基于遗传算法的恶意代码对抗样本生成方法

闫 佳 闫 佳 聂楚江 苏璞睿*

(中国科学院大学计算机科学与技术学院 北京 100190)

(中国科学院软件研究所可信计算与信息保障实验室 北京 100190)

摘 要：机器学习已经广泛应用于恶意代码检测中，并在恶意代码检测产品中发挥重要作用。构建针对恶意代码检测机器学习模型的对抗样本，是发掘恶意代码检测模型缺陷，评估和完善恶意代码检测系统的关键。该文提出一种基于遗传算法的恶意代码对抗样本生成方法，生成的样本在有效对抗基于机器学习的恶意代码检测模型的同时，确保了恶意代码样本的可执行和恶意行为的一致性，有效提升了生成对抗样本的真实性和模型对抗评估的准确性。实验表明，该文提出的对抗样本生成方法使MalConv恶意代码检测模型的检测准确率下降了14.65%；并可直接对VirusTotal中4款基于机器学习的恶意代码检测商用引擎形成有效的干扰，其中，Cylance的检测准确率只有53.55%。

关键词：恶意代码检测；机器学习；对抗样本

中图分类号：TP309.5 **文献标识码**：A **文章编号**：1009-5896(2020)09-2126-08

DOI: 10.11999/JEIT191059

Method for Generating Malicious Code Adversarial Samples Based on Genetic Algorithm

YAN Jia YAN Jia NIE Chujiang SU Purui

(*School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing* 100190, *China*)

(*Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing* 100190, *China*)

Abstract: Machine learning is widely used in malicious code detection and plays an important role in malicious code detection products. Constructing adversarial samples for malicious code detection machine learning models is the key to discovering defects in malicious code detection models, evaluating and improving malicious code detection systems. This paper proposes a method for generating malicious code adversarial samples based on genetic algorithms. The generated samples combat effectively the malicious code detection model based on machine learning, while ensuring the consistency of the executable and malicious behavior of malicious code samples, and improving effectively the authenticity of the generated adversarial samples and the accuracy of the model adversarial evaluation are presented. The experiments show that the proposed method of generating adversarial samples reduces the detection accuracy of the MalConv malicious code detection model by 14.65%, and can directly interfere with four commercial machine-based malicious code detection engines in VirusTotal. Among them, the accuracy rate of Cylance detection is only 53.55%.

Key words: Malware detection; Machine learning; Adversarial sample

1 引言

机器学习算法在恶意代码检测领域已广泛应用[1]，并取得了显著的检测效果。相比于传统的检测方法，机器学习可通过海量样本的训练，抽取恶意样本的共性特征，形成恶意样本的检测模型。Saxe等人[2]通过统计PE元数据、字符串等特征，利用深度神经网络模型，在PE文件的恶意性检测上达到了95%的准确率、0.1%的误报率。Arp等人[3]提出了一种高效可解释的恶意软件检测模型，利用安卓文件的权限请求、硬件模块等静态特征，通过

收稿日期：2019-12-31；改回日期：2020-05-30；网络出版：2020-07-21

*通信作者： 苏璞睿 purui@iscas.ac.cn

基金项目：国家自然科学基金(61902384, U1836117, U1836113)

Foundation Items: The National Natural Science Foundation of China (61902384, U1836117, U1836113)

支持向量机(Support Vector Machine, SVM)模型，达到了94%准确率。Raff等人[4]也从静态分析的角度提出了基于PE头的恶意代码检测方法，并对基于n-gram的恶意代码检测方法[5]进行了评估。基于机器学习的恶意代码检测模型目前已在多款安全产品中广泛应用，文献[6,7]都有成熟的基于机器学习的商用恶意代码检测引擎。

虽然机器学习算法提升了恶意代码检测的效率和未知代码的检测能力，但其模型往往是通过有限的样本数据训练形成，存在机器学习模型过拟合导致的泛化局限性，任何偏离训练集统计分布的样本都可以绕过检测。攻击者可以利用对抗样本消除一些恶意特征属性，实现检测逃逸。针对样本对抗逃逸的问题，国内外学者开展研究，并取得了一系列成果，梁光辉等人[8]利用沙箱规避行为在代码进化过程中产生的动静态语义上的差异，设计了基于相似度差异的判定算法，提升了检测对抗样本的能力；Grosse等人[9]利用DREBIN[3]的数据集，训练了多个分类器进行鲁棒性测试，发现采用固定维度特征的恶意代码检测模型易受对抗样本攻击的结论；Xu等人[10]提出一种自动生成对抗样本的方法，实现了对基于PDF结构特征的机器学习恶意软件分类器的逃逸。但是，Grosse的方法是基于白盒的对抗模式，需要已知恶意软件分类器架构作为前提条件；Xu的方案则是针对数据型文件PDF实现的，不能很好地延伸用于其他文件类型的恶意软件检测。这导致他们的方案无法适用于真实的网络环境。

近期也有学者采用黑盒的方式实现了面向基于机器学习的恶意代码检测对抗技术。Hu等人[11,12]先后提出了两个黑盒对抗方案，利用替补模型模拟目标模型的检测功能，一个是基于GAN的方式自动生成对抗样本的方案，另一个是通过插入无关API序列实现对抗样本生成的方案。但是，二者都存在局限性：前者需要已知目标模型采用的特征类型，后者无法保证对抗样本与原始样本的行为一致性。

因此，本文针对基于机器学习技术的恶意代码检测模型易受对抗干扰的局限性，提出了一种基于黑盒的对抗样本生成技术。在保留PE文件原始行为的条件下，通过改写文件结构、添加对抗信息，使基于机器学习的恶意代码检测引擎对样本产生误判(如将良性样本误判为恶意代码，将恶意样本误判为良性程序)。实验结果证明，利用本文的方法可以使MalConv[13]恶意代码检测模型误报率升高，检测准确率从98.88%下降至84.23%；将被MalConv模型误报的239个对抗样本上传VirusTotal[14]，其中，基于机器学习的Cylance的检测准确率为53.55%。

本文剩余章节安排如下：第2节简述机器学习模型等研究背景和相关工作；第3节描述基于遗传算法的恶意代码对抗样本生成方法；第4节进行实验验证和分析；第5节总结本文工作。

2　研究背景和相关工作

机器学习可用于解决恶意代码的检测问题，常用的模型包括：循环神经网络(RNN)，深度神经网络(DNN)和卷积神经网络(CNN)。下面将分别介绍基于机器学习的恶意代码检测模型和面向恶意代码检测模型的对抗样本生成等方面的相关工作。

2.1　基于机器学习的恶意代码检测模型

基于机器学习的恶意代码检测引擎一般从动态分析和静态分析两个角度切入：动态分析是指将样本在沙箱中执行后，基于动态行为序列进行特征提取；静态分析是在不执行样本的情况下，直接从二进制文件中提取重要特征。

恶意代码的动态行为信息可以抽象为一个独立的高级事件——对构成输入序列的各种组件的API调用进行规范化表示，有学者将这样的高级事件视为时序分类问题。Pascanu的团队[15]提出利用自然语言建模的思想，将RNN模型用于预测一串API调用序列是否具有恶意性。Kolosnjaji等人[16]提出一种将n-gram的卷积与完整的序列信息建模相结合，利用长短时记忆(LSTM)算法进行样本恶意性预测。Huang等人[17]提出一种基于DNN的多任务恶意代码分类系统——MtNet，通过沙箱动态执行PE文件，利用3-gram生成API调用的动态行为短序列，并用相互信息[18]构造特征向量空间，利用DNN训练得到检测准确率99.51%的恶意代码检测模型。

在静态分析方面，近年有学者[19,20]提出利用可视化的思想，将二进制代码转化为灰度图像，并使用机器学习对恶意代码进行分类，取得了良好的检测效果。Liu等人[21]在基于机器学习的检测器上使用数据可视化和对抗训练的方式来检测不同类型的恶意代码及其变体，提升了检测器检测对抗样本的能力。Invincea实验室的Saxe等人[2]通过静态分析的方法，提取4组离散型特征构建特征向量空间，其中两组特征源于PE头文件、导入表，两组特征源于字符串、字节在样本中的分布统计，利用DNN训练得到95%的检测准确率。NVIDIA公司研究团队[13]提出直接将PE文件的原始字节序列作为输入，通过卷积神经网络(CNN)学习、训练后，得到MalConv恶意代码检测模型。与只提取PE文件部分特征作为输入的检测模型[2]相比，该方法的特征向量空间表示方法将原本离散的特征联系起来，能检测任意大小的样本，并避免遗漏重要特征，使

得预测结果更准确。本文后续将主要针对MalConv检测模型进行对抗样本生成方法的有效性实验评估。

2.2 面向恶意代码检测机器学习模型的对抗样本生成

目前，基于机器学习的对抗技术已经取得一定发展，有基于白盒实现的对抗技术，也有基于黑盒利用替补模型检测的反馈信息进行调整的对抗策略。下面将详细阐述其进展。

基于白盒的对抗技术研究中，Skylight团队[22]通过逆向分析Cylance公司恶意代码AI检测引擎产品，在恶意样本尾部添加PE文件的可打印字符串，使得Wanncry[23]等恶意样本能逃逸Cylance PROTECT等产品的检测。Grosse等人[9]则在假设已知检测模型结构和参数权重信息的情况下，基于梯度下降算法，通过不同超参数生成的对抗样本使模型的误报率达到40%～85%。

然而，在真实网络中基于白盒方式实现对抗攻击并不可取，因为面对的大部分恶意代码检测引擎都是闭源的。Dang等人[24]根据PDF文件格式的对象特征，使用黑盒修改器变形样本，在不需要替补模型的条件下，基于爬山算法实现了到对抗样本的生成。

Hu等人[11]基于GAN的方式，通过数据训练得到一个能模拟目标模型功能的训练替补模型，然后利用替补模型的反馈生成对抗样本。但是，该方案的一个前提条件是需要知道目标恶意代码分类模型选用的特征类型。

Hu等人[12]还提出了一种基于动态分析插入无关API的对抗方案，该方案可以实现针对基于RNN的恶意代码检测引擎的逃逸。但是，该方案在进行数据预处理时简化了输入数据：首先，缩减了样本循环逻辑中API重复调用的次数；其次，输入模型的API调用序列是截取的长度为1000的原始API调用序列。这使得该方案无法保持PE文件的可执行与原始恶意行为之间的一致性。

综上所述，我们发现目前面向基于机器学习的恶意代码检测引擎的对抗技术都存在一定的局限性：(1)只能针对特定的数据型文件(如PDF)实现逃逸；(2)需预先知道目标模型的特征类型、模型参数等信息；(3)无法保持对抗样本与原始样本的行为一致性。本文针对前述黑盒模式的对抗样本生成问题，提出了一种基于遗传算法的恶意代码对抗样本生成方法，通过静态改写PE文件并保持其动态行为一致的方式实现了对基于机器学习的恶意代码检测模型的逃逸。

3 基于遗传算法的恶意代码对抗样本生成方法

目前在工业界得到广泛应用的基于机器学习的恶意代码检测系统，大多针对PE可执行文件实施静态检测，PE文件也是目前恶意代码采用最多的文件类型之一。因此，为了分析评估实际环境下基于机器学习的恶意代码检测系统的对抗能力，本文提出了一种PE文件的恶意代码对抗样本生成方法，在保留PE文件原始行为的基础上，利用遗传算法对原始PE文件实现静态改写，使生成的对抗样本能够对基于机器学习的恶意代码检测引擎形成干扰，甚至逃逸引擎的检测。

本节首先对保持行为一致的PE文件静态改写方法进行介绍，然后阐述基于遗传算法的对抗样本生成方法。

3.1 动态行为一致的PE文件静态改写

为实现保持行为一致的PE文件静态改写，需要PE文件格式(如图1[25])各字段、结构对其动态行为的影响进行分析评估。本文的思路是，通过PE文件格式预分析，评估各字段对动态执行的影响，形成静态改写素材库，在后续的样本变异生成后，通过动态沙箱校验其行为一致性，保证最终绕过检测的样本与原始样本动态行为一致。其中，PE文件格式的预分析主要是在PE格式提供的字段划分基础上，针对部分典型PE样本，利用模糊测试方法针对所有字段进行修改、已有字段的增加/删除、字段顺序重新编排等，并通过执行验证的方式对其合规性进行快速验证，筛选出不会影响样本正常加载启动的字段，继而针对上述字段更进一步进行模糊测试，并通过程序后段行为是否触发来判断字段对程序运行的影响，最终筛选出能够初步保持动态行为一致的潜在改写原子操作。

通过前述PE格式字段测试，本文筛选出PE头文件、节表信息两类PE文件改写原子操作，如表1所示，包括添加冗余模块、填充冗余信息、改变节表顺序等。上述原子操作不会改变原样本的执行逻辑，改写后的对抗样本的动态行为与原始样本保持一致。

3.2 基于遗传算法的对抗样本生成

本文在生成对抗样本的过程中，用遗传算法[26]作为样本改写的选择策略。遗传算法(Genetic Algorithm, GA)是一种启发式搜索算法，受达尔文进化论的启发，属于自然进化算法(Evolutionary Algorithm, EA)。遗传算法可以通过模拟自然选择过程中的变异、选择和交叉等操作生成用于优化和搜索问题的解决方案。该算法的主要特点是直接对目

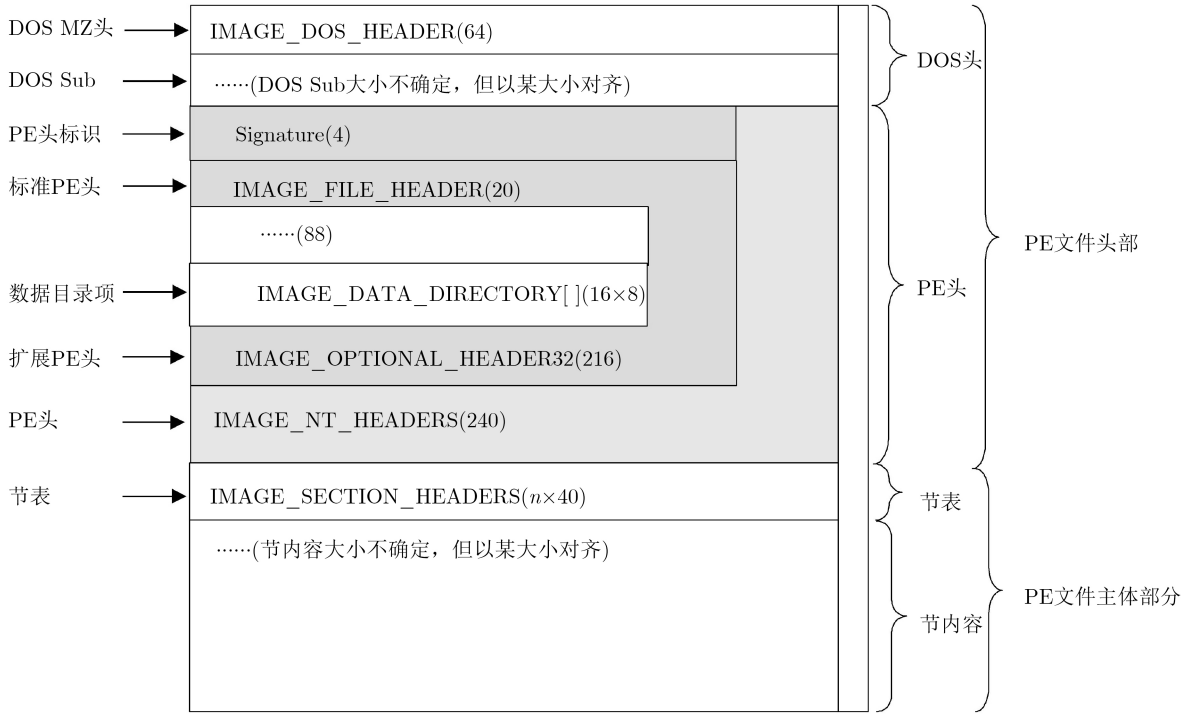

图 1 PE文件格式结构

表 1　PE文件改写原子操作

改写模块	改写内容
PE头文件	PE标志位修改
	PE文件校验和修改
节表	导入表添加冗余导入函数
	节表模块重命名
	节表冗余信息填充
	节表新模块添加
PE文件	加壳、脱壳操作

标对象进行操作，采用概率与统计的方法获得最优解，不需要特定的规则，能自适应调整、优化搜索方向。

NVIDIA公司研究团队提出的MalConv[13]是近期基于机器学习的静态检测方法的代表性工作，该方法可捕捉PE文件整体特征，检测效果明显优于业界基于n-gram[15]和基于PE头[2]的检测方案，因此本文评估实验将以MalConv恶意代码检测模型为目标，在本节中，先结合MalConv检测模型的相关知识，形式化地定义样本检测过程中的一些基本概念，然后介绍本文提出的基于遗传算法的对抗样本生成算法。

3.2.1　基于MalConv模型检测的数学定义

本文将恶意代码检测模型MalConv作为对抗目标，并做如下的数学定义：用x表示输入样本，函数$f(x)$表示MalConv模型对输入样本的恶意性评分，函数$F(x)$表示该模型对输入样本是否为恶意代码的预测，则有MalConv对输入样本的恶意性预测评分$f(x)\in[0,1.0]$，对样本是否具有恶意性的预测结果$F(x)\in\{'-','+'\}$。定义$'-'$为恶意样本，$'+'$为良性样本，预测分数区间以0.5作为临界点：当$f(x)\in[0,0.5)$时，$F(x)='-'$，样本为恶意样本；当$f(x)\in[0.5,1.0]$时，$F(x)='+'$，样本为良性样本。那么，基于机器学习的恶意代码检测模型对对抗样本实例进行预测时，会有如下情况出现：$x_{\mathrm{A}}^{-}=\left\{x_{\mathrm{A}}^{-}\in x^{-}|f(x_{\mathrm{A}}^{-})\in[0.5,1.0]\Rightarrow F(x_{\mathrm{A}}^{-})='+'\right\}$或$x_{\mathrm{A}}^{+}=\left\{x_{\mathrm{A}}^{+}\in x^{+}|f(x_{\mathrm{A}}^{+})\in[0,0.5)\Rightarrow F(x_{\mathrm{A}}^{+})='-'\right\}$。

3.2.2　基于自适应遗传算法的PE对抗样本生成方法

根据遗传算法的定义，本文将3.2.1小结的内容与遗传算法的概念相结合，得到如下映射：样本x作为遗传算法中的个体，改写样本的原子操作a作为带有遗传信息的基因，检测模型对样本的预测函数$f(x)$作为适应度，样本检测结果$F(x)\in\{'-','+'\}$表示种群——恶意文件种群、良性文件种群。

为了高效生成对抗样本，我们收集并整理了改写素材库。该素材库由2部分构成：(1)随机收集5000个PE文件，将这些PE文件按头文件、节表模块等内容进行拆分及归类；(2)随机生成部分可打印字符串作为改写素材。在生成对抗样本时，利用随机种子生成样本改写的随机操作序列，这个操作序列即“基因组”，按照“基因组”的操作改写样本，从素材库中随机选取需要的具体内容。

基于遗传算法的对抗样本生成流程图如图2所示。

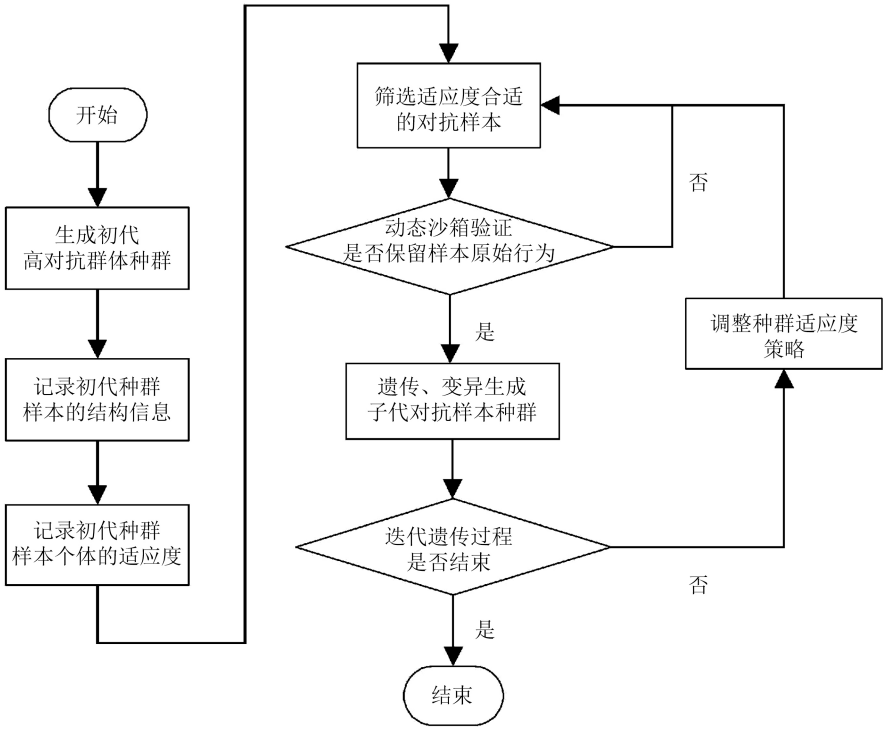

图 2 基于遗传算法的对抗样本生成算法流程图

具体步骤如下：

(1) 生成初始的对抗样本种群，首先随机生成K段“基因组”，即改写操作的随机操作序列$[a_1, a_2, \cdots, a_n]$，然后对目标样本集逐个进行改写，得到一个初代的对抗样本种群；

(2) 解析步骤(1)中生成的初代对抗样本种群，记录每个对抗样本的结构信息、与原始样本的差异信息；

(3) 通过MalConv恶意代码检测模型检测生成的对抗样本种群，将MalConv模型的预测分数作为个体的适应度，记录每个对抗样本的适应度，适应度的取值区间为$[0, 1.0]$；

(4) 根据适应度的大小选择参与进化的父体与母体，在分别设定良性、恶意样本的适应度阈值区间的前提下，选择适应度达到给定阈值的对抗样本；

(5) 在选择父体、母体的过程中，利用开源沙箱系统Cuckoo Sandbox[27]验证父体、母体是否保留原始的行为，将保留原始行为的对抗样本作为下一代进化的父体、母体；

(6) 对被选出的父体与母体执行遗传操作，即复制父体与母体的基因，采用交叉、变异等方式生成新的K段“基因组”，生成下一代对抗样本种群；

(7) 判断遗传变异是否结束：如是，结束对抗样本的生成过程，返回最终的对抗样本集；如否，调整种群适应度，再次执行步骤(4)。

进化过程的自适应选择策略体现在步骤(4)和步骤(5)，不单依靠个体适应度的大小排序进行选择，还要根据进化特征(是否保留样本原始行为)作为选择父体、母体的依据。

本文方法中的交叉是指互换父体、母体基因组中的部分基因，按照一定的交叉概率交换父体、母体样本改写操作序列中的部分操作；变异是在变异概率指导下，改变父体、母体基因组中的部分基因，替换原样本改写操作序列中的部分操作。在保留优秀基因的基础上，变异增加了基因的多样性，能够提高找到最优解的概率。

本文方法的自适应过程体现在进化过程中，父体、母体的交叉概率和变异概率是根据进化过程中种群的适应度分布进行自适应调整的。当遗传进化过程处于种群适应度较为分散的初期，自适应交叉概率和自适应变异概率选择较大的概率值，通过广泛搜索来增加种群的多样性；在种群适应度较为收敛的后期，自适应地调整为较小的概率值，通过细致搜索保证不破坏最优解。本文方法与基本遗传算法相比，自适应调整策略有利于提高收敛速度和收敛精度，能够更快获取最优解。

针对基于时序特征的恶意代码检测模型，基于遗传变异的PE文件对抗样本生成方法的逃逸检测效果明显，我们将在第4节中对实验及结果进一步阐述。

4 实验评估

为验证所提方案的有效性，评估其生成的对抗

样本是否能够绕过现有机器学习检测模型，本文针对美国NVIDIA研究团队近期提出的MalConv[13]恶意代码检测模型进行对抗样本逃逸测试，同时为进一步评估本方案在现实场景下的有效性，本文还针对VirusTotal(谷歌公司旗下业界知名的恶意软件检测平台)商用机器学习检测引擎进行了样本逃逸测试。

NVIDIA研究团队尚未公布可用的MalConv检测模型，因此本文拟依据其论文方案进行复现，针对复现的模型进行逃逸测试。为训练该模型，本实验收集了15168个样本。该样本集由奇安信技术研究院提供，通过在其用户终端进行为期7天的实时收集，并2次筛选获得的真实样本。本实验的训练集和测试集按照9:1划分，详细信息如表2所示。模型输出评分小于0.5的样本判定为良性样本，评分在0.5～1区间的样本判定为恶意样本。通过100轮的迭代训练，模型检测准确率达到98.88%。

表 2　实验数据统计信息

样本	训练集	测试集
良性样本	7059	784
恶意样本	6593	732
总数	13652	1516

针对前述自行训练的MalConv模型，本文利用遗传算法生成的初代对抗样本测试集进行测试，模型检测准确率为96.97%，体现了一定的抗干扰能力；但是针对经过遗传算法多代演化变异生成的样本，MalConv模型检测准确率为84.23%，相比于原始样本集的检测结果，准确率下降了14.65%。详细的测试数据如表3所示。

表 3　恶意代码检测引擎检测结果

评测样本集	良性样本误报	恶意样本误报	误报样本综述	模型检测准确率(%)
原始样本集	7	10	17	98.88
初代对抗样本集	37	9	46	96.97
优化后的对抗样本集	228	11	239	84.23

为进一步验证现实场景下本文所提方案的有效性，本文利用VirusTotal上4个具有代表性的基于机器学习的恶意代码检测引擎(Cylance[6], Sophos ML[7], Endgame[28], Trapmine[29])对本方法生成的能够逃逸MalConv模型检测的239个对抗样本进行2次验证。4家检测引擎对本文方法生成的239个对抗样本的测试结果如表4所示。根据表4的内容可以看出，本文方法能够有效地干扰基于机器学习的恶意代码检测引擎工作。在239个对抗样本中，有235个样本使至少1家检测引擎产生误报，即样本逃逸率为98.23%。其中，有111个样本可以使Cylance检测引擎形成误报，逃逸率高达46.45%。

本文对上述检测引擎误报的对抗样本进行手工分析，将其与原始样本进行对比，发现二者的差异主要在于，对抗样本去掉了一些PE文件的结构特征(如PE文件头部DOS Stub)，并对原文件的节表进行多次的重命名、改变顺序、增添冗余信息等操作。这些差异内容并不会改变样本的原始行为，却在一定程度上改变了原始样本作为输入数据的顺序特征。正是这些序列化特征的改变，使本文生成的对抗样本能够实现对检测引擎的干扰，甚至逃逸。需要指出的是，虽然人工修改PE文件的方法同样可以生成与本文方法类似的对抗样本，但这会增加大量的人工成本，并且人工修改的主观性会使生成的对抗样本附加过多的冗余信息，从而影响对抗样本的质量。本文方法在无需人工干预的情况下，可以高效生成对抗样本，并基于遗传算法的自适应性对其进行简化，使得对抗样本可以更具针对性。

表 4　厂商产品的检测成功率

恶意代码检测引擎	误报样本数	检测逃逸率(%)
Cylance	111	46.45
Endgame	43	17.99
Sophos ML	50	20.92
Trapmine	35	14.64

本文的实验结果表明，针对基于机器学习的恶意代码检测模型，变异策略具备干扰检测模型的能力，使良性样本被判定为恶意样本、导致检测引擎产生误报；使恶意样本被判定为良性样本、实现检测逃逸。

5　结束语

恶意代码检测机器学习模型的对抗样本生成，对于评估和完善恶意代码检测系统具有重要意义。本文通过实际对抗样本的分析和恶意代码机器学习检测模型的健壮性评估，提出了一种基于遗传算法的恶意代码对抗样本生成方法，在不改变样本原始动态行为的条件下，通过PE文件改写序列的编组变异，对样本进行静态修改，可以使基于机器学习

的恶意代码检测模型对变异后样本作出错误判定。实验表明，本文所提方案生成的样本可有效绕过MalConv恶意代码检测模型，以及VirusTotal中4款基于机器学习的商用恶意代码检测引擎，有效提升了生成对抗样本的真实性和模型对抗评估的准确性。

参 考 文 献

[1] LANDAGE J and WANKHADE M P. Malware and malware detection techniques: A survey[J]. *International Journal of Engineering Research & Technology*, 2013, 2(12): 61–68.

[2] SAXE J and BERLIN K. Deep neural network based malware detection using two dimensional binary program features[C]. The 10th International Conference on Malicious and Unwanted Software (MALWARE), Fajardo, USA, 2015: 11–20. doi: 10.1109/MALWARE.2015.7413680.

[3] ARP D, SPREITZENBARTH M, HUBNER M, *et al*. Drebin: Effective and explainable detection of android malware in your pocket[C]. Network and Distributed System Security Symposium, San Diego, USA, 2014: 23–26. doi: 10.14722/ndss.2014.23247.

[4] RAFF E, SYLVESTER J, and NICHOLAS C. Learning the PE header, malware detection with minimal domain knowledge[C]. The 10th ACM Workshop on Artificial Intelligence and Security, Dallas, USA, 2017: 121–132. doi: 10.1145/3128572.3140442.

[5] RAFF E, ZAK R, COX R, *et al*. An investigation of byte n-gram features for malware classification[J]. *Journal of Computer Virology and Hacking Techniques*, 2018, 14(1): 1–20. doi: 10.1007/s11416-016-0283-1.

[6] Cylance Inc. What's new in CylancePROTECT and CylanceOPTICS[EB/OL]. https://s7d2.scene7.com/is/content/cylance/prod/cylance-web/en-us/resources/knowledge-center/resource-library/briefs/Whats-New-CylancePROTECT-and-CylanceOPTICS.pdf, 2020.

[7] Sophos Inc. Sophos central migration tool articles, documentation and resources[EB/OL]. https://community.sophos.com/kb/en-us/122264#Product%20Information, 2020.

[8] 梁光辉, 庞建民, 单征. 基于代码进化的恶意代码沙箱规避检测技术研究[J]. 电子与信息学报, 2019, 41(2): 341–347. doi: 10.11999/JEIT180257.

LIANG Guanghui, PANG Jianmin, and SHAN Zheng. Malware sandbox evasion detection based on code evolution[J]. *Journal of Electronics & Information Technology*, 2019, 41(2): 341–347. doi: 10.11999/JEIT180257.

[9] GROSSE K, PAPERNOT N, MANOHARAN P, *et al*. Adversarial perturbations against deep neural networks for malware classification[J]. arXiv, 2016, 1606.04435.

[10] XU Weilin, QI Yanjun, and EVANS D. Automatically evading classifiers[C]. The 23rd Annual Network and Distributed System Security Symposium, San Diego, USA, 2016: 21–24. doi: 10.14722/ndss.2016.23115.

[11] HU Weiwei and TAN Ying. Generating adversarial malware examples for black-box attacks based on GAN[J]. arXiv, 2017, 1702.05983.

[12] HU Weiwei and TAN Ying. Black-box attacks against RNN based malware detection algorithms[C]. The Workshops of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, USA, 2018.

[13] RAFF E, BARKER J, SYLVESTER J, *et al*. Malware detection by eating a whole exe[C]. The Workshops of the 32nd AAAI Conference on Artificial Intelligence, New Orleans, USA, 2018: 268–276.

[14] TOTAL V. VirusTotal-free online virus, malware and url scanner[EB/OL]. https//www.virustotal.com/en, 2012.

[15] PASCANU R, STOKES J W, SANOSSIAN H, *et al*. Malware classification with recurrent networks[C]. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 2015: 1916–1920. doi: 10.1109/ICASSP.2015.7178304.

[16] KOLOSNJAJI B, ZARRAS A, WEBSTER G, *et al*. Deep learning for classification of malware system call sequences[C]. The 29th Australasian Joint Conference on Artificial Intelligence, Hobart, Australia, 2016: 137–149. doi: 10.1007/978-3-319-50127-7_11.

[17] HUANG Wenyi and STOKES J W. MtNet: A multi-task neural network for dynamic malware classification[C]. The 13th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, San Sebastián, Spain, 2016: 399–418. doi: 10.1007/978-3-319-40667-1_20.

[18] MANNING C D, RAGHAVAN P, and SCHÜTZE H. Introduction to Information Retrieval[M]. Cambridge: Cambridge University Press, 2008.

[19] HAN K S, LIM J H, KANG B, *et al*. Malware analysis using visualized images and entropy graphs[J]. *International Journal of Information Security*, 2015, 14(1): 1–14. doi: 10.1007/s10207-014-0242-0.

[20] KANCHERLA K and MUKKAMALA S. Image visualization based malware detection[C]. 2013 IEEE Symposium on Computational Intelligence in Cyber Security (CICS), Singapore, 2013: 40–44. doi: 10.1109/CICYBS.2013.6597204.

[21] LIU Xinbo, LIN Yaping, LI He, *et al*. A novel method for malware detection on ML-based visualization technique[J]. *Computers & Security*, 2020, 89: 101682. doi: 10.1016/j.cose.2019.101682.

[22] Skylight. Cylance, I kill you![EB/OL]. https://skylightcyber.com/2019/07/18/cylance-i-kill-you/, 2019.

[23] MOHURLE S and PATIL M. A brief study of wannacry threat: Ransomware attack 2017[J]. *International Journal of Advanced Research in Computer Science*, 2017, 8(5): 1938–1940. doi: 10.26483/ijarcs.v8i5.4021.

[24] DANG Hung, HUANG Yue, and CHANG E C. Evading classifiers by morphing in the dark[C]. 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, USA, 2017: 119–133. doi: 10.1145/3133956.3133978.

[25] 戚利. Windows PE权威指南[M]. 北京: 机械工业出版社, 2011: 67–68.
QI Li. Windows PE: The Definitive Guide[M]. Beijing: Machinery Industry Press, 2011: 67–68.

[26] KOZA J R. Genetic Programming II: Automatic Discovery of Reusable Subprograms[M]. Cambridge, MA, USA: MIT Press, 1994: 32.

[27] Cuckoo Sandbox. Cuckoo Sandbox–Automated malware analysis[EB/OL]. http://www.cuckoosandbox.org, 2017.

[28] BANON S. Elastic endpoint security[EB/OL]. https://www.elastic.co/cn/blog/introducing-elastic-endpoint-security, 2019.

[29] Trapmine Inc. TRAPMINE integrates machine learning engine into VirusTotal[EB/OL]. https://trapmine.com/blog/trapmine-machine-learning-virustotal/, 2018.

闫　佳：男，1991年生，博士生，研究方向为网络与系统安全.
闫　佳：男，1986年生，副研究员，研究方向为网络与系统安全.
聂楚江：男，1983年生，副研究员，研究方向为网络与系统安全.
苏璞睿：男，1976年生，研究员，研究方向为网络与系统安全.

责任编辑：陈　倩