ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO TỔNG KẾT
ĐỀ TÀI KHOA HỌC VÀ CÔNG NGHỆ SINH VIÊN NĂM 2024

Tên đề tài tiếng Việt: **PHÁT HIỆN LỖ HỔNG TRONG HỢP ĐỒNG THÔNG MINH TRÊN MẠNG LIÊN CHUỖI KHỐI BẰNG PHƯƠNG PHÁP HỌC MÁY VÀ HỌC SÂU**

.....

Tên đề tài tiếng Anh: **SMART CONTRACT VULNERABILITIES AUTOMATIC DETECTION ON THE CROSS-CHAIN NETWORK USING MACHINE LEARNING AND DEEP LEARNING**

.....

Khoa/ Bộ môn: Mạng máy tính và Truyền thông

Thời gian thực hiện: 6 tháng

Cán bộ hướng dẫn: Th.S Trần Tuấn Dũng

Tham gia thực hiện

TT	Họ và tên, MSSV	Chịu trách nhiệm	Điện thoại	Email
1.	Võ Anh Kiệt, 20520605	Chủ nhiệm	0365642317	20520605@gm.uit.edu.vn

ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Ngày nhận hồ sơ	
Mã số đề tài	
(Do CQ quản lý ghi)	

BÁO CÁO TỔNG KẾT

Tên đề tài tiếng Việt: **PHÁT HIỆN LỖ HỔNG TRONG HỢP ĐỒNG THÔNG MINH TRÊN MẠNG LIÊN CHUỖI KHỐI BẰNG PHƯƠNG PHÁP HỌC MÁY VÀ HỌC SÂU**

.....

Tên đề tài tiếng Anh: **SMART CONTRACT VULNERABILITIES AUTOMATIC DETECTION ON THE CROSS-CHAIN NETWORK USING MACHINE LEARNING AND DEEP LEARNING**

.....

Ngày ... tháng ..... năm ....
Cán bộ hướng dẫn
(Họ tên và chữ ký)

Ngày ... tháng ..... năm ....
Sinh viên chủ nhiệm đề tài
(Họ tên và chữ ký)

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài: PHÁT HIỆN LỖ HỔNG TRONG HỢP ĐỒNG THÔNG MINH TRÊN MẠNG LIÊN CHUỖI KHỐI BẰNG PHƯƠNG PHÁP HỌC MÁY VÀ HỌC SÂU
- Mã số:
- Chủ nhiệm: VÕ ANH KIỆT – 20520605
- Thành viên tham gia:
- Cơ quan chủ trì: Trường Đại học Công nghệ Thông tin.
- Thời gian thực hiện: 6 tháng

2. Mục tiêu:

Triển khai được mô hình cầu nối dựa trên phương pháp SideChain và tiến hành được quá trình chuyển đổi dữ liệu qua các mạng chuỗi khối.

Xây dựng tập dữ liệu CrossChainSentinel bao gồm các mẫu lành tính và các mẫu độc hại được gán nhãn thủ công, đồng thời tiến hành xử lý dữ liệu.

Ứng dụng các mô hình học máy và học sâu trong việc phát hiện các mẫu độc hại một cách tự động, đánh giá và nhận xét các mô hình.

3. Tính mới và sáng tạo:

Ở nghiên cứu này, nhóm nghiên cứu sinh viên đã tiến hành triển khai việc chuyển đổi dữ liệu thông qua cầu nối Sidechain, đồng thời xây dựng và cung cấp được tập dữ liệu dành riêng cho các hợp đồng thông mô hình cầu nối Sidechain bao gồm 300 mẫu (bao gồm 158 mẫu an toàn và 142 mẫu có lỗ hổng). Đồng thời, ở nghiên cứu này nhóm cũng tiến hành việc xử lý dữ liệu và tiến hành thực hiện quá trình huấn luyện các mô hình học máy và học sâu để có thể tự động hóa việc phát hiện lỗ hỏng trên hợp đồng thông minh một cách tự động.

4. Tóm tắt kết quả nghiên cứu:

Hoàn tất quá trình triển khai được mô hình cầu nối dựa trên phương pháp SideChain và tiến hành được quá trình chuyển đổi dữ liệu qua các mạng chuỗi khối

Thành công trong việc tập dữ liệu CrossChainSentinel bao gồm các mẫu lành tính và các mẫu độc hại được gán nhãn thủ công, đồng thời tiến hành xử lý dữ liệu.

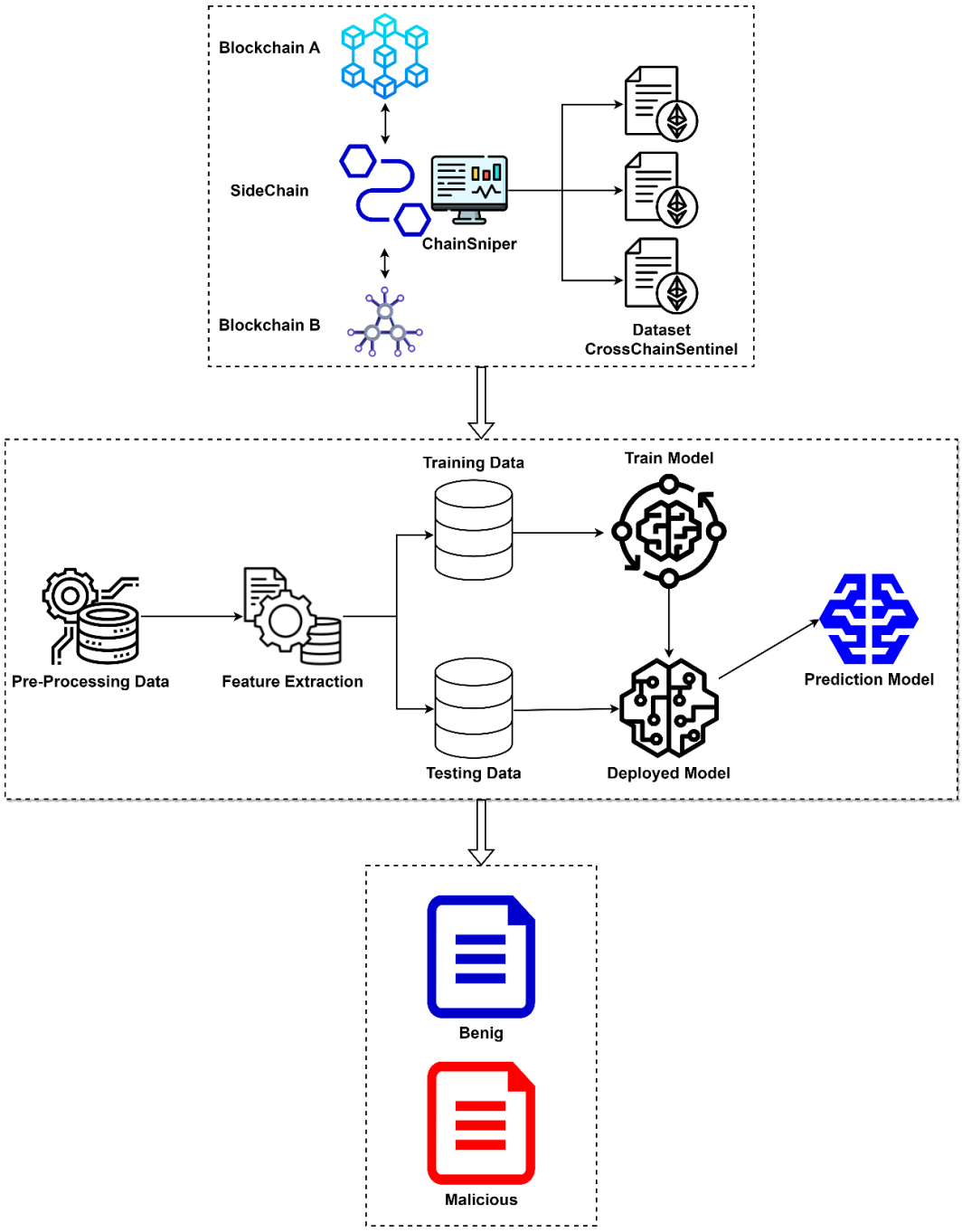
Đánh giá và nhận xét các mô hình học máy và học sâu trong việc phát hiện các mẫu độc hại một cách tự động, các mô hình.

5. Tên sản phẩm: Phát hiện lỗ hổng trong hợp đồng thông minh trên mạng liên chuỗi khối bằng phương pháp học máy và học sâu

6. Hiệu quả, phương thức chuyển giao kết quả nghiên cứu và khả năng áp dụng:

Thực nghiệm cho thấy hệ thống có khả năng phát hiện các lỗ hổng trong hợp đồng thông minh liên chuỗi với độ chính xác cao, lên tới 96,7% khi sử dụng mô hình Roberta. Các mô hình học máy và học sâu trong ChainSniper như Random Forest, XGBoost, CNN, LSTM cũng đạt hiệu suất tốt, với độ chính xác trên 70%, cao nhất với mô hình RoBERTa với 96.67%. Thời gian xử lý mẫu hợp đồng thông minh cũng khá nhanh, từ 3.3 – 4.0 giây tùy thuộc vào mô hình, phù hợp với yêu cầu đánh giá an ninh tự động. Hệ thống có thể phát hiện các lỗ hổng phổ biến như tấn công reentrancy, tràn/thiếu số nguyên và rút tiền Ether không được bảo vệ. Kết quả nghiên cứu có thể được chuyển giao dưới dạng một công cụ phần mềm mã nguồn mở và có thể được áp dụng rộng rãi trong việc kiểm tra an ninh hợp đồng thông minh liên chuỗi trước khi triển khai vào các nền tảng blockchain thực tế.

7. Hình ảnh, sơ đồ minh họa chính



Cơ quan Chủ trì
(ký, họ và tên, đóng dấu)

Chủ nhiệm đề tài
(ký, họ và tên)

MỤC LỤC

CHƯƠNG 1: MỞ ĐẦU.....10
1.1. Vấn đề về Blockchain và Crosschain..... 10
1.2. Vấn đề về học máy và học sâu..... 14
CHƯƠNG 2: CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN.....16
2.1. Các phương pháp ứng dụng học máy và học sâu 16
2.2. Các phương pháp ứng dụng công cụ..... 20
CHƯƠNG 3: ĐỀ XUẤT GIẢI PHÁP.....28
3.1. Tổng quan mô hình 28
3.1.1. Các thành phần chính của mô hình 28
3.1.2. Mô hình cầu nối..... 30
3.2. Phát hiện các lỗ hổng bằng phương pháp học máy và học sâu.. 32
3.2.1. Xây dựng tập dữ liệu..... 32
3.2.2. Gán nhãn các mẫu và xử lý dữ liệu..... 34
3.2.3. Ứng dụng các mô hình học máy trong việc phát hiện các lỗ hổng..... 36
3.2.4. Ứng dụng các mô hình học sâu trong việc phát hiện các lỗ hổng..... 37
4.1. Thiết lập thực nghiệm 38
4.1.1. Môi trường thực nghiệm 38
4.1.2. Chỉ số đánh giá..... 39
4.1.3. Kịch bản thực nghiệm..... 42
4.2. Kết quả thực nghiệm 44
4.2.1. Kết quả về mặt hiệu suất..... 44
4.2.2. Kết quả về mặt thời gian..... 45
4.3. Thảo luận..... 46
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....48
5.1. Kết luận 48
5.2. Hướng phát triển 49
TÀI LIỆU THAM KHẢO 50

DANH MỤC HÌNH ẢNH

Hình 1 Giao tiếp liên chuỗi giữa hai chuỗi khối thông qua sidechain 11
Hình 2 Công trình nghiên cứu của Xu và nhóm nghiên cứu 16
Hình 3 Công trình nghiên cứu của Deng và nhóm nghiên cứu..... 19
Hình 4 Công trình nghiên cứu của Huang và nhóm nghiên cứu 22
Hình 5 Công trình nghiên cứu của Jiang và nhóm nghiên cứu..... 24
Hình 6 Công trình nghiên cứu của Parizi và nhóm nghiên cứu..... 26
Hình 7 Mô hình tổng quan 29
Hình 8 Các bước chuyển đổi dữ liệu qua cầu nối Sidechain 32
Hình 9 Phân bố mẫu lành tính và độc hại 33
Hình 10 Phân bố mẫu lành tính và độc hại tương ứng với các lỗ hổng..... 33
Hình 11 Quá trình xử lý dữ liệu..... 35
Hình 12 Thời gian thực thi của các mô hình học máy và học sâu trong việc phát hiện lỗ hổng bảo mật trong các hợp đồng thông minh liên chuỗi..... 46

DANH MỤC BẢNG BIỂU

Bảng 1 Môi trường thực nghiệm mạng liên chuỗi.....38
Bảng 2 Môi trường thực nghiệm các phương pháp học máy và học sâu .39
Bảng 3 Hiệu suất của các mô hình học máy và học sâu trong việc phát hiện lỗ hổng bảo mật trong các hợp đồng thông minh liên chuỗi 44
Bảng 4 Thời gian thực thi của các mô hình học máy và học sâu trong việc phát hiện lỗ hổng bảo mật trong các hợp đồng thông minh liên chuỗi 45

TÓM TẮT

Trong những năm gần đây, có nhiều tiến triển trong quá trình phát triển và triển khai hợp đồng thông minh trên mạng chuỗi chéo, việc này giúp tạo điều kiện cho giao tiếp và trao đổi dữ liệu giữa các blockchain khác nhau được diễn ra một cách hiệu quả. Tuy nhiên, việc triển khai các ứng dụng này mang theo rủi ro về an toàn thông tin, đặc biệt là trong việc phát hiện lỗ hổng trong các hợp đồng thông minh, có thể gây nguy hiểm đến tính bảo mật. Trong các nghiên cứu trước đây đã tập trung vào việc xác định và phát hiện lỗ hổng trong hợp đồng thông minh bằng các phương pháp kiểm tra ký tự và thực thi, tuy nhiên, các phương pháp hiện nay vẫn chưa đạt được khả năng phân tích toàn diện. Do đó, trong nghiên cứu này, nhóm đề xuất sử dụng các phương pháp học máy và học sâu để phân tích các lỗ hổng này một cách hiệu quả hơn.

Ở công trình nghiên cứu này, em giới thiệu các phương pháp học máy và học sâu dựa trên ChainSniper - một khung phân tích tích hợp học máy dựa trên sidechain để tự động đánh giá lỗ hổng hợp đồng thông minh chéo chuỗi. Phương pháp mà nhóm đề xuất là xây dựng một tập dữ liệu quy mô lớn gồm 300 đoạn mã có gán nhãn thủ công, được gọi là CrossChainSentinel, đã được biên soạn để huấn luyện các mô hình phân biệt mã dễ bị tấn công và mã an toàn. Các đoạn mã này bao gồm các lỗ hỏng: Reentrancy, Interger Overflow/Underflow và Unprotected Ether Withdrawl. Kết quả thực nghiệm đã chứng minh được tính hiệu quả của việc ứng dụng học máy và học sâu giúp tăng sự hiệu quả của việc kiểm tra hợp đồng thông minh cho các ứng dụng phi tập trung phân tán trên nhiều blockchain. Độ chính xác phát hiện đạt mức đáng kể, khẳng định tiềm năng của ChainSniper trong việc tăng cường an ninh thông qua đánh giá tự động và toàn diện mã hợp đồng.

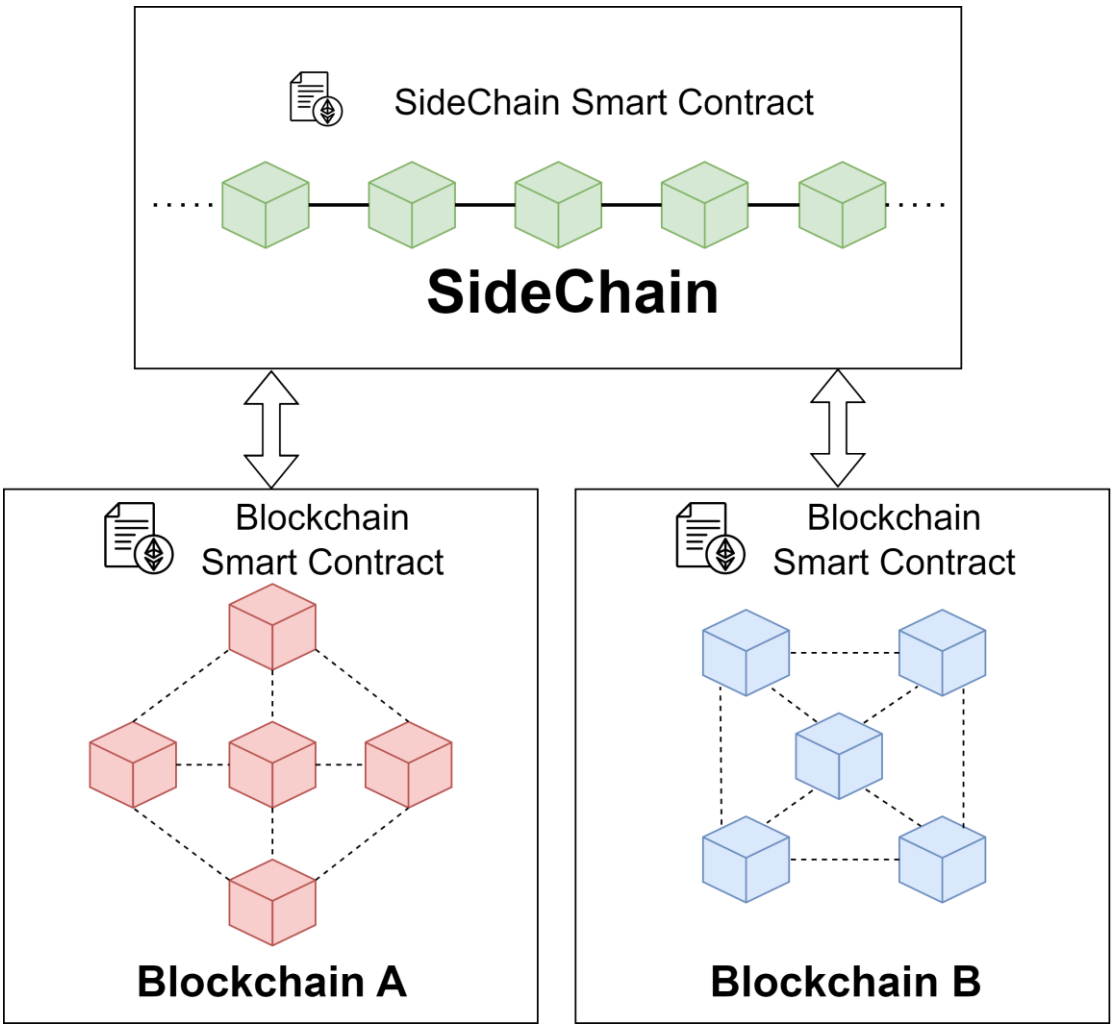
CHƯƠNG 1: MỞ ĐẦU

Trong phần này, em sẽ trình bày tổng quan về bối cảnh của vấn đề, các thách thức mà công trình này tiến hành giải quyết.

1.1. Vấn đề về Blockchain và Crosschain

Sự phát triển của công nghệ blockchain đã tạo ra bước tiến lớn trong việc hình thành các mạng không tập trung, mang lại khả năng lưu trữ và trao đổi thông tin một cách an toàn và chính xác. Điều này được thực hiện thông qua cơ chế đồng thuận phân tán giữa các nút mạng [1, 2]. Tuy nhiên, một hạn chế của hệ thống blockchain là việc chúng thường được xây dựng một cách độc lập với quy tắc và giao thức riêng, dẫn đến khó khăn trong việc tương tác và trao đổi dữ liệu giữa các blockchain khác nhau.

Đáp ứng vấn đề này, công nghệ liên chuỗi (cross-chain) đã được phát triển như một giải pháp cho phép các mạng blockchain tương tác với nhau một cách an toàn và hiệu quả [3]. Công nghệ này hỗ trợ việc chuyển giao tài sản số giữa các chuỗi khác nhau, từ đó thúc đẩy khả năng tích hợp và phát triển của hệ sinh thái blockchain [4]. Các giải pháp như Polkadot, chẳng hạn, đã cung cấp một khung làm việc cho phép các blockchain độc lập, với cấu trúc và chức năng khác nhau, giao tiếp và tương tác với nhau, mở ra một bước tiến mới trong lĩnh vực công nghệ blockchain và các ứng dụng phi tập trung [5].



Hình 1 Giao tiếp liên chuỗi giữa hai chuỗi khối thông qua sidechain

Công nghệ liên chuỗi nhằm kết nối các hệ sinh thái blockchain bị cô lập, cho phép tài sản và dữ liệu được chuyển giao và chia sẻ một cách thuận tiện giữa các blockchain khác nhau [6]. Đóng góp này tiến hành giải quyết một vấn đề rất quan trọng trong việc xử lý các điểm hạn chế về khả năng xử lý cũng như khả năng về mở rộng và đồng thời là chức năng mà các blockchain riêng lẻ thường gặp phải. Hiện tại tồn tại ba cách tiếp cận chính để kết nối và cho phép chuyển tài sản giữa các chuỗi khối khác nhau: các giải pháp công chứng, khóa băm, và các relays/sidechains [7–9]. Cơ chế công chứng là phương pháp đồng thuận trong đó các bên thứ ba tin cậy (công chứng viên) xác minh giao dịch bằng chữ ký số trước khi chúng được thêm vào blockchain, nhằm ngăn giao dịch kép. Giải pháp

khóa băm có thể triển khai thông qua cổng để truy cập các hợp đồng khóa thời gian khóa băm (Hash Time Lock Contract - HTLC) trên blockchain từ xa, đảm bảo việc nhận thanh toán trước khi phục dựng tài sản trên blockchain đích.

Sidechains là các blockchain phụ được gắn nối với blockchain chính thông qua thanh gài 2 chiều, cho phép chuyển tài sản giữa các chuỗi, thêm tính năng mới cho blockchain chính mà không cần sửa đổi giao thức, giải quyết những vấn đề thách thức như việc mở rộng hệ thống và đảm bảo vấn đề về sự riêng tư của các thông tin. Tuy nhiên, giao thức tương tác chuỗi khối này vẫn tồn tại các vấn đề về tính bí mật, tính riêng tư và sự sẵn sàng của hệ thống tư cần giải quyết.

Đặc biệt, các cuộc tấn công liên chuỗi nhằm vào việc lợi dụng các điểm yếu trong hợp đồng thông minh tiến hành thực thi trên nhiều mạng blockchain riêng biệt, có thể gây ra tổn thất tài chính nghiệm trọng. Một ví dụ điển hình là vụ việc tấn công gần đây, nơi mà lỗ hổng trong một hợp đồng thông minh liên chuỗi đã bị khai thác, dẫn đến việc mất mát tài sản trị giá hơn 600 triệu đô la [11]. Một trường hợp khác, vào năm 2016, dự án gây quỹ The DAO trên Ethereum cũng đã trở thành nạn nhân của một cuộc tấn công, mất đi hơn 50 triệu đô la do lỗ hổng trong hợp đồng thông minh của họ. Các sự cố tấn công hợp đồng thông minh tiếp tục xảy ra, ví dụ như vụ tấn công vào ví Parity tháng 7/2017 khiến mất mát 30 triệu đô la tiền điện tử Ether. Hay vụ đánh cắp gần 300.000 đô la từ nền tảng KingDice tháng 8/2017 cũng do lợi dụng lỗ hổng trong mã hợp đồng. Hơn thế nữa, trong gian đoạn gần đây, thị trường blockchain đã tiếp nhận 1 cuộc tấn công là loạt vụ tấn công vào các hợp đồng thông minh trên Binance Smart Chain năm 2021, trong đó có vụ đánh cắp hơn 200 triệu đô la thông qua hợp đồng của Venus Protocol [12, 13]. Như vậy, việc

phân tích và bảo mật hợp đồng thông minh trước khi tiến hành thực thi và sử dụng mang tính cấp thiết lớn trong việc hạn chế rủi ro mất mát tài sản. Hợp đồng thông minh, là các giao thức số được thiết kế để làm đơn giản hóa, kiểm chứng hoặc thực thi các quy trình đàm phán và thực hiện hợp đồng. Chúng được ứng dụng rộng rãi, từ dịch vụ tài chính đến thị trường dự đoán và trong lĩnh vực Internet vạn vật [14]. Những hợp đồng này hoạt động hiệu quả trên các nền tảng blockchain, tự động hoá các hành động theo các điều kiện đã đặt ra trước, giảm bớt nhu cầu cho các bên trung gian. Hợp đồng thông minh, do đó, tạo điều kiện cho việc giao dịch không dựa vào sự tin tưởng và tự động hóa thực hiện các quy trình trong hệ thống blockchain. Với sự phát triển nhanh chóng của công nghệ blockchain và việc ứng dụng nó trong nhiều ngành nghề khác nhau, việc phân tích và kiểm tra bảo mật cho hợp đồng thông minh trở nên cực kỳ quan trọng trước khi chúng được triển khai [15]. Do chúng hoạt động dựa trên mã tự thực thi, bất kỳ lỗ hổng nào cũng có thể gây ra hậu quả lớn. Mặc dù việc kiểm tra và phân tích mã bằng phương pháp thủ công là thiết yếu, nhưng quá trình này lại mất nhiều thời gian và công sức, và cũng dễ phát sinh lỗi do con người. Trong các hệ thống blockchain có liên kết chéo như Polkadot, việc bảo mật và kiểm thử càng trở nên phức tạp và thách thức [16].

Hiện nay, hệ sinh thái Ethereum đang phải đối mặt với nhiều lỗ hổng bảo mật nghiêm trọng như các cuộc tấn công lặp lại (Reentrancy) [16], vấn đề tràn số (Overflow/Underflow) [17] và các vấn đề liên quan đến việc rút token không an toàn trong các hợp đồng thông minh [18]. Những vấn đề này đều là những rủi ro lớn đối với các ứng dụng phi tập trung (dApps) trên Ethereum. Do đó, cần có các giải pháp và công cụ kiểm thử bảo mật hiệu quả hơn để đáp ứng với tốc độ phát triển của các ứng dụng blockchain hiện đại. Mạng chuỗi liên kết chéo (cross-chain) mở ra cơ hội

phát triển các ứng dụng phi tập trung phức tạp hơn bằng cách kết nối nhiều blockchain với nhau. Tuy nhiên, điều này cũng tạo ra nhiều lỗ hổng bảo mật hơn, mà kẻ tấn công có thể khai thác để gây ra các vụ tấn công [19]. Một trong những cuộc tấn công đáng được chú ý đó là cuộc tấn công lặp lại (reentrancy), cho phép đối tượng tấn công lặp đi lặp lại lời gọi hàm của hợp đồng thông minh trước khi hoàn thành yêu cầu trước đó, dẫn đến các hậu quả khó lường [20]. Bên cạnh đó, các lỗ hổng tràn số và underflow cũng rất nguy hiểm, xảy ra khi giá trị vượt ngưỡng trên hoặc ngưỡng dưới cho phép, sẽ gây ra hậu quả khôn lường [21]. Đặc biệt, lỗ hổng rút tiền điện tử mà không được xác thực (unprotected ether withdrawal) cũng đang nhận được sự quan tâm, khi một hợp đồng thông minh không xác minh chính xác yêu cầu rút tiền, cho phép hacker rút Ether một cách bất hợp pháp [22]. Những hậu quả từ các lỗ hổng bảo mật nói trên đã và đang lan rộng trong hệ sinh thái Ethereum, gây thiệt hại tài chính lớn cho nhiều bên liên quan [12]. Tình hình nghiêm trọng hiện nay nhấn mạnh sự cần thiết của việc phát triển và tuân thủ nghiêm ngặt các chính sách và quy trình bảo mật trong quá trình tạo ra các ứng dụng blockchain. Điều này đòi hỏi các doanh nghiệp và tổ chức phải tập trung đầu tư vào các hoạt động kiểm tra, phát hiện lỗ hổng và cải tiến mã nguồn của sản phẩm trước khi chúng được triển khai. Bằng cách này, có thể giảm thiểu rủi ro và thiệt hại do các lỗi bảo mật trong ứng dụng blockchain gây ra.

1.2. Vấn đề về học máy và học sâu

Học máy, một lĩnh vực của trí tuệ nhân tạo, cho phép máy tính học hỏi và cải thiện từ kinh nghiệm mà không cần sự lập trình chi tiết. Sự tiến bộ trong học máy, đặc biệt là phát triển của học sâu, đã mở ra khả năng xử lý dữ liệu lớn, dự đoán và ra quyết định trong nhiều lĩnh vực vượt trội so với

con người [23, 24]. Trong lĩnh vực hợp đồng thông minh, các mô hình học máy có thể được sử dụng để phân tích mã và nhận diện các vấn đề như lỗi lặp, tấn công tràn số, và tấn công từ chối dịch vụ (DoS) [25]. Các mô hình này, được huấn luyện từ cả các ví dụ về hợp đồng thông minh dễ tổn thương và an toàn, có thể thực hiện xác minh hợp đồng một cách hiệu quả và tự động ở quy mô lớn.

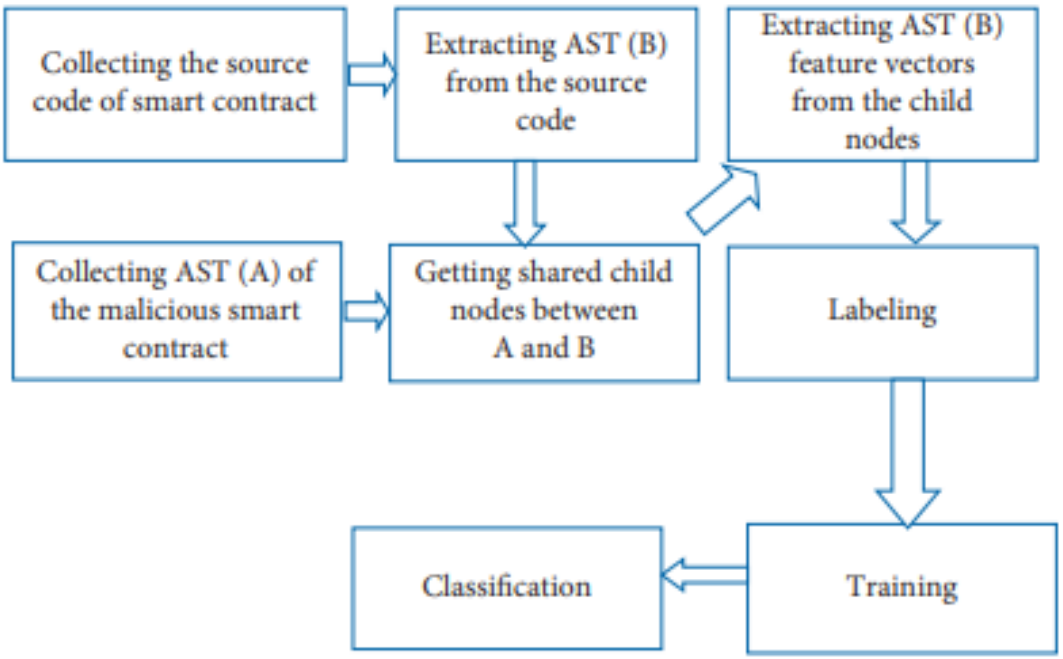
Các nghiên cứu đã tập trung phát triển mô hình học máy có khả năng phát hiện tự động các đoạn mã chứa lỗi hoặc lỗ hổng bảo mật. Tôi đã phát triển một bộ dữ liệu mã nguồn Solidity, tập trung vào việc phát hiện các lỗ hổng tiềm ẩn. Sử dụng bộ dữ liệu này, tôi đã huấn luyện nhiều mô hình phân loại khác nhau như cây quyết định, rừng ngẫu nhiên, SVM, LSTM, CNN, để cải thiện khả năng phát hiện các đoạn mã dễ bị tấn công. Học sâu, một nhánh của học máy, sử dụng mạng nơ-ron nhân tạo sâu, có khả năng tự động trích xuất đặc trưng và mẫu từ dữ liệu phức tạp như hình ảnh, âm thanh, và ngôn ngữ tự nhiên. Việc áp dụng học sâu trong phân tích mã nguồn cũng mang lại kết quả tích cực, ví dụ như mô hình Roberta đã được chứng minh là hiệu quả trong việc phân loại và dự đoán lỗ hổng. Đây là một hướng nghiên cứu tiềm năng để tự động hóa quá trình kiểm tra và bảo mật mã nguồn.

CHƯƠNG 2: CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

Trong phần này, em sẽ trình bày các công trình nghiên cứu liên quan trong việc ứng dụng học máy, học sâu và các ứng dụng trong việc phát hiện lỗ hổng.

2.1. Các phương pháp ứng dụng học máy và học sâu

Hiện nay, đã có một số nghiên cứu quan trọng liên quan đến vấn đề mà em đang tìm hiểu. Nhóm của Xu [26] đã tổng hợp việc ứng dụng các phương pháp học máy để tìm ra lỗi trong các hợp đồng thông minh. Họ đề xuất một cách phân tích mới dựa trên học máy, kết hợp với cây cú pháp trừu tượng (AST) để tự động rút ra các đặc điểm quan trọng.



Hình 2 Công trình nghiên cứu của Xu và nhóm nghiên cứu

Phương pháp của nhóm Xu gồm việc tạo AST cho các hợp đồng thông minh cần phân tích và các mẫu hợp đồng có lỗi cơ bản. Sau đó, họ dùng các nút con chung để phân tích và so sánh độ giống nhau về cấu trúc giữa

các AST. Từ đó, họ tạo ra các vector đặc trưng dựa trên những nút con chung này.

Để nâng cao khả năng phát hiện lỗi, nhóm nghiên cứu kết hợp mô hình k láng giềng gần nhất (KNN) vào quá trình phân tích. Mô hình cuối có thể dự đoán 8 loại lỗi phổ biến trong hợp đồng thông minh, bao gồm: Re-entrancy, Arithmetic, Access Control, Denial of Service, Unchecked Low Level Calls, Bad Randomness, Front Running, và Short Address.

Qua các thử nghiệm, công trình của Xu tỏ ra vượt trội hơn công cụ Oyente và SmartCheck về độ chính xác. Kết quả cho thấy mô hình KNN đạt độ chính xác, độ thu hồi và độ F1 trên 90% cho tất cả các loại lỗi được kiểm tra. Hơn nữa, phương pháp này không cần chạy hợp đồng trên môi trường Ethereum thật, giúp tăng tốc độ phân tích đáng kể.

Tuy nhiên, mô hình chủ yếu tập trung vào ngôn ngữ Solidity nên cần điều chỉnh thêm để xác định chính xác các dòng code dễ bị lỗi. Trong tương lai, nhóm nghiên cứu dự định mở rộng để phân tích các hệ thống blockchain khác ngoài Ethereum, bổ sung thêm dữ liệu mẫu về các hợp đồng có lỗi cơ bản, và phát triển thành một công cụ hoàn chỉnh để người dùng có thể dễ dàng áp dụng trong thực tế.

Một nghiên cứu khác [27] đã tiến hành đánh giá toàn diện về các thách thức bảo mật và quyền riêng tư trong việc tương tác giữa các blockchain, dựa trên phương pháp lấy ý kiến đa chiều (MLR). Nhóm tác giả đã chỉ ra một số lỗ hổng quan trọng như: tấn công wormhole: Lợi dụng lỗ hổng bảo mật để đánh cắp tài sản, tấn công từ chối dịch vụ làm tê liệt hệ thống, tấn công dựa trên thời gian giao dịch, sử dụng mật mã không tương thích:

gây mất tài sản khi chuyển đổi, rò rỉ thông tin trong hợp đồng khóa thời gian băm.

Ngoài ra, nghiên cứu cũng đề cập đến các vấn đề như tấn công thông đồng, chi tiêu kép và các mối lo ngại về quyền riêng tư khi các blockchain tương tác với nhau. Bằng cách phân tích cả tài liệu học thuật lẫn tài liệu xám, nhóm nghiên cứu đã cung cấp cái nhìn tổng quan về tình hình hiện tại, đồng thời nêu ra các giải pháp tiềm năng và thách thức còn tồn tại.

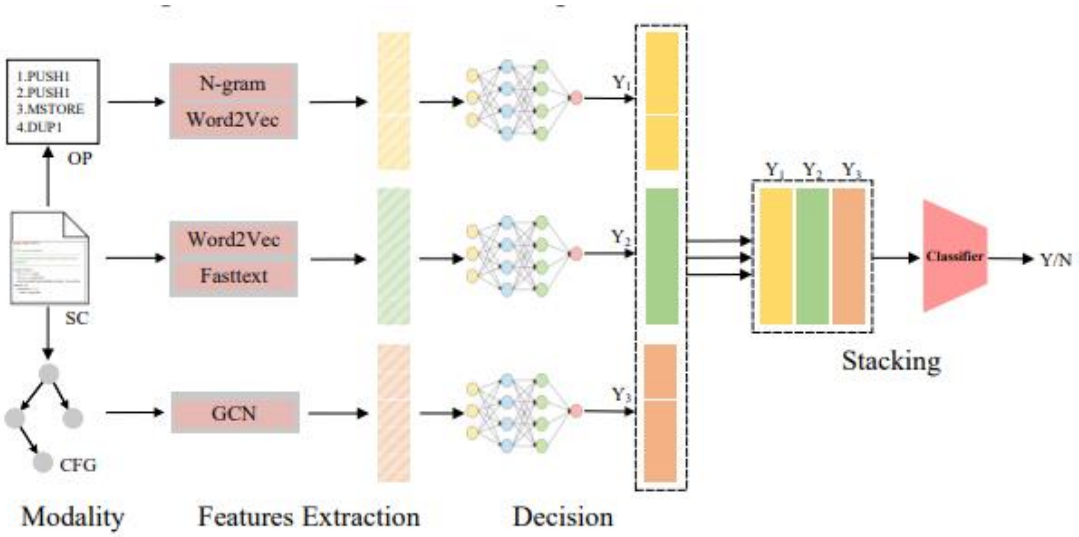
Để đối phó với những vấn đề này, bài báo đề xuất một số biện pháp như: sử dụng khóa đa bước ẩn danh (AMHL) chống tấn công wormhole, thêm lớp giao tiếp phụ ngăn tấn công thông đồng, triển khai cơ chế không khuyến khích với ba quan sát viên để chặn tấn công chi tiêu kép.

Tuy nhiên, các giải pháp này có thể tạo ra thách thức mới về khả năng mở rộng và độ phức tạp của hệ thống. Việc bổ sung các biện pháp bảo mật có thể làm chậm giao dịch và tăng độ phức tạp, từ đó có thể tạo ra lỗ hổng bảo mật mới.

Kết luận, bài báo kêu gọi cần thêm nghiên cứu để đánh giá hiệu quả của các biện pháp đề xuất và phát triển phương pháp toàn diện hơn nhằm giải quyết các thách thức về bảo mật và quyền riêng tư trong tương tác blockchain. Nhóm tác giả cũng gợi ý hướng nghiên cứu tương lai về mức độ độc lập giữa các lỗ hổng bảo mật và quyền riêng tư liên quan đến khả năng tương tác blockchain.

Deng và cộng sự [28] đã đề xuất một phương pháp mới để phát hiện lỗ hổng trong hợp đồng thông minh, kết hợp học sâu và hợp nhất quyết định

đa phương thức. Cách tiếp cận này xem xét toàn diện cả mã nguồn, mã hoạt động và cấu trúc điều khiển của hợp đồng, nhằm nâng cao độ chính xác trong việc tìm ra các lỗ hổng tiềm ẩn.



Hình 3 Công trình nghiên cứu của Deng và nhóm nghiên cứu

Nhóm nghiên cứu đã trích xuất năm đặc trưng khác nhau từ hợp đồng thông minh, bao gồm mã nguồn, mã hoạt động và các mô hình luồng điều khiển. Điều này giúp xây dựng một mô hình học sâu mạnh mẽ. Kết quả cho thấy độ chính xác cao trong việc phát hiện lỗ hổng, với các giá trị AUC đáng chú ý như 0,834 cho lỗi số học và 0,886 cho phụ thuộc thứ tự giao dịch.

Thực nghiệm chỉ ra độ chính xác và tỷ lệ hồi phục cao đối với nhiều loại lỗ hổng. Cụ thể, phương pháp đạt độ chính xác 91,6% cho lỗi số học, 90,9% cho lỗi truy hồi, 94,8% cho phụ thuộc thứ tự giao dịch và 89,5% cho khóa Ethernet. Những con số này minh chứng cho hiệu quả của phương pháp trong việc phát hiện đa dạng lỗ hổng.

Điểm nổi bật của nghiên cứu là việc sử dụng phương pháp hợp nhất quyết định đa phương thức. Các thí nghiệm cắt giảm cho thấy việc kết hợp quyết định từ nhiều bộ phân loại góp phần quan trọng vào hiệu suất tổng thể. Điều này nhấn mạnh tầm quan trọng của việc sử dụng các phương pháp hợp nhất quyết định để đạt kết quả tối ưu.

Tuy đạt được kết quả ấn tượng, nghiên cứu của Deng và cộng sự chưa khai thác tiềm năng của học không giám sát - một phương pháp có thể phát hiện lỗ hổng mà không cần nhãn dữ liệu. Điều này mở ra hướng nghiên cứu mới, tập trung vào việc tích hợp học không giám sát vào quá trình phát hiện lỗ hổng, nhằm nâng cao hơn nữa độ chính xác và hiệu quả của các phương pháp hiện có.

2.2. Các phương pháp ứng dụng công cụ

Daojing He và cộng sự đã công bố một nghiên cứu chi tiết [29] về các lỗ hổng bảo mật trong hợp đồng thông minh trên nền tảng Ethereum. Nghiên cứu tập trung vào việc khảo sát các lỗ hổng phổ biến và các biện pháp phòng ngừa, sử dụng cả phân tích tĩnh và động để phát hiện, phân loại lỗ hổng và đề xuất giải pháp cải thiện an ninh.

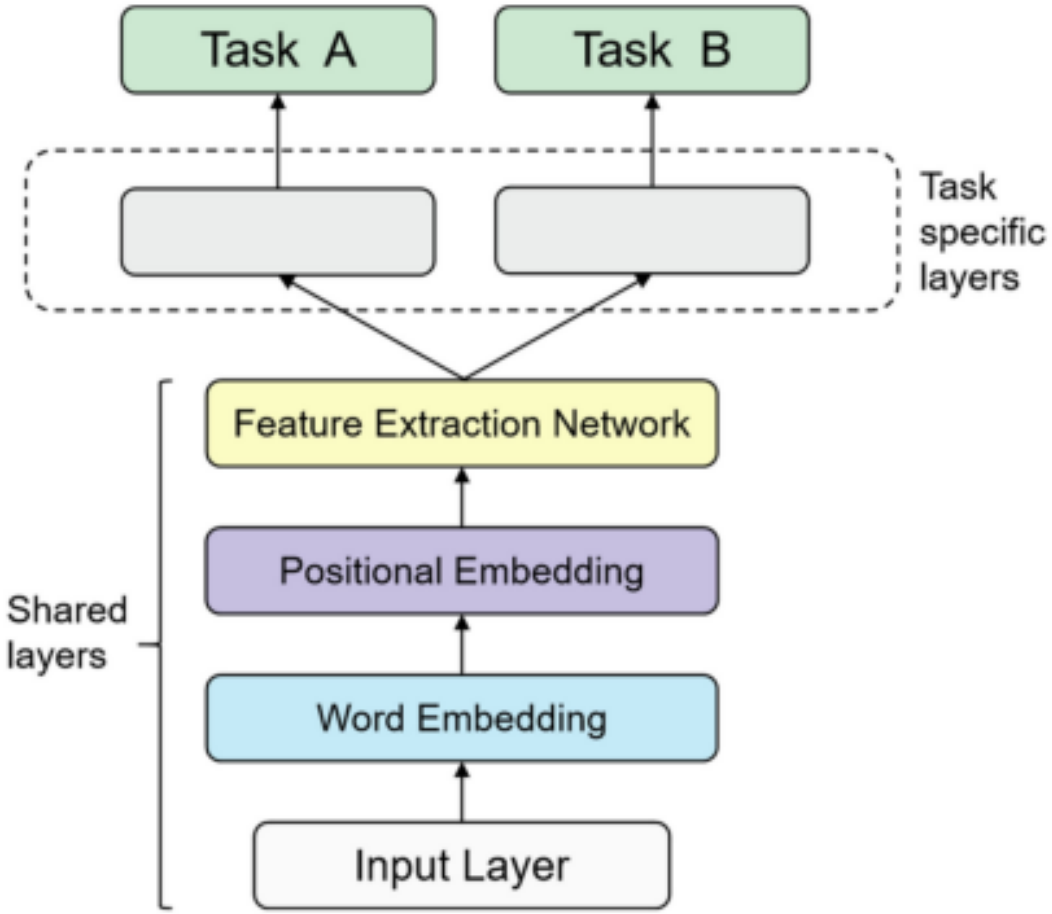
Nhóm nghiên cứu đã phân tích năm loại lỗ hổng chính, bao gồm lỗi tái nhập, lỗi tràn số, lỗi điều kiện chủng tộc, lỗi phụ thuộc thứ tự giao dịch và lỗi truy xuất không an toàn. Kết quả cho thấy lỗi tái nhập và lỗi tràn số là phổ biến và nguy hiểm nhất. Tuy nhiên, các công cụ phân tích hiện tại vẫn còn hạn chế trong việc phát hiện lỗ hổng, đặc biệt với các hợp đồng phức tạp.

Điểm nổi bật của nghiên cứu là đề xuất một khung phân tích bảo mật toàn diện, kết hợp phân tích tĩnh, động và kiểm tra thực nghiệm. Khung này không chỉ phát hiện lỗ hổng hiện có mà còn có khả năng tìm ra các lỗ hổng mới. Thực nghiệm cho thấy độ chính xác cao và tỷ lệ phát hiện tốt hơn so với phương pháp truyền thống, đặc biệt với lỗi tái nhập và lỗi tràn số.

Mặc dù đạt được nhiều kết quả ấn tượng, nghiên cứu vẫn còn một số hạn chế. Các công cụ phân tích chỉ phát hiện được lỗ hổng đã biết, không dự đoán được lỗ hổng mới. Kiểm tra an ninh chủ yếu dựa trên hợp đồng mẫu, gây khó khăn trong việc áp dụng rộng rãi.

Ngoài ra, nghiên cứu chưa khai thác tiềm năng của học máy và học sâu trong việc phát hiện lỗ hổng. Những hạn chế này mở ra hướng nghiên cứu mới cho tương lai, tập trung vào việc cải thiện khả năng phát hiện lỗ hổng bằng cách áp dụng các kỹ thuật học máy tiên tiến.

Huang và cộng sự [30] đã phát triển một mô hình phát hiện lỗ hổng hợp đồng thông minh dựa trên học đa nhiệm, nhằm nâng cao hiệu suất và độ chính xác. Mô hình này gồm hai phần chính: lớp chia sẻ học thông tin ngữ nghĩa của hợp đồng và lớp nhiệm vụ cụ thể sử dụng mạng nơ-ron tích chập (CNN) để phân loại từng nhiệm vụ. Bằng cách thiết lập các nhiệm vụ phụ trợ, mô hình có thể học các đặc trưng lỗ hổng và nhận diện chúng hiệu quả.



Hình 4 Công trình nghiên cứu của Huang và nhóm nghiên cứu

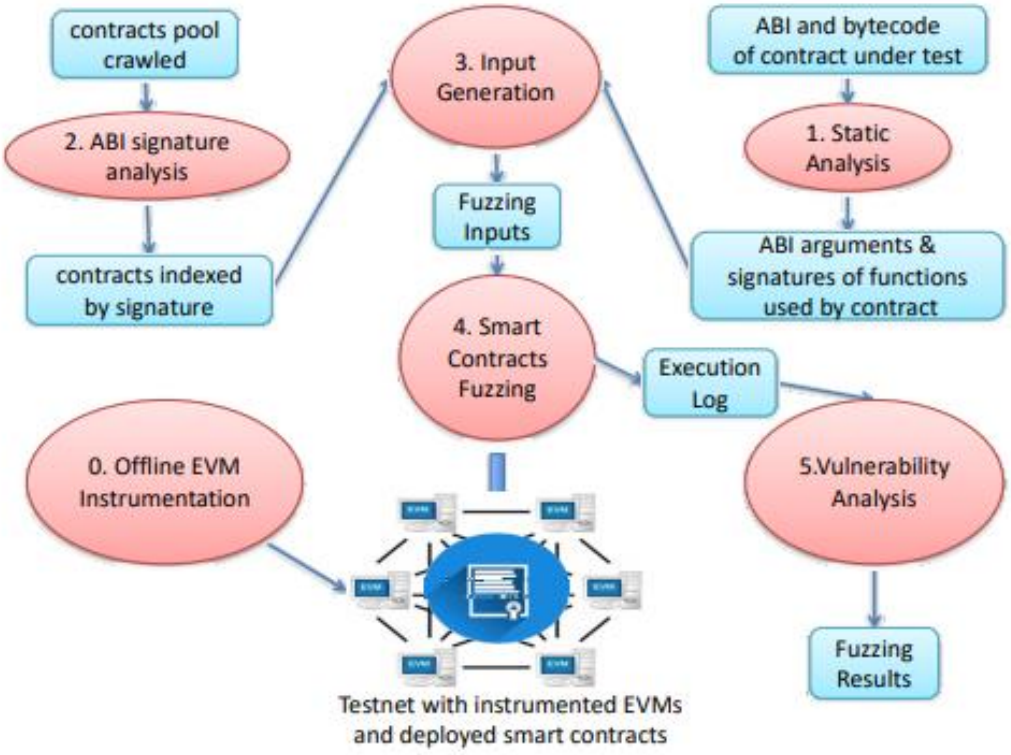
Nhóm nghiên cứu đã xây dựng một bộ dữ liệu mã nguồn mở từ 149,363 hợp đồng thông minh trên nền tảng XBlock, được dán nhãn bằng các công cụ phát hiện hiện có. Bộ dữ liệu này cung cấp thông tin quan trọng như địa chỉ, mã byte và mã nguồn, hỗ trợ việc nghiên cứu phát hiện lỗ hổng. Kết quả thực nghiệm cho thấy mô hình học đa nhiệm này không chỉ cải thiện độ chính xác mà còn tiết kiệm thời gian, tính toán và lưu trữ so với mô hình đơn nhiệm.

Phương pháp của Huang không chỉ nâng cao độ chính xác phát hiện lỗ hổng mà còn có khả năng mở rộng để hỗ trợ việc học và phát hiện các lỗ hổng mới. Nghiên cứu tập trung vào các lỗ hổng phổ biến như lỗ hổng số học, tái nhập và hợp đồng chứa địa chỉ không xác định. Kết quả cho thấy

việc sử dụng học đa nhiệm cải thiện đáng kể khả năng phát hiện và nhận diện lỗ hổng, mở ra triển vọng ứng dụng rộng rãi trong bảo mật và quản lý tài sản tài chính trên các nền tảng blockchain.

Tuy nhiên, phương pháp này vẫn còn một số hạn chế. Mô hình yêu cầu mã nguồn của hợp đồng thông minh để thực hiện việc phát hiện, giới hạn phạm vi áp dụng. Ngoài ra, nó mới chỉ dừng lại ở việc phát hiện một số loại lỗ hổng cụ thể và cần được mở rộng để phát hiện các loại lỗ hổng khác. Dù vậy, nghiên cứu của Huang và cộng sự đã mở ra hướng đi mới cho các nghiên cứu và ứng dụng tiếp theo trong lĩnh vực bảo mật hợp đồng thông minh.

ContractFuzzer [31] được phát triển bởi Jiang và cộng sự là một khung làm việc fuzzing được phát triển để phát hiện các lỗ hổng bảo mật trong hợp đồng thông minh trên nền tảng Ethereum. Công cụ này phân tích giao diện ABI của hợp đồng thông minh để tạo ra các đầu vào phù hợp với cú pháp của hợp đồng đang được kiểm tra. ContractFuzzer định nghĩa các test oracle mới cho nhiều loại lỗ hổng khác nhau và cài đặt EVM để giám sát quá trình thực thi hợp đồng, từ đó phát hiện các lỗ hổng thực sự. Kết quả thực nghiệm cho thấy ContractFuzzer đã phát hiện được hơn 459 lỗ hổng với độ chính xác rất cao, mỗi lỗ hổng đều được xác nhận thông qua phân tích thủ công.



Hình 5 Công trình nghiên cứu của Jiang và nhóm nghiên cứu

Là công cụ fuzzing đầu tiên dành riêng cho việc phát hiện lỗ hổng bảo mật của hợp đồng thông minh Ethereum, ContractFuzzer không chỉ phát hiện được nhiều loại lỗ hổng mà còn có tỷ lệ dương tính giả thấp hơn đáng kể so với công cụ xác minh bảo mật hiện đại Oyente. Trong nghiên cứu này, ContractFuzzer đã phát hiện tổng cộng 459 lỗ hổng từ 6991 hợp đồng thông minh được kiểm tra, bao gồm cả các lỗi nổi tiếng như DAO và Parity Wallet.

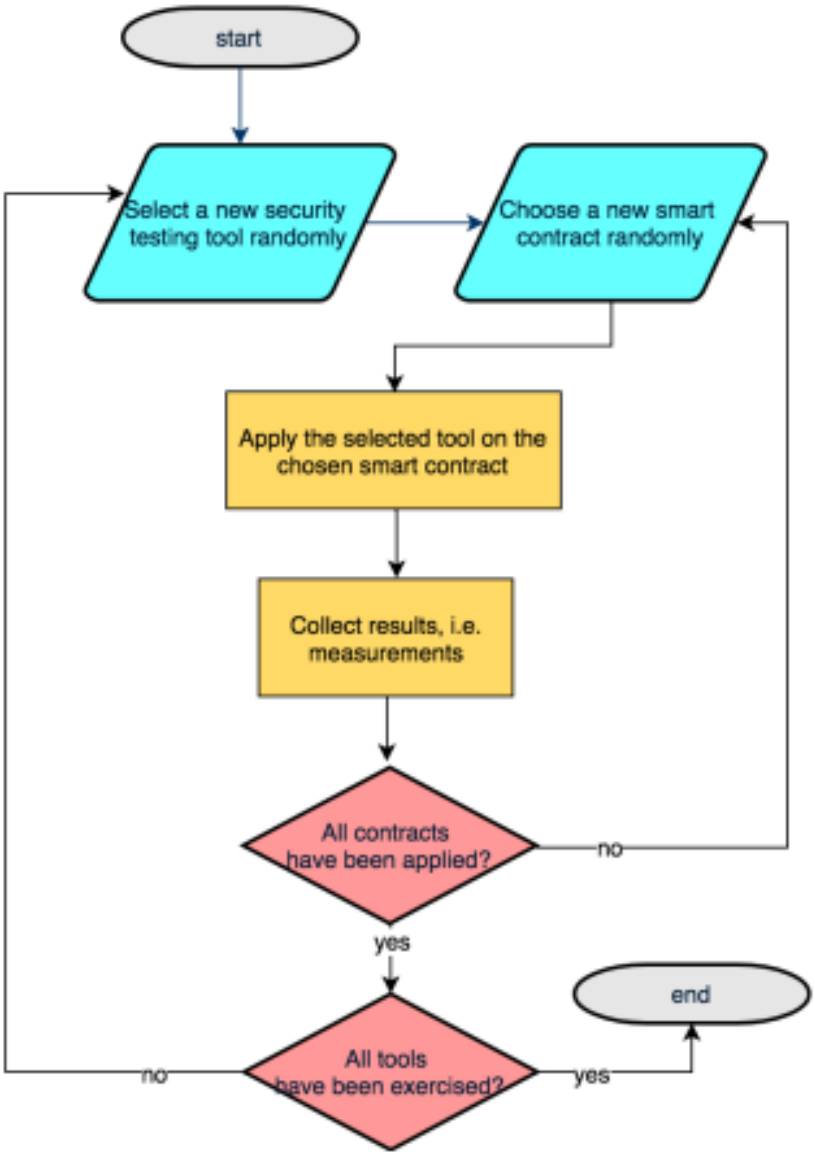
Hợp đồng thông minh trên Ethereum là các tài khoản có thể sở hữu số dư Ether và bộ nhớ riêng tư bền bỉ được quản lý bởi mã code. Chúng được lập trình bằng ngôn ngữ cấp cao như Solidity và được biên dịch thành bytecode của EVM. Mỗi giao dịch trong hợp đồng thông minh đều được tính phí gas để tránh lãng phí tài nguyên. Khi gas bị sử dụng hết trong quá

trình thực thi, một ngoại lệ out-of-gas sẽ được kích hoạt, hoàn nguyên tất cả các thay đổi đã thực hiện.

ContractFuzzer sử dụng phương pháp fuzzing để phát hiện lỗ hổng bảo mật thông qua việc phân tích giao diện ABI và tạo ra các đầu vào phù hợp. Bằng cách giám sát quá trình thực thi hợp đồng thông minh, công cụ này có thể phát hiện các lỗ hổng bảo mật với độ chính xác cao. Việc áp dụng ContractFuzzer trong thử nghiệm trên 6991 hợp đồng thông minh Ethereum đã chứng minh tính khả dụng và thực tiễn của nó.

Nghiên cứu cũng chỉ ra tiềm năng mở rộng của ContractFuzzer để phát hiện nhiều loại lỗ hổng khác liên quan đến EVM hoặc nền tảng blockchain cơ bản. Điều này cho thấy khả năng ứng dụng của ContractFuzzer trong việc kiểm tra bảo mật cho các nền tảng hợp đồng thông minh khác, mở ra hướng đi mới cho lĩnh vực này.

Công trình nghiên cứu của Parizi và cộng sự [32] tập trung vào việc đánh giá bảo mật và phát hiện lỗ hổng trong các hợp đồng thông minh trên blockchain Ethereum. Công trình bao gồm phân tích so sánh các ngôn ngữ lập trình hợp đồng thông minh, đề xuất các phương pháp xác minh chính thức, và phát triển các công cụ phân tích tĩnh để kiểm tra bảo mật.



Hình 6 Công trình nghiên cứu của Parizi và nhóm nghiên cứu

Phương pháp được sử dụng bao gồm thực nghiệm so sánh, phân tích chính thức, mô hình hóa hành vi hợp đồng, và phát triển các công cụ phân tích tĩnh. Công trình sử dụng phương pháp xác minh chính thức như chuyển đổi hợp đồng sang ngôn ngữ khác để kiểm chứng. Các công cụ phân tích tĩnh như ZEUS, Hydra Framework cũng được phát triển để phát hiện lỗ hổng bảo mật.

Phương pháp và công cụ được đề xuất cho thấy hiệu quả trong việc phát hiện nhiều loại lỗ hổng bảo mật phổ biến trong hợp đồng thông minh. Ví dụ, công cụ ZEUS phát hiện được 94.6% hợp đồng chứa lỗ hổng bảo mật nghiêm trọng. Phương pháp xác minh chính thức cũng giúp đảm bảo tính chính xác của hợp đồng so với mục đích thiết kế.

Hầu hết công trình đều thực hiện đánh giá thực nghiệm trên các bộ hợp đồng thông minh thực tế để kiểm chứng hiệu quả của phương pháp/công cụ đề xuất. Các tiêu chí đánh giá bao gồm khả năng phát hiện lỗ hổng, độ chính xác, hiệu suất thực thi. Một số công trình cũng so sánh với các phương pháp/công cụ hiện có để chứng minh ưu điểm.

Các hạn chế chung bao gồm phạm vi phát hiện lỗ hổng còn hạn chế, tỷ lệ cảnh báo sai còn cao, hiệu suất chưa tối ưu với các hợp đồng phức tạp. Một số phương pháp xác minh chính thức đòi hỏi chuyên môn cao để sử dụng. Ngoài ra, hầu hết các nghiên cứu mới chỉ tập trung vào nền tảng Ethereum và ngôn ngữ Solidity, chưa mở rộng sang các blockchain khác.

Chương 3: ĐỀ XUẤT GIẢI PHÁP

Trong phần này, em sẽ mô tả cách thức xây dựng và phát triển mô hình ChainSniper, với mục tiêu chính là tự động phát hiện các lỗ hổng trong hợp đồng thông minh trên mạng liên chuỗi. Em sẽ trình bày chi tiết về phương pháp và quy trình xây dựng mô hình, cũng như việc tạo ra tập dữ liệu dùng để đánh giá hiệu quả của các mô hình học máy trong hệ thống ChainSniper.

3.1. Tổng quan mô hình

3.1.1. Các thành phần chính của mô hình

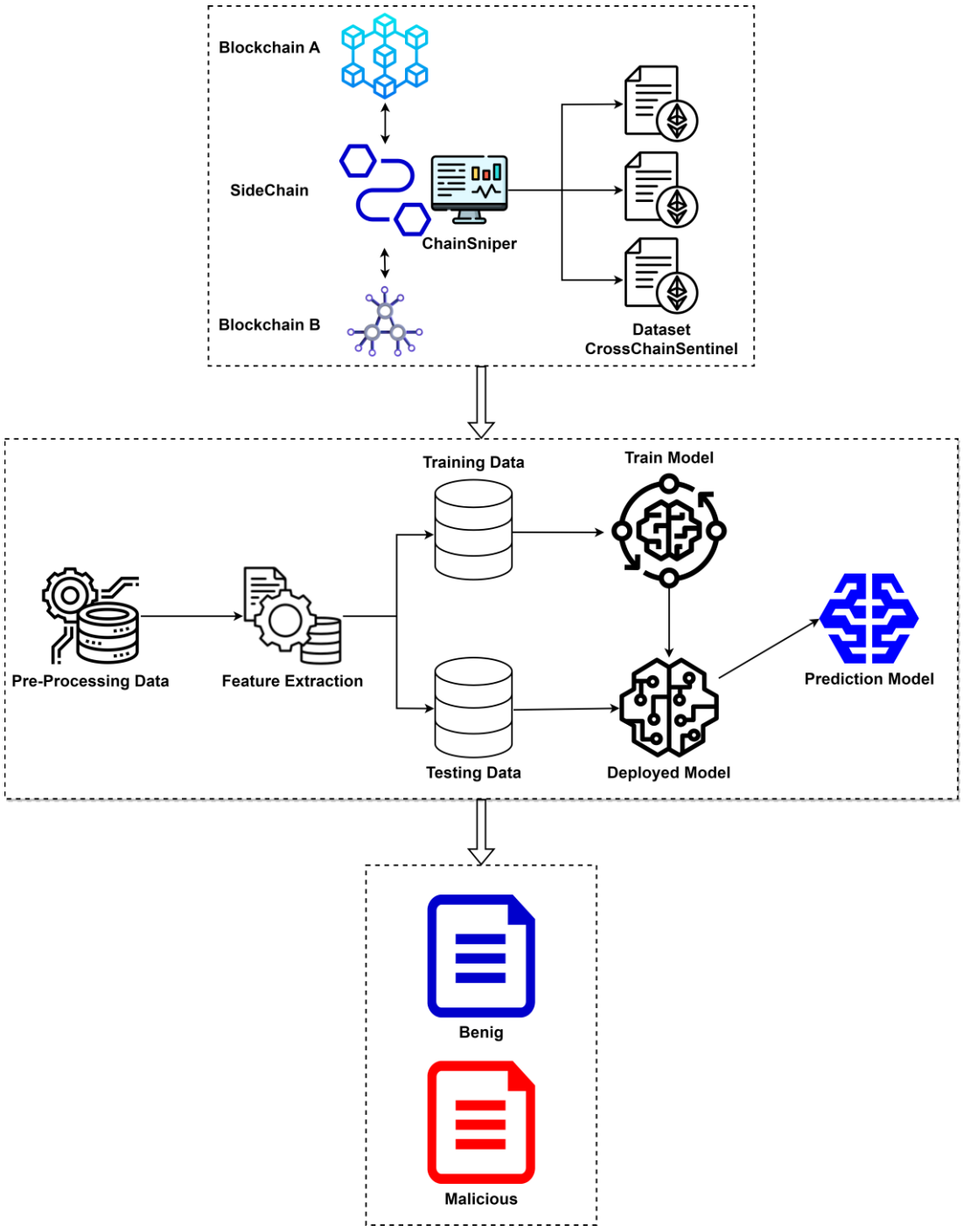
Trong phần này, em sẽ mô tả cách thức xây dựng và phát triển mô hình ChainSniper, với mục tiêu chính là tự động phát hiện các lỗ hổng trong hợp đồng thông minh trên mạng liên chuỗi. Em sẽ trình bày chi tiết về phương pháp và quy trình xây dựng mô hình, cũng như việc tạo ra tập dữ liệu dùng để đánh giá hiệu quả của các mô hình học máy trong hệ thống ChainSniper.

Mô hình chính của ChainSniper bao gồm 5 thành phần:

1. Ethereum Sepolia đóng vai trò là Blockchain A
2. Quorum đóng vai trò là Blockchain B
3. Sidechain đóng vai trò quan trọng trong việc chuyển dữ liệu và ghi lại các hợp đồng thông minh
4. Mô hình Học máy phát hiện lỗ hổng
5. Module phân loại trong hệ thống sử dụng sidechain có chức năng xác định và phân loại các hợp đồng thông minh

Hệ thống này gồm hai mạng blockchain liên kết thông qua cầu nối sidechain, giúp chuyển dữ liệu và lưu trữ thông tin của hợp đồng thông

minh. Dữ liệu này sẽ được xử lý và chuẩn bị trước khi đưa vào các mô hình học máy, đã được huấn luyện trên tập dữ liệu có nhãn, để nhận diện các hợp đồng thông minh độc hại. Sau đó, hiệu suất của các mô hình này được đánh giá một cách kỹ lưỡng. Cuối cùng, module sử dụng kết quả dự đoán từ các mô hình để phân loại hợp đồng thông minh trên sidechain thành lành tính hoặc độc hại. Hình minh họa mô hình chính:



Hình 7 Mô hình tổng quan

3.1.2. Mô hình cầu nối

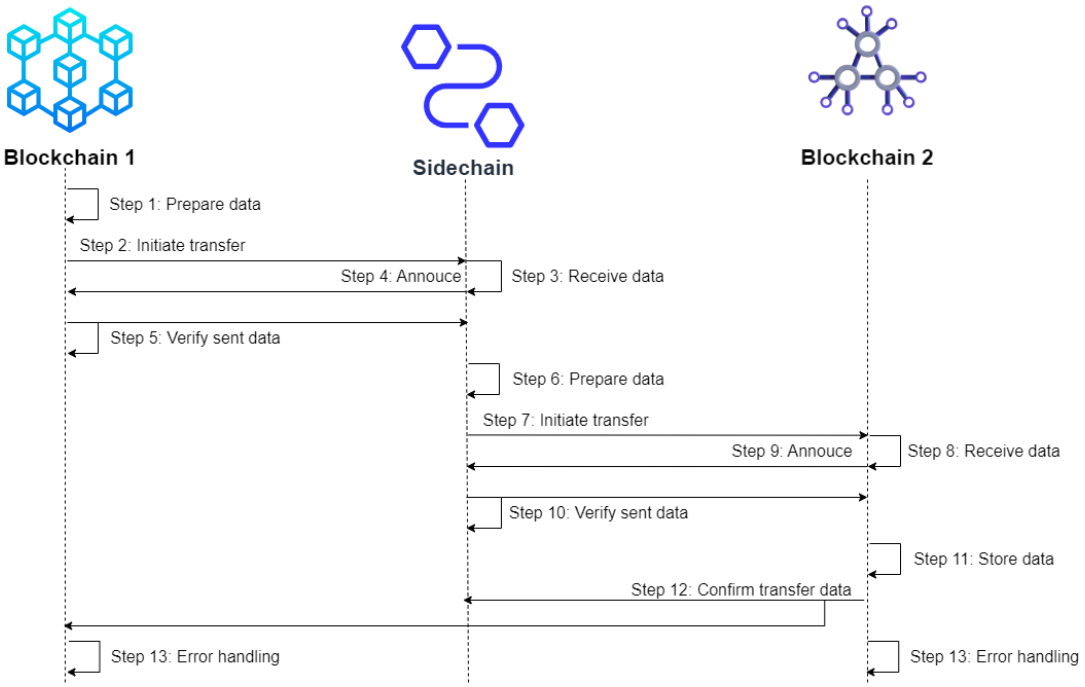
Hệ thống ChainSniper cho phép tương tác giữa các blockchain không đồng nhất qua cầu nối sidechain. Sidechain này đóng vai trò như một lớp trung gian, xử lý và chuyển dữ liệu giữa các mạng blockchain kết nối với nhau. Để có thể truyền tải dữ liệu, tài sản giữa các chuỗi được khóa lại ở một chuỗi này và giải khóa những biểu diễn tương đương ở chuỗi kia thông qua cơ chế neo hai chiều dựa trên hợp đồng đa chữ ký.

Khi có các giao dịch xuyên chuỗi diễn ra, các nút của sidechain sẽ ghi lại các thông tin, dữ liệu của giao dịch đó như địa chỉ của các hợp đồng thông minh tham gia, dấu thời gian diễn ra, các lời gọi hàm, các tham số được truyền vào, giá trị trả về, cũng như các ngoại lệ nếu có, sẽ được ghi lại trong nhật ký giao dịch. Nhật ký này sau đó được tổng hợp và xử lý để tạo nên một tập dữ liệu, cung cấp cái nhìn sâu sắc về hành vi và cách thức hoạt động của hợp đồng thông minh, cũng như các mẫu thực thi khác nhau của chúng.

Các bước thực hiện việc chuyển đổi dữ liệu qua sidechain:

1. Chuẩn bị dữ liệu: Xác định cấu trúc và bảo đảm tính nguyên vẹn của dữ liệu trước khi thực hiện chuyển đổi.
2. Khởi tạo quá trình chuyển đổi: Kết nối với sidechain để khởi tạo quá trình chuyển đổi.
3. Nhận dữ liệu ở sidechain: Sidechain nhận dữ liệu đến từ hệ thống blockchain A sau khi quá trình chuyển đổi được khởi tạo.
4. Sidechain thông báo nhận dữ liệu: blockchain A nhận thông báo từ sidechain về sự kiện chuyển đổi dữ liệu.

5. Kiểm tra Dữ liệu Trên Sidechain: Sidechain tiến hành xác nhận độ chính xác và tính nguyên vẹn của dữ liệu nhận được từ hệ thống blockchain chính.
6. Chuẩn bị dữ liệu (lần 2): Chuẩn bị lại dữ liệu cho lần chuyển đổi tiếp theo
7. Khởi tạo quá trình chuyển đổi (lần 2): Blockchain B tiếp tục kết nối với sidechain để khởi tạo lần chuyển đổi tiếp theo
8. Nhận dữ liệu (lần 2): Sidechain gửi dữ liệu mới đến Blockchain B.
9. Thông báo: Blockchain B nhận thông báo từ sidechain về sự kiện chuyển đổi dữ liệu lần thứ hai.
10. Xác minh dữ liệu đã gửi từ sidechain (lần 2): Sidechain tiến hành kiểm tra và đảm bảo tính chính xác cũng như nguyên vẹn của dữ liệu vừa được chuyển giao.
11. Bảo quản Dữ liệu: Dữ liệu được bảo lưu trên Blockchain B để sử dụng trong các hoạt động tiếp theo.
12. Xác nhận Chuyển đổi Dữ liệu: Blockchain B kiểm chứng rằng quá trình chuyển đổi đã hoàn tất thành công và dữ liệu đã được biến đổi một cách chính xác.
13. Quản lý Lỗi: Trong trường hợp xuất hiện lỗi, các mạng blockchain phải tiến hành xử lý, bao gồm cả việc gửi thông báo lỗi và ghi nhận chúng vào log để có thể theo dõi và giải quyết vấn đề.



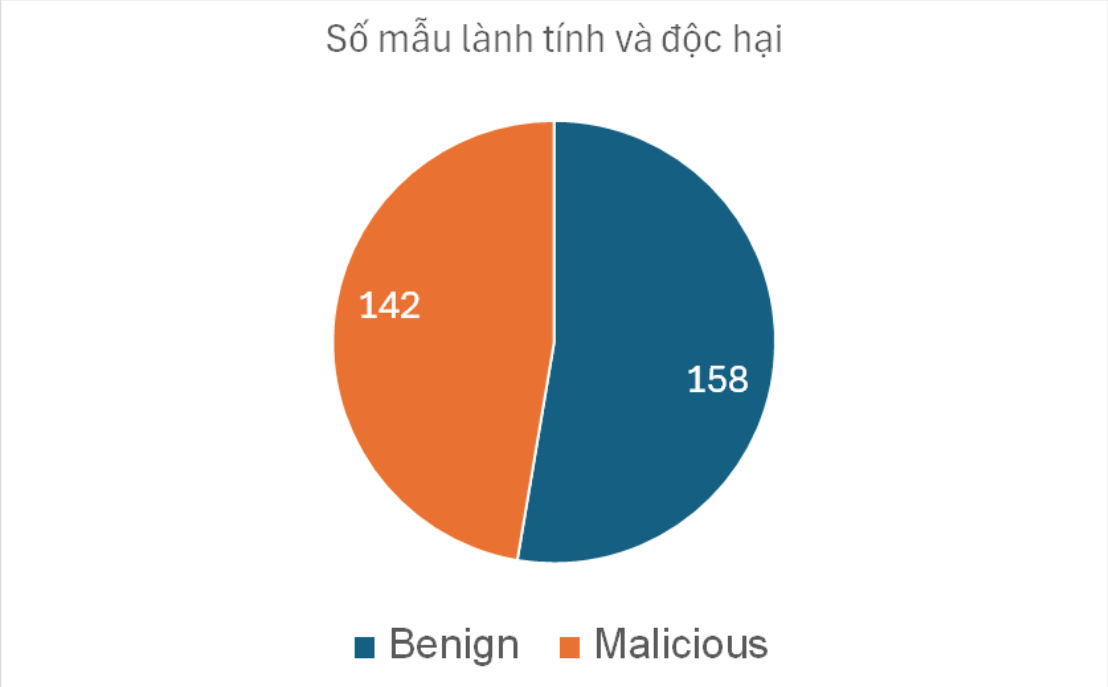
Hình 8 Các bước chuyển đổi dữ liệu qua cầu nối Sidechain

3.2. Phát hiện các lỗ hổng bằng phương pháp học máy và học sâu

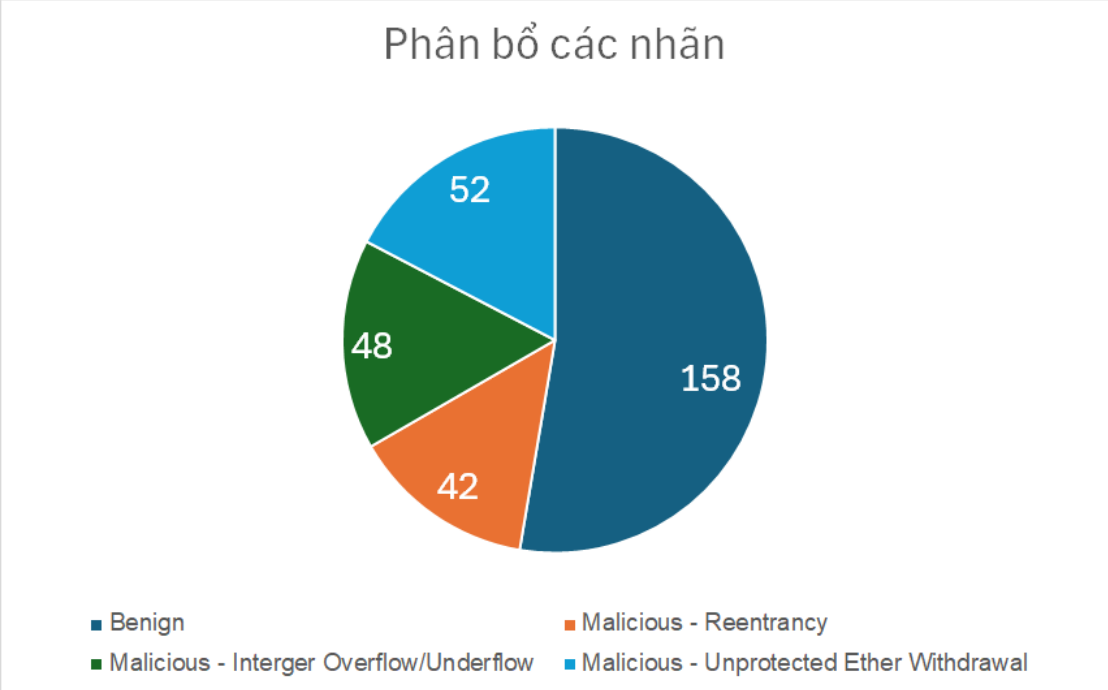
3.2.1. Xây dựng tập dữ liệu

CrossChainSentinel[1] được tạo ra với mục đích phân tích các lỗ hổng tiềm ẩn trong các hợp đồng thông minh trên cầu nối sidechain. Tập dữ liệu này chứa 300 tệp hợp đồng thông minh, trong đó 158 mẫu được xác định là lành tính, 142 mẫu còn lại được xếp vào loại độc hại. Cụ thể, số mẫu độc hại bao gồm 42 hợp đồng bị lỗ hổng tái tham chiếu (Reentrancy), 48 hợp đồng chứa lỗi tràn/cạn số nguyên (Interger Overflow/Underflow) và 52 hợp đồng có vấn đề rút Ether ra mà không được bảo vệ (Unprotected Ether Withdrawal).

[1] https://github.com/anhkiet1227/CrossChainSentinel



Hình 9 Phân bố mẫu lành tính và độc hại



Hình 10 Phân bố mẫu lành tính và độc hại tương ứng với các lỗ hổng

Các lỗ hổng này được xác định thông qua sử dụng các kỹ thuật từ các nghiên cứu trước như Smartbugs-wild[2], SolidiFi-benchmark[3] cùng Danh

[2] https://github.com/smartbugs/smartbugs-wild
[3] https://github.com/DependableSystemsLab/SolidiFI-benchmark

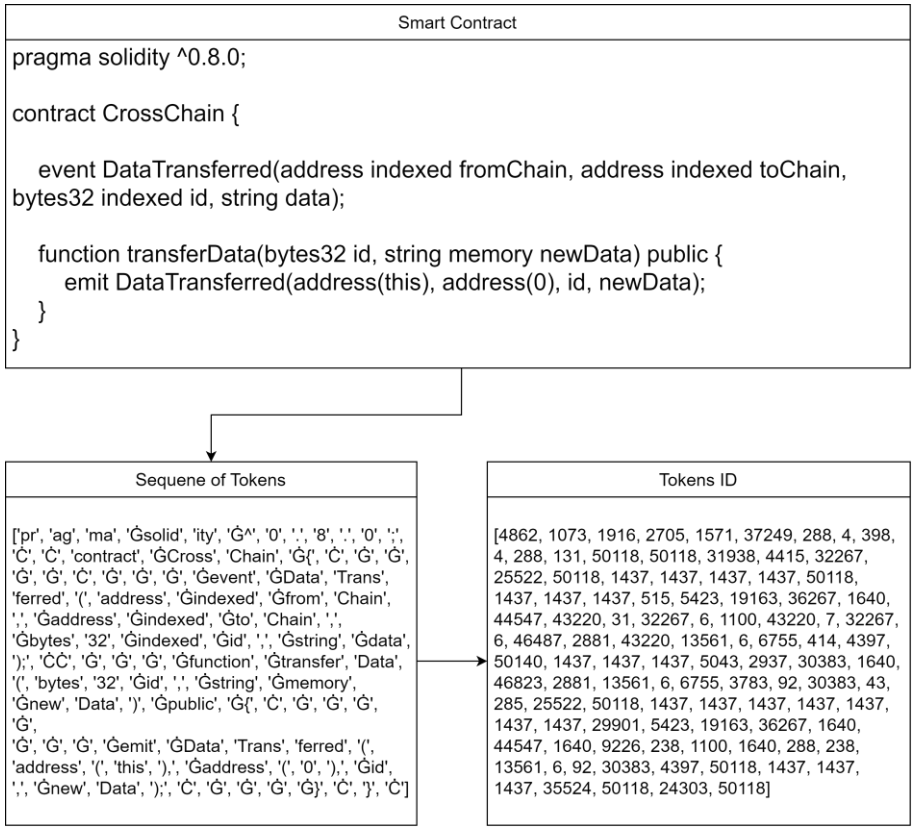
sách đen các địa chỉ Ethereum liên quan tới các cuộc tấn công đa chuỗi. Dữ liệu về cầu nối chuỗi chéo được mô phỏng theo 15 nhà cung cấp thực tế khác nhau gồm Commos, Avalanche, Chainlink... Tập dữ liệu CrossChainSentinel tập trung vào những lỗ hổng phổ biến có thể xảy ra khi tài sản được chuyển đổi giữa các blockchain thông qua hợp đồng sidechain.

Việc đưa ra các rủi ro tiềm ẩn đó trong một tập dữ liệu có nhãn rộng lớn giúp CrossChainSentinel hỗ trợ tăng cường khả năng kiểm toán, thử nghiệm và phát hiện những lỗ hổng nguy hiểm trong cơ chế cầu nối chuỗi chéo trước khi chúng được triển khai trên mạng chính. Sự đa dạng về nguồn chuỗi chéo và các loại lỗ hổng khiến đây là tập dữ liệu (dataset) lý tưởng để thực hiện công việc đánh giá công cụ và các mô hình nhằm tăng cường bảo mật chuỗi.

3.2.2. Gán nhãn các mẫu và xử lý dữ liệu

Tập dữ liệu cdhứa hai loại nhãn để phân loại hợp đồng thông minh. Tập nhãn đầu tiên là một tập nhãn nhị phân, xác định xem một hợp đồng có đặc điểm an toàn hay nguy hiểm. Tập nhãn thứ hai chi tiết hóa các nhãn đa lớp, phân loại các hợp đồng độc hại thành các loại lỗ hổng cụ thể như an toàn, có khả năng bị Reentrancy, Interger Overflow/Underflow và Unprotected Ether Withdrawal. Dữ liệu được xác định bằng các đặc điểm như tên file dự án, ID commit, nhãn mục tiêu, cấu trúc và nội dung của hợp đồng, cũng như số thứ tự file. Trong tập nhãn nhị phân, giá trị 0 biểu thị an toàn và 1 là nguy hiểm. Trong tập nhãn đa lớp, giá trị 0 là an toàn, 1 là lỗ hổng Reentrancy, 2 là lỗ hổng Integer Over/Underflow và 3 là lỗ hổng Unprotected Ether Withdrawal.

Để xử lý dữ liệu, em bắt đầu bằng việc chuyển đổi các file mã nguồn hợp đồng thông minh trở thành các token thông qua quá trình tokenization (Hình). Mỗi file code smart contract được phân chia thành các token, đơn vị nhỏ nhất của mã nguồn, như từ vựng, ký tự đặc biệt, hoặc biểu thức. Việc này giúp em biểu diễn cấu trúc và ý nghĩa của mã nguồn một cách có cấu trúc. Tiếp theo, mỗi token được chuyển đổi thành một tokenID, là một số nguyên duy nhất đại diện cho loại cụ thể của token đó. Quá trình này giúp giảm kích thước của dữ liệu và tạo ra một biểu diễn số hóa hiệu quả cho mã nguồn hợp đồng. Quá trình này tối ưu hóa việc đưa dữ liệu vào mô hình học máy, giúp mô hình có khả năng "nắm bắt" cấu trúc và ngữ cảnh của mã nguồn hợp đồng thông minh. Từ đó, em đã phát triển một bộ các tokenID, trong đó mỗi tokenID tương ứng với một phần cụ thể của mã nguồn. Điều này cho phép em tạo ra các vector đặc trưng cho mỗi hợp đồng thông minh, và cuối cùng, sử dụng chúng để đưa vào mô hình học máy nhằm phân loại các lỗ hổng một cách chính xác.



Hình 11 Quá trình xử lý dữ liệu

3.2.3. Ứng dụng các mô hình học máy trong việc phát hiện các lỗ hổng

Việc chọn các phương pháp học máy bởi chúng cho phép tự động phát hiện lỗ hổng bảo mật trong các hợp đồng thông minh bằng cách phân biệt các mẫu trong mã hợp đồng và cấu trúc. So với kiểm toán thủ công, các mô hình học máy có thể phân tích hợp đồng một cách hiệu quả và nhất quán hơn. Em thử nghiệm vài mô hình học máy cổ điển, bao gồm Cây quyết định, Rừng ngẫu nhiên, Máy vector hỗ trợ, XGBoost và Hồi quy Logistic. Nhờ khả năng giải thích, các mô hình này cho phép hiểu được lý do tại sao một số hợp đồng bị gắn cờ là dễ bị tấn công.

Decision tree và Random forest xây dựng các quy tắc phân cấp dựa trên các thuộc tính mã để phân loại hợp đồng. Trong khi đó, SVM tìm ranh giới tối ưu giữa hợp đồng dễ bị tấn công và an toàn trong không gian đặc trưng. XGBoost tiếp tục tăng độ chính xác thông qua một tập hợp các học viên yếu. Em trích xuất các đặc trưng số liệu có giá trị thông tin như độ phức tạp hàm, luồng điều khiển và các mẫu cú pháp.

Bằng cách huấn luyện trên dữ liệu đại diện thích hợp của cả ví dụ có chiều hướng tích cực lẫn chiều hướng tiêu cực, các mô hình học cách phát hiện bền vững các lỗ hổng an toàn thông tin. Em áp dụng các kỹ thuật này trong ChainSniper để xây dựng một máy quét tự động cho các lỗ hổng như Reentrancy, Interger Overflow/Underflow và Unprotected Ether Withdrawal. Tính linh hoạt của học máy cung cấp một phương pháp luận để phát hiện lỗ hổng trong khi tránh nỗ lực thủ công mở rộng

3.2.4. Ứng dụng các mô hình học sâu trong việc phát hiện các lỗ hổng

Quá trình lựa chọn các phương pháp học sâu được thực hiện vì khả năng của chúng trong việc mô hình hóa các mối quan hệ phức tạp. của logic mã và phụ thuộc mà không cần trích xuất thủ công các đặc trưng. Các kỹ thuật như CNN, RNN, LSTM và các mô hình Transformer như RoBERTa cho phép học từ đầu đến cuối trực tiếp từ mã nguồn hợp đồng thông minh thô để phát hiện các mẫu cú pháp và ngữ nghĩa phức tạp mà đặc trưng cho các lỗ hổng.

Việc thử nghiệm các mạng nơ-ron sâu bao gồm các mô hình tuần tự dựa trên LSTM có khả năng diễn giải các luồng điều khiển, CNN trích xuất các mẫu cú pháp cục bộ và bộ mã hóa Transformer như RoBERTa cung cấp các vector từ ngữ cảnh phong phú. Nhờ xử lý phân cấp theo tầng, các mạng này tự động học quan hệ tiềm ẩn giữa các token và cấu trúc dẫn đến sự hiểu biết về logic cho thấy lỗ hổng bảo mật.

Hơn thế nữa, các phụ thuộc dài hạn được mô hình hóa bởi LSTM có thể truy tìm các luồng thông tin trong hợp đồng để xác định các vấn đề kiểm tra cuộc gọi không kiểm soát. Các vector ngữ cảnh của RoBERTa có khả năng phân biệt các sắc thái giữa các cấu trúc có vẻ tương tự nhưng có thể hoặc không dễ bị tấn công. Các lớp chú ý có thể giải thích được của học sâu cũng hỗ trợ xác định các thành phần gây vấn đề. Việc áp dụng các kỹ thuật này trong ChainSniper để xây dựng các máy quét tự động có thể cờ các nguy cơ Reentrancy, Interger Overflow/Underflow và Unprotected Ether Withdrawal.

Chương 4: THỰC NGHIỆM VÀ ĐÁNH GIÁ

Trong phần này, em sẽ mô tả quá trình thiết lập thực nghiệm, bao gồm việc cấu hình môi trường thực nghiệm, xác định các chỉ số đánh giá và phác thảo các kịch bản thực nghiệm. Em cũng sẽ trình bày về kết quả đạt được, cả về hiệu quả và thời gian thực thi của mô hình.

4.1. Thiết lập thực nghiệm

4.1.1. Môi trường thực nghiệm

Để chạy mô hình liên chuỗi, cần có một môi trường với hiệu suất CPU cao, dung lượng lưu trữ lớn và bộ nhớ đủ rộng. Chính vì thế, nhóm đã thực hiện việc thiết lập một môi trường thích hợp để đáp ứng các yêu cầu như bảng bên dưới:

Thiết bị	Blockchain A	SideChain	Blockchain B
CPU	4 cores	4 cores	4 cores
GPU	N/A	N/A	N/A
RAM	8 GB	8 GB	8 GB
Hard Drive	60 GB	60 GB	60 GB
OS	Ubuntu 22.04	Ubuntu 22.04	Ubuntu 22.04

Bảng 1 Môi trường thực nghiệm mạng liên chuỗi

Dựa trên cơ sở hạ tầng mạng liên chuỗi đã được mô tả trong Bảng , có một phương pháp khác để triển khai hệ thống này mà không yêu cầu nhiều phần cứng vật lý. Đó là việc sử dụng một máy đơn lẻ với cấu hình phần cứng vừa phải để tạo ra các môi trường ảo hóa. Mỗi môi trường ảo này có thể được cấu hình để mô phỏng một node riêng biệt của Blockchain A, SideChain, và Blockchain B. Trong trường hợp này em sử dụng một máy với 4 CPU, 16 GB RAM, 500 GB ổ cứng SSD và Hệ điều

hành Window 10 có thể được sử dụng để tạo ra ba môi trường ảo, mỗi môi trường ảo chạy các mạng Blockchain A, SideChain, và Blockchain B.

Điều này không chỉ giúp tiết kiệm chi phí cho phần cứng mà còn tạo điều kiện thuận lợi cho việc quản lý và triển khai. Hơn nữa, trong khuôn khổ của dự án Axelar, phương pháp này cung cấp một cách linh hoạt và hiệu quả để mô phỏng và thử nghiệm các kịch bản liên kết chuỗi khác nhau, cho phép các nhà phát triển và nhà nghiên cứu kiểm tra và tối ưu hóa các giải pháp mạng liên chuỗi một cách dễ dàng hơn.

Để huấn luyện mô hình học máy, cần một môi trường có khả năng chạy liên tục trong thời gian dài với hiệu suất cao từ CPU và GPU, cùng với dung lượng lưu trữ lớn và bộ nhớ rộng lớn. Do vậy, để đáp ứng những yêu cầu tính toán nghiêm ngặt này, nhóm đã thiết lập môi trường như bảng bên dưới:

	Machine 1	Machine 2	Google Collab
CPU	4 cores	4 cores	4 cores
GPU	N/A	N/A	T4
RAM	8 GB	16 GB	12.7 GB
Hard drive	60 GB	60 GB	166 GB
OS	Ubuntu 22.04	Window 10	Ubuntu 22.04

Bảng 2 Môi trường thực nghiệm các phương pháp học máy và học sâu

4.1.2. Chỉ số đánh giá

Trong nghiên cứu này, em đã chọn 5 chỉ số để đánh giá hiệu suất của mô hình, bao gồm: Accuracy (Độ chính xác), Precision (Mức độ chính xác), Recall (Độ phủ), F1-score (Điểm F1) và Thời gian dự đoán. Để tính toán bốn chỉ số đầu tiên, em sử dụng ma trận confusion (confusion matrix),

một công cụ hữu ích trong việc đánh giá các mô hình phân loại. Confusion matrix cho phép xác định nhanh chóng các chỉ số này qua việc phân tích số lượng dữ liệu thực tế thuộc vào một lớp nhưng lại bị dự đoán nhầm lẫn vào lớp khác, dựa trên các khái niệm về True Positive, False Positive, True Negative và False Negative. Điều này giúp hiểu rõ hơn về cách thức mô hình phân loại và đánh giá chất lượng dự đoán của nó:

- True Positive (TP) là những mẫu được dự đoán đúng thuộc lớp positive và thực tế cũng thuộc lớp positive.
- True Negative (TN) là những mẫu được dự đoán đúng thuộc lớp negative và thực tế cũng thuộc lớp negative.
- False Positive (FP) là những mẫu được dự đoán nhầm thuộc lớp positive nhưng thực tế lại thuộc lớp negative.
- False Negative (FN) là những mẫu được dự đoán nhầm thuộc lớp negative nhưng thực tế lại thuộc lớp positive.

Các độ đo đánh giá như Accuracy, Precision, Recall, và F1-Score được xác định và tính toán dựa trên số lượng các mẫu phân loại là True Positive (TP), True Negative (TN), False Positive (FP) và False Negative (FN). Các định nghĩa và công thức của chúng như sau:

Accuracy là chỉ số đánh giá độ chính xác toàn diện của mô hình. Nó được tính bằng cách lấy tỷ lệ phần trăm của tổng số dự đoán đúng (bao gồm cả true positives và true negatives) so với tổng số quan sát. Khi giá trị Accuracy càng cao, điều đó cho thấy mô hình có khả năng dự đoán càng chính xác.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision đo lường độ chính xác của những dự đoán dương tính mà mô hình thực hiện. Nó được xác định bằng cách chia số lượng true positives (các trường hợp dự đoán đúng là dương tính) cho tổng số dự đoán dương tính (bao gồm cả true positives và false positives). Giá trị Precision cao chỉ ra rằng mô hình có độ tin cậy lớn khi dự đoán các quan sát là dương tính.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall (còn được gọi là Sensitivity) cho biết khả năng của mô hình trong việc xác định tất cả các trường hợp dương tính thực sự. Chỉ số này được tính bằng cách chia số lượng true positives (số trường hợp dương tính mà mô hình dự đoán chính xác) cho tổng số trường hợp dương tính thực tế. Khi giá trị Recall càng cao, nó cho thấy mô hình càng có khả năng phát hiện tất cả các trường hợp dương tính mà không bỏ lỡ.

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1-Score là chỉ số đánh giá sự cân bằng giữa Precision và Recall. Nó được tính bằng công thức lấy trung bình điều hòa của Precision và Recall. F1-Score cung cấp một cái nhìn toàn diện về độ chính xác và độ phủ của mô hình. Một mô hình với F1-Score cao được xem là có hiệu suất tốt, vì nó thể hiện cả hai khía cạnh quan trọng là Precision và Recall đều được cân nhắc và tối ưu hóa.

$$\text{F1-score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Thời gian dự đoán (Inference Time) cũng là một chỉ số quan trọng khác trong việc đánh giá hiệu suất của mô hình máy học. Chỉ số này đo lường thời gian mà mô hình cần để đưa ra dự đoán cho một hoặc một loạt các quan sát mới. Thời gian dự đoán càng ngắn, thì mô hình càng được xem là hiệu quả, điều này rất quan trọng trong các ứng dụng cần phản hồi nhanh chóng, như trong các tình huống yêu cầu thời gian thực. Mô hình có thời gian dự đoán nhanh sẽ trả về kết quả nhanh hơn, cho phép hệ thống đưa ra quyết định hoặc hành động kịp thời. Ngược lại, thời gian dự đoán chậm có thể khiến trải nghiệm người dùng kém hơn. Vì vậy, khi thiết kế và đánh giá mô hình, thời gian dự đoán là một yếu tố then chốt cần xem xét.

4.1.3. Kịch bản thực nghiệm

Trước khi em bước vào các giai đoạn chi tiết của quy trình này, hãy cùng xem xét tổng quan về quy trình mà em đã thiết kế để kiểm tra và phân loại lỗi trong các smart contract trên blockchain. Đầu tiên, em xây dựng một mô hình cầu nối blockchain, tạo ra một môi trường mô phỏng cho việc tạo ra các smart contract. Tiếp theo, em thu thập và gắn nhãn cho các smart contract để tạo ra một tập dữ liệu đủ lớn để huấn luyện và kiểm tra mô hình. Sau đó, em sử dụng các thuật toán học máy để xây dựng các mô hình phân loại lỗi trong smart contract, và quá trình huấn luyện được thực hiện trên tập dữ liệu đã chuẩn bị trước đó. Để đánh giá hiệu suất, em sử dụng các chỉ số như Accuracy, Precision, Recall, và F1-Score, và cuối cùng, em triển khai mô hình tốt nhất và kiểm tra nó trên giao diện ứng

dụng để đảm bảo tính thực tế và hiệu quả trong môi trường thực tế. Hãy cùng đi vào từng bước chi tiết của quá trình này:

1. Xây dựng mô hình cầu nối blockchain: Mục tiêu của giai đoạn này là mô phỏng quá trình sáng tạo các smart contract trên blockchain. em thiết lập một mô hình cầu nối để tái hiện các bước quan trọng trong việc tạo ra smart contract, tạo nên một môi trường mô phỏng cho nghiên cứu tiếp theo.
2. Thu thập và gắn nhãn mác smart contract: Sau khi mô phỏng, em thu thập các smart contract được tạo ra từ mô hình cầu nối. Các smart contract này sau đó được gắn nhãn để tạo thành tập dữ liệu cho quá trình đào tạo và kiểm tra mô hình.
3. Áp dụng thuật toán học máy: Với tập dữ liệu đã có, em áp dụng các thuật toán học máy như máy học và học sâu để xây dựng các mô hình phân loại lỗi trong smart contract. Các thuật toán này được chọn để hiểu và dự đoán các lỗi có thể xuất hiện trong smart contract.
4. Huấn luyện mô hình: Quá trình huấn luyện được thực hiện trên tập dữ liệu đã được chuẩn bị. Mô hình học từ dữ liệu để hiểu các đặc trưng quan trọng và mối quan hệ giữa chúng để có khả năng dự đoán lỗi một cách chính xác.
5. Đánh giá và so sánh mô hình: Sau khi huấn luyện, em đánh giá hiệu suất của các mô hình bằng cách sử dụng các chỉ số như Accuracy, Precision, Recall và F1-Score. Sự so sánh giữa các mô hình giúp em lựa chọn mô hình có hiệu suất tốt nhất.
6. Lựa chọn mô hình tốt nhất: Dựa trên kết quả đánh giá, mô hình có hiệu suất tốt nhất được lựa chọn để tiếp tục triển khai và kiểm tra trong các tình huống thực tế.

7. Triển khai trên giao diện ứng dụng: Mô hình tốt nhất được triển khai trên giao diện ứng dụng để kiểm tra và xác nhận tính khả thi của nó trong môi trường thực tế.

4.2. Kết quả thực nghiệm

4.2.1. Kết quả về mặt hiệu suất

Bảng kết quả cung cấp một cái nhìn tổng quan về khả năng phát hiện lỗ hổng bảo mật trong các smart contract liên chuỗi của các mô hình. Qua các độ đo này, chúng ta có thể thăm dò cận kề hiệu quả của từng mô hình. Bảng thể hiện kết quả hiệu suất. RoBERTa nổi bật như mô hình dẫn đầu về khả năng phân loại lỗ hổng trong smart contract, với độ chính xác cao nhất đạt 96.67% so với các mô hình khác. Điểm nổi bật của RoBERTa so với các mô hình khác là độ đo và F1 Score lần lượt đạt 100% và 93.33\% - cao nhất trong tất cả các mô hình, cho thấy sự cân bằng tốt giữa độ chính xác và độ nhạy.

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.8519	0.8209	0.8730	0.8462
Random Forest	0.7312	0.6324	**1.0000**	0.7748
XGBoost	0.6211	0.5485	0.9924	0.7065
SVM	0.8458	0.7551	0.9911	0.8571
Logistic Regression	0.9000	0.9071	0.8864	0.8967
CNN	0.9000	0.8235	**1.0000**	0.9032
LSTM	0.5500	0.5500	**1.0000**	0.7100
FNN	0.8125	0.8636	0.7037	0.7755
RoBERTa	**0.9667**	**1.0000**	0.8750	**0.9333**

Bảng 3 Hiệu suất của các mô hình học máy và học sâu trong việc phát hiện lỗ hổng bảo mật trong các hợp đồng thông minh liên chuỗi

4.2.2. Kết quả về mặt thời gian

Quá trình thực hiện của các mô hình cho thấy sự biến thiên đáng kể về mặt thời gian. Bảng bên dưới trình bày kết quả thời gian thực thi của chúng. Mô hình như Decision Tree, Random Forest và SVM chỉ mất khoảng 3.3 - 3.7 giây để thực hiện, phản ánh sự đơn giản trong cấu trúc của chúng, đặc biệt là Decision Tree và Random Forest.

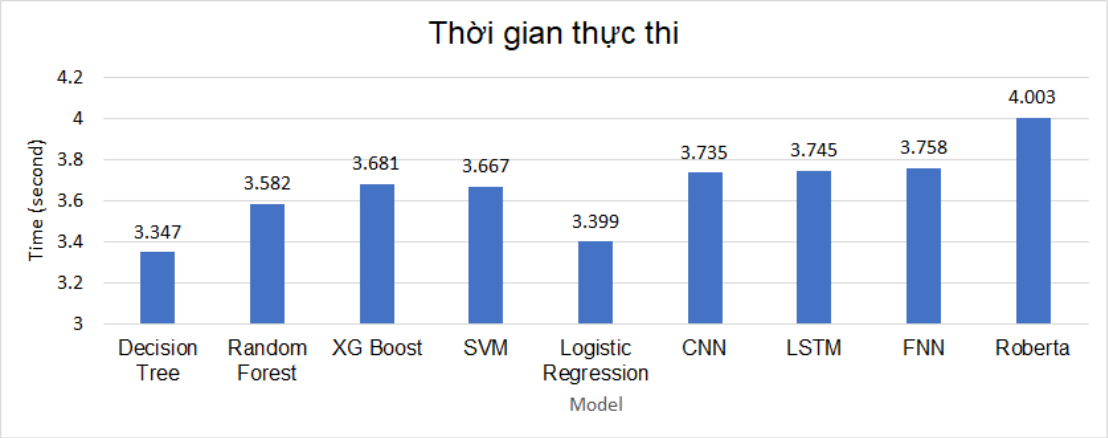
Model	Time Performance (seconds)
Decision Tree	3.347
Random Forest	3.582
XGBoost	3.681
SVM	3.667
Logistic Regression	3.399
CNN	3.735
LSTM	3.745
FNN	3.758
RoBERTa	4.003

Bảng 4 Thời gian thực thi của các mô hình học máy và học sâu trong việc phát hiện lỗ hổng bảo mật trong các hợp đồng thông minh liên chuỗi

Mặt khác, mô hình RoBERTa mất nhiều thời gian nhất để hoàn thành, đạt 4.003 giây. Sự phức tạp của kiến trúc transformer có thể là nguyên nhân, nhưng thời gian này vẫn nằm trong phạm vi chấp nhận được, đặc biệt khi xét đến khả năng dự đoán xuất sắc của mô hình.

Việc đánh giá thời gian thực hiện của mô hình là quan trọng, đặc biệt khi cân nhắc triển khai mô hình trong các ứng dụng cần phản ứng nhanh hoặc đào tạo liên tục. Trong trường hợp các ứng dụng cần thời gian phản hồi nhanh, việc chọn mô hình như Decision Tree hoặc SVM có thể phù hợp,

trong khi đó, RoBERTa vẫn là sự lựa chọn mạnh mẽ cho các nhiệm vụ đòi hỏi độ chính xác cao, mặc dù có thời gian thực hiện lớn hơn. HÌnh bên dưới minh họa sự khác biệt về thời gian thực thi giữa các mô hình.



Hình 12 Thời gian thực thi của các mô hình học máy và học sâu trong việc phát hiện lỗ hổng bảo mật trong các hợp đồng thông minh liên chuỗi

4.3. Thảo luận

Hệ thống được đánh giá về khả năng phát hiện bảo mật các lỗ hổng và hành vi độc hại trong các hợp đồng thông minh xuyên chuỗi. Các mô hình học máy, đặc biệt là Roberta, có hiệu quả rất cao trong việc phân loại và xác định lỗ hổng, với độ chính xác vượt 96%. Điều này cho thấy khả năng của hệ thống trong việc diễn giải ngữ nghĩa hợp đồng để học các mô hình dự đoán chính xác phần mã độc hại.

Khả năng phát hiện bảo mật xuất phát từ cách tiếp cận độc đáo trong việc tạo tập dữ liệu xuyên chuỗi của hệ thống. Theo đó, dữ liệu hợp đồng thô sẽ được biến đổi thành các đặc trưng mạnh mẽ. Tổng thể, kết quả khẳng định độ bảo mật của hệ thống cho các mạng blockchain liên kết thông qua các mô hình học máy được huấn luyện trên tập dữ liệu xuyên chuỗi mới.

Việc tích hợp ChainSniper vào hệ thống cũng tăng cường thêm lớp bảo vệ thông qua khả năng phân tích động các phụ thuộc và tương tác giữa các hợp đồng xuyên blockchain. Bằng cách giám sát theo thời gian thực, ChainSniper có thể xác định các dấu hiệu của nỗ lực khai thác lỗ hổng. Sự kết hợp này cho phép giám sát bảo mật chủ động trên các mạng blockchain liên kết.

Về mặt thời gian giao động, các mô hình học máy đạt mức 3.3 - 3.7 giây. Trong khi đó các mô hình học sâu cho thời gian 3.7 - 4..0 giây. Cao nhất là mô hình RoBERTa với 4.003 giây. Mặc dù vậy, xét về yếu tố như độ chính xác đều rất cao và có khả năng ứng dụng thực tế.

Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong phần kết luận này, em sẽ tổng kết những điểm chính của công trình nghiên cứu và đề cập đến các hướng phát triển tiềm năng trong tương lai cho mô hình.

5.1. Kết luận

Nghiên cứu này mang lại một bước tiến quan trọng trong việc cải thiện bảo mật cho hệ sinh thái hợp đồng thông minh liên kết chuỗi. Cụ thể, nhóm nghiên cứu đã phát triển một mô hình mới có tên ChainSniper, dựa trên công nghệ sidechain. Hơn nữa, họ cũng tạo ra một tập dữ liệu đặc biệt, CrossChainSentinel, chứa 300 mẫu mã nguồn hợp đồng thông minh đã được gán nhãn. Sử dụng tập dữ liệu này, ChainSniper được thiết kế để kết hợp nhiều mô hình máy học tiên tiến, bao gồm cả các phân loại học sâu, nhằm phát hiện các hợp đồng thông minh độc hại trong môi trường blockchain liên kết. Qua các thử nghiệm và đánh giá, mô hình này cho thấy độ chính xác ấn tượng lên tới 96\% trong việc phát hiện lỗ hổng hợp đồng thông minh trên bộ dữ liệu mới. Kết quả này là minh chứng cho tiềm năng lớn của việc kết hợp công nghệ sidechain và máy học trong việc tăng cường bảo mật cho hệ thống hợp đồng thông minh liên kết chuỗi. Nghiên cứu này không chỉ cung cấp một mô hình hiệu quả với độ chính xác cao mà còn mở ra hướng tiếp cận mới để cải thiện an toàn cho hợp đồng thông minh hoạt động trong môi trường đa blockchain. Như vậy, nghiên cứu đã định hình cơ hội và hướng phát triển mới trong việc nâng cao tính bảo mật cho hệ thống hợp đồng thông minh, góp phần thúc đẩy sự phát triển của công nghệ blockchain trong tương lai.

5.2. Hướng phát triển

Nghiên cứu này mở ra nhiều hướng phát triển tiềm năng cho mô hình ChainSniper. Một trong những khả năng chính là việc mở rộng bộ dữ liệu CrossChainSentinel, bao gồm cả việc bổ sung thêm các mẫu hợp đồng thông minh lành tính và độc hại. Việc tăng cường bộ dữ liệu sẽ cải thiện khả năng tổng quát hóa và độ chính xác của mô hình học máy. Ngoài ra, khám phá và tích hợp các kiến trúc máy học tiên tiến như mô hình Transformer và áp dụng kiến thức từ lĩnh vực xử lý ngôn ngữ tự nhiên có thể giúp cải thiện khả năng hiểu ngữ nghĩa của mã smart contract, tăng cường chất lượng trong việc phát hiện lỗ hổng. Sự kết hợp này sẽ khai thác hiệu quả ngữ liệu để phân tích mã máy, mở ra một phương thức mới trong việc đảm bảo an ninh cho hợp đồng thông minh. Mục tiêu dài hạn là liên tục cải thiện ChainSniper để nó có thể tự động phát hiện điểm yếu trong hợp đồng thông minh trước khi chúng được triển khai, giúp thực hiện kiểm định chất lượng, nâng cao độ tin cậy và an toàn cho các ứng dụng phi tập trung trên nhiều nền tảng blockchain khác nhau. Tóm lại, sự kết hợp giữa việc mở rộng bộ dữ liệu, tối ưu hóa thuật toán và nâng cao các tính năng sẽ giúp mô hình ChainSniper ngày càng hoàn thiện, trở thành một giải pháp quan trọng trong việc kiểm tra toàn diện hợp đồng thông minh, giảm thiểu rủi ro bảo mật cho các hệ thống phức tạp liên kết chuỗi.

TÀI LIỆU THAM KHẢO

[1] P. Patel and H. Patel, “Achieving a secure cloud storage mechanism using blockchain technology,” p. 130–142, 2023. [Online]. Available: http://dx.doi.org/10.7763/IJCTE.2023.V15.1342

[2] M. Kumarathunga, R. N. Calheiros, and A. Ginige, “Sustainable microfinance outreach for farmers with blockchain cryptocurrency and smart contracts,” p. 9–14, 2022. [Online]. Available: http://dx.doi.org/10.7763/IJCTE.2022.V14.1304

[3] T. Xie, J. Zhang, Z. Cheng, F. Zhang, Y. Zhang, Y. Jia, D. Boneh, and D. Song, “zkbridge: Trustless cross-chain bridges made practical,” in Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, pp. 3003–3017.

[4] P. Han, Z. Yan, W. Ding, S. Fei, and Z. Wan, “A survey on cross-chain technologies,” Distributed Ledger Technologies: Research and Practice, vol. 2,no. 2, pp. 1–30, 2023.

[5] Y. Hei, D. Li, C. Zhang, J. Liu, Y. Liu, and Q. Wu, “Practical agentchain: A compatible cross-chain exchange system,” Future Generation Computer Systems, vol. 130, pp. 207–218, 2022.

[6] R. Lan, G. Upadhyaya, S. Tse, and M. Zamani, “Horizon: A gas-efficient, trustless bridge for cross-chain transactions,” arXiv preprintarXiv:2101.06000, 2021.

[7] K. Qin and A. Gervais, “An overview of blockchain scalability, interoperability and sustainability,” Hochschule Luzern Imperial College London Liquidity Network, pp. 1–15, 2018.

[8] T. Hardjono, “Blockchain gateways, bridges and delegated hash-locks,” arXiv preprint arXiv:2102.03933, 2021.

[9] A. Singh, K. Click, R. M. Parizi, Q. Zhang, A. Dehghantanha, and K.-K. R. Choo, "Sidechain technologies in blockchain networks: An examination and state-of-the-art review," Journal of Network and Computer Applications, vol. 149, p. 102471, 2020.

[10] M. H. Miraz and D. C. Donald, "Atomic cross-chain swaps: development, trajectory and potential of non-monetary digital token swap facilities,"arXiv preprint arXiv:1902.04471, 2019.

[11] J. Zhang, J. Gao, Y. Li, Z. Chen, Z. Guan, and Z. Chen, "Xscope: Hunting for cross-chain bridge attacks," in Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, 2022, pp. 1–4.

[12] S. Sayeed, H. Marco-Gisbert, and T. Caira, "Smart contract: Attacks and protections," IEEE Access, vol. 8, pp. 24 416–24 427, 2020.

[13] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts (sok)," in Principles of Security and Trust: 6th International Conference, POST 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings 6. Springer, 2017, pp. 164–186.

[14] S. Wang, Y. Yuan, X. Wang, J. Li, R. Qin, and F.-Y. Wang, "An overview of smart contract: architecture, applications, and future trends," in 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018, pp. 108–113.

[15] T. H.-D. Huang, "Hunting the ethereum smart contract: Color-inspired inspection of potential attacks," arXiv preprint arXiv:1807.01868, 2018.

[16] L. Zhang, W. Chen, W. Wang, Z. Jin, C. Zhao, Z. Cai, and H. Chen, "Cbgru: A detection method of smart contract vulnerability based on a hybrid model," Sensors, vol. 22, no. 9, p. 3577, 2022.

[17] E. Lai and W. Luo, “Static analysis of integer overflow of smart contracts in ethereum,” in Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy, 2020, pp. 110–115.

[18] M. Staderini, C. Palli, and A. Bondavalli, “Classification of ethereum vulnerabilities and their propagations,” in 2020 Second International Conference on Blockchain Computing and Applications (BCCA). IEEE, 2020, pp. 44–51.

[19] H. Mao, T. Nie, H. Sun, D. Shen, and G. Yu, “A survey on cross-chain technology: Challenges, development, and prospect,” IEEE Access, 2022.

[20] M. Rodler, W. Li, G. O. Karame, and L. Davi, “Sereum: Protecting existing smart contracts against re-entrancy attacks,” arXiv preprint arXiv:1812.05934, 2018.

[21] P. Praitheeshan, L. Pan, J. Yu, J. Liu, and R. Doss, “Security analysis methods on ethereum smart contract vulnerabilities: a survey,” arXiv preprint arXiv:1908.08605, 2019.

[22] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, and H.-N. Lee, “Systematic review of security vulnerabilities in ethereum blockchain smart contract,” IEEE Access, vol. 10, pp. 6605–6621, 2022.

[23] J.-W. Liao, T.-T. Tsai, C.-K. He, and C.-W. Tien, “Soliaudit: Smart contract vulnerability assessment based on machine learning and fuzz testing,” in 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS). IEEE, 2019, pp. 458–465.

[24] S. Badillo, B. Banfai, F. Birzele, I. I. Davydov, L. Hutchinson, T. Kam-Thong, J. Siebourg-Polster, B. Steiert, and J. D. Zhang, “An introduction to machine learning,” Clinical pharmacology & therapeutics, vol. 107, no. 4,pp. 871–885, 2020.

[25] M. Krichen, “Strengthening the security of smart contracts through the power of artificial intelligence,” Computers, vol. 12, no. 5, p. 107, 2023.

[26] Y. Xu, G. Hu, L. You, and C. Cao, “A novel machine learning-based analysis model for smart contract vulnerability,” Security and Communication Networks, vol. 2021, pp. 1–12, 2021.

[27] T. Haugum, B. Hoff, M. Alsadi, and J. Li, “Security and privacy challenges in blockchain interoperability-a multivocal literature review,” in Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering, 2022, pp. 347–356.

[28] W. Deng, H. Wei, T. Huang, C. Cao, Y. Peng, and X. Hu, “Smart contract vulnerability detection based on deep learning and multimodal decision fusion,” Sensors, vol. 23, no. 16, p. 7246, 2023.

[29] D. He, Z. Deng, Y. Zhang, S. Chan, Y. Cheng, and N. Guizani, “Smart contract vulnerability analysis and security audit,” IEEE Network, vol. 34, no. 5, pp. 276–282, 2020.

[30] J. Huang, K. Zhou, A. Xiong, and D. Li, “Smart contract vulnerability detection model based on multi-task learning,” Sensors, vol. 22, no. 5, p. 1829, 2022.

[31] B. Jiang, Y. Liu, and W. K. Chan, “Contractfuzzer: Fuzzing smart contracts for vulnerability detection,” in Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, 2018, pp. 259–269.

[32] R. M. Parizi, A. Dehghantanha, K.-K. R. Choo, and A. Singh, “Empirical vulnerability analysis of automated smart contracts security testing on blockchains,” arXiv preprint arXiv:1809.02702, 2018.

ĐẠI HỌC QUỐC GIA TP. HCM
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc Lập - Tự do - Hạnh Phúc

TP.HCM, ngày tháng năm 2024

THÔNG TIN KẾT QUẢ NGHIÊN CỨU

1. Thông tin chung:

- Tên đề tài: PHÁT HIỆN LỖ HỔNG TRONG HỢP ĐỒNG THÔNG MINH TRÊN MẠNG LIÊN CHUỖI KHỐI BẰNG PHƯƠNG PHÁP HỌC MÁY VÀ HỌC SÂU
- Mã số:
- Chủ nhiệm: VÕ ANH KIỆT – 20520605
- Thành viên tham gia:
- Cơ quan chủ trì: Trường Đại học Công nghệ Thông tin.
- Thời gian thực hiện: 6 tháng

2. Mục tiêu:

Triển khai được mô hình cầu nối dựa trên phương pháp SideChain và tiến hành được quá trình chuyển đổi dữ liệu qua các mạng chuỗi khối.

Xây dựng tập dữ liệu CrossChainSentinel bao gồm các mẫu lành tính và các mẫu độc hại được gán nhãn thủ công, đồng thời tiến hành xử lý dữ liệu.

Ứng dụng các mô hình học máy và học sâu trong việc phát hiện các mẫu độc hại một cách tự động, đánh giá và nhận xét các mô hình.

3. Tính mới và sáng tạo:

Ở nghiên cứu này, nhóm nghiên cứu sinh viên đã tiến hành triển khai việc chuyển đổi dữ liệu thông qua cầu nối Sidechain, đồng thời xây dựng và cung cấp được tập dữ liệu dành riêng cho các hợp đồng thông mô hình cầu nối Sidechain bao gồm 300 mẫu (bao gồm 158 mẫu an toàn và 142 mẫu có lỗ hổng). Đồng thời, ở nghiên cứu này nhóm cũng tiến hành việc xử lý dữ liệu và tiến hành thực hiện quá trình huấn luyện các mô hình học máy và học sâu để có thể tự động hóa việc phát hiện lỗ hỏng trên hợp đồng thông minh một cách tự động.

4. Tóm tắt kết quả nghiên cứu:

Hoàn tất quá trình triển khai được mô hình cầu nối dựa trên phương pháp SideChain và tiến hành được quá trình chuyển đổi dữ liệu qua các mạng chuỗi khối

Thành công trong việc tập dữ liệu CrossChainSentinel bao gồm các mẫu lành tính và các mẫu độc hại được gán nhãn thủ công, đồng thời tiến hành xử lý dữ liệu.

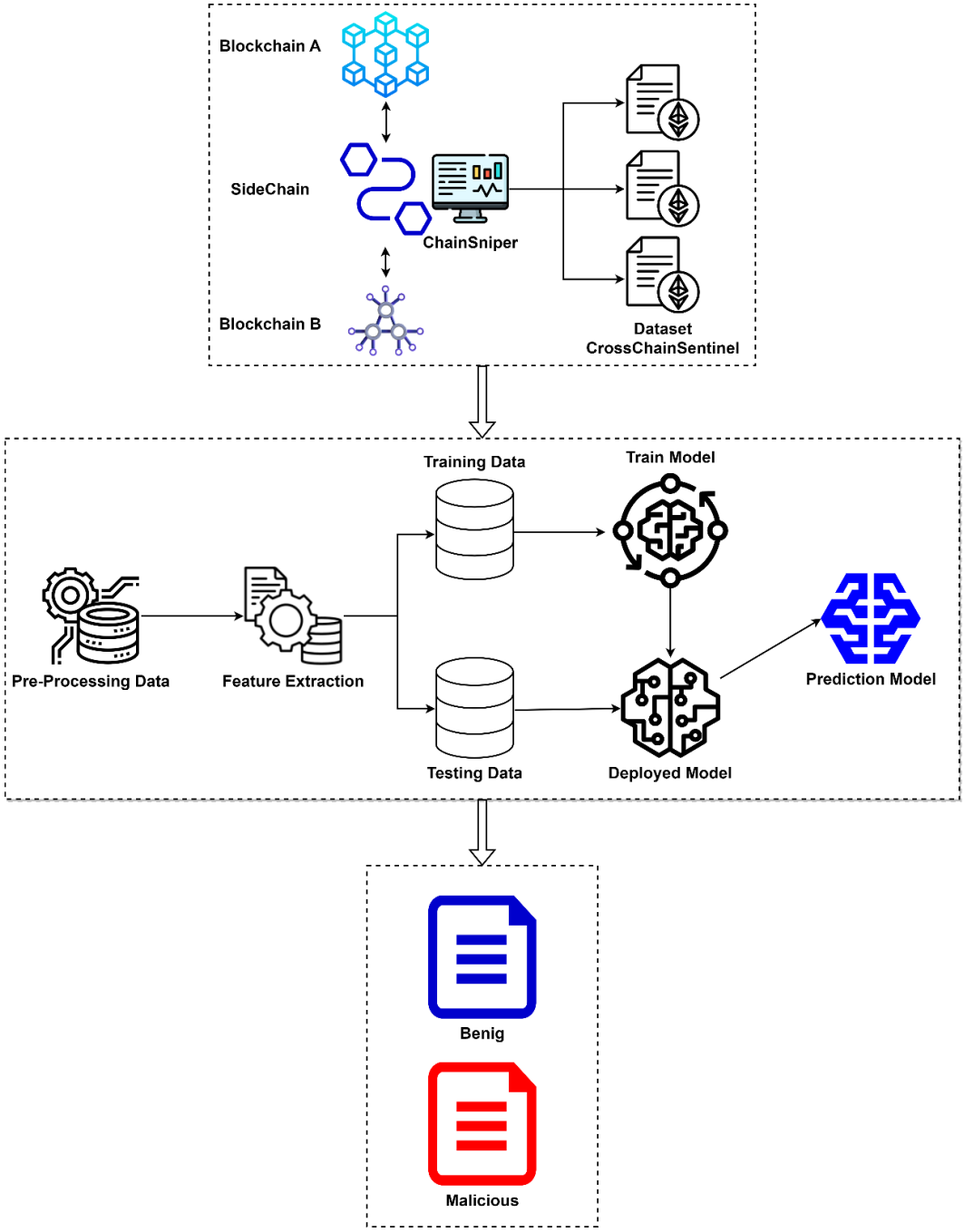
Đánh giá và nhận xét các mô hình học máy và học sâu trong việc phát hiện các mẫu độc hại một cách tự động, các mô hình.

5. Tên sản phẩm: Phát hiện lỗ hổng trong hợp đồng thông minh trên mạng liên chuỗi khối bằng phương pháp học máy và học sâu

6. Hiệu quả, phương thức chuyển giao kết quả nghiên cứu và khả năng áp dụng:

Thực nghiệm cho thấy hệ thống có khả năng phát hiện các lỗ hổng trong hợp đồng thông minh liên chuỗi với độ chính xác cao, lên tới 96,7% khi sử dụng mô hình Roberta. Các mô hình học máy và học sâu trong ChainSniper như Random Forest, XGBoost, CNN, LSTM cũng đạt hiệu suất tốt, với độ chính xác trên 70%, cao nhất với mô hình RoBERTa với 96.67%. Thời gian xử lý mẫu hợp đồng thông minh cũng khá nhanh, từ 3.3 – 4.0 giây tùy thuộc vào mô hình, phù hợp với yêu cầu đánh giá an ninh tự động. Hệ thống có thể phát hiện các lỗ hổng phổ biến như tấn công reentrancy, tràn/thiếu số nguyên và rút tiền Ether không được bảo vệ. Kết quả nghiên cứu có thể được chuyển giao dưới dạng một công cụ phần mềm mã nguồn mở và có thể được áp dụng rộng rãi trong việc kiểm tra an ninh hợp đồng thông minh liên chuỗi trước khi triển khai vào các nền tảng blockchain thực tế.

7. Hình ảnh, sơ đồ minh họa chính



Cơ quan Chủ trì
(ký, họ và tên, đóng dấu)

Chủ nhiệm đề tài
(ký, họ và tên)