北京航空航天大学学报
Journal of Beijing University of Aeronautics and Astronautics
ISSN 1001-5965,CN 11-2625/V

《北京航空航天大学学报》网络首发论文

题目：基于 LIME 的恶意代码对抗样本生成技术
作者：黄天波，李成扬，刘永志，李燈辉，文伟平
DOI：10.13700/j.bh.1001-5965.2020.0397
收稿日期：2020-08-09
网络首发日期：2021-01-21
引用格式：黄天波，李成扬，刘永志，李燈辉，文伟平．基于 LIME 的恶意代码对抗样本生成技术．北京航空航天大学学报.
https://doi.org/10.13700/j.bh.1001-5965.2020.0397

基于 LIME 的恶意代码对抗样本生成技术

黄天波，李成扬，刘永志，李燈辉，文伟平✉

（北京大学 软件与微电子学院，北京 102600）
*通信作者 E-mail： weipingwen@ss.pku.edu.cn

摘要 基于机器学习检测恶意代码技术的研究和分析，针对机器学习模型对抗样本的生成提出一种基于 LIME 的黑盒对抗样本生成方法。该方法可以对任意黑盒的恶意代码分类器生成对抗样本，绕过机器学习模型检测。该方法首先使用简单模型模拟目标分类器的局部表现，获取特征权重，然后通过扰动算法生成扰动，再根据生成的扰动对原恶意代码进行修改后生成对抗样本，并基于 2015 年微软公布的常见恶意样本数据集和收集的来自 50 多个供应商的良性样本数据对该方法进行实验：首先参照常见恶意代码分类器实现了 18 个基于不同算法或特征的目标分类器，然后使用该方法对这些目标分类器进行攻击，使这些分类器的真阳性率均降低到接近 0。此外实验还复现了 MalGAN 和 ZOO 两个先进的黑盒样本生成方法与该方法进行对比，实验结果表明本文方法能够有效生成对抗样本，且方法本身具有适用范围广泛、能灵活控制扰动和健全性的优点。

关键词 对抗样本；恶意代码；机器学习；LIME；目标分类器
中图分类号 TP309 文献标识码：A DOI: 10.13700/j.bh.1001-5965.2020.0397

Adversarial sample generation technology of malicious code based on LIME

HUANG Tianbo, LI Chengyang, LIU Yongzhi, LI Denghui, WEN Weiping✉

（School of Software & Microelectronics, Peking University, Beijing 102600, China）

Abstract Based on the research and analysis of machine learning technology to detect malicious code, a LIME-based black-box adversarial examples generation method is proposed to generate adversarial samples for any black box malicious code classifier and bypass the detection of machine learning models. The method first uses a simple model to simulate the target classifier's local performances, obtains the feature weights, and generates disturbances through the disturbance algorithm. According to them, finally, the way modifies the original malicious code to generate adversarial samples. We tested the approach using Microsoft's common malicious sample data in 2015 and the collected benign sample data from more than 50 suppliers. Firstly, 18 target classifiers based on different algorithms or features were implemented concerning common malicious code classifiers. And their classifiers' true positive rates were reduced to approximately zero when we attacked them using the way. Besides, two advanced black box sample generation methods, MalGAN and ZOO, were reproduced for comparison with this method. The experimental results show that this paper's method can effectively generate adversarial samples, and the method itself owns various strengths, including broad applicability, flexible control of disturbances, and soundness.

Key words Adversarial samples; Malicious code; Machine learning; LIME；target classifiers

近几年恶意代码检测技术的研究表明机器学习在代码检测问题上被越来越多的研究人员应用，众多学者[1][2][3][4][5]提出将机器学习技术作为下一代恶意代码分类器的关键组成部分。分类器从恶意代码中提取特征，使用机器学习算法对良性程序与恶意程序进行分类。根据使用的特征性质，可以将恶意代码检测技术分为静态检测[6][7]和动态检测[8]。然而机器学习分类器在多个领域都被证明是不安全

收稿日期：2020-08-09
基金项目：国家自然科学基金（61872011）；
作者简介：黄天波 男，硕士研究生。主要研究方向：网络空间安全、恶意代码检测、代码混淆李成扬 男，硕士研究生。主要研究方向：二进制软件安全、程序漏洞挖掘与利用刘永志 男，硕士研究生。主要研究方向：网络空间安全、二进制软件安全、程序漏洞挖掘与利用李燈辉 男，硕士研究生。主要研究方向：网络空间安全、恶意样本分析、程序漏洞挖掘与利用文伟平 男，博士，教授，博士生导师。主要研究方向：系统与网络安全，大数据与云安全，智能计算安全
网络首发时间：2021-01-21 09:57:45 网络首发地址：https://kns.cnki.net/kcms/detail/11.2625.V.20210121.0934.001.html

的。随着机器学习越来越广泛的应用，对抗样本攻击和防御的研究也在变得越来越有意义，这一领域通常被归为对抗机器学习（Adversarial Learning）。在恶意代码检测问题上，基于机器学习的分类器在面对对抗样本攻击时也可能是非常脆弱的。因此研究在恶意代码检测问题上的对抗样本生成技术可以增加对这种攻击方法的了解，避免将分类器暴露在这种攻击之下，从而针对该攻击提出针对性或者普适性的防御方法。同时研究人员可以通过攻击方法攻击其所研究的分类器，对分类器的鲁棒性进行评估。

本文提出了一种基于 LIME[9]的对抗样本生成方法。该方法可针对未知算法和参数细节的分类器生成有效的对抗样本，并且通过引入扰动常量，使该方法具有了广泛的适用范围、能灵活控制扰动大小的优点。同时在实验中验证了该方法的健全性。

1 相关研究

为了使提出的样本生成办法不仅可攻击分类器，同时有助于评估防御[10]、提供具体级别的安全产品[11]，我们针对对抗样本相关技术、攻击者对抗能力的描述和样本修改进行了相关的研究。

1.1 恶意代码对抗样本相关技术

对抗样本攻击意为通过微小地修改机器学习分类器的输入样本，诱导分类器对修改后的输入产生错误的输出结果。2013 年由 Szegedy 等[12]在神经网络模型背景下给出了严谨的问题描述，之后出现基于梯度的白盒解决方案[13]。但是因为现实中的大部分分类器都是闭源的，相较于白盒方法，黑盒更具有广泛的使用价值。

2017 年 Hu 等[14]提出了 MalGAN，是一种针对恶意代码分类器的黑盒对抗样本生成方法。使用该方法可以对一些使用 One-Hot 型（只有 0 和 1 两种取值）特征的黑盒分类器生成对抗样本。

同年，Chen 等[15]提出 ZOO（Zeroth-Order Optimization，零阶优化）方法，是一种基于零阶优化估计目标分类器梯度进而生成对抗样本的方法。由于不需要梯度，ZOO 也是一种黑盒方法，使用 ZOO 无需额外训练模型。

不过在 2016 年 Ribeiro 等[9]提出 LIME（Local Interpretable Model-Agnostic Explanations，模型无关的局部可解释方法），是一种与模型无关的方法（Model-Agnostic）。原理是使用可解释的简单模型在局部逼近目标模型，无论目标模型有多复杂，其在局部（可理解为切点）的趋势（可理解为切线）都可以用一个简单模型来刻画，从而通过简单模型解释局部的特征权重。对于给定的目标模型 f 和输入向量 x，LIME 方法可以在 f 的局部使用简单模型模拟目标模型的局部性质，从而判断 x 各分量对分类结果的影响权重。局部模拟需要一组与 x 相近的特征，使用 z 代表临近特征。LIME 对特征 x 的解释 $\xi(x)$，通过下式表达：

$$\xi(x)=\arg\min_{g\in G} L(f,g,\Pi_x)+\Omega(g) \tag{1}$$

其中 $g\in G$，G 是用于模拟目标模型的简单模型集合，通常包括线性模型和决策树；$\Pi_x(z)$ 用于度量局部特征 z 和 x 的接近程度；$\Omega(g)$ 用来衡量 g 的复杂性（不利于解释的程度），这通常取决于选用的简单模型，例如选用决策树，决策树的深度就是影响 $\Omega(g)$ 的主要因素；L 是损失函数，用来描述 g 在 x 附近模拟 f 的效果，模拟效果越好，则 L 的值越小，L 的一种实现如下：

$$L(f,g,\Pi_x)=\sum_{z,z'\in Z}\Pi_x(z)(f(z)-g(z'))^2 \tag{2}$$

在 L 中，z 和 x 的距离越小，则 $\Pi_x(z)$ 越大，也就是说，离 x 越近，误差权重越大，通过最小化，这样的损失函数使 g 在 x 局部获得较好的模拟效果，而 $L(f,g,\Pi_x)+\Omega(g)$ 最小，意味着使局部模拟和可解释的综合效果最好。作者用 K-Lasso 算法选择 k 个特征（通过正则化路径），然后基于对 x 的随机扰动生成一组数据，然后通过最小化 $L(f,g,\Pi_x)+\Omega(g)$ 学习到最优的简单模型 g，最后通过对 g 的参数 w 分析即可得到特征权重，例如当 g 是线性模型时，g 的参数 w 就对应了每个特征的

权重。如果 k 被设置为特征的总数，那么将得到所有特征的权重。由于 LIME 方法在设计时没有对目标模型做任何假设约束，所以使用时无需知道目标模型的算法和参数，理论上可用于解释任意黑盒模型。因为只在局部用简单模型模拟，效果好速度快，这也是本文所设计的对抗样本方法选用 LIME 的原因。

1.2 对攻击者能力的描述

精确的描述攻防场景，需要对攻击者的攻击能力做出合理的预估[16]。本文采用 Stokes 等[17]的提法，将特征分为正特征（Positive Feature）和负特征（Negative Feature）：正特征表示样本中有利于使分类器判断为恶意代码的特征，这些特征往往代表恶意行为；负特征表示样本中有利于使分类器判断为良性代码的特征。Crandall 等[18]指出，攻击者通常采取变构策略，使用替代代码的方式来达到所需的恶意目标，攻击者有能力删除正特征或添加负特征。在 MalGAN 方法中，假设攻击者只能添加特征，不能删除特征，从而保证不影响原样本的程序特征，但方法不对添加特征的数量进行限制，本身的假设存在低估攻击者对抗能力的可能。Inigo Incer 等[19]使用了特定的对抗能力来描述其所提供的安全机制的安全边界，他们通过分析一系列特征的修改，并评估每种类型的修改是否是简单廉价的，从而确定攻击者的对抗能力。但是在对这些类别的修改进行评估时，评估的标准缺少灵活性。事实上，对于攻击能力较强的专业黑客，即使是被认为困难的修改，也是可以完成一定数量的，而对于被认为是简单廉价的修改，在现实场景中，攻击者也不一定会不计数量的大量修改，大量未使用的特征可能增大被检测为对抗样本的概率，如 Xu 等人[20]提出一种检测对抗样本的方法，筛选被添加到原样本中却未使用到的修改作为检测标准之一。

基于上述分析，可以明显看到这些假设容易低估或者高估一些攻击者，导致只能用于特定研究背景，具有一定的局限性。为了寻找适合衡量对抗能力的方法，需要进一步分析攻击者修改样本的过程。

1.3 样本修改过程

攻击者在其能力和成本范围内对原样本进行修改，在保证新样本可执行的前提下，要确保新样本与原样本主要程序功能相同，即保证对抗样本的有效性。一般修改的过程分为两个部分：逆提取和逆预处理。逆提取中针对新旧样本的差异特征 r 和中间层表示的对应关系，通过直接修改中间层文件或者样本的源代码以满足在不影响其他特征的基础上添加新特征。逆预处理实现修改后的中间层文件到可执行文件的转变，可以通过逆向工程的方式从中间文件转换为可执行文件，甚至，若在逆提取阶段中直接修改源码，则可编译得到可执行文件。

通过对攻击过程进行分析可知：修改样本的难度和成本主要体现在逆提取过程中。于是引入扰动常量的概念，用于描述攻击者的对抗能力，并提出基于 LIME 的扰动方式的实现。

2 基于 LIME 的恶意代码对抗样本生成方法

2.1 基于 LIME 的恶意代码对抗样本生成过程

对抗样本的生成过程主要包含 4 个部分：探测（A 部分），扰动方法（B 部分），逆向过程（C 部分）和验证部分（最下方虚线部分）。

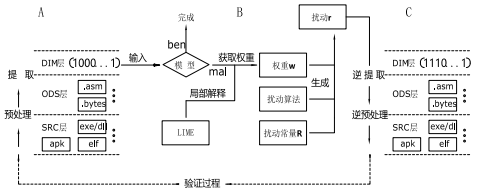

图 1 基于 LIME 的恶意代码分类器对抗样本生成过程图

Fig.1 Diagram of the generation process of anti-samples based on LIME malicious code classifier

该对抗样本生成方法所解决问题可描述为：

$$f(x)=mal \text{ and } f(x+r)=ben \quad (3)$$

$$g(x)=g(x+r) \quad (4)$$

其中 mal（Malicious）表示恶意代码分类标签，ben（Benign）表示良性代码分类标签。x 是目标分类器 f 的输入，表示恶意代码中提取的特征，r 是对 x 的扰动（Perturbation）。函数 g 表示获取样本的主要程序功能，$g(x)$ 是 x 原本的主要程序功能，$g(x+r)$ 表示修改后的主要程序功能。在保证代码语义的前提下，诱使目标分类器将恶意代码标识为良性代码。对图 1 的 4 个模块做出具体阐述如下：

1. 探测，通过特征工程获取样本特征，探测目标分类器 f 的分类结果，当结果为 ben 时结束方法，否则进行下一步。

2. 扰动，使用基于 LIME 的扰动方法生成一个扰动 r，满足 $f(x+r)=ben$，生成成功则进入下一步，失败则方法以失败结束。

3. 逆向过程，根据扰动 r 修改相应样本程序，生成最终可逃逸检测的对抗样本，使用该对抗样本再次进行第 1 步的探测，应能够正常结束方法，否则可能是在逆向过程出错，检查错误后重试方法。

4. 验证过程，如果修改后的样本满足公式 4，且能够按照探测模块的方法再次探测，并能得到探测结果 ben，则验证通过，结束方法。

2.2 扰动算法设计

扰动模块的实现，包括两个方面：扰动常量和扰动方式。扰动常量形式化的描述了攻击者在有限成本内修改样本，且保持样本主要功能不变的能力。扰动常量越大，攻击者能力越强；扰动方式指定了对于具体特征的修改，不同于使用雅可比算法[21] 等白盒方法，使用 LIME 模拟黑盒分类器在恶意样本处的局部表现，从而确定影响样本分类的关键特征，然后使用一个扰动算法，在保证不影响输入样本主要程序功能的同时，修改部分关键特征。

2.2.1 扰动常量

扰动常量 R，用于描述在特定攻击场景下攻击者的成本和能力边界，也就是对抗能力——攻击者在有限成本内修改样本，且使修改后的样本保持原样本主要程序性质不变的能力。以下是 R 的一般形式：

$$R=\{(d_i,s_{i1},s_{i2})\,|\,s_{i1}<=s_{i2},d_i\in[0,1]\}_{i=1}^{k} \quad (5)$$

其中 k 为正整数，d_i、s_{i1} 和 s_{i2} 均是实数。R 是一个 $k*3$ 的向量，包含 k 个扰动规则，*i-th* 表示第 i 个扰动规则，通常用一或两个扰动规则描述一种特征，对应添加和删除规则。不妨假设 i 规则描述一种 A 特征，则攻击者使用 i 规则对 A 特征所能修改的维数比例的最大值用 d_i 表示，d_i 取值在[0,1]之间，如果 A 特征在特征向量中一共 m 维，攻击者真实修改的 A 特征维数应不大于 $m*d_i$；我们用 s 描述对具体某维特征的修改能力，允许对某个特征在初始值的基础上加 s，这里的 s_{i1} 和 s_{i2} 用来描述 s 的取值范围。当 s_{i1} 与 s_{i2} 确定时，区间 $[s_{i1},s_{i2}]$ 表示对所有 A 特征，s 取值应在 $[s_{i1},s_{i2}]$ 范围内。这个区间往往由特征的性质、攻击者的攻击成本和能力所决定。这样 k 个扰动规则综合起来，就描述了 R 表示的范围。攻击者能力越强或越不计成本，R 表示的范围就越大；特征越容易修改，R 表示的范围也越大。

2.2.2 基于 LIME 的扰动方式实现

扰动方式指定了对于具体特征的修改。首先使用 LIME 方法求解出逼近真实情况的 ω。在 LIME 方法中需要选择简单模型 g 来模拟目标分类器，这里以线性模型为例，那么 g(x)可以表示为：

$$g(x)=\omega_1\cdot x_1+\omega_2\cdot x_2+\omega_3\cdot x_3+\ldots+\omega_m\cdot x_m \quad (6)$$

$\omega=\{\omega_1,\omega_2,\omega_3,\cdots,\omega_m\}$ 是 g 的参数。使用 K-Lasso 算法选择 k 个特征（通过正则化路径），然后

基于对 x 的随机扰动生成一组临近 x 的数据，在这组数据基础上最小化损失函数 $L(f,g,\Pi_x)$ ，可以得到此时 g 的参数 ω，用 ω 近似表示特征 x 的权重，即另 $w=\omega$，从而求解出 w，如果 k 被设置为特征的总数，那么将得到所有特征的权重。

表1 创建扰动r的算法伪码

Table 1 Perturbation algorithm

```
Algorithm 1: Perturbation
Data: w = {w1, w2, w3, ..., wm}, distance, R
1 r ← {Δx1, Δx2, Δx3, ..., Δxm},Δxi = 0
2 U ← {(1, w1), (2, w2), (3, w3), ..., (m, wm)}
3 sum ← 0
4 list sort|w|(U) // Sort by absolute value of wi
5 if not list then
6       return fail // list is empty
7 end
8 while list do
9       i, w ← list.remove(0) // remove the first element
10      Δxi ← argminΔx,Δx∈R Δx * w
11      sum ← sum + Δxi * w
12      if sum <distance then
13            break
14      end
15 end
16 f(x + r) ← ben? return r : distance ← distance * 2, go 5
```

当攻击者获取权重 w 之后，需要根据 w 和扰动算法生成一个符合公式 3，4 的扰动 r，且攻击者可以在对抗能力范围内对恶意样本做出 r 对应的修改。我们不妨设分类器给出的置信度在[0, 1]区间内，以 0.5 为分界，0 表示百分之百确定是 ben 样本，1 表示百分之百确定是 mal 样本。一个创建扰动 r 的过程的算法如表 1 所示。

攻击者首先探测到分类器 f 检测恶意样本 x 的置信度 $confidence=f(x)$，对于恶意样本，$confidence$ 在(0.5, 1]之间。由于 w 是已经求得的，攻击者需要在 x 中选择一些维度，在使 $confidence$ 减少的方向上，做一些能力和成本范围之内的改变。考虑 g 取线性模型的情况，此时有 $g(x)=w\cdot x$，且 $g(x+r)=g(x)+g(r)$，因为 $g(x)$ 是对 $f(x)$ 的局部模拟，攻击者可以通过 g 近似计算 $confidence$，所以一般满足 $g(r)=w\cdot r<-0.2$ 即可。在具体选择 r 的维度时，由于 w 已经用 LIME 方法求得，可以使用贪心法，按权重由大到小，依次判断能否对相应特征做出符合 R 要求的修改，从而挑选合适的维度，我们用 $\Delta x\in R$ 表示对某个特征进行 Δx 的变动是符合 R 要求的修改，如果符合，就加入候选维度。

3　实验评估

为了验证本文所提出的基于 LIME 的对抗样本生成方法的效果，分别进行攻击实验和对比实验：攻击实验用于测试本文方法在目标分类器上的攻击效果；对比实验用于和同类方法做比较，增强结论的说服力。本节将介绍实验环境、数据集、评估指标、目标分类器设置以及实验设计。

3.1　实验环境

表2　实验的硬件、软件环境

Table 2　Hardware and software environment

硬件环境	内存：16G CPU：Inter(R) Core(TM)i7-8550U
软件环境	IDA pro 7.0 lime 0.1.1.37 keras 2.3.1 python 3.6.3 tensorflow 1.15.0

numpy 1.18.1
sklearn 0.20.0
adversarial-robustness-toolbox 1.1.1

实验的软硬件环境信息如表 2：实验代码主要使用 Python3.6.3 实现，在数据集准备环节借助了 IDA pro 提供的 Python2.7.17 脚本执行接口,在 IBM 提供的 adversarial-robustness-toolbox 包中，封装了许多应用于机器学习分类器的攻击与防御方法，这包括本文要使用的 ZOO 方法，此外本文所要使用的 LIME 方法被封装在 lime 包中。

3.2 数据集

本文收集了两组不相交的 Win32 PE 文件，这两组文件的收集方式借鉴于文章[22]，对于每个 PE 文件，我们都使用其 IDA pro 反汇编生成的 ASM 文件来代表 PE 文件。良性样本源于系统镜像中使用 ninite[23]安装的 50 多个供应商的应用软件，进而有效避免学习和识别与特定供应商相关联的文件[24]。恶意样本包含来自 Ramnit, Lollipop 等 9 个恶意代码家族的 10868 个 Win32 PE 恶意程序对应的 ASM 文件，这些文件来自 2015 年微软举办 Kaggle 比赛[25]时公开的数据。

3.3 评估指标

因为攻击者希望原本被识别的恶意样本，经改变为对抗样本后，被标识成良性样本，攻击前后的真阳性率 TPR 之差就是有效对抗样本的比例；为了直观考虑，我们将攻击前后的 TPR 之差与攻击前的 TPR 之比称为攻击成功率，表示为 ASR，则有 $ASR = 1 - TPR_{攻击后} / TPR_{攻击前}$；除上述两个指标外，为了保证说服力，本文采用准确率 ACC 来评估自建的目标分离器，仅保留那些准确率 90%以上的分类器纳入实验。采用的评估指标如表 3 所示。

表 3 评估指标
Table 3 Evaluation indicators

评估指标	公式
真阳性率 TPR	TPR = TP/(TP+FN)
准确率 ACC	ACC = (TP+TN)/(TP+TN+FP+FN)
攻击成功率 ASR	ASR = 1-TPR $_{攻击后}$/TPR $_{攻击前}$

为充分测试方法的效果，目标分类器根据使用的算法或特征差异，可分为 18 个，如表 4 所示。算法涵盖线性、树形和深层神经网络类算法，包括：LR、RF、SVM、MLP 算法。特征包括：API、opc-2gram、opc-3gram。

表 4 目标分类器设置
Table 4 Target classifier settings

#编号	API	opc-2gram	opc-3gram
LR	#1	#2	#3
RF	#4	#5	#6
SVM	#7	#8	#9
MLP1	#10	#11	#12
MLP2	#13	#14	#15
MLP3	#16	#17	#18

将算法用集合 alg 表示，特征用集合 fea 表示，alg 与 fea 做笛卡尔积，则有 12 个有序对组合（表 4 的#1~#12），用一个数字编号表示一种组合，例如#1 号表示（API，LR），代表使用 API 特征和 LR 算法训练的一个分类器。考虑到基于 MLP 的模型根据隐层数不同可能存在较大的性质差异，我们对#10~#12 号分类器额外设置了 2 组不同层数的分类器，因此共有 18 个目标分类器。

3.4 实验设计

攻击实验中恶意样本数据集有 60%用于目标分类器的训练，在剩余的 40%中，在 9 个恶意代码家族各随机选取了 20 个样本，我们将这 180 个样本称为攻击样本，用于生成对应的对抗样本。

使用本文的对抗样本生成方法，生成上述 180 个攻击样本的对抗样本来攻击每一个目标分类器。用扰动常量控制扰动大小[26]，绘制扰动大小-TPR 图、扰动大小-ASR 图，以获得不同扰动强度下的攻

击效果。为进一步增强说服力，本文设计了对比实验：复现 MalGAN 和 ZOO 这两个先进的黑盒对抗样本生成方法，生成攻击样本的对抗样本，攻击#1、#4、#7、#10、#13、#16 号分类器，记录 TPR 攻击前后的变化，并与本文方法进行对比分析。

3.5　实验与结果

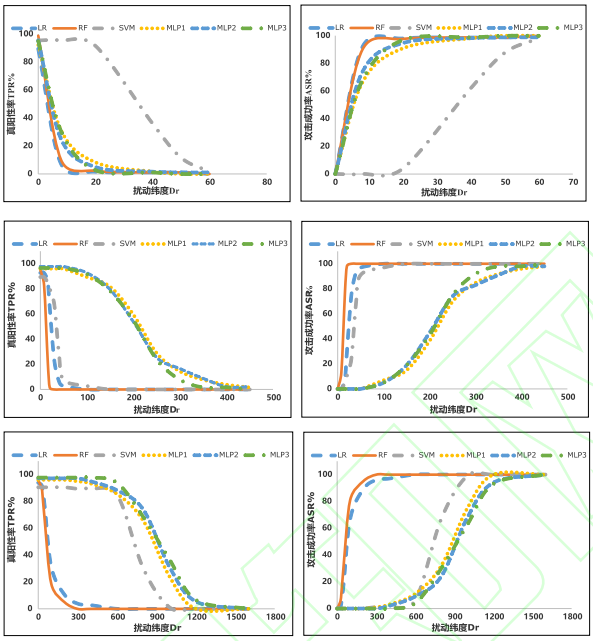

图 2　三种特征的 Dr-TPR 图，Dr-ASR 图
Fig.2 Dr-TPR and Dr-ASR of three characteristics

图 2 为基于不同扰动维度（Dr）生成对抗样本，攻击各目标分类器产生的 TPR 和 ASR 的变化图。三行图片分别对应特征 API、opc-2gram、opc-3gram 的 TPR 和 ASR 图。查看 TPR 图，可以发现 18 个分类器的 TPR 均降到了接近 0 的水平，即在较高扰动代价情况下，本文方法几乎 100%成功地攻击任意恶意代码分类器。

对比实验中使用 MalGAN 和 ZOO 生成攻击样本的对抗样本，攻击#1、#4、#7、#10、#13、#16 号分类器。每个分类器性质不同，选用参数也有差异，本文倾向于选择使对抗样本更有效的参数，经过多次实验，结合本文方法汇总如表 5。

表5　API特征的分类器，真阳性率TPR对比表
Table 5　API feature classifier and TPR comparison table

对抗样本生成方法	目标分类器序号-使用的算法					
	#1-LR	#4-RF	#7-SVM	#10-MLP1	#13-MLP2	#16-MLP3
无攻击对照	89.44	98.89	95.38	92.22	95.56	95.00
ZOO	57.78	91.11	61.67	69.44	72.78	70.56
MalGAN	0.00	1.67	0.56	0.00	0.00	0.00
本文方法	0.00	0.00	1.67	0.00	1.11	0.00

从表 5 可以看出，MalGAN 和本文方法都具有较好的效果，且攻击效果相似（将 TPR 降到更低，且降幅相似）。ZOO 在实验中表现不佳，攻击后的 TPR 都在 50%以上；MalGAN 将#1、#10、#13、#16 号分类器的 TPR 降到 0，#4、#7 号分类器的 TPR 降到 1.67%和 0.56%；本文方法将#1、#4、#10、#16 号分类器的 TPR 降到 0，而#7、#13 号分类器的 TPR 分别降到 1.67%和 1.11%。为了进一步对比这 3 种方法，我们比较了 3 种方法生成的对抗样本本身的差异。

表6 两种方法生成的对抗样本平均扰动维度

Table 6 The average perturbation dimension of the adversarial samples generated by MALGAN and LIME

对抗样本生成方法	目标分类器序号-使用的算法					
	#1-LR	#4-RF	#7-SVM	#10-MLP1	#13-MLP2	#16-MLP3
MalGAN	28.31	23.26	102.87	24.70	35.14	29.45
本文方法	9.56	11.62	43.12	15.47	13.39	13.92

实验中的 API 特征是 One-Hot 型的，对应取值应该是 0 或者 1，而 ZOO 在特征中可能出现-3、-1、2 等值，我们只能通过筛选来获取符合要求的对抗样本，这将导致有效对抗样本进一步减少；MalGAN 适用于 One-Hot 类型的特征，我们主要对比其生成的对抗样本的扰动维度。表 6 展示了 MalGAN 和本文方法在对各分类器取得较好攻击效果时的平均扰动纬度 Dr（单位：个）。其中，攻击效果相似的情况下，本文方法比 MalGAN 生成的样本平均扰动维度小，这可能因为本文方法有针对扰动大小的设计。

4 结 论

1）该方法是一种有效的黑盒对抗样本生成方法。使用该方法生成的对抗样本测试黑盒的恶意代码分类器，能显著降低分类器的真阳性率 TPR。

2）该方法适用范围广泛。使用该方法攻击 18 个不同算法或特征的目标分类器，均有不错的攻击成功率 ASR。目标分类器的算法涵盖了线性算法、树形算法和深度神经网络算法，而特征既有 One-Hot 型的也有数值型的。

3）该方法能有效控制扰动的大小。可以通过设置不同的扰动常量来控制对抗样本的扰动维数和扰动范围。

4）该方法具有健全性。随着扰动的增大，ASR 是严格递增的；随着扰动持续增大，ASR 接近或达到 100%。虽然过大的扰动可能会使对抗样本失去意义，但可以说明该方法是健全的。

5 参考文献

[1] M. Alazab, “Automated Malware Detection in Mobile App Stores Based on Robust Feature Generation,” Electronics, vol. 9, no. 3, Mar, 2020

[2] Saxe J , Berlin K . Deep neural network based malware detection using two dimensional binary program features[C]// 2015 10th International Conference on Malicious and Unwanted Software (MALWARE). IEEE, 2015.

[3] Pascanu R, Stokes J W , Sanossian H , et al. Malware classification with recurrent networks[C]// Icassp IEEE International Conference on Acoustics. IEEE, 2015: 1916-1920.

[4] HUANG Wenyi, Stokes J W . MtNet: A Multi-Task Neural Network for Dynamic Malware Classification[J]. Proceedings of Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 2016: 399-418.

[5] Kolosnjaji B , Zarras A , Webster G , et al. Deep Learning for Classification of Malware System Call Sequences[J]. Australasian Joint Conference on Artificial Intelligence. Springer, 2016: 137-149.

[6] Schultz M G , Eskin E , Zadok F , et al. Data Mining Methods for Detection of New Malicious Executables[C]// IEEE Symposium on Security & Privacy. IEEE, 2001: 38-49.

[7] Jeremy Z Kolter and Marcus A Maloof. Learning to Detect Malicious Executables in the Wild[C]. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, August 2004, Seattle, WA, USA. New York: ACM, 2004: 470-478.

[8] Kolter J Z , Maloof M A . Learning to Detect and Classify Malicious Executables in the Wild[J]. journal of machine learning research, 2006, 7(4): 2721-2744.

[9] Ribeiro M T , Singh S , Guestrin C . "Why Should I Trust You?": Explaining the Predictions of Any Classifier[C]// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA. New York: ACM, 2016: 1135-1144.

[10] SU Dong , ZHANG Huan , CHEN Hongge, YI Jinfeng , et al. Is Robustness the Cost of Accuracy? -- A Comprehensive Study on the Robustness of 18 Deep Image Classification Models[J]. The European Conference on Computer Vision (ECCV). 2018.

[11] Stokes J W , Wang D , Marinescu M , et al. Attack and Defense of Dynamic Analysis-Based, Adversarial Neural Malware Detection Models[C]// MILCOM 2018 IEEE Military Communications Conference. IEEE, 2018: 1-8.

[12] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[J]. 2013.

[13] Goodfellow I J , Shlens J , Szegedy C . Explaining and harnessing adversarial examples[J]. 3rd International Conference on Learning Representations, ICLR, 2015.

[14] HU Weiwei., AND TAN Ying. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN[J]. 2017

[15] CHEN Pinyu, ZHANG Huan, Y. Sharma, J. Yi, C.-J. Hsieh, Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models[C]// Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security(AISec’17). Dallas, Texas, USA. New York: ACM, 2017: 15-26.

[16] Carlini N , Athalye A , Papernot N , et al. On Evaluating Adversarial Robustness[J]. 2019.

[17] Stokes J W , Wang D , Marinescu M , et al. Attack and Defense of Dynamic Analysis-Based, Adversarial Neural Malware Detection Models[C]// MILCOM 2018 IEEE Military Communications Conference. IEEE, 2018: 1-8.

[18] J. Crandall, Z. Su, F. Chong, and S. Wu. On deriving unknown vulnerabilities from zero-day polymorphic and metamorphic worm exploits[C]// Proceedings of the ACM Conference on Computer and Communications Security (CCS'05). Alexandria, VA, USA. New York : ACM, 2005: 235-248.
[19] Inigo Incer, Michael Theodorides, Sadia Afroz, and David Wagner.Adversarially Robust Malware Detection Using Monotonic Classification[C]// Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics (IWSPA'18), Tempe, AZ, USA. New York: ACM, 2018: 54-63.
[20] XU Weilin, D. Evans, and QI Yanjun, et al. Feature squeezing: Detecting adversarial examples in deep neural networks[C]// 25th Annual Network and Distributed Systems Symposium February(NDSS 2018), San Diego, California, USA. Geneva: The Internet Society, 2018.
[21] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings[C]// Security and Privacy (Euro S&P), 2016 IEEE European Symposium on. IEEE, 2016: 372-387.
[22] Kreuk F , Barak A , Aviv-Reuven S , et al. Adversarial Examples on Discrete Sequences for Beating Whole-Binary Malware Detection[J]. 2018.
[23] SWIESKOWSKI, P., AND KUZINS, S. Ninite. https://ninite.com/
[24] RAF, E., BARKER, J., SYLVESTER, J., BRANDON, R.,CATANZARO, B., AND NICHOLAS, C. Malware Detection by Eating a Whole EXE[C]// The Workshops of the The Thirty-Second AAAI Conference on Artificial Intelligence, February 2018, New Orleans, Louisiana, USA. AAAI Press, 2018: 268-276.
[25] MicroSoft.Kaggle microsoft malware classification challenge 2015. https://www.kaggle.com/c/malware-classification/data
[26] Carlini N , Athalye A , Papernot N , et al. On Evaluating Adversarial Robustness[J]. 2019