

BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH

Thành viên :

- Trần Nguyễn Tiến Thành - 22521364
- Hồ Vĩnh Nhật - 22521013

Automatic black-box testing utilizing
reinforcement learning for Android application



Phòng thí nghiệm An toàn thông tin (InSecLab)
Trường ĐH Công nghệ Thông Tin, ĐHQG Tp. HCM



Nội dung báo cáo

- **Phần I: Tổng quan đề tài**
- **Phần II: Function Architecture**
- **Phần III: App Architecture**
- **Phần IV: Demo và kết quả**





I - Tổng quan đề tài

Painpoints:

1. Quy trình kiểm thử còn **thủ công**, đòi hỏi tiêu **tốn nhiều thời gian (ngày làm)**
2. Quá trình kiểm thử hầu hết là các **tác vụ cơ bản, lặp lại tạo gánh nặng không cần thiết cho tester**
3. Cân **nhiều chi phí** cho việc kiểm thử (nhân công, thời gian, công cụ)
4. Yêu cầu nhân lực có một **trình độ, kinh nghiệm nhất định** để thực hiện kiểm thử
5. Hiện tại quy trình kiểm thử tự động hầu hết là **ngẫu nhiên, dựa theo một bộ rule** có sẵn dẫn đến **ít linh hoạt** cho nhiều môi trường khác nhau.





I - Tổng quan đề tài

Painpoints:

1. Thủ công, tốn nhiều thời gian (ngày làm) → Sử dụng Reinforcement Learning để thực hiện kiểm thử tự động trong vài giờ
2. Tác vụ cơ bản, lặp lại tạo gánh nặng không cần thiết cho tester → Thực hiện kiểm thử tự động các lỗi, tính năng cơ bản, tester tập trung vào các tác vụ yêu cầu chuyên sâu, kỹ năng
3. Nhiều chi phí -> Tốn ít chi phí nhân công
4. Trình độ, kinh nghiệm -> Tự cải thiện theo thời gian
5. Ngẫu nhiên, dựa theo một bộ rule có sẵn dẫn đến ít linh hoạt → Sử dụng Reinforcement Learning có thể linh hoạt thay đổi, học từ môi trường



I - Tổng quan đề tài

Mục tiêu:

Đề xuất một chương trình kiểm thử hộp đen (black-box) ứng dụng Android tự động sử dụng học tăng cường (Reinforcement Learning)

- Sử dụng **Double Dueling DQN** để giảm giao động trong các lựa chọn (Q over-estimate) và tốt trong nhận diện các state gần giống nhau (Thay đổi nhỏ trong UI)
- Sử dụng kết hợp với **input hình ảnh** để tăng khả năng nhận diện UI của mô hình
- Sử dụng **Graph Isomorphism Network with Edge (GINE)** để trích xuất thông tin từ đồ thị trạng thái (state graph)
- Kết hợp với small LLM (**microsoft/Phi-4-mini-instruct**) để thực hiện tạo các payload cho trường input (số, chữ)





Nội dung báo cáo

- Phần I: Tổng quan đề tài
- **Phần II: Function Architecture**
- Phần III: App Architecture
- Phần IV: Demo và kết quả



II - Function Architecture



Function	Input	Output	Mô tả
GUI embedder	Ảnh chụp màn hình XML của UI Các chỉ số của state trước đó	Các action có thể thực hiện State graph của UI	Có nhiệm vụ phân tích XML của UI để tạo ra danh sách của các action có thể thực hiện và State graph của UI
State embedder	State graph của UI	Đặc trưng của state hiện tại	Trích xuất các đặc trưng của State graph sử dụng GINEConv để tận dụng edge làm đầu vào cho mô hình
DQN Agent	Đặc trưng state và Các action có thể thực hiện	Lựa chọn một action duy nhất để thực thi tương tác	Lựa chọn action tối ưu để tương tác với môi trường
Environment	Action	State	Nhận action và trả về state mới tương ứng
Reward analyzer	State, Action, State mới	Reward	Tính toán phần thưởng mà mô hình nhận được sau mỗi action
Prioritized Experience Replay	State, Action, State mới, Reward	Sample các hành động quá khứ	Dùng để huấn luyện lại mô hình DQN Agent nhằm giúp mô hình RL "học" và "nhớ" các action quan trọng, lặp lại trong quá khứ.
ACVTools	apk	Code coverage	Tính độ bao phủ câu lệnh (instruction coverage) của mô hình kiểm thử



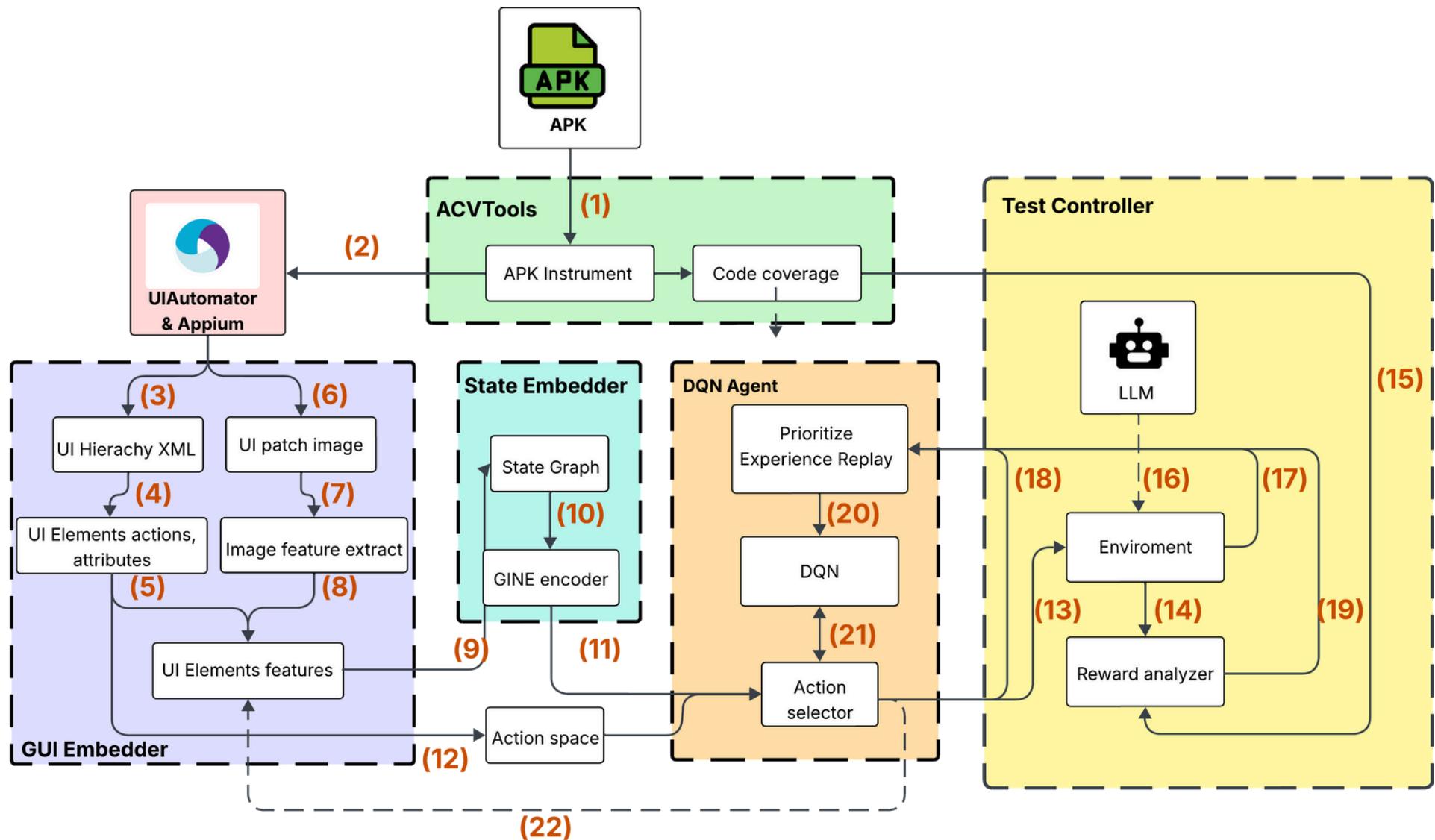


Nội dung báo cáo

- Phần I: Tổng quan đề tài
- Phần II: Function Architecture
- **Phần III: App Architecture**
- Phần IV: Demo và kết quả



III - App Architecture



III - App Architecture



1. Instrument APK bằng ACVTools để tính code coverage
2. UIAutomator và appium sẽ thực hiện trích xuất ảnh chụp màn hình và dữ liệu XML từ quá trình chạy kiểm thử
3. Dữ liệu Hierarchy thu được từ quá trình trích xuất dữ liệu từ XML
4. Từ Hierarchy XML trích xuất ra các action có thể thực hiện trên các phần tử giao diện người dùng và state
6. UI Patch Image là ảnh chụp màn hình từ quá trình chạy và được xử lý
7. Từ dữ liệu đã được xử lý của UI Patch Image trích xuất ra các đặc trưng hình ảnh như nhận diện nút, màu sắc, hoặc bối cảnh, để hỗ trợ phân tích trạng thái UI hoặc nhận diện các yếu tố không phụ thuộc vào XML.



III - App Architecture



- 5,8. Dữ liệu thu được thông qua Hierarchy và Patch image sẽ được tổng hợp lại thành các phần tử chứa dữ liệu về state và action
9. Dữ liệu về state và action sẽ được sử dụng để xây dựng nên state graph với các node là các element feature bao gồm cả action và state và mối quan hệ giữa các node
10. Sử dụng Gineconv để trích xuất các đặc trưng từ state graph và encode nó thành một tensor để hỗ trợ quá trình chọn lựa action
- 11,12. Dữ liệu bao gồm tensor từ Gineconv và một chuỗi các actions lấy được từ quá trình GUI embedded sẽ được chuyển tiếp đến DQN để thực hiện lựa chọn
13. Sau quá trình tối ưu DQN sẽ trả về một action và được đem đến môi trường để thực hiện.



III - App Architecture



14. Sau khi chạy thử action reward analyzer sẽ tính điểm reward cho action
15. Đồng thời code coverage được tính bằng acvtools cũng sẽ đóng góp trong quá trình tính điểm
16. LLM được sử dụng để generate và thực hiện các input text trong quá trình kiểm thử nhằm phát hiện các lỗi từ việc nhập liệu
- 17,18,19. Sau khi chạy dữ liệu về action, new state và reward trả về sẽ được đưa ngược lại về DQN để tối ưu hóa quá trình chọn lựa
20. Các dữ liệu được đưa về sẽ được lưu lại dưới dạng tuple và tính toán giá trị ưu tiên của các action nhằm giúp mô hình RL học và nhớ các action quan trọng
21. Thực hiện chọn lựa action dựa trên những cập nhập để tối ưu các action tiếp theo
22. Action được chọn sẽ được làm cạnh trong state graph trở tới state mới sau khi thực hiện action





Nội dung báo cáo

- **Phần I: Tổng quan đề tài**
- **Phần II: Function Architecture**
- **Phần III: App Architecture**
- **Phần IV: Demo và kết quả**





Môi trường:

- 16 Core CPU 3.8Hz
- GPU 4060 8GB và GPU 3050 4GB
- 16GB RAM
- Windows 11 và Ubuntu 24.04LTS

So sánh:

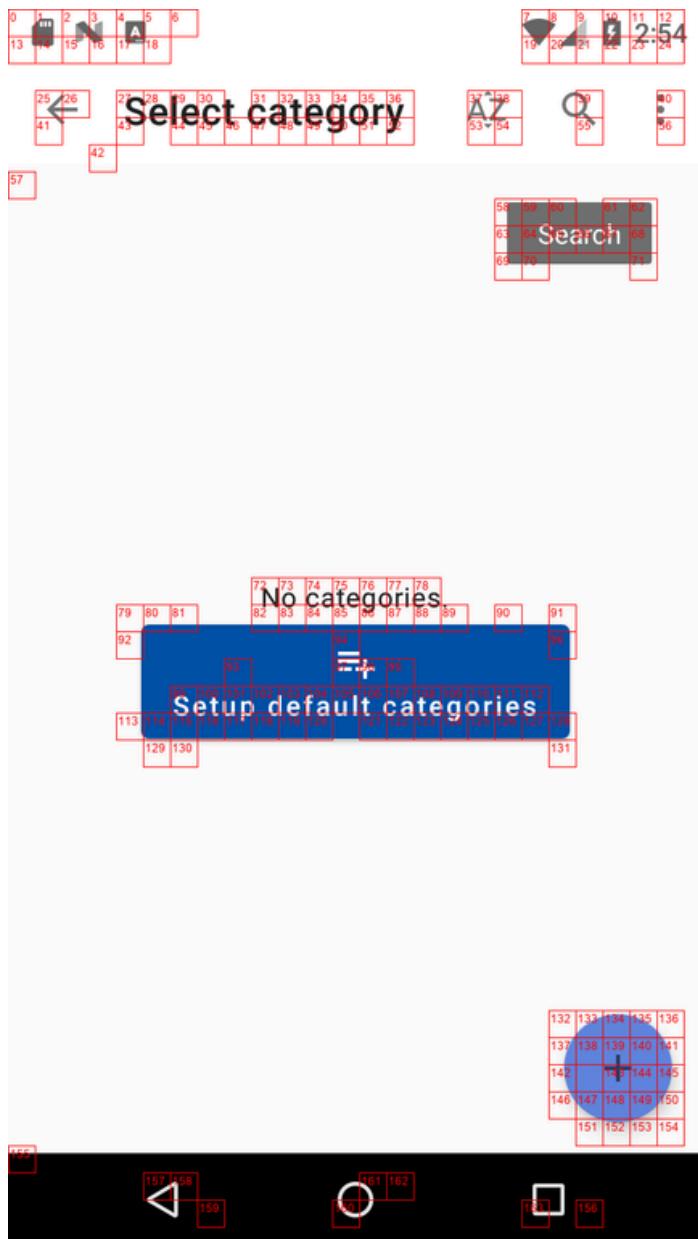
- DQT (exe)
- Monkey

So sánh dựa trên: code coverage (ACVTools) và số lỗi tìm được. Mỗi thí nghiệm thực hiện trong 2 giờ (120 phút)



IV - Demo

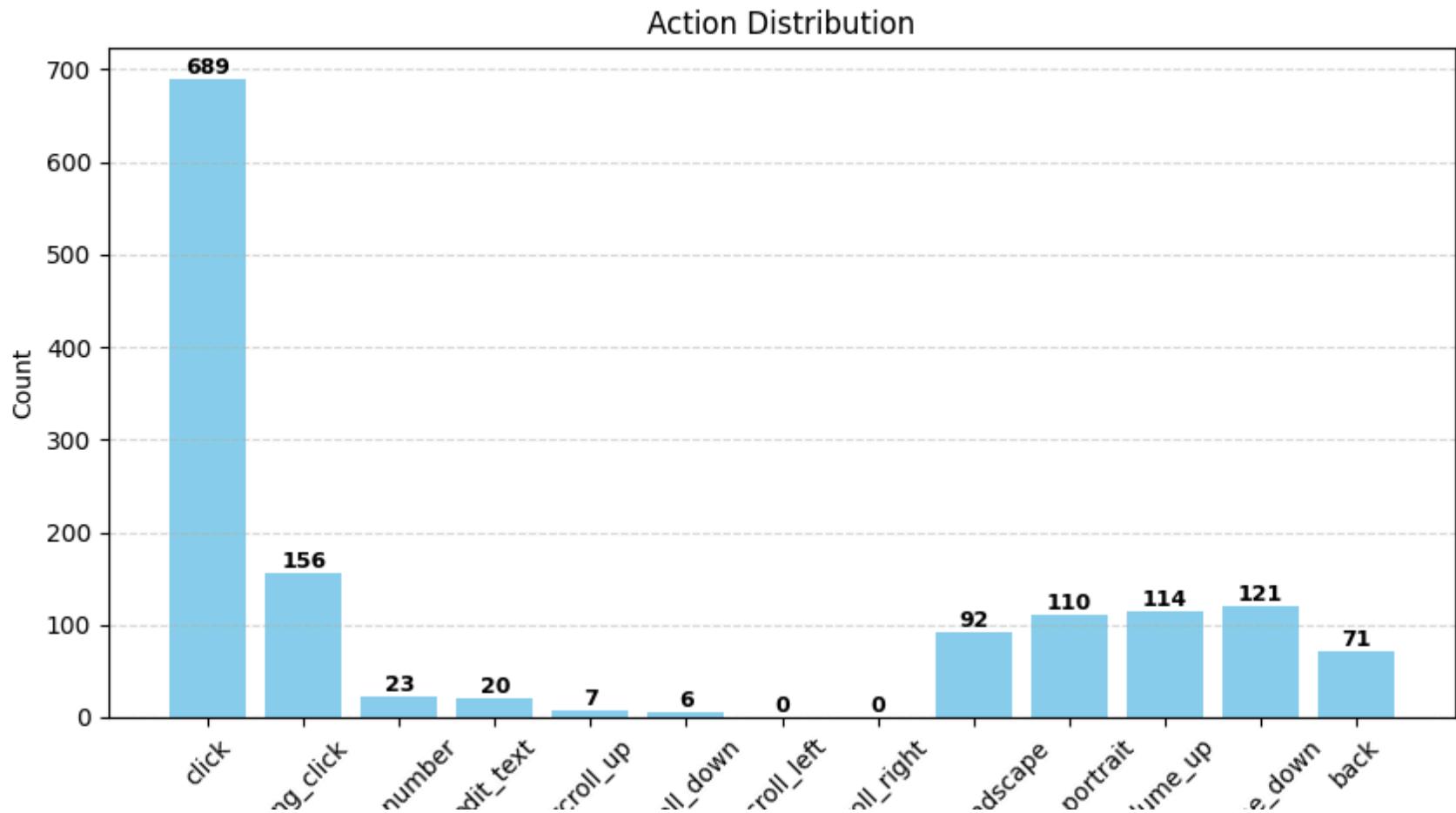
Hình ảnh



- Image patch

IV - Demo

Hình ảnh



Phân bổ 13 action đã thực hiện



IV - Demo

 org.totschnig.myexpenses.debug >  org.totschnig.myexpenses.debug

granularity level: instruction

Element	Ratio	Cov.	Missed	Lines	Missed	Methods	Missed	Classes
1		7.98038%	60790	66062	7873	8280	921	1020
2		3.15185%	33247	34329	1286	1371	103	122
3		0.60706%	16209	16308	978	984	110	112
4		21.40733%	22729	28920	851	1153	56	100
5		17.56583%	10925	13253	1274	1300	614	627
6		27.57258%	52591	72612	1386	2562	61	183
7		17.93764%	34371	41884	978	1261	140	208
8		13.07876%	22922	26371	796	1017	86	121
9		25.45820%	10005	13422	532	821	69	119
10		0.00000%	59	59	4	4	1	1
11		3.94201%	3777	3932	128	145	16	20
12		3.69318%	4746	4928	160	172	17	19
13		14.07381%	24330	28315	1392	1656	171	231
14		11.39509%	127786	144220	5014	6072	559	727
15		11.54156%	25277	28575	1015	1216	86	130
16		5.94521%	124996	132897	4970	5559	560	670
17		39.43497%	32114	53024	1319	1483	59	87
18	 	5.04728%	1036125	1091201	50710	54376	5321	6226
19	 	13.87821%	979141	1136926	44352	53713	5218	6777
20	 	2.97636%	1022959	1054340	51573	53529	5503	5856
21	 	3.55185%	4105787	4256989	51154	59125	12928	14313
Total		497681 of 8248567	6.03354%	7750886	8248567	227745	255799	32599
								37669

Coverage của ACVTool



IV - Demo

Demo:



IV - Kết quả

Coverage

Application	Version	Android SDK	DQT		Monkey		Our	
			Activity Cov	Instr Cov	Activity Cov	Instr Cov	Activity Cov	Instr Cov
AntennaPod	3.8.0	Android SDK 21	0.5 (6/12)	13.48%			28.92%	100% (6/6)
Slideshow Wallpaper	1.2.2	Android SDK 18	1 (3/3)	4.61%			5.62%	100% (6/6)
NewPipe APK	0.27.7	Android SDK 21	0.25 (3/12)	19.01%			20.52%	100% (5/5)
Seal for Android	1.13.1	Android SDK 21	0.5 (2/4)	14.03%			28.60%	33.33%(1/3)
MyExpenses-r554-debug.apk(DQT app)	N/A	N/A	0.5909(21/44)	2.06%			7.97%	84.61538% (11/13)
Spotify for Android	3.6.4(latest)	Android SDK 21	1 (2/2)	14.03%			—	100%(3/3)
								17.97%

Lỗi tìm được

Application	Version	Android SDK	DQT	Monkey	Our
AntennaPod	3.8.0	Android SDK 21	0	0	5
Slideshow Wallpaper	1.2.2	Android SDK 18	0	0	1
NewPipe APK	0.27.7	Android SDK 21	0	0	2
Seal for Android	1.13.1	Android SDK 21	0	0	1
MyExpenses-r554-debug.apk(DQT app)	N/A	N/A	2	0	0
Spotify for Android	3.6.4(latest)	Android SDK 21	0	0	2

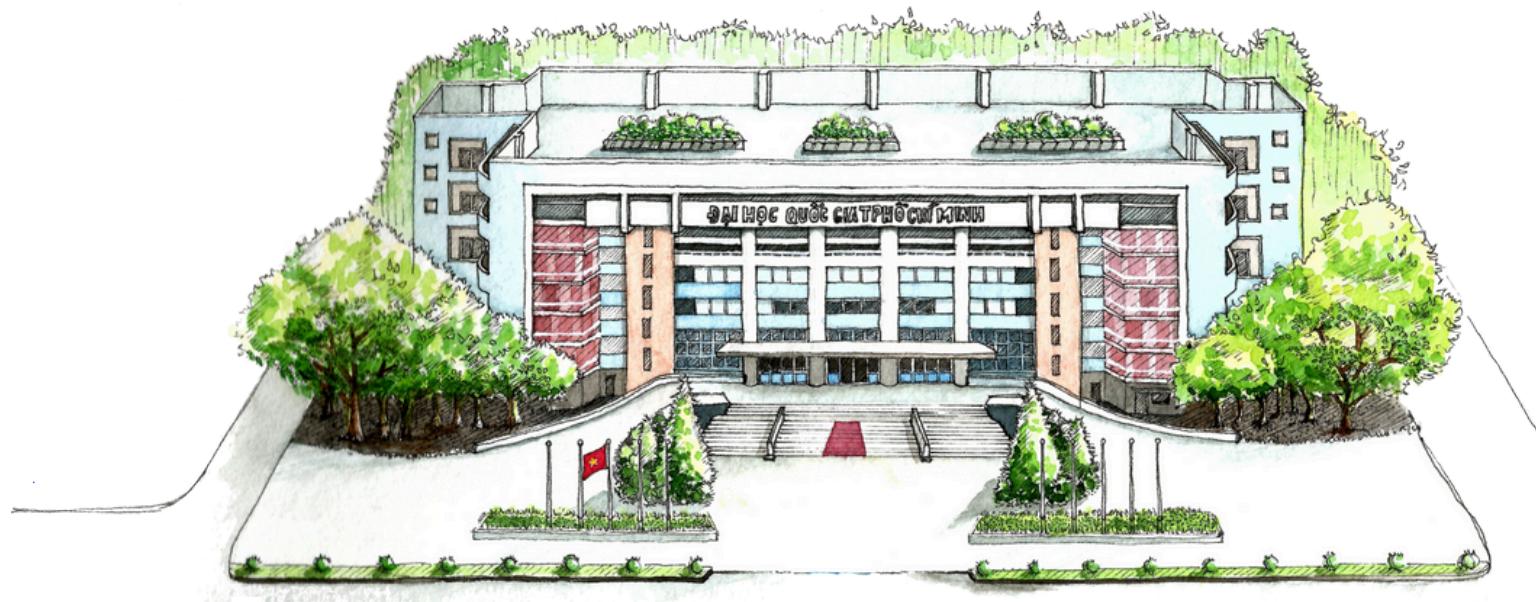




V - Khó khăn và hạn chế

- Mô hình còn nhiều biến động và ngẫu nhiên
- Mô hình chưa nắm được môi trường dẫn đến lặp lại, exploit, bỏ qua một số chỗ
- Mô hình chưa nắm được hoàn toàn nội dung (context) của UI - điền form
- Chưa khám phá được một số chức năng yêu cầu upload, tải file lên app





Xin cảm ơn.



Nhóm nghiên cứu InSecLab **Phòng Thí nghiệm An toàn thông tin**

Email: inseclab@uit.edu.vn

Website: <https://inseclab.uit.edu.vn/>

Fanpage: <https://www.facebook.com/inseclab>



Tài liệu liên quan

- Tên bài báo khoa học theo định dạng trích dẫn chuẩn IEEE (có tên tác giả, nơi công bố, năm công bố).

