计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200300127

面向恶意软件检测模型的黑盒对抗攻击方法

陈晋音　邹健飞　袁俊坤　叶林辉
浙江工业大学信息工程学院　杭州 310023

摘　要　深度学习方法已被广泛应用于恶意软件检测中并取得了较好的预测精度,但同时深度神经网络容易受到对输入数据添加细微扰动的对抗攻击,导致模型输出错误的预测结果,从而使得恶意软件检测失效。针对基于深度学习的恶意软件检测方法的安全性,提出了一种面向恶意软件检测模型的黑盒对抗攻击方法。首先在恶意软件检测模型内部结构参数完全未知的前提下,通过生成对抗网络模型来生成恶意软件样本;然后使生成的对抗样本被识别成预先设定的目标类型以实现目标攻击,从而躲避恶意软件检测;最后,在 Kaggle 竞赛的恶意软件数据集上展开实验,验证了所提黑盒攻击方法的有效性。进一步得到,生成的对抗样本也可对其他恶意软件检测方法攻击成功,这验证了其具有较强的攻击迁移性。

关键词:恶意软件检测;深度学习;生成对抗网络;对抗攻击;黑盒攻击
中图法分类号　TP391

Black-box Adversarial Attack Method Towards Malware Detection

CHEN Jin-yin,ZOU Jian-fei,YUAN Jun-kun and YE Lin-hui
School of Information Engineering,Zhejiang University of Technology,Hangzhou 310023,China

Abstract　Deep learning method has been widely used in malware detection,which also has an excellent performance in the aspect of classification accuracy. Meanwhile,deep neural networks are vulnerable to adversarial attacks in the form of subtle perturbations added on the input data,resulting in incorrect predictive results,such as escaping the malware detection. Aiming at the security of malware detection method based on deep learning,this paper proposes a black-box adversarial attack method towards the malware detection model. First,it uses the generative adversarial net model to generate the adversarial examples. Then,the generated adversarial examples are identified as the pre-set target type to achieve the target attack. Finally,experiments are carried out on the Kaggle competition malware dataset to verify the effectiveness of the black-box attack method. Furthermore,the generated adversarial examples are applied to attack other classification models to testify the strong transfer attack capacity of the proposed black-box attack method.

Keywords　Malware detection,Deep learning,Generative adversarial network,Adversarial attack,Black-box attack

1　引言

恶意软件能够有目的地实现对网络和主机的攻击、信息和隐私的窃取、网络的监视,对主机、网络和隐私的安全性都具有极大的威胁和损害,因此,对其进行检测、分析和预防一直是网络与信息安全研究工作的重点和热点之一。传统恶意软件检测模型的核心方法为签名技术[1],即从软件二进制代码中识别特定的字符串来鉴别恶意性质。但是手动分析通常需要很长时间,研究者提出若干针对恶意软件签名的自动生成方法,如基于特定漏洞[2]、payloads[3]、honeypots[4]等的签名。但是,由于它们针对的是恶意软件的特定方面,恶意软件开发人员可以对其软件稍作修改来创建一个新的未被发现的变体。深度学习的引入使恶意软件检测方法的原有技术框架得到改变,其可以充分利用特征之间的关联性,对提取到的恶意软件特征进行训练,使恶意软件检测模型的识别率得到大幅度提高。

由于深度神经网络的优异性能,已有较多基于深度神经网络模型的恶意软件检测方法。然而,Goodfellow 等[5]和 Szegedy 等[6]发现深度神经网络容易受到对输入数据添加细微扰动形式的对抗攻击。恶意软件检测模型的攻击方法主要是基于 Goodfellow 等提出的快速梯度下降法(Fast Gra-dient Sign Method,FGSM)技术。在已知识别模型结构的前提下基于梯度信息生成对抗样本的方法属于白盒攻击方法,未指定特定攻击目标类型的方法属于无目标攻击方法。在现实的信息与网络攻防场景中,恶意软件检测模型的信息往往被保护起来,无法进行白盒攻击,且其攻击是为了达到特定目标(如使得某类恶意软件被识别成特定类),因此黑盒目标攻击

到稿日期:2020-03-23　返修日期:2020-08-28　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:宁波市"科技创新 2025"重大专项(2018B10063)
This work was supported by the Major Special Funding for "Science and Technology Innovation 2025" of Ningbo,China(2018B10063).
通信作者:陈晋音(chenjinyin@zjut.edu.cn)

的技术研究具有更为重要的现实意义。

针对当下恶意软件具有隐蔽功能且难以被检测、恶意软件检测方法安全性能不高和现实场景中获取模型内部参数困难的问题，本文从攻击者的角度，提出了一种基于生成对抗网络(Generative Adversarial Nets，GAN)的恶意软件检测模型的黑盒攻击技术，本文的主要贡献如下：

(1)提出了一种基于 GAN 的恶意软件检测黑盒攻击方法，在未知检测模型的前提下通过优化扰动生成恶意软件攻击样本来逃避恶意软件检测，从而实现对恶意软件检测模型的黑盒攻击。

(2)通过输入攻击目标样本来使任何类别的恶意软件可以通过对抗训练生成相应目标的对抗样本，实现了黑盒设置下的目标攻击，验证了所提方法具有较强的目标攻击能力。

(3)采用 Kaggle 竞赛的恶意软件数据集，验证所提方法的有效性，并分析了生成对抗样本的攻击迁移性，验证了其对其他分类检测模型具有攻击迁移性。

2 相关工作

2.1 恶意软件检测方法

传统的恶意软件检测工作的主要核心方法是签名技术[1]。签名是一小段字节序列，其对于每一种已知的恶意软件类型都是独一无二的[7]，需要域名专家进行手工制作、更新和上传，效率低且易出错。

针对基于签名的传统检测方法，攻击者研究出了以加密[8]、封装[9]、混淆[8]、多态[10]、变形[10]为代表的逃逸方法。此外，研究者提出了启发式的检测算法，并且这类算法成为恶意软件检测所使用的主要技术之一[11]。基于启发式的检测是使用专家给出的规则或模式来判别软件的恶意性[12]，但模式的建立往往受到专家的主观影响，准确性不够且效率低。

为了加快恶意软件的复制，攻击者使用了 Zeus 等工具箱以每天数以千计的速度生成和更新恶意软件样本，这使得基于签名和启发式的手工检测方法不再适用[13]。基于机器学习的检测方法从原始数据中提取出具有模式信息的特征(如 API 调用、二进制字符和程序行为信息等)，并将其交给分类器进行学习，使用训练好的分类器进行预测。Berlin 等[14]使用了逻辑回归、Kong 等[15]使用了聚类、Annachhater 等[16]使用了隐马尔可夫模型、Garcia 等[17]使用了随机森林、Ye 等[18]使用了限制玻尔兹曼机、Huda 等[19]使用了 K 均值和支持向量机等进行恶意软件的检测，这些都使检测效果和检测速度得到了显著的提高。

然而手工提取的待分析特征一旦被攻击者发现，攻击者可轻易地修改样本并成功逃逸检测。深度学习以分布式的结构自动地学习数据的特征，因此被应用于恶意软件检测领域中，典型的工作如 Wang 等[20]使用了深度神经网络、Pascanu 等[21]使用了循环神经网络、Raff 等[22]使用了卷积神经网络(Convolutional Neural Network，CNN)等，这些都显著缩短了检测工作的时间和降低了劳动成本，且极大地提高了检测的效率和防御的隐蔽性。

2.2 恶意软件检测的攻击方法

恶意软件的白盒攻击方法主要是基于 Goodfellow 等[5]提出的 FGSM 技术。Kolosnjaji 等[23]针对 Raff 等[22]提出的具有代表性的 MalConv 这种基于 CNN 的恶意软件检测技术，通过计算扰动的方向，从待定的所有字节中选出最接近该方向的字节并填充到文件末尾，以此方式迭代直到软件被分类器检测为良性。Kreuk 等[24]针对具有较好检测性能的 CNN 检测器，通过在恶意代码的无效处插入一小段字节(payload)来形成扰动以保护文件功能，并使用一个距离函数解来决软件字节的离散问题，随后将该函数并入损失函数，基于 FGSM 方法通过迭代不断求导并修改 payload 直到恶意软件成功绕过检测器。

在黑盒攻击方面，Hu 等[25]提出一种基于 GAN 的恶意软件生成算法 MalGAN，该算法基于替代检测器拟合黑盒分类器以逼近其决策边界的思路，能够绕过基于机器学习的黑盒检测模型，生成对抗样本。替代检测器和黑盒分类器具有相似结构的先验假设(即不具有对不同黑盒模型的普适性)，否则将导致难以逼近黑盒分类器，降低攻击的效力。Anderson 等[26]将有效的特征信息压缩为 2 350 维的特征向量，基于强化学习技术，让智能体不断选择不会破坏恶意软件功能但有机会使其绕过检测的文件修改操作，对操作引起的环境变化和奖赏情况进行评估并启发下一个操作的选择。此方法操作简单，但是欺骗效率不够理想。Kim 等[27]提出了一种传输深度卷积生成对抗网络，将深度自动编码器(Deep Autoencoder，DAE)应用于 GAN 以稳定训练过程。在训练 GAN 之前，DAE 学习恶意软件特性，生成一般数据，并将数据传输给 GAN 生成器，经过训练的判别器使用转移学习将获取恶意软件特征的能力传递给检测器。但是基于 DAE 提取特征再使用转移学习传递特征的方式增加了成本。Rosenberg 等[28]提出滑动窗口的思路，将 API 调用序列的信息作为特征，通过对滑动窗口移动过程中的每个窗口进行分析来判断整体的恶意性质。该方法也使用替代检测器的思路，通过 FGSM 方法选择插入的 API，不断迭代直到它成功迷惑分类器。但是基于插入 API 来改变调用序列所使用的特征明显，容易被基于此方法更新后的检测器发现，且提取特征的过程增加了成本。Li 等[29]开发了一种黑盒攻击方法 E-MalGAN 来误导装有防火墙的 Android 恶意软件检测系统。该系统通过使用双目标 GAN 来生成对抗样本。双目标 GAN 有两个鉴别器，导致其生成器同时与防火墙和恶意软件检测器竞争。但是此方法仅针对 Android 恶意软件检测系统。

2.3 生成对抗网络模型

生成对抗网络[30]是一种利用博弈的思想进行对抗从而提升模型性能的方法。GAN 同时训练一个生成器和一个判别器，其中生成器(Generator，G)通过学习将噪声的分布映射为逼近于真实样本的对抗样本的分布，而判别器(Discriminator，D)需要从混有真实样本和对抗样本的数据中将对抗样本鉴别出来。整个生成对抗网络的训练过程就是生成器不断模仿真实样本学习如何生成对抗样本，同时判别器不断从样本

中找出对抗样本，直到生成器找到了能够迷惑判别器的方法。训练过程的优化目标为：

$$\min_G \max_D V(D,G)=E_{x\sim p_{data}(x)}[\log D(x)]+E_{Z\sim p_z(\boldsymbol{Z})}[\log(1-D(G(\boldsymbol{Z})))] \tag{1}$$

其中，x 为输入的样本，$p_{data}(x)$ 为真实数据的概率分布，$p_z(\boldsymbol{Z})$ 为噪声的分布。生成器(G)的目标就是找到一个映射关系，使得噪声的分布 $p_z(\boldsymbol{Z})$ 通过映射后能逼近真实数据的概率分布 $p_{data}(x)$，从而迷惑判别器 D。

3 基于 GAN 的黑盒对抗攻击方法

本文提出基于 GAN 的对抗攻击方法，输入恶意软件原始文件，通过生成器将高维噪声映射为图像，使用判别器对生成的图像和目标恶意软件图像进行判别，不断优化扰动，最终成功“欺骗”判别器，生成对抗样本。其系统框架如图 1 所示。

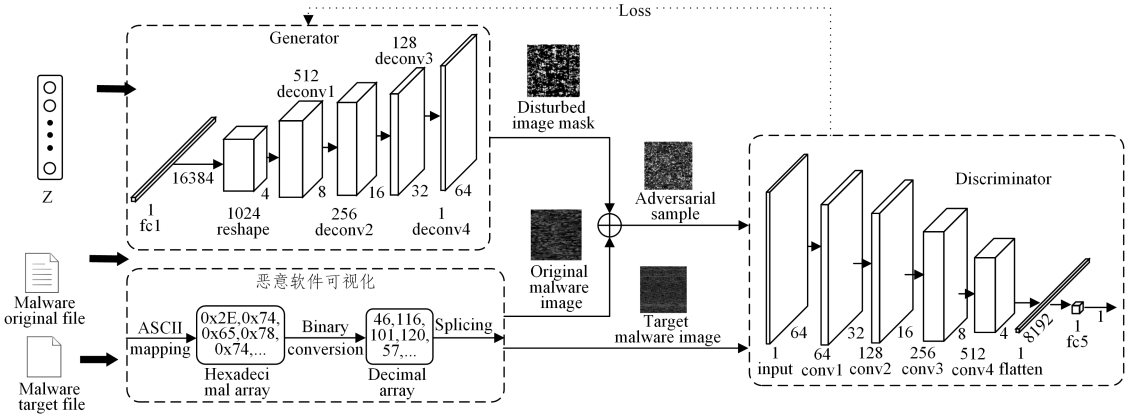

图 1 基于 GAN 的对抗样本自动生成

Fig. 1 Black-box attack example generation based on GAN

本文利用 GAN 的对抗策略优化扰动生成对抗样本。该攻击方法为目标攻击，恶意软件原始文件可通过对抗训练被检测器判定为恶意软件目标文件。虽然两者都是恶意软件，但是属于不同的恶意软件家族。本文系统首先对恶意软件原始文件进行数据预处理，映射为图像；然后生成器学习将均匀分布的噪声 $\boldsymbol{Z}$ 映射为扰动图，将扰动图加到原始样本上生成模仿目标样本的对抗样本，判别器学习如何将原始样本生成的对抗样本和目标样本区分开，双方由此对抗；最后经过迭代优化最终使生成器找到噪声到真实图像的非线性映射，生成对抗样本。

3.1 数据预处理

Nataraj 等[31]首先提出将软件转化为灰度图像的形式，以可视化的方式展示软件的纹理特征，并使用图像领域成熟的分类模型和技术来进行恶意软件的分析和检测。本文将恶意软件转化为图像形式，数据预处理过程如图 2 所示，以后缀为“.asm”的汇编语言软件为例，文件的每个字符都可以在 ASCII 编码中找到。首先通过 ASCII 标准表将其映射为十六进制的数字；然后通过进制转换转化为十进制的数字，使得恶意软件的长字符串转化为十进制的长数组；最后，将十进制长数组以一定的方式进行拼接(如定宽切割拼接、定长切割拼接等)，转换为尺寸为 $m\times m\times 1$ 的恶意软件灰度图像。

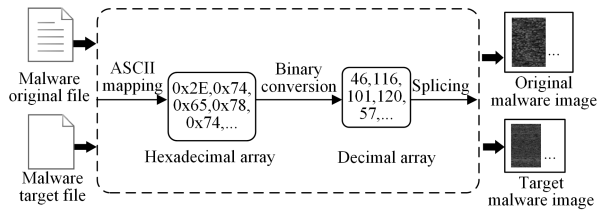

图 2 数据预处理

Fig. 2 Data preprocessing

3.2 生成器(G)

设计生成器的目的是将噪声映射为扰动图像，将扰动图像加入到原始样本中生成对抗样本，使判别器将对抗样本误分为目标样本。生成器以一个 z 维的噪声向量 $\boldsymbol{Z}$ 作为输入，其中，z 是一个超参数，代表噪声的长度。$\boldsymbol{Z}$ 的每个元素都是在[0,1)范围内服从均匀分布的随机数。噪声向量 $\boldsymbol{Z}$ 经过权重为 θ_g 的前馈神经网络，输出扰动图像 $mask$。生成器将扰动图像 $mask$ 加到原始样本中生成对抗样本，并将其被判别器分类的概率和被攻击目标样本的标签(数值 1)的交叉熵作为损失函数，使对抗样本尽可能地模仿目标样本，迷惑判别器。其中，生成器的损失函数定义如下：

$$L_G=E_{x\in S_{Original},Z\sim p_{uniform[0,1)}}\log D_{\theta_d}(G_{\theta_g}(x,\boldsymbol{Z})) \tag{2}$$

其中，x 表示当前样本，$S_{Original}$ 表示原始样本数据集，$p_{uniform[0,1)}$ 表示噪声 $\boldsymbol{Z}$ 服从均匀分布，取值范围为[0,1)，θ_g 表示判别器中网络层的参数。为了训练生成器以达到较好的效果，理论上 L_G 应趋于最小化收敛。但是由于判别器与生成器之间的对抗作用，作为一个训练良好的 GAN，损失函数值将不会平滑收敛，而是一个不断上下波动的过程。以原始样本为第 1 类恶意软件家族，目标样本为第 2 类恶意软件家族为例，生成器的损失函数曲线图如图 3 所示。横坐标表示生成对抗样本的迭代次数 $epoch$，纵坐标表示生成器的损失函数值。损失函数值在迭代过程中不断上下波动，这是因为随着迭代次数的增加，判别器的判别能力增强，使得生成器生成的对抗样本更难以迷惑判别器，生成器的损失函数值增加。随着生成器的不断优化，使得生成的对抗样本更多地成功欺骗判别器，同时又使得损失函数值减小。因此，损失函数曲线整体变化趋势是先基本不变，后增大，再减小。究其原因，是生成器采用将均匀噪声的随机采样映射为扰动图像再叠加原

始样本生成的对抗样本，该对抗样本在初始阶段与原始样本差异较大，使得判别器在开始时未能学习到足够的特征信息，导致判别器性能低于生成器，因此生成器损失函数值变化不明显。随着迭代次数的增加，判别器的判别性能增强，使得生成器生成的对抗样本难以迷惑判别器，生成器的损失函数值增加；当迭代到 400 左右时，生成器不断优化生成较多能够欺骗判别器的对抗样本，使得生成器损失函数值下降，最后围绕一个较低值上下波动。

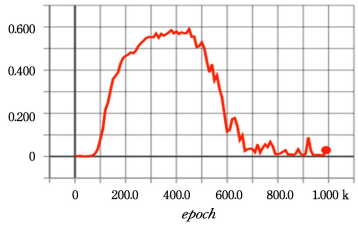

图 3 生成器损失函数曲线图

Fig. 3 Generator loss function curve

3.3 判别器(D)

设计判别器的目的是将生成器生成的对抗样本和目标样本区分开。本文设计的判别器是一个权重为 θ_d 的前馈神经网络。它以目标样本和生成器生成的对抗样本作为输入，分类概率的数值大小作为输出。判别器以当前样本 x 的分类概率和当前样本的标签(原始样本为 0，目标样本为 1)的交叉熵作为损失函数，使得判别器能抵挡住生成器的迷惑，具有良好的分类鲁棒性。判别器的损失函数定义如下：

$$L_D=-E_{x\in S_{Target}}\log(1-D_{\theta_d}(x))-E_{x\in S_{Original}}\log D_{\theta_d}(x) \tag{3}$$

其中，$D_{\theta_d}(x)$表示输入的样本 x 被判别器预测的概率，S_{Target} 表示目标样本数据集，$S_{Original}$ 表示原始样本数据集。判别器的损失函数曲线图如图 4 所示。横坐标表示生成对抗样本的迭代次数 $epoch$，纵坐标表示判别器的损失函数值。损失值在迭代过程中上下波动，整体变化趋势是先增大，后减小，再增大，这刚好与图 3 所示的生成器的损失函数数值变化相反，两者此消彼长，在训练过程中进行对抗。

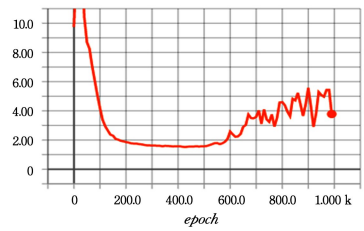

图 4 判别器损失函数曲线图

Fig. 4 Discriminator loss function curve

3.4 算法的基本流程

算法 1 基于 GAN 的对抗攻击方法

输入：原始恶意软件样本 M，目标恶意软件 M′，迭代次数 epoch，最大迭代次数 Max

输出：对抗样本 M_{adv}

1. while(epoch<Max)do
2. 生成器将随机分布 **Z** 映射为扰动图 mask
3. 原始恶意软件样本 M 叠加扰动图 mask 生成对抗样本 M_{adv}
4. 将目标恶意软件 M′与对抗样本 M_{adv} 放入判别器
5. 分别获得生成器的交叉熵 L_G 和判别器的交叉熵 L_D
6. 通过沿梯度$\nabla_{\theta_g}L_G$ 下降来更新生成器的权重 θ_g
7. 通过沿梯度$\nabla_{\theta_d}L_D$ 下降来更新判别器的权重 θ_d
8. endwhile

4 实验与结果分析

4.1 实验数据集

本文所使用的恶意软件数据集源自 Microsoft 公司于 2015 年在 Kaggle 举办的恶意软件分类竞赛[32]。数据包含了 9 个恶意软件家族，共 10 868 个恶意软件样本。数据集的划分如表 1 所列。每一个样本同时包含了“.bytes”后缀的二进制文件和“.asm”后缀的汇编语言文件两种文件类型，本文以汇编语言文件类型为例进行实验。

表 1 Kaggle 恶意软件数据集及划分

Table 1 Kaggle malware data set and partition

Malware family	Malware type	Label	Sample size	Number of samples on training set	Number of samples on testing set
Ramnit	蠕虫	1	1 541	1 233	308
Lollipop	广告	2	2 478	1 982	496
Keilhos_ver3	后门	3	2 942	2 354	588
Vundo	木马	4	475	380	95
Simda	后门	5	42	34	8
Tracur	下载软件	6	751	601	150
Kelihos_ver1	后门	7	398	318	80
Obfuscator.ACY	混淆软件	8	1 228	982	246
Gatak	后门	9	1 013	810	203

为便于后续的实验工作，将数据集按训练集占比约 80%、测试集占比约 20%的比例进行划分，训练集共 8 694 个样本，测试集共 2 174 个样本。

4.2 评价指标

为了对分类实验的结果进行分析，本文选用精度作为评价标准来评估分类器的泛化性能。精度表达式为：

$$acc(f;D)=\frac{1}{n}\sum_{i=1}^{n}I(y_i=f(x_i)) \tag{4}$$

其中，$f(x_i)$为分类器对样本 x_i 的分类结果，y_i 为样本真实的标签，样本总数为 n，精度为分类正确的样本占总样本的比例。

分类器在对抗样本上的精度降低得越多，攻击方法越有效。此外，本文还根据分类器的预测输出与真实标签的组合分成真正例(True Positive，TP)、假正例(False Positive，FP)、真反例(True Negative，TN)和假反例(False Negative，FN)。假设样本总数为 n，则可得 $TP+TN+FP+FN=1$。其中，真正率(True Positive Rate，TPR)和假正率(False Positive Rate，FPR)分别如式(5)和式(6)所示：

$$TPR=\frac{TP}{TP+TN} \tag{5}$$

$$FPR=\frac{FP}{FP+TN} \tag{6}$$

以 TPR 为纵轴，FPR 为横轴可画出接受者操作特性曲

线(Receiver Operating Characteristic, ROC)。进行分类器性能评估时,一个分类器的 ROC 曲线若能包住另一个分类器的 ROC 曲线,则该分类器往往具有更好的泛化性能。进一步通过计算 ROC 曲线与坐标轴围城的面积可以得到 AUC(Area Under ROC Curve),使用 AUC 值可进行更具体的性能评价。

4.3 实验结果分析

为了验证本文提出的基于 GAN 的黑盒攻击的有效性,在上述数据集上进行了恶意软件检测模型的黑盒攻击。本实验中,数据预处理过程将原始恶意软件样本转换为 64×64×1 的恶意软件灰度图像。GAN 模型采用深度卷积生成对抗网络(Deep Convolutional Generative Adversarial Network, DCGAN)[33]结构。深度卷积生成网络相比一般的生成对抗网络,具有生成能力更强大,训练起来更加稳定、容易,生成的样本更加多样化等优点。其中,生成器具体的结构信息如表 2 所列。将长度为 z 的噪声向量通过全连接层和尺度变化操作转换为 4×4×1 024 大小的特征图。通过 4 层卷积核为 3×3、步长为 2 的反卷积层,将噪声转变的特征图一步步进行反卷积操作,直到其和原始数据图具有一样的尺寸。整个生成器网络共 5 层,共 6194304 个带训练参数(除去偏置值和第一个全连接层,且实际训练时由于前向、反向计算和优化器的使用,参数量将有所增加)。

表 2 生成器的网络结构

Table 2 Network structure of generator

Type of network layer	Information	Output characteristic chart	Number of training parameters
Input layer	—	z	—
Full connection layer 1(fc1)	All connected to 16384 neurons	16384	16384×z
Scale transformation layer	Transform to feature map format	(4,4,1024)	—
Deconvolution layer 2(deconv2)	Convolution kernel: 3×3×1024×512 Step:2 Activation function: ReLU	(8,8,512)	4718592
Deconvolution layer 3 (deconv3)	Convolution kernel: 3×3×512×256 Step:2 Activation function: ReLU	(16,16,256)	1179648
Deconvolution layer 4 (deconv4)	Convolution kernel: 3×3×256×128 Step:2 Activation function: ReLU	(32,32,128)	294912
Deconvolution layer 5 (deconv5)	Convolution kernel: 3×3×128×1 Step:2 Activation function: ReLU	(64,64,1)	1152

其中,判别器具体的结构信息如表 3 所列。首先输入一个通道的对抗样本图像和目标样本图像,然后通过 4 层卷积核为 4×4、步长为 2 的卷积层逐层提取特征,最后通过全连接层输出分类概率的数值大小。网络共 5 层,共 2161728 个带训练参数(除去偏置值,且实际训练时由于前向、反向计算和优化器的使用,参数量将有所增加)。

表 3 判别器的网络结构

Table 3 Network structure of discriminator

Type of network layer	Information	Output characteristic chart	Number of training parameters
Input layer	—	(64,64,1)	—
Convolution layer 1 (conv1)	Convolution kernel: 4×4×1×64 Step:2 Activation function: ReLU	(32,32,64)	1024
Convolution layer 2 (conv2)	Convolution kernel: 4×4×64×128 Step:2 Activation function: ReLU	(16,16,128)	131072
Convolution layer 3 (conv3)	Convolution kernel: 4×4×128×256 Step:2 Activation function: ReLU	(8,8,256)	524288
Convolution layer 4 (conv4)	Convolution kernel: 4×4×256×512 Step:2 Activation function: ReLU	(4,4,512)	2097152
Full connection layer(fc5)	Fully connected to 1 output for classification	1	8192

本实验设置生成器和判别器的总迭代训练次数为 1000,其中,每训练一次判别器将训练 k 次生成器,k 是待研究超参数。高维空间的噪声是一维向量,其长度 z 是另一待研究的超参数。训练结束后随机选取 500 个测试样本(不足 500 的使用所有测试样本)用于生成对抗样本以进行黑盒攻击,进而测试分类器的精度变化情况。控制噪声的大小 z 和生成器的单次迭代次数 k 为控制的参数变量,实验结果如表 4 所列。

表 4 基于 GAN 的扰动生成攻击的实验结果

Table 4 Experimental results of disturbance generation attack based on GAN

Experimental group number	Parameter z	Parameter k	Malware image recognition rate/%
0	—	—	88.17±0.87
1	100	20	18.78±0.88
2	1000	20	**16.80±0.70**
3	1000	10	18.04±1.03
4	10000	10	21.08±0.04
5	10000	20	24.26±1.10
6	10000	40	20.63±0.89
7	10000	60	22.95±0.94

从参数 z 代表的高维空间的扰动初始大小来看,对表 4 列出的第 1、第 2、第 5 组实验结果进行对比,高维空间的扰动大小与扰动图像的攻击性能无明显的相关性。从参数 k 代表的生成器单次迭代次数来看,对比第 4—第 7 组实验结果可以发现,当生成器的单次迭代次数在 10～60 之间时,生成的对抗样本的攻击效力与生成器的迭代次数之间并无明显的正比关系,即并非对生成器进行更多的训练得到的对抗样本就更具攻击力。当 z=1000,k=20 时,恶意软件图像分类器的攻击效果达到最佳,分类精度降低到 16.80%左右。

原始恶意软件图像的混淆矩阵的热力图如图 5 所示。该

图横轴为预测的类别，纵轴为样本真正的标签，每个区块代表某类真正标签的样本被预测为某一类别占所有样本的比例，颜色越深，占比越大，最理想的情况为所有的比例都分布在热力图的对角线上。原始样本基本上被分到了相应的恶意软件家族中，只有第 4 类和第 5 类预测为相应恶意软件家族的比例较低，主要原因是第 4、第 5 类恶意软件家族的训练样本量过少，这使得分类器无法很好地学习它们的模式信息。

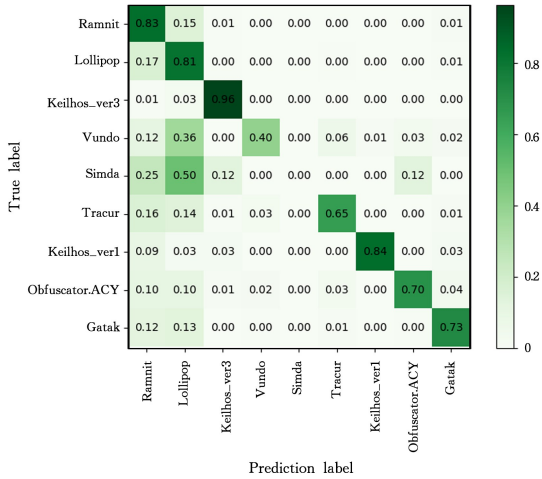

图 5　原始恶意软件图像的混淆矩阵

Fig. 5　Confusion matrix of original malware image

基于 GAN 的扰动生成攻击在参数 $k=20, z=1\ 000$ 时，其恶意软件图像生成的对抗样本的混淆矩阵的热力图如图 6 所示。由图 6 可知，添加扰动后，大多数对抗样本都被分类到第 1 个和第 2 个恶意软件家族中，少量对抗样本被分到第 9 个恶意软件家族中。由数据集划分表表 1 可知，导致此分类结果的原因是：一方面第 4－第 7 这 4 个恶意软件家族的训练样本量过少，这使得分类器无法很好地学习它们的模式信息；另一方面恶意软件中第 1－第 5、第 8－第 9 等家族的纹理特征非常相似，决策边界不够明显，导致扰动能较好地使对抗样本突破决策边界进而被错分到其他的类别中。

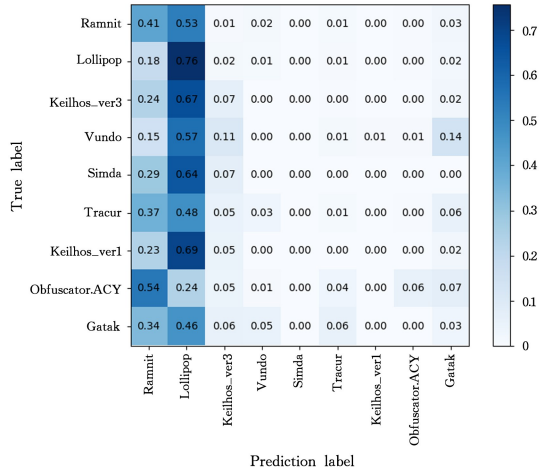

图 6　当 $k=20, z=1\ 000$ 时，对抗样本的混淆矩阵

Fig. 6　Confusion matrix of adversarial examples when $k=20, z=1\ 000$

基于 GAN 的扰动生成攻击在 $k=20, z=1\ 000$ 时生成的对抗样本的 ROC 曲线和 AUC 值如图 7 所示。

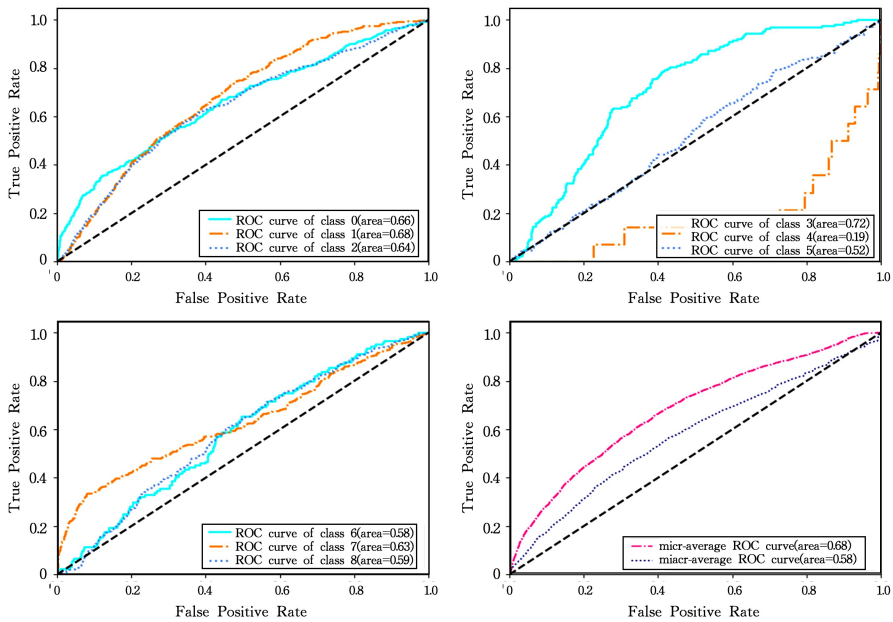

图 7　当 $k=20, z=1\ 000$ 时对抗样本的 ROC 曲线和 AUC 值

Fig. 7　ROC curve and AUC value of adversarial examples when $k=20, z=1\ 000$

恶意软件图像中，所有类型的对抗样本都达到了较好的攻击效果，这使得当前类别的样本的错分概率非常高，多数恶意软件家族的对抗样本的 ROC 曲线甚至低于随机分类的 ROC 曲线（黑色虚线），其 AUC 值低于 0.5。

基于 GAN 的扰动生成方法不仅实现了黑盒攻击，而且还能够使分类器将对抗样本分类到预先设定的类别中，即进行目标攻击，同时攻击效果非常理想，分类器的分类精度降低到 16.80%左右。

4.4　攻击的迁移性

本文提出的基于 GAN 的黑盒攻击生成的对抗样本具有攻击的迁移性。本文分别在机器学习方法（如随机森林（Random Forest，RF）、逻辑回归（Logistic Regression，LR）、

决策树(Decision Tree,DT)、支持向量机(Support Vector Machine,SVM)、KNN 分类器(*K*-Nearest Neighbor classifier,KNN)和梯度增强决策树(Gradient Boosting Decision Tree,GBDT))以及深度学习模型(如 CNN)上验证对抗样本的迁移性。其中,随机森林包含 500 棵决策树,逻辑回归使用 L1 正则化,支持向量机中的核函数采用高斯核,梯度增强决策树包含 200 棵决策树,CNN 为 LeNet-5 模型[34],模型结构如表 5 所列。

表 5 CNN 模型的网络结构

Table 5 Network structure of CNN model

Type of network layer	Information	Output characteristic chart	Number of training parameters
Input layer	—	(32,32,1)	—
Convolution layer 1 (conv1)	Convolution kernel: 5×5×1×6 Step:1 Activation function: ReLU	(28,28,6)	150
Pooling layer 1 (pool1)	filter:2×2×6×6 Step:2	(14,14,6)	144
Convolution layer 2 (conv2)	Convolution kernel: 5×5×6×16 Step:1 Activation function: ReLU	(10,10,16)	2 400
Pooling layer 2 (pool2)	filter:2×2×16×16 Step:2	(5,5,16)	1 024
Full connection layer 1(fc1)	Fully connected to 120 output	120	400
Full connection layer 2 (fc2)	Fully connected to 84 output	84	120
Full connection layer 3 (fc3)	Fully connected to 10 output for classification	10	84

首先,将恶意软件图像转化为一维数组并将其作为恶意软件的图像特征放入分类器,对测试样本和对抗样本进行预测,计算识别率。识别率的定义如下:

$$识别率=\frac{正确识别的样本数}{总样本数} \tag{7}$$

只要分类器识别的对抗样本与原始样本识别结果不同,就视为迁移成功。因此迁移成功率为:

$$迁移成功率=\frac{迁移成功的样本数}{总样本数} \tag{8}$$

表 6 列出了本文提出的基于 GAN 的黑盒攻击生成的对抗样本的迁移性实验结果。由表 6 可知,所使用的机器学习分类器对原始样本的识别率都较高,其中,随机森林的原始样本识别率达到 98.50%,这说明机器学习分类器对于提取的恶意软件图像特征具有较好的分类效果;深度学习模型的原始样本识别率达到 98.90%,优于其他机器学习方法。在对抗样本的识别率中,逻辑回归和支持向量机较低,分别为 0 和 2.27%,原因可能是逻辑回归和支持向量机的结构与神经网络非常相似。虽然随机森林、决策树和梯度增强决策树与神经网络具有的不同的结构使得对它们的对抗样本识别率相比其他分类器更高,但是梯度增强决策树的对抗样本检测率为 7.89%,仍然很低。深度学习模型的对抗样本识别准确率为 25.90%,同时,实验所使用的机器学习分类器的迁移成功率达到 96.4%以上,其中,KNN 分类器的迁移成功率达到 100%,极高的迁移成功率说明了本文提出的黑盒攻击方法产生的对抗样本能极大地影响其他分类器的识别结果,这说明大部分分类器几乎不能检测出生成器产生的任何恶意软件图像,但所提黑盒攻击方法已经成功地绕过了这些机器学习分类器。

表 6 对抗样本迁移性实验结果

Table 6 Experimental results on migration of adversarial examples

(单位:%)

Classifier	Origina lsamples recognition rate	Adversarial samples recognition rate	Migration success rate
SVM	96.50	0	97.55
KNN	97.00	28.57	100
LR	97.40	2.27	97.41
DT	97.70	20.80	98.21
GBDT	98.30	7.89	96.93
RF	98.50	0	96.40
CNN	98.90	25.90	87.70

结束语 本文提出了面向恶意软件检测模型的黑盒对抗攻击方法。实验结果表明,本文提出的基于 GAN 的黑盒攻击方法能够攻击基于深度学习的恶意软件检测器,使其将对抗样本错分到预设的类别中,实现目标攻击,同时产生的对抗样本具有攻击迁移性。但是本文采用的生成对抗样本的方法是生成随机扰动图并将其叠加到原始恶意软件图像上,这可能改变了原始恶意软件的功能。在接下来的工作中,我们将优化添加扰动方法,保证恶意软件的功能性;优化训练方法,如增加策略,使模型能在决策边界上进行对抗并生成对抗样本,以提高攻击的效率和精度;同时针对该攻击方法提出相应的防御机制,提高恶意软件检测的安全性。

参 考 文 献

[1] KEPHART J O. Automatic extraction of computer virus signatures[C]// Proc. 4th Virus Bulletin International Conference. Abingdon, England, 1994: 178-184.

[2] BRUMLEY D, WANG H, JHA S, et al. Creating Vulnerability Signatures Using Weakest Preconditions[C]// 20th IEEE Computer Security Foundations Symposium(CSF'07). Venice, 2007: 311-325.

[3] WANG K, CRETU G, STOLFO S J. Anomalous Payload-Based Worm Detection and Signature Generation[C]// Recent Advances in Intrusion Detection. RAID, 2005: 227-246.

[4] PORTOKALIDIS G, SLOWINSKA A, BOS H. Argos: an emulator for fingerprinting zero-day attacks[C]// EuroSys 2006. 2006.

[5] GOODFELLOW I J, SHLENS J, SZEGEDY C. Explaining and harnessing adversarial examples[J]. arXiv: 1412.6572, 2014.

[6] SZEGEDY C, ZAREMBA W, SUTSKEVER I, et al. Intriguing properties of neural networks[J]. arXiv: 1312.6199, 2013.

[7] YE Y, LI T, ZHU S, et al. Combining file content and file relations for cloud based malware detection[C]// Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 222-230.

[8] SUNG A H, XU J, CHAVEZ P, et al. Static analyzer of vicious

executables(save)[C]// 20th Annual Computer Security Applications Conference. IEEE,2004:326-334.
[9] KENDALL K,MCMILLAN C. Practical malware analysis[C]// Black Hat Conference. 2007:10.
[10] BAZRAFSHAN Z,HASHEMI H,FARD S M H,et al. A survey on heuristic malware detection techniques[C]// The 5th Conference on Information and Knowledge Technology. IEEE, 2013:113-120.
[11] YE Y,LI T,ADJEROH D,et al. A survey on malware detection using data mining techniques[J]. ACM Computing Surveys (CSUR),2017,50(3):41.
[12] EGELE M,SCHOLTE T,KIRDA E,et al. A survey on automated dynamic malware-analysis techniques and tools[J]. ACM computing surveys(CSUR),2012,44(2):6.
[13] FOSSI M,JOHNSON E,MACK T,et al. Symantec global Internet security threat report trends for 2008[J]. Methodology, 2005(April):1-3.
[14] BERLIN K,SLATER D,SAXE J. Malicious behavior detection using windows audit logs[C]// Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security. ACM, 2015: 35-44.
[15] KONG D, YAN G. Discriminant malware distance learning on structural information for automated malware classification [C]// Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM,2013: 1357-1365.
[16] ANNACHHATRE C,AUSTIN T H,STAMP M. Hidden Markov models for malware classification[J]. Journal of Computer Virology and Hacking Techniques,2015,11(2):59-73.
[17] GARCIA F C C,MUGA I I,FELIX P. Random forest for malware classification[J]. arXiv:1609.07770,2016.
[18] YE Y,CHEN L,HOU S,et al. DeepAM:a heterogeneous deep learning framework for intelligent malware detection [J]. Knowledge and Information Systems,2018,54(2):265-285.
[19] HUDA S,MIAH S,HASSAN M M,et al. Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data[J]. Information Sciences, 2017, 379:211-228.
[20] WANG Q,GUO W,ZHANG K,et al. Adversary resistant deep neural networks with an application to malware detection[C]// Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2017: 1145-1153.
[21] PASCANU R,STOKES J W,SANOSSIAN H,et al. Malware classification with recurrent networks[C]// 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE,2015:1916-1920.
[22] RAFF E,BARKER J,SYLVESTER J,et al. Malware detection by eating a whole exe[C]// Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence. 2018.
[23] KOLOSNJAJI B,DEMONTIS A,BIGGIO B,et al. Adversarial malware binaries:Evading deep learning for malware detection in executables[C]// 2018 26th European Signal Processing Conference(EUSIPCO). IEEE,2018:533-537.
[24] KREUK F,BARAK A,AVIV-REUVEN S,et al. Deceiving end-to-end deep learning malware detectors using adversarial examples[J]. arXiv:1802.04528,2018.
[25] HU W,TAN Y. Generating adversarial malware examples for black-box attacks based on GAN[J]. arXiv:1702.05983,2017.
[26] ANDERSON H S,KHARKAR A,FILAR B,et al. Evading machine learning malware detection[R]. USA:Black Hat. ,2017.
[27] 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,2018,460:83-102.
[28] ROSENBERG I,SHABTAI A,ROKACH L,et al. Generic black-box end-to-end attack against state of the art API call based malware classifiers[C]// International Symposium on Research in Attacks, Intrusions, and Defenses. Springer, Cham, 2018:490-510.
[29] LI H,ZHOU S,YUAN W,et al. Adversarial-Example Attacks Toward Android Malware Detection System[J]. IEEE Systems Journal,2019,14(1):653-656.
[30] GOODFELLOW I,POUGET-ABADIE J,MIRZA M,et al. Generative adversarial nets[C]// Advances in Neural Information Processing Systems. 2014:2672-2680.
[31] NATARAJ L,KARTHIKEYAN S,JACOB G,et al. Malware images:visualization and automatic classification[C]// Proceedings of the 8th International Symposium on Visualization for Cyber Security. ACM,2011:4.
[32] RONEN R,RADU M,FEUERSTEIN C,et al. Microsoft malware classification challenge[J]. arXiv:1802.10135,2018.
[33] RADFORD A,METZ L,CHINTALA S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv:1511.06434,2015.
[34] LECUN Y,BOTTOU L,BENGIO Y,et al. Gradient-based learning applied to document recognition[C]// Proceedings of the IEEE. 1998:2278-2324.

CHEN Jin-yin, born in 1982, Ph.D, associate professor. Her main research interests include artificial intelligence security, data mining and intelligent computing.