基于机器学习算法的主机恶意代码检测技术研究

张东，张尧，刘刚，宋桂香
（浪潮电子信息产业股份有限公司，北京 100085）

摘　要：对机器学习算法下主机恶意代码检测的主流技术途径进行了研究，分别针对静态、动态这 2 种分析模式下的检测方案进行了讨论，涵盖了恶意代码样本采集、特征提取与选择、机器学习算法分类模型的建立等要点。对机器学习算法下恶意代码检测的未来工作与挑战进行了梳理。为下一代恶意代码检测技术的设计和优化提供了重要的参考。

关键词：恶意代码检测；机器学习；静态分析；动态分析；分类模型
中图分类号：TP309
文献标识码：A
doi: 10.11959/j.issn.2096-109x.2017.00179

Research on host malcode detection using machine learning

ZHANG Dong, ZHANG Yao, LIU Gang, SONG Gui-xiang
(Inspur Electronic Information Industry Co.,Ltd, Beijing 100085, China)

Abstract: Main trends of host malcode detection using machine learning were focused on, and two categories of detection models(namely static analysis and dynamic analysis) were well discussed. Moreover, the critical stages such as malcode samples collection, feature extraction and selection, the construction of machine learning classifiers were considered fully. At last, some future work and challenges in this field were listed. The work can serve as a practical reference for establishing next-generation malcode detection techniques.

Key words: malcode detection, machine learning, static analysis, dynamic analysis, classification model

1　引言

随着互联网的繁荣和发展，包括虚拟化、软件定义网络、物联网在内的新兴技术逐步得到应用，进一步加速了万物互联的步伐。同时，海量的核心数据和网络应用也不断向云端、数据中心等关键信息基础设施整合和迁移，主机安全也因此成为网络攻防战的焦点。恶意代码是指运行在目标主机中，按照攻击者所规定逻辑执行的指令，其类别包括计算机病毒、蠕虫、木马、僵尸网络、勒索软件等。恶意代码攻击可以窃取核心数据和敏感信息，甚至对计算机系统和网络造成破坏，是当今网络安全的最大威胁之一[1,2]。

现阶段，恶意代码呈现变种数量多、传播速度快、影响范围广的特点。尤其需要注意的是，恶意代码常针对新型漏洞（如零日漏洞）进行设计，是敌手发动高级持续性威胁（APT，advanced persistent threat）[3]的主要技术手段。而传统的恶意代码检测方法，如基于签名特征码（signature）的检测和基于启发式规则（heuristic）的检测，在应对数量繁多的未知恶意代码时，正面临越来越大的挑战：签名特征码检测方法通过维护一个已知的恶意代码库，将待检测代码样本的特征码与恶意代码库中的特征码进行比对，如果特征码出现匹配，则样本为恶意代码。该方法需要耗费大量的人力、

收稿日期：2017-06-12；**修回日期：**2017-07-02。**通信作者：**张尧，zhangyaobj@inspur.com

物力对恶意代码进行研究并要求用户及时更新恶意代码库，检测效率和效果越来越力不从心，并且很难有效抵御未知恶意代码；另外，启发式规则检测方法通过专业的分析人员对现有的恶意代码进行规则提取，并依照提取出的规则对代码样本进行检测。但面对现阶段恶意代码爆炸式的增长趋势，仅依赖人工进行恶意代码分析，在实施上变得愈发困难。

为了应对上述挑战，机器学习算法下的恶意代码检测思想被提出。基于机器学习算法的防护技术为实现高准确率、自动化的未知恶意代码检测提供了行之有效的技术途径，已逐渐成为业内研究的热点。本文围绕机器学习算法下的主机恶意代码检测，系统地梳理和探讨该方向的主流技术途径与解决方案。

根据 Cohen[4]对恶意代码的研究结果，可知恶意代码检测的本质是一个分类问题，即把待检测样本区分成恶意或合法的程序。因此机器学习算法驱动下的主机恶意代码检测技术，其核心步骤为：1) 采集数量充分的恶意代码样本；2) 对样本进行有效的数据处理，提取特征；3) 进一步选取用于分类的主要数据特征；4) 结合机器学习算法的训练，建立分类模型；5) 通过训练后的分类模型对未知样本进行检测。如图 1 所示，根据检测过程中样本数据采集角度的不同，可以将检测分为静态分析与动态分析：静态分析不运行待检测程序，而是通过程序（如反汇编后的代码）进行分析得到数据特征，而动态分析在虚拟机或仿真器中执行程序，并获取程序执行过程中所产生的数据（如行为特征），进行检测和判断。

2 恶意代码的样本采集

恶意代码样本的有效采集是进行代码分析工作的基础。当结合机器学习算法进行检测时，只有通过充分的样本数据训练，分类模型才能更准确地实现检测功能。一般来讲，恶意代码样本的获取途径有如下几种。

1) 用户端采样：这是大多数杀毒软件厂商的主要获取方法，使用杀毒软件的终端用户将恶意代码样本上传至厂商，该方法具有较好的实时性，但安全厂商的样本数据往往选择不对外开放，很难直接获取。

2) 公开的网络数据库：如 VirusBulletin[5]、Open Malware[6]、VX Heavens[7]等，但相比恶意代码的更新速度，现阶段公开在线样本系统较有限，且站点存在隐蔽性不足、易遭到攻击的问题。因此，建立威胁情报的共享机制，日益突显出其重要性。

3) 其他技术途径：通过蜜罐（如 Nepenthes 蜜罐[8]）等捕获工具进行搜集，即设计一个专门的具有脆弱性的系统，诱导攻击者进行攻击进而得到恶意代码样本。一些木马和网络后门等也可以通过垃圾邮件陷阱或安全论坛（如卡饭论坛[9]）的方式得到。不过，上述技术途径的捕获样本规模较有限。

3 基于机器学习的静态分析方法

为了提取恶意代码的静态特征，往往需要先对程序代码进行逆向分析。常用的工具包括 IDA Pro[10]、Hopper、OllyDbg 等，其中，IDA Pro 是交互式的反汇编器，它不仅可以产生恶意程序的

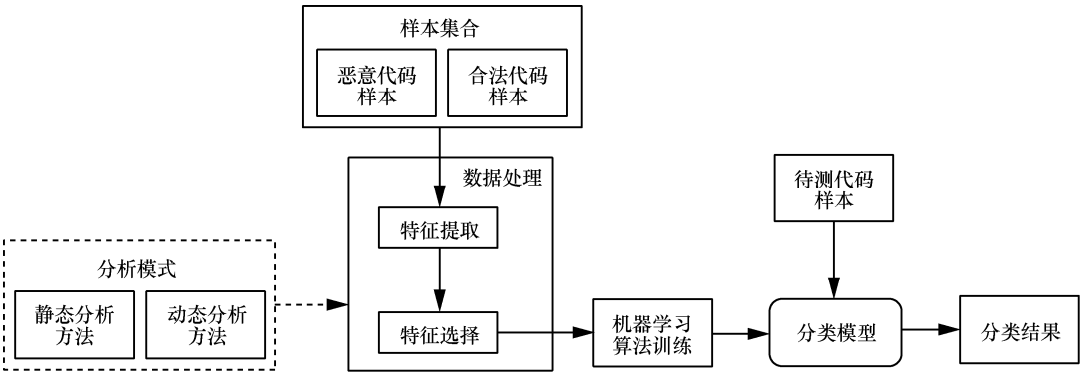

图 1 基于机器学习算法的恶意代码检测步骤

汇编代码，也具有识别函数块、获取输入函数、描述函数流程图等功能。

3.1　样本特征的提取

1) 基于序列的特征类型

基于序列的特征类型在样本特征的提取上应用最为广泛，其代表技术为 N 元语法模型（N-gram）。N-gram 假定 N 个出现的词只与之前出现的 N−1 个词相关，其中，N 代表一个特征序列的长度。如果考虑一个长为 L 的词组集合，则 N 元语法模型会通过滑动窗口的形式，将词组划分为 L−N+1 个特征序列。例如，当 3-gram 被应用在词集{PUSH, SUB, SAL, AND, DIV, LDS, POP}上（此时 L=7）时，如图 2 所示，会得到 5 个特征序列，每个序列包含 3 个词元。

在词元的选择上，Abou-Assaleh 等[11]首先提出了基于字节（Byte）序列的特征提取框架，并使用 K 近邻分类方法实现了恶意代码的有效检测。此外，Kolter[12]和 Moskovitch[13]等也分别对基于字节的 N-gram 方法进行了测试和评估。另一类词元选择方式是基于操作码（Opcode）的，Opcode 是描述程序执行操作的机器语言指令，相对于字节序列来讲，具有更强的实际意义和可靠性，结合 Opcode 的特征提取可以更好地表征恶意代码。文献[14]基于 Opcode 序列完成了对恶意代码进化的追踪。Siddiqui 等[15]结合操作码序列的方式，通过神经网络、决策树等分类算法，实现了 98.4%的检测准确率。Moskovitch 等[16]进一步使用超过 3×10^4 个文件的测试集，对基于 Opcode 序列的 5 种不同分类器进行评估，对恶意代码的检测准确率高达 99%。

2) 基于字符串的特征类型

另一种特征类型的提取方式是基于程序代码中的可输出字符串[17,18]，因为可输出字符串在某种程度上反映了待测程序的意图。例如，从代码中的“https://..”字符串可以推测程序的网络连接意图；而包含目录路径的字符串则说明程序可能尝试读取用户文档或注册表信息等。文献[18]选取了可执行文件中 100 个可输出字符串，以此为特征训练基于支持向量机的分类器，实现了 99.38%的准确率。与基于序列的特征类型相比，代码中的字符串数量有限，因此提取的特征集具有较少的维度，在计算成本上可以实现有效的控制。

3) 基于 API 调用的特征类型

程序对应用程序编程接口（API，application programming interface）调用也可以作为特征类型。文献[19]对 API 调用进行了讨论，指出程序 PE（portable executable）文件头中的 API 信息不具有准确性，因为恶意代码会在 PE 文件头中夹杂错误的 API 信息。Ding 等[19]对反汇编后的代码进行 API 调用分析，利用恶意代码和合法代码应用程序编程接口分布的差异性提取了基于 API 调用的程序特征。文献[20]将代码中的 API 调用序列转化为对应的马尔可夫（Markov）链，有向图中边的权重表示调用 API 的状态转移概率。通过基于 Markov 链的特征提取，实现了对未知恶意软件的有效分类。

3.2　样本特征的选取

由于提取的数据特征常包含冗余信息，容易引起过度拟合问题，本节对数据特征选取的主要方法进行介绍，其种类主要包括信息增益（IG，information gain）、增益率（GR，gainratio）、文档频率（DF，document frequency）、主成分分析（PCA，principal component analysis）等。

信息增益使用信息熵度量特征使用与否而导致的信息量差异。式(1)给出了样本集合 S 信息熵

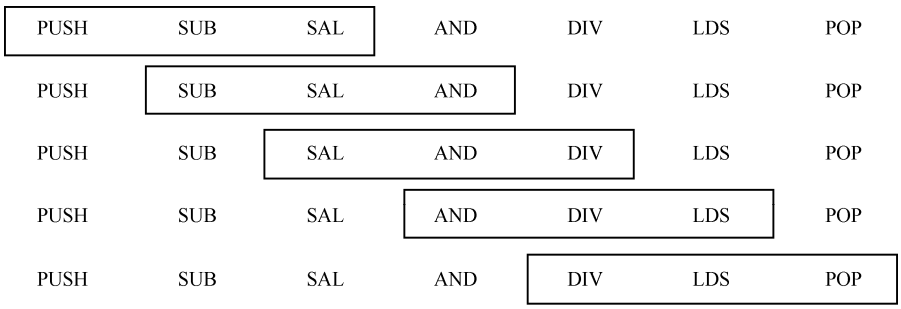

图 2　基于操作码的 3 元语法模型应用示例

（基于子集合，如对集合 S 的一个分类 C）的计算方式。

$$E(S)=\sum_{c\in C}-\frac{|S_c|}{|S|}\mathrm{lb}\frac{|S_c|}{|S|} \tag{1}$$

式(2)为信息增益的计算方法。

$$IG(S,A)=E(S)-\sum_{v\in V(A)}\frac{|S_v|}{|S|}E(S_v) \tag{2}$$

$IG(S,A)$度量了在特征 A 的基础上对样本进行划分后，样本信息熵的减少量，其中，$V(A)$是特征 A 的值域，S_v 是 S 中特征 A 上值等于 v 的样本集合。对于计算结果，信息增益越显著的特征属性越重要，在选择时具有更强的倾向性。

然而，信息增益方法会让取值过多的特征拥有过大优势。增益率方法结合了特征在样本数据分类中的影响，起到了降低上述偏差的作用。如式(3)和式(4)所示，S_i 是包含 d 个值的特征 A 分割样本而形成的子集，$SI(S,A)$是 S 关于特征 A 各值的熵。因此，对于信息增益相近的特征 A_1 和 A_2，分组能力显著（即 $SI(S,A)$较小）的特征会被优先选择。

$$GR(S,A)=\frac{IG(S,A)}{SI(S,A)} \tag{3}$$

$$SI(S,A)=-\sum_{i=1}^{d}\frac{|S_i|}{|S|}\mathrm{lb}\frac{|S_i|}{|S|} \tag{4}$$

文档频率方法统计某特征项在样本集中的出现频率，并设定一个门限值，选取在文本中出现频率超过门限值的特征项。且与 IG 和 GR 不同，DF 不需要大量的先验信息，且原理简单，计算效率高，因此在实现上常与其他方法（如 IG 方法）一起使用，以提升特征选取的准确度[19]。

主成分分析也是一类常见的特征选取方法，在静态、动态分析中常被用于实现对样本数据的降维[20]。PCA 通过线性变换，将原始数据投射到新的坐标系下，并通过新空间中最大线性无关组对数据样本进行表达，该线性无关组特征值的空间坐标即 PCA 方法所选取的特征。与 IG、DF 等方法不同，PCA 使用变换后的特征，而非原始特征的子集。

4 基于机器学习算法的动态分析技术

恶意代码的静态分析技术，在应对代码混淆或加壳等情形时，具有一定的局限性。为了保证代码评估的准确性，动态分析技术利用虚拟机或仿真器执行待测程序，监控并收集程序运行时显现的行为特征，并根据特征数据实现恶意代码的分类。

4.1 行为特征的提取

沙箱技术是收集行为特征的重要技术途径，许多安全公司提供了 Web 版的沙箱接口，用以对上传的程序样本进行动态分析，生成行为分析报告。目前常见的沙箱工具有 Anubis、Joe Sandbox、Cuckoo Sandbox、CWSandbox 等。

动态分析的重点是对监控行为的类型进行合理选择。一般来讲，基于行为分析的方案主要考察程序运行过程中所涉及的以下方面：1) 系统文件的改变，如创建或修改文件；2) 注册表键值的操作行为；3) 动态链接库（DLL，dynamic link library）的加载情况；4) 进程访问的情况；5) 系统服务行为，如开启、创建或删除服务；6) 网络访问情况；7) 应用程序编程接口（API）的调用。此外，一些解决方案[21]还对程序调用函数的数据信息进行分析，这时污点标签设置方法常被结合使用。

文献[22,23]结合行为报告的分析结果，对恶意代码的行为特征进行识别，借助机器学习算法对可执行文件进行分类。杨轶等[24]通过分析污点传播的过程，识别不同的恶意代码行为间控制指令和数据的依赖关系，从而比较恶意代码的相似性。Imran 等[25]通过隐马尔可夫模型对待测样本的动态行为特征进行描述，并借助机器学习算法实现分类。Anderson 等[26]则通过动态方式搜集程序指令序列，进而生成基于马尔可夫链的有向图。根据组合图核方法，得到指令序列图的相似性矩阵，最终使用支持向量机对恶意代码进行判定，检测准确率达到 96.41%。

4.2 行为特征的选取

许多沙箱工具，如 Anubis 和 CWSandbox 的输出格式为文本或可扩展标记语言（XML，extensible markup language），这两类格式更适用于小规模样本的人工分析。具体来说，文本格式报告对行为特征的刻画过于简单，粒度较粗，一些重要的行为不再可见；而 XML 格式下的分析报告表述又过于繁冗，不便于开展自动化分析。

为了高效处理行为分析数据，Trinius 等[27]提出基于恶意软件指令集（MIST，malware instruction set）的行为数据描述方法，常被用来对其他格式（如 XML 格式）的行为报告进行转换，从而达到在行为数据中选取主要特征的目的。

MIST 将程序行为的监控结果描述为一系列指令，其中每个线程和进程的执行流被分组在一个连续的报告中。每条指令都对应监控到的一个系统调用（system call）及其调用到的参数（argument），指令以短数值的方式予以标识。此外，系统调用的具体参数被分隔在不同等级的块中，反映不同程度的行为粒度。

如图 3 所示，CATEGORY 表示系统调用的类别，而 OPERATION 对应某个特定的系统调用。如‘12 0a’代表类别 Winsock（12）和对应的系统调用 connect_socket（0a）。ARGBLOCK1~ARGBLOCK*N* 代表各个参数，在参数块的编排上，低级别参数块编码相对稳定的、具备高区分度的属性（如产生新文件的目录），而高级别参数块包含更易变化的“噪声”属性（如生成的文件名称）。MIST 指令序列结构确保在少数指令不同的情况下，不同的恶意代码变种可以被迅速识别。

MIST 报告可以进一步通过向量空间模型（VSM，vector space model）进行向量化，生成可用于机器学习算法分类的数据。在特征项和特征项权重的计算上，可运用词袋模型（BOW，bag of words）。词袋模型的示例如下，假设有下述 2 个文件。

1) Samuel detected a malware. I detected the malware too.

2) The malware was detected by us.

基于上述 2 个文件，可以构建一个词汇表。

词汇表={1.“Samuel”，2.“detected”，3.“a”，4.“malware”，5.“I”，6.“the”，7.“too”，8.“was”，9.“by”，10.“us”}。这个词汇表一共包含 10 个不同的单词，利用索引号，上面 2 个文件可分别用 10 维向量表示（向量中元素为词表单词在文件中出现的频率）。

1) [1，2，1，2，1，1，1，0，0，0]

2) [0，1，0，1，0，1，0，1，1，1]

利用词袋模型，经过 MIST 处理后的指令语句将作为 VSM 模型中的特征项，指令的出现频率即为特征项的权重，以建立恶意代码的向量空间数据，这样就可以利用机器学习算法（如支持向量机）进行恶意代码的分类。

5　恶意代码分类算法

恶意代码进行静态、动态分析后得到的特征数据，可以作为机器学习算法训练的输入，产生相应的恶意代码分类器。本节对常见的算法，如 *K* 近邻（KNN，*k* nearest neighbor）[11]、支持向量机（SVM，support vector machine）[26]、朴素贝叶斯（Naïve Bayes）[16]、决策树（DT，decision tree）[15]、随机森林（RF，randomforest）[20]，以及深度学习算法（如卷积神经网络（CNN，convolutional neural network））[28,29]进行介绍。

KNN 算法是最直观的机器学习算法之一，样本的分类由距样本点最近的 *K* 个邻居决定，*K* 个邻居中大多数节点所在的类别即为分类结果。恶意代码检测常用到二分分类，这时一个有效方式是将 *K* 设置为奇数，有利于避免出现距离相同的两类节点。距离计算上，习惯使用的方式包括 Euclidean 距离和 Manhattan 距离。KNN 的一大优势是支持“增进式学习”，即训练集的新增样本可以作为增量进行训练，不需要重新对模型进行重训练。

SVM 算法尝试找寻一个线性超平面进行二分分类，如图 4 所示，该平面距离两类样本（两条虚线）的距离相同，通过超平面可以将样本进行划分。其中，虚线是通过支持向量的方式构建的。SVM 和 KNN 算法在处理规模较小的样本时

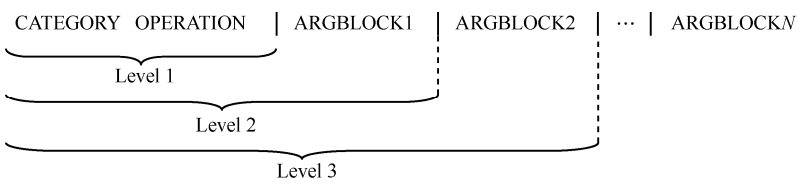

图 3　MIST 指令示意

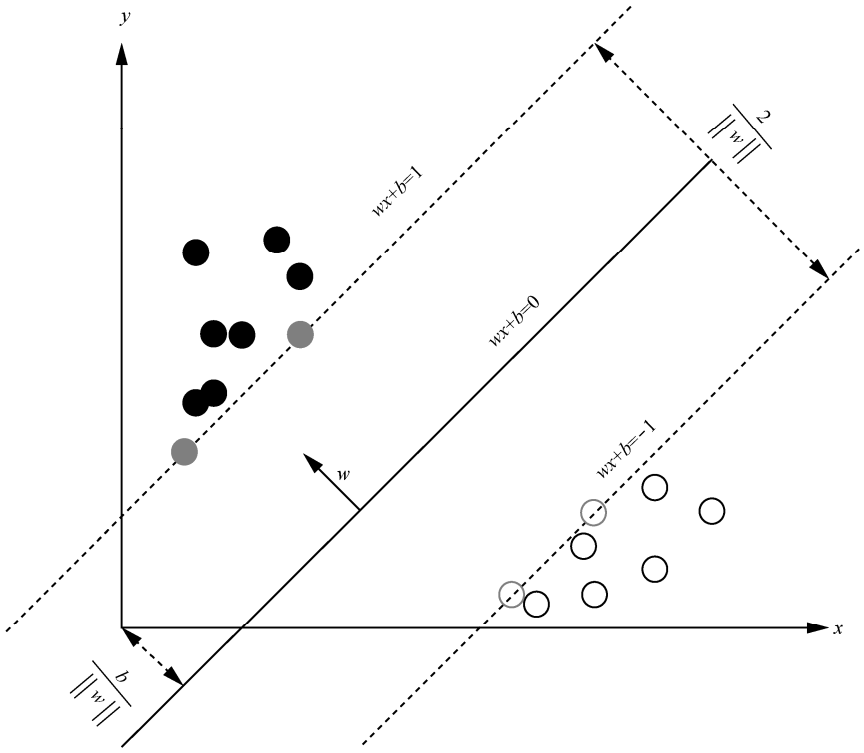

图 4　2 维空间下的 SVM 算法

较为有效，但随着数据集的增加，SVM 和 KNN 的计算成本均较大。

朴素贝叶斯是计算复杂度较低的分类方法，该方法基于贝叶斯定理，并默认特征之间相互独立。当某样本具有 t 个特征，总共类别数量为 n 时，朴素贝叶斯分类器分别计算每个分类的后验概率，如式(5)所示，算法根据计算结果选择概率最大的分类。

$$P(C_i \mid F_1F_2\cdots F_t)=\frac{P(F_1F_2\cdots F_t \mid C_i)P(C_i)}{P(F_1F_2\cdots F_t)} \quad (5)$$

由于 $P(F_1F_2\cdots F_t)$ 对所有类别来说都是一样的，根据假设，式(5)可进一步转化为求 $P(C_i)\prod_1^t P(F_t \mid C_i)$ 的最大值。朴素贝叶斯算法的一个优势是概率性的分类结果描述，避免了非是即否的定性判断。这样，分类结果可以进一步提交给人工进行决策，从而减少对合法代码的误报。

决策树算法是另一类重要的机器学习算法，分类器以树结构的形式表达，树的中间节点（包括根节点）是不同的特征，从根节点往下逐一匹配，判断得到的叶子节点即为分类结果。一般可以通过启发式贪心搜索获得最小的决策树，其实现方式之一是利用信息增益方法对数据集进行划分。

随机森林算法是由一系列决策树构成的分类器，森林中每棵树都由一部分独立取样的数据样本进行训练，待测样本在每棵树中进行判断，所有输出中频次最高的分类即为 RF 算法的判断结果。DT 和 RF 分类算法的优势在于特征的类型可以不一致，因此能够结合多类特征（字符串特征、序列特征等）进行判断。

深度学习框架下包含多类算法，其中卷积神经网络是一个多层的神经网络结构，由卷积层、次抽样层和全连接层组成，卷积层和次抽样层在中间层重复出现，全连接层在底层用于输出分类结果。CNN 的训练一般借助反向传播 BP 算法，BP 算法通过正向数据流和反向误差信号这 2 个传播过程，逐步调整各层的函数参数。虽然可以提供不错的分类效果，但 CNN 训练过程相对比较复杂[28]。此外，深度学习算法也可用于恶意代码的特征提取。在文献[29]中，Huang 等使用微软提供的目前为止最大的样本集（4.5×10^6 个文件），利用深度学习框架构造恶意软件的有效分类工具 MtNet。此外，由文献[29]可知，就恶意

代码检测的效果来说，训练层数的增加并不能显著提升检测率和误判率（1~2 层时，即可实现较优检测）。

6　结束语

在网络攻击日益复杂、恶意代码层出不穷的今天，机器学习算法在恶意代码检测中的应用逐渐受到学术界和众多安全厂商的重视。本文对基于机器学习算法恶意代码检测的技术方法和主流方案进行了梳理和讨论，这一工作将为新型主机恶意代码检测技术的设计和实现提供重要参考。但该领域仍属于发展阶段，还存在着许多未来工作和挑战，对其归纳如下。

1) 静态分析检测速度快、系统资源占用少，但随着代码混淆、加壳等反检测技术的发展，静态分析的准确性受到一定程度的限制。动态分析技术需要运行被测代码，在效率上存在局限性。一个主流的发展方向是将静态、动态分析技术进行有效结合，全方位地对待测代码进行评估。

2) 机器学习算法可以提供高准确率的恶意代码分类，但分类器一般作为黑盒机制被加以使用，安全人员缺乏对结果的理解。结果往往在不质疑分类器性质的情况下直接被使用，因此分类结果受经验阈值和数据特征的影响，出现一定倾向性。研究传统量化分析（如准确率、误报率）之外的统计学方法，如可信度（credibility），科学地评价和比较底层的机器学习算法，是未来一项重要的研究工作。

3) 在考虑敌手视角时，如果攻击者也通过机器学习技术优化恶意代码的设计，对攻击目标画像并实现精准攻击，该如何应对？同时，又该如何保证机器学习引擎不被攻击者“投毒”，防止出现干扰项致使训练出错产生误判，这些都是需要进一步研究和思考的问题。

参考文献：

[1] 国家互联网应急中心. 2015 年中国互联网网络安全报告[EB/OL]. http://www.cert.org.cn/publish/main/upload/File/2015annualreport.pdf.
CNCERT/CC. 2015 China cyber security report[EB/OL]. http://www.cert.org.cn/publish/main/upload/File/2015annualreport.pdf.

[2] ZHANG Y, WANG X, PERRIG A, et al. Tumbler: adaptable link access in the bots-infested Internet[J].Computer Networks, 2016, 105: 180-193.

[3] 360 威胁情报中心. 2016 中国高级持续性威胁（APT）研究报告[EB/OL]. http://zt.360.cn/1101061855.php?dtid=1101062514&did=490274251.
360 Threat Intelligence Center. 2016 China APT research report[EB/OL]. http://zt.360.cn/1101061855.php?dtid=1101062514&did=490274251.

[4] COHEN P. Models of practical defenses against computer viruses[J]. Computers & Security,1989, 8(2):149-160.

[5] VirusBulletin[EB/OL]. https://www.virusbulletin.com/resources/wildlists/.

[6] Open Malware[EB/OL]. http://www.offensivecomputing.net/.

[7] VX Heavens[EB/OL]. http://vxheaven.org/.

[8] BAECHER P, KOETTER M, HOLZ T, et al. The nepenthes platform: an efficient approach to collect malware[C]//The International Symposium on Recent Advances in Intrusion Detection (RAID). 2006:165-184.

[9] 卡饭论坛[EB/OL]. http://bbs.kafan.cn.
Kaspersky Forum [EB/OL]. http://bbs.kafan.cn.

[10] HEX-RAYS SA. IDA pro introduction[EB/OL]. http://www.hex-rays. com/products.shtml/.

[11] ABOU-ASSALEH T, CERCONE N, KESELJ V, et al. N-gram-based detection of new malicious code[C]//The 28th Annual International Computer Software and Applications Conference (COMPSAC). 2004:41-42.

[12] KOLTER J Z, MALOOF M A. Learning to detect and classify malicious executables in the wild[J]. The Journal of Machine Learning Research, 2006(7):2721-2744.

[13] MOSKOVITCH R, STOPEL D, FEHER C, et al. Unknown malcode detection via text categorization and the imbalance problem[C]//IEEE International Conference on Intelligence and Security Informatics (ISI). 2008:156-161.

[14] KARIM M E, WALENSTEIN A, LAKHOTIA A, et al. Malware phylogeny generation using permutations of code[J]. Journal in Computer Virology, 2005,1(1/2):13-23.

[15] SIDDIQUI M, WANG M C, LEE J. Data mining methods for malware detection using instruction sequences[C]//The Artificial Intelligence and Applications (AIA). 2008.

[16] MOSKOVITCH R, FEHER C, TZACHAR N, et al. Unknown malcode detection using opcode representation[C]//European Conference on Intelligence and Security Informatics(EuroISI). 2008: 204-215.

[17] SCHULTZ M G, ESKIN E, ZADOK F, et al. Data mining methods for detection of new malicious executables[C]//IEEE Symposium on Security and Privacy (S&P). 2001:38-49.

[18] LAI Y. A feature selection for malicious detection[C]//The 9th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. 2008: 365-370.

[19] DING Y, YUAN X, TANG K, et al. A fast malware detection algo-rithm based on objective-oriented association mining[J]. Computers &Security, 2013, 39:315-324.

[20] MARICONTI E, ONWUZURIKE L, ANDRIOTIS P, et al. MA-MADROID: detecting android malware by building Markov chains of behavioral models[C]//The Symposium on Network and

Distributed System Security (NDSS). 2017.

[21] SCHWARTZ E J, AVGERINOS T, BRUMLEY D. All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask)[C]//IEEE Symposium on Security and Privacy (S&P). 2010:317-331.

[22] CHRISTODORESCU M, JHA S, KRUEGEL C. Mining specifications of malicious behavior[C]//The 1st India Software Engineering Conference. 2008:5-14.

[23] RIECK K, HOLZ T, WILLEMS C, et al. Learning and classification of malware behavior[C]//The International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA). 2008:108-125.

[24] 杨轶, 苏璞睿, 应凌云, 等. 基于行为依赖特征的恶意代码相似性比较方法[J]. 软件学报, 2011, 22(10): 2438-2453.
YANG Y, SU P, YING L, et al. Dependency-based malware similarity comparison method[J]. Journal of Software, 2011, 22(10): 2438-2453.

[25] IMRAN M, AFZAL M T, QADIR M A. Malware classification using dynamic features and hidden markov model[J]. Journal of Intelligent & Fuzzy Systems, 2016, 31(2):837-847.

[26] ANDERSON B, QUIST D, NEIL J, et al. Graph-based malware detection using dynamic analysis[J]. Journal in Computer Virolo-gy, 2011, 7(4): 247-258.

[27] TRINIUS P, WILLEMS C, HOLZ T, et al. A malware instruction set for behavior-based analysis[C]//The 5th GI Conference on Sicherheit, Schutz und Zuverl assigkeit. 2010:205-216.

[28] 杨晔. 基于行为的恶意代码检测方法研究[D]. 西安: 西安电子科技大学, 2015.
YANG Y. Research on detection method of malware based on behavior[D]. Xi'an: Xidian University, 2015.

[29] HUANG W, STOKES J W. MtNet: a multi-task neural network for dynamic malware classification[C]//The International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment(DIMVA). 2016: 399-418.

作者简介：

张东（1974-），男，山东威海人，浪潮电子信息产业股份有限公司高级工程师，主要研究方向为系统软件安全。

张尧（1988-），男，湖北襄阳人，博士，浪潮电子信息产业股份有限公司研究员，主要研究方向为网络安全、主机系统安全与应用密码学。

刘刚（1979-），男，四川德阳人，硕士，浪潮电子信息产业股份有限公司工程师，主要研究方向为操作系统安全、可信计算与云安全。

宋桂香（1978-），女，山东郓城人，浪潮电子信息产业股份有限公司工程师，主要研究方向为安全测评。