对抗训练驱动的恶意代码检测增强方法

刘延华[1,2]，李嘉琪[1,2]，欧振贵[1,2]，高晓玲[1,2]，刘西蒙[1]，MENG Weizhi[3]，刘宝旭[4]

（1. 福州大学计算机与大数据学院，福建 福州 350108；2. 福建省网络计算与智能信息处理重点实验室，福建 福州 350108；
3. 丹麦科技大学应用数学和计算机系，哥本哈根 2800；4. 中国科学院信息工程研究所，北京 100093）

摘 要：为了解决恶意代码检测器对于对抗性输入检测能力的不足，提出了一种对抗训练驱动的恶意代码检测增强方法。首先，通过反编译工具对应用程序进行预处理，提取应用程序接口（API）调用特征，将其映射为二值特征向量。其次，引入沃瑟斯坦生成对抗网络，构建良性样本库，为恶意样本躲避检测器提供更加丰富的扰动组合。再次，提出了一种基于对数回溯法的扰动删减算法。将良性样本库中的样本以扰动的形式添加到恶意代码中，对添加的扰动进行二分删减，以较少的查询次数减少扰动的数量。最后，将恶意代码对抗样本标记为恶意并对检测器进行重训练，提高检测器的准确性和稳健性。实验结果表明，生成的恶意代码对抗样本可以躲避目标检测器的检测。此外，对抗训练提升了目标检测器的准确率和稳健性。

关键词：对抗训练；检测增强；生成对抗网络；扰动删减

中图分类号：TN92

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2022171

Adversarial training driven malicious code detection enhancement method

LIU Yanhua[1,2], LI Jiaqi[1,2], OU Zhengui[1,2], GAO Xiaoling[1,2], LIU Ximeng[1], MENG Weizhi[3], LIU Baoxu[4]

1. College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China
2. Fujian Provincial Key Laboratory of Networking Computing and Intelligent Information Processing, Fuzhou University, Fuzhou 350108, China
3. Department of Applied Mathematics and Computer Science, Technical University of Denmark, Copenhagen 2800, Denmark
4. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

Abstract: To solve the deficiency of the malicious code detector's ability to detect adversarial input, an adversarial training driven malicious code detection enhancement method was proposed. Firstly, the applications were preprocessed by a decompiler tool to extract API call features and map them into binary feature vectors. Secondly, the Wasserstein generative adversarial network was introduced to build a benign sample library to provide a richer combination of perturbations for malicious sample evasion detectors. Then, a perturbation reduction algorithm based on logarithmic backtracking was proposed. The benign samples were added to the malicious code in the form of perturbations, and the added benign perturbations were culled dichotomously to reduce the number of perturbations with fewer queries. Finally, the adversarial malicious code samples were marked as malicious and the detector was retrained to improve its accuracy and robustness of the detector. The experimental results show that the generated malicious code adversarial samples can evade the detector well. Additionally, the adversarial training increases the target detector's accuracy and robustness.

Keywords: adversarial training, detection enhancement, generative adversarial network, perturbation reduction

收稿日期：2022−06−08；**修回日期：**2022−08−29

通信作者：刘宝旭，liubaoxu@iie.ac.cn

基金项目：国家自然科学基金资助项目（No.62072109, No.U1804263）；福建省自然科学基金资助项目（No.2021J01625, No.2021J01616）；福建省科技重大专项（科教联合）项目（No.2021HZ022007）

Foundation Items: The National Natural Science Foundation of China (No.62072109, No.U1804263), The Natural Science Foundation of Fujian Province (No.2021J01625, No.2021J01616), Major Science and Technology Project of Fujian Province (No.2021HZ022007)

0 引言

根据我国互联网网络安全监测数据分析报告，在 2021 年上半年，我国境内感染计算机恶意程序的主机约有 446 万台，同比增长 46.8%，发现新增移动互联网恶意程序 86.6 万余个。随着恶意代码及其变种数量的增加，恶意代码检测面临着巨大的挑战[1]。基于特征码的检测技术无法应对新型恶意代码，以人工分析为主要方式的检测技术存在检测效率低等明显问题，无法适应当前的网络安全环境，自动化和智能化的恶意代码检测具有一定的必要性。

在智能化恶意代码检测中，相关特征（如任务、意图、应用编程接口调用、系统调用以及字节特征等）被提前提取并用于恶意代码检测器的训练，取得了较好的结果[2-3]。然而，机器学习本身存在一些安全性问题[4]。机器学习模型的有效性取决于训练数据和测试数据遵循相同分布的假设，这种假设很可能遭到攻击者的破坏，损害模型的安全性。攻击者在输入样本上施加微小的扰动便能迫使分类模型输出错误的预测，这种方式称为对抗样本攻击[5]。在恶意代码领域，攻击者利用模型的不足，生成恶意代码样本，达到绕过恶意代码检测器的目的[6-7]。

随着恶意代码反检测能力的提高，增强恶意代码检测器识别对抗样本的能力，提高检测器的稳健性，是现阶段提升恶意代码检测水平的关键。防御蒸馏[8]、对抗训练[9]和对抗样本拒绝[10]等是常见的对抗样本防御措施。其中，对抗训练被认为是抵抗对抗攻击的最佳解决方案，它利用训练好的模型来生成对抗样本，然后将它们添加到训练集中以重新训练模型，从根本上增强目标分类器的稳健性[11]。Wang 和 Liu 等[12]通过对抗训练方法提升面向恶意软件 C2 流量的检测能力。Wang 和 Zhang 等[13]提出了一个用于生成对抗样本进行对抗训练的框架，通过重训练提高分类器在安卓恶意软件检测和家庭分类中的有效性。这些研究证明了对抗训练增强恶意代码检测器的可行性。

在恶意代码检测器对抗训练的过程中，对抗样本的生成是一个重要环节。如何利用对抗样本知识，以较小攻击成本和较高攻击成功率生成具有现实意义的恶意代码对抗样本是恶意代码领域的一个重要问题。Goodfellow 等[14]提出的生成对抗网络（GAN, generative adversarial network）在样本生成上具有一定的优势。GAN 由生成器和判别器构成，通过生成器和判别器之间的博弈，生成器将学习到数据的潜在规律并生成新的数据。Kim 等[15]利用 GAN 生成基于灰度图像的恶意代码样本。之后，Kim 等[16]在文献[15]的基础上利用深度卷积 GAN 模型生成恶意代码样本，并基于图像结构相似性模拟零日恶意代码的生成。文献[17]利用辅助分类生成对抗网络生成恶意代码灰度图像，但没有考虑生成器生成的恶意代码质量。由于恶意代码的相邻字节之间存在结构上的相互依赖关系，对于恶意代码文件的任何更改都可能破坏可执行文件的功能，影响恶意代码的恶意功能[18]。恶意代码对抗样本与对抗性图像不同，即使成功地欺骗了检测模型，这些对抗样本在现实世界中也是不可行的。

在恶意代码的执行性问题上，Hu 等[19]提出了一种基于 GAN 的恶意软件生成算法，通过引入一个替代检测器，对恶意代码检测器实现黑盒攻击，并通过在导入表中添加扰动应用程序接口（API, application programming interface）实现恶意代码对抗样本的生成。但是，基于原始 GAN 模型的恶意代码生成，容易面临梯度消失以及训练不稳定等问题[20]。而且该研究在对抗生成的过程中并未考虑到攻击成本的问题。唐川等[21]提出了一种基于最小修改成本的对抗样本生成算法，利用深度卷积 GAN 模型生成良性扰动，通过修改反编译文件并对安卓应用程序包（APK, Android application package）进行重打包，生成可执行的恶意软件对抗样本，成功绕过目标检测器的检测。但是该方法只考虑到了恶意代码特征的修改成本，未考虑到对抗样本生成过程中恶意代码检测器的查询次数。由于检测器的多次重复查询，容易引起安全人员的察觉，攻击者在对检测器进行攻击时，需要考虑到目标检测器查询效率问题。

针对上述问题，本文提出了一种对抗训练驱动的恶意代码检测增强方法。首先，基于沃瑟斯坦生成对抗网络[22]（WGAN, Wasserstein generative adversarial network）和扰动删减方法，生成低扰动数量、高查询效率的恶意代码对抗样本。然后，利用生成的对抗样本对目标检测器进行再训练，增强恶意代码检测器性能。本文的主要研究工作包含以下几个方面。

1) 提出一种基于 WGAN 的良性样本生成算法，构建面向 API 调用的良性样本库。利用 WGAN 在一定程度上解决原始 GAN 训练不稳定的问题。通过生成器和判别器之间的博弈训练，模拟良性样

本的分布，构建良性样本库，进而为恶意代码对抗提供更加丰富的扰动组合。

2) 提出一种基于对数回溯法的扰动删减算法，构造恶意代码对抗样本。将生成的良性样本以扰动的形式添加到恶意代码，利用对数回溯法对添加的扰动进行删减，以较少的扰动数量和目标检测器查询次数绕过恶意代码检测器。

3) 基于对抗训练对目标检测器进行增强。利用生成的恶意代码对抗样本对恶意代码检测器进行重训练，提高恶意代码检测器对于对抗样本的检测率。最后，选取不同的恶意代码检测器进行实验，验证了本文方法的有效性和通用性。

1　相关理论

1.1　WGAN

WGAN 是 GAN 的一种变体，不同于 GAN 使用具有突变性的詹森香农散度作为生成数据与真实数据间的距离衡量标准，WGAN 引入沃瑟斯坦距离作为损失函数，能够对 GAN 模型梯度消失以及训练不稳定问题进行优化。沃瑟斯坦距离更加平滑，即使 2 个分布互不重叠，也能够很好地反映二者的远近。沃瑟斯坦距离的计算方法如式(1)所示。

$$W(P_r,P_g)=\frac{1}{K}\sup_{\|f\|_L\leqslant K}\mathrm{E}_{\chi\sim P_r}[f(x)]-\mathrm{E}_{\chi\sim P_g}[f(x)] \quad (1)$$

其中，P_r 和 P_g 分别表示真实样本的分布和生成器生成样本的分布，K 表示利普希茨常数。$\|f\|_L\leqslant K$ 等价于 $\|f(x_1)-f(x_2)\|\leqslant K\|x_1-x_2\|$，若 f 的定义域为实数集合，则 $\|f\|_L\leqslant K$ 表示 f 的导函数绝对值不超过 K。

将判别器表示为函数 f，设定一个固定常数 c（c>0），以 c 的绝对值截断判别器的参数 ω，限制判别器的最大局部变动幅度，使其满足 $\|f\|_\omega\leqslant c$。在判别器和生成器双方的博弈中，判别器的目标是尽可能正确地区分真实样本与生成器生成的假样本，即最大化沃瑟斯坦距离。相反地，生成器的目标是最小化沃瑟斯坦距离，尽可能输出与真实样本相似的样本以欺骗判别器。判别器的损失函数、生成器的损失函数以及 WGAN 的目标函数分别如式(2)~式(4)所示。

$$L_D=E_{\chi\sim P_r}[D(x)]-E_{z\sim P_z}[D(G(z))] \quad (2)$$

$$L_G=-E_{\chi\sim P_g}[f_\omega(x)]=-E_{z\sim P_z}[D(G(z))] \quad (3)$$

$$\min_G\max_D V(D,G)=E_{\chi\sim P_r}[D(x)]-E_{z\sim P_z}[D(G(z))] \quad (4)$$

其中，G 和 D 分别表示生成模型和判别模型，θ 表示生成器的参数，ω 表示判别器的参数。P_r、P_g 和 P_z 分别表示真实样本分布、生成器生成样本分布和随机噪声分布。

WGAN 的训练是一个零和博弈的过程，生成器和判别器通过交替迭代训练，最终达到纳什均衡。训练判别器时，固定生成器的参数，将生成器生成的样本和真实样本作为判别器的输入，根据损失函数 L_D，更新判别器的参数并将梯度反向传播给生成器。每次判别器参数 ω 更新后将其按固定常数 c 的绝对值截断，将判别器的参数限制在固定范围内，即 $\omega\in[-c,c]$。训练生成器时，固定判别器的参数，输入一批随机噪声向量，然后输出虚拟数据，由判别器对生成的虚拟数据进行评估，根据损失函数 L_G 更新参数和反传梯度。

1.2　API 调用特征表示

本文主要研究面向 API 调用的恶意代码检测，通过获取应用程序的 API 调用特征判断应用程序是否具有盗取隐私信息、恶意删除文件等恶意行为。定义一个应用程序 API 特征集合 $S=\{s_1,s_2,\cdots,s_n\}$。将应用程序的 API 调用特征映射为一个二值特征向量，若该应用程序包含 API 调用 s_i，则对应位置的特征向量元素值为 1；若该应用程序未调用 s_j，则对应位置的特征向量元素值为 0。

例如，应用程序 API 特征集合包含 5 个 API，即 $S=\{s_1,s_2,s_3,s_4,s_5\}$，应用程序 x 使用了 s_1,s_3,s_4 这 3 个 API，则应用程序 x 的特征向量 $\boldsymbol{S}_x$ 可以表示为 $\boldsymbol{S}_x=[1,0,1,1,0]^{\mathrm{T}}$。

2　模型构建

为了增强恶意代码检测模型的稳健性和对抗样本识别能力，本文提出了对抗训练驱动的恶意代码检测增强方法。模型框架如图 1 所示，主要由数据预处理、良性样本库构建、对抗样本生成和对抗训练组成。

2.1　数据预处理

首先，使用反汇编工具对应用程序进行反编译，获取应用程序的 API 调用特征。由于总体 API 数量较多，使用从恶意样本数据中提取的 API 调用构建特征。同时，参考安卓开发者官网提供的 API 包索引名对提取的 API 调用特征进行过滤。最后，

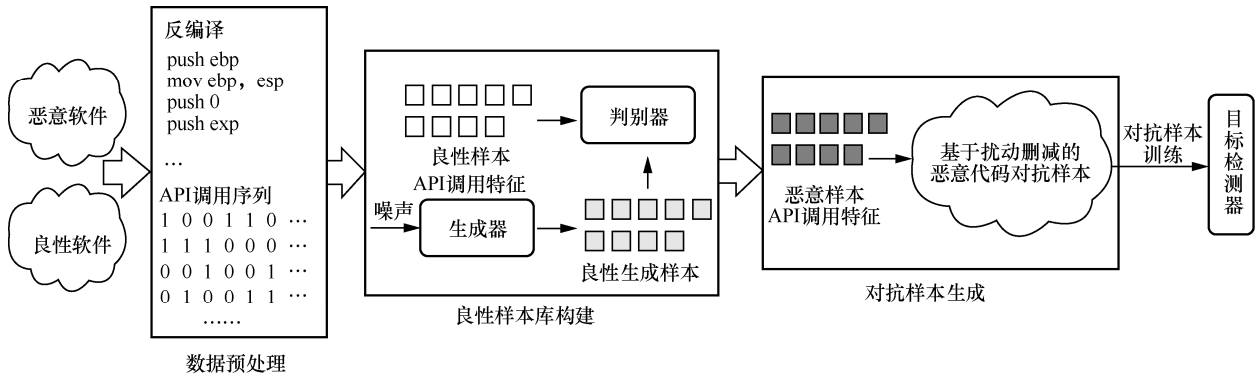


图 1　对抗训练驱动的恶意代码检测增强模型框架

采用卡方检验法降低特征维度，为每一个应用程序样本生成二值特征向量。

2.2　良性样本库的构建

WGAN 模型在一定程度上能够解决 GAN 训练不稳定和模式崩溃导致生成数据多样性不足的问题。在构建良性样本库的过程中，基于 WGAN 模型学习真实良性样本的特征分布，在满足真实样本分布的前提下，实现在真实良性样本基础上细微的变化，在一定程度上模拟良性样本的变种生成，进而提供更加丰富的扰动组合。

为了区分不同模块生成的样本，本文定义了不同的样本名称，具体描述如表 1 所示。

表 1　　各模块生成样本的详细描述

样本名称	描述
良性样本	基于WGAN模型生成的被恶意代码检测器判断为良性的样本
扰动样本	将良性样本的 API 特征以扰动的形式添加到恶意代码样本中，生成扰动样本
对抗样本	对扰动样本进行筛选，针对可以成功绕过恶意代码检测器的样本执行扰动删减算法，得到具有绕过恶意代码检测器检测的能力和较低攻击成本的对抗样本

在模型的结构设计方面，使用多层全连接网络构建 WGAN 生成模型和判别模型，网络结构分别如图 2 和图 3 所示。

生成模型由一个输入层、2 个隐藏层和一个输出层组成。输入层的输入向量为服从标准正态分布的随机噪声向量。隐藏层使用非线性函数 ReLU 作为激活函数，能够减少计算量和降低过拟合。输出层使用的激活函数为 Sigmoid。

判别模型由一个输入层、2 个隐藏层和一个输出层组成。输入层的输入来自真实样本。不同于生成模型使用的激活函数为 ReLU，判别模型的隐藏层使用的激活函数为 LeakyReLU。并且，在每个隐藏层后各添加一个 Dropout 层，防止模型过拟合。

2.3　对抗样本生成和对抗训练

本文采用模拟对抗样本攻击的方式，通过对恶意代码添加扰动生成对抗样本，达到绕过目标检测器检测的目的。其中，攻击者的能力设定为攻击者掌握了目标检测器所使用的算法和特征集合，但是无法获取检测器的训练数据；攻击者通过增加 API 调用的方式修改恶意软件；攻击者只能查询目标检

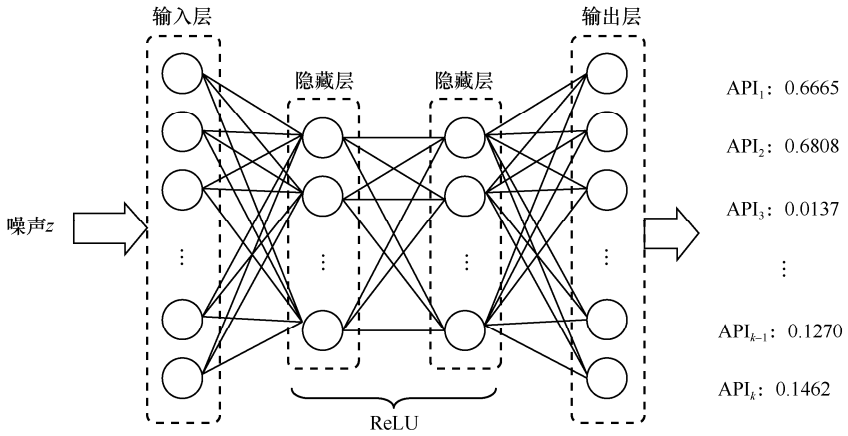


图 2　生成模型的网络结构

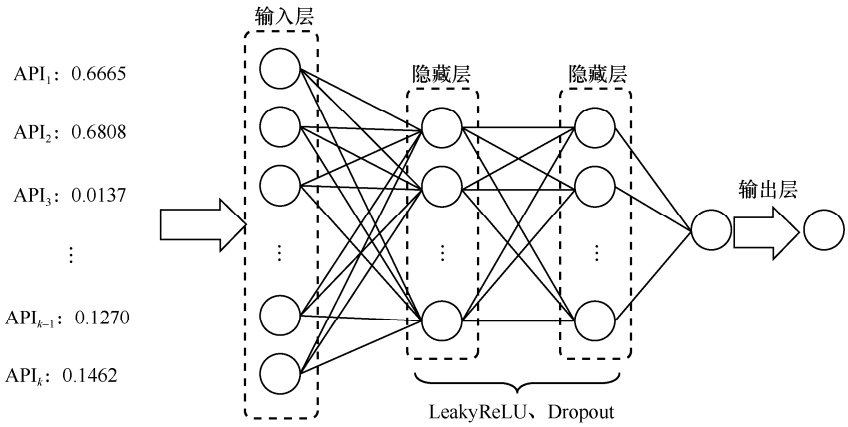


图 3　判别模型的网络结构

测器预测的类别。

由于恶意代码的特殊性，直接从原始恶意代码中删除一个特征可能会导致恶意功能消失，甚至程序崩溃。为了保留恶意代码的原始功能，只对原始样本添加 API 调用，不删除或修改原本存在的特征。对抗样本的生成流程如图 4 所示。

首先，将良性样本库中的样本以扰动的形式添加进恶意样本中，以躲避恶意代码检测器的检测。扰动方式如式(5)所示。

$$X'(i)=\max\left(X,B(i)\right),i\in\left(0,\cdots,n-1\right) \tag{5}$$

其中，X 为原始恶意样本，$B(i)$为良性样本库中的第 i 个样本，$X'(i)$ 为对应的添加扰动后的扰动样本，n 为良性样本库的规模。max(·) 代表 2 个特征向量间逐元素的或运算，若 X 的某一元素值为 1，则 $X'(i)$ 对应位置的元素值也为 1，即保留恶意样本中的原始 API 调用；若 X 的某一元素值为 0，而 $B(i)$ 对应位置的元素值为 1，则 $X'(i)$ 对应位置的元素值也为 1，即添加良性扰动。

其次，为了更加真实地模拟恶意代码制作者的攻击思路，本文从攻击者的角度出发，使用对数回溯法进行扰动删减，实现以较少的查询次数和较少的扰动数量生成恶意代码对抗样本。

最后，通过将生成的恶意代码对抗样本标注为恶意，扩充恶意代码检测器训练数据，完成检测器再训练，达到增强恶意代码检测器的目的。

3　基于 WGAN 的良性样本生成算法

本文提出基于 WGAN 的良性样本生成算法，WGAN 模型通过生成器和判别器之间的博弈训练，生成近似真实良性样本的数据。良性样本生成过程如算法 1 所示。

算法 1　良性样本生成算法

输入　真实良性样本集 X_{ben}，批次大小 m，每次生成器迭代中判别迭代次数 n_{critic}，判别器的初始化参数 ω_0，生成器的初始化参数 θ_0，生成样本个数 n，学习率 α，权值剪裁 c。

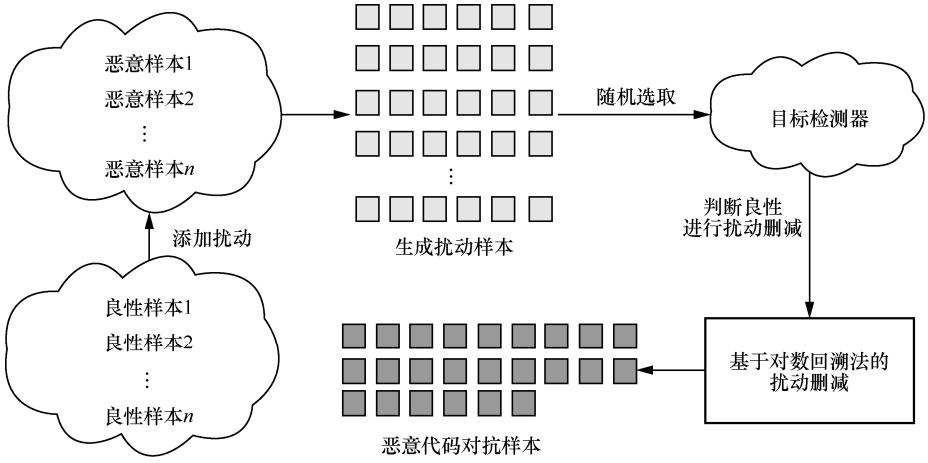


图 4　恶意代码对抗样本生成流程

输出 生成样本集 B_{gen}

1) while θ 未收敛 do
2) for $j \leftarrow 1,\cdots,n_{\text{critic}}$ do
3) 从真实良性样本集 X_{ben} 中抽取 m 个样本 $\{x^{(1)},x^{(2)},\cdots,x^{(m)}\}$
4) 从前置随机分布中抽取 m 个样本 $\{z^{(1)},z^{(2)},\cdots,z^{(m)}\}$
5) 计算参数 ω 的梯度
6) $\omega \leftarrow \omega + \alpha\text{RMSProp}(\omega, g_\omega)$
7) $\omega \leftarrow \text{clip}(\omega, -c, c)$
8) end for
9) 从前置随机分布中抽取一个批次的 m 个样本 $\{z^{(1)},z^{(2)},\cdots,z^{(m)}\}$
10) 计算参数 θ 的梯度
11) $\theta \leftarrow \theta - \alpha\text{RMSProp}(\theta, g_\theta)$
12) end while
13) 由训练好的生成器输出 n 个生成样本 $B_{\text{gen}} = \{o^{(1)},o^{(2)},\cdots,o^{(n)}\}$

首先，通过选择生成器生成的数据和真实样本来训练判别器，更新判别器的参数。然后，利用生成器生成的数据欺骗判别器，将判别器的判断结果反馈给生成器，并更新生成器的参数。通过生成器和判别器之间的博弈训练，生成器可以生成更真实的样本。

由于样本数据是由 0 和 1 构成的二进制特征向量，而生成样本的数值介于 0～1，需要对生成器的输出进行二值化处理。二值化处理如式(6)所示。

$$b_i = \begin{cases} 0, & o_i \leqslant 0.5 \\ 1, & o_i > 0.5 \end{cases} \tag{6}$$

其中，o_i 为生成网络的输出向量的第 i 个特征值，b_i 为对应特征二值化结果。算法 2 展示了二值化处理的过程。

算法 2 二值化处理算法

输入 生成样本集 B_{gen}

输出 二值化生成样本集 B'

1) 定义一个 $n \times m$ 的数组 B'，并将所有元素赋值为 1
2) for $i \leftarrow 1,\cdots,n$ do
3) for $j \leftarrow 1,\cdots,m$ do
4) $b_j^{(i)} = (o_j^{(i)} > 0.5)?b_j^{(i)} : 0$
5) end for
6) end for

运用算法 1 和算法 2，能够生成近似良性样本的生成样本，构建良性样本库，为恶意样本提供良性扰动。

4 基于对数回溯法的扰动删减算法

由于攻击者在攻击的过程中需要对检测器的结果进行查询，以判断攻击的有效性。减少查询目标检测器的次数可以防止被目标检测器发现其攻击行为而拒绝提供服务。攻击者在制作对抗样本时倾向于降低攻击成本、减少扰动数量和提高查询效率。为了从攻击者的角度模拟对抗样本的生成，提出基于对数回溯法的扰动删减算法，实现以较少的查询次数和较少的扰动数量生成恶意代码对抗样本。基于对数回溯法的扰动删减算法如算法 3 所示。

算法 3 基于对数回溯法的扰动删减算法

输入 恶意代码扰动样本集 X，扰动集 I，目标检测器的查询结果 $f(\cdot)$

输出 恶意代码对抗样本 $X_{\min}$

1) $S \leftarrow 0$
2) 打乱 X 的顺序
3) for 遍历 X_i in X do
4) $X_{\min} \leftarrow X_i$
5) $R \leftarrow I$
6) loop：
7) while $f(X_{\min}) = 1$ do //检测器结果为良性
8) $S \leftarrow 1$
9) $I \leftarrow R$
10) if $\text{len}(I) \leqslant 1$ then
11) break
12) end if
13) 将扰动集 I 随机等分为 R 和 D，R 为保留的扰动集，D 为删除的扰动集
14) $X_{\min}$ 对应 D 的所有元素修改为 0
15) if $f(X_{\min}) = 0$ then
16) $\text{swap}(R, D)$//交换 R,D
17) $X_{\min}$ 对应 D 的所有元素修改为 0，R 修改为 1
18) end if
19) end while
20) while $f(X_{\min}) = 0$ do
$D' \leftarrow D$
21) if $\text{len}(D') = 0$ then
22) break

23)　　end if
24)　　将扰动集 D' 随机等分为保留的扰动集 R 和删除的扰动集 D
25)　　$X_{\min}$ 对应 R 的元素修改为 1
26)　　if $f(X_{\min})=1$ then
27)　　$I \leftarrow R$
28)　　go to loop
29)　　end if
30)　end while
31) end for
32) if $S=1$ do
33)　return $X_{\min}$
34) else
35)　return null
36) end if

对数回溯法是一种与二分查找思路相似的方法。二分查找法假定原始数据是一个有序的状态，通过数据的中间值与目标值的对比选取执行方向。而在本文中，对数回溯法面向的数据是 API 调用列表，是一种无序的数据。本文模仿二分查找的思路，在迭代过程中，将原始数据随机减少一半。

首先，选取可以躲避恶意代码检测器的扰动样本，计算扰动集。然后，随机减少一半的扰动，加入恶意样本进行查询，并记录当前删除的扰动集。如果查询结果为良性，则重复进行此过程，直至检测器结果为恶意，交换删除数据和当前保留数据，重复上述迭代过程。若交换数据后查询结果仍为恶意，则恢复移除数据的一半数据进行查询，重复迭代，直至检测器结果为良性。当前数据集为删减后所得 API。

5　实验

5.1　数据集

实验使用了 2 个数据集，一个是安卓平台应用程序的数据集（后文简称为安卓数据集），一个是 Windows 可执行应用程序的数据集（后文简称为 Windows 数据集）。

安卓平台应用程序的数据集包含 2 932 个恶意样本和 2 165 个良性样本。其中，恶意样本来自开源恶意程序样本库 VirusShare。良性样本来自小米应用商店，并且所有的良性样本都经过 VirusTotal 平台的检测。VirusTotal 是一个在线检测平台，它通过将文件分发给多种反病毒引擎进行扫描，扫描结果准确率优于单一产品扫描，具有较高的可靠性。

Windows 可执行应用程序的数据集为天池阿里云安全恶意程序检测比赛数据，包含 8 909 个恶意样本和 4 978 个良性样本。数据来自文件（Windows 可执行程序）经过沙箱程序模拟运行后的经过脱敏处理的 API 指令序列。

5.2　实验设置

实验选取随机森林（RF, random forest）、逻辑回归（LR, logistic regression）、决策树（DT, decision tree）、支持向量机（SVM, support vector machine）和多层感知器（MLP, multilayer perceptron）作为目标检测器，验证本文提出的对抗训练驱动的恶意代码检测增强方法的有效性。首先通过对良性样本和对抗样本进行评估，验证对抗样本生成方法的有效性；然后通过对抗训练前后的检测器对比，验证对抗训练的有效性。

在生成对抗网络模型的构建中，生成模型的节点数设置为 100-128-128-196，判别模型的节点数设置为 196-128-128-1。WGAN 模型的实验参数如表 2 所示。

表 2　　WGAN 实验参数

参数名	参数值	描述
epoch	20 000	训练次数
optimizer	RMSProp	优化器
α	0.000 05	学习率
m	64	批次大小
c	0.01	判别器的权值剪裁阈值
ncritic	5	每轮训练中判别迭代次数

5.3　评价指标

5.3.1　生成模型的有效性评估

对于生成模型的有效性评估，使用模型生成样本的良性样本检测率作为评估指标，即生成样本被检测器判断为良性的概率，记作 GEN_TPR，定义如式(7)所示。

$$\text{GEN\_TPR}=\frac{\text{num}\left(f(x')=\text{benign}\right)}{\text{num}\left(G(\boldsymbol{z})\right)} \tag{7}$$

其中，num(·) 为样本的数量，$f(\cdot)$ 为目标检测器对样本的预测结果。G 为 WGAN 的生成器，$\boldsymbol{z}$ 为从标准正态分布的随机噪声向量。

5.3.2　对抗样本的有效性

对抗样本的评估包含攻击成功率和攻击成本这 2 个方面。

攻击成功率也称为绕过率，即对抗样本成功躲避目标恶意代码检测器检测，被检测器识别为良性的概率，记作 ASR，计算方法如式(8)所示。

$$\mathrm{ASR}=\frac{\mathrm{num}\left(f(x')=\mathrm{benign}\right)}{\mathrm{num}(x)} \tag{8}$$

其中，num(·) 为样本的数量，$f(\cdot)$ 为目标检测器对样本的预测结果；x 为恶意样本，x' 为 x 经过扰动后的样本。

对于攻击成本，本文综合考虑了添加的扰动数量和检测器查询次数，计算方法如式(9)所示。

$$\mathrm{cost}=\alpha p+\beta q \tag{9}$$

其中，cost 为攻击成本；p 为扰动数量，即添加的 API 调用数量；q 为恶意代码检测器的查询次数，α 和 β 分别为 p 和 q 的权重，本文设定扰动数量与查询次数占相等权重，即 $\alpha=\beta=0.5$。

5.3.3 对抗训练的有效性

恶意代码检测器性能的评估，采用模型准确率来衡量检测器的检测率，通过对抗训练前后的准确率对比，验证对抗训练的有效性。准确率的计算方法如式(10)所示。

$$\mathrm{Accuracy}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}} \tag{10}$$

其中，TP 表示正确检测的恶意样本数量，TN 表示正确检测的良性样本数量，FN 表示被判断为良性的恶意样本数量，FP 表示被判断为恶意的良性样本数量。

5.4 生成模型的有效性评估

在训练 WGAN 模型的过程中，将生成器的输出样本进行二值化处理后，作为目标检测器的输入，计算良性样本检测率。

在安卓数据集中，实验选取 5.2 节中的 5 种目标检测器，计算不同训练次数下的良性样本检测率，实验结果如图 5 所示。

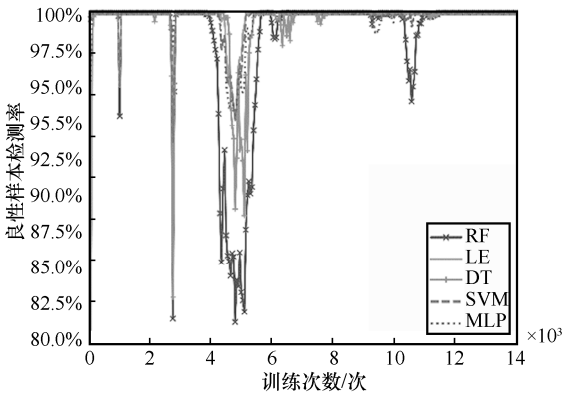


图 5 安卓数据集下 WGAN 生成样本的良性检测率

在 Windows 数据集中，实验选取 DT、MLP 和 LR 这 3 种目标检测器，计算不同训练次数下的良性样本检测率，实验结果如图 6 所示。

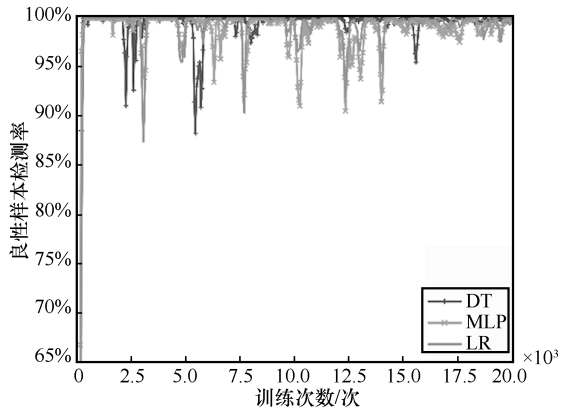


图 6 Windows 数据集下 WGAN 生成样本的良性检测率

由于 WGAN 的训练过程是生成器和判别器的博弈过程，在迭代初始阶段，生成器还没有学习真实样本的分布，良性样本检测率较低，且存在较大波动，当生成器和判别器进行了多次博弈后，生成器模型生成更加满足真实样本分布的数据。

从图 5 和图 6 可以看出，安卓数据集在经过约 12 500 次训练后，在不同的目标检测器下，良性样本检测率均在趋近 100%处保持稳定。Windows 数据集在经过 16 000 次训练后，生成样本的良性检测率维持在 97%以上。基于 WGAN 的生成样本较好地学习了真实良性样本的分布特征，在构建良性样本库上具有一定的有效性。

5.5 对抗样本生成结果评估

5.5.1 攻击成功率评估

在攻击成功率评估的实验中，对 2 个数据集选取与 5.4 节中相同的目标检测器进行实验。实验设置不同的良性样本库规模，评估恶意代码对抗样本的攻击成功率。图 7 和图 8 分别为面向安卓数据集和 Windows 数据集的攻击成功率结果。其中，横坐标表示良性样本库的规模，纵坐标为恶意代码对抗样本攻击成功率。

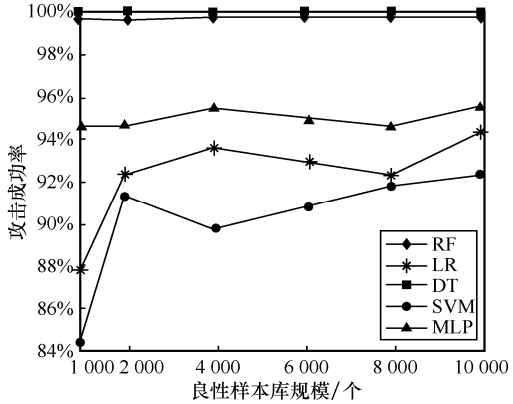


图 7 安卓数据集下不同良性样本库规模的攻击成功率

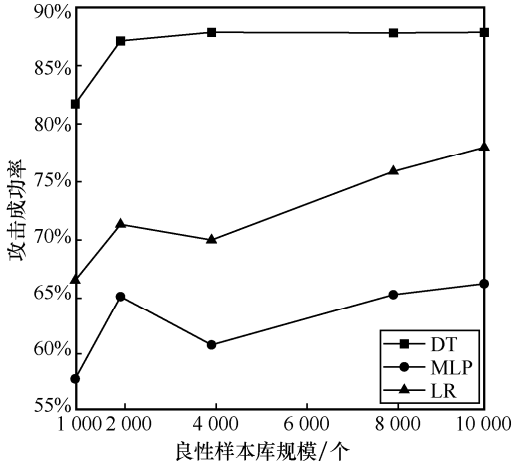


图 8　Windows 数据集下不同良性样本库规模的攻击成功率

在安卓数据集中，当良性样本库规模大于 2 000 个时，攻击成功率维持在一个相对稳定的状态，对抗样本具有较高攻击成功率。

在 Windows 数据集中，MLP 的攻击成功率较低，但在良性样本库规模大于 2 000 个时，仍达到 60%以上的攻击成功率。在 4 000~10 000 个的良性样本库规模中，攻击成功率呈现出一定的增长趋势。当良性样本库规模达到 10 000 时，攻击成功率取得了较好的结果。

实验结果发现，当良性样本库规模为 1 000 个时，2 个数据集下不同的目标检测器的攻击成功率均最低，过少的良性样本会影响当前对抗样本攻击方法的效果。当目标检测器为 DT 时，2 种数据集都具有最高攻击成功率。决策树算法通过对训练数据进行分析，对特征生成规则，利用规则对新数据进行判断。本文的攻击方法通过向恶意样本添加扰动，容易对基于生成规则的决策产生干扰，达到攻击效果。

5.5.2　攻击成本评估

为了验证基于对数回溯法的扰动删减在扰动成本和查询效率上的有效性，本文对安卓数据集进行 3 组实验，即实验 1、实验 2 和实验 3，分别计算安卓数据集恶意代码对抗样本在不同良性样本库规模的攻击成本。实验 1 按照扰动数量从小到大的顺序选取扰动样本，进行检测器查询，直到样本成功躲避恶意代码检测器的检测。实验 2 按照扰动数量从小到大的顺序选取扰动样本，进行检测器查询，并对成功躲避恶意代码检测器的样本进行扰动删减。实验 3 随机选取扰动样本，对成功躲避恶意代码检测器的样本执行基于对数回溯法的扰动删减。3 组实验统一使用 RF 算法作为恶意代码目标检测器，评估在不同规模的良性样本库下，对抗样本生成的攻击成本，采用箱型图展示攻击成本结果，如图 9~图 11 所示，其中，▲表示数据平均值，●表示数据异常值。

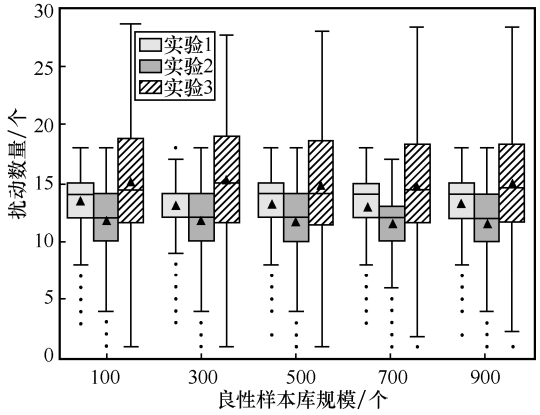


图 9　不同良性样本库规模的扰动数量

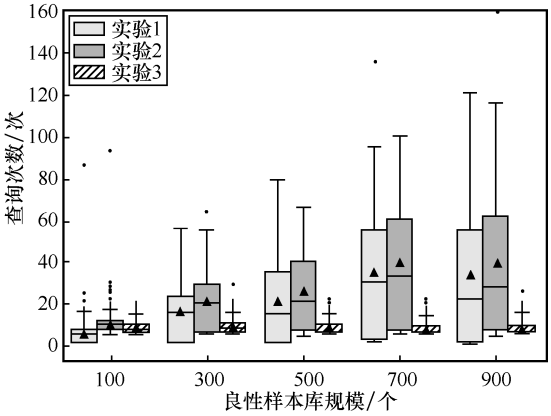


图 10　不同良性样本库规模的查询次数

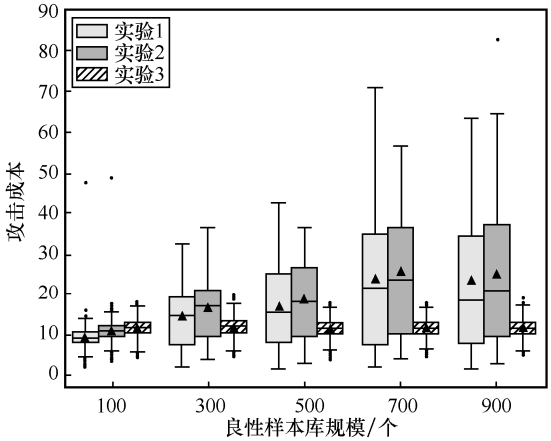


图 11　不同良性样本库规模的攻击成本

在扰动数量上，3 组实验均受良性样本库规模的大小影响不大。实验 1 是在文献[21]的对抗样本生成算法的基础上增加扰动样本的排序操作，扰动数量与文

献[21]一致。实验 2 在实验 1 的基础上进行扰动删减，扰动数量最少。实验 3 采取随机选取扰动样本的方式，扰动数量不稳定，跨度较大。从整体扰动数量的平均值看，3 组实验的扰动数量差距在 5 个以内。

在目标检测器查询次数上，实验 3 的查询次数最少。实验 1 和实验 2 的查询次数随着良性样本库规模的增加而增加，实验 3 的扰动数量受良性样本库规模影响不大，始终保持较低查询次数。

本文通过理论分析，计算基于对数回溯法的扰动删减方法的查询次数。在进行扰动删减时，基于对数回溯法从最大的扰动集开始，不断从扰动样本中删除一半添加的扰动，直到样本被目标检测器错误分类，实现在尽可能少的查询次数内减少扰动数，当删减到只剩一个扰动且删减过程中每次迭代都需要交换保留集和删除集时，所需查询次数最多，设原扰动数为 p，特征维度为 k，则查询次数为 $2\log p \leqslant 2\log k$。而在文献[21]的对抗样本生成算法中，对抗样本生成只关注扰动数量，并不关注查询次数，检测器查询次数等于良性样本库规模大小。

为了综合考虑扰动数量和恶意代码检测器查询次数，采用式(9)的攻击成本计算方法，实验结果如图 11 所示。从图 11 中可知，当良性样本库规模为 100 个时，由于良性样本库规模较小，实验 3 在查询次数上的优势并没有得到体现。并且，由于实验 3 在选择扰动样本时具备一定的随机性，扰动数量存在一定的浮动，在良性样本库规模为 100 个时，攻击成本的平均值略大于实验 1 和实验 2。在其他良性样本库规模中，实验 3 的攻击成本最小，且实验 3 的攻击成本不受良性样本库规模的影响。

在安卓数据集中，当良性样本库规模大于 2 000 时，攻击成功率维持在一个相对稳定的状态，对抗样本具有较高攻击成功率。为了验证本文方法对于不同分类器具有通用性，选取 5.2 节中的目标检测器进行实验，在良性样本库规模为 2 000 个的条件下，计算对抗样本生成的扰动数量与查询次数，结果如图 12 和图 13 所示。

实验结果表明，本文提出的基于对数回溯法的扰动删减算法能够以较小的扰动数量和查询次数，躲避多种恶意代码检测器的检测，对不同的恶意代码检测器具有一定的通用性。

结合图 7 和图 12 中安卓数据集不同目标检测器的攻击成功率和扰动数量实验结果可以发现，具有最高攻击成功率的 DT 检测器对应的平均扰动数量最少，而具有最低攻击成功率的 SVM 检测器对应的平均扰动数量最多。检测器的对抗攻击难度与对抗样本扰动数量具有一定的正相关性。

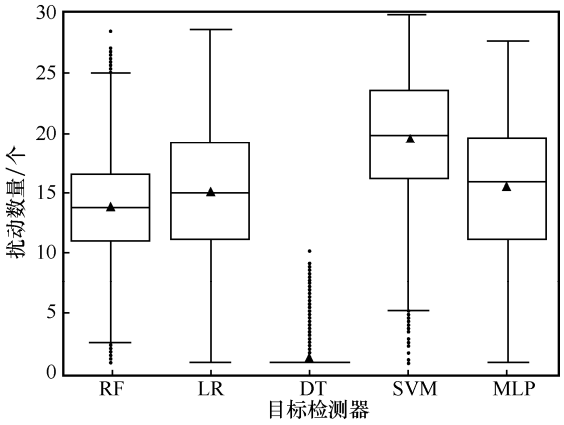


图 12 不同目标检测器的扰动数量

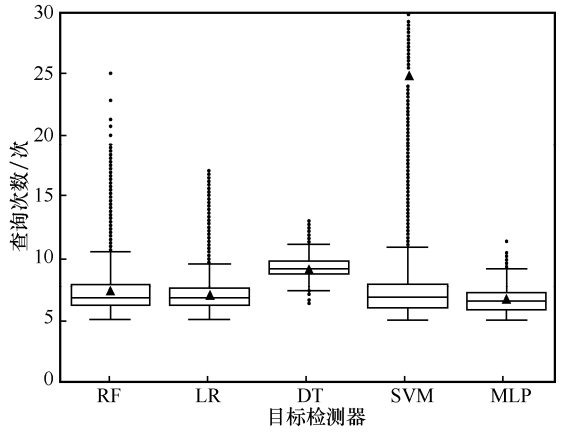


图 13 不同目标检测器的查询次数

在查询次数方面，各目标检测器模型没有体现出明显的差距，攻击过程中的检测器查询次数集中在 5~10 次，具有较高的查询效率。

5.5.3 对抗样本的有效性评估

对于安卓应用程序，本文通过在反汇编文件中文件中插入扰动 API，并利用工具对文件进行重打包和重签名。

以恶意 APK “VirusShare_ffb376be1e8d8311d320f7a107caee9a”为例，利用本文提出的对抗样本生成算法，得到扰动 API，在反汇编文件中添加扰动 API 调用代码，并进行重打包和重签名。最后，利用 VirusTotal 对扰动生成的 APK 进行检测，实验结果表明，与原始恶意样本相比，将扰动生成的对抗样本识别为恶意文件的反病毒引擎数量减少了 10 个，验证了本文所提出的对抗样本生成的有效性。

5.6 对抗训练结果评估

为了验证对抗训练对恶意代码检测器的增强

作用，本文对 2 组数据集分别进行 4 组实验。首先，计算原始目标检测器的准确率，并对目标检测器进行对抗攻击。然后，将生成的对抗样本加入恶意代码检测器进行对抗训练。最后，再次攻击对抗训练后的检测器。

2 个数据集对抗训练前后检测器的准确率如表 3 和表 4 所示。其中，D 为初始检测器，D_{AT} 为对抗训练后的检测器。对 D 和 D_{AT} 检测器进行对抗样本攻击模型分别表示为 AE_1 和 AE_2。表 5 和表 6 为对 D 和 D_{AT} 检测器进行对抗攻击的攻击成功率，即 AE_1 和 AE_2 的攻击成功率。

表 3　对抗训练前后检测器的准确率（安卓数据集）

检测器	RF	LR	DT	SVM	MLP
D	99.93%	99.96%	99.93%	99.65%	99.82%
D_{AT}	100.00%	99.84%	100.00%	99.76%	99.94%

表 4　对抗训练前后检测器的准确率（Windows 数据集）

检测器	DT	MLP	LR
D	96. 90%	96.97%	94.34%
D_{AT}	99.75%	98.33%	94.99%

表 5　对抗训练前后对抗攻击成功率（安卓数据集）

攻击模型	RF	LR	DT	SVM	MLP
AE_1	99.60%	87.97%	99.97%	91.38%	94.68%
AE_2	14.40%	32.54%	89.23%	9.18%	29.78%

表 6　对抗训练前后对抗攻击成功率（Windows 数据集）

攻击模型	DT	MLP	LR
AE_1	87.11%	64.80%	71.09%
AE_2	65.96%	17.56%	23.79%

从表 3 和表 4 可知，对于不同的目标检测器，在经过对抗训练后，检测器的准确率大都得到了一定的提升。结果表明，通过加入生成的恶意代码对抗样本进行对抗训练，提升了目标检测器的恶意代码识别能力。

从表 5 和表 6 可知，对抗训练后的恶意代码检测其攻击成功率明显低于原始恶意代码检测器。结果表明，通过对抗训练，恶意代码检测器识别对抗样本的能力有明显提高，提高了模型的抗干扰能力，增强了模型的稳健性。

6　结束语

针对机器学习模型的脆弱性问题，对恶意代码检测模型的增强方法展开了研究，提出了对抗训练驱动的恶意代码检测增强方法。首先，基于 WGAN 构建面向 API 调用的良性样本库，以扰动的形式添加进恶意样本。然后，基于对数回溯法进行扰动删减以降低攻击成本。最后，基于主动防御思想将生成的对抗样本用于恶意代码检测器的重训练，提高恶意代码检测器防御对抗性攻击的能力。实验表明，本文提出的恶意代码对抗样本生成方法能够以较低的扰动成本和较少的查询次数生成具有较高躲避率的恶意代码对抗样本。通过生成的恶意代码对抗样本来丰富恶意样本库，重训练恶意代码检测模型，能够达到增强模型稳健性和提高模型检测率的效果。

在未来研究工作中，将进一步对本文方法进行改进和完善。一方面，针对对抗样本生成攻击成本最小化问题进行优化，考虑多个因素对攻击成本的影响并赋予合适的权重。另一方面，进一步探索基于代码混淆技术和躲避动态恶意代码检测模型的对抗样本生成方法。

参考文献：

[1] 胡建伟, 车欣, 周漫, 等. 基于高斯混合模型的增量聚类方法识别恶意软件家族[J]. 通信学报, 2019, 40(6): 148-159.
HU J W, CHE X, ZHOU M, et al. Incremental clustering method based on Gaussian mixture model to identify malware family[J]. Journal on Communications, 2019, 40(6): 148-159.

[2] WANG S S, CHEN Z X, YAN Q B, et al. Deep and broad URL feature mining for android malware detection[J]. Information Sciences, 2020, 513: 600-613.

[3] ONWUZURIKE L, MARICONTI E, ANDRIOTIS P, et al. MaMaDroid: detecting android malware by building Markov chains of behavioral models[J]. ACM Transactions on Privacy and Security, 2019, 22(2): 1-34.

[4] 刘奇旭, 王君楠, 尹捷, 等. 对抗机器学习在网络入侵检测领域的应用[J]. 通信学报, 2021, 42(11): 1-12.
LIU Q X, WANG J N, YIN J, et al. Application of adversarial machine learning in network intrusion detection[J]. Journal on Communications, 2021, 42(11): 1-12.

[5] 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.
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.

[6] DEMETRIO L, COULL S E, BIGGIO B, et al. Adversarial EXEmples: a survey and experimental evaluation of practical attacks on machine learning for windows malware detection[J]. ACM Transactions on Privacy and Security, 2021, 24(4): 1-31.

[7] LI D Q, LI Q M, YE Y F, et al. Arms race in adversarial malware detection: a survey[J]. ACM Computing Surveys, 2021, 55(1): 1-35.

[8] MIRZAEIAN A, KOSECKA J, HOMAYOUN H, et al. Diverse knowledge distillation (DKD): a solution for improving the robustness

of ensemble models against adversarial attacks[C]//Proceedings of 2021 22nd International Symposium on Quality Electronic Design. Piscataway: IEEE Press, 2021: 319-324.
[9] KWON H, LEE J. Diversity adversarial training against adversarial attack on deep neural networks[J]. Symmetry, 2021, 13(3): 428.
[10] WANG D R, LI C R, WEN S, et al. Defending against adversarial attack towards deep neural networks via collaborative multi-task training[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 19(2): 953-965.
[11] LI D Q, LI Q M. Adversarial deep ensemble: evasion attacks and defenses for malware detection[J]. IEEE Transactions on Information Forensics and Security, 2020, 15: 3886-3900.
[12] WANG J N, LIU Q X, LIU C G, et al. GAN-based adversarial patch for malware C2 traffic to bypass DL detector[C]//Information and Communications Security. Berlin: Springer, 2021: 78-96.
[13] WANG C Y, ZHANG L L, ZHAO K, et al. AdvAndMal: adversarial training for android malware detection and family classification[J]. Symmetry, 2021, 13(6): 1081.
[14] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.
[15] KIM J Y, BU S J, CHO S B. Malware detection using deep transferred generative adversarial networks[C]//Neural Information Processing. Berlin: Springer, 2017: 556-564.
[16] KIM J Y, BU S J, CHO S B. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders[J]. Information Sciences, 2018, 460/461: 83-102.
[17] LIU Y H, LI J Q, LIU B X, et al. Malware detection method based on image analysis and generative adversarial networks[J]. Concurrency and Computation: Practice and Experience, 2022: doi.org/10.1002/cpe.7170.
[18] SUCIU O, COULL S E, JOHNS J. Exploring adversarial examples in malware detection[C]//Proceedings of 2019 IEEE Security and Privacy Workshops. Piscataway: IEEE Press, 2019: 8-14.
[19] HU W W, TAN Y. Generating adversarial malware examples for black-box attacks based on GAN[J]. arXiv Preprint, arXiv: 1702.05983, 2017.
[20] 王万良, 李卓蓉. 生成式对抗网络研究进展[J]. 通信学报, 2018, 39(2): 135-148.
WANG W L, LI Z R. Advances in generative adversarial network[J]. Journal on Communications, 2018, 39(2): 135-148.
[21] 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(S1): 64-69.
TANG C, ZHANG Y, YANG Y X, et al. DroidGAN: Android adversarial sample generation framework based on DCGAN[J]. Journal on Communications, 2018, 39(S1): 64-69.
[22] ARJOVSKY M, CHINTALA S, BOTTOU L. Wasserstein generative adversarial networks[C]//Proceedings of the 34th International Conference on Machine Learning.[S.l.]: JMLR.org, 2017: 214-223.

[作者简介]

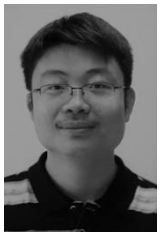
刘延华（1972− ），男，山东济宁人，博士，福州大学副教授、硕士生导师，主要研究方向为网络空间安全、网络数据分析、网络系统故障分析、智能计算及应用等。

李嘉琪（1998− ），女，福建漳州人，福州大学硕士生，主要研究方向为恶意代码检测、网络安全等。

欧振贵（1998− ），男，福建莆田人，福州大学硕士生，主要研究方向为知识图谱融合、实体对齐、知识图谱补全、链接预测等。

高晓玲（1995− ），女，福建漳州人，福州大学硕士生，主要研究方向为网络安全。

刘西蒙（1988− ），男，陕西西安人，博士，福州大学研究员，主要研究方向为隐私计算、密文数据挖掘、大数据隐私保护、可搜索加密等。



MENG Weizhi（1986− ），男，博士，丹麦科技大学副教授，主要研究方向为入侵检测、生物认证、恶意程序检测、人工智能安全、区块链应用等。

刘宝旭（1972− ），男，山东沂水人，博士，中国科学院信息工程研究所研究员、博士生导师，主要研究方向为网络攻防、威胁情报、态势感知、威胁发现、网络溯源等。