

kTrans:Biến áp nhận thức kiến thức để nhúng mã nhị phân

Văn Ngọc Chúc
Đại học Thanh Hoa

Hạo Vương
Đại học Thanh Hoa

Chu Ngọc Thần
Đại học Công nghệ Bắc Kinh

Gia Minh Vương
Đại học Khoa học và Công nghệ Huazhong

Tử Hàm Sa
Đại học Kỹ thuật Thông tin

Zeyu Gao
Đại học Thanh Hoa

Triệu Chương
Đại học Thanh Hoa

trường

Nhúng mã nhị phân (BCE) có các ứng dụng quan trọng trong các nhiệm vụ kỹ thuật đảo ngược khác nhau như phát hiện sự tương tự mã nhị phân, khôi phục kiểu, khôi phục luồng điều khiển và phân tích luồng dữ liệu. Các nghiên cứu gần đây đã chỉ ra rằng mô hình Transformer có thể hiểu được ngữ nghĩa của mã nhị phân để hỗ trợ các tác vụ tiếp theo. Tuy nhiên, các mô hình hiện có đã bỏ qua kiến thức trước đó về ngôn ngữ hợp ngữ. Trong bài báo này, chúng tôi đề xuất một phương pháp tiếp cận mới dựa trên Transformer, cụ thể là kTrans, để tạo ra việc nhúng mã nhị phân nhận biết kiến thức. Bằng cách cho *kiến thức rõ ràng* làm đầu vào bổ sung cho Máy biến áp và kết hợp *kiến thức tiềm ẩn* với một nhiệm vụ đào tạo trước mới lạ, kTrans cung cấp một góc nhìn mới để kết hợp kiến thức miền vào khung Transformer. Chúng tôi kiểm tra các phần nhúng được tạo bằng tính năng phát hiện và trực quan hóa ngoại lệ, đồng thời áp dụng kTrans đến 3 tác vụ xuôi dòng: Phát hiện sự tương tự mã nhị phân (BCSD), Khôi phục loại chức năng (FTR) và Nhận dạng cuộc gọi gián tiếp (ICR). Kết quả đánh giá cho thấy kTrans có thể tạo ra các phần nhúng mã nhị phân chất lượng cao và vượt trội hơn các phương pháp tiếp cận hiện đại (SOTA) đối với các tác vụ hạ nguồn lần lượt là 5,2%, 6,8% và 12,6%. kTrans được cung cấp công khai tại: <https://github.com/Learner0x5a/kTrans-release>

1. Giới thiệu

Nhúng mã nhị phân (BCE), còn được gọi là học biểu diễn mã nhị phân, nhằm mục đích ánh xạ mã nhị phân không có cấu trúc vào một không gian có chiều thấp trong đó mã nhị phân được biểu diễn dưới dạng nhúng. Thông qua việc nhúng mã nhị phân, nhiều nhiệm vụ phân tích mã nhị phân truyền thống có thể được cải thiện bằng cách sử dụng các phương pháp học sâu, chẳng hạn như phát hiện sự giống nhau của mã nhị phân [5, 11, 28–30, 35, 50, 52, 54], nhận dạng ranh giới hàm [41, 48], phục hồi chữ ký hàm [số 8], phân tích tập giá trị [17] và nhận dạng cuộc gọi gián tiếp [55], v.v. Với sự phát triển của các mô hình ngôn ngữ lớn, Transformer [10] đã được chứng minh là một mô hình biểu diễn ngôn ngữ hiệu quả và cũng đã đạt được hiệu suất xuất sắc trong các nhiệm vụ phân tích mã nhị phân khác nhau.

Các phương pháp nhúng mã nhị phân hiện tại có thể được chia đại khái thành hai loại: nhúng thủ công và nhúng dựa trên học tập. Việc nhúng thủ công liên quan đến việc biểu diễn mã nhị phân với các đặc điểm số được xây dựng thủ công, chẳng hạn như Gemini [52], Hướng dẫn2Vec [28], v.v. Tuy nhiên, những phương pháp như vậy đòi hỏi chuyên môn sâu rộng về miền và thường có nhiệm vụ cụ thể, dẫn đến khả năng chuyển giao kém. Mặt khác, việc nhúng dựa trên học tập sẽ tự động hóa quá trình tạo các phần nhúng cho mã nhị phân bằng các phương pháp học biểu diễn, chẳng hạn như Asm2Vec [11], AN TOÀN [35], v.v. Những phương pháp này tự động tìm hiểu các tính năng từ dữ liệu, tránh những sai lệch do kỹ thuật tính năng thủ công gây ra. Hơn nữa, với ứng dụng rộng rãi của Transformers, nhiều phương pháp tiếp cận dựa trên Transformer đã được đề xuất và đạt được hiệu suất tiên tiến trên các tác vụ hạ nguồn khác nhau, chẳng hạn như PalmTree [29], jTrans [50], COMBO [54], v.v. Bất chấp sự thành công của các phương pháp tiếp cận hiện có trong nhiệm vụ phân tích mã nhị phân, chúng vẫn có một số hạn chế.

Thứ nhất, các phương pháp tiếp cận hiện tại thiếu việc sử dụng kiến thức sẵn có về mã nhị phân. Hầu hết các phương pháp tiếp cận đều coi mã nhị phân là ngôn ngữ tự nhiên và áp dụng trực tiếp các mô hình ngôn ngữ tự nhiên vào hợp ngữ [5, 50]. Tuy nhiên, hợp ngữ chứa kiến thức về kiến trúc tập lệnh (ISA), bao gồm các loại mã lệnh, loại toán hạng, mối quan hệ giữa các mã lệnh và thông tin khác. Ví dụ, đối với `rax, eax, r10, al`, một mô hình ngôn ngữ tự nhiên sẽ coi chúng là các mã thông báo độc lập, nhưng trên thực tế, ba mã sau đều là một phần của `rax`. Vì vậy, nếu mô hình có thể hiểu được mối quan hệ giữa các r10 và tất cả, nó sẽ có thể nắm bắt được mối quan hệ luồng dữ liệu giữa các hướng dẫn khi lập mô hình các chuỗi như `movzx eax, byte [rdx]; cmp byte [rdx], al`.

Thứ hai, các phương pháp tiếp cận hiện tại thiếu hiểu biết về hướng dẫn và do đó không thể mô hình hóa hành vi thực hiện chương trình. Lệnh là đơn vị cơ bản của việc thực hiện chương trình, nhưng hầu hết các phương pháp đều áp dụng trực tiếp các mô hình ngôn ngữ tự nhiên vào hợp ngữ mà không hiểu rõ về ranh giới lệnh. Ví dụ: PalmTree yêu cầu người dùng cung cấp ranh giới hướng dẫn, trong khi BinBert [5] thiếu

Bảng 1: So sánh các mô hình hợp ngữ.

Phương pháp	LÀ MỘT Kiến thức	Chỉ dẫn ranh giới	ngầm phụ thuộc	Theo ngữ cảnh Sự suy luận
word2vec	N	N	N	N
Cây cọ	N	N	một phần	N
BinBert	N	N	N	Y
jTrans	N	một phần	một phần	Y
kTrans	Y	Y	Y	Y

thông tin về ranh giới hướng dẫn hoàn toàn. Nếu không hiểu đúng về hướng dẫn, mô hình có thể không phân biệt được giữa ['bật', 'rbp'] và ['rbp', 'pop'].

Hơn nữa, các phương pháp tiếp cận hiện tại thiếu mô hình hóa các phụ thuộc tiềm ẩn trong mã nhị phân. Các hướng dẫn hợp ngữ không phải là các thực thể độc lập mà có các phụ thuộc tiềm ẩn, chẳng hạn như thanh ghi cờ toàn cầu EFLAGS. Các phương pháp tiếp cận hiện tại giải quyết một phần sự phụ thuộc tiềm ẩn thông qua thiết kế thủ công. Ví dụ: các mối quan hệ nhảy lệnh của mô hình jTrans bằng cách chia sẻ các tham số giữa các phần nhúng mã thông báo và các phần nhúng vị trí, nhưng nó thiếu sự xem xét đối với các phần phụ thuộc khác. PalmTree mô hình hóa sự phụ thuộc dữ liệu bằng cách xây dựng các tác vụ Dự đoán trình tự tiếp theo (NSP) trên biểu đồ luồng dữ liệu, nhưng nó hy sinh khả năng mô hình hóa ngữ cảnh hợp ngữ hoàn chỉnh. Về chiến lược che giấu của Transformers [10], mặt nạ cấp mã thông báo đơn giản chỉ có thể mô hình hóa các phần phụ thuộc giữa các mã thông báo riêng lẻ và không thể nắm bắt được phần phụ thuộc giữa các hướng dẫn. Hãy xem xét đoạn lắp ráp kiểm tra eax, eax; mov RCx, qword [rbp-num]; jne addr. Trong những trường hợp như kiểm tra eax, [MẶT NẠ] hoặc jne [MẶT NẠ], mô hình có thể dự đoán chính xác các mã thông báo bị che giấu mà không cần thông tin theo ngữ cảnh. Tuy nhiên, mô hình không tìm hiểu được mối quan hệ giữa các lệnh kiểm tra eax, eax và tôi thêm vào, mà thực sự ngụ ý một sự phụ thuộc nhánh có điều kiện. Tóm lại, như thể hiện trong Bảng 1 vẫn còn thiếu một giải pháp nhúng mã nhị phân toàn diện.

Trong bài báo này, chúng tôi đề xuất kTrans, một phương pháp nhúng mã nhị phân dựa trên Transformer mới. kTrans kết hợp kiến thức trước đây về ngôn ngữ hợp ngữ vào mô hình Transformer, đồng thời có thể mô hình hóa sự phụ thuộc ngầm giữa các hướng dẫn với một nhiệm vụ đào tạo trước mới.

kTrans kết hợp kiến thức rõ ràng và tiềm ẩn theo hai cách khác nhau tương ứng:

- Đưa kiến thức về mã thông báo một cách rõ ràng. Loại kiến thức này được tạo ra dựa trên định nghĩa của Kiến trúc tập lệnh (ISA), chẳng hạn như loại lệnh và mối quan hệ giữa các thanh ghi rax, eax, r10, al, và được đưa vào Máy biến áp dưới dạng đầu vào bổ sung. Bằng cách đưa vào loại kiến thức này, mô hình có được tiềm năng tìm hiểu các thuộc tính mã thông báo và ranh giới lệnh.
- Đưa kiến thức giảng dạy ngầm vào. Loại kiến thức này được ngầm đưa vào bằng cách sử dụng cấp độ hướng dẫn

mặt nạ. Bằng cách che giấu toàn bộ lệnh, mô hình sẽ hiểu được ranh giới lệnh và có thể mô hình hóa các mối quan hệ ngầm giữa các lệnh, chẳng hạn như các phụ thuộc nhánh có điều kiện.

Chúng tôi đã tiến hành đào tạo trước Tran trên quy mô lớn tập dữ liệu mã nhị phân và đánh giá hiệu suất của nó đối với các nhiệm vụ nội tại và nhiệm vụ phân tích mã nhị phân trong thế giới thực. Kết quả thực nghiệm chứng minh rằng kTrans vượt qua các phương pháp trước đó trong việc mô hình hóa hợp ngữ với độ phức tạp thấp nhất ($1 + 3.3e^{-4}$) và tạo ra các mã nhúng mã nhị phân chất lượng cao với độ chính xác phát hiện ngoại lệ cao nhất (86.6%). Thông qua các thí nghiệm cắt bỏ, chúng tôi đã xác nhận tính hiệu quả của thiết kế nhận biết kiến thức. Hơn nữa, trên 3 tác vụ xuôi dòng, tức là phát hiện sự tương tự mã nhị phân, khôi phục loại chức năng và nhận dạng cuộc gọi gián tiếp, kTrans vượt trội hơn các giải pháp tiên tiến nhất với 0,745 MRR@10000 (75.2%), độ chính xác 0,876 (76.8%) và 0,28 MRR@32 (71.2% tương ứng là 6%).

Nhìn chung, chúng tôi chứng minh rằng việc đưa kiến thức vào có thể nâng cao hiệu quả khả năng của các mô hình hợp ngữ, cung cấp một hướng mới cho việc phát triển các mô hình ngôn ngữ cho mã nhị phân. Và với sự tiến bộ của các mô hình quy mô lớn có mục đích chung cung cấp những hiểu biết mới cho các mô hình miền trong lĩnh vực nhúng mã nhị phân, chúng tôi thảo luận về ba hướng nghiên cứu trong tương lai để nhúng mã nhị phân: mô hình miền lớn hơn, mô hình hiệu quả về chi phí và kết hợp với các mô hình lớn nói chung. các mô hình ngôn ngữ

Tóm lại, công trình này có những đóng góp sau:

- Chúng tôi đề xuất một cách tiếp cận mới để kết hợp kiến thức có sẵn vào Transformers, có thể mô hình hóa các phụ thuộc tiềm ẩn trong hợp ngữ. Nó cải thiện đáng kể khả năng dự đoán của các mô hình hợp ngữ, cho thấy triển vọng đầy hứa hẹn của việc hợp nhất kiến thức.
- Chúng tôi đã tiến hành nhiều thử nghiệm và kết quả chứng minh rằng các phần nhúng được tạo ra bởi mô hình được đào tạo trước của chúng tôi vượt trội so với các công việc trước đó và nâng cao đáng kể hiệu suất của các tác vụ tiếp theo.
- Chúng tôi thảo luận về một số hướng nghiên cứu trong tương lai cho việc nhúng mã nhị phân.
- Chúng tôi mã nguồn mở kTrans để tạo thuận lợi cho việc nghiên cứu sau này.

2 Bối cảnh và công việc liên quan

2.1 Nhúng mã nhị phân

Nhúng mã nhị phân (BCE) là một kỹ thuật nhằm ánh xạ mã nhị phân không có cấu trúc vào một không gian có chiều thấp, trong đó mã nhị phân được biểu diễn dưới dạng nhúng. BCE có nhiều ứng dụng như phát hiện độ tương tự mã nhị phân, khôi phục kiểu, khôi phục luồng điều khiển và phân tích luồng dữ liệu, v.v. Trong khi đó, với sự phát triển của ngôn ngữ lớn

mô hình, mô hình dựa trên máy biến áp [10] đã nổi lên như một phương pháp phổ biến cho BCE về khả năng tự động tìm hiểu các tính năng phức tạp. Phần này sẽ cung cấp tổng quan về các phương pháp khác nhau để nhúng mã nhị phân, bao gồm nhúng mã nhị phân thủ công, nhúng mã nhị phân dựa trên học tập không dựa trên Transformer và nhúng mã nhị phân dựa trên học tập dựa trên Transformer.

2.2 Nhúng mã nhị phân thủ công

Việc nhúng mã nhị phân thủ công liên quan đến việc biểu diễn mã nhị phân bằng cách sử dụng các đặc điểm số được xây dựng thủ công. Thông thường, các tính năng này được thiết kế để nắm bắt thông tin cú pháp và ngữ nghĩa có trong mã nhị phân. Một số tính năng thường được sử dụng bao gồm:

- **Lệnh N-gram:** Lệnh n-gram biểu thị chuỗi N lệnh liên tiếp trong mã nhị phân. Các tính năng này có thể nắm bắt các mẫu cục bộ và sự phụ thuộc giữa các hướng dẫn nhưng có thể không thể mô hình hóa các phần phụ thuộc tầm xa.
- **Đồ thị luồng điều khiển (CFG):** CFG thể hiện cấu trúc luồng điều khiển của mã nhị phân dưới dạng đồ thị có hướng, với các khối cơ bản là các nút và các kết nối luồng điều khiển là các cạnh giữa các nút. CFG có thể nắm bắt cấu trúc luồng điều khiển của mã nhị phân nhưng không thể mô hình hóa các phụ thuộc dữ liệu và thông tin ngữ nghĩa khác.
- **Đồ thị gọi hàm (FCG):** FCG thể hiện mối quan hệ gọi hàm giữa các hàm trong mã nhị phân dưới dạng đồ thị có hướng, với các hàm là nút và lệnh gọi hàm là các cạnh. FCG có thể nắm bắt được mối quan hệ cấp cao giữa các chức năng nhưng không thể mô hình hóa cấu trúc chi tiết và ngữ nghĩa trong các chức năng riêng lẻ.

Mặc dù tính đơn giản và dễ dàng trích xuất nhưng các phương pháp nhúng mã nhị phân thủ công vẫn có một số hạn chế. Chúng đòi hỏi kiến thức chuyên môn sâu rộng về miền và thường có nhiệm vụ cụ thể, dẫn đến khả năng chuyển đổi kém giữa các nhiệm vụ phân tích mã nhị phân khác nhau. Ngoài ra, các phương pháp này có khả năng thích ứng và khả năng mở rộng hạn chế đối với những thách thức và tiến bộ mới trong lĩnh vực phân tích mã nhị phân.

2.3 Nhúng mã nhị phân dựa trên học tập

Loại phương pháp tiếp cận này tự động hóa quá trình tạo các phần nhúng cho mã nhị phân thông qua học biểu diễn, học các tính năng tự động từ dữ liệu, tránh những sai lệch do kỹ thuật tính năng thủ công gây ra.

Không dựa trên máy biến ápLoại phương pháp tiếp cận này có thể được chia thành nhiều loại dựa trên các thuật toán học tập cơ bản mà chúng sử dụng:

- **Dựa trên RNN:** Các phương thức như AN TOÀN [35] sử dụng mạng thần kinh tái phát (RNN) để mô hình hóa các chuỗi mã nhị phân. Các phương pháp dựa trên RNN, chẳng hạn như LSTM [20] và GRU [9], có thể nắm bắt các phụ thuộc tuần tự trong dữ liệu và rất phù hợp để mô hình hóa các chuỗi mã nhị phân có độ dài thay đổi. Tuy nhiên, RNN có thể gặp phải vấn đề biến mất độ dốc khi xử lý các chuỗi dài, hạn chế khả năng mô hình hóa các phụ thuộc tầm xa của chúng.
- **Dựa trên CNN:** Các phương pháp như adiff [30] sử dụng mạng thần kinh tích chập (CNN) để mô hình hóa các mẫu và mối quan hệ cục bộ trong mã nhị phân. CNN có thể nắm bắt thông tin không gian và phân cấp bằng cách sử dụng các lớp tích chập và hoạt động gộp. Tuy nhiên, CNN có thể không hiệu quả trong việc mô hình hóa các phụ thuộc tuần tự và các mối quan hệ tầm xa trong mã nhị phân như RNN hoặc Transformers.
- **Dựa trên GNN:** Các phương pháp dựa trên mạng thần kinh đồ thị (GNN), chẳng hạn như Gemini [52] và VulSeeker [16], mô hình hóa mã nhị phân bằng cách sử dụng các biểu diễn đồ thị, chẳng hạn như đồ thị luồng điều khiển (CFG) và đồ thị luồng dữ liệu (DFG). GNN có thể nắm bắt các mối quan hệ và phụ thuộc phức tạp giữa các phần khác nhau của mã nhị phân bằng cách xử lý dữ liệu có cấu trúc biểu đồ. Tuy nhiên, GNN có thể yêu cầu các bước tiền xử lý bổ sung để trích xuất các biểu diễn đồ thị từ mã nhị phân và có thể tốn kém về mặt tính toán đối với các đồ thị lớn.

Dựa trên máy biến ápCác phương pháp nhúng mã nhị phân dựa trên máy biến áp tận dụng cơ chế tự chú ý mạnh mẽ có trong Transformers, cho phép chúng nắm bắt các phần phụ thuộc tầm xa và các mẫu phức tạp trong mã nhị phân. Ví dụ về các phương pháp dựa trên Transformer bao gồm PalmTree [29], jTrans [50] và COMBO [54]. Các phương pháp này thường sử dụng các mô hình tinh chỉnh trước khi huấn luyện, cho phép chúng tìm hiểu các cách biểu diễn mã nhị phân có mục đích chung từ dữ liệu quy mô lớn, không được gắn nhãn và sau đó tinh chỉnh các cách biểu diễn này cho các nhiệm vụ phân tích mã nhị phân cụ thể.

Một số ưu điểm chính của các phương pháp dựa trên Transformer so với các phương pháp không dựa trên Transformer bao gồm:

- **Cải thiện mô hình hóa các phần phụ thuộc tầm xa:** Máy biến áp sử dụng cơ chế tự chú ý để lập mô hình mối quan hệ giữa tất cả các cặp mã thông báo trong chuỗi đầu vào, cho phép chúng nắm bắt các phần phụ thuộc tầm xa hiệu quả hơn RNN hoặc CNN.
- **Khả năng mở rộng:** Máy biến áp có thể xử lý các chuỗi đầu vào song song, làm cho chúng hiệu quả tính toán hơn và có khả năng mở rộng hơn so với RNN vốn yêu cầu xử lý tuần tự.
- **Học chuyển giao:** Các phương pháp dựa trên máy biến áp thường sử dụng các mô hình học chuyển giao như tinh chỉnh tiền đào tạo, cho phép chúng tận dụng quy mô lớn,

dữ liệu không được gắn nhãn để tìm hiểu cách biểu diễn mã nhị phân có mục đích chung và cải thiện hiệu suất trên các tác vụ phân tích mã nhị phân cụ thể.

Tuy nhiên, các phương pháp nhúng mã nhị phân dựa trên Transformer cũng gặp phải một số thách thức:

- Độ phức tạp tính toán: Mặc dù Transformers có thể xử lý các chuỗi đầu vào song song, nhưng cơ chế tự chú ý của chúng có độ phức tạp bậc hai đối với độ dài chuỗi, khiến chúng tốn kém về mặt tính toán đối với các chuỗi dài. Điều này có thể được giảm thiểu ở một mức độ nào đó bằng cách sử dụng các kỹ thuật như chú ý thưa thớt hoặc chú ý cục bộ.
- Tích hợp kiến thức trước: Hầu hết các phương pháp tiếp cận dựa trên Transformer đều xử lý mã nhị phân như ngôn ngữ tự nhiên và áp dụng trực tiếp các mô hình ngôn ngữ vào hợp ngữ. Điều này có thể bỏ qua những kiến thức sẵn có trong mã nhị phân, chẳng hạn như các loại mã lệnh, các loại toán hạng, v.v.
- Hiểu các hướng dẫn: Nhiều phương pháp tiếp cận dựa trên Transformer thiếu sự hiểu biết rõ ràng về ranh giới hướng dẫn, điều này có thể cản trở khả năng mô hình hóa hành vi thực hiện chương trình của chúng. Một số phương pháp, như PalmTree, yêu cầu người dùng cung cấp ranh giới hướng dẫn, trong khi các phương pháp khác, như BinBert [5], thiếu thông tin hoàn toàn về ranh giới hướng dẫn.

Tóm lại, việc nhúng mã nhị phân đóng một vai trò quan trọng trong việc cải thiện hiệu suất của các nhiệm vụ phân tích mã nhị phân khác nhau. Trong khi các phương pháp nhúng mã nhị phân thủ công có những hạn chế nhất định, các phương pháp nhúng mã nhị phân dựa trên học tập, cả không dựa trên Transformer và dựa trên Transformer, đã cho thấy nhiều hứa hẹn trong việc nắm bắt tốt hơn cấu trúc và ngữ nghĩa của mã nhị phân. Việc tiếp tục nghiên cứu và phát triển trong lĩnh vực này có thể dẫn đến các phương pháp nhúng mã nhị phân hiệu quả và toàn diện hơn, có thể giải quyết những thách thức và hạn chế của các phương pháp tiếp cận hiện có.

Trong công việc này, để giải quyết những thách thức này và phát triển các phương pháp nhúng toàn diện, chúng tôi tập trung vào việc kết hợp kiến thức có sẵn về mã nhị phân, cải thiện sự hiểu biết về lệnh và lập mô hình các phụ thuộc tiềm ẩn.

2.4 Công việc liên quan

2.4.1 Học biểu diễn trong NLP

Học biểu diễn trong xử lý ngôn ngữ tự nhiên (NLP) nhằm mục đích học các cách biểu diễn vector liên tục của các từ, cụm từ hoặc câu nắm bắt được ý nghĩa ngữ nghĩa của chúng. Các kỹ thuật như word2vec [37], Găng tay [42] và ELMo [43] đã được sử dụng rộng rãi cho các nhiệm vụ NLP khác nhau, bao gồm dịch máy, phân tích tình cảm và trả lời câu hỏi. Các mô hình này dựa trên mạng thần kinh nông hoặc mạng thần kinh tái phát và dựa vào số liệu thống kê về sự xuất hiện để tìm hiểu các phần nhúng. Gần đây hơn,

Các mô hình dựa trên máy biến áp như BERT [10], GPT [7,45,46], và RoBERTa [31] đã đạt được những kết quả tiên tiến nhất trong nhiều tiêu chuẩn NLP, thể hiện sức mạnh của cơ chế tự chú ý trong việc học các cách biểu diễn theo ngữ cảnh có thể mô hình hóa các phụ thuộc tầm xa.

2.4.2 Học biểu diễn cho mã nhị phân

Học biểu diễn mã nhị phân là một lĩnh vực nghiên cứu tích cực trong những năm gần đây. Công việc ban đầu tập trung vào các phương pháp trích xuất đặc trưng thủ công [12,13,15,30,52], trong đó các tính năng như rawbyte, biểu đồ luồng điều khiển, số liệu thống kê về hướng dẫn và biểu đồ lệnh gọi hàm được sử dụng để nắm bắt cấu trúc và ngữ nghĩa của mã. Tuy nhiên, những cách tiếp cận thủ công này có xu hướng tốn nhiều công sức và có thể bỏ sót những thông tin ngữ nghĩa quan trọng.

Các nghiên cứu gần đây chủ yếu tập trung vào khám phá các kỹ thuật dựa trên học tập, bao gồm doc2vec [11], CNN [30], LSTM [34,35], mạng lưới thần kinh đồ thị [16,33,52] và Máy biến áp [5,32,50,53,54]. Các kỹ thuật này tận dụng các thuật toán học sâu để tìm hiểu các phần nhúng, nắm bắt cấu trúc và ngữ nghĩa của mã nhị phân tốt hơn. Mục tiêu của việc học biểu diễn cho mã nhị phân là cho phép cải thiện hiệu suất trong các tác vụ như phát hiện phần mềm độc hại, kỹ thuật đảo ngược, phân tích lỗ hổng và phát hiện tính tương tự của phần mềm, v.v.

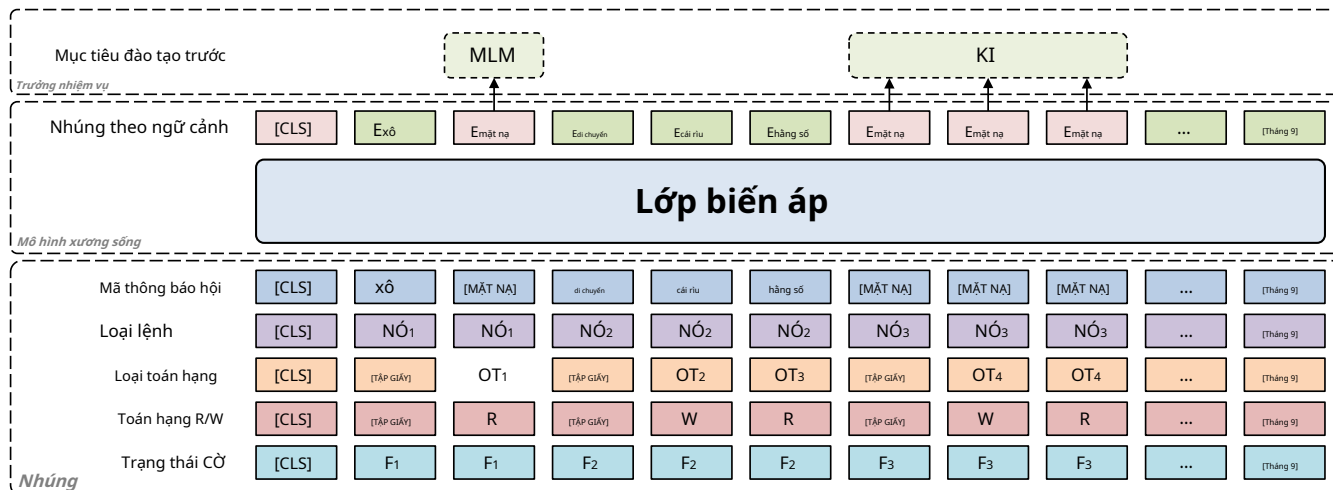
2.4.3 Học biểu diễn cho mã nguồn

Học cách biểu diễn cho mã nguồn nhằm mục đích tìm hiểu các cách biểu diễn có ý nghĩa của mã nguồn có thể được sử dụng trong các tác vụ kỹ thuật phần mềm khác nhau, chẳng hạn như hoàn thành mã [49], phát hiện lỗi [44], tổng hợp chương trình [6] và tóm tắt mã [24]. Các kỹ thuật như code2vec [4], code2seq [3] và vẽ đồ thị mạng lưới thần kinh [2] đã được đề xuất để tìm hiểu cách nhúng mã nguồn bằng cách nắm bắt thông tin cú pháp và ngữ nghĩa có trong mã, chẳng hạn như cây cú pháp trừu tượng, biểu đồ luồng điều khiển và biểu đồ luồng dữ liệu.

Công việc gần đây cũng đã khám phá việc sử dụng các mô hình dựa trên Transformer cho việc học biểu diễn mã nguồn [1,14]. Các mô hình này chứng minh tiềm năng của cơ chế tự chú ý trong việc nắm bắt các mẫu phức tạp và sự phụ thuộc lâu dài trong mã nguồn, giúp cải thiện hiệu suất đối với các tác vụ như tóm tắt mã, dịch mã và phát hiện lỗ hổng bảo mật tự động. Ngoài ra, mô hình tinh chỉnh tiền đào tạo được sử dụng bởi các mô hình dựa trên Transformer này cho phép tận dụng dữ liệu mã nguồn không được chú thích ở quy mô lớn để cải thiện tính tổng quát và độ tin cậy.

2.4.4 Ứng dụng của Học biểu diễn vào phân tích mã nhị phân

Biểu diễn mã nhị phân có thể được sử dụng để giải quyết các nhiệm vụ phân tích mã nhị phân khác nhau. Cây cọ [29] đề xuất kỹ thuật nhúng lệnh chung dựa trên Transformer cho nhiều tác vụ phân tích mã nhị phân. Song Tử [52] mở rộng



Hình 1: Tổng quan về Trans.

CFG với các tính năng được trích xuất thủ công (ví dụ: số lượng hướng dẫn) để biểu thị mã nhị phân và jTrans [50] kết hợp một biểu diễn nhận biết bước nhảy duy nhất cho thông tin luồng điều khiển trong mã nhị phân. Trọng lượng Byte [48] Và [41] xác định ranh giới hàm trong mã nhị phân tương ứng với RNN và Transformers và EKLAVYA [số 8] khôi phục chữ ký loại hàm bằng word2vec để hướng dẫn. CALLEE [55] thực hiện nhận dạng cuộc gọi gián tiếp với doc2vec để nhúng các lát mã nhị phân và DeepVSA [17] tạo điều kiện thuận lợi cho việc phân tích tập hợp giá trị bằng cách sử dụng hướng dẫn được thiết kế thủ công.

Ngoại trừ các nhiệm vụ được đề cập trước đó, biểu diễn mã nhị phân cũng có thể được áp dụng cho nhiều nhiệm vụ khác như phát hiện lỗi hỏng [16,30,50,52], xuất xứ của trình biên dịch [18,40], phát hiện phần mềm độc hại [22,51], v.v.

3 Phương pháp

3.1 Tổng quan

Để vượt qua những thách thức nêu ở phần 1, chúng tôi đề xuất một giải pháp mới gọi là kTrans để tự động tích hợp nâng cao kiến thức về mã nhị phân. kTrans được dựa trên Kiến trúc bộ mã hóa biến áp và tuân theo thiết kế mô-đun để đạt được khả năng mở rộng cao. Như sự bày tỏ n trong hình 1, kTrans bao gồm ba mô-đun chính: mô-đun nhúng, mô hình đường trục và mô-đun đầu nhiệm vụ. Đầu tiên, mô-đun nhúng chịu trách nhiệm đưa các thông tin thực tế về mã thông báo một cách rõ ràng, mô hình xương sống chịu trách nhiệm tạo ra biểu diễn nhúng và mô-đun đầu nhiệm vụ là phản hồi cho các nhiệm vụ đưa ra kiến thức tiềm ẩn.

Mô-đun nhúng tạo ra các phần nhúng cho Máy biến áp và kết quả thu được bằng cách tính tổng của chúng.

- Nhúng mã thông báo hội: Đối với chuỗi văn bản của

hướng dẫn lắp ráp, chúng tôi sử dụng một phương pháp mã thông báo phổ biến để thu được chuỗi mã thông báo và chuyển đổi nó thành vector, tức là nhúng mã thông báo.

- Nhúng kiến thức rõ ràng: Đối với kiến thức rõ ràng có trong các lệnh hợp ngữ, chẳng hạn như các loại mã hoạt động và các loại toán hạng, chúng tôi xây dựng một từ vựng dựa trên định nghĩa của ISA. Chúng tôi mã hóa kiến thức này dưới dạng một chuỗi được liên kết với các mã thông báo lắp ráp để tính toán mức độ nhúng kiến thức.
- Mã hóa vị trí: Để mã hóa vị trí, chúng tôi áp dụng phương pháp mã hóa vị trí được sử dụng rộng rãi từ BERT. Đối với việc nhúng phân đoạn, vì Máy biến áp của chúng tôi cần một chức năng duy nhất làm đầu vào và không liên quan đến nhiều bản án ử, phân đoạn em là định thì không

Mô hình xương sống sử dụng mô hình Transformer để tích hợp các phần nhúng khác nhau và tạo ra các phần nhúng theo ngữ cảnh. Mô hình Transformer nắm bắt thông tin theo ngữ cảnh trong chuỗi bằng cơ chế tự chú ý để tạo ra các phần nhúng theo ngữ cảnh, sau đó được sử dụng cho các biến thể nhiệm vụ hạ nguồn quan trọng.

Nhiệm vụ học mô-đun là để phân bổ trách nhiệm cho các hướng dẫn trước nhiệm vụ đào tạo. Trong quá trình đào tạo trước, kTrans bao gồm một mô-đun học mã thông báo (MLM) nhiệm vụ học mã thông báo (KI) nhiệm vụ.

3.1.1 Mô-đun nhúng

Một tập hợp nhúng định theo dõi các mô hình Paradigm, we first bình thường đầu tiên ze tập hợp chỉ dẫn st ở thời xướng ban of-vocabu ầu từ y (OOV) p cươp. Sau đó, we xây dựng từ vựng và chuyển đổi hướng dẫn lắp ráp thành mã thông báo dựa trên từ vựng. Chúng tôi thêm mã thông báo đặc biệt










































vào chuỗi mã thông báo của từng chức năng lắp ráp, bao gồm [CLS] ở đầu và [SEP] ở cuối. Mã thông báo [PAD] bổ sung được thêm vào cuối chuỗi mã thông báo để đảm bảo độ dài bằng nhau cho mỗi chuỗi. Chuỗi mã thông báo vượt quá giới hạn độ dài tối đa sẽ bị cắt bớt. Đối với các mã thông báo không có trong từ vựng, chúng tôi biểu thị chúng một cách thống nhất bằng mã thông báo [UNK] đặc biệt. Sau đó, chuỗi mã thông báo cuối cùng được chuyển qua lớp nhúng có thể học được để có được các mã nhúng mã thông báo tập hợp.

Nhúng kiến thức rõ ràng: Để tận dụng kiến thức rõ ràng có sẵn nhằm nâng cao Máy biến áp, chúng tôi xây dựng một số chuỗi kiến thức dựa trên ISA và thực hiện các hoạt động nhúng tương tự như chuỗi mã thông báo tập hợp. Như thể hiện trong hình 2, chúng tôi xem xét bốn loại kiến thức rõ ràng: loại opcode, loại toán hạng, trạng thái đọc/ghi toán hạng và trạng thái đăng ký FLAGS. Đối với mỗi loại kiến thức, chúng tôi cũng xây dựng vốn từ vựng và thêm các mã thông báo đặc biệt như [CLS], [SEP], [PAD], [UNK], v.v. Sau khi mã hóa, các chuỗi được chuyển qua các lớp nhúng có thể học tương ứng để tạo ra nhúng kiến thức.

Phần nhúng cuối cùng có được bằng cách tổng hợp các phần nhúng từ các nguồn khác nhau:

$$E_{\text{Kiến thức}} = E_{\text{Loại hướng dẫn}} + E_{\text{Loại toán hạng}} + E_{\text{Toán hạng RW}} + E_{\text{Cờ}}$$

$$E = E_{\text{Mã thông báo}} + E_{\text{Kiến thức}} + E_{\text{Chức vụ}}$$

Lớp biến áp							
							
Cuộc họp	...	XÔ	chuột chũi	đi chuyển	cải riu	0x80	...
Mã thông báo							
Chỉ dẫn	...	10	10	24	24	24	...
Kiểu							
Toán hạng	...	0	42	0	42	128	...
Kiểu							
Toán hạng	...	0	2	0	1	0	...
R/W							
CỜ	...	10	10	7	7	7	...
Trạng thái							

Hình 2: Tích hợp kiến thức rõ ràng của kTrans.

3.1.2 Mô hình xương sống

Chúng tôi áp dụng bộ mã hóa Transformer nhiều lớp làm mô hình xương sống. Transformer là mô hình ngôn ngữ hai chiều dựa trên cơ chế chú ý. Với dữ liệu đào tạo mở rộng và quá trình học tập tự giám sát, Transformer có thể sở hữu khả năng chuyển giao kiến thức hiệu quả. Mô hình xương sống thực hiện việc nhúng cuối cùng đầu vào và tạo ra các phần nhúng theo ngữ cảnh thông qua nhiều lớp Máy biến áp hai chiều.

3.1.3 Mô-đun đầu nhiệm vụ

Theo kết quả nghiên cứu của RoBERTa [31], nhiệm vụ Dự đoán câu tiếp theo (NSP) không phải là nhiệm vụ đào tạo trước hiệu quả. Do đó, chúng tôi chỉ sử dụng nhiệm vụ Mô hình ngôn ngữ đeo mặt nạ (MLM) để đào tạo trước và thay thế nhiệm vụ NSP bằng nhiệm vụ Tích hợp kiến thức (KI). Mục tiêu của nhiệm vụ KI là nâng cao Trans's sự hiểu biết về ranh giới hướng dẫn cũng như các ràng buộc tiềm ẩn giữa các hướng dẫn.

Người đứng đầu MLM: Chúng tôi áp dụng nhiệm vụ MLM từ BERT, trong đó 15% mã thông báo được che dấu ngẫu nhiên để mô hình dự đoán. Trong số các mã thông báo bị che giấu này, 80% được thay thế bằng mã thông báo đặc biệt [MASK], 10% được thay thế bằng mã thông báo ngẫu nhiên và 10% còn lại không thay đổi. Giả sử hàm lắp ráp được ký hiệu là $f = [x_1, \dots, x_N]$, Ở đâu $x_{TôI}$ là $TôI$ -mã thông báo thứ của f , và M là số lượng token. Đầu tiên chúng tôi chọn một tập hợp ngẫu nhiên các vị trí cho để che giấu (tức là, $tôix$).

$$f_{MLM} = \text{THAY THẾ}(f, tôix, [\text{MẶT NẠ}]) \quad (1)$$

Dựa trên những định nghĩa này, mục tiêu MLM của việc xây dựng lại các mã thông báo bị che giấu có thể được xây dựng như sau:

$$\underset{\theta}{\text{phút}} L_{MLM}(\theta) = \sum_{Tôix \in \text{tôix}} -\text{nhật ký } P(x_{Tôix} / f_{MLM}) \quad (2)$$

Ở đâu $tôix$ chứa các chỉ số của mã thông báo bị che. Một ví dụ về quy trình tạo mặt nạ được trình bày trong Hình 3.

đầu KI: Việc che dấu cấp độ mã thông báo không thể buộc mô hình tìm hiểu khái niệm "hướng dẫn" và do đó không thể mô hình hóa sự phụ thuộc giữa các hướng dẫn. Do đó, chúng tôi sử dụng chiến lược che giấu mức hướng dẫn. Đối với một hàm lắp ráp nhất định, chúng tôi chọn ngẫu nhiên 15% hướng dẫn và thay thế tất cả mã thông báo của chúng bằng mặt nạ, trong khi mã thông báo của các hướng dẫn còn lại không thay đổi. Giả sử hàm lắp ráp được ký hiệu là $f = [TôI, \dots, TôI]$, Ở đâu $TôI$ là $TôI$ -chỉ dẫn thứ của f , và M là số lượng hướng dẫn. Đầu tiên chúng tôi chọn một tập hợp ngẫu nhiên các vị trí cho để che giấu (tức là, $tôitôI$).

$$f_{KI} = \text{THAY THẾ}(f, tôitôI, [\text{MẶT NẠ}]) \quad (3)$$

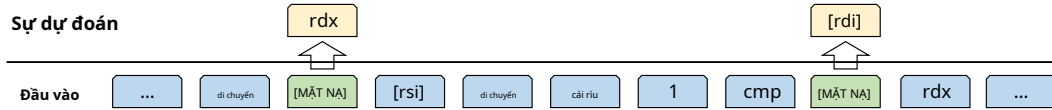
Dựa trên những định nghĩa này, mục tiêu KI của việc xây dựng lại các mã thông báo bị che giấu có thể được xây dựng như sau:

$$\underset{\theta}{\text{phút}} L_{KI}(\theta) = \sum_{j \in \text{tôitôI}} -\text{nhật ký } P(TôI_j / f_{KI}) \quad (4)$$

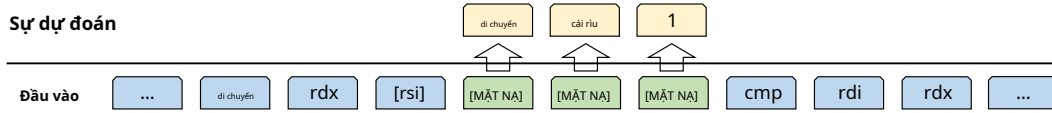
Ở đâu $tôitôI$ chứa các chỉ số của các hướng dẫn đeo mặt nạ. Một ví dụ về quy trình tạo mặt nạ được trình bày trong Hình 4.

Cách *hàm tổn thất tổng thể* của kTrans trong giai đoạn tiền đào tạo là tổng của các hàm mục tiêu MLM và KI:

$$\underset{\theta}{\text{phút}} L_{\theta} = L_{MLM}(\theta) + L_{KI}(\theta) \quad (5)$$



Hình 3: Mô hình ngôn ngữ đeo mặt nạ (MLM).



Hình 4: Tích hợp kiến thức (KI).

3.2 Cơ sở lý luận

Tích hợp kiến thức rõ ràng. Các mã thông báo và từng loại kiến thức rõ ràng có thể được coi là các phương thức khác nhau của tập hợp và dữ liệu từ một phương thức duy nhất có thể không đầy đủ hoặc mơ hồ. Bằng cách tích hợp thông tin từ nhiều phương thức, học tập đa phương thức có thể làm giảm tác động của những vấn đề đó. Nếu chúng ta có đủ dữ liệu, học tập đa phương thức đã được chứng minh là vượt trội hơn so với học tập đơn phương thức [23]. Do đó, chúng tôi kỳ vọng việc tích hợp kiến thức rõ ràng sẽ nâng cao tổng thể mô hình.

Tích hợp kiến thức tiềm ẩn. Mục tiêu của nhiệm vụ MLM tương đương với việc tối đa hóa thông tin lẫn nhau giữa đầu vào và đầu ra của bộ mã hóa Transformer [26], và các chiến lược che giấu khác nhau (MLM và KI) thực sự cung cấp các góc nhìn khác nhau về quá trình lắp ráp. Do đó, việc tối ưu hóa Máy biến áp với nhiều chế độ xem, tức là tối ưu hóa chung, thường có thể mang lại thông tin lẫn nhau lớn hơn so với một chế độ xem đơn lẻ. Nói cách khác, nếu các phần nhúng được tạo có thể dẫn đến tổn thất thấp cho nhiều chế độ xem, điều đó cho thấy khả năng biểu diễn của Transformer tốt hơn, tức là khả năng biểu diễn của các phần nhúng được tạo ra là tốt hơn.

3.3 Bộ dữ liệu

Để đào tạo trước Trans, chúng tôi sử dụng BinaryCorp [50] tập dữ liệu để trích xuất mã hợp ngữ và kiến thức. Bộ dữ liệu BinaryCorp bao gồm hơn 10 nghìn dự án và 26 triệu hàm mã nhị phân, bao gồm nhiều loại mục tiêu khác nhau như trình soạn thảo, trình nhắn tin tức thời, máy chủ HTTP, trình duyệt web, trình biên dịch, thư viện đồ họa, thư viện mật mã, v.v. Chúng tôi duyệt qua tập dữ liệu BinaryCorp và trích xuất các hàm từ mỗi chương trình nhị phân. Đối với mỗi hàm, chúng tôi trích xuất các loại opcode, loại toán hạng, trạng thái đọc/ghi toán hạng và trạng thái thanh ghi EFLAGS trong các hướng dẫn của nó. Trong quá trình này, các hàm quá nhỏ (dưới 5 lệnh) sẽ bị lọc ra.

4 Thực hiện

Để đạt được khả năng mở rộng tốt hơn, Trans không dựa vào phân tích chương trình hạng nặng. Nó chỉ yêu cầu sử dụng một công cụ tháo gỡ tùy ý (ví dụ IDA Pro [19]) để lấy các hướng dẫn chức năng và truy xuất kiến thức như loại opcode và loại toán hạng dựa trên ISA.

Khắc phục vấn đề OOV Để khắc phục vấn đề Out-Of-Vocabulary (OOV), sau khi lấy được ranh giới hàm trong chương trình nhị phân, chúng tôi sử dụng Capstone [38] để phân tách các hướng dẫn và trích xuất kiến thức rõ ràng. Dựa trên ISA, có một số lượng hữu hạn các trạng thái cho loại opcode, loại toán hạng, trạng thái đọc/ghi toán hạng và trạng thái đăng ký EFLAGS. Vì vậy, chúng ta chỉ cần chuẩn hóa văn bản hướng dẫn lắp ráp. Chúng tôi áp dụng ba chiến lược chuẩn hóa: (1) coi các từ ghi nhớ và toán hạng là mã thông báo, (2) thay thế tất cả các hằng số bằng "const" và (3) xóa dấu phẩy giữa các toán hạng.

Tóm lại, kích thước từ vựng (bao gồm cả mã thông báo đặc biệt) của mã thông báo lắp ráp là 15.511 và kích thước từ vựng (bao gồm mã thông báo đặc biệt) của các loại kiến thức khác nhau được thể hiện trong Bảng 2.

Bảng 2: Kích thước từ vựng.

từ vựng	Kích cỡ
Mã thông báo hội	15.511
Loại lệnh	1.002
Loại toán hạng	12
Toán hạng R/W	9
Trạng thái CỜ	56

5 Sự đánh giá

5.1 Phương pháp đánh giá

Trong quá trình đánh giá, chúng tôi giải quyết bốn câu hỏi nghiên cứu sau:

- RQ1: Làm thế nào Transt thực hiện bằng ngôn ngữ lắp ráp mô hình? (nhìn thấy 5.3)
- RQ2: Tác động của Thiết kế Nhận thức Tri thức là gì? (nhìn thấy 5.4)
- RQ3: Chất lượng của các phần nhúng được tạo ra như thế nào kTrans? (nhìn thấy 5.5)
- RQ4: Có thể kTrans cải thiện hiệu quả việc thực hiện các nhiệm vụ hạ nguồn? (nhìn thấy 5.6)

Để trả lời RQ1, chúng tôi tính toán độ phức tạp của kTrans và so sánh nó với một số đường cơ sở. Để trả lời RQ2, chúng tôi tiến hành các thí nghiệm cắt bỏ để đánh giá tác động của việc tích hợp kiến thức rõ ràng và tích hợp kiến thức tiềm ẩn một cách riêng biệt. Để trả lời RQ3, chúng tôi thực hiện đánh giá định lượng bằng cách sử dụng phát hiện ngoại lệ và đánh giá định tính bằng cách sử dụng trực quan hóa t-SNE. Cuối cùng, để trả lời RQ4, chúng tôi áp dụng kTrans vào loại nhiệm vụ phân tích mã nhị phân: nhận dạng độ tương tự mã nhị phân, phục hồi loại chức năng và nhận dạng cuộc gọi gián tiếp. Chúng tôi đánh giá hiệu suất của phương pháp học không cần bản (trực tiếp sử dụng các phần nhúng được tạo bởi kTrans) cũng như hiệu suất sau khi tinh chỉnh kTrans.

5.2 Thiết lập đánh giá

5.2.1 Bộ dữ liệu

Để đào tạo trước kTrans, chúng tôi chọn tập dữ liệu BinaryCorp, bao gồm một tập hợp lớn các chương trình nhị phân thu được thông qua quá trình biên dịch tự động. Bộ dữ liệu chứa tổng cộng 48.130 chương trình nhị phân, khoảng 26 triệu hàm nhị phân, bao gồm hai trình biên dịch (gcc và clang) và năm mức tối ưu hóa biên dịch (O0-O3, Os). Số liệu thống kê chi tiết về các tập huấn luyện và kiểm tra được cung cấp trong Bảng 3.

Bảng 3: Thống kê về số lượng dự án, nhị phân và chức năng của bộ dữ liệu.

Bộ dữ liệu	# dự án	# nhị phân	# Chức năng
Tàu lửa nhị phân	7,845	38,455	21,085,338
Kiểm tra tập đoàn nhị phân	1,974	9,675	4,791,673

So với các bộ dữ liệu như BinKit [25] và Cisco [33], BinaryCorp bao gồm số lượng dự án lớn hơn đáng kể, khiến nó trở thành một tập dữ liệu đa dạng và thực tế hơn. Kết quả là nó phù hợp hơn để đánh giá khả năng mở rộng và hiệu quả của các mô hình ngôn ngữ mã nhị phân.

5.2.2 Đường cơ sở trong mô hình hóa hội

Về mặt lắp ráp mô hình, chúng tôi so sánh kTrans với hai đường cơ sở đại diện:

- Đường cơ sở không dựa trên máy biến áp: word2vec [37]. Bằng cách xử lý các từ ghi nhớ và toán hạng dưới dạng mã thông báo, chúng ta có thể có được các phần nhúng của chúng bằng cách sử dụng word2vec. Việc nhúng lệnh lắp ráp và việc nhúng chuỗi lệnh có thể thu được bằng cách lấy mức trung bình của các lần nhúng mã thông báo của chúng tương ứng.
- Đường cơ sở dựa trên máy biến áp: PalmTree [29]. PalmTree cung cấp các phần nhúng ở cấp độ hướng dẫn bằng cách sử dụng Transformer. Để đảm bảo tính công bằng, chúng tôi đã đào tạo lại mô hình PalmTree trên bộ dữ liệu BinaryCorp. Tương tự, việc nhúng chuỗi lệnh có thể thu được bằng cách lấy mức trung bình của các lần nhúng lệnh.

5.2.3 Cấu hình hệ thống

kTrans bao gồm 12 lớp Transformer với chiều ẩn là 768. Số lượng đầu trong cơ chế chú ý nhiều đầu được đặt thành 8. Trong quá trình đào tạo trước, tỷ lệ khởi động là 0,01, tốc độ học ban đầu là $3e-5$ và tốc độ học giảm dần là 0,01.

Tất cả các thử nghiệm được thực hiện trên máy chủ Linux chạy Ubuntu 18.04 với CPU Intel(R) Xeon(R) Gold 6154 @ 3.00GHz, RAM 512GB và 4 GPU NVIDIA V100.

5.3 RQ1: Làm thế nào Transt thực hiện bằng ngôn ngữ lắp ráp mô hình?

Chúng tôi sử dụng sự bối rối để đánh giá hiệu suất của từng mô hình trong quá trình lắp ráp mô hình. Tất cả các mô hình được đào tạo trong 3 kỷ nguyên trên tập huấn luyện của tập đoàn nhị phân và sau đó độ phức tạp được tính toán trên tập kiểm tra.

Độ phức tạp được định nghĩa là lũy thừa của xác suất logarit trung bình trên mỗi ký hiệu của bộ kiểm tra, được chuẩn hóa bằng số lượng ký hiệu:

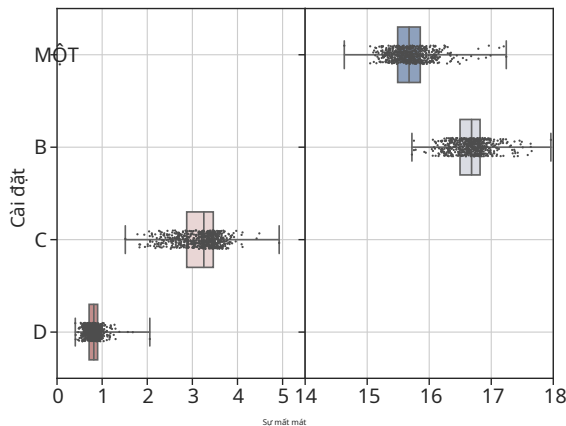
$$PPL = 2^{-\frac{1}{N} \sum_{i=1}^N \log_2(P(x_{T(i)}))}$$

trong đó N là tổng số mã thông báo trong bộ thử nghiệm và $P(x_{T(i)})$ biểu thị xác suất được mô hình ngôn ngữ gán cho mã thông báo $x_{T(i)}$. Sự bối rối đo lường mức độ ngạc nhiên của mô hình ngôn ngữ khi nó gặp các mã thông báo mới. Độ phức tạp thấp hơn cho thấy mô hình dự đoán mã thông báo tiếp theo tốt hơn, trong khi độ phức tạp cao hơn cho thấy mô hình không chắc chắn hơn và kém chính xác hơn trong các dự đoán của nó. Về mặt lý thuyết, giá trị thấp nhất có thể có của độ phức tạp là 1 và giá trị độ phức tạp càng gần 1 thì hiệu suất của mô hình trong việc lắp ráp mô hình càng tốt.

Như thể hiện trong bảng 4, word2vec có độ phức tạp cao nhất, tiếp theo là PalmTree với độ phức tạp thấp hơn một chút và kTrans có mức độ bối rối thấp nhất, $1 + 3.30e-4$. Vì thế, kTrans thực hiện tốt nhất trong việc lắp ráp mô hình.

Bảng 4: Đánh giá độ phức tạp.

Người mẫu	PPL-1
word2vec	$9.01e-3$
Cây cọ	$9.05e-3$
kTrans	$3.30e-4$



Hình 5: Sự phân bố tổn thất trên bộ thử nghiệm trong các cài đặt khác nhau.

5.4 RQ2: Tác động của Kiến thức-

Nhận thức Thiết kế?

Gửi định giá thứ để đóng góp quan điểm của kn cú tôi tiêm t ò các
model's capab Tôi, chúng tôi dẫn như e của ablati ò trải nghiệm tont
Và đánh giá t người mẫu điện tử biểu diễn nce của ablati ò trải nghiệm tont
Nhược điểm đáng nghĩ về c sự cắt bỏ chi phí cuối cùng, nơi ẩn náu n lớp di đản ông-
sion của tháng dels trong thứ e sự cắt bỏ thử nghiệm Ns được đặt ò 128
tới SP ăn xong rồi Một Chúng tôi so sánh sự mất mát f bốn khác biệt ethuê
định cư gs: (A) Ò Tr gốc Một Chúng tôi cung cấp thông (B) Chuyển giới trí thông minh trước đây
plici tôi không biết gđ; (CTR Một Với hàm ý Tôi edge;
(D) T tiền chuộc ở với người yêu cũ hợp pháp và tôi kiến thức rõ ràng Owledge.
Qua so sánh Ng (A, B) Một (C, D), chúng ta có thể như Xin chào tôi lập ước
của người biết rõ ràng wgo. Sĩ tôi, bởi so sánh g (A, C) một N (B,
D), w tôi có thể trốn tránh chúc mừng e ác dụng của tôi biết rõ ràng cái nềm.
Một N Nhân vật 5 và bảng 5, cho một, C) và (B, Đ),
phần giới thiệu phục tôi tôi rõ ràng ò nềm r e kết thúc trong một ý nghĩa N-
nếp gấp e trong tadel's lo Ss, chỉ thị ng đó trong Corporatin tôi
plici tôi không biết đi qua h cái KI t hồi lon Tôi kinh ngạc y en-
han và mod el's capab sự linh hoạt, trong trường hợp của Compariso n (A,
Ba đing ex Biết rõ ràng nềm le Một kiến một si tăng chiến đấu tham gia
tôi ôi, có thể bị do bởi v e giới thiệu thêm vào ò kiến thức cuối cùng bởi rìa
trong đó và MLM t hồi một mình có thể cau bạn quá phù hợp ting. Hồ w bao giờ,
cho compariso N (C, D), một giới thiệu sau cing K Tôi làm nhiệm vụ, B tiếp pháp
biết dấu hiệu gở một cách nhanh chóng, làm giảm người mẫu mất mát, chỉ số ăn uống
cái đó sự kết hợp quốc gia của e rõ ràng k Nhanh cú Một KI nhiệm vụ
có thể g thực sự đẩy h trước đó tôi của odel khả năng.

Bảng 5: Tổn thất trung bình trên bộ thử nghiệm ở các cài đặt khác nhau.

Cài đặt	giải thích Kiến thức	ngầm Kiến thức	Sự đánh giá Sự mất mát
MỘT	\times	\times	15,8
B	\checkmark	\times	16,7
C	\times	\checkmark	3.11
D	\checkmark	\checkmark	0,858

Nhìn chung, những kết quả này cho thấy rằng việc kết hợp kiến thức bên ngoài vào mô hình hóa trình tự hợp ngữ là rất có lợi cho mô hình của chúng tôi.

5.5 RQ3: Chất lượng của các phần nhúng được tạo ra như thế nào kTrans?

5.5.1 Phát hiện ngoại lệ

Phát hiện ngoại lệ là một phương pháp đánh giá định lượng để đánh giá chất lượng của các phần nhúng. Chính thức, được cấp một bộ mã thông báo $T = t_1, t_2, \dots, t_N, t_{N+1}$, giả sử t_1, \dots, t_N thuộc cùng một cụm và t_{N+1} là ngoại lệ, nhiệm vụ nhằm xác định ngoại lệ không thuộc cùng nhóm với các mã thông báo còn lại.

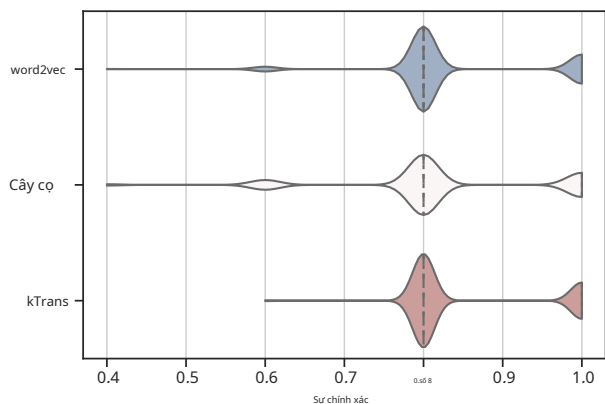
Hiệu suất của nhiệm vụ phát hiện ngoại lệ có thể được đánh giá bằng độ chính xác. Lý tưởng nhất là nếu các ngoại lệ trong tất cả các nhóm được xác định chính xác thì độ chính xác phải là 1. Sau khi thiết lập trong PalmTree, chúng tôi đánh giá cả khả năng phát hiện ngoại lệ opcode và hoạt động và ngoại lệ de sự tiếp xúc.

Chúng tôi hướng dẫn một trong tập cấu trúc f rom bina ries trong tập kiểm tra một mẫu thứ 50, 000 nhóm dữ liệu. cho opca ngoại lệ phát hiện n, mỗi nhóm nhược điểm của dữ liệu Tôi bốn trong công trình với cùng một opcode và một lần cuộc đấu tranh với một sự khác biệt opcode. Đối với toán hạng phát hiện ngoại lệ Tôi, mỗi nhóm dữ liệu lên bao gồm của bốn hướng dẫn C quan hệ với thứ e cùng một vở opera d loại và một lần tương tác với một hoạt động khác nhau Một thứ.

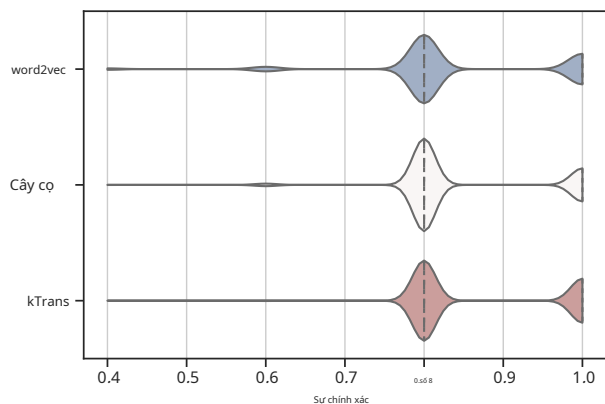
Bàn 6 cho thấy độ chính xác trung bình acy của ngoại lệ phát hiện cho mỗi người mẫu. Hình 6 hiển thị t h phân phối điện tử của ngoại lệ phát hiện n độ chính xác cho r Mỗi mô hình. Như được hiển thị, k Dịch vẫn đạt được S sự hoàn hảo nhất vũ khí, trí thông minh h độ chính xác trong số 86,1% cho opco de det outlier e hành động và 86. 56% dành cho opera thử ngoại lệ phát hiện N Cây cọ p e nhậm chỉ còn có hình dạng worse hơn wo thứ 2vec bất nhiệm vụ này .

Bảng 6: Acc trung bình sự khác biệt nt mô hình cho ngoại lệ de- sự tiếp xúc. Một opcode d biểu thị ave cơn thịnh nộ chính xác mã lệnh ngoại lệ d sự phát hiện và Một opnd den ò es mức trung bình độ chính xác điện tử của vở opera thử nd ngoại lệ e hành động.

Người mẫu	Mã Acc	Acc-opnd
word2vec	0,840	0,846
Cây cọ	0,829	0,847
kTrans	0,861	0,866

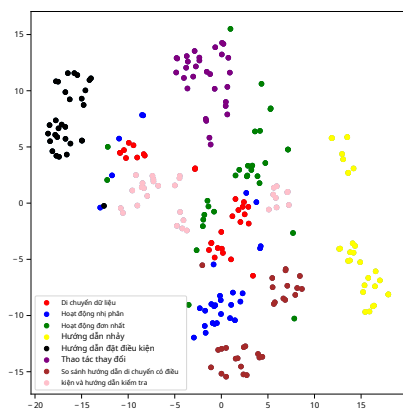


(a) Mã lệnh

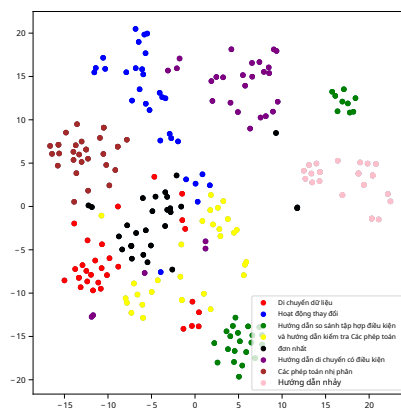


(b) Toán hạng

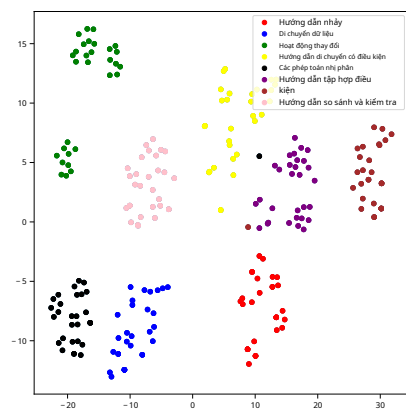
Hình 6: Độ chính xác của việc phát hiện ngoại lệ.



(a) word2vec



(b) Cây cọ



(c) kTrans

Hình 7: trực quan hóa t-SNE của các phần nhúng.

5.5.2 Phân tích định tính thông qua trực quan hóa t-SNE

Chúng tôi cũng quan sát sự phân bố của các phần nhúng hướng dẫn và phân tích chất lượng của các phần nhúng bằng cách sử dụng hình ảnh trực quan.

Nhúng hàng xóm ngẫu nhiên phân phối t (t-SNE) là một phương pháp thống kê giúp trực quan hóa dữ liệu nhiều chiều bằng cách gán cho mỗi điểm dữ liệu một vị trí trên bản đồ hai hoặc ba chiều. Chúng tôi xây dựng một tập lệnh từ các tập nhúng phân phối t trong tập kiểm tra và lấy mẫu 30 trường hợp từ mỗi lớp trong số tám lớp lệnh chung. Sau đó, chúng tôi sử dụng t-SNE để quan sát sự phân phối các phần nhúng của chúng.

Như thể hiện trong hình 7, so với word2vec và PalmTree, các phần nhúng được tạo bởi kTrans thể hiện ranh giới lớp rõ ràng hơn và phân bố dày đặc hơn trong cùng một lớp. Trong khi PalmTree không phân biệt được giữa "Data Movement", "Hướng dẫn so sánh và kiểm tra" và "Hoạt động đơn nhất" và word2vec kết hợp "Di chuyển dữ liệu" và "Hướng dẫn di chuyển có điều kiện" với nhau.

Thông qua cả phân tích định lượng và định tính, chúng ta có thể kết luận rằng kTrans có thể tạo ra mã nhị phân chất lượng cao

nhúng vượt qua pr

công việc hiển nhiên.

5.6 RQ4: Có thể cải thiện hiệu quả các hiệu suất của nhiệm vụ tiếp theo?

5.6.1 Mã nhị phân Simil phát hiện tính chất

Độ giống nhau của mã nhị phân phân phối t (BCSD) nhằm mục đích xác định mức độ giống nhau giữa định hai đoạn mã nhị phân. chúng

Dựa trên BinaryCorp, tôi đánh giá hiệu suất của word2vec kTrans trên BCSD. Ngoài ra còn có PalmTree, chúng tôi tiếp cận kTrans với giải pháp BCSD Gemini và tions jTrans dựa trên N. tiên tiến nhất của GN được sử dụng. Việc đánh giá đã đáp ứng Xếp hạng được đối ứng) và Recall@k, xác định như sau:

$$\text{Nhớ lại}@k = \frac{1}{|F|} \sum_{f \in F} \text{TỔ I}(\text{Thứ hạng}_{f_{gt} \leq k}^{\text{Tổ I}})$$

$$\text{MRR} = \frac{1}{|F|} \sum_{f \in F} \frac{1}{\text{Thứ hạng}_{f_{gt}^{\text{Tổ I}}}}$$

Bảng 7: Kết quả của các phương pháp phát hiện sự tương đồng nhị phân khác nhau trên BinaryCorp (Poolsize=32).

Người mẫu	MRR							Nhớ lại@1						
	O0,O3	O1,O3	O2,O3	O0,Os	O1,Os	O2,Os	Trung bình	O0,O3	O1,O3	O2,O3	O0,Os	O1,Os	O2,Os	Trung bình
Song Tử	0,402	0,643	0,835	0,469	0,564	0,628	0,590	0,263	0,528	0,768	0,322	0,441	0,518	0,473
word2vec	0,433	0,841	0,952	0,474	0,778	0,849	0,721	0,315	0,789	0,936	0,352	0,716	0,798	0,651
jTrans-zero	0,594	0,841	0,962	0,649	0,850	0,891	0,797	0,499	0,803	0,945	0,566	0,808	0,853	0,746
kTrans-số không	0,537	0,951	0,981	0,570	0,908	0,910	0,810	0,402	0,932	0,973	0,435	0,877	0,881	0,750
Cây cọ	0,688	0,873	0,956	0,729	0,850	0,904	0,833	0,567	0,813	0,936	0,618	0,777	0,859	0,762
jTrans	0,947	0,976	0,985	0,956	0,979	0,977	0,970	0,913	0,960	0,974	0,927	0,964	0,961	0,949
kTrans	0,975	0,991	0,996	0,984	0,992	0,991	0,988	0,956	0,984	0,993	0,971	0,985	0,984	0,979

Bảng 8: Kết quả của các phương pháp phát hiện sự tương đồng nhị phân khác nhau trên BinaryCorp (Poolsize=10.000).

Người mẫu	MRR							Nhớ lại@1						
	O0,O3	O1,O3	O2,O3	O0,Os	O1,Os	O2,Os	Trung bình	O0,O3	O1,O3	O2,O3	O0,Os	O1,Os	O2,Os	Trung bình
Song Tử	0,072	0,189	0,474	0,069	0,147	0,202	0,192	0,058	0,148	0,420	0,051	0,115	0,162	0,159
word2vec	0,119	0,423	0,693	0,097	0,369	0,446	0,358	0,099	0,367	0,628	0,086	0,307	0,403	0,315
jTrans-zero	0,215	0,570	0,759	0,233	0,571	0,563	0,485	0,167	0,503	0,701	0,175	0,507	0,500	0,426
kTrans-số không	0,129	0,702	0,835	0,111	0,645	0,664	0,514	0,105	0,633	0,789	0,090	0,538	0,610	0,461
Cây cọ	0,130	0,403	0,677	0,152	0,355	0,496	0,369	0,083	0,326	0,609	0,097	0,281	0,420	0,303
jTrans	0,584	0,734	0,792	0,627	0,709	0,710	0,693	0,499	0,668	0,736	0,550	0,648	0,648	0,625
kTrans	0,588	0,800	0,870	0,657	0,777	0,777	0,745	0,493	0,732	0,820	0,565	0,712	0,716	0,673

Ở đây A là một nhóm hàm nhị phân, G là nhóm hàm nhị phân thực tế, T là chức năng chỉ báo. Và chúng tôi biểu thị một hàm truy vấn là $f: \mathcal{O} \rightarrow \mathcal{F}$ và sự thật nền tảng tương ứng của nó hoạt động như $f: \mathcal{O} \rightarrow \mathcal{G}$.

Như thể hiện trong bảng 7, với kích thước bể bơi là 32, kTrans đạt được MRR trung bình là 0,988 và Recall@1 trung bình là 0,979, vượt qua tất cả các mẫu khác. Trong kịch bản không bắn, ngoại trừ (O0, O3), kTrans vượt trội hơn jTrans zero-shot trong tất cả các cài đặt khác. PalmTree hoạt động tốt hơn word2vec và Gemini nhưng chỉ vượt trội hơn zero-shot kTrans bằng 0,01-0,02, kém xa jTrans và kTrans. Khi kích thước nhóm lên tới 10.000 (Bảng số 8), kTrans vẫn có hiệu suất tốt nhất trong số tất cả các phương pháp với MRR là 0,745 và Recall@1 là 0,673, trong khi MRR của PalmTree giảm mạnh xuống 0,369, khiến nó không thực tế trong kịch bản BCSD trong thế giới thực.

Chúng tôi cũng nghiên cứu mối quan hệ giữa quy mô nhóm và hiệu suất của các phương pháp BCSD khác nhau. Như thể hiện trong hình số 8, kTrans đạt được hiệu suất tốt nhất trong mọi cài đặt và thậm chí kTrans-zero có thể hoạt động tốt hơn jTrans trong (O1, O3) và (O2, O3). Khi quy mô nhóm lên tới 10.000, kTrans-zero cũng có thể hoạt động tốt hơn jTrans trong (O1, Os) và (O2, Os).

Do đó, những kết quả này chứng minh rằng việc kết hợp kiến thức bên ngoài vào phần nhúng có thể nâng cao hiệu suất của nhiệm vụ BCSD.

5.6.2 Phục hồi loại chức năng

Phục hồi kiểu hàm (FTR) nhằm mục đích xác định số lượng và loại đối số cho một hàm trong mã nhị phân.

Chúng tôi so sánh kTrans, PalmTree và EKLAVYA trong nhiệm vụ này. EKLAVYA sử dụng word2vec làm phương pháp nhúng lệnh và dự đoán các loại hàm bằng cách sử dụng Recurrent

Bảng 9: Kết quả của các phương pháp khôi phục loại hàm khác nhau.

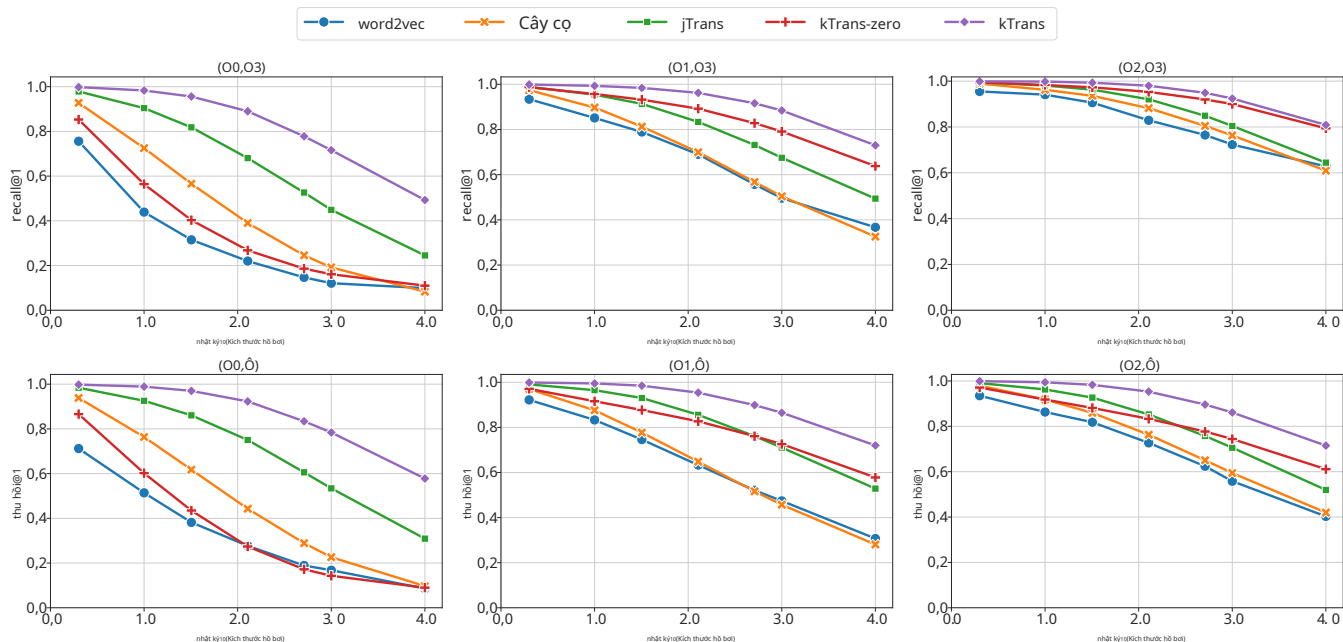
Người mẫu	Độ chính xác của tàu	Kiểm tra độ chính xác
EKLAVYA (word2vec)	0,987	0,809
Cây cọ	0,946	0,808
kTrans-số không	0,939	0,856
kTrans	0,948	0,876

Mạng thần kinh (RNN). Do đó, chúng tôi thay thế phương pháp nhúng lệnh bằng kTrans và PalmTree để so sánh.

Bộ dữ liệu được sử dụng trong thử nghiệm này được cung cấp bởi EKLAVYA và bao gồm 8 dự án và 2.312 tập nhị phân. Trong các bài báo của EKLAVYA và PalmTree, họ sử dụng phân tách nhị phân để chia tập dữ liệu thành các tập huấn luyện và kiểm tra, điều này có thể dẫn đến sự chồng chéo giữa hai bộ do mã được chia sẻ giữa các nhị phân khác nhau trong cùng một dự án. Do đó, chúng tôi thực hiện phân chia mới dựa trên các dự án, chọn ngẫu nhiên 6 dự án làm tập huấn luyện và 2 dự án còn lại làm tập thử nghiệm. Số liệu đánh giá được sử dụng là độ chính xác của dự đoán loại hàm.

Như thể hiện trong bảng 9, kTrans đạt độ chính xác cao nhất trên bộ thử nghiệm là 87,6%. PalmTree và EKLAVYA (word2vec) đạt được hiệu suất tương tự ở mức 80%, nhưng vẫn kém hơn so với zero-shot kTrans. Ngoài ra, bằng cách so sánh độ chính xác giữa tập huấn luyện và tập kiểm tra, chúng tôi nhận thấy rằng PalmTree và EKLAVYA bị trang bị quá mức đáng kể.

Do đó, những kết quả này chỉ ra rằng việc kết hợp kiến thức bên ngoài vào phần nhúng có thể nâng cao hiệu suất của nhiệm vụ FTR.



Hình 8: Màn trình diễn tài xuất hiện của các mô hình khác nhau cho phát hiện sự tương tự mã nhị phân về kích thước hồ bơi.

Tkhai năng 10: Kết quả khác nhau tài để nhận dạng cuộc gọi gián tiếp (Kích thước hồ bơi=2).

Người mẫu	MRR	Nhớ lại@1
CALLEE (doc2vec)	0,777	0,553
word2vec	0,754	0,540
Cây cọ	0,765	0,530
kTrans-số không	0,790	0,580
kTrans	0,808	0,616

Bảng 11: Kết quả của sự khác biệt mô hình phong phú cho cuộc gọi gián tiếp nhận ra ni- (Kích thước hồ bơi=32).

Người mẫu	MRR	Nhớ lại@1
CALLEE (doc2vec)	0,172	0,057
word2vec	0,155	0,051
Cây cọ	0,154	0,046
kTrans-zero	0,231	0,094
kTrans	0,280	0,133

5. 6.3 Nhận dạng cuộc gọi gián tiếp sự

Tôi nhận dạng cuộc gọi gián tiếp (tôi C) nhằm mục đích xác định tiềm năng tôi callee của hàm gián tiếp cal Is ở dạng mã nhị phân.

Chúng tôi so sánh kTrans, Palm Tree và CALLEE trong phần k. CALLEE sử dụng doc2vec [27] Một là trình tự tập hợp của - phương pháp edding và thực hiện callsite-callee khớp với Một Smạng lưới thần kinh iamese. trước hết, chúng tôi thay thế tập y Có phương pháp nhúng đẳng hợp bằng kTrans, word2vec, một d Pthức almTree để so sánh.

Chúng tôi xây dựng một tập dữ liệu cho nhiệm vụ ICR dựa trên SPE C CPU 2006. Bằng cách chạy thử nghiệm ở SPEC CPU 2006 và sử dụng g hành vi không theo thời gian của công Vi vậy, chúng tôi ghi lại sự thật cơ bản của S rnhệ thiết bị đo động, bao gồm tổng số chương trình đối với các cuộc gọi gián S Tôi chúng tôi đã thu thập được 52.062 Niếp, các cuộc gọi trực tiếp từ 21 dự án.

Thay vì sử dụng Precision/ Recall/F1 của phân loại nhị phân S tion, chúng tôi áp dụng tw sau o thước đo toàn diện hơn S

$$\text{Nhớ lại}@k = \frac{1}{|F_{\text{callee}}|} \sum_{T \in F_{\text{callee}}} \text{TÔI}(\text{Thứ hạng}_{T \in F_{\text{callee}}}^{\text{Tôi}})$$

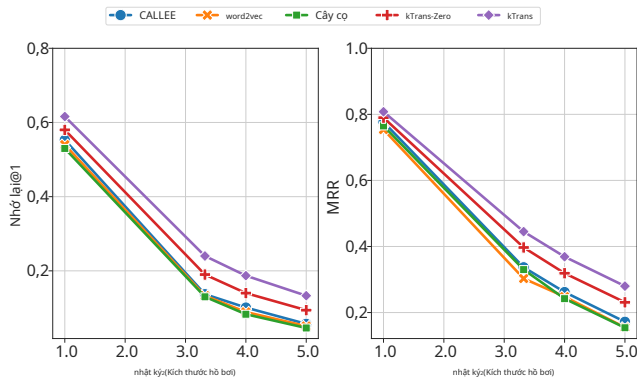
$$\text{MRR} = \frac{1}{|F_{\text{callee}}|} \sum_{T \in F_{\text{callee}}} \frac{1}{\text{Thứ hạng}_{T \in F_{\text{callee}}}^{\text{Tôi}}}$$

Ở đâu F_{callee} là một hồ hồ bơi, G_{callee} là mặt đất tru th cal Lee callee, $T \in F_{\text{callee}}$ chỉ số f là f unction. Và chúng tôi biểu thị một ry cal Lee tại $T \in F_{\text{callee}}$ và c của nó ầu trả lời cho sự thật cơ bản fu N hành động BẢNG tại $T \in G_{\text{callee}}$.

Như thể hiện trong Bảng 10 và bảng 11, kTransachi ool có đêm trước Anh ta hiệu suất tốt nhất với P kích thước là 2 và 32. Chúng tôi examine mỗi quan hệ giữa ecũng n MRR, Recall@1 và pool kích cỡ vì Mỗi mô hình. Như miêu tả d bảng số 9, như kích thước hồ bơi tôi Nếp gấp ừ, có sự suy giảm ở M RR và Recall@1 cho tất cả các tháng dels, Nhưng kTrans duy trì s của nó banviệt hơn các mô hình khác.

Vi vậy, những lý do này Điều đó chỉ ra rằng việc kết hợp bên ngoài cuối cùng kiến thức về em giường ngủ có thể nâng cao hiệu quả ormance của nhiệm vụ ICR.

Tóm lại là, ktra N có thể cải thiện hiệu quả e hoàn hảo hoặc sức mạnh của hạ lưu các nhiệm vụ như BCSD, FTR, d ICR.



Hình 9: Hiệu suất của các mô hình khác nhau để nhận dạng cuộc gọi gián tiếp liên quan đến kích thước nhóm.

6 định hướng tương lai cho việc nhúng mã nhị phân-định

6.1 Mô hình miền lớn hơn

Như thể hiện trong 5.6, mẫu Transformer hiện tại vẫn còn chỗ dành cho những người cải tiến trong các nhiệm vụ phân tích khó khăn như ICR. theo s Luật tăng dần [21] của Transformers, mạng lan lớn-mô hình đo có quy mô tốt hơn với các mô hình lớn hơn và nhiều dữ liệu hơn. Do đó, bằng cách tận dụng những tiến bộ trong công nghệ đào tạo trước, Nq es, chẳng hạn như AMP [36] và DeepSpe ed [47], chúng ta có thể ăn tồn để độ mạnh mẽ ls cụ thể là tại cầu thu thập dữ liệu ca ngời. T Anh Se tên miền lớn hơn mô hình có thể nời en phân lần đầu tiên tôi dùng Tời mô hình cate và phụ thuộc w Tời mông các nhiệm vụ ca ngời, tôi đang được cải thiện Phiệu suất trong va quan trọng nhị phân.

6.2 Chi phí ảnh hưởng Tới người mẫu

TraNinh và chạy g lan quy mô lớn g mô hình uage ca N là cồ kinh nghiệm chính thức Nanh mẽ và tài nguyên e-căng. Ở đó f, quảng, expl hiệu quả chi phí quảng Tầm kỹ thuật cho r mã nhị phân em giườg, đing là một sự tái quan trọng hướng tìm kiếm. Ồ không có khả năng ồioi Tời để điều tra ma túy đã od để chuyển l kiểm tiền, trong đó một trước traN mô hình ed trên một ge mục đích chung Tập dữ liệu điện tử rá bared Ồ không nhị phân nhỏ hơn-s P tập dữ liệu hiệu quả Qua tốt khi tận dụng kn con cú-edge học được từ thứ đó e miền chung, người mẫu có thể p Ồ mười-tài tất cả đạt được tốt hơn Phiệu suất với ít tính toán hơn Ồ cuối cùng resources. Một cái khác cách tiếp cận là đào P truyền thuyết giáo viên-student Parào cho hiệu quả t tạo người mẫu cũ. T phương pháp của anh ấy lves traN ng một cái nhỏ hơn, si tồ h mô hình pler (các Shọc sinh) để học f rom Perfrang bị cho các trầ udent với h b ít tính toán hơn Ồ cuối cùng resources. Những cos này t tập luyện hiệu quả g phương pháp sẽ tôi gười ake t, nhữ phát triển và triển khai bi Nhúng mã ary ding tồo els có nhiều quyền truy cập hơn b, ừ các nhà nghiên cứu và những người thực hành.

6.3 Kết hợp với các mô hình ngôn ngữ lớn chung

Các mô hình ngôn ngữ lớn chung, chẳng hạn như GPT-4 [39], đã chứng tỏ khả năng vượt trội trong việc hiểu và tạo ngôn ngữ tự nhiên. Việc kết hợp các mô hình ngôn ngữ lớn chung này với việc nhúng mã nhị phân có thể dẫn đến các mô hình thậm chí còn mạnh mẽ hơn để hiểu và phân tích mã nhị phân. Bằng cách tận dụng điểm mạnh của cả hai loại mô hình, chúng ta có thể hưởng lợi từ khả năng hiểu ngữ cảnh và mô hình hóa ngôn ngữ của các mô hình ngôn ngữ lớn, đồng thời kết hợp kiến thức miền cụ thể được mã hóa trong việc nhúng mã nhị phân. Sự kết hợp này có thể cho phép phạm vi ứng dụng rộng hơn, chẳng hạn như tổng hợp mã, phát hiện lỗi hổng và hiểu chương trình. Tuy nhiên, điều quan trọng là phải giải quyết những thách thức trong việc sắp xếp kiến thức miền cụ thể của mã nhị phân với kiến thức tổng quát hơn về các mô hình ngôn ngữ lớn, đảm bảo kết quả chính xác và đáng tin cậy trong ngữ cảnh mã nhị phân.

7. Kết luận

Trong bài báo này, chúng tôi đề xuất kTrans, nhúng mã nhị phân mô hình ding kết hợp kiến thức bên ngoài. Chúng tôi tận dụng Kiến trúc dựa trên máy biến áp để nắm bắt ngữ nghĩa của hướng dẫn lắp ráp và sử dụng các kỹ thuật đưa kiến thức vào để nâng cao hiệu suất của mô hình. Thông qua mở rộng-ive ev Một đa dạng tasks và bộ dữ liệu, kTrans là-a-sistentl yo làm tốt hơn ba self-ai mô hình, hiển thị bao hàm hiệu quả của nó hư hổng Nừ trong hội đồng ngu hiểu tuổi g. Kết quả con quý Strat vượt trội mỗi vì tence trong cry nhị Ồ sự giống nhau phát hiện Ồn, foại ghi chú trườg phân và ca t tiềm gián sẽ công nhận nhiệm vụ. T Anh tanh ừg phát hiện gợi ý thứ đó Một từ bên ngoài tất cả kiến thức vào mông em ngôn ngữ ngớ ngẩn m Ồ thật đáng kể là tôi tôi ứng minh của họ khả năng nóS. Hơn nữa, we thảo luận về tương lai r eim kiểm trực tiếp các vấn đề trong cluding khám phá la r genô hình miền r, tiết kiệm chi phí mô hình , Một d kết hợp mưu m Ờng các vấn đề lớn nói Một age mod-els. Qua quầng mắ quần áo này giần c, chúng tôi có thể bảo có quảng bá tập hợp ly l mô hình ngôn ngữ ng f hoặc bảo mật phần mềm Tời và nhị phân mã một N an thàh ời.

Có sẵn Một kTrans

kTrans là công khai có sẵn le.

Tham khảo eNes

- [1] Wasiuddin Ahmad, S aiko Chakraborty, Baishakhi Ray, Một Kai-Wei Chang. b, luyện tập trước g cho chương trình b, đứng và ge neration. ArXiv, ab S/2103.06333, 2021.

- [2] Miltiadis Allamanis, Marc Brockschmidt và Mahmoud Khademi. Học cách biểu diễn chương trình bằng đồ thị. *ArXiv*, abs/1711.00740, 2017.
- [3] Uri Alon, Shaked Brody, Omer Levy và Eran Yahav. code2seq: Tạo chuỗi từ các biểu diễn mã có cấu trúc. *bản in trước arXiv arXiv:1808.01400*, 2018.
- [4] Uri Alon, Meital Zilberstein, Omer Levy và Eran Yahav. code2vec: Học cách biểu diễn mã phân tán. *Kỷ yếu của ACM về Ngôn ngữ lập trình*, 3(POPL):1–29, 2019.
- [5] Fiorella Artuso, Marco Mormando, Giuseppe A Di Luna và Leonardo Querzoni. Binbert: Hiểu mã nhị phân với một máy biến áp có thể điều chỉnh và nhận biết thực thi. *bản in trước arXiv arXiv:2208.06692*, 2022.
- [6] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quốc Lê, và những người khác. Tổng hợp chương trình với mô hình ngôn ngữ lớn. *bản in trước arXiv arXiv:2108.07732*, 2021.
- [7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever và Dario Amodei. Các mô hình ngôn ngữ là những người học ít lần. Trong H. Larochelle, M. Ranzato, R. Hadsell, MF Balcan và H. Lin, các biên tập viên, *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, tập 33, trang 1877–1901. Hiệp hội Curran, Inc., 2020.
- [8] Zheng Leong Chua, Shiqi Shen, Prateek Saxena và Zhenkai Liang. Mạng lưới thần kinh có thể học chữ ký loại hàm từ các tập nhị phân. TRONG *Hội nghị chuyên đề về bảo mật USENIX*, trang 99–116, 2017.
- [9] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho và Yoshua Bengio. Đánh giá thực nghiệm về mạng lưới thần kinh tái phát có kiểm soát trên mô hình trình tự, 2014.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee và Kristina Toutanova. Bert: Đào tạo trước các máy biến áp hai chiều sâu để hiểu ngôn ngữ. *bản in trước arXiv arXiv:1810.04805*, 2018.
- [11] Steven HH Ding, Benjamin CM Fung và Philippe Charland. Asm2vec: Tăng cường độ mạnh của biểu diễn tĩnh cho tìm kiếm bản sao nhị phân dựa trên mã obfusca-
tối ưu hóa và trình biên dịch. TRONG *Hội nghị chuyên đề IEEE 2019 về bảo mật và quyền riêng tư (SP)*, trang 472–489. IEEE, 2019.
- [12] Thomas Dullien và Rolf Rolles. So sánh dựa trên đồ thị của các đối tượng thực thi (phiên bản tiếng Anh). *Sstic*, 5(1):3, 2005.
- [13] Tiền Phong, Chu Nhuận Đông, Thành Thành Xu, Yao Cheng, Brian Testa và Heng Yin. Tìm kiếm lỗi dựa trên biểu đồ có thể mở rộng cho hình ảnh phân sụn. TRONG *Kỷ yếu Hội nghị ACM SIGSAC 2016 về An ninh Máy tính và Truyền thông*, trang 480–491, 2016.
- [14] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiao Cheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang và Ming Chu. CodeBERT: Một mô hình được đào tạo trước về lập trình và ngôn ngữ tự nhiên. TRONG *Kết quả của Hiệp hội Ngôn ngữ học tính toán: EMNLP 2020*, trang 1536–1547, Trực tuyến, tháng 11 năm 2020. Hiệp hội Ngôn ngữ học Tính toán.
- [15] Debin Gao, Michael K Reiter, và Dawn Song. Binhunt: Tự động tìm kiếm sự khác biệt về ngữ nghĩa trong chương trình nhị phân. TRONG *Hội nghị quốc tế về an ninh thông tin và truyền thông*, trang 238–255. Mùa xuân, 2008.
- [16] Jian Gao, Xin Yang, Ying Fu, Yu Jiang và Jianguang Sun. Vulseeker: công cụ tìm kiếm lỗ hổng dựa trên học tập ngữ nghĩa cho hệ nhị phân đa nền tảng. TRONG *Hội nghị quốc tế IEEE/ACM lần thứ 33 năm 2018 về Kỹ thuật phần mềm tự động (ASE)*, trang 896–899. IEEE, 2018.
- [17] Wenbo Guo, Dongliang Mu, Xinyu Xing, Min Du và Dawn Song. Deepvsa: Hỗ trợ phân tích tập hợp giá trị bằng phương pháp học sâu để phân tích chương trình sau khi chết. TRONG *Hội nghị chuyên đề về bảo mật USENIX*, trang 1787–1804, 2019.
- [18] Xu He, Shu Wang, Yunlong Xing, Pengbin Feng, Hained Wang, Qi Li, Songqing Chen và Kun Sun. Binprov: Nhận dạng xuất xứ mã nhị phân mà không cần tháo gỡ. TRONG *Kỷ yếu Hội nghị chuyên đề quốc tế lần thứ 25 về nghiên cứu tấn công, xâm nhập và phòng thủ*, trang 350–363, 2022.
- [19] Tia lục giác SA. IDA Pro: trình gỡ lỗi và dịch ngược đa bộ xử lý đa nền tảng. [http://www. hex-rays.com/products/ida/index.shtml](http://www.hex-rays.com/products/ida/index.shtml).
- [20] Sepp Hochreiter và Jürgen Schmidhuber. Trí nhớ ngắn hạn dài. *Tính toán thần kinh*, 9(8):1735–1780, 1997.
- [21] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl,

- Aidan Clark và cộng sự. Đào tạo tính toán tối ưu lớn bản các mô hình ngôn ngữ *in trước arXiv arXiv:2203.15556*, 2022.
- [22] Xin Hu, Tzi-cker Chiueh và Kang G Shin. mở rộng việc lập lớn-mục phần mềm độc hại bằng cách sử dụng biểu đồ lệnh gọi hàm. TRONG *Kỷ yếu hội nghị ACM lần thứ 16 về An ninh máy tính và truyền thông*, trang 611–620, 2009.
- [23] Yu Huang, Chenzhuang Du, Zihui Xue, Hiényao Chen, Hang Zhao và Longbo Huang. Điều gì làm cho việc học đa phương thức tốt hơn học đơn phương thức (có thể chứng minh được). *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, 34:10944– 10956, 2021.
- [24] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung và Luke Zettlemoyer. Tóm tắt mã nguồn bằng mô hình chú ý thần kinh. TRONG *Kỷ yếu Hội nghị thường niên lần thứ 54 của Hiệp hội Ngôn ngữ học tính toán (Tập 1: Các bài báo dài)*, trang 2073–2083, Berlin, Đức, tháng 8 năm 2016. Hiệp hội Ngôn ngữ học tính toán.
- [25] Dongkwan Kim, Eunsoo Kim, Sang Kil Cha, Sooel Son và Yongdae Kim. Xem lại phân tích độ tương tự mã nhị phân bằng cách sử dụng kỹ thuật tính năng có thể hiểu được và bài học kinh nghiệm. *Giao dịch của IEEE về Kỹ thuật phần mềm*, trang 1–23, 2022.
- [26] Lingpeng Kong, Cyprien de Masson d'Autume, Wang Ling, Lei Yu, Zihang Dai và Dani Yogatama. Quan điểm tối đa hóa thông tin lẫn nhau của việc học biểu diễn ngôn ngữ. *bản in trước arXiv arXiv:1910.08350*, 2019.
- [27] Quốc Lê và Tomas Mikolov. Đại diện phân phối của câu và tài liệu. TRONG *Hội nghị quốc tế về học máy*, trang 1188–1196. PMLR, 2014.
- [28] Young Jun Lee, Sang-Hoon Choi, Chulwoo Kim, Seung-Ho Lim và Ki-Woong Park. Học mã nhị phân với học sâu để phát hiện điểm yếu của phần mềm. TRONG *Hội nghị chuyên đề quốc tế về Internet lần thứ 9 (ICONI) 2017 KSII*, 2017.
- [29] Xuezixiang Li, Yu Qu, và Heng Yin. Palmtree: Học mô hình hợp ngữ để nhúng lệnh. TRONG *Kỷ yếu Hội nghị ACM SIGSAC 2021 về Bảo mật Máy tính và Truyền thông*, trang 3236–3251, 2021.
- [30] Bingchang Liu, Wei Huo, Chao Zhang, Wenchao Li, Feng Li, Aihua Piao và Wei Zou. adiff: phát hiện sự giống nhau của mã nhị phân phiên bản chéo với dnn. TRONG *Kỷ yếu Hội nghị quốc tế ACM/IEEE lần thứ 33 về Kỹ thuật phần mềm tự động*, trang 667–678, 2018.
- [31] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer và Veselin Stoyanov. Roberta: Phương pháp đào tạo trước bert được tối ưu hóa mạnh mẽ. *bản in trước arXiv arXiv:1907.11692*, 2019.
- [32] Zhenhao Luo, Pengfei Wang, Baosheng Wang, Yong Tang, Wei Xie, Xu Zhou, Danjun Liu và Kai Lu. Vulhawk: Phát hiện lỗ hổng đa kiến trúc bằng tìm kiếm mã nhị phân dựa trên entropy. TRONG *Hội nghị chuyên đề về bảo mật hệ thống phân tán và mạng thường niên lần thứ 30, NDSS 2023, San Diego, California, Hoa Kỳ, ngày 27 tháng 2 - ngày 3 tháng 3 năm 2023*. Hiệp hội Internet, 2023.
- [33] Andrea Marcelli, Mariano Graziano, Xabier Ugarte-Pedrero, Yanick Fratantonio, Mohamad Mansouri và Davide Balzarotti. Học máy đang giải quyết vấn đề tương tự hàm nhị phân như thế nào TRONG *Hội nghị chuyên đề về bảo mật USENIX lần thứ 31 (USENIX Security 22)*, trang 2099–2116, 2022.
- [34] Luca Massarelli, Giuseppe A Di Luna, Fabio Petroni, Leonardo Querzoni và Roberto Baldoni. Điều tra biểu đồ nhúng mạng lưới thần kinh với việc trích xuất các tính năng không được giám sát để phân tích nhị phân. TRONG *Kỷ yếu Hội thảo lần thứ 2 về Nghiên cứu Phân tích nhị phân (BAR)*, 2019.
- [35] Luca Massarelli, Giuseppe Antonio Di Luna, Fabio Petroni, Roberto Baldoni và Leonardo Querzoni. An toàn: Tích hợp chức năng tự chú ý để đảm bảo tính tương tự nhị phân. TRONG *Hội nghị quốc tế về phát hiện sự xâm nhập và phần mềm độc hại cũng như đánh giá lỗ hổng*, trang 309– 329. Springer, 2019.
- [36] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, và những người khác. Huấn luyện độ chính xác hỗn hợp. *bản in trước arXiv arXiv:1710.03740*, 2017.
- [37] Tomas Mikolov, Kai Chen, Greg Corrado và Jeffrey Dean. Ước tính hiệu quả các biểu diễn từ trong không gian vectơ. *bản in trước arXiv arXiv:1301.3781*, 2013.
- [38] Nguyễn Anh Quỳnh. Capstone: Khung tháo gỡ thể hệ tiếp theo.
- [39] OpenAI. Báo cáo kỹ thuật Gpt-4, 2023.
- [40] Yuhei Otsubo, Akira Otsuka, Mamoru Mimura, Takeshi Sakaki và Hiroshi Ukegawa. o-glassx: Khôi phục nguồn gốc trình biên dịch với cơ chế chú ý từ một đoạn mã ngắn. TRONG *Kỷ yếu Hội thảo Nghiên cứu Phân tích nhị phân lần thứ 3*, 2020.

- [41] Kexin Pei, Jonas Guan, David Williams-King, Junfeng Yang và Suman Jana. Xda: Tháo gỡ chính xác, mạnh mẽ với transfer learning. *bản in trước arXiv arXiv:2010.00770*, 2020.
- [42] Jeffrey Pennington, Richard Socher, và Christopher D Manning. Găng tay: Các vectơ toàn cầu để biểu diễn từ. TRONG *Kỷ yếu hội nghị các phương pháp thực nghiệm trong xử lý ngôn ngữ tự nhiên (EMNLP) năm 2014*, trang 1532–1543, 2014.
- [43] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee và Luke Zettlemoyer. Biểu diễn từ theo ngữ cảnh sâu sắc. TRONG *Kỷ yếu Hội nghị năm 2018 của Chi hội Bắc Mỹ của Hiệp hội Ngôn ngữ học tính toán: Công nghệ ngôn ngữ con người, Tập 1 (Bài báo dài)*, trang 2227–2237, New Orleans, Louisiana, tháng 6 năm 2018. Hiệp hội Ngôn ngữ học Tính toán.
- [44] Michael Pradel và Koushik Sen. Deepbugs: một phương pháp học tập để phát hiện lỗi dựa trên tên. *Kỷ yếu của ACM về Ngôn ngữ lập trình*, 2:1 – 25, 2018.
- [45] Alec Radford, Karthik Narasimhan, Tim Salimans và Ilya Sutskever. Cải thiện sự hiểu biết ngôn ngữ bằng cách đào tạo trước.
- [46] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Các mô hình ngôn ngữ là những người học đa nhiệm không được giám sát. *Blog OpenAI*, 1(8):9, 2019.
- [47] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, và Yuxiong He. Deepspeed: Tối ưu hóa hệ thống cho phép đào tạo các mô hình deep learning với hơn 100 tỷ tham số. TRONG *Kỷ yếu Hội nghị quốc tế ACM SIGKDD lần thứ 26 về khám phá tri thức và khai thác dữ liệu*, trang 3505–3506, 2020.
- [48] Eui Chul Richard Shin, Dawn Song, và Reza Moazzzi. Nhận dạng các hàm nhị phân với mạng lưới thần kinh. TRONG *Hội nghị chuyên đề về bảo mật (USENIX) lần thứ 24 (USENIX) Bảo mật 15*, trang 611–626, 2015.
- [49] Alexey Svyatkovskiy, Ying Zhao, Shengyu Fu và Neel Sundaresan. Pythia: Hệ thống hoàn thiện mã được hỗ trợ bởi Ai. TRONG *Kỷ yếu hội nghị quốc tế ACM SIGKDD lần thứ 25 về khám phá tri thức và khai thác dữ liệu*, trang 2727–2735, 2019.
- [50] Hao Wang, Wenjie Qu, Gilad Katz, Wenyu Zhu, Zeyu Gao, Han Qiu, Jianwei Zhuge và Chao Zhang. jtrans: biến áp nhận biết bước nhảy để phát hiện sự tương tự mã nhị phân. TRONG *Kỷ yếu Hội nghị chuyên đề quốc tế ACM SIGSOFT lần thứ 31 về kiểm thử và phân tích phần mềm*, trang 1–13, 2022.
- [51] Ke Xu, Yingjiu Li, Robert H Đặng, và Kai Chen. Deeprefiner: Hệ thống phát hiện phần mềm độc hại Android nhiều lớp áp dụng mạng lưới thần kinh sâu. TRONG *Hội nghị chuyên đề Châu Âu của IEEE 2018 về Bảo mật và Quyền riêng tư (EuroS&P)*, trang 473–487. IEEE, 2018.
- [52] Xiaojun Xu, Chang Liu, Qian Feng, Heng Yin, Le Song và Dawn Song. Nhúng biểu đồ dựa trên mạng thần kinh để phát hiện sự giống nhau của mã nhị phân đa nền tảng. TRONG *Kỷ yếu Hội nghị ACM SIGSAC 2017 về An ninh Máy tính và Truyền thông*, trang 363– 376, 2017.
- [53] Zeping Yu, Rui Cao, Qiyi Tang, Sen Nie, Junzhou Huang và Shi Wu. Vấn đề về thứ tự: Mạng thần kinh nhận biết ngữ nghĩa để phát hiện sự giống nhau của mã nhị phân. TRONG *Kỷ yếu Hội nghị AAAI về Trí tuệ nhân tạo*, tập 34, trang 1145–1152, 2020.
- [54] Yifan Zhang, Chen Huang, Yueke Zhang, Kevin Cao, Scott Thomas Andersen, Huajie Shao, Kevin Leach, và Vũ Hoàng. Combo: Huấn luyện trước các biểu diễn mã nhị phân bằng cách học tương phản. *bản in trước arXiv arXiv:2210.05102*, 2022.
- [55] Wenyu Zhu, Zhiyao Feng, Zihan Zhang, Jianjun Chen, Zhijian Ou, Min Yang và Chao Zhang. Callee: Khôi phục biểu đồ cuộc gọi cho các tệp nhị phân với phương pháp học chuyển giao và đối chiếu. TRONG *Hội nghị chuyên đề IEEE 2023 về Bảo mật và Quyền riêng tư (SP)*, trang 1953–1970. Hiệp hội máy tính IEEE, 2023.