Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

HinDroid：An Intelligent Android Malware Detection System Based on Structured Heterogeneous Information Network

张寒青　硕士

2018年11月11日

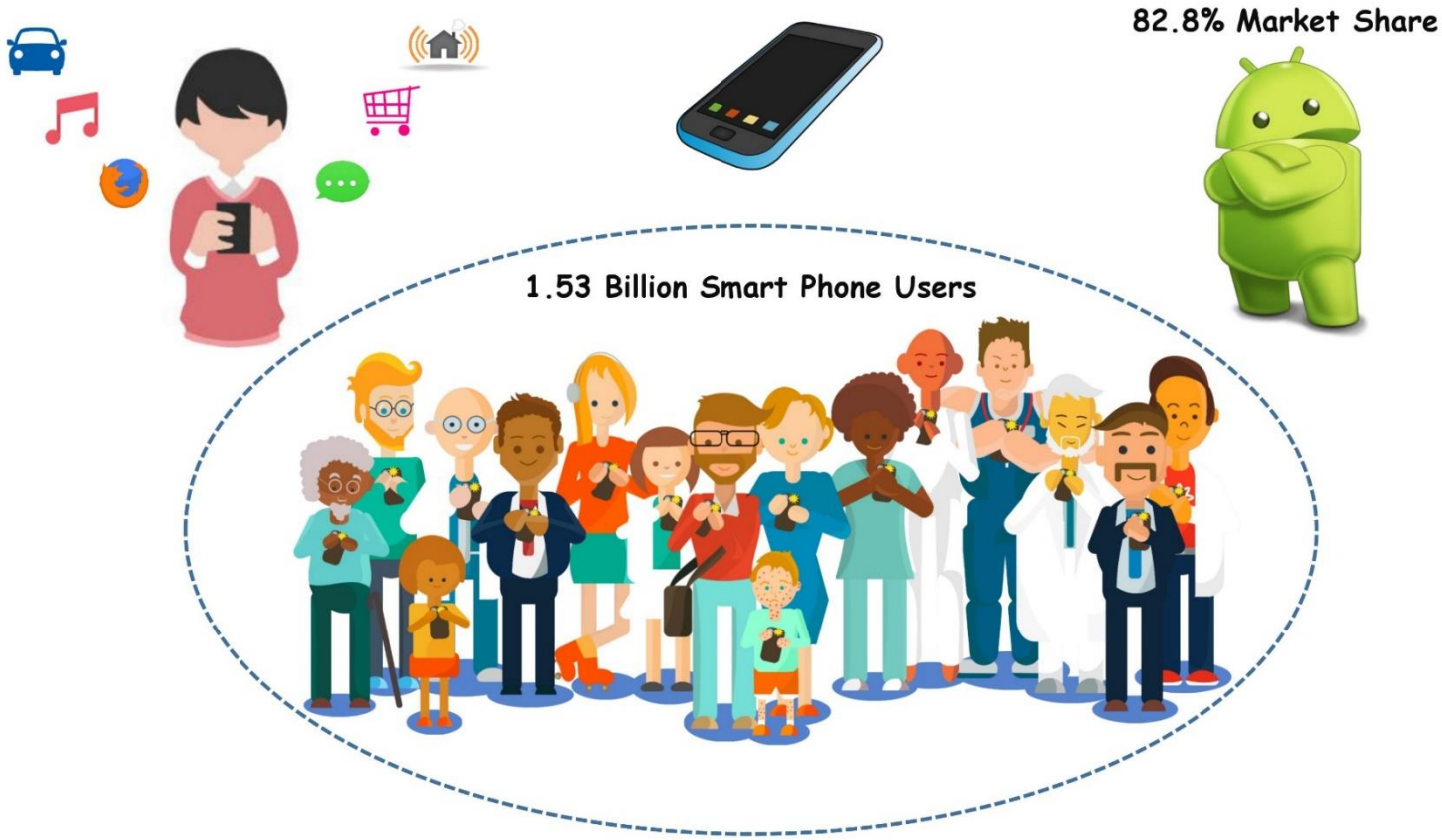
内容提要

- 背景简介
- 基本概念
- 框架原理
- 实验分析
- 优劣分析
- 应用总结
- 参考文献

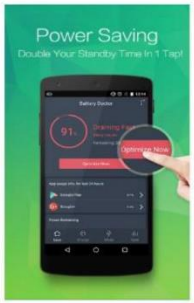
背景简介

- 预期收获
  - 1.了解一种最新恶意软件智能检测方法
  - 2.理解异构信息网络、SVM、多核学习等基础知识
  - 3. 学习信息网络在数据挖掘问题中的运用

背景简介



背景简介


Power Saving
Double Your Standby Time In 1 Tap!


BRING FRIENDS
TO BATTLE

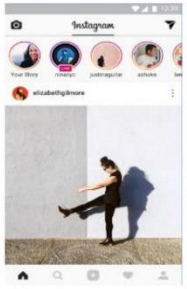

Mario runs automatically
Tap to jump


Express yourself with free stickers


COLOR SWITCH
GAME MODES

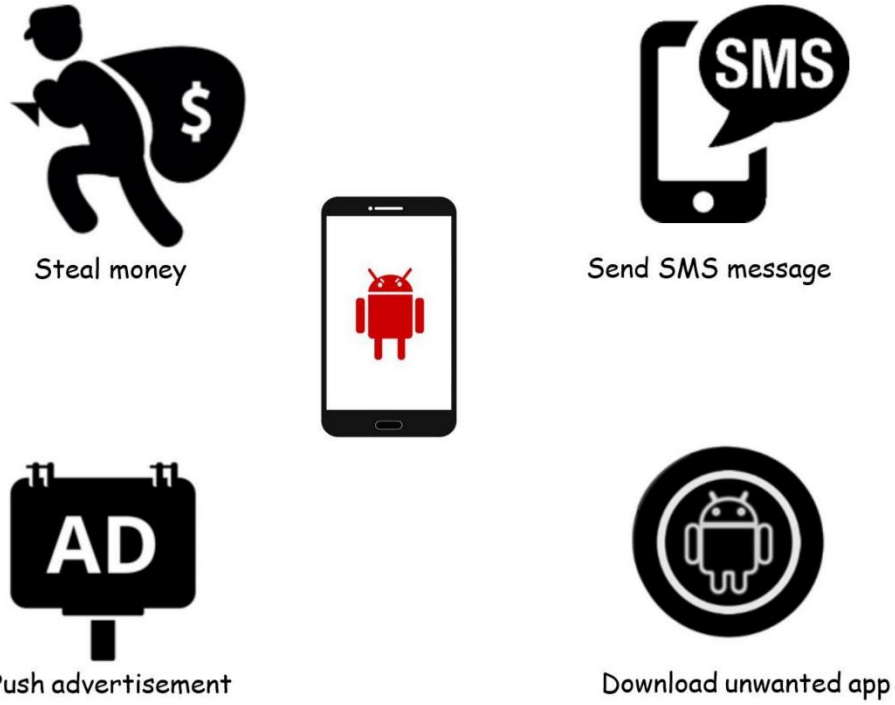

Simple. Personal.
Real time messaging.


ANGRY BIRDS
#1 APP OF ALL TIME




Get the very latest on the things you care about most.

背景简介

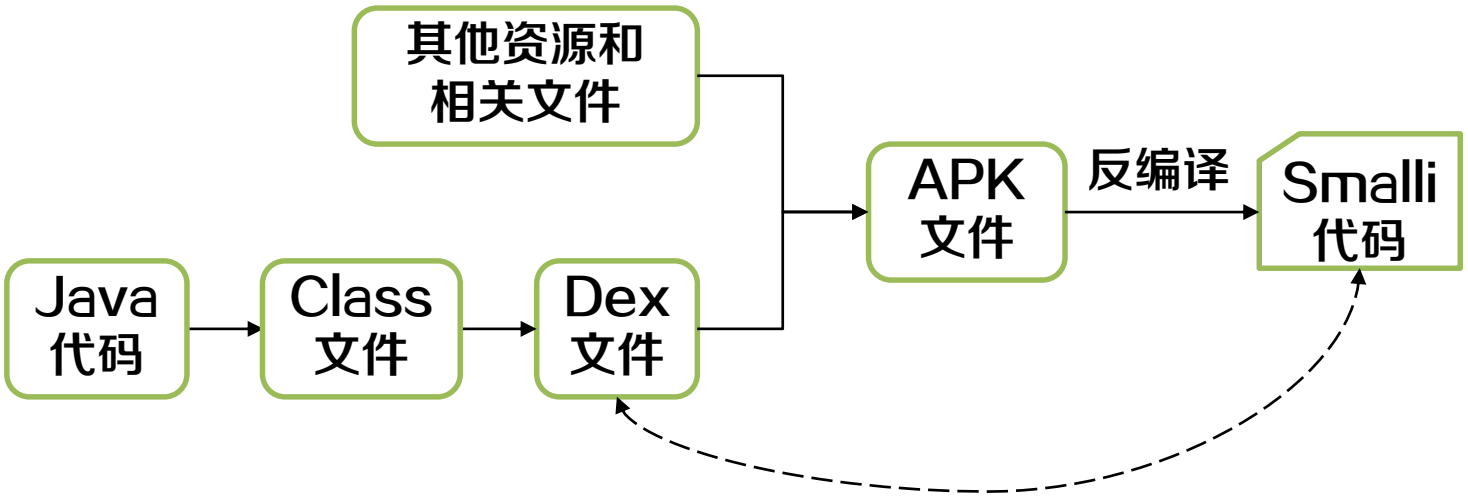

Steal money
SMS
Send SMS message
AD
Push advertisement
Download unwanted app

背景简介

Abdulhayoglu, Melih, et al. "HinDroid: An Intelligent Android Malware Detection System Based on Structured Heterogeneous Information Network." *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* ACM, 2017:1507-1515.

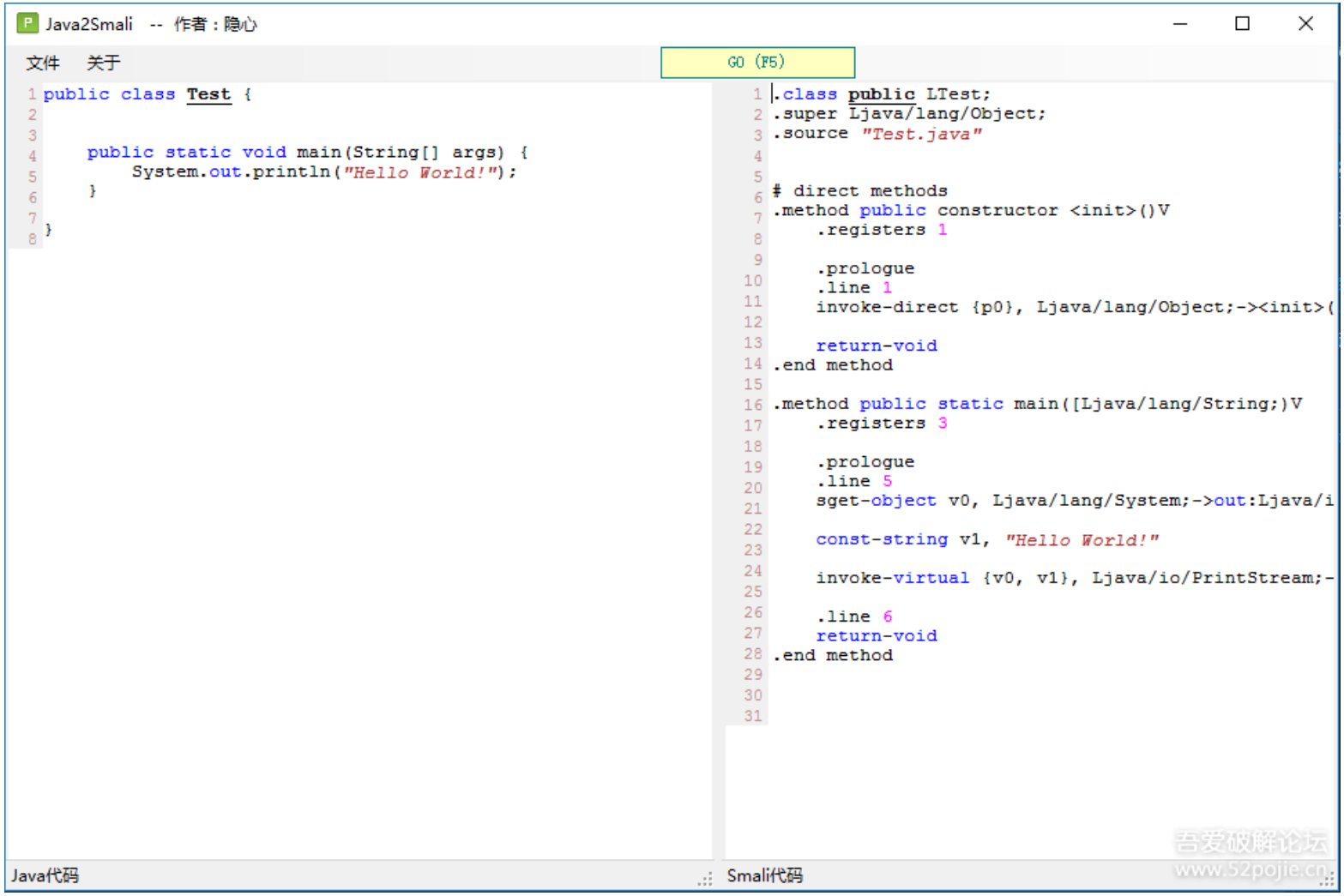
基本概念

- Smali语言
  - Dalvik虚拟机的寄存器语言
  - Dex文件反编译后得到Smalli代码



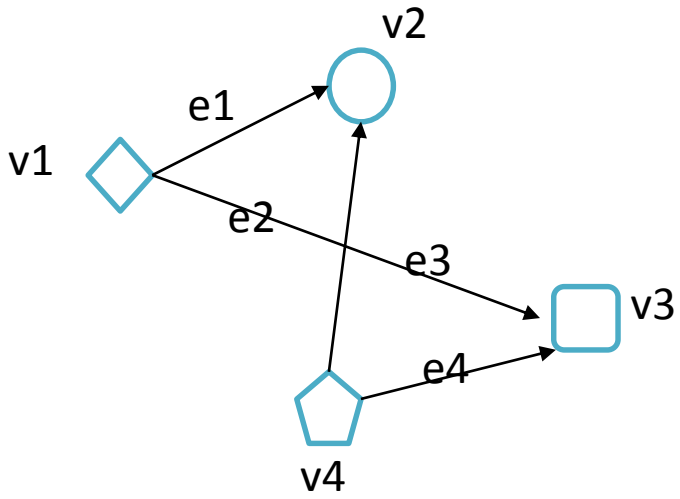
基本概念

- Smalli代码实例



基本概念

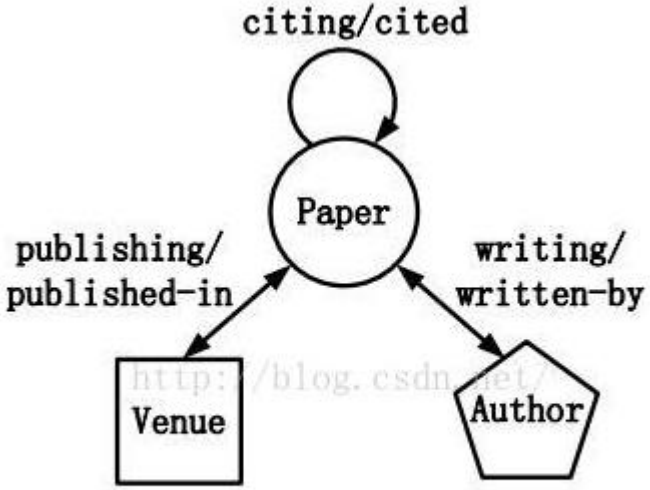
- 信息网络
  - 信息网络可以用一个有向图G = (V, E) 来表示



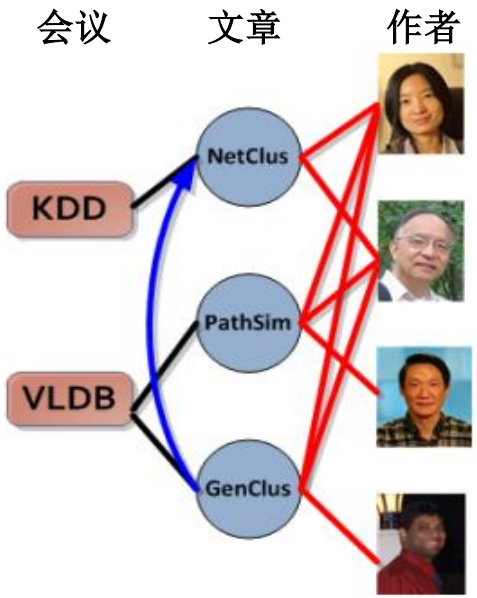
V= {v1,v1,v3,v4}
E= {e1,e1,e3,e4}

基本概念

- 异构信息网络(HIN)
  - 如果|A| > 1 或者 |R| > 1，则该信息网络为异构信息网络，或简称为异构网络，否则为同构网络。



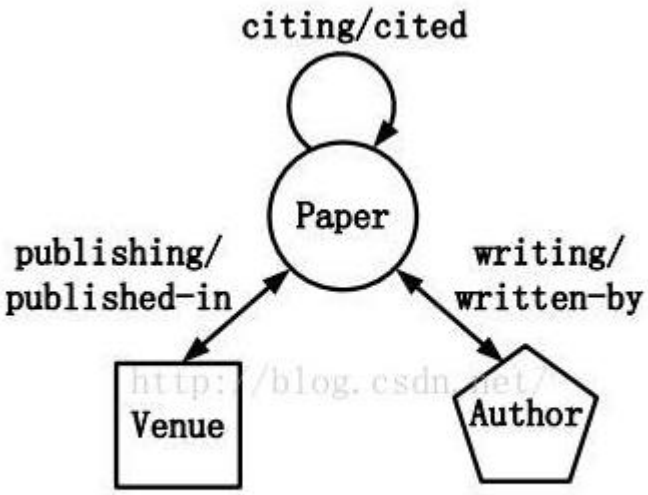
Network Schema
TG=(A,R)



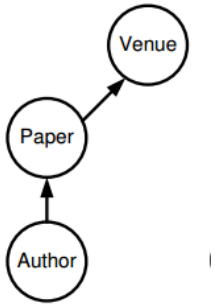
A=｛会议、文章、作者｝
R=｛publishing、published-in、writing、written-by、citing、cityed｝

基本概念

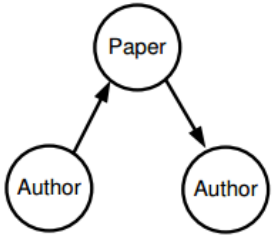
- 元路径
  - 元路径P定义在网络模式TG=(A,R)之上
  - 链接两类对象的一条路径



Network Schema
TG=(A,R)



元路径：APV



元路径：APA

基本概念

- 交换矩阵
  - 元路径中实体邻接矩阵乘积

元路径: $P = (A_1 A_2 A_3 \cdots\cdots A_l)$

交换矩阵： $M = W_{A_1A_2} W_{A_2A_3} \cdots\cdots W_{A_{l-1}A_l}$

例如：P=(AVA) → $M = W_{AV} W_{VA}$

W_{AV} =

	SIGMOD	VLDB	ICDE	KDD
Mike	2	1	0	0
Jim	50	20	0	0
Mary	2	0	1	0
Bob	2	1	0	0
Ann	0	0	1	1

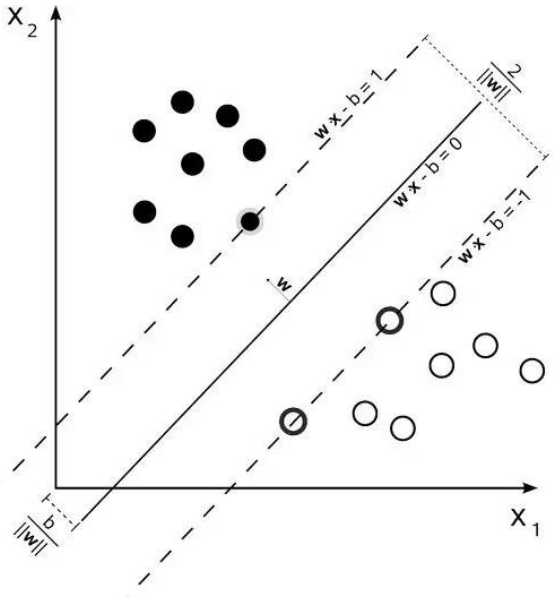
$$M = W_{AV} * W_{AV}{}^{T} = \begin{bmatrix} 5 & 120 & 4 & 5 & 0 \\ 120 & 2900 & 100 & 120 & 0 \\ 4 & 100 & 5 & 4 & 1 \\ 5 & 120 & 4 & 5 & 0 \\ 0 & 0 & 1 & 0 & 2 \end{bmatrix}$$

- SVM分类原理
  - 特征空间里寻找超平面，以最小的错分率把正负样本分开

$$D = \{(x_1, y_1), (x_2, y_2), \ldots\ldots (x_m, y_m)\}$$

$$y_i \in (-1, +1)$$

$$f(x) = w^T x + b$$

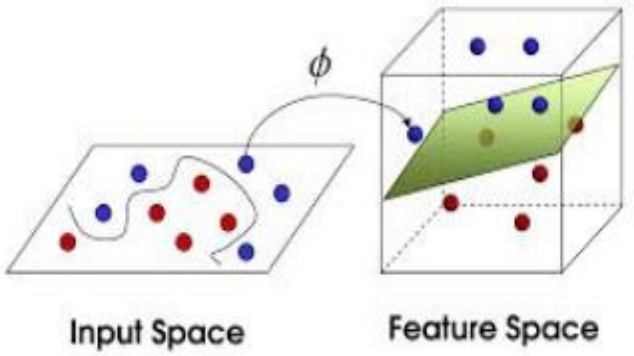


基本概念

- 核技巧
  - 将输入空间映射到高维特征空间

$$f(x) = w^T x + b$$

$$f(x) = w^T \varphi(x) + b$$



$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j \, y_i \, y_j \, \varphi(x_j)^T \varphi(x_i)$$

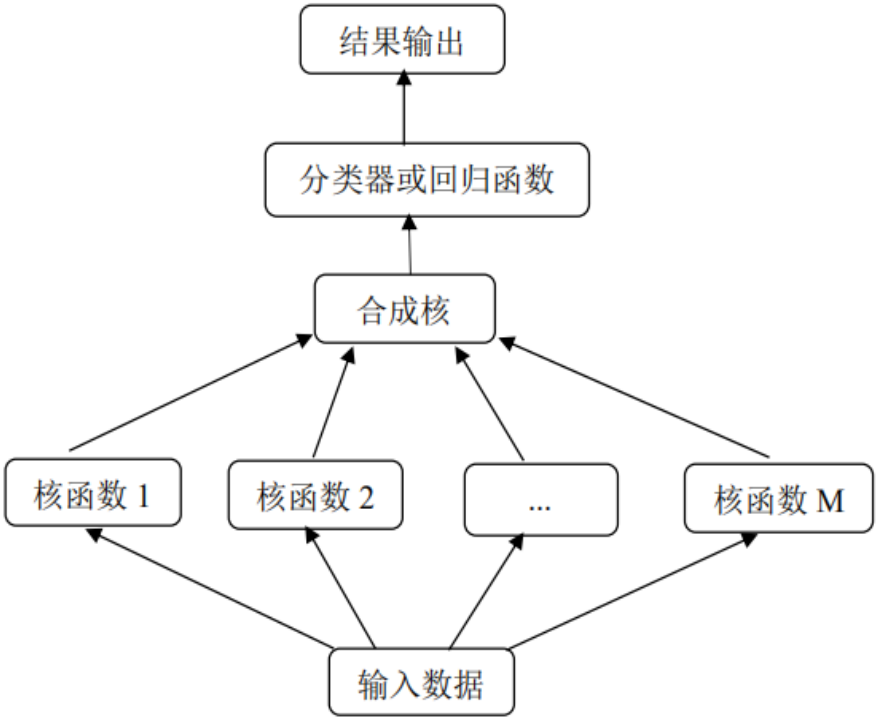
$$k(x_j, x_i) = \varphi(x_j)^T \varphi(x_i)$$

核函数!

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0$$

基本概念

- 多核学习
  - 针对不同的特征，采用不同核函数
  - 充分发挥了各个基本核的不同特征映射能力



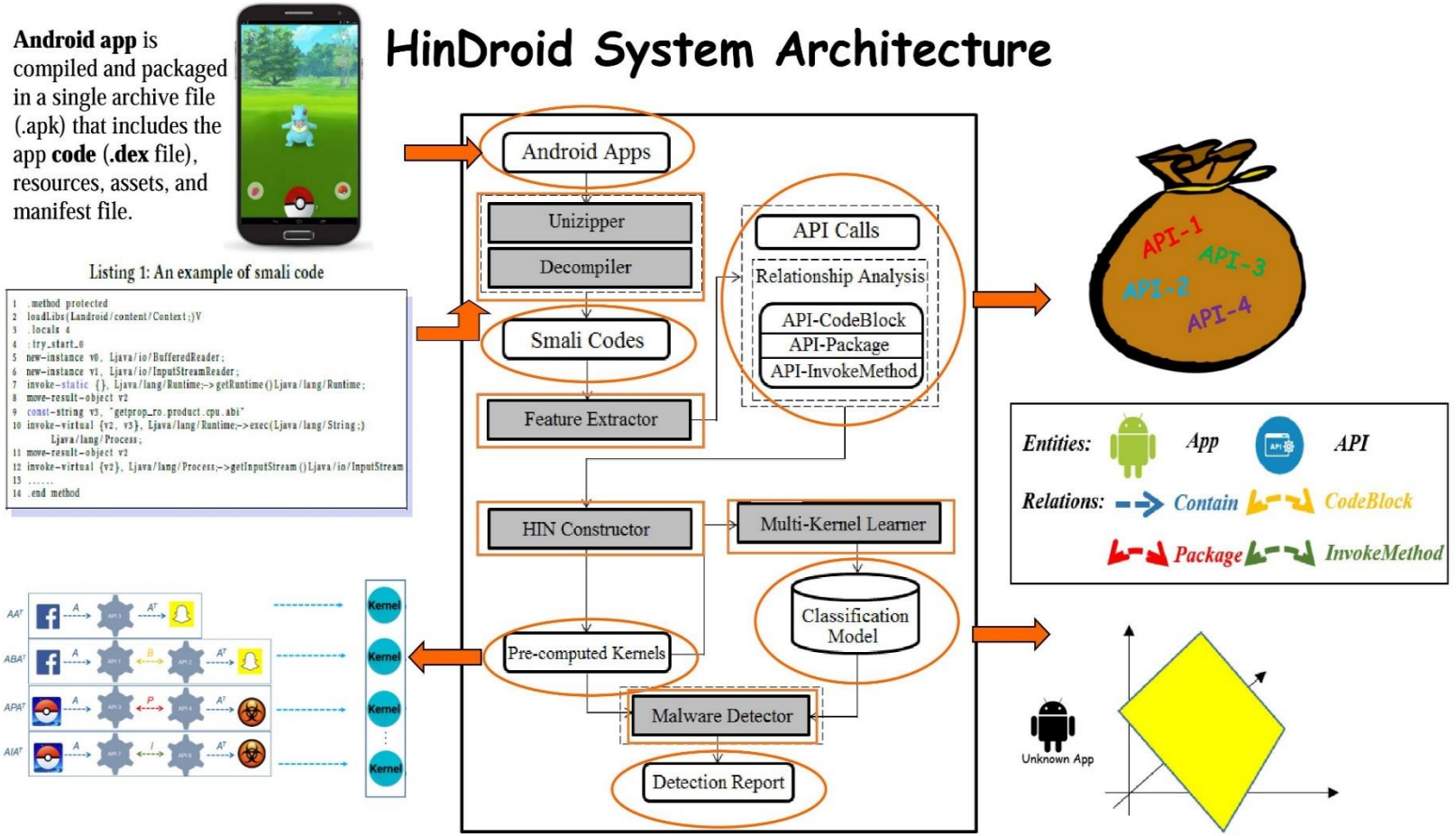
框架原理

T	对Android应用文件进行分析，判断是否为恶意软件
I	APK文件
P	1.预处理 2.特征构建 3.机器学习模型分类
O	APK文件是否为恶意

P	基于常规特征检测方式容易被代码改写等方式绕过
C	具备大量有标签APK样本
D	构建表征能力强的特征
L	2017 KDD 顶级会议

框架原理

- 整体框架图

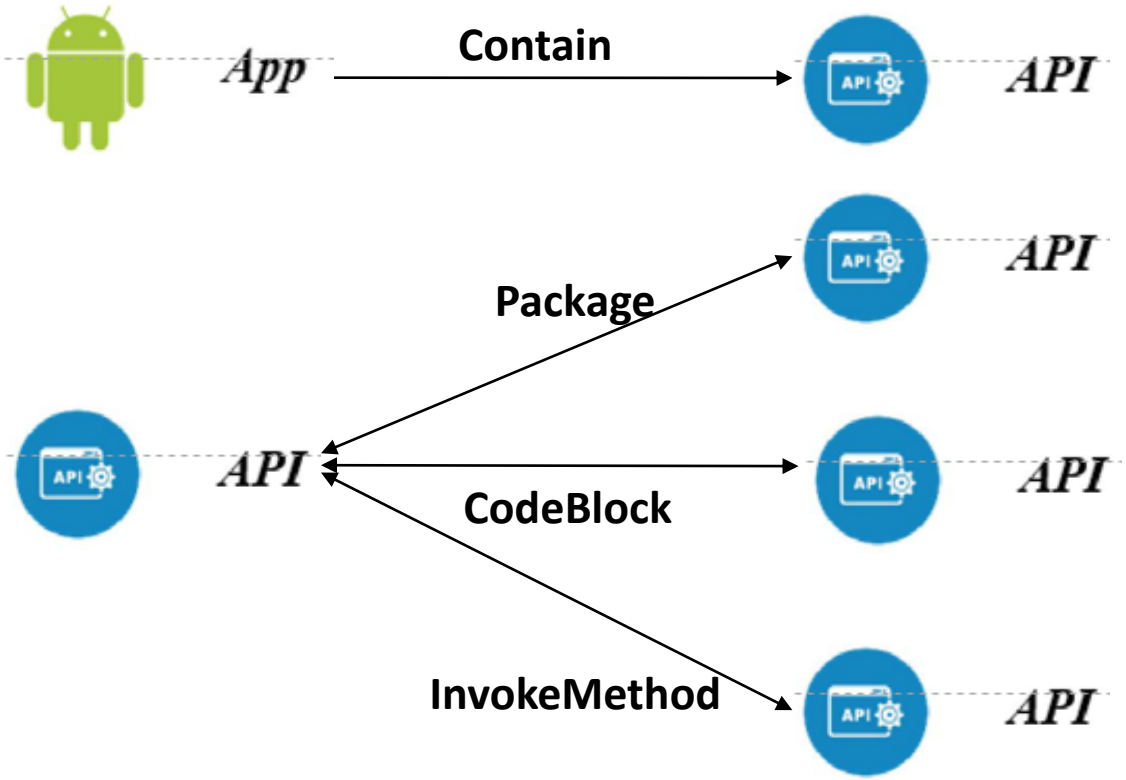

Android app is compiled and packaged in a single archive file (.apk) that includes the app code (.dex file), resources, assets, and manifest file.
HinDroid System Architecture
Listing 1: An example of smali code
1 .method protected
2 loadLibs(Landroid/content/Context;)V
3 .locals 4
4 :try_start_0
5 new-instance v0, Ljava/io/BufferedReader;
6 new-instance v1, Ljava/io/InputStreamReader;
7 invoke-static {}, Ljava/lang/Runtime;->getRuntime()Ljava/lang/Runtime;
8 move-result-object v2
9 const-string v3, "getprop_ro.product.cpu.abi"
10 invoke-virtual {v2, v3}, Ljava/lang/Runtime;->exec(Ljava/lang/String;)
Ljava/lang/Process;
11 move-result-object v2
12 invoke-virtual {v2}, Ljava/lang/Process;->getInputStream()Ljava/io/InputStream
13 .....
14 .end method
Android Apps
Unizipper
Decompiler
Smali Codes
Feature Extractor
API Calls
Relationship Analysis
API-CodeBlock
API-Package
API-InvokeMethod
HIN Constructor
Multi-Kernel Learner
Classification Model
Pre-computed Kernels
Malware Detector
Detection Report
API-1
API-3
API-2
API-4
Entities:
App
API
Relations:
Contain
CodeBlock
Package
InvokeMethod
Kernel
Unknown App

框架原理

- 特征构建
  - 2种实体四种关系
  - 异构信息网络
- 基于HIN的多核学习模型构建

框架原理

- 2种实体
  - APP(微信、QQ、微博等)
  - API(文件读写、网络连接等关键API)
- 四种关系



框架原理

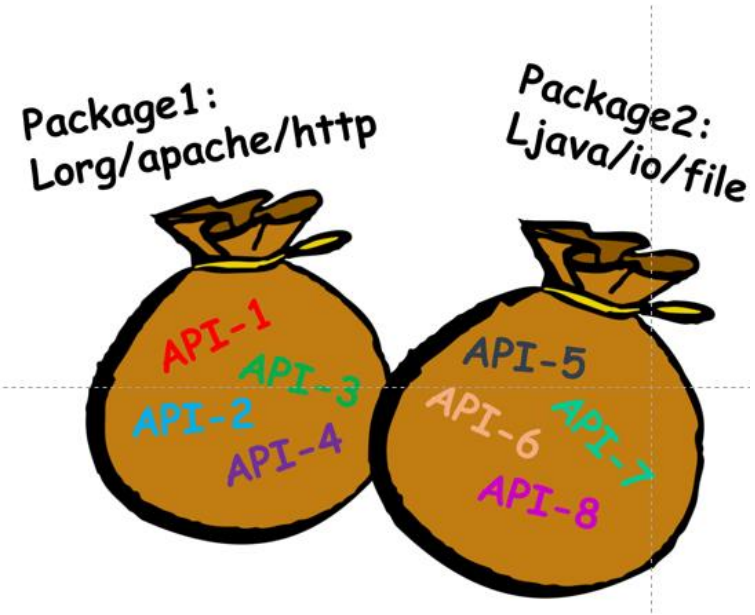
- 关系1 -> Contain关系（APP-API）

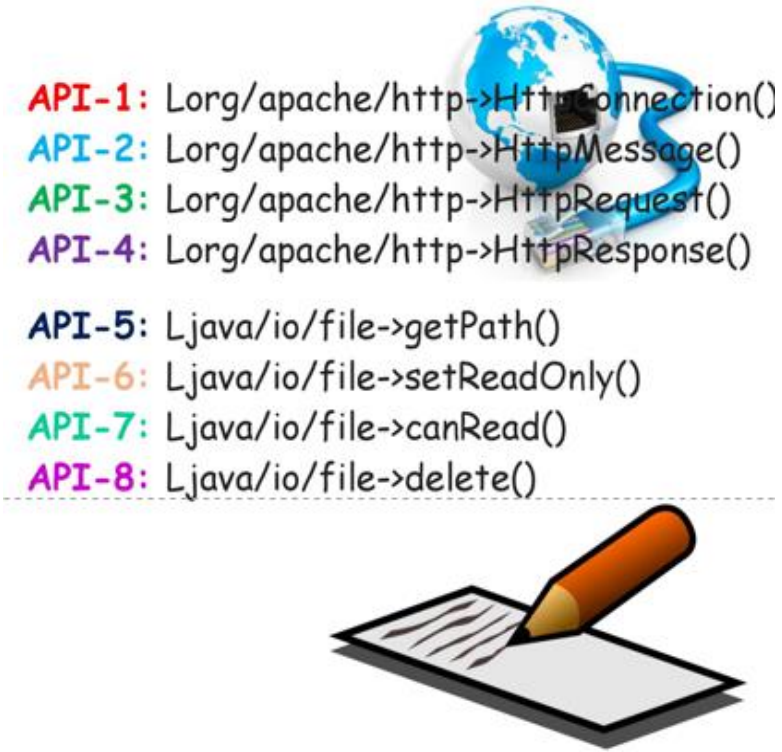
Smali Code:

APIs

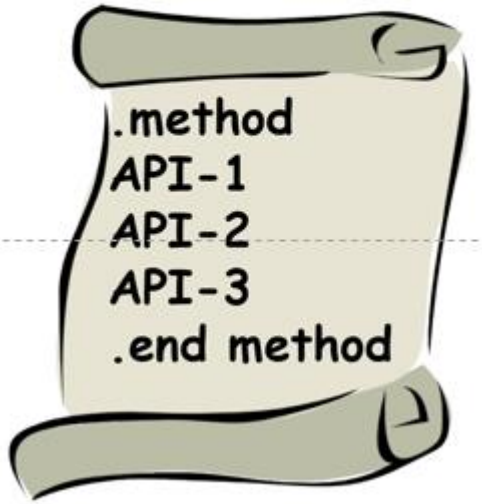
```
.method protected
loadLibs(Landroid/content/Context;)V
.locals 4
:try_start_0
new-instance v0, Ljava/io/BufferedReader;
new-instance v1, Ljava/io/InputStreamReader;
invoke-static {}, Ljava/lang/Runtime;->getRuntime()Ljava/lang/Runtime;
move-result-object v2
const-string v3, "getprop ro.product.cpu.abi"
invoke-virtual {v2, v3}, Ljava/lang/Runtime;->exec(Ljava/lang/String;)
    Ljava/lang/Process;
move-result-object v2
invoke-virtual {v2}, Ljava/lang/Process;->getInputStream()Ljava/io/InputStrea
.....
.end method
```

- 关系2 -> Package关系（API-API）





- 关系3 -> CodeBlock关系（API-API）



Smali Code:

APIs

```
.method protected
loadLibs(Landroid/content/Context;)V
.locals 4
:try_start_0
new-instance v0, Ljava/io/BufferedReader;
new-instance v1, Ljava/io/InputStreamReader;
invoke-static {}, Ljava/lang/Runtime;->getRuntime()Ljava/lang/Runtime;
move-result-object v2
const-string v3, "getprop ro.product.cpu.abi"
invoke-virtual {v2, v3}, Ljava/lang/Runtime;->exec(Ljava/lang/String;)
    Ljava/lang/Process;
move-result-object v2
invoke-virtual {v2}, Ljava/lang/Process;->getInputStream()Ljava/io/InputStrea
.....
.end method
```

- 关系4 -> InvokeMethod关系（API-API）

Smali Code:

```
.method protected
loadLibs(Landroid/content/Context;)V
.locals 4
:try_start_0
new-instance v0, Ljava/io/BufferedReader;
new-instance v1, Ljava/io/InputStreamReader;
invoke-static {}, Ljava/lang/Runtime;->getRuntime()Ljava/lang/Runtime;
move-result-object v2
const-string v3, "getprop ro.product.cpu.abi"
invoke-virtual {v2, v3}, Ljava/lang/Runtime;->exec(Ljava/lang/String;)
    Ljava/lang/Process;
move-result-object v2
invoke-virtual {v2}, Ljava/lang/Process;->getInputStream()Ljava/io/InputStrea
.....
.end method
```

APIs

1.invoke-static：调用实例的静态方法

2.invoke-virtual:调用实例的虚方法

3.invoke-direct：调用实例的直接方法

4.invoke-super：调用实例的父类方法

5.invoke-interface：调用实例接口方法

算法原理

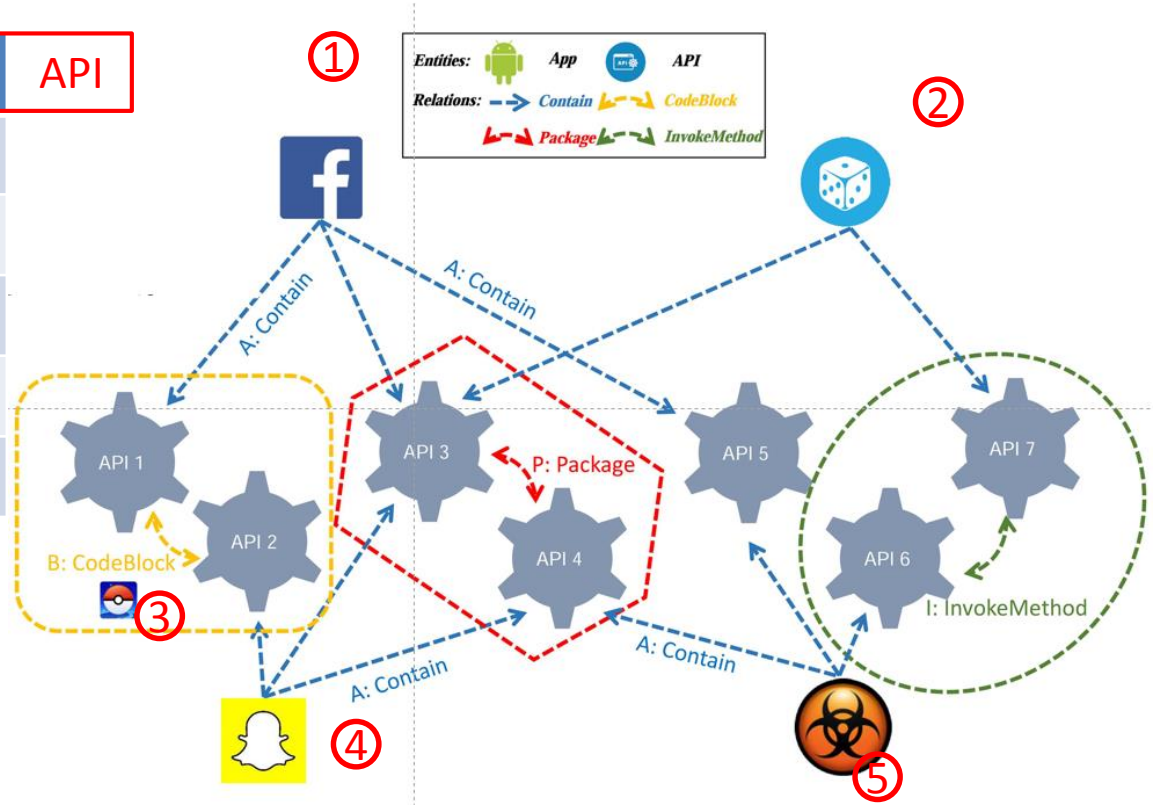
- 四个邻接矩阵

矩阵	元素	含义
A	a_{ij}	APP_i包含API_j, a_{ij}=1 否则， a_{ij}=0
B	b_{ij}	API_i和API_j在同一个函数中, b_{ij}=1 否则， b_{ij}=0
P	p_{ij}	API_i和API_j ,有相同的包名 p_{ij} =1 否则, p_{ij}=0
I	i_{ij}	API_{ij}和API_{ij}被调用的方式相同, i_{ij}=1 否则， i_{ij}=0

框架原理

- A矩阵(5x7)

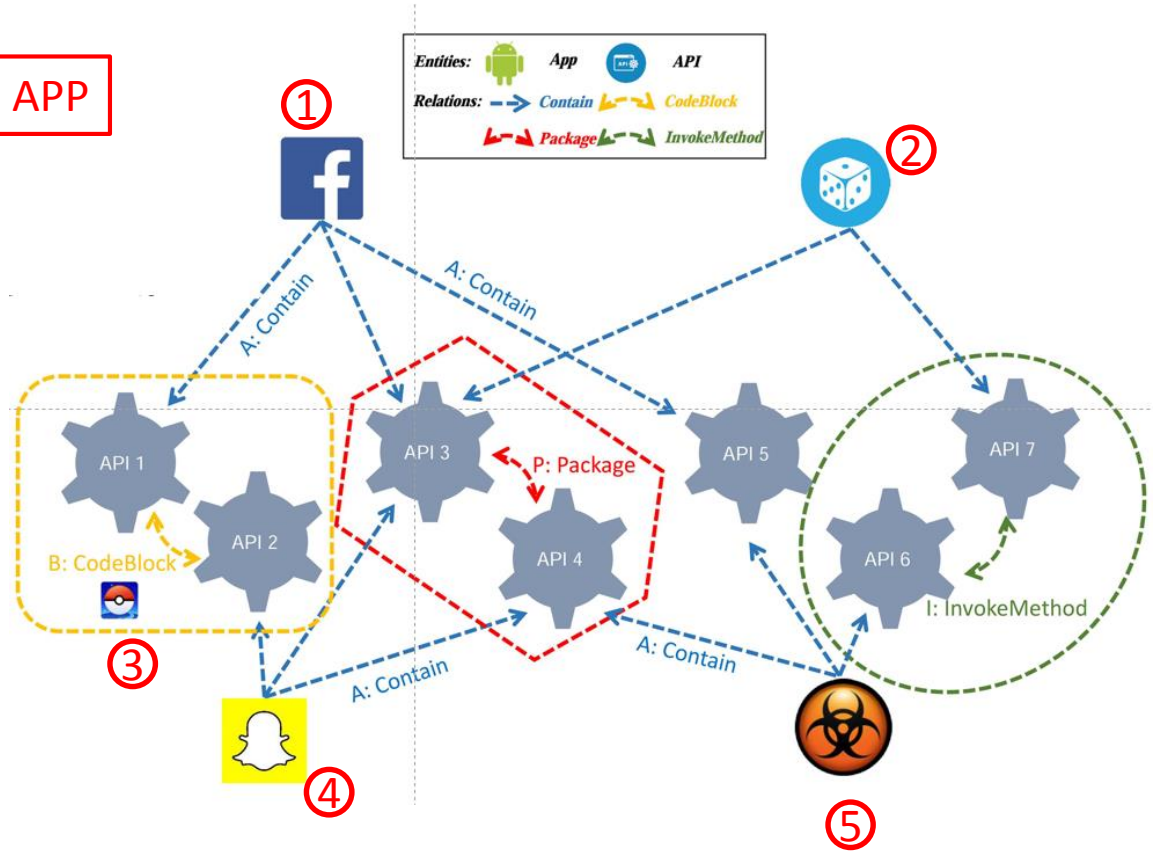
	1	2	3	4	5	6	7	API
1	1	0	1	0	1	0	0	
2	0	0	1	0	0	0	1	
3	1	1	0	0	0	0	0	
4	0	1	1	1	0	0	0	
5	0	0	0	1	1	1	0	
APP								



框架原理

- B矩阵（7x7）API_i和API_j在同一个函数里，p_{ij}=1 否则，p_{ij}=0

	1	2	3	4	5	6	7	APP
1	0	1	0	0	0	0	0	
2	1	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	
6	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	
APP								



I矩阵、P矩阵行列的大小都是7x7

框架原理

- 交换矩阵构建（只选取对称元路径）

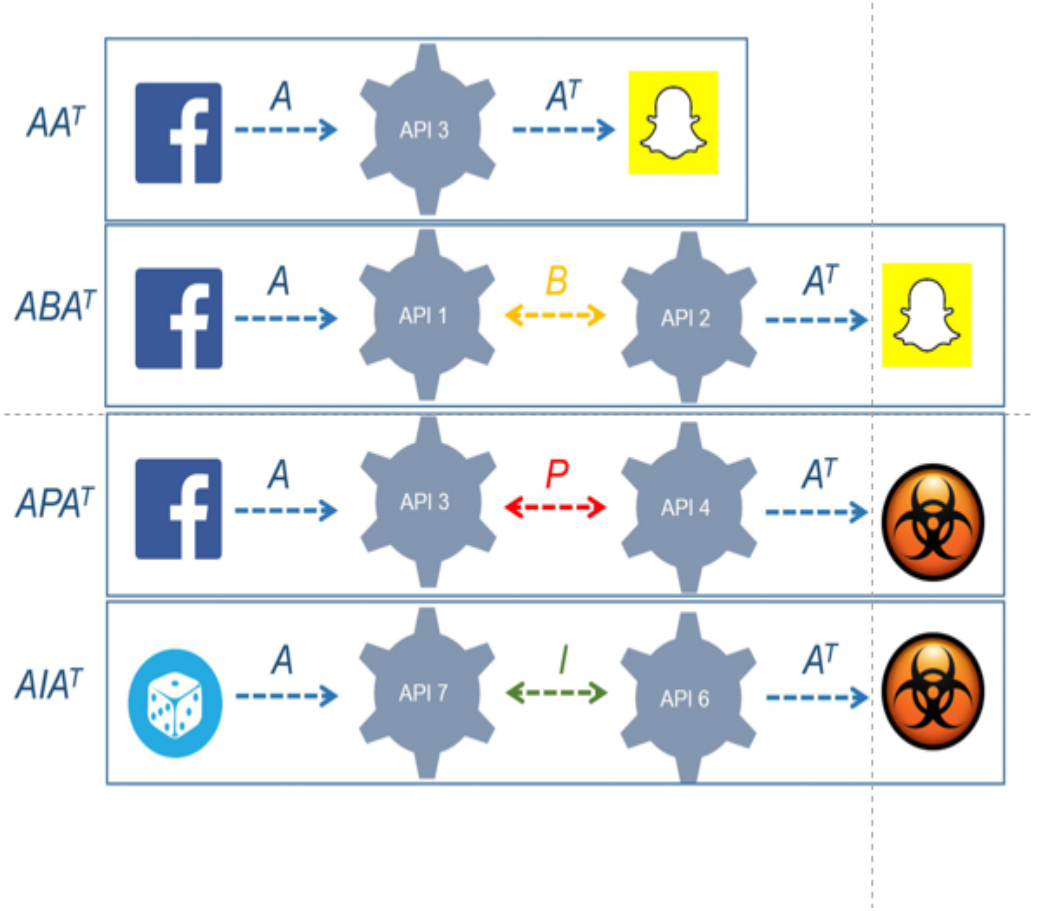
Commuting matrix: M_{φ}

$M_{AA^T}=AA^T = 1 \xrightarrow{A} 2 \xrightarrow{A^T} 1$

$M_{ABA^T}=ABA^T = 1 \xrightarrow{A} 2 \xrightarrow{B} 2 \xrightarrow{A^T} 1$

$M_{APA^T}=APA^T = 1 \xrightarrow{A} 2 \xrightarrow{P} 2 \xrightarrow{A^T} 1$

$M_{AIA^T}=AIA^T = 1 \xrightarrow{A} 2 \xrightarrow{I} 2 \xrightarrow{A^T} 1$



框架原理

- Commuting matrix(M_φ)含义

$M_\varphi(\mathrm{i},\mathrm{j})$物理含义?

	1	2	3	4	5	APP
1						
2						
3						
4						
5						
APP						

以AA^T 为例：
$t(x_i)$=$[t_{i1}, t_{i1} \dots\dots, t_{in}]$
t(x_i)表示特定元路径下一个APP的特征

在AA^T 路径下：t(x_i)表示两个APP之间具有的相同的API

$M_\varphi(\mathrm{i},\mathrm{j}) = \mathrm{t}(x_i)\ \mathrm{t}(x_j)$：具有的相同的API为特征，计算两个APP之间的相似度

$M_\varphi(\mathrm{i},\mathrm{j}) = \mathrm{t}(x_i)\ \mathrm{t}(x_j)$

表示在元路径表征的关系特征下，两个APP之间的相似度。

框架原理

- SVM分类
  - 假如有数据集合D：m个有标签的APP文件

$D = \{(x_1, y_1), (x_2, y_2), \ldots\ldots (x_m, y_m)\}$　　$y_i \in (-1, +1)$

其中，x_m代表APP的id，y_m代表APP标签

条件一

Commuting matrix M_φ(m x m)

条件二

$f(x) = w^T \varphi(x) + b$

目标

框架原理

- 模型构建
  - 假设存在一个核函数$k(x_j, x_i) = \varphi(x_j)^T \varphi(x_i)$

$$\min_{w,b} \frac{1}{2} ||w||^2$$

$$\text{s.t. } y_i*(w^T \varphi(x_i) + b) \geq 1, i = 1,2,3 \ldots\ldots m$$

通过拉格朗日乘子，转换为对偶问题

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j \, y_i \, y_j \, \varphi(x_j)^T \varphi(x_i)$$

$M_{\varphi}(i,j) = k(x_j, x_i)$

$$s.t. \sum_{i=1}^{m} \alpha_i y_i = 0$$

二次规划算法

求解出α_i

框架原理

- 模型构建
  - 判别函数生成

$$f(x) = w^T \varphi(x) + b$$

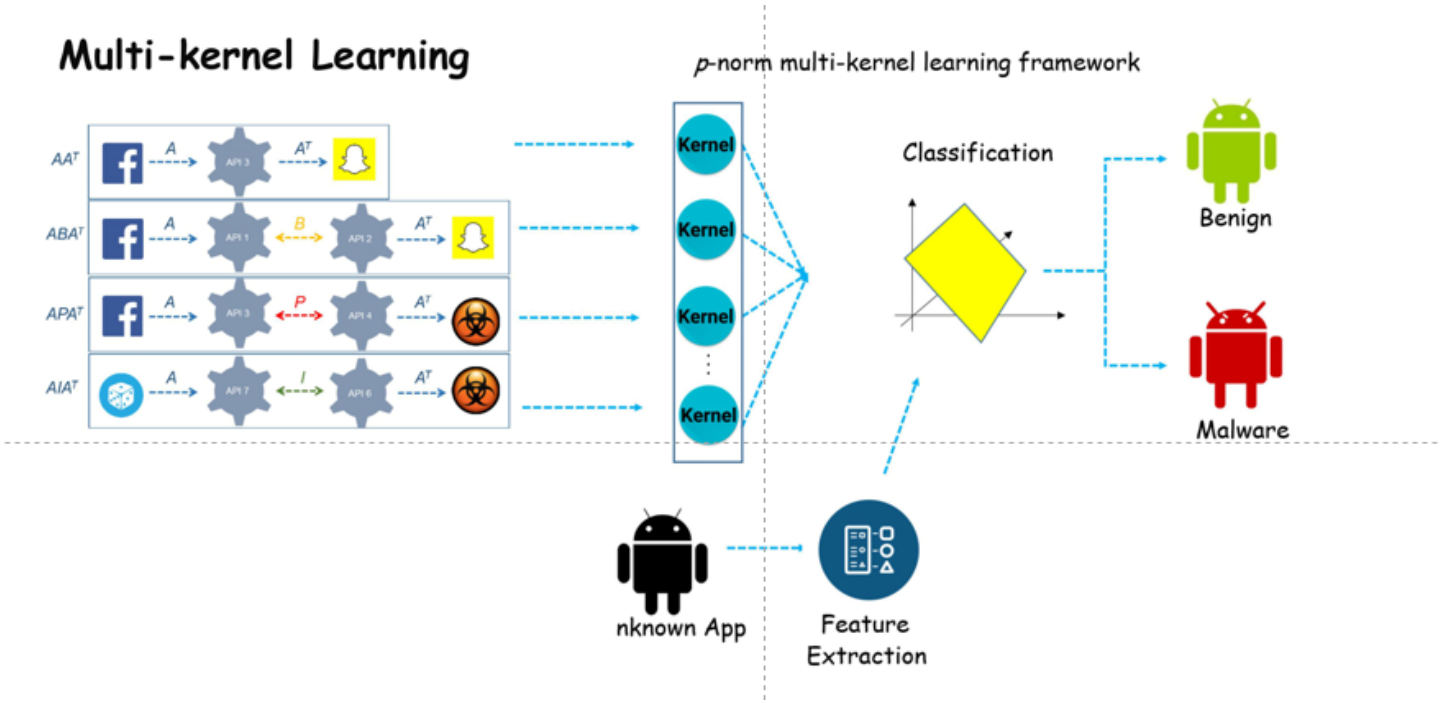
$$= \sum_{i=1}^{m} \alpha_i y_i \varphi(x_i)^T \varphi(x) + b$$

$$= \sum_{i=1}^{m} \alpha_i y_i k(x, x_i) + b$$

对于新来的样本x，可以根据$f(x)$完成分类

框架原理

- 基于HIN的多核学习模型



实验分析

- 评价指标

Table 2: Performance indices of Android malware detection

Indices	Description
TP	# of apps correctly classified as malicious
TN	# of apps correctly classified as benign
FP	# of apps mistakenly classified as malicious
FN	# of apps mistakenly classified as benign
$Precision$	$TP/(TP+FP)$
$Recall$	$TP/(TP+FN)$
ACC	$(TP+TN)/(TP+TN+FP+FN)$
$F1$	$2*Precision*Recall/(Precision+Recall)$

实验分析

- 单核和多核学习效果对比分析

Table 3: Detection performance evaluation

PID	Method	F1	β	ACC	TP	FP	TN	FN
1	AA^T	0.9529	0.1069	94.40%	283	19	189	19
2	ABA^T	0.9581	0.0900	95.00%	286	9	189	16
3	APA^T	0.9495	0.0858	94.20%	273	0	198	29
4	AIA^T	0.9183	0.0623	90.40%	270	16	182	32
5	$ABPB^TA^T$	0.9479	0.0670	94.00%	273	1	197	29
6	$APBP^TA^T$	0.9502	0.0565	94.20%	277	4	194	25
7	$ABIB^TA^T$	0.8683	0.0639	84.60%	254	29	169	48
8	$AIBI^TA^T$	0.8722	0.0639	85.00%	256	29	169	46
9	$APIP^TA^T$	0.8373	0.0445	81.20%	242	34	164	60
10	$AIPI^TA^T$	0.8761	0.0572	86.60%	237	2	196	65
11	$ABPIP^TB^TA^T$	0.9184	0.0616	90.80%	259	3	195	43
12	$APBIB^TP^TA^T$	0.8597	0.0617	84.60%	236	11	187	66
13	$ABIPI^TB^TA^T$	0.9284	0.0426	91.80%	266	5	193	36
14	$AIBPB^TI^TA^T$	0.8237	0.0426	82.60%	218	3	195	84
15	$AIPBP^TI^TA^T$	0.8597	0.0469	81.60%	215	5	193	87
16	$APIBI^TP^TA^T$	0.8597	0.0458	84.60%	236	11	187	66
17	Combined-kernel (5)	0.9214	—	91.20%	258	0	198	44
18	Combined-kernel (16)	0.9740	—	96.80%	300	14	184	2
19	Multi-kernel (5)	0.9834	—	98.00%	297	5	193	5
20	Multi-kernel (16)	**0.9884**	—	**98.60%**	299	4	194	3

实验分析

- 对比分析

Table 5: Comparisons with other mobile security products

Family	Sample #	Norton	Lookout	CM	*HinDroid*
Lotoor	78	75	74	76	78
RevMob	52	46	50	48	52
Malapp	33	29	32	30	33
Fakebank	31	29	30	29	30
Generisk	29	29	29	29	29
GhostPush	19	15	16	18	18
Fakegupdt	16	15	14	14	16
Danpay	21	19	20	20	21
HideIcon	12	11	9	8	12
Idownloader	11	10	9	9	10
Total	302	278	283	281	299
DetectionRate	–	92.05%	93.71%	93.05%	**99.01%**

For the comparisons, we use all the latest versions of the mobile security products (i.e., Clean Master (CM): 2.08, Lookout: 10.9-7f33b3e, and Norton: 3.17.0.3205)

优劣分析

- 表征能力强。让恶意软件变种难以逃脱检测。
- 基于静态分析，无法处理加固或者具有动态加载功能的恶意软件。

应用总结

- 框架具有较好的扩充性。可运用与医疗或者生物信息等各个方面。

参考文献

1. Abdulhayoglu, Melih, et al. "HinDroid: An Intelligent Android Malware Detection System Based on Structured Heterogeneous Information Network." *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* ACM, 2017:1507-1515.
2. https://blog.csdn.net/swy520/article/details/78962895
3. http://home.cse.ust.hk/~yqsong/papers/2017-HIN-Metagraph.pdf

谢谢！

大成若缺，其用不弊。大盈若冲，其用不穷。大直若屈。大巧若拙。大辩若讷。静胜躁，寒胜热。清静为天下正。