

引用格式:陈镭,杨章静,黄璞.融合特征选择的 Android 恶意逃避攻击研究 [J]. 微电子学与计算机,2021,38(3):89-94. [CHEN L, YANG Z J, HUANG P. Research on android malicious evasion attack based on feature selection[J]. Microelectronics & Computer, 2021, 38(3): 89-94.]

融合特征选择的 Android 恶意逃避攻击研究

陈 镭^{1,2}, 杨章静¹, 黄 璞¹

(1 南京审计大学 信息工程学院, 江苏 南京 211815;

2 计算机软件新技术国家重点实验室(南京大学), 江苏 南京 210023)

摘 要: 机器学习系统以其强大的自适应性、自学习能力,越来越多的应用到 Android 恶意软件检测领域,取得了显著的检测效果.然而,机器学习算法和样本本身还面临着诸多安全威胁,一些经过精心策划的攻击,希望颠覆这些算法并允许恶意行为对抗检测.首先以 Drebin 系统为例介绍了基于机器学习的 Android 恶意软件检测方法的原理,然后在攻击目标、攻击策略的基础上提出了针对机器学习分类器的逃避攻击模型.在综合考虑特征权重、可修改性、修改成本的基础上,提出了一种恶意对抗样本生成方法.实验结果表明,只需要修改很少量的特征,就能够逃避线性 SVM 分类器的检测,最后用一个具体的恶意样本逃避实例验证了提出方法的有效性.

关键词: 机器学习;支持向量机;安卓恶意软件;逃避攻击;特征重要性

中图分类号: TP311

文献标识码: A

文章编号: 1000—7180(2021)03—0089—06

Research on android malicious evasion attack based on feature selection

CHEN Lei^{1,2}, YANG Zhang-jing¹, HUANG Pu¹

(1 School of Information Engineering, Nanjing Audit University, Nanjing 211815, China;

2 State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China)

Abstract: With its strong self-adaptability and self-learning ability, machine learning systems are increasingly used in the field of Android malware detection and have achieved remarkable detection results. However, machine learning algorithms and samples themselves also face many security threats, some well-designed attacks, hoping to subvert these algorithms and allow malicious behavior to resist detection. Firstly, the principle of machine learning-based detection method is introduced by taking the Drebin system as an example, and then an evasion attack model for machine learning classifier is proposed based on the attack target and attack strategy. Based on the comprehensive consideration of feature weight, modifiability and modification cost, proposes a malicious countermeasure sample generation method. The experimental results show that only a few features need to be modified to evade the detection of the linear SVM classifier. Finally, a specific example of malicious sample evasion attack verifies the effectiveness of the proposed method.

Key words: machine learning; SVM; android malware; evasion attack; feature importance

收稿日期: 2020—07—02; 修回日期: 2020—07—27

基金项目: 国家自然科学基金(U1831127);物联网产业化与智能生产协同创新中心(闽江学院)基金(IIC1705);南京审计大学
高教研究课题(2020JG049);江苏省高校“青蓝工程”优秀青年骨干教师培养对象资助

1 引言

近年来机器学习技术越来越多的应用于 Android 恶意软件检测领域^[1-3]. 在这些系统中, 基于不同的特征表示(如权限、系统调用图、动态行为或 Api 调用), 采用各种分类方法, 如支持向量机、随机森林和深层神经网络等, 用于模型构建来检测恶意应用程序. 虽然这些技术在 Android 恶意软件检测方面提供了无与伦比的灵活性, 但攻击者利用获得的一些先验知识、观察的组合, 积极地操纵数据误导分类器^[4-5](例如, 针对在线更新系统, 通过改变数据分布或特征重要性, 最大限度地逃避和欺骗分类器的检测, 将恶意软件误分类为良性), 使得机器学习方法本身成为攻击对象. 目前, 在对抗环境中理解机器学习安全性的问题开始被重视^[6]. 攻击和防御总是一场永无休止的竞赛, 在每一轮迭代中, 他们试图分析彼此的方法和弱点, 并制定自己的最佳策略以克服对手^[7].

本文针对基于机器学习的 Android 恶意软件检测系统展开逃避攻击^[8-9]问题进行研究. 以近似的 Drebin^[1] 系统为攻击目标系统, 在分析攻击目标、攻击能力、攻击策略的基础了提出了针对机器学习分类器的逃避攻击模型. 和以往大多对抗样本生成方法类似, 采用修改特征向量而非直接修改源代码的方式, 模拟攻击者制作对抗样本, 逃避分类系统的检测. 为了保证原有的恶意功能不受影响, 先前的方法^[10]往往忽略特征消除的场景, 只考虑增加特征的方式来修改特征向量. 本文在综合考虑特征权重、可修改性(增加和消除)、修改成本的基础上, 提出了一种恶意对抗样本生成方法. 并且根据此方法生成的恶意软件可以实际运行起来而不会影响其原有的恶意功能.

2 Drebin 系统介绍

2.1 数据集

Drebin 收集 2010 年 8 月至 2012 年 10 月期间的大量公开 App 样本. 通过将每个应用上传到 VirusTotal 进行标记, 如果被 10 个常用的反病毒工具中的 2 个及以上识别为恶意, 则标记为恶意应用. Drebin 的最终数据集中共包含 123 453 个良性应用、5 560 个恶意应用.

2.2 特征向量

提取特征字符串. Drebin 从 Manifest 文件和 Dex 文件中尽可能多的提取静态特征, 通过向每类

特征中的字符串添加唯一前缀, 确保属于不同类别的特征元素不会发生冲突, 最终构成由 8 种不同类型特征字符串组成的集合 S , Drebin 在其数据集中提取的特征字符串超过 545 000 个.

$$S = S_1 \cup S_2 \cup \dots \cup S_8$$

构建特征向量. Drebin 定义了一个 $|S|$ 维的向量, 其中每个维度的值取为 0 或是 1. 对于每一个单独的应用软件 x 来说, 是通过构造一个特征向量 $\varphi(x)$ 来对其进行描述, 这样对于从 x 中提取的特征字符串 s , 若 s 出现在 S 中, 特征向量相应的维度设置为 1, 否则为 0. 形式上, 如

$$\varphi: X \rightarrow \{0, 1\}^{|S|}, \varphi(x) \mapsto (I(x, s))_{s \in S},$$

$$I(x, s) = \begin{cases} 1, & x \text{ 含有特征 } s \\ 0, & x \text{ 不含特征 } s \end{cases}$$

2.3 分类算法

一旦 Android 应用被表示为特征向量, Drebin 使用线性 SVM 分类器区分良性和恶意样本. 线性分类器一般表示为线性函数, 如公式(1):

$$f(x) = w^T x + b \quad (1)$$

分类器最终会在特征空间中识别出一个超平面, 将两种类别的样本最大距离的分开. 其中 $w \in \mathbb{R}^{|S|}$ 为特征权重的向量, 代表了超平面的方向, w 的每个元素都可以解释为相应功能的权重, b 为调节参数. 这些参数在训练过程中不断的调整优化获得.

$$f(x) = \sum_{s \in S} I(x, s) \cdot w_s = \sum_{s \in x} w_s \quad (2)$$

对于每个待检测的应用 x , 如公式(2) 所示, 都是通过计算 $f(x)$ 得到, 给定临界值 t , 如果 $f(x) > t$, 分类为恶意软件, $f(x) \leq t$ 则分类为正常软件.

3 逃避攻击模型

3.1 攻击目标

逃避攻击最常见的攻击场景, 恶意软件作者通常希望在不损害恶意功能的情况下, 逃避机器学习分类器的检测, 误导分类器将恶意应用识别为良性应用. 相对的情况是将良性应用错误的分类为恶意应用, 虽然也存在这种情况, 但攻击者一般没有太大兴趣, 因此暂不讨论此种情况, 本文的攻击目标是将恶意应用误分类为良性应用. 即给定待检测的恶意应用 x , $f(x) > t$, 逃避攻击的目标是找到一个尽可能小的扰动 δ , 使得 $f(x') = f(x + \delta)$ 的分类结果不同于原来的结果, 即 $f(x') \leq t$.

3.2 攻击策略

本文提出的攻击策略包括以下 2 个部分:

(1)复制目标分类器,得到恶意分类中每个特征的重要性. 攻击者敌对能力的大小首先取决于对敌对知识掌控的程度. 攻击者可能完全、部分或完全不了解分类器的训练数据 D、特征集合 F、学习算法 A. 攻击能力不同的攻击者,成功实施攻击的差别主要在复制目标分类器这个步骤. 例如,在具有完整知识的情况下,即 DFA,可以简单地复制目标系统. 在不能够访问原始训练数据的情况下,包括 FA、F、A 和 NONE,需要生成与原始数据类似的替代训练集. 在不能够获取原始特征的情况下,包括 DA、D、A 和 NONE,需要构造与原始特征类似的替代特征集(Android 恶意软件检测领域, Permission 和 Api 这两类特征最常用). 在不了解的对手目标分类器使用的分类算法的情况下,包括 DF、D、F 和 NONE,需要选择近似于原始分类器的替代算法.

(2)从目标恶意应用中提取特征功能集,消除顶级恶意特征或者增加顶级良性特征. 从公式(2)中不难看出,单个特征 f_i 对分类的贡献度与其特征权重 w_i 的大小正相关, w_i 的值可以从训练好的模型参数中得到,按其值大小由高到低进行排序. 在恶意软件检测领域,应用程序中提取特征功能集被映射为特征向量 $\varphi(x)$,其特征值由离散的数值 0 或 1 构成. 通过迭代的消除顶级恶意特征(特征值由 1 变为 0)或增加顶级良性特征(特征值由 0 变为 1)达到攻击目的.

4 对抗样本生成方法

攻击者成功实施逃避攻击的关键在于能够找到那些关键特征,在能力允许的范围內对其进行修改,并保证修改后的应用程序能够正常运行,且不容易被检测系统发现. 本节在综合考虑以上因素的情况下,在分析特征可修改性、修改成本的基础上,提出了一种对抗样本生成方法.

4.1 可修改性分析

修改特征的方式一般包括特征添加和特征消除. 对 Android 应用程序文件进行修改并不是简单的在样本中加入一些噪声就可以了,在图像中加入少许噪声后并不影响人们对图像的识别,但是对应用程序文件修改后很有可能会导致程序崩溃甚至无法安装运行. 表 1 给出了 Drebin 提取的 8 类不同特征集的可修改性.

特征添加:通常使用注入的方式添加特征,一般可以保留原始恶意程序的语义和入侵功能,比特征

表 1 Drebin 特征集的可修改性

类别	描述	添加	方式	消除	方式
S_1	Hardware components	是	注入	否	
S_2	Requested permissions	是	注入	否	
S_3	Application components	是	注入	是	改名
S_4	Filtered intents	是	注入	否	
S_5	Restricted API calls	是	注入	是	反射加密
S_6	Used permission	是	注入	否	
S_7	Suspicious API calls	是	注入	是	反射加密
S_8	Network addresses	是	注入	是	加密

消除更不易被察觉. 例如,可以在 Manifest 文件中添加权限,在 Dex 文件中添加字符串、注入 Api 调用特征等. 在不会被执行的源代码文件中注入特征很容易被程序执行图发现,语法分析阶段就能够被识别,迫使攻击者不得不向可执行的程序中注入特征,这就需要攻击者考虑更多、更严格的约束条件.

特征消除:特征消除更复杂的操作,攻击者不能消除 Hardware 组件、Permission 和 Intent 特征,因为这可能会限制应用程序的功能. 除入口组件外,一些 Application 组件的名称可以修改,但必须保证 Dex 文件中相应名称同步修改. 对于消除 Api 和 Network addresses 特征,攻击者尝试通过加、解密的方式实现隐藏的目的,但可能会引入其他可疑特征.

4.2 修改成本

在实际场景中,不损害程序功能的情况下操纵大量的特征是不太可行的. 定义为特征功能集中改变某一个特征元素相应的成本.

对于攻击者,每个特征的操作成本 c_i 是不同的, c_i 主要和特征类型(Permission、Api)、特征修改方式(增加、消除)等因素有关. 文献[11]给出了由经验得出的几种不同类别特征的归一化处理后的修改成本. 除表 1 中列举的不能被消除的特征外,本文假定修改 Manifest 文件中 S_1 - S_4 类特征的成本(添加为 0.5、消除为 0.95)的和 Dex 文件中 S_5 - S_8 类特征的成本(添加为 0.4、消除为 0.7).

而攻击者为逃避检测,修改特征需要花费的总成本可表示为:

$$c(x) = \sum_{i=1}^n c_i$$

(3)

假设攻击者拥有可用于逃避攻击的资源上限为 $c_{\max}$,若 $c(x) > c_{\max}$,则攻击自动终止,攻击任务失败.

4.3 逃避算法

特征的权重越大,特征被操纵的成本越低,特征越重要.单个特征的重要性可定义如下,其中 λ 为调节参数:

$$I_i = \lambda \cdot \frac{|w_i|}{c_i}$$

(4)

越重要的特征意味着攻击者在逃避攻击中操纵它的可能性越大.相反的,如果想要构造一个更安全的分类器,其原理就是减少这些重要特征被选择用于模型构建的可能性,迫使攻击者去操纵其他不太重要的特征.以特征重要性为依据,本节提出一种恶意对抗样本生成算法,步骤大致如下:

输入:恶意 App 样本特征向量 $\varphi(x)$,参数 λ ,并且根据公式(2), $f(x) > t$

Step1:复制目标分类器,提取特征,训练模型,获得模型为每个特征指定的权重 w_i ;

Step 2:根据特征类别、修改方式等,确定每个特征的修改成本 c_i ;

Step 3:根据公式(4) 计算每个特征的重要性 I_i ,按 I_i 从高到低排序;

Step 4:通过向特征向量中添加 I_i 最大的良性特征或消除 I_i 最大的恶意特征;

Step 5:while $f(x') > t$ 或 $c(x' - x) \leq c_{\max}$ do
1:执行步骤 4;

2:根据公式(2) 计算 $f(x')$,根据公式(3) 计算 $c(x' - x)$;

end while

输出:对抗样本特征向量 $\varphi(x')$.

5 实验验证

5.1 构建目标系统

替代训练集:Drebin 公开的数据集中共包含 123 453个良性应用、5 560个恶意应用,共129 013个样本,由于良性应用和恶意应用的比例相差过大,恶意应用的数目只占良性应用不到 5%的比列,造成样本类别分布偏差过大,我们随机抽取良性应用的 1/10,即 1 2 346个良性应用和全部的5 560个恶意应用,总共17 906个样本的数据集.按照和 80%、20%的比列划分训练集和测试集,最终训练集包含 14 324个样本,测试集包含3 582个样本.

替代特征集:Drebin 原始样本过多且不均衡,特征集合异常庞大,总计超过 54 万维,这些特征占用了大量分类器的训练时间.本文从17 906个样本的数据集中,提取特征共计100 234个,加上标记位,

组成100 235维的特征向量.

替代分类器:由于 Drebin 没有公开其使用的 SVM 分类器细节,考虑到特征数和样本数比较大,我们使用 libLinear 分类器作为替代分类器,为了保证分类结果的稳定性,采用十折交叉验证的方式进行训练.本文克隆的分类器,分类效果如表 2 所示.恶意样本分类的真阳率(TPR)为 96.9%,略高于 Drebin 本身 93.9%的 TPR,恶意样本分类的假阳率(FPR)为 1%,和 Drebin 本身 1%的 FPR 保持一致,整体准确率(Accuracy) 98.3%,略低于 Drebin 本身 99.2%的 Accuracy.

表 2 替代分类器和 Drebin 检测率对比情况

	Drebin	分类器 1	分类器 2
数据集	129 013	17 906	17 906
特征集	545 000	100 234	100 234
分类器	SVM	SVM(L1)	SVM(L2)
TPR(malware)/%	93.9	96.9	96.5
FPR(malware)/%	1	1	1.6
Accuracy/%	99.2	98.3	97.7

构建逃避攻击目标集:本文研究的目标是逃避分类器检测,因此目标集合中的样本应满足以下两个基本条件:(1)不能出现在分类器的训练集中.(2)样本本身为恶意应用,而且被分类器正确识别为恶意应用.替代测试集共包含3 582个样本,其中标记为恶意样本个数为1 132,被分类器正确识别的为恶意样本为1 092个,最终这1 092个恶意样本作为本文逃避攻击的目标.

5.2 实验效果

5.2.1 特征权重对分类的影响

我们使用 libLinear 分类器作为替代分类器.实验中分别在损失函数中添加 L_1 正则化项或 L_2 正则化项的方式对 SVM 分类器进行训练,得到的模型权重 w_i 最大的 5 个良性特征或恶意特征,详细情况如表 3 所示.

表 3 分类器的特征权重 Top5

训练方式	类别	Top1	Top2	Top3	Top4	Top5
L_1 正则化	Malware+	3.03	2.78	2.35	2.3	2.18
	Benign-	3.03	2.27	1.59	1.57	1.56
L_2 正则化	Malware+	1.81	1.71	1.67	1.61	1.44
	Benign-	1.22	1.13	0.97	0.96	0.95

从表 3 中可以看出, L_1 正则化得到的特征权重值更离散,而 L_2 正则化得到的特征权重值更集中.最初的模型对1 097个恶意样本的分类准确率均为

100%,在不考虑特行修改方式的情况下,分别对 L_1 正则化和 L_2 正则化训练出的模型排名前 5 的特征,仅仅改变 1 个特征的情况下,得到线性 SVM 分类器对恶意样本检测率的下降程度,如图 1 所示.

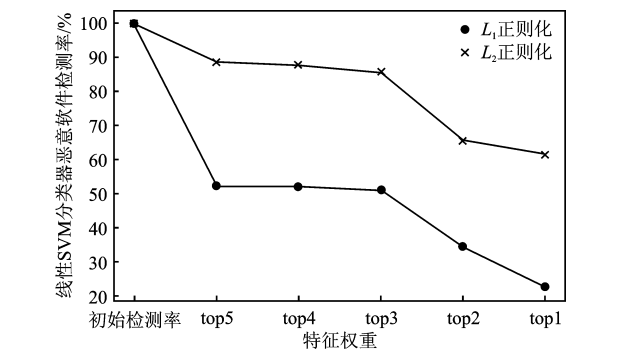


图 1 修改 1 个特征后分类器检测率

从图 1 可以看出,最坏的情况,攻击者仅仅修改了 L_1 正则化方式下 SVM 分类器排名第一的特征,其权重值 w 为 3.03,分类的检测率从 100% 迅速下降到 22.4%. 换句话说,如果想要构造一个更安全的线性 SVM 分类器来增加算法的鲁棒性,其原理就是减少这些权重大的特征被选择用于模型构建的可能性. 另外,采用 L_2 正则化的方式也会迫使攻击者花费更多的资源,操纵相对更多的特征.

5.2.2 验证对抗样本生成方法

实验中以 DroidKungFu 恶意家族一个具体恶意 App 样本为例,通过采用本文提出的方法修改特征,成功逃避目标分类器的检测. 采用 L_2 正则化的方式训练模型,得到恶意样本中权重排名前 10 的特征,如表 4 所示,其中类别+代表对恶意分类的贡献,类别-则是对良性分类的贡献.

表 4 恶意样本中特征权重 Top10

类别	权重	特征名
+	1.17	call::system/bin/su
-	0.98	feature::android.hardware.touchscreen
+	0.81	intent::android.intent.action.SIG_STR
+	0.52	call::getSubscriberId
-	0.47	activity:: activity
+	0.41	intent:: android.intent.action.BATTERY_CHANGED_ACTION
-	0.33	url::http://r2.adwo.com/ad
-	0.32	activity:: com.adwo.adsdk.AdwoAdBrowserActivity
+	0.31	api_call:: android/net/wifi/WifiManager; >setWifiEnabled
+	0.29	url::https://market.android.com

而模型总体特征权重排名前 30 的情况如下: (+1.81,+1.71,+1.67,+1.61,+1.44,+1.41,+1.37,-1.22,+1.2,+1.18,+1.17,+1.17,-1.13,+1.06,+1.06,+1.03,+1,+0.99,+0.99,-0.98,+0.98,+0.98,+0.98,-0.97,+0.96,-0.96,-0.95,+0.94,+0.94,+0.91),按照 3.3 节算法思想,按权重从高到底的顺序修改特征(添加或删除),权重排名前 7 的特征都为恶意类别,不能添加,因本样本中也没有这 7 个特征,所以也不能消除,故跳过对这 7 个特征的操作(以下遇到类似情况特征均按跳过处理),权重排名第 8 的特征为 service_receiver::collepepics.iconosys.eng.LocationSender,其权重为-1.22,良性类别,可添加此特征,为方便计算,公式 4 中 λ 取值为 1,可计算出此特征重要性简单记为 1.22/0.5. 权重排名第 9、10 的特征为跳过. 权重排名第 11 的特征为 call::system/bin/su,其权重为+1.17,恶意类别,可消除此特征,计算出此特征重要性为 1.17/0.7. 权重排名第 12 的特征为跳过. 权重排名第 13 的特征为 activity::Autopalya,其权重为-1.13,良性类别,可添加此特征,计算出此特征重要性为 1.13/0.5. 权重排名第 13-19 的特征为跳过. 权重排名第 20 的特征为 feature::android.hardware.touchscreen,其权重为-0.98,良性类别,因本样本中已有了这个特征,所以不能再添加,故跳过. 权重排名第 21-23 的特征为跳过. 权重排名第 24 的特征为 activity::APhoto 其权重为-0.97,良性类别,可添加此特征,计算出此特征重要性为 0.97/0.5. 权重排名第 25 的特征为跳过. 权重排名第 26 的特征为 activity::DoodleRunnerActivity,其权重为-0.96,良性类别,可添加此特征,计算出此特征重要性为 0.96/0.5.

故按照算法思想,根据特征重要性排序修改特征,首先修改权重排名第 8 的特征(重要性第 1),但添加此特征后,对抗样本仍然被分类器识别为恶意,在没有达到特征修改上限 c_{max} 的情况下,依次修改权重排名第 13(重要性第 2)、第 24(重要性第 3)和第 26(重要性第 4)的特征后,分类器成功被欺骗,完成逃避攻击.

6 结束语

本文针对基于机器学习的 Android 恶意软件检测系统展开逃避攻击问题开展研究. 为了保证原有的恶意功能不受影响,先前的方法往往忽略特征消除场

景,只考虑向特征向量中增加特征,修改特征时也没有区分修改成本.本文在综合考虑特征权重、可修改性、修改成本的基础上,提出了一种恶意对抗样本生成方法.实验部分证明了我们提出方法的有效性,本文中虽然只演示了攻击特定的分类器(线性 SVM),但逃避攻击的策略是通用的,对其他以机器学习为基础的分类器同样适用,包括逻辑回归、随机森林、神经网络等.至于主要以破坏模型可用性和完整性为目的,针对训练数据集进行的欺骗攻击(poisoning attack)问题,可作为我们下一步的研究方向.

参考文献:

- [1] ARP D, SPREITZENBARTH M, HUBNER M, et al. DREBIN: effective and explainable detection of android malware in your pocket[C]//NDSS Symposium. 2014. DOI: 10.14722/ndss.2014.23247.
- [2] HOU S F, YE Y F, SONG Y Q, et al. HinDroid: an intelligent android malware detection system based on structured heterogeneous information network[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Halifax, Nova Scotia, Canada: Association for Computing Machinery, 2017: 1507-1515. DOI: 10.1145/3097983.3098026.
- [3] 张蕾,崔勇,刘静,等.机器学习在网络空间安全研究中的应用[J].计算机学报,2018,41(9):1943-1975. DOI: 10.11897/SP.J.1016.2018.01943.
ZHANG L, CUI Y, LIU J, et al. Application of machine learning in cyberspace security research[J]. Chinese Journal of Computers, 2018, 41(9): 1943-1975. DOI: 10.11897/SP.J.1016.2018.01943.
- [4] 李盼,赵文涛,刘强,等.机器学习安全性问题及其防御技术研究综述[J].计算机科学与探索,2018,12(2):171-184. DOI: 10.3778/j.issn.1673-9418.1708038.
LI P, ZHAO W T, LIU Q, et al. Security issues and their countermeasuring techniques of machine learning: a survey[J]. Journal of Frontiers of Computer Science and Technology, 2018, 12(2): 171-184. DOI: 10.3778/j.issn.1673-9418.1708038.
- [5] SUCIU O, MaRGINEAN R, KAYA Y, et al. When does machine learning FAIL? Generalized transferability for evasion and poisoning attacks[C]//Proceedings of the 27th USENIX Conference on Security Symposium. Berkeley: USENIX Association, 2018: 1299-1316.
- [6] 张思思,左信,刘建伟.深度学习中的对抗样本问题[J].计算机学报,2019,42(8):1886-1904. DOI: 10.11897/SP.J.1016.2019.01886.
ZHANG S S, ZUO X, LIU J W. The problem of the adversarial examples in deep learning[J]. Chinese Journal of Computers, 2019, 42(8): 1886-1904. DOI: 10.11897/SP.J.1016.2019.01886.
- [7] CHEN L W, YE Y F, BOURLAI T. Adversarial machine learning in malware detection: arms race between evasion attack and defense[C]//2017 European Intelligence and Security Informatics Conference (EISIC). Athens, Greece: IEEE, 2017: 99-106. DOI: 10.1109/EISIC.2017.21.
- [8] ABAID Z, KAAFAR M A, JHA S. Quantifying the impact of adversarial evasion attacks on machine learning based android malware classifiers[C]//2017 IEEE 16th International Symposium on Network Computing and Applications (NCA). Cambridge, MA, USA: IEEE, 2017: 1-10. DOI: 10.1109/NCA.2017.8171381.
- [9] SRNDIC N, LASKOV P. Practical evasion of a learning-based classifier: a case study[C]//2014 IEEE Symposium on Security and Privacy. San Jose, CA, USA: IEEE, 2014: 197-211. DOI: 10.1109/SP.2014.20.
- [10] 唐川,张义,杨岳湘,等. DroidGAN: 基于 DCGAN 的 Android 对抗样本生成框架[J].通信学报,2018,39(Z1):64-69. DOI: 10.11959/j.issn.1000-436x.2018178.
TANG C, ZHANG Y, YANG Y X, et al. DroidGAN: android adversarial sample generation framework based on DCGAN[J]. Journal on Communications, 2018, 39(Z1): 64-69. DOI: 10.11959/j.issn.1000-436x.2018178.
- [11] CHEN L W, HOU S F, YE Y F, et al. SecureDroid: enhancing security of machine learning-based detection against adversarial android malware attacks[C]//Proceedings of the 33rd Annual Computer Security Applications Conference. Orlando, FL, USA: Association for Computing Machinery, 2017: 362-372. DOI: 10.1145/3134600.3134636.

作者简介:

陈 镭(通讯作者) 男,(1987-),博士研究生,工程师.研究方向为机器学习、信息安全. E-mail: chenlei@nau.edu.cn.

杨章静 男,(1979-),博士,副教授.研究方向为模式识别与人工智能.

黄 璞 男,(1985-),博士,副教授.研究方向为模式识别与计算机视觉.