Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

加密移动流量分析方法

硕士研究生　吴泽瀚

2022年04月23日

内容提要

- 背景简介
- 基本概念
  - 网络流量
  - 流量分析方法
  - 图与图神经网络
- 算法原理
  - AppScanner
  - MAppGraph
- 应用总结
- 参考文献

预期收获

- 预期收获
  - 1. 了解加密移动流量
  - 2. 了解流量分析的常用方法
  - 3. 了解利用流量分析进行应用程序分类
  - 4. 了解应用领域和发展方向等

背景简介

- 加密移动流量分析
  - 5G技术在内的通信技术的普及使智能移动设备数量显著增加
  - 移动设备及其应用的安全性对底层网络的安全性有很大影响
  - 超过80%的移动通信流量加密或采用传输层安全协议TLS
  - 加密移动流量分析提供了一种无需访问设备的非侵入性方法，来检测和监控移动设备上的应用程序

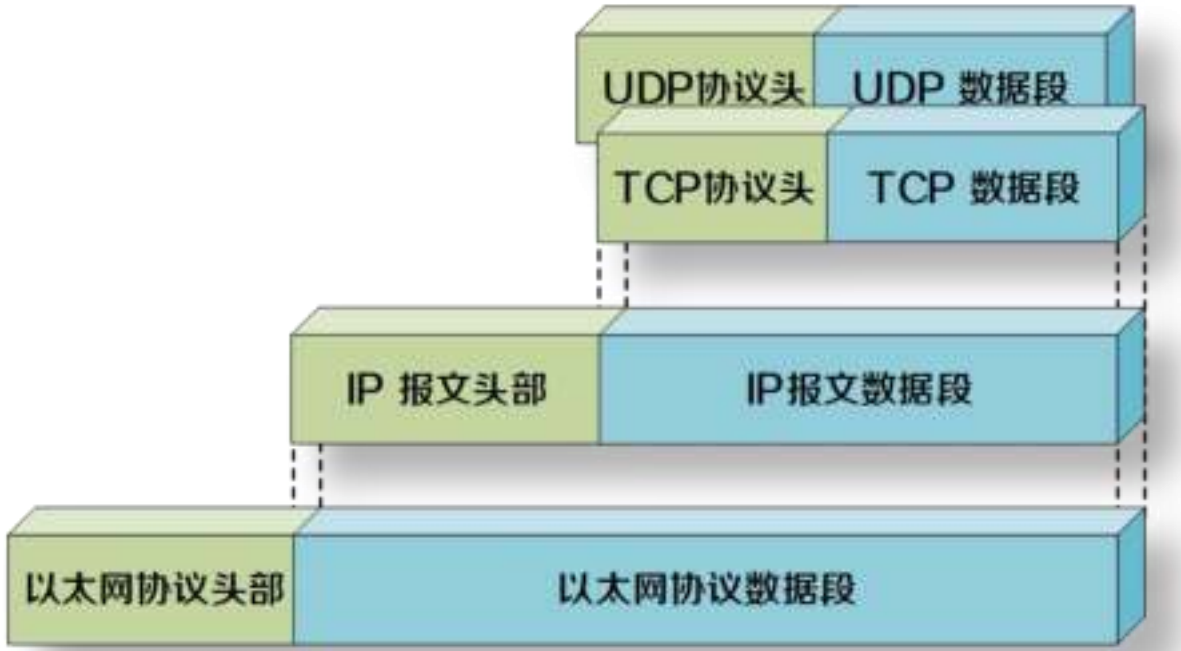
背景简介

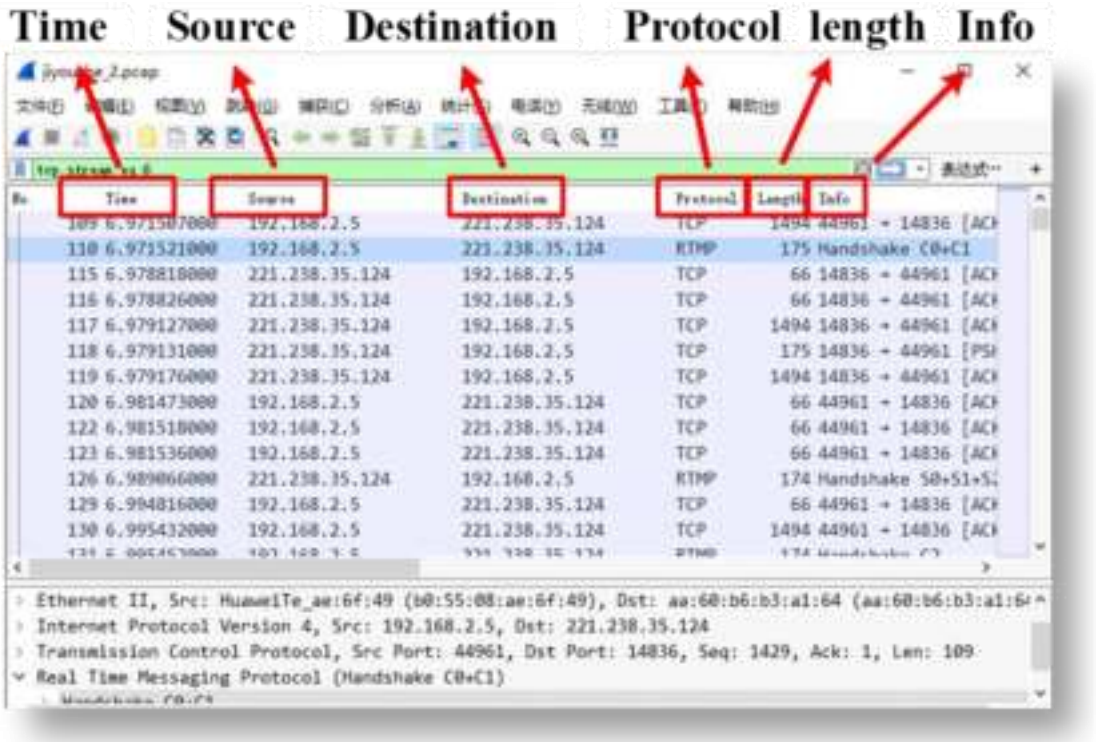
- 传统方法分析加密移动流量存在困难
  - 流量经过加密
    - 基于分析有效载荷的方法无法分析加密流量
  - 移动流量大多使用https协议传输
    - 基于端口的分类方法无法对移动流量进行分类
  - 用户行为随时间动态变化
    - 移动应用在短时间内捕获的流量可能并不代表其完整的流量模式
  - 当前移动应用程序使用内容交付网络CDN和三方库
    - 无法利用域名解析或ip地址查找等策略，同时存在较多模糊流量

基本概念

基本概念

- 网络流量
  - 通过特定网络节点的数据包和网络请求数量
  - 常见的流量分析通常分析udp/tcp



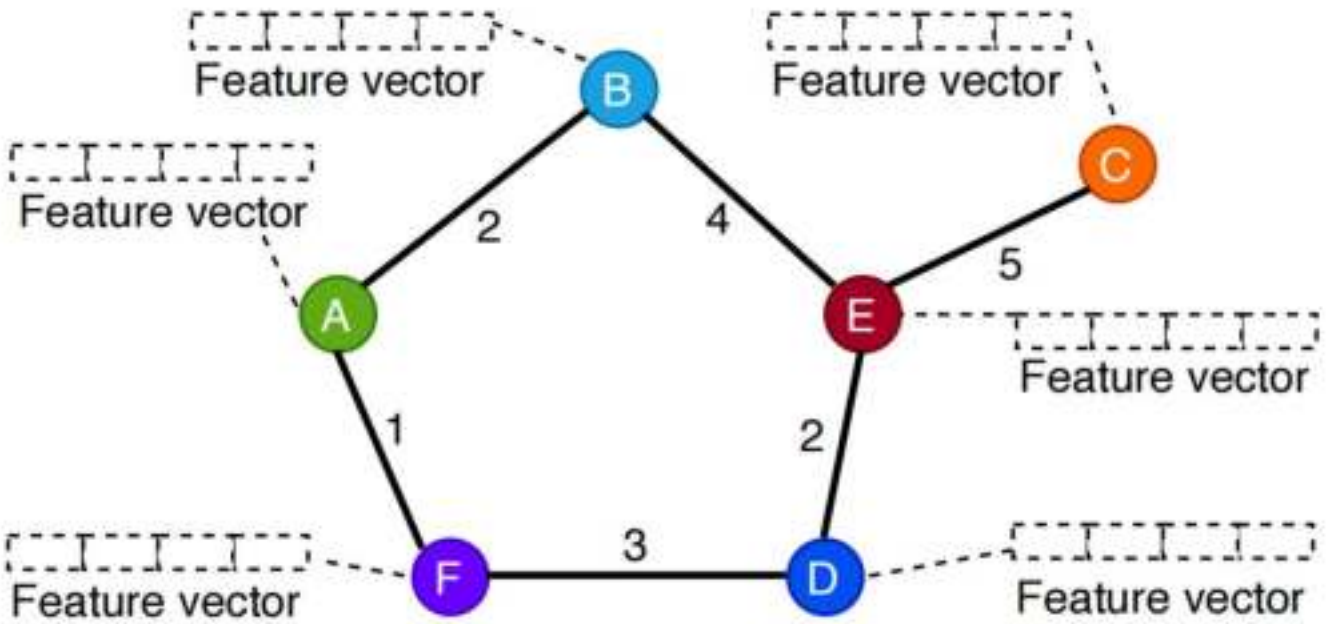


基本概念

- 流量分析方法
  - 基于负载
    - 分析出现在流量数据有效负载中的应用程序模式或者签名，如常用的“深度包检测DPI”
    - 深度包检测：对应用层数据的应用协议识别，数据包内容检测与深度解码
  - 基于特征
    - 根据捕获的流量特征的统计特性建立分类模型，不依赖负载，只提取数据包头的有关信息，例如移动流量的数据包大小、流持续时间、到达时间等，学习流量的行为模式

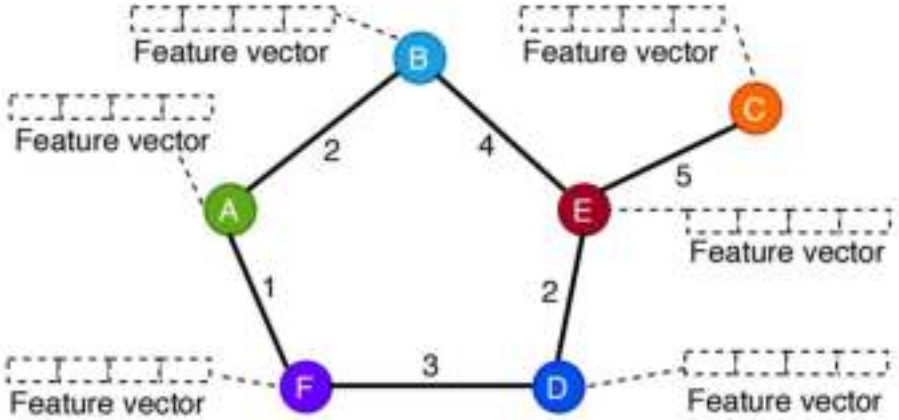
基本概念

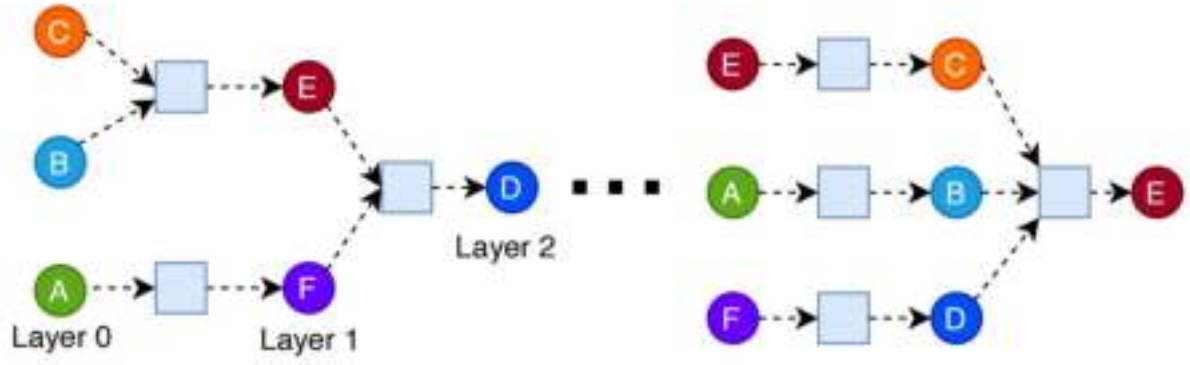
- 图基本介绍
  - 结构化数据
  - 作为一种非欧几里得数据，传统的卷积网络无法分析，分析图结构需要图神经网络的帮助
  - 图分析可以应用到下游的节点分类、链路预测、图分类等问题上



基本概念

- 图神经网络
  - 利用邻接矩阵表示图结构
  - 聚合（获取节点的邻接信息）
  - 更新（通过激活函数以及权重确定节点最新信息）
  - 循环（获取更多层的邻接信息）
- 图卷积神经网络
  - 修改了聚合部分，引入度矩阵
  - $H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)})$
  - 最后更新的节点信息包括邻接的节点的度信息





算法原理

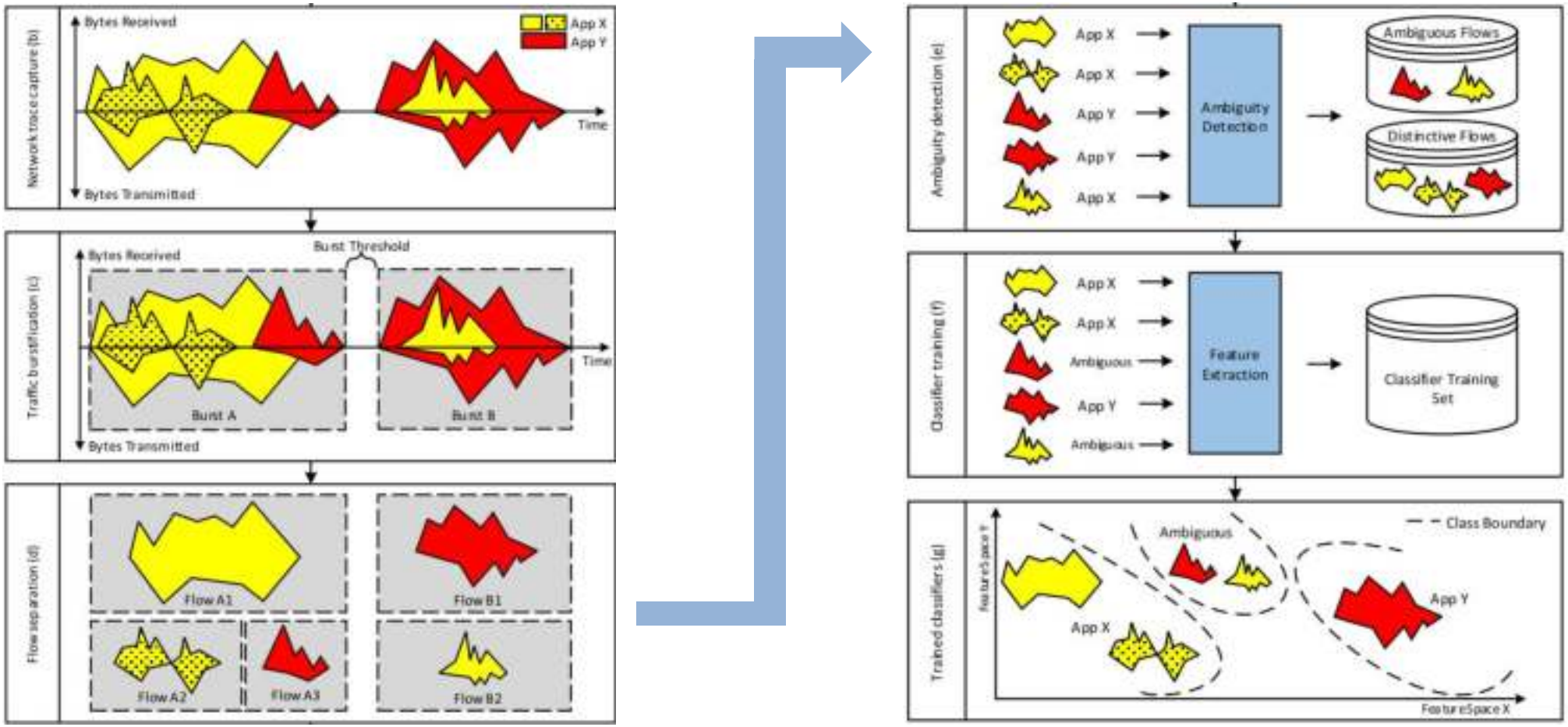
算法原理 AppScanner

T	高鲁棒性移动应用程序分类
I	加密移动流量
P	1.根据burst和flow将流量进行划分 2.提取流特征训练分类器 3.通过模糊检测以及分类验证提高精度
O	加密移动流量所属应用程序

P	识别加密移动流量对应的应用程序
C	应用程序存在模糊流之外的独特流量
D	模糊流量导致应用程序分类效果较差
L	IEEE TIFS 2018 (CCF-A类期刊)

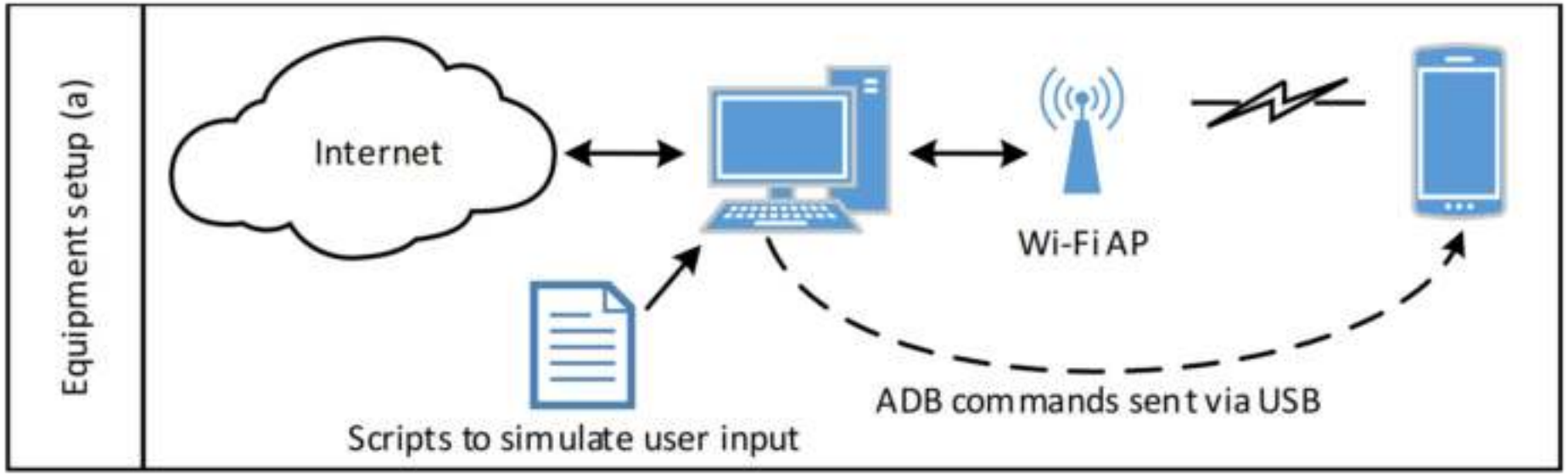
算法原理 AppScanner

- 系统概述
  - 流量捕获->流量突发划分->流划分->模糊检测->分类器训练->分类验证



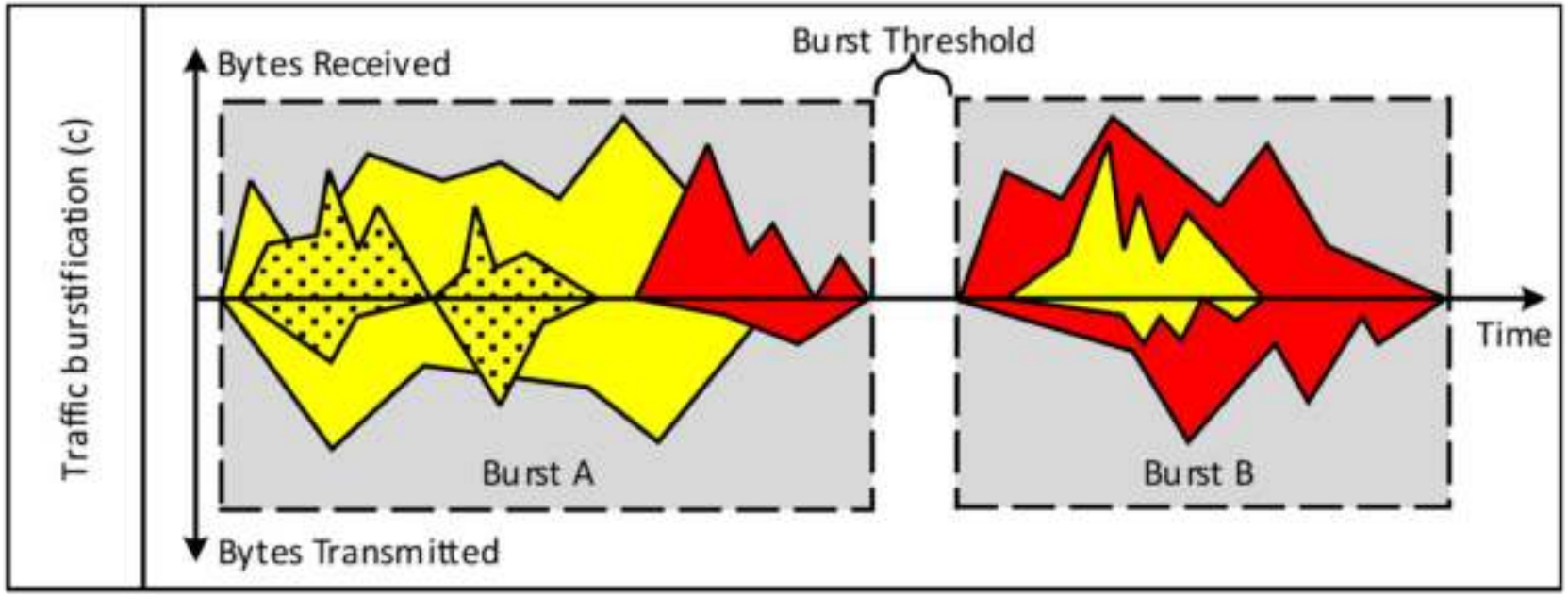
算法原理 AppScanner

- 数据收集（UI模糊）
  - 将工作站配置在wifi接入点与互联网之间
  - 编写脚本模拟应用程序中的用户操作，借助ADB通过USB与目标移动设备相连
    - ADB为安卓调试桥，支持在PC上对安卓手机进行操作
  - 通过脚本运行，可以实现大规模的自动化和并行化



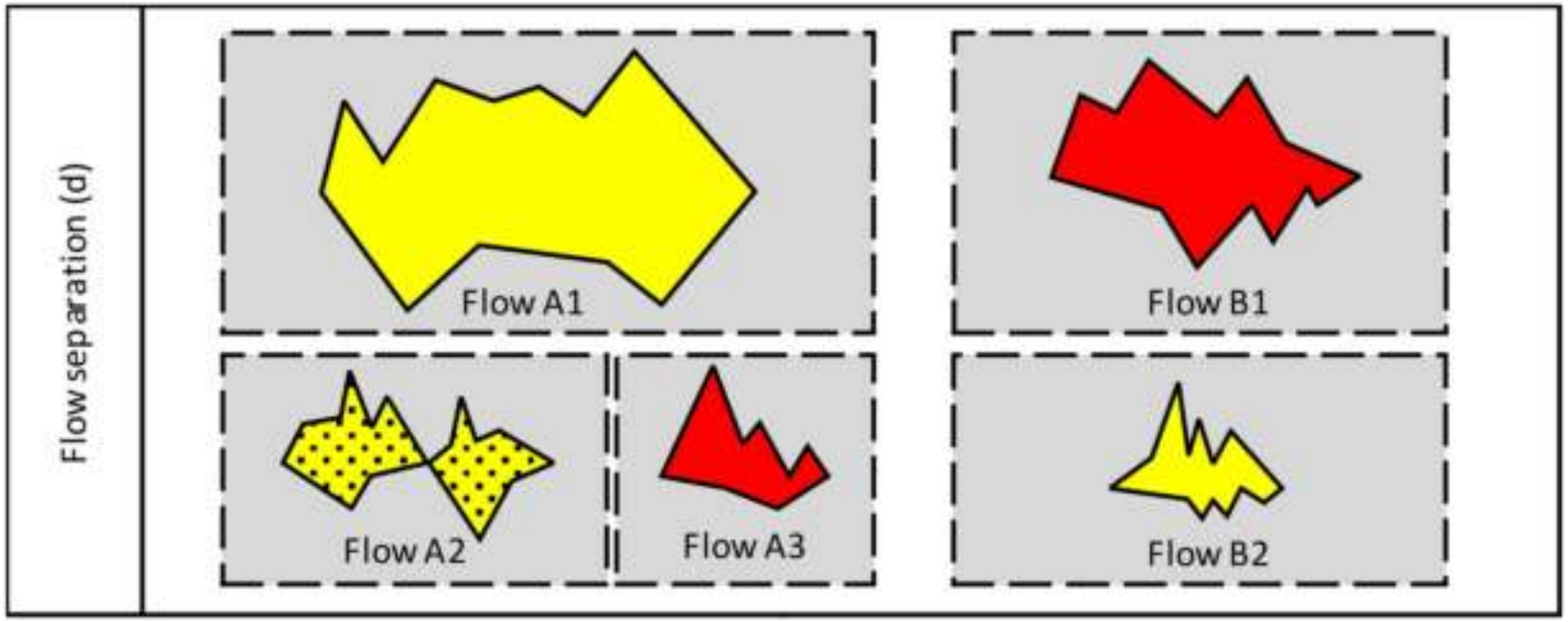
算法原理 AppScanner

- Burst（突发）
  - 一组同时发生的所有网络数据包（不考虑源地址或目的地址）
  - 当且仅当设置的突发阈值内没有新分组到达时，创建新的突发分组
  - 使用突发，将网络流量逻辑上划分为离散的、可管理的部分



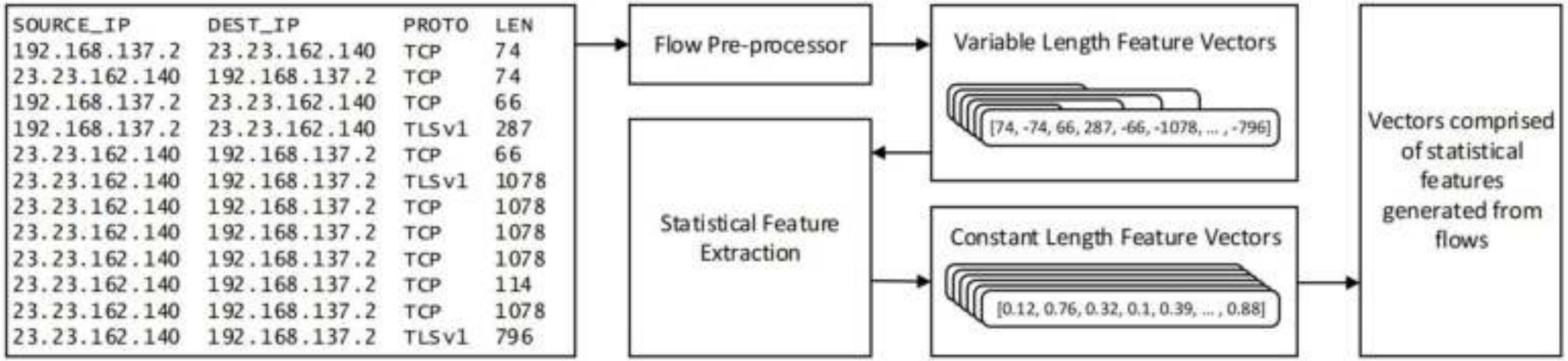
算法原理 AppScanner

- Flow（流）
  - 具有相同远程IP地址的数据包序列（在一个突发中）
  - 流在突发结束时结束，而TCP会话可以跨越突发存在
  - 在突发中存在流的重叠的情况



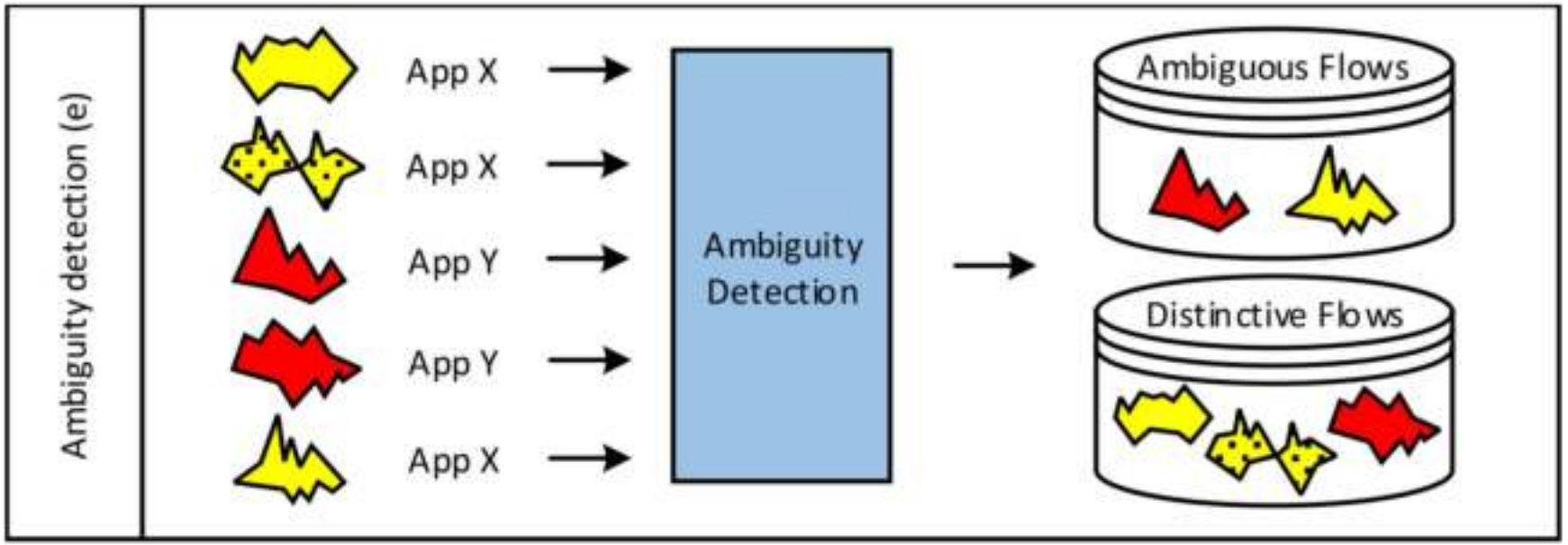
算法原理 AppScanner

- 特征生成
  - 由流生成统计特征
  - 选择流的三个特征，传入数据包的大小、传出数据包的大小、传入传出数据包的总大小（流中包含多个数据包）
  - 计算每个特征的最小值、最大值……峰度等统计数值
  - 任意长度的流被转化为长度54维的特征向量



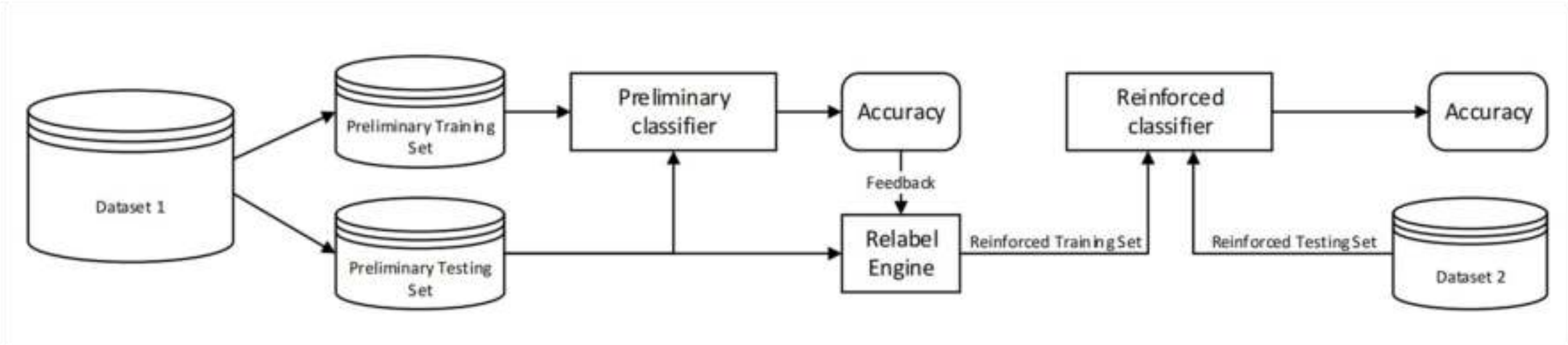
算法原理 AppScanner

- 模糊检测
  - 许多应用程序都共享相同的三方库（如广告库），这些库本身会产生流量
  - 跨应用的公共库会产生类似的网络流量，阻碍应用程序识别
  - 需要利用模糊检测来首先识别应用程序之间共享的流量
    - 共享的流量称为模糊流量，流中剩余的流量称为独特流量



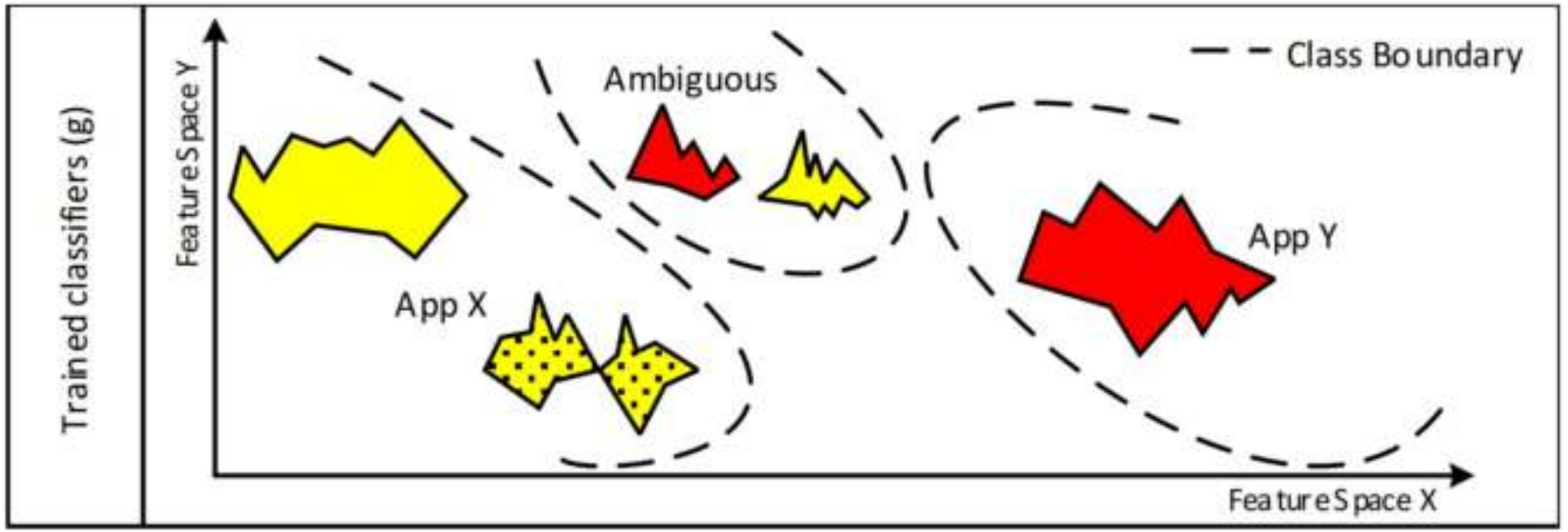
算法原理 AppScanner

- 模糊检测
  - 利用强化学习识别和重新标记模糊流量
  - 分类器训练分为初级分类器和强化分类器
  - 初级分类器和初始训练集仅用作识别模糊流量，向强化分类器提供额外的知识



算法原理 AppScanner

- 分类验证
  - 机器学习总是试图将一个未标记的示例放入最相似的类中
  - 不进行分类验证，会导致假阳性率的提升
  - 利用预测概率来衡量每个分类的确定程度，并和预设的分类验证阈值比较
    - 分类验证阈值：分类结果是否被采纳的界限
    - 分类验证阈值的设定影响着AppScanner的精度



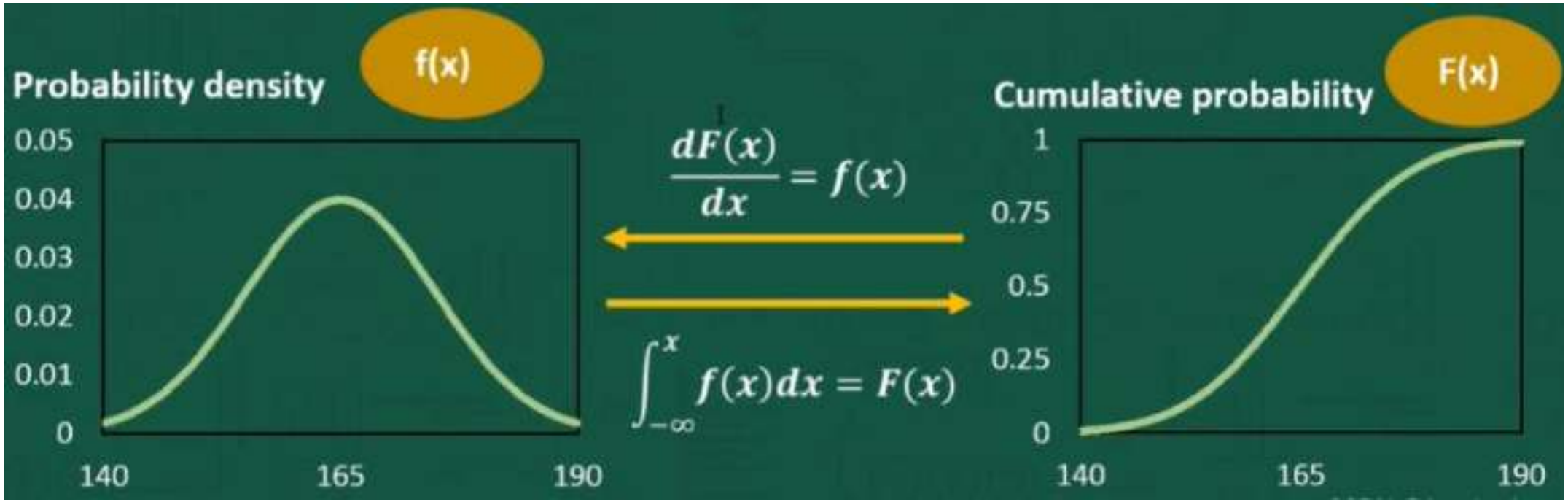
算法原理 实验结果

- 数据集生成
  - Dataset-1作为基线数据集
  - 不同的数据集改变不同的因素，时间、应用程序版本、设备与操作系统
  - 1a、4a、5a作为数据集的变体存在，主要是为了保证app数量一致

Name	Device	Operating System	Number of apps	App versions	Time of data collection
Dataset-1	Motorola XT1039	Android 4.4.4	110	Latest versions as at T_0	T_0
Dataset-1a	Motorola XT1039	Android 4.4.4	65	Latest versions as at T_0	T_0
Dataset-2	Motorola XT1039	Android 4.4.4	65	Latest versions as at T_0	T_0 + 6 months
Dataset-3	LG E960	Android 5.1.1	65	Latest versions as at T_0	T_0 + 6 months
Dataset-4	Motorola XT1039	Android 4.4.4	110	Latest versions as at T_0 + 6 months	T_0 + 6 months
Dataset-4a	Motorola XT1039	Android 4.4.4	65	Latest versions as at T_0 + 6 months	T_0 + 6 months
Dataset-5	LG E960	Android 5.1.1	110	Latest versions as at T_0 + 6 months	T_0 + 6 months
Dataset-5a	LG E960	Android 5.1.1	65	Latest versions as at T_0 + 6 months	T_0 + 6 months

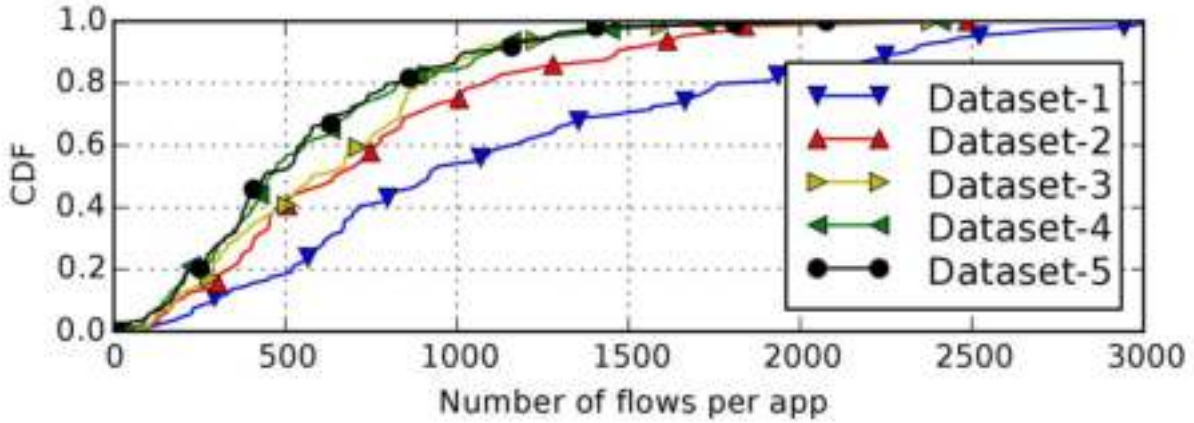
算法原理 实验结果

- CDF累积分布曲线
  - 概率密度函数pdf的积分
  - 可使用 CDF 确定取自总体的随机观测值将小于或等于特定值的概率。还可以使用此信息来确定观测值将大于特定值或介于两个值之间的概率



算法原理 实验结果

- 系统评估
  - 分两步评估系统的分类效果以及分析影响系统的因素的影响力度
  - 第一步使用单个数据集导出的训练集合和测试集对系统性能进行基线评估
  - 多个数据集上的基线效果比较
    - Dataset-1数据集上的基线效果最好，与每个应用中所含流数最大有关



Dataset	Precision(%)	Recall(%)	F1(%)	Accuracy (%)
Dataset-1	74.5	72.8	73.1	73.1
Dataset-1a	74.4	73.2	73.4	73.7
Dataset-2	68.8	67.6	67.7	65.5
Dataset-3	71.3	69.5	69.8	70.4
Dataset-4	68.6	66.6	66.9	67.4
Dataset-4a	68.5	66.8	67.1	66.9
Dataset-5	69.7	68.1	68.3	69.6
Dataset-5a	67.7	65.5	66.0	67.5

算法原理 实验结果

- 系统评估
  - 第二步使用完全独立的训练和测试集来进行全面测试
  - 了解时间、设备（包括操作系统）、应用程序版本以及设备和应用程序的版本组合等因素如何影响分类器分类的性能

Name	Device	Operating System	Number of apps	App versions	Time of data collection
Dataset-1	Motorola XT1039	Android 4.4.4	110	Latest versions as at T_0	T_0
Dataset-1a	Motorola XT1039	Android 4.4.4	65	Latest versions as at T_0	T_0
Dataset-2	Motorola XT1039	Android 4.4.4	65	Latest versions as at T_0	T_0 + 6 months
Dataset-3	LG E960	Android 5.1.1	65	Latest versions as at T_0	T_0 + 6 months
Dataset-4	Motorola XT1039	Android 4.4.4	110	Latest versions as at T_0 + 6 months	T_0 + 6 months
Dataset-4a	Motorola XT1039	Android 4.4.4	65	Latest versions as at T_0 + 6 months	T_0 + 6 months
Dataset-5	LG E960	Android 5.1.1	110	Latest versions as at T_0 + 6 months	T_0 + 6 months
Dataset-5a	LG E960	Android 5.1.1	65	Latest versions as at T_0 + 6 months	T_0 + 6 months

Test Name	Training Set	Testing Set	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)	Independent Variable	Apps
TIME	Dataset-1a	Dataset-2	44.5	43.1	42.6	40.9	Time	65
D-110	Dataset-4	Dataset-5	40.9	36.6	36.3	37.6	Device	110
D-110A	Dataset-4a	Dataset-5a	38.4	35.2	35.1	37.7	Device	65
D-65	Dataset-2	Dataset-3	43.5	38.3	39.0	39.6	Device	65
V-LG	Dataset-3	Dataset-5a	33.0	31.0	30.0	30.3	App versions	65
V-MG	Dataset-2	Dataset-4a	34.8	32.1	32.1	32.7	App versions	65
DV-110	Dataset-1	Dataset-5	23.3	19.5	19.3	19.2	Device & App versions	110
DV-65	Dataset-1a	Dataset-5a	22.3	19.7	19.3	19.5	Device & App versions	65

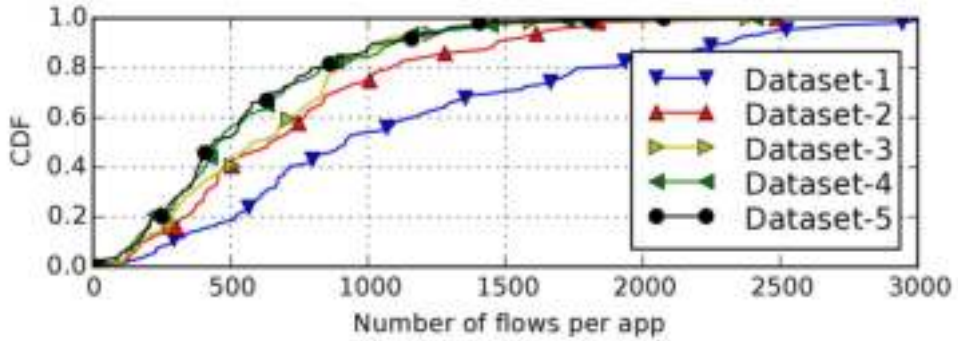
算法原理 实验结果

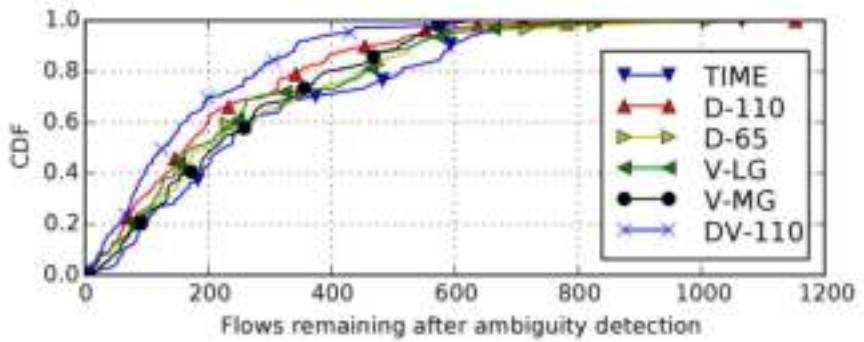
- 实验结论
  - 时间的影响：应用程序生成流量的逻辑在经过一段时间后基本不变
  - 设备的影响：设备以及操作系统的变化会对流量生成方式产生较小的影响
  - 应用程序版本的影响：应用程序代码逻辑的变化对应用程序生成网络流的方式影响较大

Test Name	Training Set	Testing Set	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)	Independent Variable	Apps
TIME	Dataset-1a	Dataset-2	44.5	43.1	42.6	40.9	Time	65
D-110	Dataset-4	Dataset-5	40.9	36.6	36.3	37.6	Device	110
D-110A	Dataset-4a	Dataset-5a	38.4	35.2	35.1	37.7	Device	65
D-65	Dataset-2	Dataset-3	43.5	38.3	39.0	39.6	Device	65
V-LG	Dataset-3	Dataset-5a	33.0	31.0	30.0	30.3	App versions	65
V-MG	Dataset-2	Dataset-4a	34.8	32.1	32.1	32.7	App versions	65
DV-110	Dataset-1	Dataset-5	23.3	19.5	19.3	19.2	Device & App versions	110
DV-65	Dataset-1a	Dataset-5a	22.3	19.7	19.3	19.5	Device & App versions	65

算法原理 实验结果

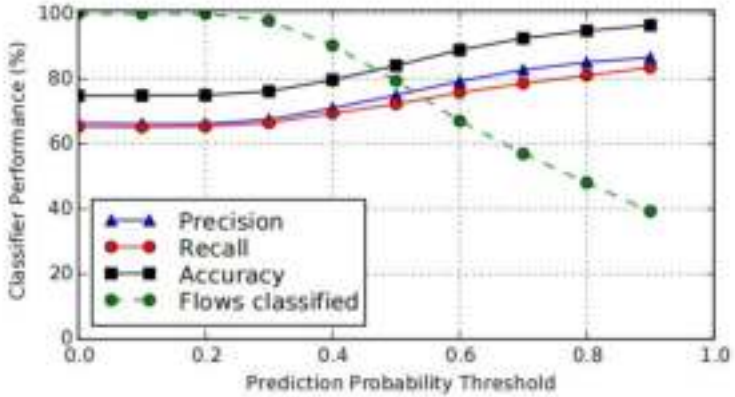
- 系统提升
  - 模糊检测：考虑了噪声对分类精度的影响，采用了三种噪声设置



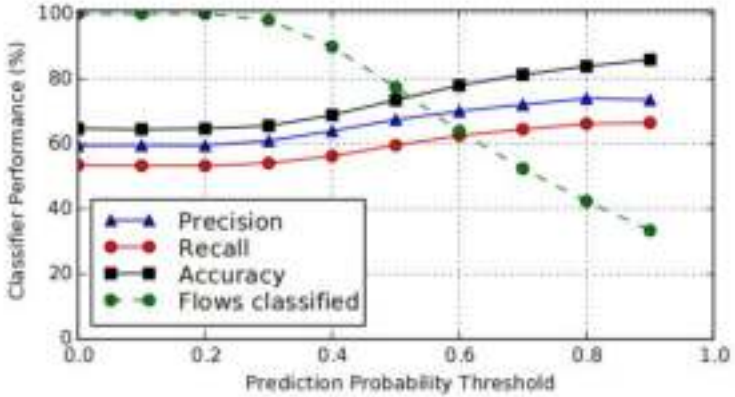


Test Name	Noise filtered				Noise ignored				Noise managed				Testing Set Details	
	P(%)	R(%)	F1(%)	A(%)	P(%)	R(%)	F1(%)	A(%)	P(%)	R(%)	F1(%)	A(%)	#Flows	%Ambiguous
TIME	66.9	66.4	65.7	72.9	61.4	65.3	61.9	64.5	66.4	65.3	64.6	74.8	54240	58.3
D-110	62.2	57.2	56.3	66.2	56.1	56.9	53.0	55.4	59.5	53.5	53.0	64.7	73811	59.6
D-110A	60.5	55.9	55.6	65.9	53.7	55.0	51.6	54.5	57.3	54.0	53.1	63.4	51995	60.4
D-65	64.4	59.8	59.4	67.5	59.4	59.0	56.0	59.0	63.8	58.8	58.5	68.1	47018	58.6
V-LG	47.8	47.3	44.6	52.8	41.4	46.6	41.4	43.9	44.7	45.6	42.5	54.5	51995	61.9
V-MG	52.4	50.6	49.5	58.1	48.9	50.0	47.2	50.2	52.8	50.3	49.3	62.4	51731	61.8
DV-110	37.0	35.0	33.2	41.0	32.8	34.8	31.0	32.4	35.0	33.7	32.0	46.3	73811	67.5
DV-65	35.8	34.8	32.9	39.8	30.9	34.0	30.2	31.3	33.8	33.8	31.5	44.0	51995	67.2

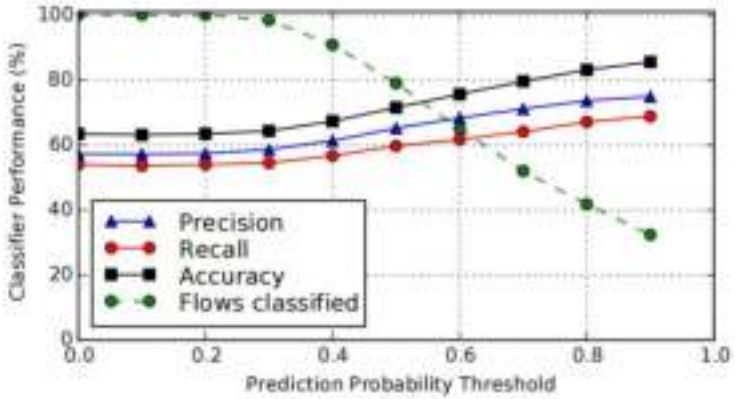
- 分类验证
  - 预定义的验证阈值：不同的验证阈值对分类效果的影响



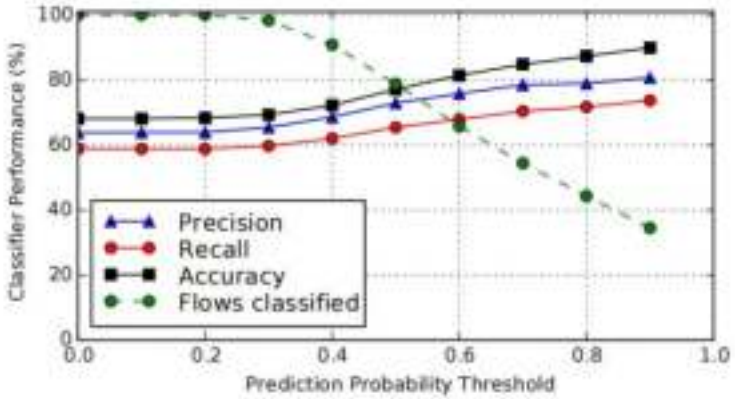
(a) Performance for the TIME test.



(b) Performance for the D-110 test.



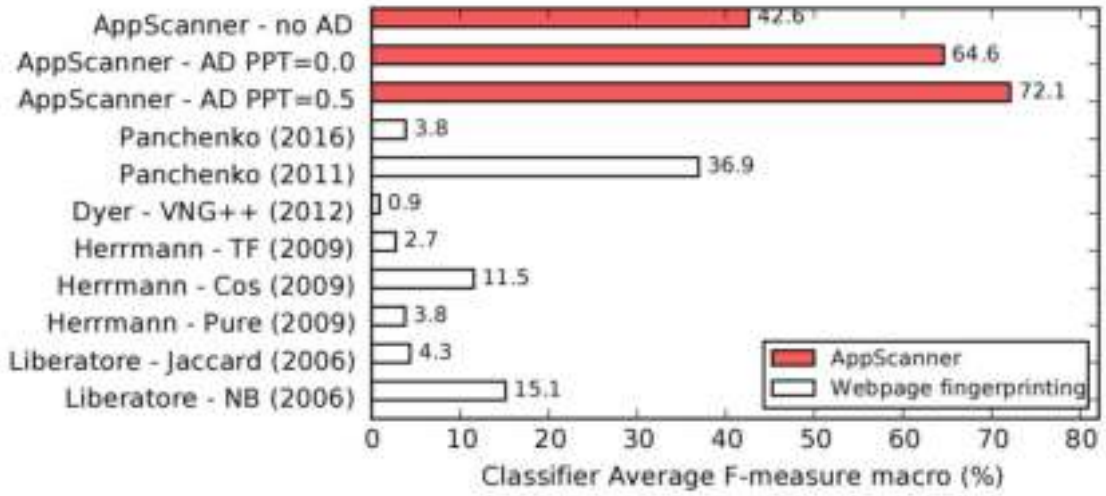
(c) Performance for the D-110A test.



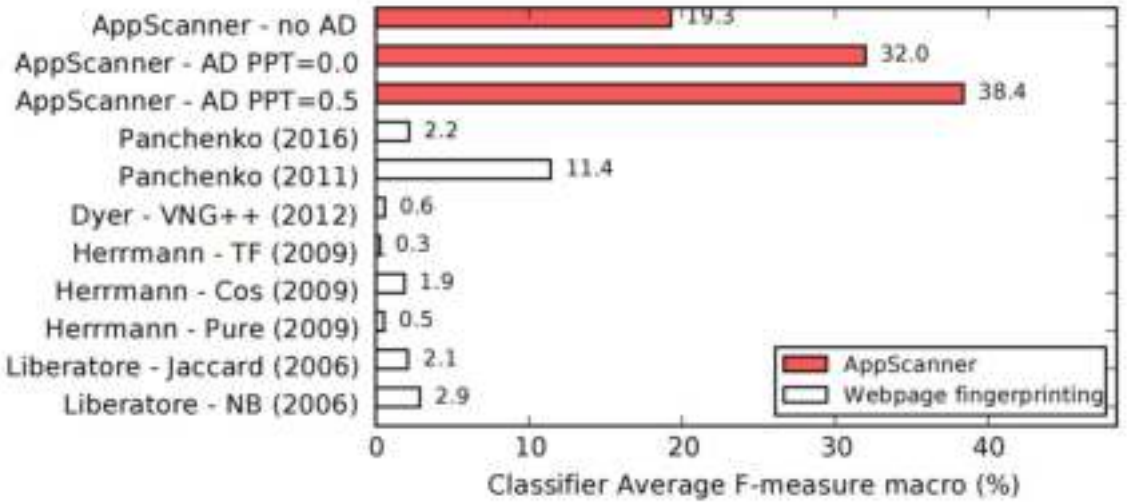
(d) Performance for the D-65 test.

算法原理 实验结果

- 效果比较
  - 与传统的流量分析及现有的移动流量分析技术比较



(a) Performance for the TIME test.



(b) Performance for the DV-110 test.

算法原理 优劣分析

- 优势
  - 模糊检测：结合强化学习，识别现代应用程序中共享的三方库产生的共享流量，即模糊流量
  - 应用程序分类的鲁棒性：分析时间、应用程序版本变化、设备和操作系统变化对应用程序分类的影响程度，在多种因素的影响下，仍能保持较高的分类精度
- 劣势
  - 流量覆盖率，因为采用脚本生成流量，产生的流量不能完全覆盖应用程序的流量行为
  - 某些应用程序并不产生独特的、不模糊的流量，导致模糊检测效果不佳

算法原理

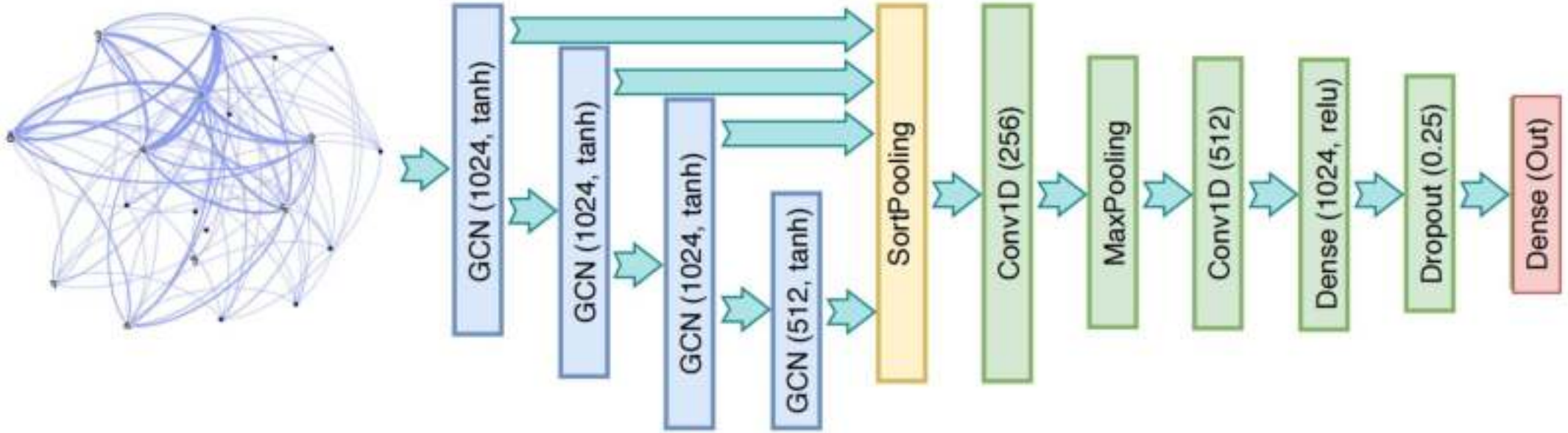
算法原理 MAppGraph

T	加密移动流量的应用程序分类
I	加密移动流量
P	1.收集应用程序各个时间段的流量组成流量块 2.加密移动流量块提取特征 3.构建加密移动流量的通信行为图 4.使用深度图卷积神经网络学习应用程序的通信行为 5.对应用程序类进行快速分类
O	加密移动流量所属应用程序

P	分析加密移动流量所属应用程序类别
C	应用程序连接的服务不完全相同
D	流量通信行为的模式构建
L	ACSAC 2021 (CCF-B类会议)

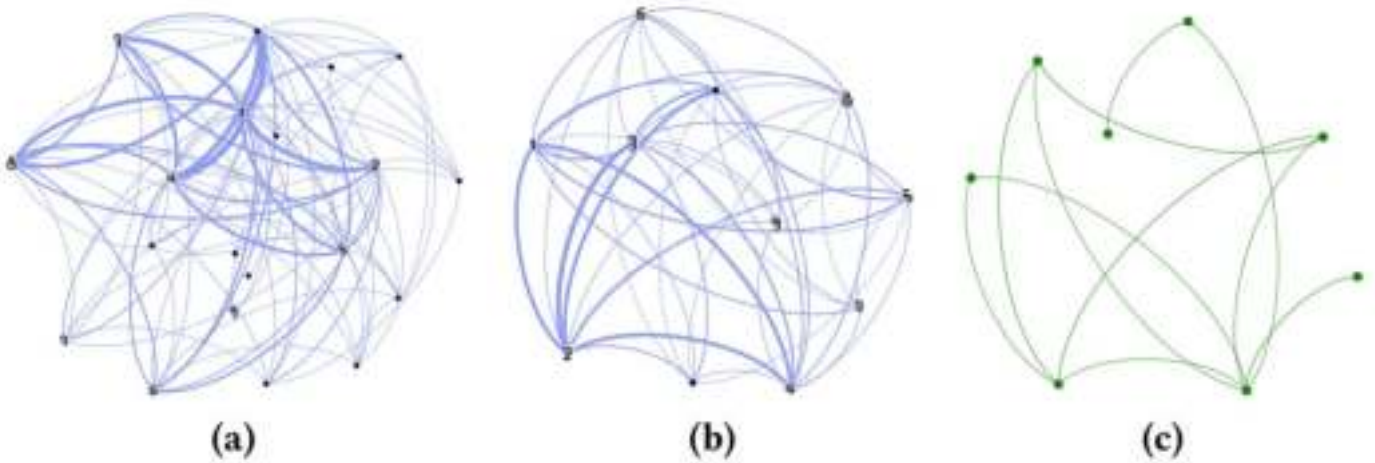
算法原理 MAppGraph

- 基本框架
  - 流量行为图构建
  - 深度图卷积神经网络学习
  - 输出加密移动流量所属程序类别



算法原理 MAppGraph

- 流量行为图构建
  - 节点的定义
    - 仅仅通过IP地址定义图的节点时，许多应用程序共享第三方服务，会导致应用程序的流量图在节点数相同时十分相似
    - 通过IP地址和端口的元组来定义一个节点
  - 边的构建
    - 利用时间相关性
  - 节点特征的提取



算法原理 MAppGraph

- 流量行为图构建
  - 时间相关性
    - 给定在时间窗口上捕获的流量
    - 将时间窗口划分为片段，用T表示片段的数量
    - 如果某个节点i向另一个节点发送或者接收一个分组，则认为在这个片段内此节点活跃，$ai(t) = 1$，否则为0
    - 节点i和节点j的时间相关性被定义为：$C_{i,j} = \sum_{t=1}^{T} ai(t) \cdot aj(t)$

(a) (b) (c)

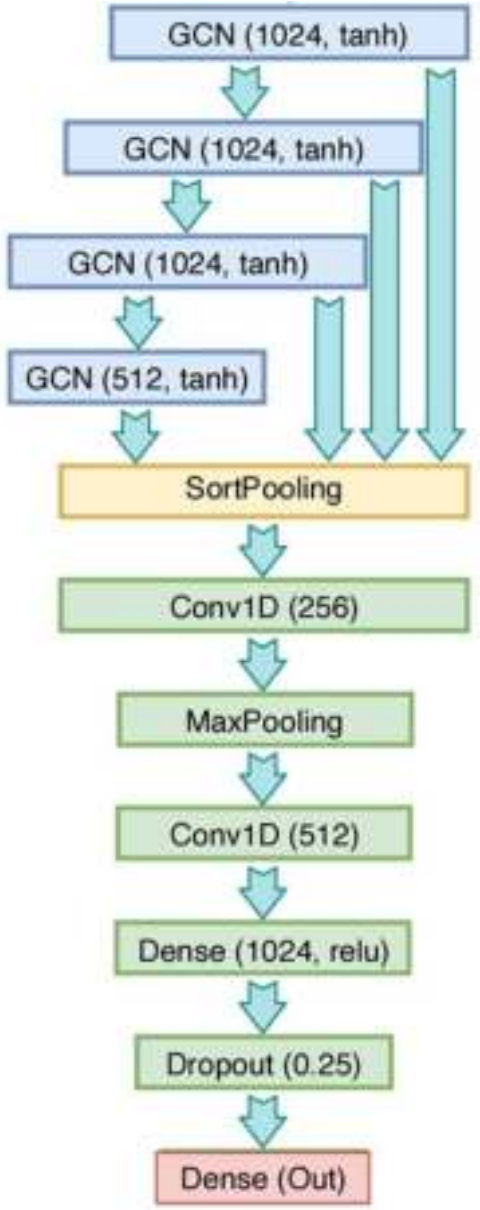
算法原理 MAppGraph

- 流量行为图构建
  - 节点特征提取
    - 从流量中提取和衍生出63维特征，将这些特征分为四类
    - 聚合特征：基本特征的综合、总计数、最大值等
    - 时间特征：流持续时间、流持续时间的平均值和标准差
    - 统计特征：数据包大小数量的中位数、平均值等
    - 分类特征：传输协议、ip地址等被分解并便准化为多个特征

No.	Feature	Category
1	Number of incoming packets	Aggregate
2	Max of incoming packet size	Aggregate
3	Min of incoming packet size	Aggregate
4	Mean of incoming packet size	Statistical
5	MAD of incoming packet size	Statistical
6	Stand. dev. of incoming packet size	Statistical
7	Variance of incoming packet size	Statistical
8	Skew of incoming packet size	Statistical
9	Kurtosis of incoming packet size	Statistical
10-18	10 – 90 percentile of incoming packet size	Statistical
19-37	Features 1-18 for outgoing packets	
38-56	Features 1-18 on both types of packets	
57	Mean flow size (in bytes)	Statistical
58	Mean flow duration (in seconds)	Temporal
59	Stand. dev. of flow duration (in seconds)	Temporal
60	Total number of flows	Aggregate
61	Mean number of packets in each flow	Statistical
62	Transmission protocol	Categorical
63	IP address of the node	Categorical

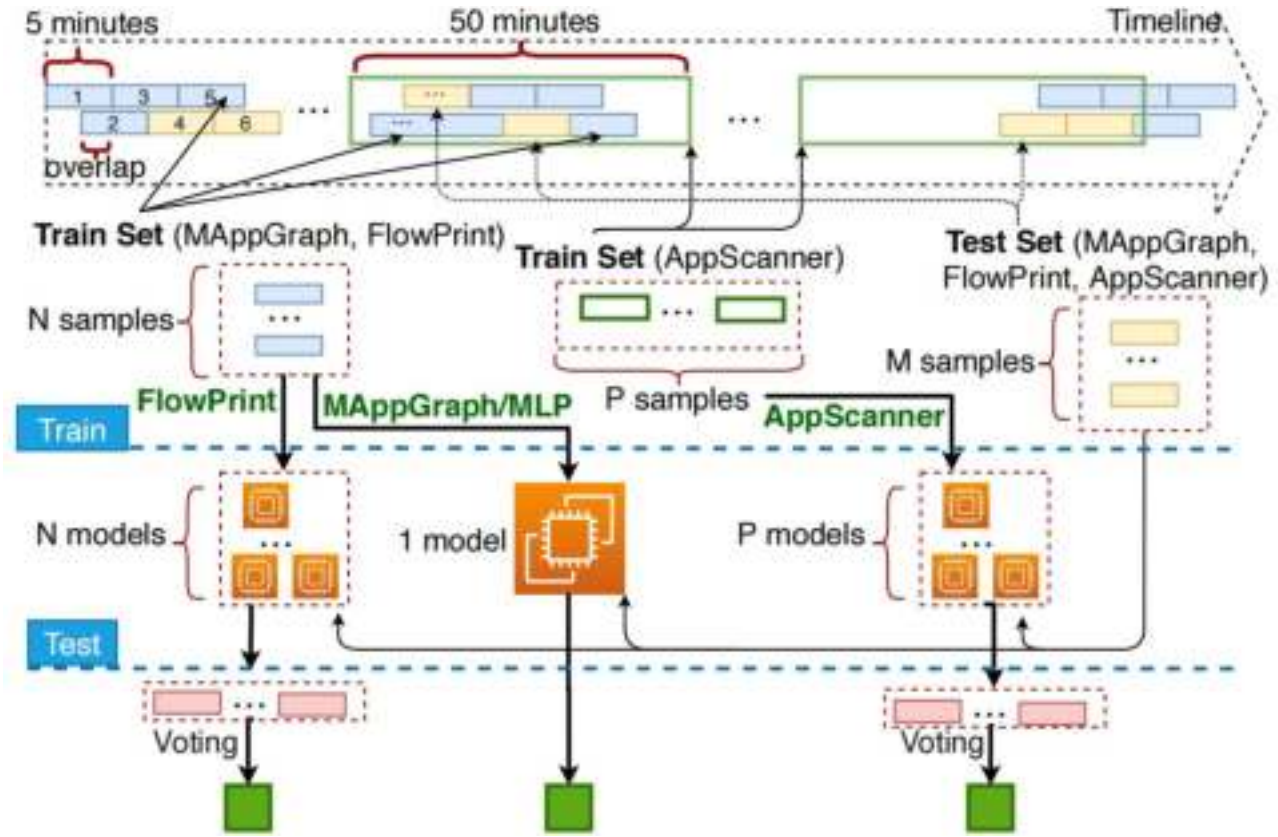
算法原理 MAppGraph

- 深度图卷积神经网络DGCNN
  - 图卷积层
    - 提取有用的特征以及图结构来描述图中的丰富信息
  - SortPooling层
    - 以有意义且一致的顺序读取图的顶点，对顶点特征进行排序
  - 池化层
    - 将图的潜在表示形式截断或者扩展到预定义的大小
  - 全连接层



算法原理 实验结果

- 数据收集
  - AppScanner
    - 5分钟的流量窗口
  - Flowprint
    - 50分钟的流量捕获窗口
  - MAppGraph
    - 5分钟的流量捕获窗口，但是获取应用程序各个时间段的数据，以构成图的集合



算法原理 实验结果

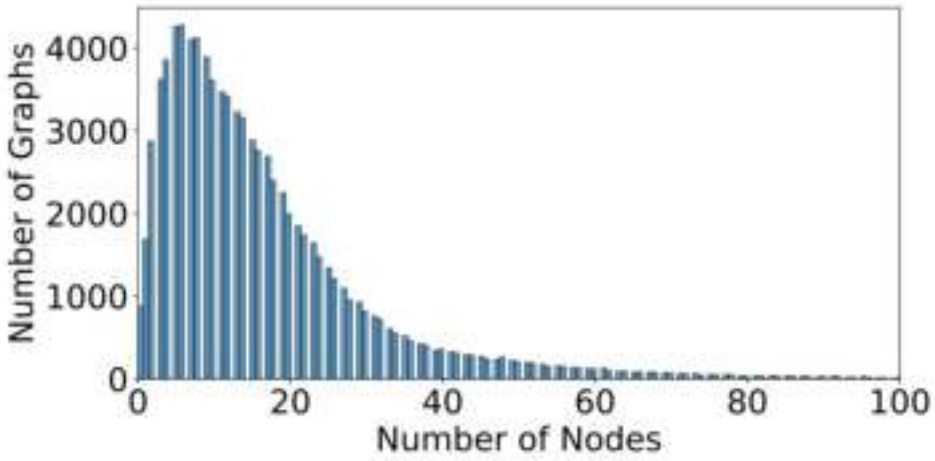
- 对比结果
  - AppScanner
    - 效果最差，按照流进行应用程序分类可能并不是一个有效的办法
  - MLP
    - 只考虑节点的特征，不考虑节点与边的相关性
  - FlowPrint
    - 只使用了5分钟的流量，程序的流量行为不完整
  - MAppGraph
    - 多个流量窗口训练，考虑应用程序多样性行为

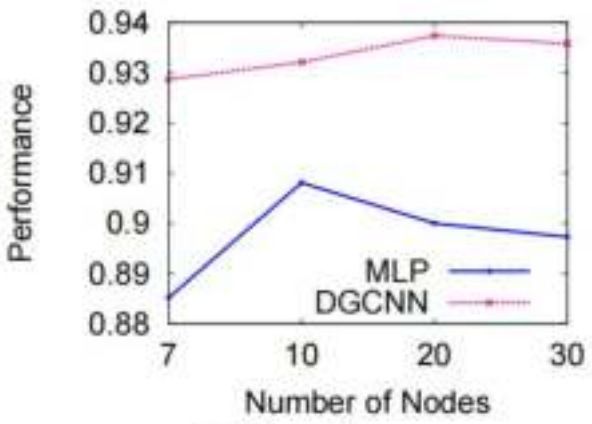
Technique	Precision	Recall	F1-Score	Accuracy
MLP	0.9081	0.9075	0.9074	0.9075
Enhanced AppScanner	0.8634	0.7938	0.7828	0.7938
Enhanced FlowPrint	0.8759	0.8341	0.8275	0.8341
MAppGraph	**0.9364**	**0.9346**	**0.9347**	**0.9346**

Model No.	Precision	Recall	F1-Score	Accuracy
1	0.7527	0.6660	0.6468	0.6660
2	0.7476	0.6483	0.6304	0.6483
3	0.7791	0.6757	0.6556	0.6757
4	0.7405	0.6468	0.6299	0.6468
5	0.7779	0.6577	0.6460	0.6577
6	0.7358	0.6652	0.6479	0.6652
7	0.7281	0.6493	0.6308	0.6493
8	0.7354	0.6559	0.6398	0.6559
9	0.7322	0.6485	0.6323	0.6485
10	0.7590	0.6515	0.6381	0.6515
11	0.7666	0.6515	0.6418	0.6515
12	0.7306	0.6527	0.6419	0.6527
13	0.7873	0.6563	0.6571	0.6563
14	0.7618	0.6415	0.6357	0.6415
15	0.7420	0.6346	0.6258	0.6346
16	0.7592	0.6633	0.6534	0.6633
Mean	0.7522	0.6541	0.6408	0.6541
Stan. Dev.	0.0187	0.0101	0.0097	0.0101
Enhanced AppScanner	**0.8634**	**0.7938**	**0.7828**	**0.7938**

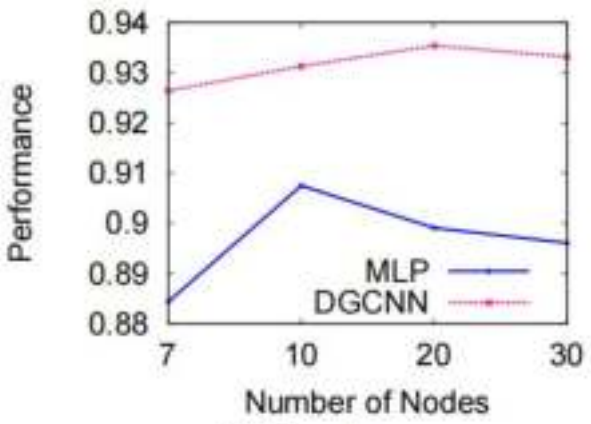
算法原理 实验结果

- 影响因素
  - 构成图的节点数量对分类效果的影响
    - 池化层会根据预定义的节点数量对流量图的潜在表示形式进行扩展或截断

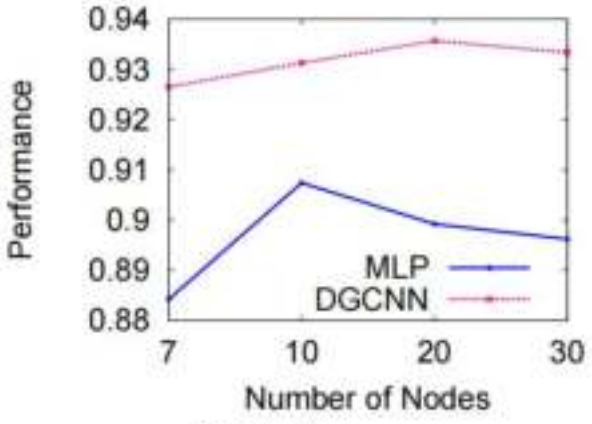




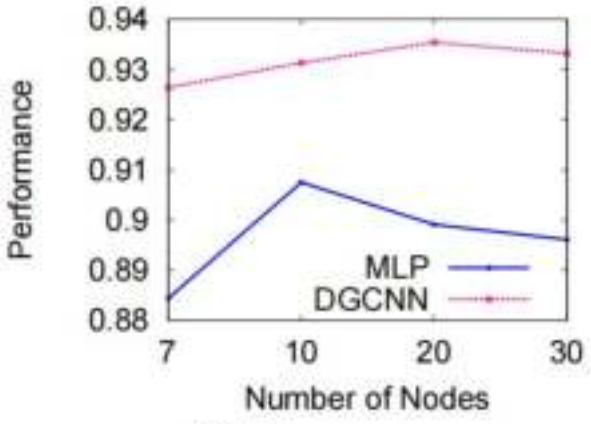
(a) Precision.



(b) Recall.



(c) F1-Score.



(d) Accuracy.

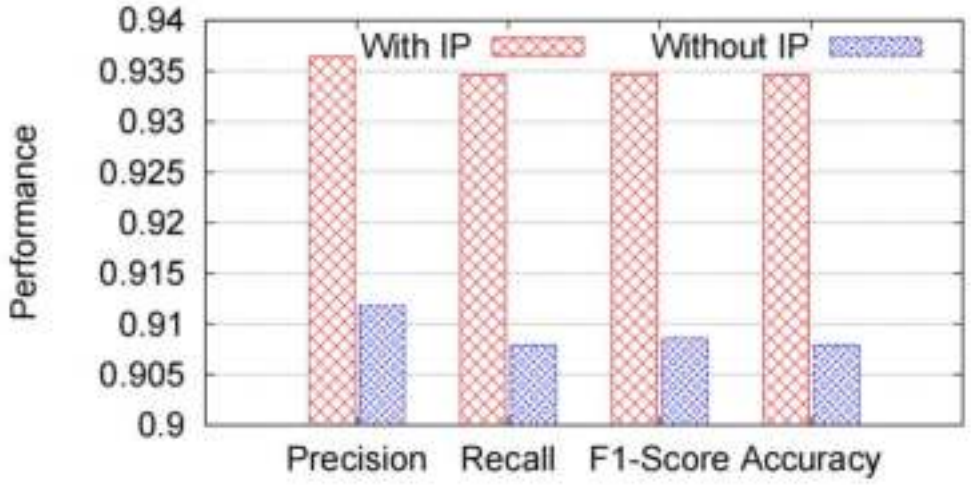
算法原理 实验结果

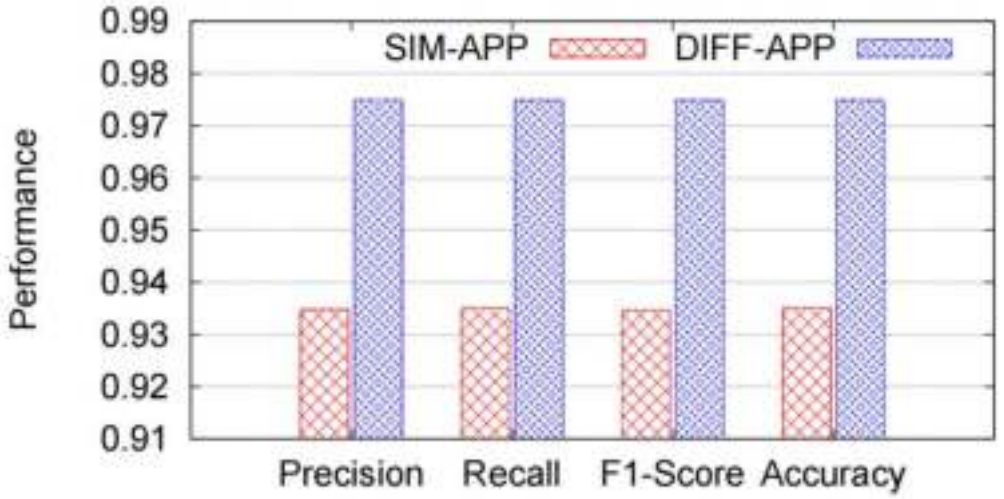
- 影响因素
  - 流量收集时间窗口：分类性能下降但是检测速度变快，性能的损失可以接受（可设置）
  - 片段持续时间：片段持续时间越短，图中的边越少

t_{slice}	Technique	Precision	Recall	F1-Score	Accuracy
1	Enhanced FlowPrint	0.7773	0.7325	0.7176	0.7325
	MAppGraph	**0.9342**	**0.9327**	**0.9328**	**0.9327**
5	Enhanced FlowPrint	0.8414	0.8112	0.8065	0.8112
	MAppGraph	**0.9353**	**0.9335**	**0.9337**	**0.9335**
10	Enhanced FlowPrint	0.8759	0.8341	0.8275	0.8341
	MAppGraph	**0.9364**	**0.9346**	**0.9347**	**0.9346**

T_{window}	Technique	Precision	Recall	F1-Score	Accuracy
5	MLP	0.9081	0.9075	0.9074	0.9075
	Enhanced FlowPrint	0.8759	0.8341	0.8275	0.8341
	MAppGraph	**0.9364**	**0.9346**	**0.9347**	**0.9346**
4	MLP	0.8905	0.8894	0.8896	0.8894
	Enhanced FlowPrint	0.8559	0.8296	0.8258	0.8296
	MAppGraph	**0.9181**	**0.9174**	**0.9171**	**0.9174**
3	MLP	0.8705	0.8671	0.8681	0.8671
	Enhanced FlowPrint	0.8546	0.8175	0.8179	0.8175
	MAppGraph	**0.8977**	**0.8932**	**0.8935**	**0.8932**
2	MLP	0.8648	0.8625	0.8631	0.8625
	Enhanced FlowPrint	0.8447	0.7945	0.7978	0.7945
	MAppGraph	**0.8759**	**0.8738**	**0.8734**	**0.8738**
1	MLP	0.8478	0.847	0.8466	0.8470
	Enhanced FlowPrint	0.7785	0.6535	0.6711	0.6535
	MAppGraph	**0.8698**	**0.8683**	**0.8679**	**0.8683**

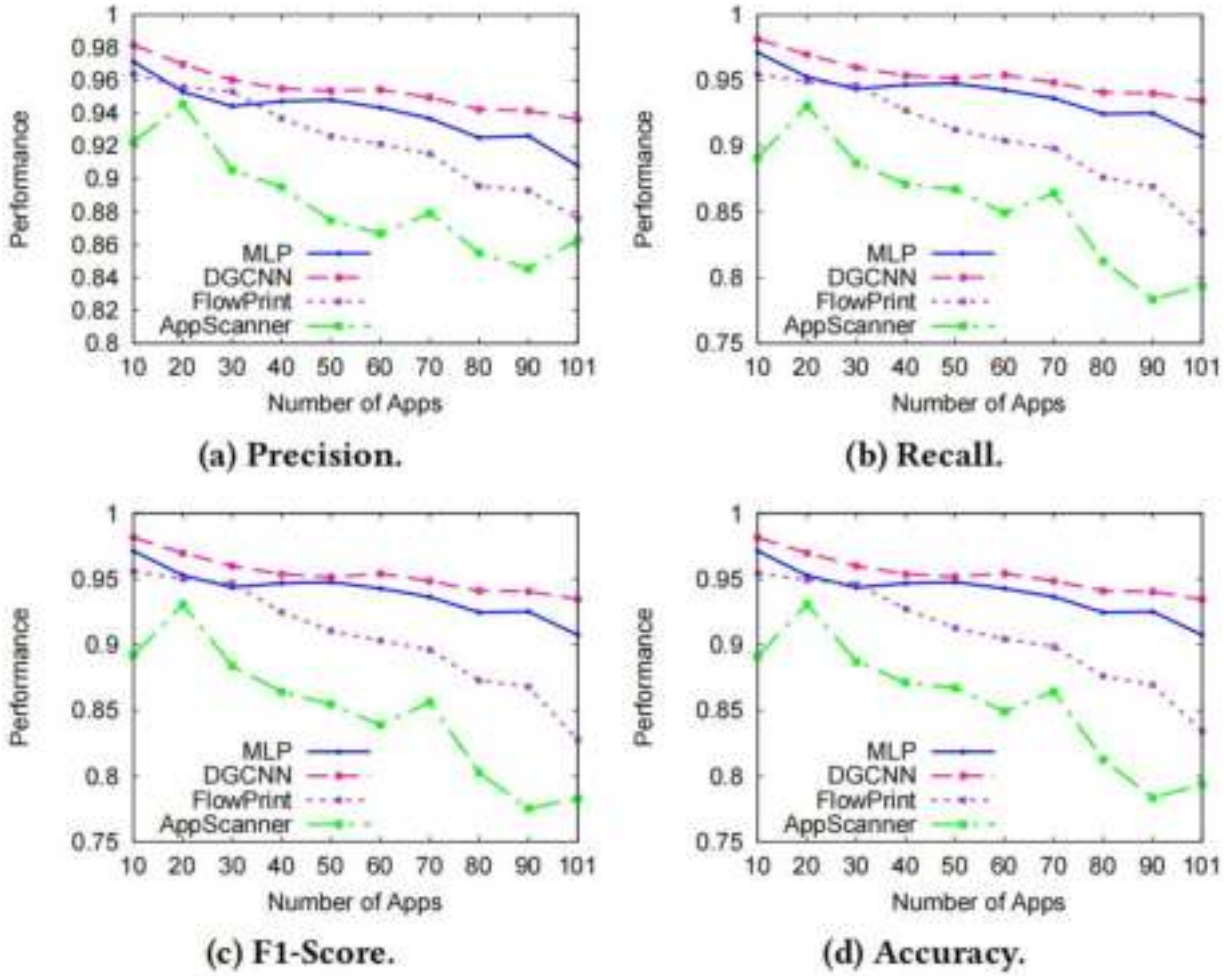
- 影响因素
  - 特征向量中是否含有IP地址：训练时不使用IP地址，性能略有下降，但是可以在目标服务的不同网络域中部署，而不需要重新训练模型
  - 数据中是否具有类似应用程序：通过考虑应用程序使用的服务之间的相关性，并结合图节点特征属性，可以更好地区分类似程序





算法原理 实验结果

- 影响因素
  - 训练时APP数量的影响
    - APP数量越多，越容易出现功能行为类似的APP
    - AppScanner受模糊流量影响较大，出现功能类似的APP时，性能下降较多
    - MAppGraph性能下降较少



算法原理 MAppGraph

- 优势
  - 流量通信行为图更好地代表了移动应用程序的通信行为
  - 引入应用程序使用服务的相关性，并结合流量的特征属性，很好地区分应用程序，即使功能类似

- 劣势
  - 对于使用服务数目相同且功能相同的应用程序的分类效果较差

应用总结

应用总结

- 应用
  - 攻击者可以通过分析移动网络的流量来了解用户的行为，定制个性化的钓鱼攻击
  - 相关机构分析移动网络流量监控非法行为，帮助发现用户的行为模式
  - 网络管理员通过分析移动网络流量，进行网络规划和管理

- 拓展方向
  - 提高流量覆盖率，构建更能体现用户真实行为的移动应用程序流量数据集，进一步分析各个因素影响分类器的性能
  - 采用无监督学习方法，例如使用图聚类，减少构建标注数据集的工作量
  - 提取流量通信行为模式，进一步减弱内容交付网络CDN、共享三方库产生的类似流量对分类的影响

参考文献

[1] Taylor V F , Spolaor R , Conti M , et al. AppScanner: Automatic Fingerprinting of Smartphone Apps from Encrypted Network Traffic.[C]// IEEE. IEEE, 2016.

[2] Taylor V F , Spolaor R , Conti M , et al. Robust Smartphone App Identification Via Encrypted Network Traffic Analysis[J]. IEEE Transactions on Information Forensics and Security, 2018, 13(1):63-78.

[3] Pham T D, Ho T L, Truong-Huu T, et al. MAppGraph: Mobile-App Classification on Encrypted Network Traffic using Deep Graph Convolution Neural Networks[C]//Annual Computer Security Applications Conference. 2021: 1025-1038.

[4] https://www.cnblogs.com/SivilTaram/p/graph_neural_network_2.html

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。