基于数据挖掘的恶意代码检测综述

黄海新[1,2] 张 路[1,3] 邓 丽[1]
(沈阳理工大学信息科学与工程学院 沈阳 110159)[1] (中国科学院沈阳自动化研究所 沈阳 110016)[2]
(安天实验室 武汉 430074)[3]

摘 要 数据挖掘是一种基于统计学的自动发掘数据规律的方法,它能通过分析海量样本的统计规律来建立判别模型,从而让攻击者难以掌握免杀的规律,近年来得到了广泛关注和快速发展。综述了数据挖掘技术应用于恶意代码检测领域所取得的研究成果;对所涉及的特征提取、特征选择、分类模型及其性能评估方法等方面的研究成果进行了深入分析和比较;最后提出了基于数据挖掘的恶意代码检测所面临的挑战,并对研究方向进行了展望。

关键词 数据挖掘,机器学习,恶意代码检测,特征提取,特征选择
中图法分类号 TP393.08 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.7.002

Review of Malware Detection Based on Data Mining

HUANG Hai-xin[1,2] ZHANG Lu[1,3] DENG Li[1]
(College of Information Science and Engineering, Shenyang Ligong University, Shenyang 110159, China)[1]
(Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China)[2]
(Antiy Labs, Wuhan 430074, China)[3]

Abstract Data mining is a method for automatically discovering data rule based on statistics which can analyze huge amounts of sample statistics to establish discriminative model, so that an attacker can not master the law to avoid detection. It has attracted widespread interests and has developed rapidly in recent years. In this paper, the research on malware detection based on data mining was summarized. The research results on feature extraction, feature selection, classification model and its performance evaluation methods were analyzed and compared in detail. At last, the challenges and prospect were provided in the field.

Keywords Data mining, Machine learning, Malware detection, Feature extraction, Feature selection

1 引言

恶意代码(Malicious Code)被定义为运行在目标计算机上,使系统按照攻击者意愿执行任务的一组指令[1]。它们不仅影响个人计算机的正常使用,而且可能导致网络瘫痪,给网络用户和企业造成巨大的经济损失。常见的恶意代码类型有病毒(Virus)、蠕虫(Worm)、木马(Trojan) 3种。

计算机病毒是指编制或者在计算机程序中插入的破坏计算机功能或者毁坏数据,影响计算机使用,并且能够自我复制的一组计算机指令或者程序代码[37]。病毒通常可自行执行和自我复制,并具有很强的感染性和潜伏性,同时也必须由用户干预来触发执行。

蠕虫是一种通过网络传播的恶意代码,它具有传播性、隐蔽性、破坏性等特点,能够自我传播,无需用户干预而自行触发,也不需要利用文件寄生,通常利用漏洞进行传播。网络的高速发展使得蠕虫可在短时间内蔓延至整个网络,并具有很强的主动攻击性,因此它的危害远大于普通病毒。

木马是指附着在应用程序中或者单独存在的一种恶意程序,它可以利用网络远程响应网络另一端的控制命令,实现对被植入了木马程序的目标计算机的控制,或者窃取机密资料。木马通常具有欺骗性、隐蔽性和非授权性特点,与病毒的主要区别是,木马不具有传染性,不能实现自我复制[38]。

常用商业安全软件对恶意代码的检测主要基于两种方法:1)基于签名(Signature-based)的方法,通过识别程序所独有的二进制字符串来检测,该方法主要依赖于已知的签名数据库[1],因此无法查杀新的未知程序,而且需要持续更新签名数据库;2)基于启发式规则(Heuristic-based)的方法,该方法通过分析人员对已知的恶意代码提取具有启发式的规则,并通过该规则发现新的恶意代码。这两种方法的缺点是都只能在计算机被恶意代码感染后才能被检测到,而不能及时发现新的未知恶意代码[3]。另一方面,传统方法的维护成本较高,需要大量人工经验进行样本分析并提取规则,面对未来海量样本的趋势,这无疑是巨大的挑战。近年来,基于数据挖掘的恶意代码检测方法得到了广泛认可,这种方法能有效弥补传统方法的不足。

所谓恶意代码检测,其本质是基于先验信息对未知应用

到稿日期:2015-06-19 返修日期:2015-11-11 本文受国家自然科学基金(61233007)资助。
黄海新(1973-),女,副教授,主要研究方向为智能决策、智能电网、模糊控制,E-mail:shepherd7@163.com;张 路(1989-),男,硕士生,主要研究方向为信息安全、数据挖掘、分布式计算,E-mail:zhanglu@antiy.cn;邓 丽(1990-),女,硕士生,主要研究方向为数据挖掘。

程序是否具有恶意属性的一种决策过程，这与数据挖掘的特点十分相似。数据挖掘能够自动寻找数据中的模式特点，然后使用所发现的模式来预测将来的数据，或者在各种不确定的条件下进行决策。而与传统检测方法不同的是，数据挖掘方法大多基于统计学，即通过分析海量样本的统计规律建立判别模型，从而让攻击者难以掌握免杀规律。近年来，一些学者将数据挖掘应用于恶意代码检测领域，并取得了一些成果[1,4,5,8,13,16]。常见的基于数据挖掘的恶意代码检测流程如图1所示。

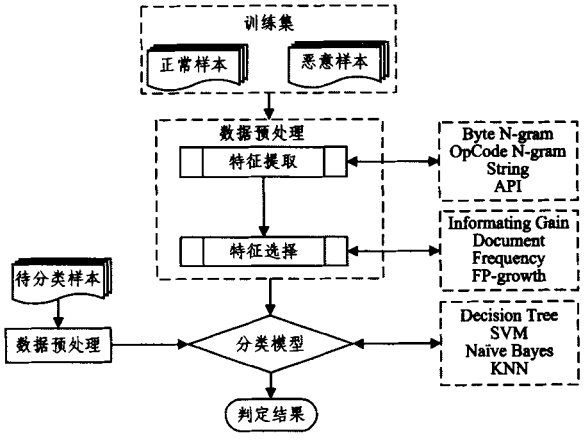

图1 基于数据挖掘的恶意代码检测流程图

该过程通常分为训练和预测两部分。在训练阶段，首先选取一批已知标签的正常样本和恶意样本构成训练集；然后进行数据预处理，包括特征提取和特征选择；最后利用得到的特征集训练分类模型。在预测阶段，将待分类样本通过数据预处理后输入训练完成的分类模型，最终得到相应的判定结果。

2 特征提取

在常见的数据挖掘应用场景中，特征的表现形式通常为数值、自然语言或逻辑关系等，而应用程序通常则是由逻辑关系复杂的代码及相关文件构成的。因此，如何对应用程序进行特征提取十分关键[3]。

2001年，Schultz等[5]首次将数据挖掘算法引入恶意代码检测领域，并采用二进制码表示应用程序，其中包括3种特征：Win32 DLL文件的调用、ASCII字符串和字节序列。最后，采用字节序列和字符串序列分别取得了96.88%和97.11%的正确率。Schultz的研究为该领域奠定了基础。

N元语法模型(N-grams)是一种基于统计语言模型(Statistical Language Model，SLM)的方法，在自然语言处理(Natural Language Processing，NLP)领域得到了广泛应用[6,19,20,30,31]。对于可执行文件，N-grams可作为一个长度为N的滑动窗口，收集一系列重叠子字节序列作为特征[8]。而与自然语言处理不同的是，这种方法所获取到的特征通常缺乏实际意义，解释性不强，但一些学者的确使用这种方法取得了不错成果[4,7]。

Abou-Assaleh等[7]首次发现基于字节序列的N-grams(Byte N-grams)可作为很好的特征，并选取L个频繁项集(Frequent Itemsets)作为特征，最终采用k-Nearest Neighbors(kNN)算法对65个可执行文件分类，得到98%的正确率。Kolter等[4]将可执行文件转换成十六进制ASCII形式，然后使用N-grams对每N个字节进行截取并分别转换为独立的序列，接着通过实验得出$N=4$时效果最好，并根据信息增益(Information Gain，IG)选出了500个最相关N-grams作为特征集合，最终通过Boosted J48得到99.58%的最高正确率。

另有学者提出了OpCode N-grams[8-11,19]，OpCode(Operational Code)是机器语言指令的一部分。Karim[9]率先使用OpCode N-grams进行特征提取，他认为OpCode序列比Byte序列的可靠性更高。Bilar[10]将67个恶意程序与20个正常程序进行反汇编(Disassemble)后，发现它们的OpCode序列的概率分布的区别很大，而且出现频率越低的OpCode代表性越强。Asaf Shabtai[8]将30000个文件用不同的OpCode N-grams集合来表示，并通过5种分类算法取得超过99%的正确率，该结果比他之前基于Byte N-grams的结果要好[12]，他认为字节序列相当于文本分类(Text Categorization)中的字母或字母序列，而OpCode序列则相当于是单词或单词序列，这使得OpCode序列比Byte序列有更明确的意义[8]。

字符串也可作为特征。LAI[13]提取了可执行文件中所有可输出字符串(Printable Strings)，并通过4种方法对所提取的特征进行评估，选出相关度最高的100个字符串作为特征，最终通过SVM取得99.38%的准确率。相对于N-grams，字符串特征集维度更低，意义更加明确，因此可获得更小的计算开销、更快的运行速度以及更高的准确率。

可执行文件所调用的API排列在PE文件头(Portable Executable File Head)中，针对不同API在恶意程序与正常程序中分布不同的特点，Ding Yuxin等[14]计算了各个API在所有可执行文件中的分布，然后通过信息增益进行特征选择。

总而言之，恶意代码的特征类型可概括为基于N-grams的序列类型，以及可输出字符串类型，广义上也包括API。

3 特征选择

特征质量对分类模型的准确率影响较大，而特征选择是特征质量的决定因素之一。通常特征提取后会得到冗余特征，它们将直接影响分类的准确率并增加计算复杂度[8,14]，甚至可能导致过拟合(Overfitting)[13]。因此，需要通过一种特征质量评价的方法选出相关度最高的特征。常见的特征选择算法有：信息增益、文档频率、频繁项集、TF-IDF(Term Frequency-Inverse Document Frequency)[21]等，它们大多借鉴于文本分类法[4,8,22]。

3.1 信息增益

信息增益是最常见的特征选择算法之一，它表示某个特征项出现或者不出现时为分类模型所带来的信息量的差别。其中，信息量由熵(Entropy)来度量[6]，信息增益越大，该特征的重要程度也就越高[23]。

Kolter首先在恶意代码检测领域引入信息增益，他将每个Byte N-grams在恶意程序或正常程序中出现与否用布尔变量表示，因此每个N-grams的信息增益值$IG(j)$可表示为：

$$IG(j)=\sum_{v_j\in\{0,1\}}\sum_{C_i}P(v_j,C_i)\log\frac{P(v_j,C_i)}{P(v_j)P(C_i)} \tag{1}$$

其中，C_i表示第i个类别，v_j表示第j个特征的布尔值，联合概率$P(v_j,C_i)$表示第j个特征出现在第i类中的概率，$P(v_j)$表示第j个特征出现在所有训练集中的概率，$P(C_i)$表示第i个类别占所有类别的比例。该算法也叫做互信息(Mutual

Information)。Kolter 最终根据 *IG* 值选取不同数量的 N-grams,将其输入 5 种不同分类模型,选出 500 个最佳的 N-grams。

3.2 文档频率

文档频率表示在训练集中包含某个特征项的文档的出现频率。这种衡量特征重要程度的方法是基于一种假设:出现频率较低的特征项对分类结果影响较小[20]。因此,通常设定一个阈值,当某个特征项的出现频率小于该阈值时,从特征空间中去掉该特征项。该方法实现简单,可降低计算复杂度,并能一定程度提高分类的准确率,但不符合信息检索理论:某些特征虽然出现频率低,却往往包含较大信息量,对分类的重要性很大[6]。

3.3 频繁项集

频繁项集的挖掘用于发现隐藏在大型数据集中有意义的联系。这种方法旨在通过特征间的相互关系挖掘出相关度较大的一些特征集[25,34]。

FP-growth 是一种不产生候选项集而采用频繁项集增长的方法来挖掘频繁项集的算法[15]。它首先将数据库存储在一种称为 FP-tree(Frequent Pattern Tree)的紧凑数据结构中,然后利用 FP-tree 来挖掘频繁项集。构建 FP-tree 时需要对原始数据集遍历两次,第一次会获得每个元素的出现频率,去掉不满足最小支持度的元素项;第二次将从空集开始,向其中不断添加频繁项集。最后从 FP-tree 中获得条件模式基,利用其构建一个条件 FP-tree,迭代到包含最后一个元素项为止。

上述方法中,IG 应用最为广泛,具有良好的通用性,对于大部分特征都有突出表现,尤其是针对解释性不强的 N-grams 序列,需要通过这种基于概率的方法来衡量每个特征项对各自类别的贡献大小。而 IG 的不足之处在于它考虑了特征项未发生的情况,特别是在正负类别样本数量以及特征值分布高度不均衡的情况下,*IG* 值将受到出现频率较低的特征项的极大干扰。而实际场景中,恶意样本与正常样本的数量分布存在失衡现象,即恶意样本数量远小于正常样本,这正是 IG 的风险点之一。

DF 相对于 IG 在理论上略显欠缺,但它的最大优势在于计算效率极高,对解决目前爆炸式增长的海量数据十分有利,特别适用于在线计算这类需要极高响应速度的应用场景。DF 的另一特点是它属于无监督算法,即不需已知类别信息的样本便可完成特征选择;而 IG 则需要依赖于大量先验信息。在实际场景中,对于新出现样本的类别标签的获取往往存在一定延时,这使得模型在特征选择上存在一定滞后。因而在这种场景下,无监督的特征选择方法尤为重要。

N-grams 序列特征的最大缺点是可解释性不强,无法描述函数调用的语义(Semantic)与控制结构的关系,缺乏对执行流程中上下文(Context)关系的充分利用。而可输出字符串则可以很好地弥补 N-grams 序列在这方面的不足,例如在 while()中若存在未成对出现的 recv()和 send(),则可能产生拒绝服务攻击,而不在循环结构中的相同函数则没有这样的语义[39]。可输出字符串能描述这种上下文关系的前提是,需要一种挖掘出字符串之间关联关系的方法,FP-growth 正是其中一种。FP-growth 是一种挖掘频繁项集的方法,也可用作特征选择,其本质是:挖掘出频繁且共同出现的特征子集,每个子集内的字符串之间共同描述了一定的上下文关系,而这种"频繁"且"共同出现"的关系是 IG、DF 所无法获取的。

另外,以上特征选择方法经常被同时使用,例如 Ding Yunxin[14]分别使用了 DF 和 IG 两种方法进行特征选择。他以 API 作为特征项,并将阈值设定为 0.6%,即去除 *DF* 值小于 0.6%的特征项,最终从 6181 个 API 中选取 *DF* 值最高的前 1000 个作为特征集;同时选取 *IG* 值最高的前 1000 个特征项,发现这些特征项与采用 DF 法选出的 1000 个特征项只有 351 个不同。最终通过这两种方法选出的特征项共同构成特征集。LAI[13]也同时使用了 DF 和 FP-tree 等方法进行特征选择。

4 分类模型的原理

所谓恶意代码检测,其本质是基于先验信息,对未知应用程序是否具有恶意属性的一种决策过程,因而机器学习中的分类算法起到了关键作用。数据预处理所得到的特征集可作为分类算法的输入,用以训练分类模型,训练完成后可将待分类样本输入模型,得到最终判定结果。常见的分类算法有决策树、支持向量机、朴素贝叶斯、k 近邻等。

4.1 决策树

决策树(Decision Tree)模型是一种描述对实例进行分类的树形结构,它通过 if-then 规则集合进行决策[23,33]。这里给出一个实例来说明决策树的生成过程。表 1 列举了 15 个客户的贷款申请信息,包括 4 个特征:客户的年龄、是否有工作、是否有房、信贷情况,以及申请结果。可利用这些少量的样本来学习一个贷款申请的决策树,用于对新客户贷款申请分类。

表 1 贷款申请样本数据表

序号	年龄	有工作	有房子	信贷情况	类别
1	青年	否	否	一般	否
2	青年	否	否	好	否
3	青年	是	否	好	是
4	青年	是	是	一般	是
5	青年	否	否	一般	否
6	中年	否	否	一般	否
7	中年	否	否	好	否
8	中年	是	是	好	是
9	中年	是	是	非常好	是
10	中年	是	是	非常好	是
11	老年	是	是	非常好	是
12	老年	是	是	好	是
13	老年	否	否	好	是
14	老年	否	否	非常好	是
15	老年	否	否	一般	否

特征选择是生成决策树的首要步骤,高质量的特征可极大提高决策效率,因此将优先使用这类特征进行划分。特征质量通常定义为:将无序的数据变得更加有序,可通过划分数据集前后信息量发生变化的大小来衡量,即信息增益,特征的信息增益值越高,其质量越高,这与 3.1 节所阐述的观点也是一致的。而在某些决策树算法中也会使用信息增益比(Information Gain Ratio)来评价特征质量。通过式(1)计算各个特征对数据集的信息增益,得到"是否有房"这个特征的信息增益值最大,为最优特征,并作为决策树的根节点。同理,计算剩余 3 个特征的信息增益值,得到"是否有工作"这个特征为当前最优特征,并作为父节点的子节点。同时,发现仅通过前两个特征的决策便可实现对数据集中所有样本的分类,而无

需依靠剩余两个特征，这也从侧面体现了特征选择的优势，可让决策更加简捷、高效。如此生成一个如图 2 所示的决策树，该决策树仅由两个特征构成[26]。

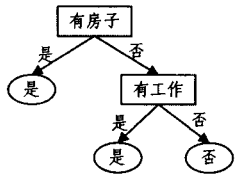

图 2　决策树模型

决策树生成后，为了提高泛化能力，避免过拟合，常采用剪枝(Pruning)的方法从已生成的树上裁掉一些子树或节点。根据不同的特征选择方法，经典的决策树变种有 J48、CART 和 C4.5 等。

随机森林(Random Forest)是决策树的一种改进算法。顾名思义，它是用随机的方式建立一个由若干决策树构成的森林，随机森林的每一棵决策树之间是没有关联的。其决策机制是：根据其中所有决策树的判定结果进行投票，票数最多的类别即为最终判定结果。决策树和随机森林都是恶意代码检测中常用的方法[1,4,8,16]。

4.2　支持向量机

支持向量机(Support Vector Machine，SVM)是一种典型的二分类(Binary Classification)模型。它是在特征空间中找到一个最优分离超平面 $\boldsymbol{w}\cdot\boldsymbol{x}+b=0$，将样本分到不同的类，分离超平面由法向量 $\boldsymbol{w}$ 和截距 b 共同决定[27,32]。在如图 3 所示的二维特征空间中，分别存在正、负两类样本，假设训练样本线性可分，那么图中的直线则表示最优分离超平面。一旦该超平面的法向量 $\boldsymbol{w}^*$ 和截距 b^* 确定后，对于新的未知样本 $\boldsymbol{x}$，将其代入：

$$f(x)=\text{sign}(\boldsymbol{w}^*\cdot\boldsymbol{x}+b^*) \tag{2}$$

即可通过 $f(x)$ 的正负情况判断该样本所属的类别。

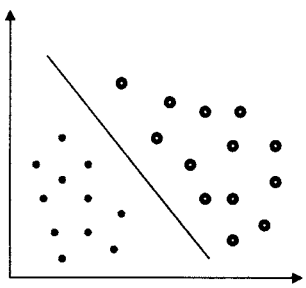

图 3　二维空间中的 SVM 模型

而对于非线性可分问题，可引入核技巧(Kernel Trick)将样本映射至高维空间，使得样本在高维空间线性可分。

4.3　朴素贝叶斯

朴素贝叶斯(Naïve Bayes)算法是基于贝叶斯定理与特征条件独立假设的分类方法[28]。它的特点是实现简单，学习与预测的效率都很高。对于给定的训练集，首先基于特征条件独立假设学习每个类别在训练集中的比例 $P(C_i)$ 和各类别下各个特征属性的条件概率估计 $P(x_j|C_i)$；然后基于此模型，对给定的输入 $\boldsymbol{x}=\{x_1,x_2,\cdots,x_n\}$，利用贝叶斯定理求出后验概率最大的输出 C：

$$C=\arg\max_{C_i}\frac{P(C_i)P(\boldsymbol{x}|C_i)}{P(\boldsymbol{x})} \tag{3}$$

而对于不同类别，$P(\boldsymbol{x})$ 都是相等的；另外由于假设特征间相互独立，即 $P(\boldsymbol{x}|C_i)=\prod_j P(x_j|C_i)$，因此式(3)可简化成：

$$C=\arg\max_{C_i}P(C_i)\prod_j P(x_j|C_i) \tag{4}$$

值得注意的是，如果待分类样本中未出现某个特征项 x_j，则 $P(x_j)=0$，导致 $\prod_j P(x_j|C_i)=0$，最终造成该样本属于任何类别的概率均为 0，而这点有时会不符合自然规律，因此需引入了平滑(Smoothing)项。Laplace Smoothing 是最常见的平滑方法之一，它对所有特征的条件概率加入了平滑因子 λ，从而一定程度上避免了零概率特征的产生。

4.4　k 近邻

k 近邻(k-Nearest Neighbor，kNN)是最简单的分类算法之一。它是一种基于距离的分类算法，该方法无需得到显式表达式。假设 n 维空间中存在由 m 个样本构成的训练集 $\boldsymbol{S}=\{\boldsymbol{s}_1,\boldsymbol{s}_2,\cdots,\boldsymbol{s}_m\}$，对于待分类样本 $\boldsymbol{s}_i$，在由训练集构成的特征空间中找到与之最近邻的 k 个样本，统计这 k 个样本的所属类别，属于同一类别样本个数最多的类别即为 $\boldsymbol{s}_i$ 的类别标签。而所谓的“近邻”实际上是一种距离的度量结果，特征空间中两个样本点的距离反映这两点的相似程度。常见的距离度量方式为欧氏距离：

$$D=\sqrt{\sum_{i=0}^{n}(x_i-y_i)^2} \tag{5}$$

其中，$\boldsymbol{x}=\{x_1,x_2,\cdots,x_n\}$，$\boldsymbol{y}=\{y_1,y_2,\cdots,y_n\}$ 分别表示两样本点，D 为两点的欧氏距离。

Schultz[5]使用 Naïve Bayes 和 Multi-Naïve Bayes 算法对比传统检测方法，发现数据挖掘方法比传统方法更加准确，从而为该领域奠定了基础。Kolter[4]后来使用了一系列分类算法，如 IBK(kNN)、Naïve Bayes、SVM、Boosted Naïve Bayes、J48、Boosted J48 等，最终 Boosted J48 取得了最好成绩。另外，Kolter 还证明了他所提出的 Byte N-grams 优于 Schultz 的特征提取方法。LAI[13]使用 FP-Tree 算法选出最相关的字符串作为特征，并根据所提取的特征，使用向量空间模型(Vector Space Model，VSM)[29]将样本集映射为由 0 和 1 构成的特征矩阵，其中 1 表示某个特征项出现在了该样本中，0 表示未出现。然后将特征矩阵输入 SVM 分类模型，使用 5 折交叉验证检测模型性能，最终以 99.38% 的准确率超过 Kolter 所使用的 SVM 方法。Igor Santos[40]同时使用了 kNN、J48、Random Forest、采用不同核函数的 SVM、Naïve Bayes 及其改进算法，以不同长度的 OpCode 为特征，最终 kNN 在准确率、误报率、AUC 值以及训练时间 4 项指标上均取得了绝对优势。Zhao Zongqu[16]同时使用了 J48、Bagging、Random Forest 等算法分别对病毒、木马、蠕虫 3 种类型的恶意代码进行检测，最终 Random Forest 以 96% 的准确率优于其他算法。Moskovitch[8]使用 DF 方法选取不同数量的 OpCode N-grams 作为特征集，然后采用 Random Forest、ANN(Artificial Neutral Network)、SVM、Naïve Bayes 等方法，最终 Random Forest 以大于 95% 的正确率表现最佳。

5　分类模型的性能比较

从上面的分析不难看出，各种方法在不同场景下均各有优势，没有绝对的优劣之分。而算法所表现的性能往往取决于应用场景，以及所选取的特征，或是样本自身的特性[36]。

通常，对于小规模数据集，SVM 的表现更为突出，这得益于它较高的准确率以及较强的泛化能力，并能通过核函数有

效解决线性不可分问题，同时对过拟合问题有很好的理论保证。但 SVM 较大的内存开销以及繁琐的调参过程，导致它并不合适处理大规模数据集。同样，不适合大规模数据集的还有 kNN，因为对于每一个待分类的样本，其都需要计算它到全体已知样本的距离，而一旦数据量增多，计算将十分缓慢。与之相反的是，Naïve Bayes 计算复杂度低，训练过程简单，对大规模数据集具有较高的计算效率；同时，由于 Naïve Bayes 从原理上依赖于极限定理，恰恰需要足够大量的样本来计算先验概率。据卡巴斯基实验室的统计，每日新增恶意样本量为 31.5 万个，如此大规模的数据量是对这些分类算法的考验，值得我们进一步探索。

从特征属性角度，Decision Tree 和 Random Forest 对特征的类型并没有严格限定，它可以同时处理字符串类型、数值型以及 OpCode N-grams 等不同类型的特征，也不需要考虑特征间量纲不同的问题。而 SVM 则必须将特征转换为数值型矩阵，如此便造成了特征量化的困难以及特征矩阵稀疏的可能性。但 Decision Tree 对样本空间的划分只能是垂直或平行于坐标轴的，而不能很好地解决倾斜于坐标轴的划分问题[36]。Naïve Bayes 建立在特征间相互独立的假设之上，而这种假设在实际中一般很难满足，因此该算法的效果往往难以达到理论上的最大值。

Naïve Bayes 的优势在于能以概率的形式解释判定结果。在实际场景中，往往需要以一定的置信度满足某种要求，例如样本被误报的代价可能高于漏报，对于恶意属性不确定的样本，需要模型“更倾向于”对其不做处理，即要求模型将样本判定为恶意的结果具有更高置信度，因此可将阈值在 0.5 附近做适当的偏移；亦或者对于恶意性不高的样本，其以概率的形式呈现，而是否需要做进一步处理，可由用户自行决策。

在某些场景下，需要建立多分类模型，例如将样本分为正常、蠕虫、病毒、木马 4 种类型。而 SVM 是典型的二分类模型，解决该问题的方法是通过组合多个二分类模型来完成多分类，典型的有“one against one”和“one against all”两种方法[35]。

在模型复杂度以及参数方面，kNN 有它独特的优势，它并不需要任何前期训练过程，而是在程序开始运行时将已知数据集全部载入内存即可开始计算。在参数方面，kNN 只有唯一参数 k 需要预先确定，通常采用交叉验证法。而其他算法都需要复杂的训练过程，尤其是 SVM 的核函数中相关参数更是缺少通用的方法来确定。而且，kNN 并不依赖于对数据集的任何假定，具有较强的通用性。

此外，kNN 能很好地解决模型维护问题。例如在实际中，随着时间变化，恶意样本的特征通常也会不断发生改变，因此不断更新训练集十分必要。由于 kNN 支持增量学习（Incremental Learning），当训练集中新增了一批样本后，对参数 k 的影响并不大，模型依然可以稳定地工作；而对于基于规则的模型（如 Decision Tree，Random Forest，SVM）来说，却意味着需要频繁更新复杂的模型，否则新增样本将不会对决策作出任何贡献。

而随之而来的问题是，当数据集不均衡时，例如某一类样本容量很大而其他类的容量很小时，很可能导致待预测样本邻域的 k 个样本中属于大容量类别的样本占多数，存在过拟合风险。相反的是，Decision Tree 可通过“剪枝”过程有效避免过拟合，同时具有较强的抗噪能力。另外，Random Forest 中每棵树的训练样本是随机的，树中每个节点的分类属性也是随机选择的，因而也不易产生过拟合。

总之，以上分类算法均存在各自的优缺点，并没有一种算法能在任何应用场景下具有绝对的优势，因此结合实际场景最为关键。同时，不可忽视的是特征提取与特征选择，这些共同决定了模型的最终表现。

6 评价方法

分类模型通常是基于已知的样本集训练得到的，而该模型对未知样本的预测能力则需要通过未出现在训练集中的测试集进行性能评估。通常引入如下指标评估分类模型的性能[5,13,14,40]：

1）真正类（True Positive，TP），将恶意程序判定正确的样本数量；

2）真负类（True Negative，TN），将正常程序判定正确的样本数量；

3）假正类（False Positive，FP），将正常程序判定错误的样本数量；

4）假负类（False Negative，FN），将恶意程序判定错误的样本数量。

根据以上指标，可对分类模型性能做总体评价：

1）准确率$=\frac{TP}{TP+FN}$

2）误报率$=\frac{FP}{TN+FP}$

3）总体准确率$=\frac{TP+TN}{TP+TN+FP+FN}$

不同类别被误判的代价不同，而准确率评估方法却默认所有的误判代价都是相同的，这点通常与实际场景不符，例如实际中误报一个样本的代价是高于漏报的。因此，一些学者引入 ROC（Receiver Operating Characteristic）曲线作为另一种评价方法[4,5,16]。ROC 曲线是以误报率为横轴、准确率为纵轴构成的二维空间中的一条曲线，用来刻画 TP 和 FP 间的折衷关系[17]。一个分类模型在一次实验中所产生的正负判定结果可形成一条 ROC 曲线，而多个分类模型对同一问题的实验结果将产生多条 ROC 曲线，此时可引入接收者操作特性曲线下的面积（Area Under Curve，AUC）来对各个分类模型进行总体评价，AUC 越大，其分类性能越好[6,20]。

为了防止数据集随机性的偏差或模型产生过拟合，常用的评价方法还有交叉验证（Cross Validation，CV）法[4,7,8,16,18,40]。其中最经典的方法是 K 折交叉验证（K-fold Cross Validation）法，其基本思想是：首先将数据集随机分成 K 个互不相交的子集，然后选取其中 $K-1$ 个作为训练集，剩下 1 个为测试集，对可能的 K 次不同组合重复这一过程，最终以 K 次实验的平均结果来衡量分类模型的性能。

结束语 基于数据挖掘的恶意代码检测领域已逐渐被广大学者以及一些安全厂商所重视，并取得了一定成功。这一领域虽然处于发展初期，但它的发展无疑会对恶意代码检测领域带来革命性的改变，与此同时它也将面临更多的挑战。

1）该领域的大多算法均借鉴于自然语言处理领域，而不同的是自然语言中的特征集（例如英文单词、汉子）是一个相对收敛的集合，一个有限的、可穷尽的集合。而在恶意代码领

域，无论是 OpCode、Byte Code，还是字符串等，都是一个相对发散的集合。这点是在借鉴方法的同时需要克服的困难。

2）许多文献中实验所用到的样本集数量有限，而现实场景中的样本更具有多样性，其特征更加丰富。因此，这些算法能否胜任实际中大规模样本的考验，有待验证。

3）本文所提到的检测算法均为分类算法，属于有监督（Supervised Learning）算法，即需要基于已知判定结果的样本进行训练，并需要不断更新训练集，以达到最佳的检测效果，这无疑会增加维护成本。因此，无监督学习（Unsupervised Learning）算法将是未来的研究重点。

参 考 文 献

[1] Lee D H, Song I S, Kim K J, et al. A study on malicious codes pattern analysis using visualization [C] // 2011 International Conference on Information Science and Applications (ICISA). IEEE, 2011: 1-5

[2] Zhang Jia, Guan Yun-tao, Jiang Xiao-xin, et al. AMCAS: An Automatic Malicious Code Analysis System[C] // Proc. of the 9th International Conference on Web-age Information Management. IEEE Press, 2008: 501-507

[3] Shabtai A, Moskovitch R, Elovici Y, et al. Detection of malicious code by applying machine learning classifiers on static features: A state-of-the-art survey [J]. Information Security Technical Report, 2009, 14(1): 16-29

[4] 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

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

[6] 宋宗成. 统计自然语言处理[M]. 北京：清华大学出版社，2013

[7] Abou-Assaleh T, Cercone N, Keselj V, et al. N-gram-based detection of new malicious code[C] // Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004 (COMPSAC 2004). IEEE, 2004, 2: 41-42

[8] Shabtai A, Moskovitch R, Feher C, et al. Detecting unknown malicious code by applying classification techniques on opcode patterns[J]. Security Informatics, 2012, 1(1): 1-22

[9] 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

[10] Bilar D. Opcodes as predictor for malware[J]. International Journal of Electronic Security and Digital Forensics, 2007, 1(2): 156-168

[11] Moskovitch R, Feher C, Tzachar N, et al. Unknown malcode detection using OPCODE epresentation[M] // Intelligence and Security Informatics. Springer Berlin Heidelberg, 2008: 204-215

[12] 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, 2008 (ISI 2008). IEEE, 2008: 156-161

[13] Lai Y. A feature selection for malicious detection[C] // Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008 (SNPD'08). IEEE, 2008: 365-370

[14] Ding Y, Yuan X, Tang K, et al. A fast malware detection algorithm based on objective-oriented association mining[J]. Computers & Security, 2013, 39: 315-324

[15] Wang Xin-yu, Du Xiao-ping, Xie Kun-qing. Research on Implementation of the FP-growth Algorithm[J]. Computer Engineering and Application, 2004, 40(9): 174-176 (in Chinese)
王新宇，杜孝平，谢昆青. FP-growth 算法的实现方法研究[J]. 计算机工程与应用，2004，40(9)：174-176

[16] Zhao Z, Wang J, Wang C. An unknown malware detection scheme based on the features of graph[J]. Security and Communication Networks, 2013, 6(2): 239-246

[17] Wang Yun-yun, Chen Song-can. A Survey of Evaluation and Design for AUC Based Classifier[J]. Pattern Recognition and Artificial Intelligence, 2011, 24(1): 64-71 (in Chinese)
汪云云，陈松灿. 基于 AUC 的分类器评价和设计综述[J]. 模式识别与人工智能，2011，24(1)：64-71

[18] Komashinskiy D, Kotenko I. Malware detection by data mining techniques based on positionally dependent features[C] // 2010 18th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP). 2010: 617-623

[19] Brown P F, Desouza P V, Mercer R L, et al. Class-based n-gram models of natural language[J]. Computational Linguistics, 1992, 18(4): 467-479

[20] Cavnar W B, Trenkle J M. N-gram-based text categorization[J]. Ann Arbor MI, 1994, 48113(2): 161-175

[21] Sebastiani F. Machine learning in automated text categorization [J]. ACM Computing Surveys (CSUR), 2002, 34(1): 1-47

[22] Su J S, Zhang B F, Xu X. Advances in machine learning based text categorization[J]. Journal of Software, 2006, 17(9): 1848-1859 (in Chinese)
苏金树，张博锋，徐昕. 基于机器学习的文本分类技术研究进展[J]. 软件学报，2006，17(9)：1848-1859

[23] Mitchell T M. 机器学习[M]. 北京：机械工业出版社，2003

[24] Yang Y, Pedersen J O. A comparative study on feature selection in text categorization[C] // ICML. 1997, 97: 412-420

[25] Jiawei H, Kamber M. Data mining: concepts and techniques [M]. San Francisco, CA, itd: Morgan Kaufmann, 2001

[26] 李航. 统计学习方法[M]. 北京：清华大学出版社，2012

[27] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning, 1995, 20(3): 273-297

[28] Kim Y H, Hahn S Y. Text filtering by boosting naive Bayes classifiers[C] // Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2000: 168-175

[29] Salton G, Wong A. A vector space model for automatic indexing [J]. Communications of the ACM, 1975, 18(11): 613-620

[30] Zhang Xiao-kang. The Study of Malicious Code Detection Based on Data Mining and Machine Learning[D]. Hefei: School of automation, University of Science and Technology of Chaina, 2009 (in Chinese)
张小康. 基于数据挖掘和机器学习的恶意代码检测技术研究[D]. 合肥：中国科学技术大学自动化学院，2009

[31] Siddiqui M, Wang M C, Lee J. Data mining methods for malware detection using instruction sequences[C] // Proceedings of Artificial Intelligence and Applications (AIA). 2008

[32] Gu Ya-xiang, Ding Shi-fei. Advances of Support Vector Machines[J]. Computer Science, 2011, 38(2): 14-17 (in Chinese)

（下转第 56 页）

在利用 Ranking SVM 进行学习时，如果学习的样本不够多，则可能对实验结果产生负面影响。虽然实验结果证实，通过引入这种跨网络关联模式能将两个不同平台相关联，更好地解决跨平台冷启动推荐问题，但是如何解决用户随机性造成的关联模式误差，使跨网络关联模式更加稳定有效，都还有待进一步思考和解决。在往后更深入的研究中，我们计划采用数量更加庞大的用户群进行实验，并对用户的行为信息进行筛选，选取对异构知识关联贡献更大的信息数据，以减小用户随机性造成的误差并提高实验效率，从而提高跨网络关联的准确性和有效性。

结束语 本文提出了一种基于关联规则挖掘的跨网络知识关联方法，即利用跨网络关联用户的集体智慧，运用关联规则构建异构网络平台的关联模式，使不同网络的异构行为能在用户层上进行跨网络关联，同时通过引入主题模型和用户感知，使该关联突破语义关联的局限性，在更细的粒度下进行感知。在这种关联模式下，我们提出一种基于权重学习的 YouTube 视频推荐应用，将关联模式与机器学习相结合，对新用户进行冷启动视频推荐。通过实验证明，本文提出的关联模式是有效的，这种跨网络知识关联模式能更好地理解和满足用户需求，有助于跨网络的个性化服务。在未来的研究中，我们将使用更多具有丰富行为信息的用户信息进行关联模式的构建，也期待能提出更好的算法挖掘跨网络关联模式。

参 考 文 献

[1] Sang Ji-tao, Lu Dong-yuan, Xu Chang-sheng. Overlapped user-based cross-network analysis: Exploring variety in big social media data[J]. Science Chinese, 2014, 59(36): 3354-3560(in Chinese)
桑基韬，路冬媛，徐常胜. 基于共同用户的跨网络分析：社交媒体大数据中的多源问题[J]. 中国科学，2014，59(36)：3554-3560

[2] Roy S D, Mei Tao, Zeng Wen-jun, et al. Socialtransfer: Cross-domain transfer learning from social streams for media applications [C]// Proceedings of the 20th ACM International Conference on Multimedia. Noboru Babaguchi, Kiyoharu Aizawa, 2012: 649-658

[3] Abel F, Araújo S, Gao Q, et al. Analyzing cross-system user modeling on the social Web[C]// Proceedings of the 2011 IEEE International Conference on Multimedia and Expo. Barcelona, Spain, 2011: 28-43

[4] Abel F, Gao Qi, Houben G J, et al. Analyzing user modeling on twitter for personalized news recommendations [M] // User Modeling, Adaption and Personalization. Berlin: Springer Berlin Heidelberg, 2011: 1-12

[5] Szomszor M N, Cantador I, Alani H. Correlating user profiles from multiple folksonomies[C]// Proceedings of the nineteenth ACM conference on Hypertext and hypermedia. Boston, Academic, 2008: 33-42

[6] Yan Ming, Sang Ji-tao, Xu Chang-sheng. Mining Cross-network Association for YouTube Video Promotion[C]// Proceedings of the ACM International Conference on Multimedia. New York, USA, 2014: 557-566

[7] Blei D M, Ng A Y, Latent M I. Dirichlet Allocation[J]. Journal of Machine Learning Research, 2003, 3: 993-1022

[8] Zhang Yong-zheng, Pennacchiotti M. Recommending branded products from social media[C]// Proceedings of the meeting of RecSys. 2013: 77-84

[9] Winoto P, Tang T. If You Like the Devil Wears Prada the Book, Will You also Enjoy the Devil Wears Prada the Movie? A Study of Cross-Domain Recommendations[J]. New Generation Computing, 2008, 26(3): 209-225

[10] Pan W, Liu N N, Xiang E W, et al. Transfer Learning to Predict Missing Ratings via Heterogeneous User Feedbacks[C]// Proceedings of the Twenty-Second International Joint Conference on Artificial Intelli. Barcelona, Catalonia, Spain, 2011

[11] Joachims T. Optimizing search engines using clickthrough data [C]// Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA, 2002: 133-143

[12] Herlocker J L, Konstan J A, Terveen L G, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 5-53

(上接第 18 页)

顾亚祥，丁世飞. 支持向量机研究进展[J]. 计算机科学，2011，38(2)：14-17

[33] Liang Dao-lei, Huang Guo-xing, Jin Jian. A New Multivariate Decision Tree Algorithm[J]. Computer Science, 2008, 35(1): 211-212(in Chinese)
梁道雷，黄国兴，金健. 一种多变量决策树方法研究[J]. 计算机科学，2008，35(1)：211-212

[34] Liu Jun-qiang, Sun Xiao-ying, Pan Yun-he. Survey on Association Rules Mining Technology[J]. Computer Science, 2004, 31(1): 40-47(in Chinese)
刘君强，孙晓莹，潘云鹤. 关联规则挖掘技术研究的新进展[J]. 计算机科学，2004，31(1)：40-47

[35] Hsu C W, Lin C J. A comparison of methods for multiclass support vector machines[J]. IEEE Transactions on Neural Networks, 2002, 13(2): 415-425

[36] Kotsiantis S B, Zaharakis I, Pintelas P. Supervised machine learning: A review of classification techniques[J]. Informatica, 2009, 33(3): 249-268

[37] 国务院. 中华人民共和国计算机信息系统安全保护条例[Z]. 1994

[38] 傅建明，彭国军，张焕国. 计算机病毒分析与对抗[M]. 武汉：武汉大学出版社，2004

[39] Li Jia-jing, Liang Zhi-yin, Wei Tao, et al. A Malicious Behavior Analysis Method Based on Program Semantic[J]. Acta Scientiarum Naturalium Universitatis Pekinensis, 2008, 44(4): 537-542 (in Chinese)
李佳静，梁知音，丰韬，等. 一种基于语义的恶意行为分析方法[J]. 北京大学学报：自然科学版，2008，44(4)：537-542

[40] Santos I, Brezo F, Ugarte-Pedrero X, et al. Opcode sequences as representation of executables for data-mining-based unknown malware detection[J]. Information Sciences, 2013, 231: 64-82