Học thủ tục mạnh mẽ để phát hiện bất thường

và Khả năng quan sát trong 5G RAN

# Tobias Sundqvist[, Thành](https://orcid.org/0000-0001-9013-6603) *viên, IEEE,* Monowar Bhuyan[,](https://orcid.org/0000-0002-9842-7840) *Thành viên, IEEE* và Erik Elmroth

*Tóm tắt* — Hầu hết các hệ thống phân tán lớn hiện có khả năng quan sát kém và không thể sử dụng toàn bộ tiềm năng của phân tích hành vi dựa trên máy học. Nhật ký hệ thống, chứa nguồn thông tin chính, không có cấu trúc và thiếu ngữ cảnh cần thiết để theo dõi các thủ tục và tìm hiểu hành vi của hệ thống. Công việc này trình bày một hướng dẫn theo dõi mới cho phép phân tách nhật ký hệ thống dựa trên thành phần và quy trình cho Mạng truy cập vô tuyến 5G (RAN) trong tương lai. Vì hệ thống có thể được chia thành các phần nhỏ hơn, các mô hình có thể tìm hiểu chính xác hơn hành vi của hệ thống và sử dụng ngữ cảnh để cải thiện khả năng phát hiện và quan sát bất thường. Kết quả đánh giá thật đáng kinh ngạc; Trong khi các phương pháp hiện đại trước đây gặp khó khăn trong việc học hành vi, một thuật toán nhanh, dựa trên từ điển có thể phát hiện tất cả các bất thường và giữ dương tính giả gần bằng không. Trình khắc phục sự cố cũng có thể nhanh chóng xác định các điểm bất thường và có được thông tin chi tiết hữu ích về tương tác thành phần trong RAN.

*Thuật ngữ chỉ mục* — Khả năng quan sát, hướng dẫn theo dõi, phát hiện bất thường, mạng truy cập vô tuyến, 5G.

## I. GIỚI THIỆU

M

Học tập ACHINE (ML) có thể hỗ trợ các nhà phát triển hiểu hành vi của các hệ thống phân tán lớn như Mạng truy cập vô tuyến 5G (RAN). Để đạt được toàn bộ tiềm năng của phân tích hành vi dựa trên ML, điều cần thiết là phải thiết bị phần mềm với phạm vi lớn hơn so với các nhà phát triển quen thuộc. Nhật ký hệ thống là nguồn thông tin chính khi khắc phục sự cố hệ thống. Chúng có thể tiết lộ chức năng của hệ thống và hướng dẫn các trình khắc phục sự cố trong việc tìm kiếm các điểm bất thường và nguyên nhân gốc rễ. Thật không may, nhật ký hệ thống thường không có cấu trúc và có thể thay đổi trong một hệ thống lớn, gây khó khăn cho việc phân biệt giữa hành vi bình thường và bất thường. ML đã được nhiều người coi là một giải pháp có thể gợi ý nơi các nhà phát triển nên thêm nhật ký vào mã [1], cách khai thác nhật ký [2] và cũng tìm hiểu hành vi [3]. Một vấn đề cơ bản trong nhiều hệ thống phân tán lớn là chất lượng dữ liệu thường quá kém để cho phép ML học các hành vi phức tạp của hệ thống [4]. Khi hàng trăm hoặc hàng nghìn nhà phát triển phát triển và duy trì các phần nhỏ của hệ thống, không có sự đồng thuận về việc đo lường phần mềm từ góc độ ML.

Bản thảo nhận được ngày 19 tháng 4 năm 2023; sửa đổi ngày 28 tháng 8 năm 2023; chấp nhận ngày 24 tháng 9 năm 2023. Ngày xuất bản 2 Tháng Mười 2023; ngày của phiên bản hiện tại 15 Tháng Tư 2024. Công việc này được hỗ trợ một phần bởi Chương trình Phần mềm và Hệ thống Tự trị AI, Wallenberg (WASP), được tài trợ bởi Quỹ Knut và Alice Wallenberg. Phó biên tập viên điều phối việc xem xét bài báo này và phê duyệt nó để xuất bản là D. Puthal. *(Tác giả tương ứng: Tobias Sundqvist.)*

Các tác giả làm việc tại Khoa Khoa học Máy tính, Đại học Umeå, 901 87 Umeå, Thụy Điển (e-mail: sundqtob@cs.umu.se; monowar@cs.umu.se; elmroth@cs.umu.se).

Mã định danh đối tượng kỹ thuật số 10.1109/TNSM.2023.3321401

Việc thống nhất thiết bị đo đạc và cho phép theo dõi quy trình thông qua toàn bộ hệ thống cũng là cần thiết. Gần đây, đã có những nỗ lực để xác định các hướng dẫn về thiết bị phần mềm, giải thích và sử dụng nhật ký hệ thống cho mục đích phát hiện bất thường [5]. Các hướng dẫn hỗ trợ trích xuất thông tin hữu ích từ nhật ký hệ thống, nhưng vẫn khó nắm bắt các bộ phận bị ảnh hưởng và tương tác như thế nào khi hệ thống giải quyết các tác vụ khác nhau. Tiêu chuẩn OpenTracing[[1]](#footnote-1) với OpenTelemetry2 cung cấp hướng dẫn về cách thiết bị đo lường hệ thống để theo dõi tương tác vi dịch vụ. Ví dụ, nghiên cứu trước đây dựa trên các hướng dẫn này đã minh họa cách tìm thấy các nút thắt cổ chai và độ trễ bất thường [6], [7]. Tuy nhiên, kinh nghiệm trong các ngành công nghiệp lớn, chẳng hạn như RAN, cho thấy rất khó để thực hiện theo dõi phân tán trong các hệ thống lớn [4], [8]. Thay vào đó, chúng tôi đề xuất một hướng dẫn thiết bị phần mềm dựa trên quy trình cho phép các phương pháp mới học hành vi RAN từ các sự kiện nhật ký hệ thống. Phương pháp được đề xuất của chúng tôi nâng cao khả năng quan sát của RAN và xác định các điểm bất thường nhanh hơn và chính xác hơn so với các phương pháp hiện đại.

Những đóng góp chính của công việc này là:

* Xác định hướng dẫn theo dõi dựa trên mô tả chức năng của hệ thống. Các hướng dẫn tạo điều kiện cho việc hiểu cách các đối tượng được kết nối với nhau và cho phép theo dõi các thủ tục thông qua một hệ thống phân tán rộng rãi, chẳng hạn như

5G CHẠY.

* Giải thích và trực quan hóa khả năng quan sát được cải thiện như thế nào khi thiết bị đo lường hệ thống với các hướng dẫn mới.
* Giới thiệu phương pháp phát hiện bất thường dựa trên thủ tục có thể tận dụng bối cảnh bổ sung mà các hướng dẫn mới cung cấp. Phương pháp học tương tác thành phần trong RAN cùng với hành vi của từng thành phần đối với các thủ tục được hệ thống hóa. Độ chi tiết tăng lên giúp cải thiện độ chính xác phát hiện bất thường, vượt qua các phương pháp hiện đại trước đây và nâng cao khả năng quan sát của hệ thống.
* Đánh giá khả năng phát hiện bất thường và cải thiện khả năng quan sát khi cập nhật giường thử nghiệm 5G nâng cao với các nguyên tắc theo dõi mới.

|  |
| --- |
| 1932-4537 c 2023 IEEE. Cho phép sử dụng cá nhân, nhưng tái xuất bản/phân phối lại cần có sự cho phép của IEEE. Xem https://www.ieee.org/publications/rights/index.html để biết thêm thông tin. |

*Tổ chức:* Phần còn lại của bài báo này có tổ chức sau: Phần II xem xét các nghiên cứu liên quan và Phần III cung cấp một mô tả sâu sắc về vấn đề nền tảng. Phần IV giải thích cách thực hiện các hướng dẫn theo dõi mới và cách cải thiện khả năng phát hiện và quan sát bất thường bằng cách sử dụng chúng. Phần V trình bày các kết quả thí nghiệm, được thảo luận với những hạn chế trong Phần VI. Cuối cùng, Phần VII kết thúc bài báo và đề cập đến các chủ đề thú vị trong tương lai.

## II. CÔNG VIỆC LIÊN QUAN

Công việc này cung cấp các phương pháp mới để cải thiện khả năng phát hiện bất thường và nâng cao khả năng quan sát của RAN bằng cách sử dụng hướng dẫn theo dõi phần mềm mới. Phần này khám phá những hạn chế của nghiên cứu trước đây trong ba lĩnh vực này. Các kỹ thuật liên quan cũng được khám phá thêm trong Phần IV.

### A. Hướng dẫn theo dõi

Khai thác nhật ký hệ thống để trích xuất thông tin hữu ích và sự phụ thuộc giữa các sự kiện là một lĩnh vực nghiên cứu sâu rộng; một cuộc khảo sát về một số nghiên cứu này có thể được đọc trong [2], [9]. Thay vì đề xuất một cách tiếp cận khác để khai thác nhật ký phi cấu trúc, chúng tôi xác định các hướng dẫn theo dõi, nếu tuân theo, cho phép nâng cao khả năng quan sát và thêm ngữ cảnh có giá trị cho các phương pháp học máy. Các hướng dẫn cũng có thể hỗ trợ giảm nhiễu thường được đưa vào khi thu thập và xử lý trước dữ liệu [10].

Một cuộc khảo sát ngành gần đây [4] về phân tích và theo dõi microservice kết luận nhu cầu lớn về phân tích dấu vết thông minh. Tuy nhiên, rào cản chính là thiếu chú thích dữ liệu theo dõi và chất lượng dữ liệu kém. Một nỗ lực để cấu trúc nhật ký hệ thống và làm cho chúng phù hợp hơn với các phương pháp học máy được trình bày trong [5]. Các ứng dụng trong một công ty viễn thông hàng đầu sau đó được thiết bị để tạo ra một định dạng văn bản mà các phương pháp học máy có thể dễ dàng diễn giải. Nhược điểm là định dạng mới khiến con người khó giải thích nhật ký và hiểu hệ thống hơn. Một số nỗ lực tồn tại để sử dụng tiêu chuẩn OpenTracing với OpenTelemetry để có thêm thông tin chi tiết từ các hệ thống phân tán với ngữ cảnh span và trace [6], [7], [11]. Tuy nhiên, chúng tôi chưa tìm thấy bất kỳ nghiên cứu nào điều tra xem tiêu chuẩn sẽ cải thiện các phương pháp phân tích trình tự nhật ký và khả năng quan sát như thế nào.

Biết cái gì, làm thế nào và ở đâu để thiết bị mã ngày nay là một lĩnh vực nghiên cứu lớn mà một cuộc khảo sát rộng rãi có sẵn tại [12]. Hầu hết thông tin gỡ lỗi được thêm vào trong quá trình phát triển ban đầu của các tính năng để xác minh rằng việc triển khai hoạt động như dự kiến. Thông tin gỡ lỗi bổ sung có thể được thêm vào sau khi tính năng được kiểm tra ở góc độ lớn hơn, trước khi phát hành phần mềm hoặc khi khắc phục sự cố hệ thống. Các hướng dẫn theo dõi được đề xuất trong công việc này nhằm mục đích cải thiện nhận thức về hệ thống khi tất cả các bộ phận hoạt động cùng nhau vào cuối chu kỳ phát triển.

### B. Phát hiện bất thường trong nhật ký hệ thống

Nhật ký hệ thống chứa thông tin gỡ lỗi phản ánh hành vi của hệ thống. Theo định nghĩa, bất thường là thứ gì đó đi chệch khỏi hành vi bình thường hoặc mong đợi. Các điểm bất thường có thể được phát hiện bằng cách phân tích nhật ký theo cách thủ công, nhưng lượng dữ liệu khổng lồ trong RAN khiến những người khắc phục sự cố RAN khó tìm ra sai lệch. Các phương pháp học máy có thể học hỏi từ các tính năng trong nhật ký hệ thống để giúp người dùng cuối xác định các điểm bất thường này [12], [13], [14], [15]. Một thách thức đối với các phương pháp này là chất lượng kém của dữ liệu đầu vào. Nếu không có cấu trúc và bối cảnh được thảo luận sau này trong Phần II-A, các phương pháp cần học hỏi từ dữ liệu nhiễu thiếu bối cảnh, dẫn đến hiệu suất kém. Thêm ngữ cảnh thủ tục bị thiếu bằng cách sử dụng các hướng dẫn theo dõi có thể cải thiện khả năng phát hiện bất thường và tăng khả năng quan sát của hệ thống. Một ngữ cảnh được thêm vào cũng có thể cung cấp cơ hội để tìm ra các loại dị thường mới [6], [7].

Công việc này chỉ giới hạn trong việc xác định các điểm bất thường từ cách tiếp cận có giám sát, trong đó dữ liệu đào tạo được dán nhãn là bình thường. Cách tiếp cận này phù hợp với môi trường tích hợp liên tục, nơi chức năng phần mềm được kiểm tra liên tục. Sau đó, các mô hình có thể tìm hiểu hành vi từ các trường hợp thử nghiệm đã vượt qua và xác định các điểm bất thường trong các trường hợp thử nghiệm không thành công. Các phương pháp hiện đại trước đây như LogRobust [10], DeepLog [16] hoặc BoostLog [17] có thể học hành vi từ các môi trường tích hợp liên tục như vậy, nhưng không thể khác nhau giữa nhiều kịch bản được hệ thống hóa trong RAN. Mỗi mô hình sau đó phải tìm hiểu số lượng cấu hình và tính năng ngày càng tăng trong RAN, dẫn đến các mô hình lớn hơn và phức tạp hơn. Các phương pháp được đề xuất trong công việc này sử dụng thông tin quy trình được thêm vào nhật ký hệ thống để tự động nhóm các hành vi tương tự và phát hiện sai lệch nhanh hơn và chính xác hơn so với các phương pháp trước đó.

### C. Khả năng quan sát

Hiểu cách các bộ phận hệ thống tương tác là điều cần thiết và nhiều điều đã được thực hiện để cải thiện khả năng quan sát. Một cách tiếp cận là khai thác nhật ký hệ thống từ các hệ thống phân tán để tìm sự phụ thuộc giữa các phần hệ thống [18]. Một phương pháp khác là tăng cường hoặc giám sát giao thức giữa các ứng dụng để theo dõi đường dẫn tin nhắn. Đây là trường hợp, ví dụ, các công cụ như X-Trace [19] hoặc CloudSeer [20]. CloudSeer cũng thiết bị cho các ứng dụng để trích xuất thông tin khi tin nhắn được gửi giữa các phần phân tán. Một ví dụ khác về cách tiếp cận này được trình bày trong [21], trong đó các tác giả đo lường một hệ thống tìm nguồn cung ứng sự kiện để theo dõi các đường dẫn yêu cầu thông qua một hệ thống. Nhiều giao thức 5G RAN không thể được mở rộng để bao gồm thông tin bổ sung được đề xuất bởi nghiên cứu trước đây. Công trình này đề xuất một giải pháp thay thế trong đó tương tác hệ thống có thể được xác định tự động thông qua học thủ tục.

Nhiều ứng dụng microservice công nghiệp đã triển khai theo dõi phân tán để cải thiện khả năng quan sát của hệ thống [22]. Thông tin theo dõi cung cấp thông tin hữu ích có thể giúp người dùng cuối hiểu được tương tác microservice, nhưng việc tích hợp vào các hệ thống lớn là có vấn đề [4], [8]. Nhiều giao thức trong RAN không thể được thay đổi để thêm thông tin bổ sung mà theo dõi phân tán yêu cầu. Thay vào đó, công trình này khám phá cách tăng cường khả năng quan sát bằng cách sử dụng thông tin trong nhật ký hệ thống. Mặc dù theo dõi phân tán chỉ cung cấp thông tin về tương tác vi dịch vụ, nhưng nhật ký hệ thống cũng chứa thông tin hành vi có giá trị giúp các nhà phát triển khắc phục sự cố hệ thống.

Một điểm khả năng quan sát *đặc biệt* được định nghĩa trong Phần IV-D. Điểm số đo lường tốc độ trình khắc phục sự cố hiểu các quy trình đang diễn ra và xác định các điểm bất thường. Các nhà nghiên cứu khác, ví dụ, [23], đã sử dụng thuật ngữ này để giám sát các cuộc tấn công bảo mật và điểm số sau đó phản ánh số lượng các tính năng được phát hiện. Trong [24] và [25], điểm số đại diện cho việc theo dõi một robot di động, tương ứng, một vật thể. Vì khả năng quan sát là một lĩnh vực rộng, các ý kiến khác nhau về cách đo lường khả năng quan sát. Theo hiểu biết tốt nhất của chúng tôi, đây là lần đầu tiên khả năng quan sát được đo lường từ góc độ phát hiện thủ tục và bất thường, nơi mọi người, không phải thuật toán, cố gắng giải thích các trạng thái bên trong của hệ thống.

## III. BỐI CẢNH

Phần này giải thích những thách thức của việc xác định các bất thường nhật ký trong các hệ thống phân tán lớn. Nó cũng xác định thuật ngữ được sử dụng thêm trong tác phẩm này.

### A. Thuật ngữ

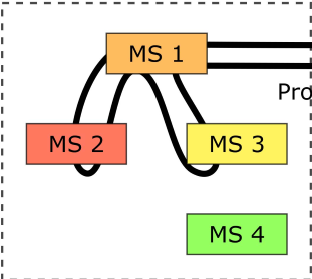
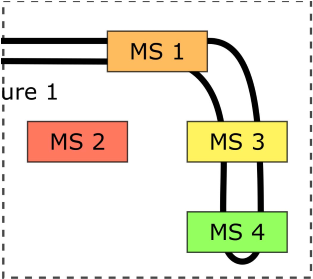
Để dễ hiểu hơn những thách thức và phương pháp trong công việc này, một số thuật ngữ cần được định nghĩa:

* *Thành phần* - Một phần của hệ thống, ví dụ: một microservice, một phần của microservice, container hoặc một khu vực được hệ thống hóa thực hiện một tác vụ cụ thể.
* *Thủ tục* - Một nhiệm vụ được xử lý bởi hệ thống, thường được xác định trong mô tả chức năng của hệ thống.
* *Quy trình phụ* - Một số thành phần hệ thống có thể hoạt động cùng nhau để thực hiện một quy trình, ví dụ: trong một hệ thống phân tán với một số vi dịch vụ hoặc bộ chứa. Chúng tôi định nghĩa một thủ tục phụ là công việc được thực hiện bởi một thành phần duy nhất.
* *Nhật ký hệ thống* - Nhật ký chứa thông tin gỡ lỗi có giá trị. Thông tin gỡ lỗi thường được thêm vào trong quá trình thiết kế hoặc kiểm tra hệ thống để hỗ trợ khắc phục sự cố.
* *Sự kiện* - Thiết bị đo phần mềm lưu trữ thông tin quan trọng dưới dạng một hoặc một số dòng trong nhật ký hệ thống; chúng tôi gọi các dòng này là sự kiện.

### B. Định nghĩa vấn đề

Để hiểu sự cần thiết của một hướng dẫn theo dõi mới, trước tiên chúng tôi mô tả sự phát triển của các hệ thống lớn. Khi lập kế hoạch các tính năng mới, đặc điểm kỹ thuật thiết kế chức năng (FDS) chỉ định các yêu cầu và mô tả triển khai cấp cao. FDS chứa lưu đồ, ảnh chụp màn hình, sơ đồ trình tự hoặc trường hợp sử dụng và là nguồn thông tin chính cho các nhà phát triển triển khai tính năng này. Tính năng này có thể ảnh hưởng đến một số bộ phận hệ thống và nhiều nhóm và nhà phát triển thường tham gia vào việc triển khai. Các thử nghiệm ở các cấp hệ thống khác nhau đảm bảo rằng các bộ phận riêng lẻ và toàn bộ hệ thống hoạt động. Sau đó, thiết bị đo phần mềm cung cấp thông tin có giá trị trong nhật ký hệ thống hoặc đầu ra tiêu chuẩn của hệ thống.

Việc xác định lỗi thường nhanh chóng trong các hệ thống nhỏ, nơi các nhà phát triển là chuyên gia về toàn bộ hệ thống. Tuy nhiên, việc xác định vị trí lỗi trong các hệ thống phân tán lớn có thể rắc rối hơn nhiều, trong đó trình khắc phục sự cố chỉ biết các phần nhỏ của hệ thống. Mỗi nhà phát triển quyết định thông tin gỡ lỗi để lưu trữ và thiết bị đo lường có thể khác nhau đáng kể giữa các thành phần. Để theo dõi các quy trình ảnh hưởng đến một số thành phần, trình khắc phục sự cố cần phải là chuyên gia về tất cả các bộ phận hoặc tham khảo ý kiến của các chuyên gia khác để giải thích chính xác các sự kiện và liên hệ chúng với tính năng được hệ thống hóa trong FDS. Sự thay đổi to lớn trong thiết bị đo đạc là rắc rối đối với



Hình 1. Một ví dụ đơn giản về cách bốn microservice bị ảnh hưởng bởi hai quy trình khác nhau.



Hình 2. Ví dụ nhật ký đơn giản hóa từ hai thủ tục đồng thời.

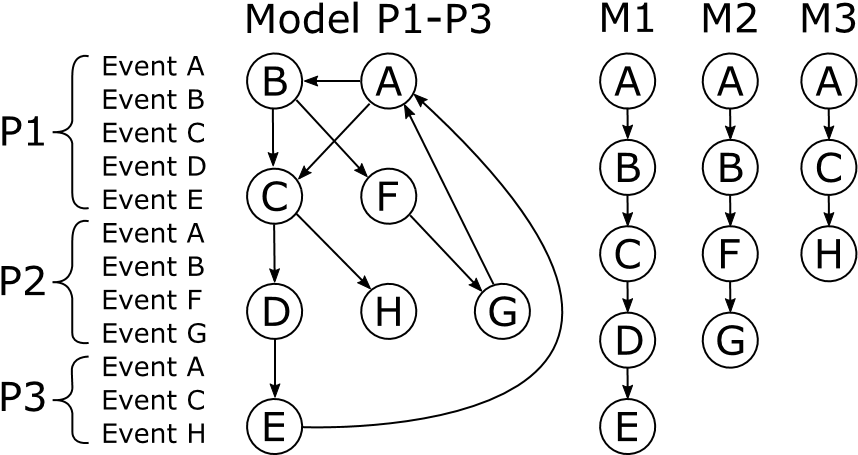
xã hội học máy, và thường rất khó để trích xuất thông tin chính xác [5]. Các nhà phát triển cũng có xu hướng thay đổi thiết bị trong quá trình phát triển và các nghiên cứu trước đây đã chỉ ra rằng 20-45% các câu lệnh ghi nhật ký thay đổi trong suốt vòng đời của hệ thống [10], [26]. Công việc này nhấn mạnh tầm quan trọng của việc thống nhất thiết bị đo lường trong toàn bộ hệ thống và làm cho các nhà phát triển nhận thức được rằng một thay đổi nhỏ trong thiết bị đo lường có thể ảnh hưởng đến các mô hình máy học học từ nhật ký hệ thống.

Một vấn đề khác là thiếu ngữ cảnh trong nhật ký hệ thống cho phép liên kết giữa các sự kiện và một thủ tục cụ thể. Điều này làm giảm khả năng quan sát của hệ thống và tăng nhiễu cho các thuật toán học máy. Để minh họa cho vấn đề, chúng tôi xem xét một ví dụ đơn giản hóa trong đó một hệ thống phân tán bao gồm bốn microservices (MS) xử lý hai thủ tục khác nhau, xem Hình. 1.

Các mũi tên trong hình minh họa thứ tự kích hoạt của MS. Các quy trình có thể xảy ra song song với nhau và ảnh hưởng đến các đối tượng khác nhau. Ví dụ đơn giản hóa chỉ lưu trữ một sự kiện mỗi khi MS bị ảnh hưởng bởi một quy trình, xem Hình. 2.

Từ danh sách sự kiện được lưu trữ ở bên trái, có thể xác định rằng một số thủ tục đồng thời đang diễn ra. Tuy nhiên, rất khó để nắm bắt những sự kiện nào thuộc về một thủ tục cụ thể mà không có bối cảnh thủ tục. Một vấn đề khác là một quy trình có thể kích hoạt các sự kiện khác nhau tùy thuộc vào đối tượng bị ảnh hưởng và các quy trình trước đây. Trong những trường hợp như vậy, chúng ta cũng cần ngữ cảnh đối tượng để nhận ra các thủ tục đã ảnh hưởng đến đối tượng như thế nào.

Từ góc độ học máy, điều cần thiết là phải phân biệt giữa các đối tượng trong nhật ký hệ thống, ví dụ: được sử dụng trong [16], [17], [27]. Việc tách các đối tượng cũng giúp các trình khắc phục sự cố theo dõi và so sánh chuỗi sự kiện, nhưng sẽ hiệu quả hơn nếu quy trình đang diễn ra được biết. Nếu không có thông tin về quy trình, trình khắc phục sự cố cần nhiều kinh nghiệm để liên hệ từng sự kiện với một quy trình cụ thể. Các



Hình 3. Chuỗi các sự kiện dễ học hơn khi bối cảnh thủ tục được biết.

Vấn đề là chuyên môn của trình khắc phục sự cố thường bị giới hạn trong một phần nhỏ của một hệ thống mở rộng. Cũng có thể khó phân biệt giữa hành vi bình thường và bất thường khi quy trình ảnh hưởng đến một số bộ phận của hệ thống. Để minh họa thêm vấn đề khi xử lý nhiều thủ tục, chúng tôi kiểm tra trình tự các sự kiện trong Hình. 3.

Một phương pháp học máy không nhận thức được bối cảnh thủ tục phải học hành vi phức tạp được coi là Model P1-P3. Công việc này giải thích cách ngữ cảnh thủ tục được thêm vào cho phép chúng ta đào tạo các mô hình riêng biệt cho từng thủ tục và thành phần.

Khi các quy trình ảnh hưởng đến một số thành phần trong hệ thống, như minh họa trong Hình. 2, điều cần thiết là phải biết mối quan hệ giữa các đối tượng và các thành phần. Nghiên cứu trước đây đã cố gắng sử dụng ID đối tượng và các thông điệp nhật ký cụ thể để tạo ra các mối quan hệ phân cấp giữa các đối tượng [28], [ 29]. Thay vì sử dụng các phương pháp thống kê để đoán đối tượng và thành phần nào thuộc về, chúng tôi đề xuất sử dụng thiết bị bổ sung để liên kết các bộ phận lại với nhau.

## IV. THIẾT KẾ HỆ THỐNG

Để đối phó với các vấn đề được mô tả trong Phần III-B, công việc này bắt đầu bằng cách xác định một hướng dẫn theo dõi mới trong Phần IV-A. Các hướng dẫn mới bổ sung thông tin thủ tục vào thông tin gỡ lỗi hiện có và làm nổi bật tầm quan trọng của nhật ký có cấu trúc. Phần IV-B sau đó giải thích cách thông tin thủ tục này có thể được sử dụng bằng một phương pháp dựa trên thủ tục mới để cải thiện tốc độ và độ chính xác của việc phát hiện bất thường. Phần IV-C mô tả thêm cách tăng cường khả năng quan sát bằng cách sử dụng các phương pháp dựa trên thủ tục mới để trực quan hóa hành vi của hệ thống và làm nổi bật nơi xảy ra bất thường. Cuối cùng, quá trình đánh giá được mô tả trong Phần V-A.

### A. Hướng dẫn theo dõi

Có nhiều mức độ trưởng thành khác nhau khi nói đến khả năng gỡ lỗi hệ thống. Với khả năng gỡ lỗi, chúng tôi muốn nói đến khả năng quan sát một hệ thống bằng cách phân tích nhật ký hệ thống. Độ trưởng thành của hệ thống có thể được chia thành các cấp độ sau:

1. Không có sự kiện nào được lưu trữ; Không thể sử dụng nhật ký hệ thống để gỡ lỗi hệ thống.
2. Một số sự kiện được lưu trữ, nhưng không có mức độ đo lường nào được sử dụng.
3. Có thể phân biệt giữa DEBUG, INFO, WARNING và ERROR.

Hình 4. Ví dụ về các sự kiện phi cấu trúc.

1. Các cấp độ thiết bị đo lường khác nhau được sử dụng cho các dấu vết DEBUG, giúp giám sát các bộ phận cụ thể của một hệ thống mở rộng dễ dàng hơn.
2. Các sự kiện được cấu trúc để giúp trích xuất thông tin có ý nghĩa dễ dàng hơn.
3. Các sự kiện được cung cấp một bối cảnh cho phép theo dõi quy trình và hiểu cách các bộ phận hoạt động cùng nhau để thực hiện các thủ tục.

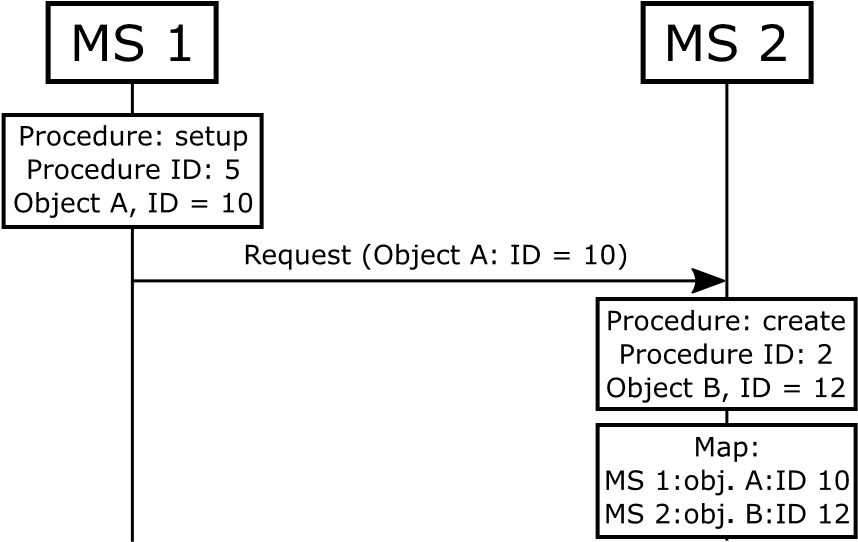
Tất cả các cấp độ thấp hơn được đáp ứng cho mỗi cấp độ và khi các cấp độ tăng lên, khả năng quan sát của hệ thống cũng được cải thiện. Hầu hết các hệ thống lớn không bao giờ đạt đến cấp độ 4, nơi các phương pháp học máy có thể trích xuất dữ liệu từ nhật ký hệ thống một cách hiệu quả. Công việc này thể hiện lợi thế của việc đạt đến cấp độ 5 bằng cách sử dụng các hướng dẫn theo dõi mới.

Nhật ký đơn giản hóa trong Hình. 4 minh họa nhược điểm của việc chỉ đạt đến cấp độ 3 về khả năng gỡ lỗi.

Cả hai sự kiện trong hình đều chỉ ra cùng một điều và một chuyên gia con người, người tự động lọc ra các từ khóa như trạng thái, người mang và thiết bị người dùng, có thể dễ dàng giải thích chúng. Để sử dụng thông tin này trong các thuật toán học máy, các chuyên gia phải tạo ra các quy tắc phân tích cú pháp thông minh được áp dụng cho phiên bản hệ thống cụ thể. Khi thông tin gỡ lỗi được thay đổi hoặc thêm vào trong quá trình phát triển, có nhu cầu liên tục cập nhật trình phân tích cú pháp và đây không phải là một giải pháp lâu dài tốt. Một cách tiếp cận phù hợp hơn cho học máy là tách thông tin thành một phần nhận dạng và một phần tham số [5]. Mục đích của phần nhận dạng là xác định vị trí phần bị ảnh hưởng của hệ thống, có thể tùy thuộc vào hệ thống, ví dụ: máy chủ, tệp, hàm, thành phần, lớp hoặc thực thể của một đối tượng. Phần tham số phải chứa thông báo thông tin về sự kiện hoặc cung cấp các tham số chính quan trọng từ góc độ gỡ lỗi. Trong ví dụ trong Hình. 4, phần nhận dạng và tham số sẽ là *thiết bị người dùng 5* và *trạng thái: ENABLED, số người mang 1*, tương ứng. Biết mối liên hệ giữa các sự kiện và thủ tục là rất quan trọng để nâng cao hơn nữa khả năng quan sát của hệ thống. Các phương pháp học máy có thể tự động xác định các quy trình khác nhau và đào tạo mô hình cho từng quy trình bằng cách thêm ngữ cảnh thủ tục. Tóm lại, chúng tôi đề xuất các nhà phát triển sử dụng thông tin gỡ lỗi sau khi đo lường phần mềm:

* *Thời gian* - Thời gian sự kiện xảy ra, còn được gọi là dấu thời gian. Nó cho phép sắp xếp các sự kiện và đo thời gian giữa các thủ tục và sự kiện.
* *Nhận dạng* - Phần này thường chứa một hệ thống phân cấp các đối tượng được sử dụng để xác định vị trí đối tượng cấp thấp nhất. Ví dụ, nó có thể là một máy chủ, thành phần hoặc đối tượng xử lý một yêu cầu.
* *Tham số* - Thông báo cung cấp thông tin chi tiết về hệ thống. Ví dụ: cập nhật trạng thái, biến, tín hiệu nhận được hoặc dịch vụ đã hoàn thành và bắt đầu.
* *Thủ tục* - Chứa hai trường: danh tính và tên thủ tục. Danh tính là một số tuần tự được sử dụng

Hình 5. Ví dụ về một sự kiện được viết với các nguyên tắc theo dõi mới.



Hình 6. Ánh xạ giữa đối tượng A trong microservice 1 và đối tượng B trong microservice 2.

để kết nối các sự kiện với một quy trình nhất định. Tên thủ tục liên quan đến các sự kiện với các thủ tục được mô tả trong FDS. Một số sự kiện trong một quy trình có thể sử dụng trường tên thủ tục. Trong những trường hợp này, sự kết hợp của các tên thủ tục sẽ tạo ra một tên duy nhất cho thủ tục đó. Một sự kiện có thể nêu Thiết *lập*, một sự kiện khác: *Phương án 2* và tên thủ tục kết hợp sau đó sẽ là *Thiết lập:Phương án 2*.

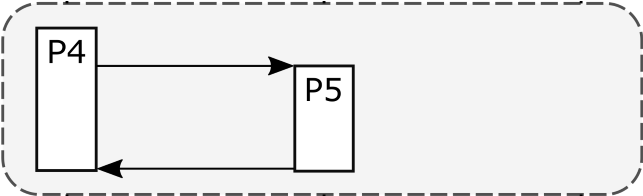
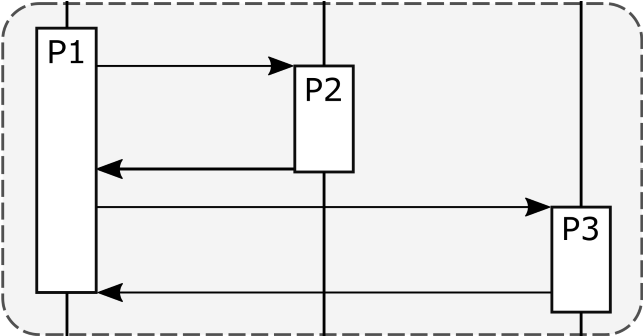
Sung. 5 minh họa một ví dụ về một sự kiện được viết với các hướng dẫn mới. Sự kiện bắt đầu bằng dấu thời gian và tiếp tục với Danh tính, Tham số và Thủ tục. Các sự kiện cũng có thể chứa thông tin như hàm, dòng mã hoặc tên ứng dụng. Loại thông tin này có thể hữu ích trong việc nhanh chóng xác định vị trí của các sự kiện trong phần mềm, nhưng nó không được điều tra thêm trong công việc này.

Trong các hệ thống mà một số thành phần xử lý một quy trình, điều quan trọng là phải cung cấp thông tin cho phép theo dõi quy trình trên toàn bộ hệ thống. Một giải pháp dễ dàng là mở rộng các giao thức giữa các thành phần để bao gồm ID thủ tục. Thật không may, điều này là không thể trong các hệ thống như RAN, nơi nhiều giao thức tiêu chuẩn hóa không cho phép thông tin này. Các giao thức chỉ chứa danh tính của các đối tượng bị ảnh hưởng nhưng tùy thuộc vào mỗi nhà cung cấp để hệ thống hóa các thủ tục và quyết định cách các bộ phận hệ thống sẽ tương tác để thực hiện các thủ tục. Để khắc phục vấn đề này khi các giao diện bị hạn chế, công trình này đề xuất một giải pháp thay thế để theo dõi các thủ tục thông qua hệ thống. Hãy xem xét hai microservice trong Hình. 6.

Microservices 1 và 2 là một phần của cùng một quy trình, nhưng giao thức tín hiệu không cho phép gửi ID thủ tục giữa các microservices. Thay vào đó, hai ID thủ tục duy nhất được đặt trong hai microservice. Thông tin thủ tục trong ví dụ sẽ nói rằng có một thủ tục phụ được gọi là *thiết lập* với ID thủ tục 5 ảnh hưởng đến đối tượng A với ID 10 và một thủ tục phụ khác được gọi là *tạo với* ID thủ tục 2 cho đối tượng B có ID 12.

Một kỹ thuật lập bản đồ đặc biệt được giới thiệu để kết nối các đối tượng và quy trình cho phép theo dõi tương tác quy trình thông qua hệ thống. Hãy xem xét ví dụ trong Hình. 6

Hình 7. Ví dụ về sự kiện ánh xạ kết nối hai đối tượng.



Hình 8. Nhóm các thủ tục thuộc về nhau, dựa trên các tín hiệu được gửi.

trong đó một quy trình phụ trong MS1 kích hoạt một quy trình phụ trong MS2. Các đối tượng A và B trong Hình. Ví dụ, 6 có thể là cùng một đối tượng tế bào hoặc một thiết bị người dùng yêu cầu cập nhật một đường hầm hoặc bộ mang vô tuyến cụ thể. Để ánh xạ các thủ tục phụ với nhau và kích hoạt các thủ tục theo dõi thông qua hệ thống, một sự kiện đặc biệt được kích hoạt khi các thành phần tương tác:

• *Mapping* - Một sự kiện đặc biệt kết nối các đối tượng từ hai thành phần. Ánh xạ phải chứa danh tính và tên của hai đối tượng và cho phép ghép nối các đối tượng và thành phần. ID đối tượng thường được gửi giữa các thành phần trong RAN và thông tin ánh xạ có thể được lưu trữ khi nhận được một số yêu cầu hoặc tín hiệu chỉ báo nhất định.

Trong khi thông tin thủ tục kết nối ID thủ tục với ID đối tượng, thông tin ánh xạ kết nối hai đối tượng. Một ví dụ về dấu vết ánh xạ có thể được nhìn thấy trong Hình. 7.

Thông tin ánh xạ nói rằng đối tượng A có ID 10 được kết nối với đối tượng B có ID 12. Kết hợp ánh xạ và thông tin thủ tục ngụ ý rằng thiết lập các thủ tục phụ cho MS1 và *tạo cho* MS2 có thể là một phần của cùng một thủ tục. Sau đó, các thủ tục phụ nên được nhóm thành các thủ tục. Hãy xem xét ví dụ trong Hình. 8 trong đó các quy trình phụ P1, P2 và P3 là một phần của quy trình 1 và các quy trình phụ P4 và P5 là một phần của quy trình 2.

Luôn có một điểm khởi đầu cho các thủ tục được hệ thống hóa; trong Hình. 8, nó xảy ra ở đầu P1 và P4. Các thủ tục phụ này được định nghĩa là *Thủ tục kích hoạt* và có thể hoạt động như một bộ chia thủ tục. Một giải pháp dễ dàng để đánh dấu chúng là thêm một thẻ chuỗi duy nhất vào thông tin tên thủ tục. Tất cả các thủ tục phụ được ánh xạ đến Triggerprocedure thuộc cùng một thủ tục. Phần IV-B1 giải thích thêm về quy trình nhóm.

### B. Phát hiện bất thường

Trong tác phẩm này, một sự bất thường được định nghĩa là một chuỗi các sự kiện khác với hành vi bình thường. Các phương pháp học được từ hành vi nhật ký hệ thống và các sai lệch có thể phát hiện được phân loại như sau:

* *Thứ tự sự kiện* - Đối với một số tính năng, các sự kiện phải xảy ra theo một thứ tự được xác định rõ ràng; nếu không, thủ tục sẽ thất bại.
* *Sự kiện bị thiếu hoặc bổ sung* - Các sự kiện không được nhìn thấy trước đó hoặc đường dẫn thực thi mới có thể gây ra các sự kiện bổ sung thường không thuộc về một thủ tục. Chức năng cũng có thể bị lỗi, dẫn đến rút ngắn các thủ tục và thiếu các thủ tục phụ.
* *Độ trễ* - Cập nhật phần mềm, tải nặng hoặc xung đột trong hệ thống có thể làm trì hoãn một phần của quy trình. Đo thời gian giữa các sự kiện trong một quy trình có thể phát hiện những độ trễ này và tổng thời gian thủ tục có thể được tính bằng cách sử dụng dấu thời gian.
* *Kiểm tra thông số* - Phạm vi thông số thông thường trong một quy trình có thể là một phần của các sự kiện và được sử dụng để phát hiện sai lệch.

Vì hầu hết các nghiên cứu trước đây đã phát triển các thuật toán để phát hiện các dị thường trong hai loại đầu tiên, phạm vi của công việc này chỉ giới hạn trong việc bao gồm hai loại này.

Trong mười năm qua, việc sử dụng các mạng nơ-ron tái phát đã trở nên phổ biến, chẳng hạn như Bộ nhớ ngắn hạn dài (LSTM) [30] hoặc Đơn vị tái phát có cổng (GRU) [31], để phát hiện các bất thường trong nhật ký hệ thống. Vào năm 2017, DeepLog [16] là một ví dụ hiện đại về cách nhật ký hệ thống có thể được chia thành một chuỗi các sự kiện cho các đối tượng khác nhau và phương pháp LSTM đã học thứ tự của các sự kiện tuần tự. Công việc của chúng tôi sau đó đã cải thiện độ chính xác phát hiện bất thường bằng cách sử dụng tập hợp AdaBoost của các mô hình LSTM [17]. Lớp chú ý [32] được giới thiệu vào năm 2017 và đã giúp tăng cường phát hiện bất thường trong nhật ký hệ thống hơn nữa [10], [28], [33], [34], [35], [36], [37].

Phần còn lại của phần này mô tả thuật toán học thủ tục mới của chúng tôi cùng với các phương pháp cơ bản là một phần của đánh giá.

*1) Học thủ tục:* Kiến trúc của phương pháp dựa trên thủ tục mới của chúng tôi được trình bày trong Hình. 9. Hệ thống này là một nền tảng thử nghiệm 5G RAN lưu trữ nhật ký hệ thống cho các kịch bản điều khiển và mặt phẳng người dùng. Phương pháp truyền thống để phân tích nhật ký là tạo quy tắc hoặc sử dụng máy học để phân tích cú pháp nhật ký và truy xuất thông tin hữu ích. Trong các hệ thống xử lý nhiều đối tượng đồng thời, đôi khi có thể tách các sự kiện thuộc về cùng một đối tượng [16], [17]. Sau đó, các phương pháp học máy (ML) có thể tìm hiểu thứ tự của các sự kiện của một đối tượng. Một vấn đề với cách tiếp cận này là các quy tắc và trình phân tích cú pháp sẽ mất một số thông tin [10] và các phương thức ML cần tìm hiểu tất cả các sự kiện thuộc về một đối tượng. Độ chính xác của các phương pháp có thể sẽ giảm đi khi hệ thống phát triển với nhiều tính năng hơn.

Nguyên tắc theo dõi của chúng tôi cho phép tự động trích xuất thông tin quan trọng mà không cần trình phân tích cú pháp hoặc quy tắc phức tạp. Nó cũng có thể giúp phân biệt giữa các thủ tục và tự động tìm hiểu hành vi cho từng thủ tục và loại đối tượng.

Chúng tôi định nghĩa một phương pháp dựa trên từ điển mới để chứng minh cách một phương pháp ít phức tạp hơn có thể được hưởng lợi từ ngữ cảnh thủ tục và thành phần. Chúng tôi tạo ra hai từ điển trong đó từ điển đầu tiên có thể phát hiện các bất thường trong các thủ tục phụ và từ điển thứ hai xử lý các bất thường phát sinh do các thủ tục con mới, thiếu hoặc bổ sung. Các từ điển lần lượt được gọi là *từ điển trình tự* và từ *điển thủ tục*. Các phương pháp tìm hiểu sự xuất hiện của các sự kiện trong các thủ tục phụ và các thủ tục phụ nào xảy ra cho một thủ tục.

Quy trình chính để đào tạo *từ điển trình tự* tuân theo bốn bước:

*Bước 1:* Các thủ tục phụ được nhóm dựa trên ID thủ tục và tên quy trình.

*Bước 2:* Tất cả các phần tử tên thủ tục cho một thủ tục con được nối với tên quy trình để tạo thành khóa thủ tục (*process-name$procedure-names*). *Bước 3:* Tạo vectơ sự kiện cho mỗi thủ tục phụ nơi lưu trữ số lần xuất hiện của mỗi sự kiện. *Bước 4:* Nối vectơ sự kiện thành một chuỗi và thêm nó làm phần tử từ điển trong thủ tục.

Sau đó, quy trình phụ có thể được nhóm thành các thủ tục bằng các bước sau:

*Bước 1:* Sắp xếp tất cả các thủ tục phụ được tìm thấy trong bước 1 của giai đoạn đào tạo cho từ *điển trình tự*, với dấu thời gian tăng dần. Kiểm tra tất cả các tên thủ tục trong danh sách và lặp lại các bước 2-3 cho mỗi tên.

*Bước 2:* Kiểm tra tên thủ tục để xác định xem thủ tục phụ có phải là thủ *tục kích hoạt* hay không. Nếu vậy, hãy tạo một vectơ trống với kích thước của tất cả các thủ tục con duy nhất ở bước 1. Vectơ này chứa số lần xuất hiện của mỗi thủ tục con trong thủ tục. Nếu một vectơ chứa một thủ tục con đã được xác định (xem bước 3), thì lưu trữ vectơ trước đó dưới dạng một chuỗi được nối với tên *Trigger-procedures* trong từ *điển thủ tục*.

*Bước 3:* Nếu thủ tục phụ không phải là thủ *tục kích hoạt*, hãy tăng số lượng thủ tục phụ trong vectơ thủ tục hiện tại.

*Bước 4:* Cuối cùng, kiểm tra thông tin ánh xạ để đảm bảo rằng tất cả các thủ tục phụ thuộc về nhau. Nếu có nhiều quy trình con trong vectơ thủ tục, hãy loại bỏ những thủ tục con không thể được ánh xạ với quy trình con khác.

Các quy trình phụ đầu tiên được xác định trong dữ liệu trong các bước tương tự như đối với từ *điển trình tự* để phát hiện các bất thường. Mỗi thủ tục phụ sau đó được kiểm tra theo các bước: *Bước 1:* Tạo khóa *process-name$procedure-names* và truy xuất tất cả các vectơ sự kiện từ từ *từ điển trình tự*. Nếu *process-name$procedure-names* không có trong từ *điển trình tự,* hãy đánh dấu quy trình con là bất thường và bỏ qua các bước còn lại. *Bước 2:* Tạo vectơ sự kiện và so sánh nó với vectơ sự kiện từ bước 2.

*Bước 3:* Đánh dấu quy trình phụ là bất thường nếu không có vectơ sự kiện nào khớp.

*Bước 4:* Nếu vectơ sự kiện bất thường, hãy tính khoảng cách Euclid giữa tất cả các vectơ sự kiện và chọn vectơ có khoảng cách nhỏ nhất. Trình bày quy trình phụ tương tự nhất với trình khắc phục sự cố; tìm hiểu thêm về điều này trong Phần IV-C.

Phần thứ hai sử dụng từ *điển thủ tục* để phát hiện các bất thường của thủ tục. Các thủ tục được phát hiện đầu tiên bằng cách sử dụng các bước để tạo từ *điển thủ tục*. Quy trình được xác định sau đó được so sánh với các thủ tục trong từ *điển thủ tục* bằng cách sử dụng các bước:

|  |
| --- |
| Hình 9. Các hướng dẫn theo dõi mới cho phép tự động trích xuất các sự kiện cho một số quy trình nhất định. |

Hình 10. Kiến trúc LSTM, X là đầu vào và h là đầu ra cho mỗi dự đoán.

*Bước 1:* Kiểm tra xem tên *Trigger-procedures* có được tìm thấy trong từ *điển thủ tục hay không*. Nếu không, hãy đánh dấu quy trình là một bất thường.

*Bước 2:* Nếu *tìm thấy Trigger-procedures* , hãy so sánh các lần xuất hiện của quy trình con với các vectơ được lưu trữ trong từ *điển procedure*.

*Bước 3:* Đánh dấu quy trình là bất thường nếu không tìm thấy vectơ giống hệt nhau.

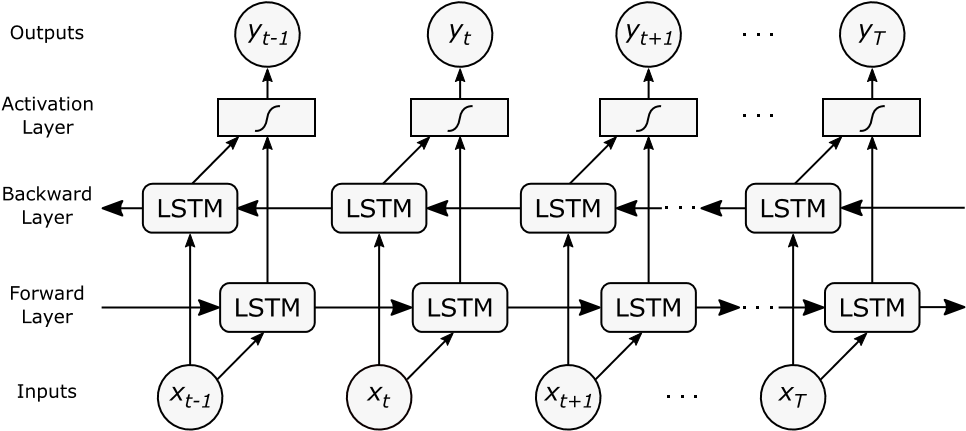
*Bước 4:* Nếu phát hiện thấy bất kỳ điểm bất thường nào, hãy sử dụng khoảng cách Euclid để tìm vectơ quy trình tương tự nhất và trình bày kết quả cho trình khắc phục sự cố.

*2) Bi-LSTM-Chú ý:* Như đã thảo luận, một số phương pháp phát hiện bất thường trạng thái của tim sử dụng các biến thể mạng nơ-ron tái phát (RNN) kết hợp với lớp chú ý. Chúng tôi chọn thực hiện một trong những phương pháp phổ biến nhất được gọi là LogRobust [10] và giải thích cách Bi-LSTM và sự chú ý được sử dụng để phát hiện các điểm bất thường trong nhật ký.

LSTM có bộ nhớ trong và có thể ghi nhớ các đầu vào trong một thời gian dài. Thay vì chỉ có một mạng nơ-ron trong mỗi mô-đun, như RNN, LSTM sử dụng một số mạng nơ-ron tương tác. Bằng cách này, LSTM có thể chọn đầu vào, xóa thông tin không quan trọng và kiểm soát đầu ra của dữ liệu, xem Hình. 10.

LSTM sử dụng một chuỗi các sự kiện trước đó để dự đoán sự kiện tiếp theo. Nếu toàn bộ chuỗi sự kiện là *S* = [*s1,s2,...,sN*], thì các dãy con *Si* =

[*sm−1,sm−2,...,si*] được sử dụng làm đầu vào, trong đó *i* ∈ [*m,N*], m là độ dài dãy con. Ví dụ: nếu một chuỗi sự kiện là (1, 2, 3, 4, 5) và *m* = 4, thì các dãy con là (1, 2, 3, 4) và (2, 3, 4, 5). Sử dụng đầu ra bằng số lượng sự kiện duy nhất, LSTM trở thành một bộ phân loại đa số. Các *m* − 1



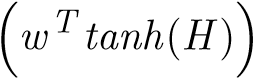
Hình 11. Kiến trúc Chú ý Bi-LSTM, x là đầu vào và y là đầu ra tại một thời điểm nhất định t.

đóng vai trò là đầu vào cho mỗi chuỗi con và giá trị cuối cùng được mã hóa một nóng và được sử dụng làm nhãn đầu ra. Sau khi được đào tạo, các dự đoán có xác suất lớn hơn ngưỡng xác suất *PT* được chọn làm đầu ra. Nếu sự kiện thực sự không nằm trong số các sự kiện được dự đoán, nó được phân loại là bất thường. *PT* cũng được điều chỉnh để đạt được Điểm F1 tốt nhất cho từng phương pháp.

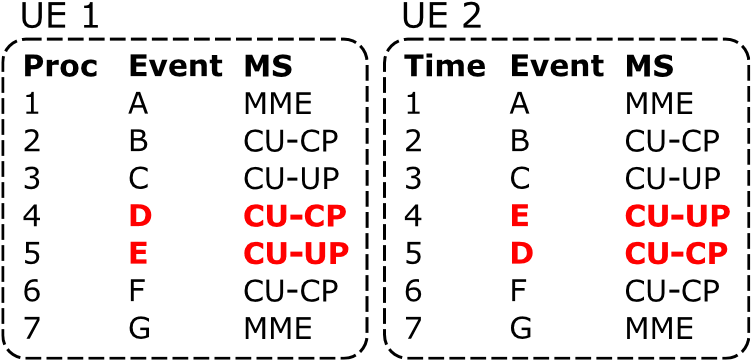
Để nâng cao hơn nữa độ chính xác của mô hình LSTM, các tế bào thần kinh ẩn trong lớp LSTM được chia thành hai phần, trong đó mỗi phần xử lý đường chuyền tiến và lùi riêng biệt. Bằng cách này, có thể nắm bắt đủ thông tin từ các chuỗi nhật ký từ cả hai hướng. Một mô hình như vậy thường được gọi là mô hình Bi-LSTM; xem Hình. 11.

Sau đó, lớp kích hoạt sử dụng luồng thông tin qua các lớp lùi và chuyển tiếp để cải thiện đầu ra dự đoán. Mô tả chi tiết hơn về cách LSTM học và ghi nhớ dữ liệu có thể được đọc trong [30].

Lớp chú ý gần đây đã được sử dụng trong một số biến thể để cải thiện độ chính xác phát hiện bất thường của RNN [10], [28], [38]. Việc nhúng nắm bắt mối quan hệ giữa các sự kiện và lớp chú ý cho phép các phương pháp LSTM có được cái nhìn sâu sắc từ toàn bộ chuỗi sự kiện. Lớp chú ý xây dựng mối tương quan giữa các sự kiện trong vectơ đầu ra LSTM ([*h1,h2,...,hT*]), trong đó *T* là độ dài trình tự. Nếu *H* là ma trận của tất cả các vectơ đầu ra, cơ chế chú ý sẽ tạo ra trọng số chú ý *α* và biểu diễn có trọng số ẩn *r*.

*α* = *mềm tối đa* (1)

*r* = *HαT*(2)



Hình 12. So sánh một phần của quy trình cho hai thiết bị người dùng (UE).

trong đó *H* ∈RdwxT, *dw* là chiều của các vectơ sự kiện, *w* và *wT* là một vectơ tham số được đào tạo và chuyển vị của nó. Một lớp softmax được áp dụng cho đầu ra chú ý để tạo vectơ xác suất cuối cùng.

LogRobust sử dụng từ điển được đào tạo trước để trích xuất các từ tiếng Anh phổ biến từ các sự kiện trong nhật ký hệ thống. Các vectơ của các từ sau đó được chuyển đổi thành một vectơ có kích thước cố định đại diện cho từng dòng sự kiện. Các nhật ký được tìm nạp trong quá trình đánh giá có cấu trúc hơn và việc trích xuất thông tin tin nhắn cho mỗi sự kiện rất dễ dàng. Thay vào đó, chúng tôi chọn chuyển đổi mỗi tin nhắn thành một số duy nhất trực tiếp.

Số lượng bất thường mà các mô hình LSTM tìm thấy phụ thuộc nhiều vào ngưỡng xác suất *PT*. Ngưỡng cần được điều chỉnh để tìm một số lượng dương tính đúng và giả phù hợp. Mỗi phương pháp được đánh giá bằng cách sử dụng điểm F1 xem xét các bất thường dương tính giả và dương tính thực, nhưng đây có thể không phải lúc nào cũng là lựa chọn tốt nhất trong môi trường sản xuất. Để chứng minh thêm tầm quan trọng của việc giữ dương tính giả ở mức thấp, hai biến thể chú ý Bi-LSTM với PT khác nhau được đánh giá.

1. *Phân tách vi dịch vụ LSTM:* Khi một số MS tham gia vào cùng một quy trình, nhiều luồng song song đôi khi có thể thay đổi thứ tự của các sự kiện. Sung. 12 minh họa sự khác biệt này.

Ví dụ được đơn giản hóa để hiển thị thời gian, số sự kiện và tên MS. Các sự kiện bắt nguồn từ cùng một quy trình nhưng có thứ tự khác nhau cho các sự kiện D-E. Lý do cho thứ tự hỗn hợp thường là các luồng hoạt động song song trong đó phản hồi thỉnh thoảng được nhận theo một thứ tự hơi khác. Các phương pháp học thứ tự trình tự, chẳng hạn như các phương pháp dựa trên LSTM, cần học cả thứ tự của  *ABCD-E* và ABCD. Trong một hệ thống như RAN, nơi nhiều luồng song song xử lý hàng nghìn thành phần, việc học hành vi từ các thành phần riêng lẻ có thể thuận lợi hơn so với hành vi kết hợp. Để khám phá xem liệu khả năng phát hiện bất thường có thể được cải thiện bằng cách sử dụng phân tách microservice hay không, một mô hình LSTM được đào tạo cho mỗi microservice. Một nhược điểm của cách tiếp cận này là số lượng các sự kiện nhật ký để học hỏi giảm. Nếu có quá ít sự kiện, các phương pháp học từ thứ tự trình tự sẽ không thể dự đoán các sự kiện trước đó.

1. *BoostLog:* Trong công trình trước đây của chúng tôi [17], chúng tôi đã phát triển một tập hợp các phương pháp LSTM AdaBoost có thể học hành vi phức tạp hơn các phương pháp LSTM riêng lẻ và đạt được điểm F1 cao hơn so với các phương pháp hiện đại trước đây. AdaBoost đã được chứng minh là phát hiện nhiều loại bất thường trong thử nghiệm 5G tiên tiến nhưng yêu cầu nhiều thời gian CPU hơn trong quá trình đào tạo vì một số mô hình LSTM được đào tạo cùng nhau. AdaBoost



Hình 13. Chế độ xem đối tượng - Cấu trúc của các đối tượng bị ảnh hưởng bởi các thủ tục.



Hình 14. Chế độ xem thủ tục - Hiển thị các thủ tục phụ trong mỗi thủ tục. Nó cũng có thể được nâng cao để hiển thị tương tác thành phần.

được đưa vào đánh giá để so sánh độ chính xác với các mô hình LSTM nâng cao sử dụng Bi-LSTM và các lớp chú ý.

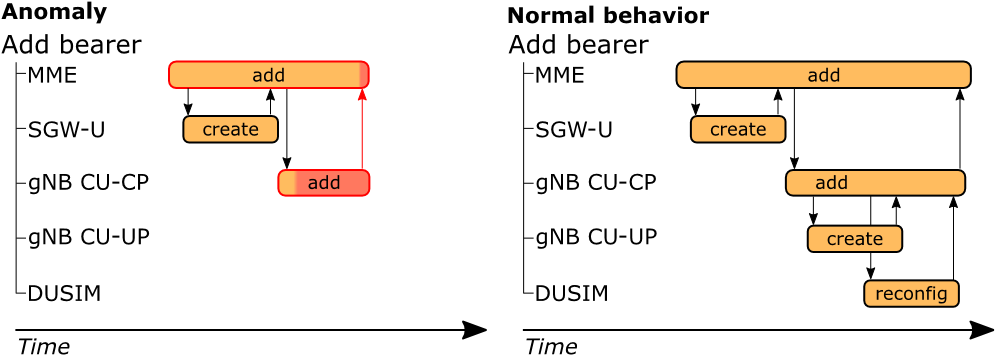
### C. Khả năng quan sát

Các trạng thái bên trong của một hệ thống có thể được hiểu tốt như thế nào từ đầu ra bên ngoài của nó thường được gọi là khả năng quan sát. Đầu ra bên ngoài, trong trường hợp này, là số liệu và nhật ký. Các chỉ số có thể cung cấp thông tin chi tiết về, ví dụ: mức sử dụng bộ nhớ, tải CPU hoặc Chỉ số Hiệu suất Chính (KPI), đo lường các tính năng đặc biệt của hệ thống. Chúng có thể được sử dụng để đo lường hiệu suất của hệ thống nhưng hiếm khi đưa ra câu trả lời cho lý do tại sao hệ thống hoạt động theo một cách nhất định. Thay vào đó, nhật ký là nơi đầu tiên để xem xét khi có sự cố nào đó. Một vấn đề được nêu bật trong Phần III-B là các sự kiện có thể khác nhau đáng kể tùy thuộc vào nhà phát triển và thành phần. Một vấn đề khác là các nhà phát triển thường thiết bị phần mềm từ góc độ thành phần và quên đi phạm vi lớn hơn mà các đối tượng và thủ tục là quan trọng.

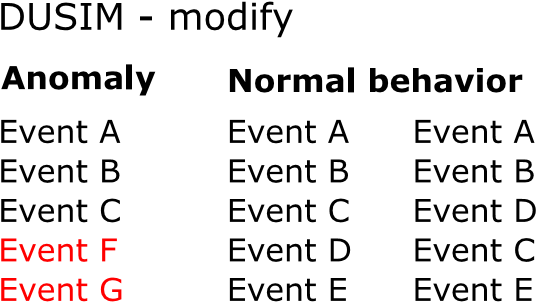
Các hướng dẫn theo dõi mới cho phép tách các đối tượng và quy trình và giúp hình dung cách các bộ phận hệ thống hoạt động cùng nhau. Các điểm bất thường cũng được trình bày bằng cách sử dụng các bước trong Phần IV-B1 để nâng cao khả năng quan sát hơn nữa. Việc trực quan hóa được thực hiện từ góc độ từ trên xuống, nơi các thủ tục ảnh hưởng đến các thành phần trong hệ thống. Theo quan điểm này, cũng có thể thấy một sự bất thường xảy ra trong quy trình nào; xem Hình. 13.

Ví dụ là từ một thử nghiệm 5G, nơi các quy trình và danh sách thiết bị người dùng (UE) được hiển thị trong chế độ xem dòng thời gian. Màu đỏ cho biết nơi xảy ra bất thường trong quy trình và trình khắc phục sự cố có thể dễ dàng điều hướng sâu hơn để xem các thành phần bị ảnh hưởng như thế nào, xem Hình. 14.

Chế độ xem quy trình trình bày tín hiệu giữa các thành phần và cách các thủ tục phụ đóng góp vào thủ tục. Khi xảy ra bất thường, có thể dễ dàng xác định các quy trình phụ không thành công và so sánh chúng với các quy trình bình thường; xem Hình. 15.



Hình 15. Chế độ xem thủ tục - So sánh các quy trình bất thường và bình thường.



Hình 16. Chế độ xem quy trình phụ - So sánh chuỗi sự kiện cho hành vi bình thường và bất thường.

Đối với mỗi quy trình phụ, có thể đi sâu hơn và so sánh các nhật ký về hành vi bình thường và bất thường; xem Hình. 16.

Ví dụ này minh họa cách so sánh các quy trình phụ để phát hiện sự khác biệt ở cấp độ sự kiện.

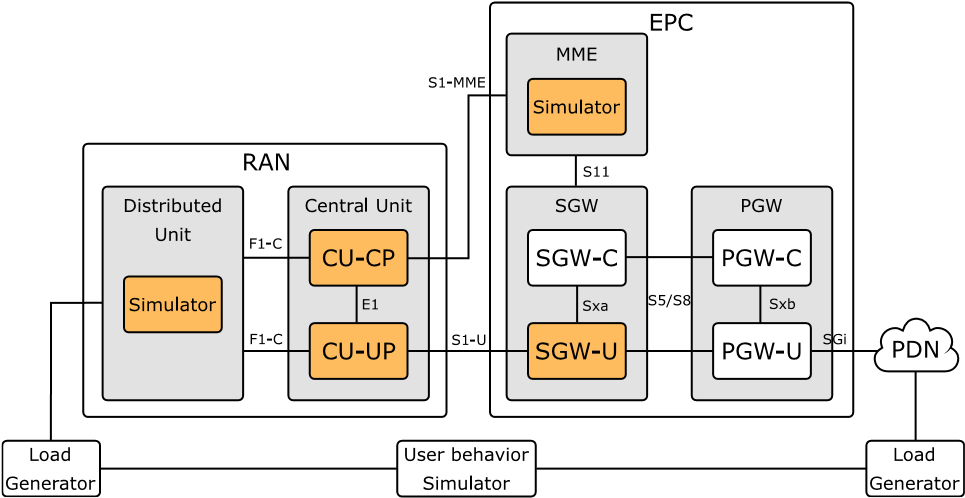
### D. Chỉ số đánh giá

Mỗi bất thường được phát hiện cần được các nhà phát triển phân tích và điều quan trọng là phải giữ số lượng dương tính giả (FP) thấp trong khi xác định dương tính thực (TP). Để xem xét cả TP và FP, các phương pháp được điều chỉnh để tối đa hóa điểm F1. Mặc dù, các giá trị độ chính xác, khả năng thu hồi, FP và TP cũng được trình bày trong kết quả. Tất cả các phương pháp phân tích từng trình tự UE và đếm số lượng trình tự bất thường. Thời gian đào tạo từng phương pháp cũng được đánh giá.

Một khía cạnh quan trọng đối với các trình khắc phục sự cố RAN là hiểu hành vi của hệ thống và nhanh chóng xác định thời điểm và địa điểm xảy ra bất thường. Để đo lường khả năng quan sát của giường thử nghiệm 5G, năm câu hỏi được xác định:

* *Thủ tục thất bại* - Thủ tục nào không thành công trong trường hợp thử nghiệm? Điều này cho thấy trình khắc phục sự cố hiểu loại kịch bản nào xảy ra khi bất thường xảy ra.
* *Quy trình lịch sử* - Các thủ tục trước đó có thể thay đổi trạng thái của đối tượng được thử nghiệm. Hiểu các quy trình trước đây có thể hỗ trợ người khắc phục sự cố trong phân tích nguyên nhân gốc rễ.
* *Microservice thất bại* - Khi một số MS làm việc cùng nhau để hoàn thành một quy trình, MS nào là một phần của quy trình và MS nào có nhiều khả năng gây ra bất thường?
* *Sự kiện bình thường* - Điều gì nên xảy ra thay vì sự bất thường? Khi kịch bản bình thường được hiểu, hành vi bất thường có thể được so sánh với bình thường.
* *Loại bất thường* - Loại bất thường có thể được xác định nhanh như thế nào? Hiểu được sự khác biệt giữa hành vi bình thường và bất thường có thể rất quan trọng để hiểu đầy đủ vấn đề.

Khi trình khắc phục sự cố tìm kiếm các điểm bất thường, họ nên cố gắng trả lời từng câu hỏi càng nhanh càng tốt. Các



Hình 17. Tổng quan về kiến trúc của nền thử nghiệm 5G.

Thời gian cần thiết để trả lời các câu hỏi sau đó được sử dụng để tính điểm *khả năng quan sát* nằm trong khoảng từ 0 đến 10 điểm. Mười điểm được cho nếu câu hỏi được trả lời trong vòng một phút, chín điểm trong vòng hai phút, v.v., giảm xuống 0 điểm khi cần hơn mười phút. Vì kinh nghiệm của trình khắc phục sự cố cũng xác định mức độ hiểu hệ thống, nên sự kết hợp giữa trình khắc phục sự cố có kinh nghiệm và người mới làm quen là một phần của đánh giá.

## V. ĐÁNH GIÁ

### A. Thiết lập testbed và thu thập dữ liệu

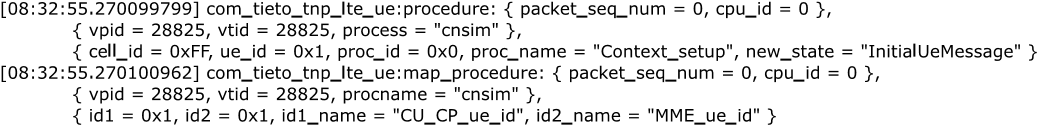
Để đánh giá xem các hướng dẫn dựa trên thủ tục có thể cải thiện khả năng phát hiện và quan sát bất thường hay không; thiết bị phần mềm hiện có của một giường thử nghiệm 5G tiên tiến đã được cập nhật để tuân theo các hướng dẫn dựa trên quy trình. Giường thử nghiệm được phát triển bởi TietoEVRY[[2]](#footnote-2) và là một hệ thống phân tán mở rộng để kiểm tra hiệu suất phần cứng và các khái niệm đám mây khác nhau cho 5G RAN trong tương lai. Một thành phố mô phỏng tạo ra giao thông khi cư dân gọi điện thoại và di chuyển xung quanh thành phố. Thiết bị người dùng và mô phỏng mạng lõi sử dụng việc sử dụng di động mô phỏng để tạo lưu lượng cho các trạm gốc; xem Hình. 17.

Phạm vi của công việc này được giới hạn ở năm quy trình chính và bao gồm phân tích thông tin thủ tục từ các trình mô phỏng, CU-CP, CU-UP và SGW-U. Tất cả các quy trình đều ảnh hưởng đến UE và danh tính của UE được bao gồm trong thông tin gỡ lỗi mới. Khả năng gỡ lỗi của nền thử nghiệm 5G đã được tăng từ cấp 4 lên 5 bằng cách cập nhật 199 thiết bị phần mềm. Việc cập nhật và xác minh thiết bị đo lường mất 40 giờ. Một trường hợp kiểm tra lưu lượng truy cập chuyên sâu chạy trong năm phút để thu thập nhật ký hệ thống với thiết bị đo lường được cập nhật. Nhật ký chứa trung bình 863.153 sự kiện từ 5.750 UE duy nhất, với 150 sự kiện trên mỗi UE, xem Bảng I. Tất cả các trình tự bắt đầu với quy trình Thiết lập và kết thúc bằng Phát hành hoặc Không hoạt động. Thêm E-RAB và Sửa đổi có thể xảy ra bất kỳ số lần nào ở giữa. Sung. 18 trình bày một ví dụ về ánh xạ thủ tục. Dữ liệu được cung cấp công khai tại github.[[3]](#footnote-3) Thông tin thủ tục bổ sung chứa trung bình 30 ký tự và tăng tổng kích thước nhật ký lên 13%. Một đánh giá bổ sung đã được tiến hành để đo lường xem kích thước nhật ký tăng lên có ảnh hưởng đến việc sử dụng bộ nhớ hoặc CPU hay không, nhưng không ảnh hưởng đến

## BẢNG I

QUY TRÌNH TRONG TEST SUITE

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |



Hình 18. Ví dụ về thủ tục và sự kiện ánh xạ trong nhật ký hệ thống bằng cách sử dụng các nguyên tắc theo dõi mới.

sự khác biệt thống kê đã được tìm thấy. Chúng tôi cũng không thể phát hiện ra bất kỳ sai lệch nào về số lượng thủ tục được xử lý do thông tin thủ tục bổ sung.

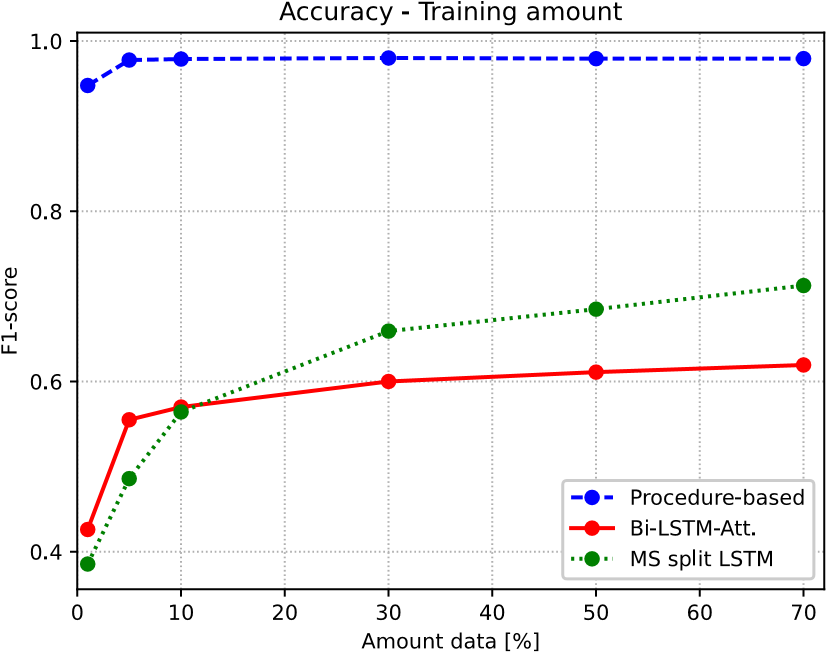
Dựa trên kinh nghiệm của chúng tôi với testbed và RAN thực của nhà cung cấp, chúng tôi đã thấy rằng các kịch bản thất bại có thể được phát hiện trong nhật ký hệ thống bằng cách thiếu, thêm hoặc thay đổi thứ tự của các sự kiện. Sự chậm trễ hoặc treo của thủ tục cũng có thể được phát hiện bằng cách sử dụng dấu thời gian của mỗi sự kiện [27], nhưng khía cạnh thời gian không được bao gồm trong công việc này. Thay vì thêm các điểm bất thường mới có thể tạo ra những thay đổi này trong nhật ký hệ thống, các điểm bất thường được đưa trực tiếp vào nhật ký. Ưu điểm của cách tiếp cận này là hành vi bất thường có thể được kiểm soát chi tiết hơn và nhiều bất thường có thể được kiểm tra. Các thay đổi được giới thiệu ngẫu nhiên giữa các UE và bao gồm hầu hết các điểm bất thường trước đây được thấy trong các hệ thống tương tự. Mười một loại dị thường, được định nghĩa trong Bảng III, được giới thiệu trong nhật ký 100 lần mỗi loại. Chúng tôi đã chọn 70% dữ liệu để đào tạo và đưa ra các điểm bất thường trong dữ liệu còn lại. Tổng cộng, có 4.025 chuỗi UE trong dữ liệu đào tạo, trong đó 3.076 có một chuỗi sự kiện duy nhất. Dữ liệu thử nghiệm chứa 1.725 trình tự UE bình thường và 1.100 trình tự UE bất thường.

### B. Thiết lập thử nghiệm

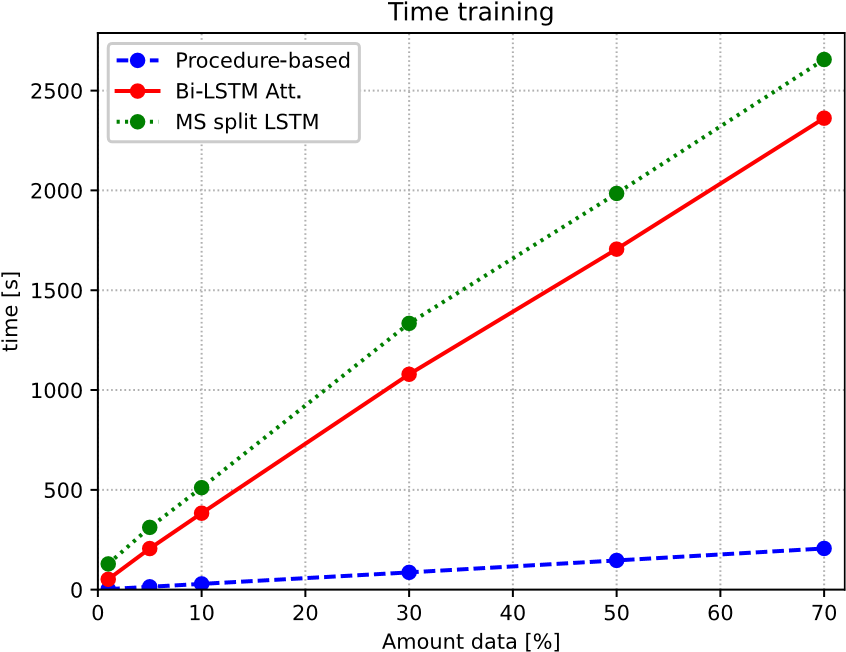
Các thông số kỹ thuật phần cứng của thử nghiệm 5G không thể được tiết lộ do bí mật thương mại, nhưng LTTng phiên bản 2.12 đã được sử dụng để thiết bị cho hệ thống. Tất cả các phương pháp đều sử dụng hệ điều hành Ubuntu 18.04.5 với một CPU Intel(R) Xeon(R) @ bộ xử lý 2.20GHz và bộ nhớ 13GB khi đào tạo và phát hiện bất thường. Xác nhận chéo 5 lần đảm bảo tính ổn định của kết quả trong tất cả các thí nghiệm.

### C. Phát hiện bất thường

Ba mô hình LSTM cơ bản được đánh giá cùng với mô hình dựa trên thủ tục mới. Phương pháp Bi-LSTM-Attention được đào tạo trong 30 kỷ nguyên, sử dụng kích thước lô tối thiểu là 200 và *PT* = 1e − 3. Hai lớp LSTM sử dụng 128 tế bào thần kinh và lớp đầu vào sử dụng giá trị m = 5. Phương pháp LSTM phân tách MS có cùng cấu hình và đào tạo một LSTM cho mỗi trong số năm vi dịch vụ. Mô hình AdaBoost đã sử dụng 10 bộ phân loại LSTM khác nhau, từ 2-3 lớp ẩn, 32-200 tế bào thần kinh và *m* = 5. Không có siêu tham số nào để điều chỉnh cho phương pháp dựa trên quy trình mới có thể học hành vi nhanh chóng với dữ liệu đào tạo ít ỏi. Để điều tra cách thức



Hình 19. Độ chính xác cho lượng dữ liệu đào tạo ngày càng tăng.



Hình 20. Thời gian đào tạo cần thiết cho lượng dữ liệu đào tạo ngày càng tăng.

lượng dữ liệu đào tạo ảnh hưởng đến độ chính xác, điểm F1 được so sánh khi lượng dữ liệu tăng lên; xem Hình. 19.

Trong khi phương pháp dựa trên thủ tục đạt mức tối đa ở mức sử dụng dữ liệu năm phần trăm, các phương pháp cơ bản tiếp tục cải thiện tới 70% mức sử dụng dữ liệu. Thời gian cần thiết để đào tạo trên dữ liệu lịch sử cũng là một yếu tố thiết yếu, và thời gian đào tạo tăng tuyến tính cho tất cả các phương pháp; xem Hình. 20. Thời gian đào tạo của phương pháp Adaboost quá lớn để đưa vào đánh giá này, và đây cũng là lý do tại sao nó không được nhìn thấy trong Hình. 19, Hình. 20, và Hình. 21.

Chế độ xem chi tiết về độ chính xác, thời gian đào tạo và các điểm bất thường được tìm thấy được trình bày trong Bảng II, trong đó 70% dữ liệu được sử dụng để đào tạo.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BẢNG II  DỮ LIỆU ĐÁNH GIÁ CHO TỪNG PHƯƠNG PHÁP   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  |   BẢNG III  CÁC LOẠI BẤT THƯỜNG ĐƯỢC GIỚI THIỆU VÀ TÌM THẤY CHO TỪNG PHƯƠNG PHÁP   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |  |  |  |  |  |  |  | |

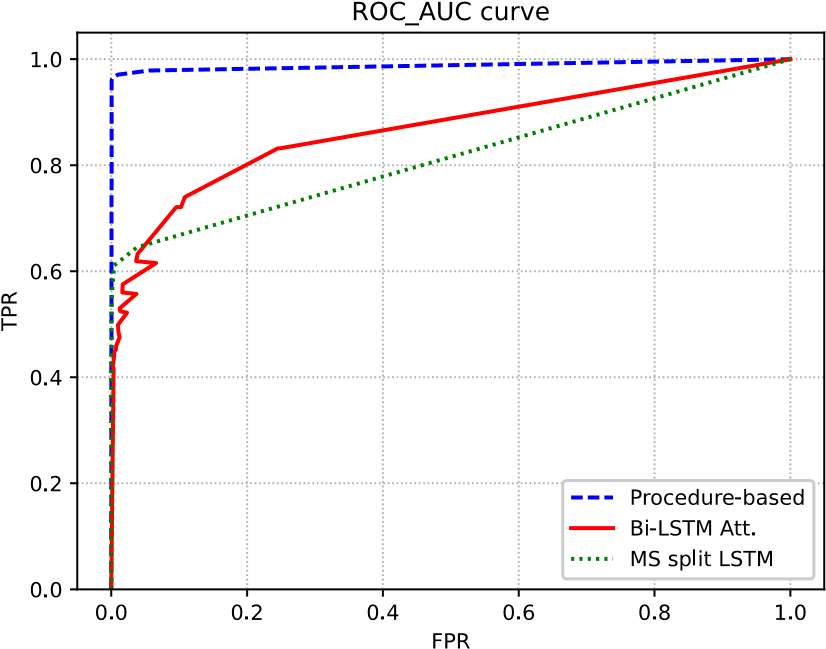
Các phương pháp cơ sở có độ chính xác cao khi dự đoán các sự kiện đơn lẻ và độ chính xác xác nhận lớn hơn 0,9 cho cả ba phương pháp cơ sở khi kết thúc đào tạo. Độ chính xác này liên quan đến dự đoán các sự kiện đơn lẻ, nhưng độ chính xác sẽ thấp hơn nhiều khi dự đoán toàn bộ chuỗi sự kiện, trung bình, bao gồm 150 sự kiện. Phương pháp cơ bản có điểm F1 cao nhất đã tìm thấy hơn một nửa số điểm bất thường và nhiều trường hợp FP. Giữ số lượng trường hợp FP thấp là điều cần thiết đối với những người khắc phục sự cố, những người không muốn lãng phí thời gian phân tích các trình tự bình thường. Vì các trường hợp FP rất nhiều, một phiên bản khác của phương pháp chú ý Bi-LSTM với *PT* = 1e−8 được đánh giá. Các phương pháp phân tách AdaBoost và MS sử dụng cùng một ngưỡng thấp. Đáng chú ý là độ chính xác tăng lên khi một phương pháp chú ý Bi-LSTM được sử dụng cho mỗi MS. Ưu điểm lớn của cách tiếp cận dựa trên thủ tục cũng là rõ ràng; hầu hết tất cả các dị thường đều được tìm thấy và chỉ có một trường hợp FP được báo cáo.

Ngoài ra còn có một sự cải thiện đáng kể về thời gian đào tạo, trong đó các phương pháp học sâu cần ít nhất 2.360 giây thời gian CPU và dựa trên quy trình chỉ cần 210 giây. Bảng III trình bày phân tích chi tiết hơn về các bất thường mà mỗi phương pháp có thể phát hiện.

Ảnh hưởng của LSTM phân tách MS rõ ràng hơn trong bảng này, nơi nhiều điểm bất thường được phát hiện đối với hầu hết các loại bất thường. Tuy nhiên, một nhược điểm là nó không thể xác định một số điểm bất thường mà một phần lớn các sự kiện bị thiếu. Một nhược điểm khác là phương pháp LSTM phân tách MS không thể phát hiện các bất thường với các sự kiện đầu vào không đủ. Các phương pháp LSTM cũng có thể dễ dàng phát hiện các sự kiện bổ sung hơn so với các sự kiện bị thiếu. Đây là một phần của bản chất LSTM, trong đó các sự kiện xảy ra gần sau sự kiện dự đoán có xác suất cao. Phương pháp dựa trên thủ tục có thể dễ dàng phát hiện những thay đổi trong nhóm quy trình con và các sự kiện mới. Tuy nhiên, một số trong số năm loại bất thường đầu tiên dường như không được phát hiện và cần phân tích thêm. Sau đó, người ta phát hiện ra rằng tất cả những dị thường này đã được đưa vào sao cho trình tự kết quả giống như kịch bản bình thường được thấy trong quá trình huấn luyện. Phương pháp dựa trên quy trình rất mạnh mẽ trong quá trình xác nhận chéo của chúng tôi và đường cong ROC-AUC cho tất cả các kết quả có thể được nhìn thấy trong Hình. 21.

### D. Khả năng quan sát

Trong việc đánh giá điểm *khả năng quan sát* , có hai khía cạnh của các phương pháp quan trọng hơn các phương pháp khác; Cách các điểm bất thường và hành vi được trình bày cho người dùng cuối và số lượng trường hợp FP/TP mà mỗi phương pháp tìm thấy. Mỗi trường hợp FP cần thêm thời gian trong việc phân tích và nếu không có trường hợp TP thì rất khó để xác định hành vi lệch lạc. Vì tất cả các phương pháp cơ bản đều trình bày hành vi và bất thường theo cùng một cách,



Hình 21. Đường cong ROC-AUC cho các phương pháp LSTM phân tách dựa trên thủ tục, Bi-LSTM và MS.

chỉ có sự chú ý BI-LSTM với ngưỡng thấp được chọn do số lượng trường hợp FP thấp. Để đưa ra đánh giá công bằng giữa phương pháp đường cơ sở và phương pháp dựa trên thủ tục, phân tích được thực hiện cho trình tự mà cả hai phương pháp đều tìm thấy trường hợp TP. Một nhóm thử nghiệm gồm 4 nhà phát triển có kinh nghiệm hỗn hợp đã được chọn để trả lời các câu hỏi được nêu trong Phần IV-D. Một điểm bất thường ngẫu nhiên được chọn từ mỗi loại bất thường và tất cả các nhà phát triển đều trả lời năm câu hỏi về khả năng quan sát từ Phần IV-D càng nhanh càng tốt. Người khắc phục sự cố được yêu cầu trả lời chúng theo thứ tự được nêu trong Phần IV-D. Mỗi người khắc phục sự cố đã làm bài kiểm tra hai lần với các điểm bất thường khác nhau và giá trị trung bình với một độ lệch chuẩn được đưa ra cho mỗi điểm số và phương pháp trong Bảng IV.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BẢNG IV  ĐIỂM KHẢ NĂNG QUAN SÁT CHO TỪNG PHƯƠNG PHÁP (TRÌNH KHẮC PHỤC SỰ CỐ CÓ KINH NGHIỆM)   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  | |  |  |  |  |  |  | |  |  |  |  |  |  | |

Tất cả các trình khắc phục sự cố đều cải thiện điểm *khả năng quan sát* bằng cách sử dụng phương pháp dựa trên thủ tục. Khả năng hình dung và so sánh các thủ tục đã làm tăng đáng kể sự hiểu biết của hệ thống. Nếu không có sự hỗ trợ trực quan trong phương pháp Bi-LSTM-attention, các trình khắc phục sự cố đã tìm kiếm thủ công các mẫu tương tự trong các UE khác và so sánh chúng theo từng sự kiện để phân biệt các trình tự bình thường và bất thường. Một số trình khắc phục sự cố cũng phải tiến hành tài liệu và mã để xác định loại quy trình nào đã xảy ra trong nhật ký hệ thống. Kết quả cũng cho thấy phương pháp dựa trên thủ tục có thể giúp cung cấp sự hiểu biết nhanh chóng về hệ thống, trong đó ba câu hỏi đầu tiên có thể được trả lời trong vòng ba phút.

## VI. THẢO LUẬN VÀ HẠN CHẾ

Công trình này mô phỏng những bất thường mà chúng ta đã thấy trong suốt 20 năm làm việc trong ngành viễn thông. Tuy nhiên, các điểm bất thường được đưa ra không dành riêng cho viễn thông, vì chúng đại diện cho một chuỗi sự kiện bất thường trong nhật ký hệ thống. Chúng tôi đã quan sát cách các hướng dẫn theo dõi và phương pháp dựa trên quy trình của chúng tôi có thể cải thiện khả năng phát hiện bất thường trong môi trường sản xuất Cloud RAN [39]. Chúng tôi tin rằng nó cũng có thể cải thiện khả năng phát hiện và quan sát bất thường trong nhiều hệ thống phức tạp khác.

Các ứng dụng thực sự đi kèm với cả sự kết hợp đơn giản hơn hoặc phức tạp hơn của các sự kiện dị thường trong nhật ký so với những sự kiện được mô phỏng trong tác phẩm này. Mặc dù vậy, chúng tôi cảm thấy tự tin rằng phương pháp dựa trên quy trình sẽ đạt được độ chính xác cao trong môi trường sản xuất bằng cách quan sát từ hiệu suất cao của nó trong các bất thường mô phỏng khác nhau và giải pháp Cloud RAN tích hợp.

Chúng tôi đã chọn so sánh phương pháp dựa trên quy trình mới của mình với các phương pháp LSTM hiện đại gần đây trong công trình này. Các phương pháp học sâu khác có thể đã vượt qua các phương pháp cơ sở này về độ chính xác hoặc thời gian CPU ở một mức độ nào đó, nhưng rất có thể nó sẽ không thay đổi kết luận của chúng tôi. Chúng tôi đã tìm cách điều tra xem liệu việc đo lường ứng dụng với tư duy lớn hơn, có hệ thống hóa có thể cải thiện khả năng quan sát và phát hiện bất thường dựa trên ML hay không. Các hướng dẫn theo dõi mới có thể hướng dẫn các nhà phát triển đi đúng hướng và hỗ trợ các nhà nghiên cứu khác phát triển các phương pháp có thể tăng cường hơn nữa khả năng phát hiện và quan sát bất thường.

Một *điểm khả năng quan sát mới* đã được xác định để đo lường mức độ hiểu của người khắc phục sự cố về các quy trình đang diễn ra và liệu có thể phân biệt giữa hành vi bình thường và bất thường hay không. Cũng có thể có các câu hỏi cụ thể về hệ thống thử nghiệm, nhưng chúng tôi đã giới hạn điều tra của mình trong các câu hỏi chung hơn liên quan đến các hệ thống phân tán lớn khác. Nghiên cứu có thể bao gồm nhiều trình khắc phục sự cố hơn để đạt được độ chắc chắn thống kê cao hơn trong *điểm quan sát*. Tuy nhiên, kết quả là thỏa mãn và tỷ lệ các nhà phát triển mới làm quen sẽ tăng lên nếu một nhóm lớn hơn các nhà phát triển đã tiến hành thử nghiệm. Điều đáng nói là các nhà phát triển nghĩ rằng cả hai phương pháp đều nâng cao khả năng quan sát của hệ thống. Hầu hết các hệ thống không có phân tích hành vi dựa trên ML và nếu không có LSTM hoặc hỗ trợ dựa trên quy trình, các nhà phát triển phải tìm kiếm qua nhật ký để phát hiện các điểm bất thường theo cách thủ công.

Các phương pháp trong công việc này sử dụng học có giám sát để học hành vi từ các trình tự không có bất thường. Các phương pháp cũng có thể không được giám sát nếu các điểm bất thường được giữ ở mức thấp. Trong trường hợp này, phương pháp dựa trên thủ tục sẽ cần một siêu tham số mới để điều chỉnh xác suất mà một thủ tục hoặc thủ tục phụ xuất hiện và sau đó lọc ra những thủ tục xảy ra không thường xuyên hơn. Các trình tự bình thường hiếm khi xảy ra có thể làm tăng tỷ lệ FP. Tuy nhiên, chúng tôi tin rằng sẽ có lợi nếu sử dụng cách tiếp cận dựa trên thủ tục vì số lượng trình tự duy nhất có thể được giữ ở mức thấp khi các trình tự được tách ra bằng cách sử dụng thủ tục, đối tượng và thành phần.

Đánh giá một số mô hình LSTM đòi hỏi hàng giờ đào tạo và thử nghiệm cho mỗi thí nghiệm và chúng tôi chỉ chọn xác minh kết quả của mình bằng cách sử dụng xác nhận chéo 5 lần. Những lợi thế của học thủ tục trong RAN vẫn còn rõ ràng và không có sự xác nhận nào nữa có thể thay đổi thực tế đó.

Các hướng dẫn theo dõi được đề xuất có thể dễ dàng được thực hiện để cải thiện nền tảng thử nghiệm 5G, nhưng có thể có các hệ thống phân tán lớn mà quy trình không thể được xác định cho một số sự kiện. Những trường hợp như vậy có thể là dấu hiệu của thiết kế hoặc kiến trúc xấu, và có lẽ các hệ thống như vậy sẽ được hưởng lợi từ việc thiết kế lại và các hướng dẫn theo dõi mới.

Thông tin quy trình bổ sung được lưu trữ trong nhật ký hệ thống làm tăng kích thước nhật ký và có thể là vấn đề đối với các hệ thống đã gặp sự cố với lưu trữ nhật ký. Đánh giá trong công việc này sử dụng LTTng để lưu trữ dữ liệu, được biết đến là một khung theo dõi hiệu quả cao. Tuy nhiên, có thể có vấn đề với việc lưu trữ dữ liệu nếu một số lượng lớn các thủ tục được xử lý cùng một lúc. Chúng tôi khuyên những người được truyền cảm hứng từ công việc của chúng tôi trước tiên hãy đánh giá xem hệ thống có thể xử lý lượng dữ liệu bổ sung được thêm vào bởi thông tin thủ tục hay không.

Phương pháp dựa trên thủ tục mới của chúng tôi đã được tích hợp và đánh giá các đường ống hiện có cho 5GcloudRAN, và kết quả rất hứa hẹn. Độ chính xác của phát hiện bất thường đã tăng lên và ít phát hiện bất thường dương tính giả hơn. Tuy nhiên, vẫn cần cải thiện hỗ trợ trực quan hóa để giúp người dùng cuối hiểu nơi xảy ra bất thường trong quy trình.

Phương pháp dựa trên thủ tục mới của chúng tôi cùng với hướng dẫn theo dõi được cải thiện, đã đặt nền móng cho luận án thạc sĩ mới được xuất bản tại Đại học Umeå[[4]](#footnote-4) [39]. Luận án kết luận rằng phương pháp dựa trên thủ tục đang hoạt động tốt trong các đường ống tích hợp liên tục Cloud RAN và có thể cải thiện độ chính xác phát hiện bất thường. Điều này củng cố niềm tin của chúng tôi rằng chúng tôi cần cấu trúc nhật ký theo những cách chung chung hơn để giúp cả người dùng cuối và phương pháp học máy. Sự hợp tác trong tương lai với các đối tác trong ngành sẽ khám phá xem liệu có bất kỳ bất lợi nào được dự đoán với phương pháp dựa trên thủ tục mới này hay không.

Các hướng dẫn được nêu trong công việc này được sử dụng để thêm bối cảnh thủ tục cho một nền thử nghiệm 5G phân tán lớn. Dựa trên kinh nghiệm thiết kế các hệ thống lớn khác, chúng tôi tin rằng các hướng dẫn của chúng tôi có thể cải thiện khả năng quan sát và phát hiện bất thường cho nhiều hệ thống khác. Tuy nhiên, điều quan trọng là thông tin thủ tục phải được thêm vào với sự hợp tác của các chuyên gia hệ thống và các nhà phát triển biết các thủ tục ảnh hưởng đến hệ thống như thế nào. Sau khi thiết bị đo lường được thực hiện, các phương pháp dựa trên quy trình có thể tự động tìm hiểu hành vi của các thủ tục được hệ thống hóa. Điều này cũng giúp các nhà phát triển và người khắc phục sự cố tìm hiểu hành vi và ánh xạ nó với tài liệu của hệ thống.

## VII. KẾT LUẬN VÀ CÔNG VIỆC TRONG TƯƠNG LAI

Nghiên cứu này trình bày cách hướng dẫn theo dõi dựa trên quy trình có thể cải thiện khả năng phát hiện và quan sát bất thường dựa trên ML trong một nền thử nghiệm 5G phân tán tiên tiến. Bối cảnh thủ tục bổ sung cho phép phương pháp dựa trên thủ tục mới tìm hiểu hành vi của các phần nhỏ hơn của hệ thống và sau đó kết hợp chúng để nhận ra tốt hơn cách tất cả các phần hoạt động cùng nhau. Kết quả thật đáng kinh ngạc: trong khi các phương pháp học sâu hiện đại trước đây cần hàng giờ đào tạo để phát hiện một số điểm bất thường, phương pháp dựa trên quy trình có thể, trong vòng vài phút, phát hiện tất cả các điểm bất thường. Thông tin thủ tục cũng tăng cường khả năng quan sát của hệ thống, nơi tất cả các trình khắc phục sự cố hiểu rõ hơn về hành vi của nền thử nghiệm 5G phân tán lớn và xác định các điểm bất thường nhanh hơn.

Chúng tôi tin rằng công việc này có thể cung cấp một số thông tin chi tiết về cách thiết bị đo lường nên được thực hiện để cho phép các phương pháp học máy tìm hiểu hành vi của các hệ thống lớn và cải thiện khả năng quan sát dễ dàng hơn. Trong nghiên cứu trong tương lai, chúng tôi sẽ điều tra xem liệu các hướng dẫn này có thể hỗ trợ phát hiện nguyên nhân gốc rễ của sự bất thường trong các hệ thống phân tán lớn hay không.

THỪA NHẬN

Cảm ơn TietoEvry đã tài trợ một phần của nghiên cứu này.

## THAM KHẢO

1. Z. Li, T.-H. Chen, và W. Shang, "Chúng ta sẽ ghi nhật ký ở đâu? Nghiên cứu và đề xuất ghi vị trí trong các khối mã," trong *Proc. 35th IEEE/ACM Int. Conf. Autom. Mềm mại. Eng.*, 2020, trang 361–372.
2. J. Zhu và cộng sự, "Các công cụ và điểm chuẩn để phân tích cú pháp nhật ký tự động," trong *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng. Softw. Eng. Thực hành.*, 2019, trang 121–130.
3. J. Soldani và A. Brogi, "Phát hiện bất thường và phân tích nguyên nhân gốc rễ lỗi trong các ứng dụng đám mây dựa trên dịch vụ (MICRO): Một cuộc khảo sát," *ACM Comput. Khảo sát*, tập 55, số 3, trang 1–39, 2022.
4. B. Li và cộng sự, "Tận hưởng khả năng quan sát của bạn: Một cuộc khảo sát công nghiệp về theo dõi và phân tích microservice," *Softw theo kinh nghiệm. Eng.*, tập 27, số 1, trang 1–28, 2022.
5. N. Bosch và J. Bosch, "Ghi nhật ký phần mềm cho máy học," 2020, *arXiv:2001.10794*.
6. C. Cassé, P. Berthou, P. Owezarski và S. Josset, "Mô hình dựa trên theo dõi để xác định các nút thắt cổ chai trong các ứng dụng phân phối vật lý," trong *Proc. Int. Conf. Inf. Netw.*, 2022, trang 226–231.
7. C. Cassé, P. Berthou, P. Owezarski và S. Josset, "Sử dụng theo dõi phân tán để xác định thành phần tài nguyên không hiệu quả trong các ứng dụng đám mây," trong *Proc. IEEE 10th Int. Conf. Cloud Netw.*, 2021, trang 40–47.
8. X. Yu, P. Joshi, J. Xu, G. Jin, H. Zhang và G. Jiang, "CloudSEER: Giám sát quy trình làm việc của cơ sở hạ tầng đám mây thông qua nhật ký xen kẽ," *ACM SIGARCH Comput. Kiến trúc sư. Tin tức*, tập 44, số 2, trang 489–502, 2016.
9. Z. S. Ageed và cộng sự, "Khảo sát toàn diện về các phương pháp khai thác dữ liệu lớn trong hệ thống đám mây," *Qubahan Acad. J.*, tập 1, số 2, trang 29–38,

2021.

1. X. Zhang và cộng sự, "Phát hiện bất thường dựa trên nhật ký mạnh mẽ trên dữ liệu nhật ký không ổn định," trong *Proc. Cuộc họp chung ACM lần thứ 27 Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2019, trang 807–817.
2. A. Bento, J. Correia, R. Filipe, F. Araujo và J. Cardoso, "Phân tích tự động truy vết phân tán: Thách thức và hướng nghiên cứu," *J. Grid Comput.*, Tập 19, Số 1, trang 1–15, 2021.
3. S. He, P. He, Z. Chen, T. Yang, Y. Su và MR Lyu, "Một cuộc khảo sát về phân tích nhật ký tự động cho kỹ thuật độ tin cậy," *ACM Comput. Khảo sát*, tập 54, số 6, trang 1–37, 2021.
4. M. Landauer, S. Onder, F. Skopik và M. Wurzenberger, "Học sâu để phát hiện bất thường trong dữ liệu nhật ký: Một cuộc khảo sát," 2022, *arXiv:2207.03820*.
5. R. B. Yadav, P. S. Kumar và S. V. Dhavale, "Một cuộc khảo sát về phát hiện bất thường log bằng cách sử dụng deep learning," trong *Proc. 8th Int. Conf. Rel. Infocom Technol. Optim. (ICRITO)*, 2020, trang 1215–1220.
6. X. Zhao, Z. Jiang và J. Ma, "Khảo sát phát hiện bất thường sâu cho nhật ký hệ thống," trong *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, 2022, trang 1–8.
7. M. Du, F. Li, G. Zheng và V. Srikumar, "DeepLog: Phát hiện và chẩn đoán bất thường từ nhật ký hệ thống thông qua học sâu," trong *Proc. ACM Special Interest Group Security Audit Control Conf. Comput. Công xã. An ninh*, 2017, trang 1285–1298.
8. T. Sundqvist, M. Bhuyan, J. Forsman và E. Elmroth, "Tăng cường học tập hợp để phát hiện bất thường trong 5G RAN," trong *Proc. IFIP Int. Conf. Artif. Thông minh. Ứng dụng Innov.*, 2020, trang 15–30.
9. W. Gan, JCW. Lin, HC Chao, và J. Zhan, "Khai thác dữ liệu trong môi trường phân tán: Một cuộc khảo sát," *Wiley Interdiscipl. Rev. Data Min. Knowl. Đĩa.*, tập 7, số 6, 2017, Điều số e1216.
10. R. Fonseca, G. Porter, RH Katz và S. Shenker, "X-Trace: Khung theo dõi mạng phổ biến," trong *Proc. 4th USENIX Symp. Netw. Hệ thống thiết kế thực hiện.*, 2007, tr. 9.
11. Y. Gan và cộng sự, "SEER: Tận dụng dữ liệu lớn để điều hướng sự phức tạp của việc gỡ lỗi hiệu suất trong microservices đám mây," trong *Proc. 24th Int. Conf. Architect. Chương trình hỗ trợ. Lang. Oper. Hệ thống.*, 2019, trang 19–33.
12. S. Lima, J. Correia, F. Araujo và J. Cardoso, "Cải thiện khả năng quan sát trong hệ thống tìm nguồn cung ứng sự kiện," *J. Syst. Softw.*, tập 181, tháng 11 năm 2021, Điều số 111015.
13. A. Janes, X. Li và V. Lenarduzzi, "Công cụ theo dõi mở: Tổng quan và so sánh quan trọng," 2022, *arXiv: 2207.06875*.
14. J. Halvorsen, J. Waite và A. Hahn, "Đánh giá khả năng quan sát của các chiến lược giám sát an ninh mạng với TOMATO," *IEEE Access*, tập 7, trang 108304–108315, 2019.
15. A. Tagliabue và cộng sự, "LION: Bộ điều hướng nhận thức khả năng quan sát quán tính LiDAR cho môi trường bị từ chối tầm nhìn," trong *Proc. Int. Symp. Exp. Robot.*, 2020, trang 380–390.
16. R. Drouilly, P. Rives và B. Morisset, "Biểu diễn ngữ nghĩa để điều hướng trong môi trường quy mô lớn," trong *Proc. IEEE Int. Conf. Robot. Tự động.*, 2015, trang 1106–1111.
17. S. Kabinna, C.-P. Bezemer, W. Shang, MD Syer và AE Hassan, "Kiểm tra tính ổn định của các tuyên bố ghi nhật ký," *Thực nghiệm Softw. Eng.*, tập 23, số 1, trang 290–333, 2018.
18. T. Sundqvist, M. Bhuyan và E. Elmroth, "Khám phá các bất thường về độ trễ trong 5G RAN — Phương pháp tiếp cận người học kết hợp," trong *Proc. 14th Int. Conf. Commun. Syst. Netw.*, 2022, trang 621–629.
19. X. Li, P. Chen, L. Jing, Z. Anh ấy và G. Yu, "SwissLog: Phát hiện và định vị bất thường mạnh mẽ cho các nhật ký không có cấu trúc xen kẽ," *IEEE Trans. Depend. Tính toán an toàn.*, tập 20, số 4, trang 2762–2780, Tháng Bảy / Tháng Tám năm 2023.
20. A. Pi, W. Chen, S. Wang và X. Zhou, "Xây dựng và phân tích quy trình làm việc nhận biết ngữ nghĩa cho các hệ thống phân tích dữ liệu phân tán," trong *Proc. 28th Int. Symp. Hiệu suất cao. Phân phối song song.*, 2019, trang 255–266.
21. S. Hochreiter và J. Schmidhuber, "Trí nhớ ngắn hạn dài," *Tính toán thần kinh.*, tập 9, số 8, trang 1735–1780, 1997.
22. K. Cho, B. Van Merriënboer, D. Bahdanau và Y. Bengio, "Về các thuộc tính của dịch máy thần kinh: Phương pháp tiếp cận bộ mã hóa-giải mã," 2014, *arXiv: 1409.1259*.
23. A. Vaswani và cộng sự, "Chú ý là tất cả những gì bạn cần," trong *Proc. Adv. Neural Inf. Process. Syst.*, 2017, trang 5998–6008.
24. R. Yang, D. Qu, Y. Gao, Y. Qian và Y. Tang, "NLSALog: Khung phát hiện bất thường cho trình tự nhật ký trong quản lý bảo mật," *IEEE Access*, tập 7, trang 181152–181164, 2019.
25. S. Gu, Y. Chu, W. Zhang, P. Liu, Q. Yin và Q. Li, "Nghiên cứu về phát hiện bất thường nhật ký hệ thống kết hợp GRU lát cắt hai chiều và cơ chế chú ý GA," trong *Proc. 4th Int. Conf. Artif. Thông minh. Dữ liệu lớn*, 2021, trang 577–583.
26. L. Yang và cộng sự, "Phát hiện bất thường dựa trên nhật ký bán giám sát thông qua ước tính nhãn xác suất," trong *Proc. IEEE / ACM 43rd Int. Conf. Softw. Eng.*, 2021, trang 1448–1460.
27. S. Huang và cộng sự, "Dị thể: Máy biến áp phân cấp để phát hiện bất thường trong nhật ký hệ thống," *IEEE Trans. Netw. Quản lý dịch vụ.*, tập 17, số 4, trang 2064–2076, tháng 12 năm 2020.
28. C. Zhang, X. Wang, H. Zhang, H. Zhang và P. Han, "Phát hiện bất thường trình tự nhật ký dựa trên trích xuất thông tin cục bộ và mô hình máy biến áp thưa thớt toàn cầu," *IEEE Trans. Netw. Quản lý dịch vụ.*, tập 18, số 4, trang 4119–4133, tháng 12 năm 2021.
29. D. Yu, X. Hou, C. Li, Q. Lv, Y. Wang và N. Li, "Phát hiện bất thường trong nhật ký phi cấu trúc bằng cách sử dụng mạng bi-LSTM dựa trên sự chú ý," trong *Proc. 7th IEEE Int. Conf. Netw. Thông minh. Ngón. Nội dung*, 2021, trang 403–407.
30. C. Elfving, "Người học theo thời gian và tuần tự để phát hiện bất thường nhật ký hệ thống: Phát triển và đánh giá mô hình máy học mới để phát hiện bất thường dựa trên nhật ký hệ thống 5G RAN", 2023.

Tobias Sundqvist (Thành viên, IEEE) nhận bằng Tiến sĩ về khoa học máy tính và kỹ thuật tại Đại học Umeå, Thụy Điển. Ông đã phát triển mạng truy cập vô tuyến trong 20 năm qua và chuyên môn của ông là phát hiện bất thường và khả năng quan sát trong các hệ thống phân tán lớn.

Monowar Bhuyan (Thành viên, IEEE) nhận bằng tiến sĩ về khoa học máy tính và kỹ thuật tại Đại học Tezpur, Ấn Độ, vào năm 2014. Ông là Trợ lý Giáo sư tại Khoa Khoa học Máy tính, Đại học Umeå, Thụy Điển, từ tháng 1 năm 2020 và là một trong những trưởng nhóm nghiên cứu tại Phòng thí nghiệm Hệ thống Phân tán Tự động. Trước đó, ông làm việc với Viện Khoa học và Công nghệ Nara, Nhật Bản; Đại học Umeå; Đại học Assam Kaziranga, Ấn Độ; và Tezpur

### Đại học từ tháng 1 năm 2009 đến tháng 12 năm 2019. Nó

đã xuất bản hơn 70 bài báo trên các tạp chí quốc tế hàng đầu và kỷ yếu hội nghị và đã viết một cuốn sách với Springer. Kinh nghiệm lãnh đạo / đồng lãnh đạo các dự án nghiên cứu của ông đã thu hút hơn 25 khoản tài trợ của MSEK trong nước, quốc tế và Liên minh Châu Âu. Các lĩnh vực nghiên cứu chính của ông bao gồm học máy, phát hiện bất thường, bảo mật hệ thống và hệ thống phân tán. Ông cũng là thành viên của ACM.

Erik Elmroth là Giáo sư Khoa học Máy tính của Đại học Umeå. Ông đã là Trưởng khoa và Phó Trưởng khoa Khoa học Máy tính trong 13 năm và phó giám đốc một trung tâm siêu máy tính quốc gia trong 13 năm nữa. Ông đã thành lập nghiên cứu của Đại học Umeå về các hệ thống phân tán, giải quyết các hệ thống quản lý tự động cho cơ sở hạ tầng điện toán ảo, chẳng hạn như đám mây và môi trường biên, xem http://www.cloudresearch.org. Kinh nghiệm của ông với các nhóm quản lý và điều hành trên quy mô lớn

các dự án nghiên cứu bao gồm các điểm nổi bật, chẳng hạn như Wallenberg AI trị giá 550 triệu euro, hệ thống tự động và chương trình phần mềm, và lĩnh vực nghiên cứu chiến lược eSSENCE. Ông đã phát triển hai chiến lược nghiên cứu cho Hội đồng Bộ trưởng Bắc Âu. Kinh nghiệm quốc tế bao gồm một năm tại NERSC, Phòng thí nghiệm Quốc gia Lawrence Berkeley, Đại học California tại Berkeley, Berkeley và một học kỳ tại Viện Công nghệ Massachusetts, Cambridge, MA, Hoa Kỳ. Ông cũng là thành viên của Ủy ban Hội đồng Nghiên cứu Thụy Điển về cơ sở hạ tầng nghiên cứu và Chủ tịch hội đồng chuyên gia về Khoa học điện tử cũng như Chủ tịch Hội đồng Cơ sở hạ tầng Quốc gia Thụy Điển về Máy tính. Ông là thành viên trọn đời của Viện Hàn lâm Khoa học Kỹ thuật Hoàng gia Thụy Điển và là Phó Chủ tịch bộ phận Công nghệ Thông tin.

1. https://opentracing.io/specification/ 2https://opentelemetry.io [↑](#footnote-ref-1)
2. https://www.tietoevry.com [↑](#footnote-ref-2)
3. https://github.com/Tobbis/qvistigt [↑](#footnote-ref-3)
4. https://www.umu.se/en/ [↑](#footnote-ref-4)