát hiện bất thường dựa trên đồ thị. Phương pháp này sẽ chuyển đổi dữ liệu từ dạng chuỗi sang dạng biểu đồ. Sử dụng thông tin về tính năng nút và thông tin thu được từ biểu đồ để phục vụ cho việc phân loại các điểm bất thường dựa trên mô hình mạng Graph Attention Networks (GAT) [7]. Phương pháp phát hiện bất thường dựa trên biểu đồ của học viên bao gồm ba bước chính:

- B1: Xây dựng biểu đồ: Là bước tính toán chuyển đổi dữ liệu thành dạng biểu đồ

- B2: Biểu diễn biểu đồ: Là bước tính toán thông tin của biểu đồ
- B3: Phân loại biểu đồ: Là bước phân loại nhãn của dữ liệu.



Hình 2.1: Các bước xử lý trong giải pháp

2.3.2 Các bước xử lý

Trong hệ thống máy tính, các mục nhật ký thường được phát triển và triển khai bởi các lập trình viên hệ thống. Các mục nhật ký này thường chứa cả thông tin ngữ nghĩa bằng ngôn ngữ tự nhiên và thông tin về các quy trình được ghi lại, chẳng hạn như ID tiến trình. Do đó, các công cụ phân tích dữ liệu nhật ký được thiết kế để trích xuất từng phần thông tin riêng lẻ từ nhật ký. Trong thử nghiệm này, sẽ sử dụng Drain [8], một công cụ phổ biến được nhiều mô hình sử dụng để phân tích dữ liệu nhật ký.

Quá trình xử lý của Drain gồm:

- Phân tích cú pháp: Chuyển đổi dữ liệu nhật ký sang định dạng có cấu trúc.
- Loại bỏ nhiễu: Loại bỏ các phần dữ liệu không liên quan hoặc không chính xác.
- Chuyển đổi kiểu: Chuyển đổi các kiểu dữ liệu sang định dạng phù hợp cho phân tích.
- Bổ sung trường: Thêm các trường mới vào dữ liệu nhật ký để hỗ trợ phân tích.

- **Xây dựng đồ thị**

Đầu vào của bài toán là chuỗi log sau khi phân tích và nhóm dữ liệu (S) theo từng phần hoặc cửa sổ trượt như đã trình bày trên. Với thông tin đầu vào này, sẽ xây dựng một biểu đồ có hướng để biểu diễn chúng. Cho $G = (V, E)$ là một đồ thị, trong đó V là tập hợp các nút trong đồ thị và E là tập hợp các cạnh có hướng. X_v là tập hợp các đặc điểm nút của đồ thị và F_v là tập hợp các đặc điểm cạnh của đồ thị. Mỗi nút trong biểu đồ G tương ứng với một sự kiện nhật ký trong chuỗi nhật ký. Tính năng nút sẽ được xây dựng dưới dạng thông tin ngữ nghĩa của các sự kiện nhật ký. Ở đây, việc thực hiện tính toán thông qua các mô hình NLP hiện có để phân tích các sự kiện nhật ký. Với xây dựng tập cạnh, mỗi e_{ij} tương ứng với sự xuất hiện của sự kiện V_i, theo sau là sự kiện V_j trong chuỗi nhật ký. Mỗi đặc điểm cạnh biểu thị thông tin cho cạnh có kích thước bằng độ dài của chuỗi nhật ký. Kích thước của tính năng cạnh bằng với độ dài của chuỗi nhật ký bởi vì sẽ đảm bảo rằng tất cả thông tin về thời gian xuất hiện của sự kiện nhật ký trong chuỗi nhật ký không bị mất trong quá trình xây dựng biểu đồ.

- **Biểu diễn đồ thị**

Biểu diễn đồ thị là một quá trình quan trọng trong các nhiệm vụ phân loại đồ thị và các nhiệm vụ liên quan khác. Để mô hình có thể phân biệt tốt giữa các sự kiện bình thường và bất thường, cần phải thiết kế cẩn thận các thuộc tính sẽ được

sử dụng. Mỗi nút trong biểu đồ đại diện cho một sự kiện duy nhất trong chuỗi nhật ký thu được từ quá trình phân tích và xử lý. Mỗi sự kiện trong nhật ký đều do nhà phát triển xác định nên chúng đều có một ý nghĩa ngôn ngữ nhất định. Theo nhiều nghiên cứu trước đây, thông tin ngữ nghĩa từ các sự kiện nhật ký có thể rất hiệu quả cho nhiều mục đích khác nhau. Có nhiều mô hình NLP đạt hiệu suất cao trong trích xuất ngữ nghĩa như Word2vec [9], BERT [10], Fastext [11], v.v. Trong thử nghiệm này, sẽ sử dụng mô hình BERT(Bidirectional Encoder Representations from Transformers) là một mô hình ngôn ngữ mạnh mẽ được phát triển bởi Google AI, có khả năng hiểu và biểu diễn văn bản ngôn ngữ tự nhiên một cách hiệu quả, với kích thước nhúng là 128 để biểu diễn log sự kiện. Các từ trong mỗi sự kiện nhật ký sẽ được trích xuất các đặc điểm thông qua BERT và các vectơ được mã hóa sẽ được tổng hợp để thu được vectơ biểu diễn thông tin 128 chiều cho mỗi sự kiện nhật ký.

Dưới đây là các bước chính trong hoạt động của mô hình:

- Sử dụng kiến trúc mạng nơ-ron Transformer để xây dựng mô hình BERT.
- Áp dụng hai nhiệm vụ học tập chính:
    - Dự đoán từ tiếp theo (Masked Language Modeling - MLM): Ẩn một số từ trong đoạn văn bản và yêu cầu mô hình dự đoán những từ bị ẩn dựa trên ngữ cảnh.
    - Dự đoán câu tiếp theo (Next Sentence Prediction - NSP): Cung cấp cho mô hình hai câu văn bản và yêu cầu mô hình xác định xem hai câu đó có liên quan với nhau hay không.
- Lặp lại quá trình học tập trên tập dữ liệu khổng lồ để tinh chỉnh mô hình BERT, giúp nó học cách hiểu các mối quan hệ ngữ nghĩa giữa các từ và câu.

Trong nghiên cứu đã phân tích ở trên, việc sử dụng chuỗi sự kiện theo trình tự nhật ký đã đạt được kết quả rất tốt. Trình tự của các sự kiện nhật ký này phản ánh

mối tương quan giữa các mối quan hệ giữa các sự kiện nhật ký. Ví dụ: Neurallog [14] sử dụng mã hóa vị trí để biểu diễn chuỗi sự kiện nhật ký hoặc DeepLog [13] sử dụng các mô hình như LSTM để tính toán theo cách tuần tự. Để đảm bảo thông tin về thứ tự các sự kiện nhật ký không bị mất trong quá trình biểu diễn đồ thị, chúng ta biểu diễn thời gian xuất hiện của hai sự kiện nhật ký liên tiếp v_i và v_j vào e_{ij}.

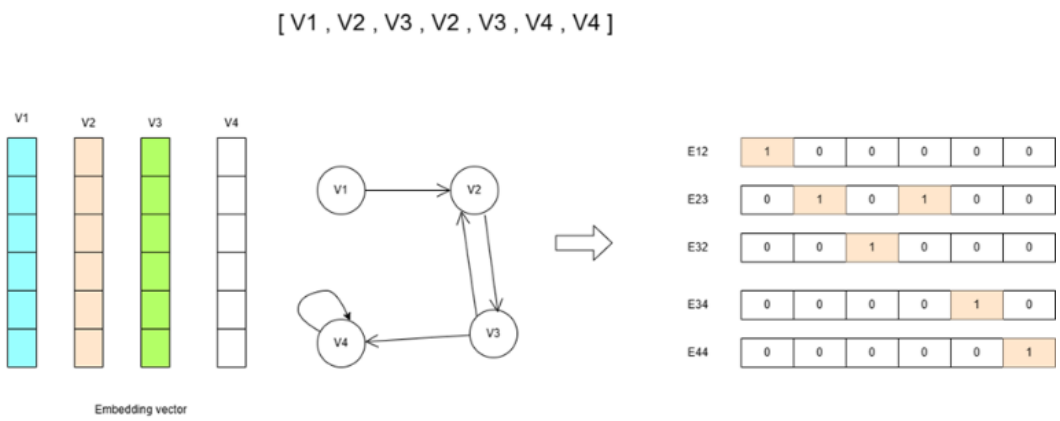
Đặt $e_{ij} = \{e_{ij}{}^{(0)}, e_{ij}{}^{(1)}, \dots e_{ij}{}^{(n)}\}$

$$e_{ij}{}^{(t)} = \{1\ if\ S_t = (v_i, v_j)\ \ 0\ another \quad (5)$$

Trong đó $t\ in\ [1, \dots, n]$ và S_t là một cặp sự kiện nhật ký liên tiếp trong S. Ví dụ như trên hình dưới. Sự kiện V2 và V3 xuất hiện liên tiếp hai lần trong dãy nhật ký nên ta có vectơ E_{23} = [0,1,0,1,0,0]. Cách biểu diễn này đảm bảo rằng biểu đồ vẫn duy trì tính tuần tự của dữ liệu gốc cũng như đảm bảo rằng việc biểu diễn dữ liệu ngắn gọn hơn khi các sự kiện trùng lặp chỉ cần được biểu diễn một lần. Trong thực tế, khi số lượng sự kiện chuỗi nhật ký lớn, các cuộc tấn công cũng có thể xảy ra trong thời gian dài, việc sử dụng kích thước cửa sổ nhỏ có thể dẫn đến khó phát hiện các sự kiện bất thường. Tuy nhiên, nếu nguồn là một chuỗi nhật ký dài thì việc biểu diễn nó theo cách này sẽ gây lãng phí tài nguyên khi có quá nhiều số 0 trong tính năng cạnh. Do đó, thay vì biểu thị số chiều của đối tượng địa lý cạnh bằng số chiều của chuỗi log, chúng ta sẽ sử dụng:

$$e_{ij}^{(t)'} = \sum e_{ij}^{(t\ mod\ m)} \quad (6)$$

Trong đó m là chu kỳ hoạt động của quá trình. Trong nghiên cứu này, sẽ sử dụng m = 60.



Hình 2.2: Xây dựng tính năng cạnh

- **Phân loại đồ thị**

Để có đầu vào cho phần phân loại đồ thị, phải cần chuyển đổi đầu ra từ phần biểu diễn đồ thị thành vectơ. Để có được vectơ này, cần sử dụng bộ tổng hợp để đọc. Có một số trình tổng hợp có thể được sử dụng, chẳng hạn như set2set, giá trị trung bình, tổng, v.v. Kết quả sau khi tổng hợp thông tin sẽ được sử dụng làm đầu vào của khối MLP. Entropy chéo được sử dụng làm hàm mất mát cho nhiệm vụ.

$$h_g = Agg(h_n) \quad (7)$$

$$\hat{y} = MLP(h_g) \quad (8)$$

Cuối cùng, mô hình dựa trên đề xuất có thể dự đoán liệu đồ thị có bất thường hay không.

2.4 Kết chương

Chương 2 đã trình bày các khái niệm cơ bản về học máy, học sâu và một số phương pháp về học sâu, bên cạnh đó cũng đã nghiên cứu một số mô hình phát hiện bất thường trong hệ thống thông tin hiện nay. Từ đó, đưa ra mô hình đề xuất phù hợp để giải quyết được bài toán hiện nay.

Chương cuối sẽ trình bày về việc áp dụng mô hình đề xuất đã trình bày ở Chương 2 để cài đặt thử nghiệm và đánh giá kết quả.

CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

3.1 Bộ dữ liệu thử nghiệm

Trong thử nghiệm này, học viên đã sử dụng 4 bộ dữ liệu phổ biến [15]: HDFS, BGL, Spirit và Thunderbird. Chúng đã được sử dụng trong nhiều nghiên cứu khác trước đây. Các bộ dữ liệu này được thu thập trong điều kiện thực tế ở các hệ thống lớn khác nhau và được chính quản trị viên hệ thống gắn nhãn. Để đảm bảo tính khách quan, việc sử dụng tập dữ liệu gốc sẽ được tải xuống từ trang web của nhà cung cấp. Chi tiết về các tập dữ liệu như sau:

- HDFS (Hệ thống tệp phân tán Hadoop) [16]
- BGL (Siêu máy tính BlueGene/L) [17]
- Spirit [17]
- Thunderbird [17]

Bảng 3.1: Mô tả cơ sở dữ liệu

Bộ dữ liệu	Nodes	Windows
HDFS	48	section
BGL	1847	100 logs
		20 logs
Spirit	1229	100 logs
		20 logs
Thunderbird	4992	100 logs
		20 logs

3.2 Tiêu chuẩn đánh giá

Để đánh giá tính hiệu quả của mô hình được đề xuất, sẽ sử dụng các số liệu về F1, khả năng thu hồi và độ chính xác:

- False Negative (FN): Số trường hợp mà mô hình dự đoán là bình thường nhưng thực tế là bất thường.
- True Negative (TN): Số trường hợp mà mô hình dự đoán đúng là bình thường.

- False Positive (FP): Số trường hợp mà mô hình xác định sai, báo cáo là bất thường, trong khi chúng thực sự là bình thường.
- True Positive (TP): Số trường hợp mà mô hình phát hiện đúng và xác định là bất thường.

Chi tiết về các chỉ số như sau:

- Recall là thước đo để đánh giá hiệu suất của mô hình phân loại, tập trung vào khả năng mô hình tìm thấy tất cả các trường hợp tích cực:

$$Recal\ (rec.)\ =\ \frac{TP}{TP\ +\ FN}(9)$$

- Precision là thước đo để đánh giá hiệu suất của mô hình phân loại, tập trung vào khả năng của mô hình trong việc dự đoán chính xác các trường hợp tích cực

$$Precision\ (prec.)\ =\ \frac{TP}{TP\ +\ FP}(10)$$

- F1-score là thước đo để đánh giá hiệu suất của mô hình phân loại, kết hợp cả độ chính xác và khả năng thu hồi.

$$F1-score(prec.)\ =\ \frac{2}{\frac{1}{Recal}+\frac{1}{Precision}}(11)$$

3.3 Cài đặt, thử nghiệm

Các thử nghiệm được thực hiện trên máy tính với thông số cấu hình của mô trường được mô tả cụ thể trong bảng sau:

Bảng 3.2: Cấu hình môi trường thử nghiệm

Thông tin	Môi trường máy huấn luyện
Vi xử lý	CPU Intel(R) Xeon(R) E5-2650 v2 @ 2.60GHz
Dung lượng RAM	62GB
Dung lượng bộ nhớ	256GB
GPU	Quandro GTX 4000 8GB

Hệ điều hành	Ubuntu 22.04.3 LTS
Python	3.8.0
Pytorch	1.13.0
torchgeometric	2.4.0

Trong nghiên cứu của học viên, học viên thử nghiệm và đặt mạng chú ý đồ thị(Graph Attention Networks - GAT) trên biểu đồ với số lần lặp là 2 và kích thước của mạng chuyển tiếp nguồn cấp dữ liệu là 512. Học viên sử dụng trình tối ưu hóa Adam cho trạng thái đào tạo và không sử dụng tính năng giảm trọng số trong giai đoạn huấn luyện và tốc độ học được sử dụng là 0,001. Học viên đặt mini-batch là 32 và tỷ lệ dropout là 0,1. Trong nghiên cứu này, mô hình được huấn luyện trong 50 epochs. Học viên sử dụng entropy chéo để tính hàm mất mát.

Học viên đã tóm tắt kích thước của một số mô hình Sota hiện tại trong Bảng 3.3. Chúng ta có thể quan sát rằng kích thước của các mô hình như NeuralLog[14] hoặc LogGD[6] sử dụng một số lượng lớn tham số. Các mô hình như DeepLog[13] và LogRobust [18] sử dụng ít tham số hơn mô hình học viên đề xuất. Tuy nhiên, số lượng tham số ít dẫn đến kết quả phân loại với tập dữ liệu của hai mô hình này không cho kết quả tốt nhất. Mô hình học viên đề xuất có số lượng tham số nhỏ hơn NeuralLog và LogGD 20 lần. Điều này đảm bảo tính khả thi khi triển khai trên các thiết bị nhỏ.

Bảng 3.3: Thông số của mô hình

Mymodel	**NeuralLog**	**LogRobust**	**DeepLog**	**LogGD**
1156610	22067714	563602	181800	19371521

3.1 Phân tích và đánh giá kết quả

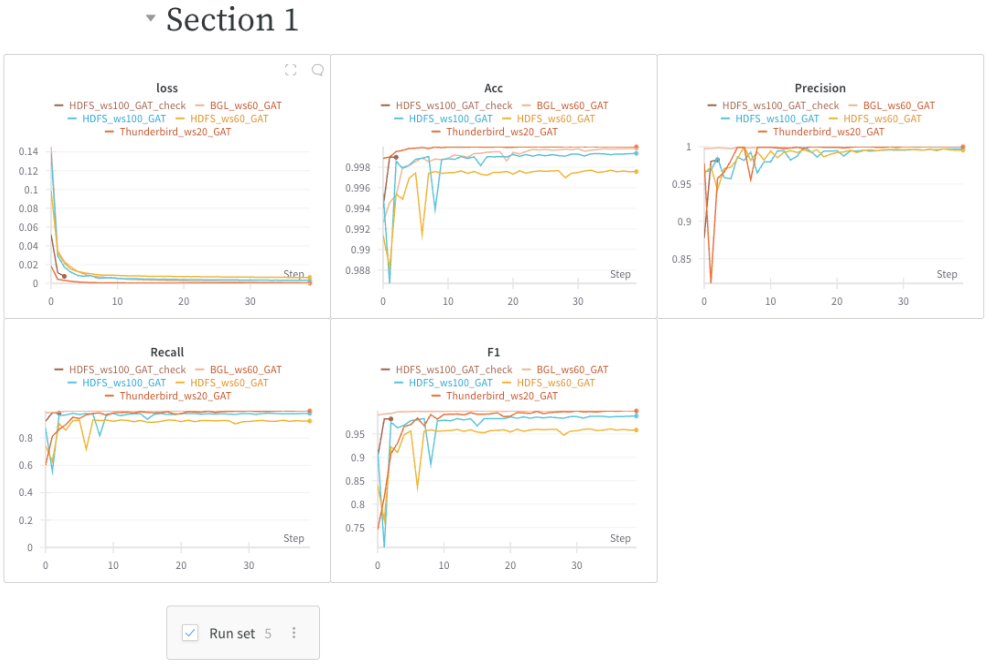
Kết quả trong Bảng 3.4 cho thấy mô hình đề xuất của học viên đạt được điểm F1 tốt nhất trên cả 4 bộ dữ liệu - HDFS, BGL, Spirit và Thunderbird. Cụ thể, mô hình của học viên đạt được điểm F1 là 0,9886, 0,9998, 0,995 và 0,992 trên các bộ dữ liệu này. Điều này chứng tỏ rằng mô hình của học viên hoạt động tốt trong việc phát

hiện sự bất thường. So với LogGD, cũng sử dụng cách tiếp cận dựa trên biểu đồ, kết quả của học viên tốt hơn trong tất cả các thử nghiệm, điều này cho thấy rằng phương pháp biểu diễn biểu đồ mà học viên đề xuất đã trích xuất tất cả thông tin liên quan giữa các sự kiện nhật ký và chuỗi sự kiện xảy ra. Các mô hình chỉ sử dụng chuỗi sự kiện nhật ký như DeepLog, NeuralLog và LogRobust cũng đạt kết quả cao với các bộ dữ liệu như HDFS hoặc BGL. Tuy nhiên, với các bộ dữ liệu không chia thành các phiên như Spirit và Thunderbird thì kết quả không ổn định. Điều này có thể là do các bộ dữ liệu này là sự kết hợp của các quy trình khác nhau trong hệ thống. Việc chỉ sử dụng trình tự khiến các mô hình thiếu sự kết nối giữa các tiến trình khác nhau, dẫn đến kết quả không cao bằng các mô hình sử dụng Graph-Based.

Bảng 3.4: Kết quả thực nghiệm

Model	**HDFS**			**BGL**			**Spirit**			**Thunderbird**		
	F1	*Rec.*	*Prec.*	*F1*	*Rec.*	*Prec.*	*F1*	*Rec.*	*Prec.*	*F1*	*Rec.*	*Prec.*
My model	0.9886	0.9803	0.997	0.9998	0.9996	1	0.995	0.992	0.999	0.992	0.986	0.998
NeuralLog	0.9827	1	0.96	0.9535	0.9586	0.9484	0.97	0.98	0.96	0.96	1	0.93
DeepLog	0.908	0.994	0.835	0.927	0.903	0.952	0.929	0.992	0.871	0.369	1	0.232
LogGD	0.9877	0.9982	0.9774	0.9719	0.9708	0.9731	0.979	0.989	0.969	0.9284	0.8889	0.9772
LogRobust	0.9819	1	0.9688	0.9402	0.9229	0.9596	0.9757	0.9957	0.9566	0.941	0.921	0.803

Nhìn chung, kết quả thử nghiệm thể hiện rõ ràng ưu điểm của mô hình đề xuất trong việc phát hiện sự bất thường từ nhật ký hệ thống. Với khả năng biểu diễn thông tin và trình tự kết nối, mô hình của học viên đạt kết quả tốt nhất về độ chính xác và điểm F1 được mô tả chi tiết dưới hình sau:



Hình 3.1: Biểu đồ dữ liệu chạy trên các tập dữ liệu

3.2 Kết chương

Chương cuối trình bày tập dữ liệu thử nghiệm, cài đặt và thử nghiệm từ đó tiến hành đánh giá hiệu quả mô hình đã được trình bày trong chương 2 và đề xuất phương pháp nghiên cứu, phát triển sau này.

KẾT LUẬN

Đề án nghiên cứu và đề xuất mô hình hình phát hiện bất thường trong các hệ thống thông tin thông qua dữ liệu log hệ thống dựa trên mô hình mạng chú ý đồ thị. Với mô hình đề xuất, các nghiên cứu đi kèm chứng minh tính hiệu quả của nó trong việc phát hiện các bất thường cho hệ thống thông tin. Học viên hy vọng rằng mô hình đề xuất có thể được áp dụng vào các ứng dụng thực tế trong tương lai. Dưới đây là một số kết quả đạt được và định hướng tương lai của học viên:

Kết quả đạt được:

- Giới thiệu và trình bày tổng quan về dữ liệu log hệ thống, phát hiện bất thường trong hệ thống thông tin, đồng thời tìm hiểu các vấn đề liên quan tới phát hiện bất thường dựa trên phân tích log.
- Giới thiệu và trình bày một số các mô hình sử dụng học máy và học sâu phát hiện bất thường dựa trên phân tích dữ liệu log hệ thống hiện nay.
- Nghiên cứu và trình bày chi tiết về mô hình phát hiện bất thường trong các hệ thống thông tin thông qua dữ liệu log hệ thống dựa trên mô hình mạng chú ý đồ thị (Graph Attention Networks – GAT).
- Tiến hành thực hiện và so sánh đánh giá kết với các một số phương pháp tiên tiến hiện có.

Hướng phát triển trong tương lai:

Với mô hình đề xuất kích thước không quá lớn và các nghiên cứu đi kèm chứng minh tính hiệu quả của nó trong việc phát hiện các bất thường cho các hệ thống thông tin. Học viên định hướng sẽ áp dụng và phát triển mô hình cho việc phát hiện bất thường dựa trên dữ liệu log hệ thống trong các thiết bị IoT vừa và nhỏ.