当一棵树落下时：在集合分类器中使用分集来识别恶意软件探测器的规避行为

|  |  |
| --- | --- |
| Charles Smutz  乔治梅森大学csmutz@gmu.edu | Angelos Stavrou  乔治梅森大学astavrou@gmu.edu |

*摘要 -*机器学习分类器是现代恶意软件和入侵检测系统的重要组成部分。然而，过去的研究表明基于分类器的检测系统在实践中容易受到逃避攻击。提高基于学习的系统的逃避阻力是一个悬而未决的问题。为了解决这个问题，我们引入了一种用于识别集合分类器表现不佳的观测的新方法。在检测过程中，当来自各个分类器的足够数量的投票不同意时，集成分类器预测显示为不可靠。所提出的方法 - 集合分类器相互协议分析允许在没有额外的外部事实真相的情况下检测许多形式的分类器逃避。

我们使用PDFrate，一款PDF恶意软件检测器来评估我们的方法。将我们的方法应用于从真实网络中获取的数据，我们表明绝大多数预测可以用高集成分类器协议进行。然而，大多数分类器逃避尝试，包括来自两项最近研究的九个有针对性的模拟情景，都给出了不确定的结果，表明这些意见不能由分类器给出可靠的预测。为了展示我们方法的普遍适用性，我们针对Drebin Android恶意软件检测器进行了测试，在该检测器中，对大多数新颖攻击正确地给出了不确定预测。我们的评估包括超过100,000个PDF文档和100,000个Android应用程序。此外，我们证明我们的方法可以推广到弱化支持向量机的梯度下降和核密度估计攻击的有效性。我们发现特征装袋是实现基于集合分类器多样性的逃避检测的最重要属性。

# Introduction

机器学习的使用已经成为解决各种渎职和恶意活动的主要技术之一。机器学习的应用包括恶意软件家族的聚类[7]，[20]，恶意下载的检测[12]，[34]，社交网络中帐户滥用的检测[14]，[44]以及检测常用文件格式，如Java档案[36]和文档[24]，[25]，[39]。此外，统计或机器学习技术已成功用于确定垃圾邮件[11]，[21]，[35]多年。

在敌对环境中使用机器学习分类的系统的主要弱点之一是它们易受到逃避攻击。对于规避攻击，我们指的是利用有关机器学习系统如何运行的知识的攻击类别，并且在许多情况下利用对训练集和特征的访问来被动地或主动地逃避检测[8]，[9] ，[15]，[33]，[45]。

用于对机器学习者进行回避攻击的常用技术是模仿。根据入侵检测系统使用的模型，通过使攻击数据显得良性，模仿攻击会阻止检测。这通常是通过编码或加密来隐藏恶意内容来实现的[28]，或者通过数据滥用或代码重用攻击来最小化恶意内容的覆盖范围[17]，[37]。例如，添加与良性观察对齐的内容以掩盖或淹没恶意内容。许多检测系统通过利用检测系统和受保护系统的差异而被避开[16]，[19]。即使防务系统的操作细节是保密的，但通常只能从外部测试中获得足够的知识来逃避[18]。

我们的方法并不是要阻止所有可能的逃避攻击，而是要引入一种机制来检测劣质分类器的性能。我们分析在集成分类器中使用自省以检测分类器在分类时何时提供不可靠结果。集合分类器相互协议分析的使用依赖于这样的直觉：当集合中的单个分类器为相同的预测投票时，预测可能是准确的。当足够数量的选票相反时，则分类器预测不可靠。在这种内部分类器不一致的状态下，检测器返回不确定的结果，而不是预测良性或恶意。在操作中，以一小部分被标记为不确定的样本为代价来改进对分类器预测的置信度，表明分类器不适合提供准确的响应。这种分离的准确不确定预测的预测是可能的，因为大部分错误分类（包括逃避尝试）都具有与准确预测不同的分类器投票分数分布。

为了评估我们的技术，我们将相互协议分析应用于两个研究深入的恶意软件检测系统：PDFrate [40]和Drebin [4]。PDFrate使用派生于随机森林分类器中的文档结构和元数据派生的特征来检测特洛伊木马PDF。PDFrate用于真实世界的入侵检测系统，可以通过提交给pdfrate.com进行评估。之所以选择PDFrate是因为它可公开访问，有据可查，使用了一个返回原始投票分数的集合分类器，并且已经受到多个最近发布的模仿攻击[26,27,43]。我们的评估包括来自操作环境的超过100,000份文档以及来自两次独立逃避研究的九种独特逃避场景中的数百种恶意文档。为了演示我们方法的普遍适用性，我们使用超过100,000个应用程序对Drebin Android恶意软件检测器进行了相互协议分析，其中包括超过20个标记为恶意软件系列的超过5,000个恶意应用程序。我们发现相互协议分析能够识别否则无法可靠检测到的新型恶意软件。

在使用支持向量机（SVM）作为基分类器构建避让集合时，我们发现特征装袋或者构造具有整个特征集的随机子集的许多单独分类器对于提供这种区分能力是至关重要的。使用这种方法，我们对抗梯度下降和核密度估计（GD-KDE）攻击，这对于传统的SVM分类器是非常成功的。

R ELATED W ORK

敌对学习是一个活跃的研究课题[18]。一些研究已经提出了创建有效的基于分类器的入侵检测系统的方法[6,15,16]。许多研究已经在训练时间解决了数据消毒或敌对影响的重要性[5]，[13]，[23]，[30]。还有一些人则关注逃避部署的分类器[8]，[27]，[43]。我们还专注于在操作过程中逃避分类器，但我们并没有专注于逃避策略，而是**提出了一种检测这些逃避尝试的方法**。

基于人口知识的信心估计一直是静态方法的基础[31]。当代研究已经证明了这些置信估计如何应用于在线设置中部署的基于机器学习的分类器[38]。然而，由于这些方法依赖于与已知基础真实的训练样本的分布相匹配的新观察，所以它们不适用于面临新观察和模仿攻击的入侵检测系统。我们不是试图量化分类器的整体准确性，而是确定分类器无法提供可靠响应的个体观察结果。我们的方法利用了分类器已经提供的数据，而没有对基础事实或独立异常值分析的额外吸引力。

最近的研究表明，集成分类器的多样性可以提高恶意软件的检测率[22-29] [46]。然而，很少有研究提出检测对这些集合分类器进行规避行为的实用策略。Chinvale等人 提出使用少量独立SPAM滤波器之间的相互协议来优化由于输入数据漂移而需要的单个分类器重新训练[11]。我们将这种方法扩展到集成分类器的反思，以便在测试时提供每个观察置信度估计值。我们与Chinvale等人有根本的不同。因为他们使用他们的合唱团的大多数结果作为重新训练单个分类器的基础事实，而我们专注于确定集合预测不可信的具体例子。简而言之，Chinvale等人 使用合唱团的多样性来提高分类器的性能。我们使用多样性来确定何时需要诉诸外部的基本事实。我们研究了使用套袋变化来实现集合中基于多样性的信心估计的因素。除了自然漂移或新型攻击之外，我们还将相互协议分析应用于针对特征提取器，训练数据和特定分类器的攻击组成的目标回避企图。

我们的实证评估依赖于目前在基于机器学习的恶意软件检测器[4]，[40]和模仿攻击[26]，[27]，[43]中的研究。我们试图减轻这些恶意软件检测器的漏洞。

III B ACKGROUND

我们将相互协议分析应用于两个恶意软件检测器：PDFrate和Drebin。我们对PDFrate的研究包括对PDFrate的两种模仿攻击：Mimicus和Reverse Mimicry。

*A. PDFrate*

PDFrate是一个基于机器学习的恶意软件检测器，在PDF文档上运行。pdfrate.com网站允许用户提交并为这些提交的文件返回评级。PDFrate对于这项研究非常有用，因为其基础机制已有详细记录[40]，它可公开提供在线攻击，并提供有关每个提交的PDF的大量信息。由于这种透明度，PDFrate一直是实际模仿攻击研究的目标[26]，[27]，[43]。

PDFrate根据对结构和元数据属性的分析对PDF文档进行分类。恶意文档的风险因素包括诸如存在Javascript对象或格式不正确的时间戳等项目。另一方面，良性文档包含惰性内容，例如文本内容或字体对象。使用应用于原始文档的正则表达式提取基本的结构和元数据信息。从文档中提取的这一小部分结构信息在文档扫描报告中显示给用户。从这些基本信息中提取功能。功能的例子包括Javascript对象的数量和文档中文件结束标记的相对位置。总而言之，使用了202个功能。

随机森林被用作PDFrate中的分类器。随机森林由数百或数千个单独的分类树构建而成。对于每棵树，训练集的一个子集用于构建。在树中的每个节点上，试图确定哪些特征和阈值最好地划分类。重复该过程直到每个叶节点包含单个类。因此，在随机森林中，每棵树都基于随机选择的训练数据子集和特征。新的观察结果贯穿每棵树，叶节点决定该树的投票。PDFrate的一个区别特征是它提供评分或评分，而不是简单的良性/恶意评估。PDFrate提供的分数是投票给正面（恶意）类的树的一部分。

PDFrate网站根据多个训练集提供来自分类器的分数。Contagio数据集取自研究人员指定的广泛可用数据集[32]。它包含10,000个文档，在良性和恶意之间平分。该数据集中的文档列表公开发布。第二套数据由乔治梅森大学的研究人员撰写，被称为大学数据集。它包含了数量超过100,000的文档，但是训练集的确切组成并未公布。在这项研究中，我们使用这两种训练集以及从它们派生的分类器。

*B. Mimicus*

Mimicus [1]是对PDFrate进行模仿攻击的框架。这是Srndi c和Laskov所描述的“部署的基于学习的系统的第一次经验性安全性评估”[43]的实现。这是一个独立的，全面的，公开可用的框架，用于在线实施PDFrate的攻击。

Mimicus通过修改现有的恶意文档来实现模仿攻击，使其看起来更像良性文档。Mimicus为文档添加额外的结构和元数据项目标记。这些附加内容不涉及添加由符合标准的PDF阅读器解释的实际内容，而是这些附加内容利用PDFrate的特征提取器中的弱点。无用的PDF属性被添加到文档尾部（文档结尾处的结构）之前的松弛或未使用的空间中，这是PDF规范不禁止的。这种方法在回避攻击中提供了相当大的灵活性，因为附加元素不一定有效。Mimicus为攻击者提供了一个简单的过程。攻击者构造一个恶意文件而不关心PDFrate逃避。Mimicus然后添加必要的诱饵结构元素。这种模仿攻击只会将伪元素添加到文档文件中 - 不会删除或修改现有的元素。

Mimicus通过将恶意文档与多个不同的良性文档进行比较来构造这些诱饵元素。调整恶意文档的特征向量以反映良性文档的特征向量。这些调整是由Mimicus使用的修改方法限定的。候选模仿特征向量通过本地PDFrate副本运行以确定分数。选择最佳特征向量。通过添加诱饵结构和元数据元素，该特征向量被用作修改原始恶意文档的目标。由于相互关联的特征和其他复杂性，构建完全匹配目标模仿特征向量的最终模仿恶意文档是不可行的。由此产生的恶意文档的特征向量介于原始木马文档和良性文档之间。模仿文档创建完成后，将其提交给pdfrate.com进行评估。

Mimicus研究的一个重要观察结果是，PDFrate特征的相互依赖性使得模仿攻击更加困难，因为修改一个特征必然会影响其他特征。一般认为，机器学习方法不需要不相关或冗余的特征。然而，在PDFrate的情况下，冗余特征看起来像通过Mimicus实现的那样进行规避攻击，通过使构建与目标特征向量匹配的PDF更加困难更困难。

Mimicus攻击模型需要知道PDFrate使用的特征集。前提是模仿攻击要成功，至少需要了解特征的类型。另外，由于这种攻击利用了普通PDF阅读器和PDFrate特征提取器之间的差异，因此如何利用这种差异的知识也是必要的。因此，所有Mimicus攻击场景都标有“F”，表示攻击者使用了该功能集的知识。

依靠特征提取的共同基础，Mimicus攻击展示了攻击者使用的各种知识水平。在攻击者知道训练数据和分类器的情况下，使用非常接近原始的副本。当对系统知识有限的攻击者进行建模时，会采用合理的替代品。标签“T”和“C”分别用来表示攻击者对训练数据和分类器的知识。因此，标签为“FTC”的攻击场景表示攻击者对PDFrate的三个主要方面的知识。

Contaigo分类器使用的PDFrate训练集已公开记录，并且随时可供研究人员使用。因此，在攻击者知道训练数据的攻击场景中，PDFrate和Mimicus使用相同的数据集。对于攻击者不了解训练集的情况，Srndi c和Laskov编写了一份代理培训集，其中包含源自VirusTotal的恶意文档以及源自Internet的良性文档。此外，他们从Contagio培训集中选择了100份恶意文档作为基准攻击文档。为了复制结果，列出了Srndi c和Laskov使用的所有数据集。

最后，为了完成离线PDFrate副本，Srndi c和'Laskov在知道分类器的知识时使用了随机森林分类器，并使用支持向量机分类器来模拟天真攻击者的情况。Mimicus的研究表明，尽管各种实施差异，但当PDFrate的三个细节都被欺骗后，结果与PDFrate online和Mimicus脱机分类器的分数几乎相同。Mimicus还实施GD-KDE攻击，旨在直接攻击SVM替代分类器。此攻击不适用于随机森林分类器，因此不会直接应用于PDFrate。

*C.反向模仿*

Maiorca等人。也研究了对PDFrate和其他PDF文档分类器的规避[26]，[27]。他们推进反向模仿技术。为了使内容看似良性（如Mimicus所做的那样），他们将恶意内容嵌入到良性PDF中，而不是将内容添加到恶意文档中，而是尽可能少地修改。反向模仿攻击实施针对PDFrate的独立逃避方法。

Maiorca等人提出了三种不同的逃避方案。在EXEembed方案中，恶意可执行文件被植入到现有的良性PDF文档中。打开文档时会执行恶意软件。这些文档利用CVE-2010-1240。在PDFembed方案中，恶意PDF被嵌入到良性PDF中。这些嵌入式文档在文档打开时自动呈现。为了评估，Maiorca等人 将利用CVE-2009-0927的文档嵌入到现有的良性PDF文档中。最后，在JSinject场景中，恶意JavaScript（与PDFembed嵌入式文档中使用的相同）直接注入根良性文档。

为了逃避检测，反向模仿攻击重点在于尽可能少地改变文档结构。例如，在EXEembed攻击中，一个新的逻辑版本的PDF被构造成几乎没有新的结构元素，但是原始PDF中的所有内容都保留在文件中。符合标准的阅读器不会显示与以前版本文档相关的内容，但是文档将由PDFrate和类似检测器的特征提取器进行分析。

除了最大限度地减少麦克罗德注射剂的结构伪影外，Maiorca等 利用PDF编码，特别是流压缩来隐藏插入的内容。例如，在PDFembed攻击中，恶意文档被嵌入到压缩的PDF流中。不解压PDF流的检测工具（如PDFrate）无法从嵌入式恶意PDF中提取功能。

Mimicus和Reverse Mimicry攻击使用两条独立路径来逃避PDFrate。Mimicus使用不会被普通PDF阅读器处理的诱饵对象的添加，而是通过基于简单正则表达式的PDFrate处理进行分析。另一方面，Reverse Mimicry攻击使用有效的PDF结构来最小化和隐藏恶意指标。Mimicus通过添加伪装来操作，而反向模仿攻击试图使恶意元素变为隐身。Mimicus利用PDFrate的丰富知识，而Reverse Mimicry方法使用PDF文件格式特有的数据隐藏技术。

*D. Drebin Android恶意软件检测器*

集合分类器相互协议分析应适用于所有可能逃避的情况，包括其他恶意软件分类器。我们评估了Drebin Android恶意软件检测器上相互协议分析的效用[4]。Drebin补充了PDFrate，因为它在软件包而不是文档上运行，并使用许多基于字符串的特征而不是数字特征。由于原始Drebin研究中使用的数据可供研究人员使用，因此我们将此数据用于评估。

Drebin通过执行快速扫描来从Android应用程序清单和反汇编代码中提取功能。这些功能被格式化为字符串。从清单中提取的功能包括硬件组件的名称/值，请求的权限，应用程序组件和意向（消息框架）。API调用，使用权限和网络地址/ URL的值取自反汇编代码。字符串值被映射到包含超过500,000个唯一值的二进制特征向量。

线性支持向量机离线培训，用于提供权重

（与超平面的距离），以便在分类过程中观察到每个特征。将每个预测变量权重组合起来以提供总体分数并与阈值进行比较以确定结果。由于这个方案，Drebin提供了一个恶意分数，并且可以识别出对这个分数有贡献的变量。对Drebin进行了超过100,000个良性和5,000个恶意样本的评估，提供1％的误报率和近94％的恶意软件检测率。

IV APPROACH方法

整体分类器由许多基类分类器构成。为了在集合中提供有意义的多样性，每个单独的分类器使用诸如对训练数据和特征的随机抽样（装袋）的机制来构建。通常情况下，结果通过投票结合起来，每个独立分类器得到相同的投票。将选票数相加得出一个分数。如果得分超过50％，那么观察被标记为恶意。否则，结果是良性的。

在许多使用案例中，组合已被证明可以提高准确性，包括恶意软件检测。然而，我们发现集合分类器的主要优势在于它们可以提供内部一致性的度量，该度量用作分类器对各个预测的置信度的估计。

在良好的预成型集合中，大多数单个分类器提供相同的投票。如果基础分类器提供相互矛盾的投票，那么集合处于不一致的状态，并且预测不太可靠。对集体中的个人贡献者进行投票的一致或不同意提供了对集合预测的置信度的估计。

分类器可能无法为某些观察结果提供准确的响应。例如，当在传统乐团中发生50/50投票分裂时，使用诸如随机选择的方法来提供预测。大多数应用程序会在分类器与所有贡献者为同一个类投票时发生的分歧完全不一致时对待随机选择的预测。但是，在完全不同意的情况下，唯一合理的解释是分类器无法做出合理的预测。

集合分类器的多样性是促进基于双方协议的信心估计的核心属性。这种多样性是由各个分类器的外推引起的。除了分类器方案和特征质量的局限性，当观察接近训练集中的样本时，分类得到很好的支持，应该是准确的。然而，随着新观测值与训练样本的距离越来越远，分类器被迫推断。对于有效地使用套袋的集合分类器，来自分类器训练的更新的观察结果，在集合中将存在更多的分歧。

外推中的这种多样性可以在PDFrate中使用的基于随机森林的分类器中观察到。表I显示了应用于各种模仿攻击的Contagio分类器中前25棵树（分数为1000）的分类性能。报告的表现相对于正确班级的森林平均得票数，除以± 0.5标准差。据观察，绝大多数树木都具有

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表I                                                   - [R IN个别树木的ELATIVE性能Ç ONTAGIO CLASSIFIER表明为（+），BELOW （ - ）， 或在（0）0.5  森林平均值的标准偏差   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 规避情景 |  |  |  |  |  |  |  |  |  | 个人树木表现 | | | | | | |  |  |  |  |  |  |  |  |  | | F模仿 | 0 | + | + | - | 0 | 0 | - | + | 0 | + | - | 0 | + | - | + | 0 | 0 | 0 | - | + | + | + | + | - | - | | FC模仿 | + | + | + | - | + | 0 | - | + | 0 | + | - | - | + | 0 | 0 | 0 | 0 | 0 | 0 | + | + | + | 0 | - | - | | FT模仿 | 0 | + | + | - | - | 0 | 0 | + | 0 | 0 | - | 0 | 0 | 0 | + | - | - | 0 | 0 | + | + | 0 | 0 | - | 0 | | FTC模仿 | - | + | + | - | 0 | + | 0 | - | - | + | 0 | - | + | 0 | + | + | - | + | 0 | + | - | + | 0 | - | 0 | | F gdkde | - | + | + | + | + | + | - | - | + | + | 0 | 0 | + | - | + | - | - | + | - | + | 0 | - | - | - | 0 | | FT gdkde | + | + | + | + | 0 | + | - | - | + | + | + | - | + | + | - | - | - | 0 | - | 0 | - | + | + | - | 0 | | JSinject | + | - | - | 0 | + | + | - | 0 | + | + | + | 0 | 0 | + | 0 | 0 | 0 | + | - | 0 | + | - | 0 | + | - | | PDFembed | 0 | - | - | + | 0 | 0 | 0 | - | - | - | - | + | + | - | - | - | - | - | 0 | - | - | - | - | + | - | | EXEembed | - | 0 | 0 | - | - | - | + | 0 | + | 0 | - | - | - | + | 0 | + | - | - | + | + | - | 0 | 0 | - | - | |

                   表二。              E NSEMBLE C LASSIFIER O UTCOMES

|  |  |  |  |
| --- | --- | --- | --- |
| 投票分数 | 结果 |  | 规避类型 |
| [0,25] | 良性 |  | 强回避 |
| （25,50） | 不确定 | （良性） | 弱回避 |
| [50,75） | （恶意） |
| [75，100] | 恶毒 |  | 没有逃避 |

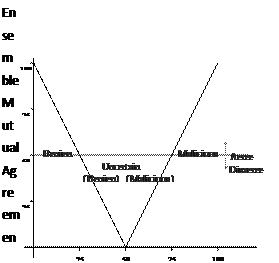
取决于逃避情景的三个结果：平均值（0），低于平均值（ - ）和高于平均值（+）。因此，当应用于远离训练数据的数据时，每棵树的精度在观察值之间变化很大。没有普遍强或弱的树木。当远离训练数据进行外推时出现的随机噪声是什么使相互协议分析成为可能。

为了一般地应用相互协议分析，除了良性和恶意的预测之外，我们还提出了一个不确定的新结果。例如，我们不把投票区域分成两半，而是将它分成4个象限。在0％到25％的地区，大多数选票认为结果是负面的（良性的）。同样，在75％到100％的地区，大部分选票都认为结果是肯定的（恶意）。但是，如果得分在25％和75％之间，个别分类不同意，结果是不确定的。为了支持与简单集合投票预测的比较，该区域可以分为另外两个象限：不确定性（良性）从25％到50％，不确定（恶意）从50％到75％。表II显示了这些分类结果。

为了更准确地说明这个概念，我们引入一个度量标准来量化集合分类器中单个投票之间的一致性：

*A* = | *v* - 0 *。*5 | \* 2

其中*A*是集合分类器双向协议率，*v*是任一类别的投票部分。图1展示了该功能，该图还显示了50％相互协议阈值导致的分类结果。末端和中间点驱动此功能的一般形状。如果分类投票比率为0或1，那么分类者就结果达成完全一致，相互协议应为1（或100％）。如果分类器按照每个类别的0.5票进行分割，那么相互协议应该至少为0（或0％）。只要使用单个阈值，在这些端点和中点之间的线所用的形状是什么并不重要 - 任何连续的曲线将允许选择给定的



集合分类器票数（％）图1基于集合投票结果的相互协议

分类器投票分数的阈值。例如，该功能不需要遵循分数分布。我们选择一个线性函数是因为它很简单。

相互协议的界限是分类器被认为处于集体协议状态的界限，并且所得到的分类应该被认为是有效的。在这个相互协议评级下面，分类是似是而非的。在本文的大部分篇幅中我们使用了50％的边界。但是，这个值应该由操作员来调整。降低此阈值会减少分歧或不确定分类区中的观察次数。在第六节中详细讨论了调整这个阈值。

相互一致的分析可以有效识别分类器表现不佳的特定样本。在规避攻击的背景下，集合共识作为将新型攻击和弱模仿攻击与有效模仿攻击分开的标准。对于新型攻击，投票结果分布在50％左右是很常见的，这表明考察中的观察地图与训练集中的良性或恶意样本一致。由于这些攻击陷入了相对罕见的不确定范围，因此它们很容易被识别，并被视为弱逃避。强烈的模仿攻击是那些攻击投票的分布接近于良性观察的分布的攻击。因此，典型的新型攻击是通过相互协议分析来确定的，但强烈的模仿攻击是不可能的。由于不确定观测得到的训练集支持性很差，因此这些观测值对于增加训练集以提高分类器的准确性是最有效的。

表三。     PDF RATE Ø UTCOMES ˚F 或乙ENIGN d OCUMENTS ˚F ROM Ø PERATIONAL Ë 估价小号ET

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 良性 | | 恶毒 | |
| 分类 |  | 不确定 | |  |
| Contagio | 98076 | 1408 | 203 | 40 |
| 大学 | 99217 | 360 | 95 | 55 |

表IV。     PDF RATE Ø UTCOMES ˚F 或中号ALICIOUS d OCUMENTS ˚F ROM Ø PERATIONAL Ë 估价小号ET

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 良性 | | 恶毒 | |
| 分类 |  | 不确定 | |  |
| Contagio | 0 | 0 | 19 | 254 |
| 大学 | 0 | 0 | 0 | 273 |

在操作中，采用相互协议分析来防止逃避入侵检测系统。相互一致率是由检测发生时整体分类器提供的结果得到的。操作员可以通过多种方式使用集合分类器协议，包括调整投票阈值以防止误报或漏报，过滤观察值以进行隔离或更昂贵的分析，并优先处理警报。相互协议分析的优势在于它可以用来识别在逃跑企图时可能发生的入侵检测逃逸行为。

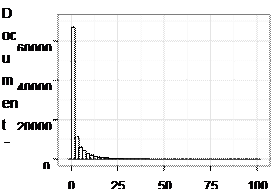
 五  Ë 估价

为了评估我们的方法，我们使用来自真实世界传感器的操作数据集以及从Mimicus和Reverse Mimicry攻击中获取的模仿攻击数据对PDFrate应用相互协议分析。我们研究相互一致性分析将分类器可靠性与分类器规避的观察结果分开的程度。我们还评估了使用Drebin Android恶意软件检测器检测新型恶意软件系列时相互协议分析的效用。

*A. PDFrate操作数据集*

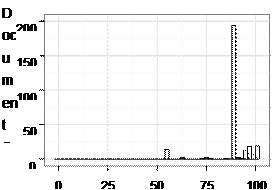
我们对从网络监视器处理的文件采取了相互协议分析，处理通过网络和电子邮件传输的文件。这个数据集包括110,000 PDF文件，我们随机分成两个数据集。操作评估集包含100,000个文档，操作培训集包含10,000个文档。在收集之后的几个月内通过扫描许多防病毒引擎来确定文档的基本事实。这些数据集分别包括273和24个恶意文件。表III和表IV显示了使用Contaigo和PDFrate的大学分类器的操作评估数据集的分数。图2和图3显示了该操作评估数据集的良性和恶意样本的PDFrate Contagio分类器分数的分布。

重要的是要注意，良性和恶意的例子的分数在各自分数范围的最远端很重，而分布很快就会下降。在典型的系统部署中，不确定范围内的观测数量非常少，大部分误分类属于不确定区域。因此，相互协议分析可以用来做出估计



PDFrate Contagio分类器分数

图2.      来自操作评估集的良性文档的分数



PDFrate Contagio分类器分数

图3.       来自运营评估集的恶意文件分数

至少在没有强大的逃避攻击的情况下，错误分类数量的上限。

不仅整体分类器相互协议分析对于识别分类器表现不佳的时间是有用的，而且对识别将提供最需要的支持以改进分类器的具体示例也是有效的。为了证明这一点，我们试图重复改进操作评估数据集中会发生的分类评分，因为将附加样本添加到分类器训练集中。我们从Contagio分类器开始，并添加了来自操作培训集的样本。

使用原来的Contagio训练数据集，我们确定了操作训练集中所有观察结果的评分。在操作环境中，所有高于不确定阈值（高于25分）的观察结果通常需要进一步调查，无论结果是不确定的还是恶意的。操作培训套件中有200个文件符合该标准。在这200个样本中，43个是假阳性，14个是使用传统阈值的假阴性。我们用正确的基本事实将这200个观察值添加到了Contagio训练集中，并创建了另一个分类器。

为了进行比较，我们还创建了具有不同大小随机选择的操作训练集子集的附加分类器，以模拟随机选择的Con-

表V.                                小号的CORES 乙ENIGN d OCUMENTS ˚F ROM ö PERATIONAL

Ë 估价小号ET ù SING Ç ONTAGIO Ç LASSIFIER 小号UPPLEMENTED W¯¯ ITH ö PERATIONAL Ť 汛期d ATA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 良性 | | 恶毒 | |
| 其他培训数据 | 训练集大小 |  | 不确定 | |  |
| 无（原Contagio） | 10000 | 98076 | 1408 | 203 | 40 |
| 随机子集2500 | 12500 | 99332 | 265 | 98 | 32 |
| 随机子集5000 | 15000 | 99444 | 200 | 71 | 12 |
| 随机子集7500 | 17500 | 99502 | 169 | 49 | 7 |
| 不确定和恶意 | 10200 | 99506 | 183 | 26 | 12 |
| 全面的训练分区 | 20000 | 99540 | 134 | 48 | 五 |

             表六。              小号作者CORES 中号ALICIOUS d OCUMENTS ˚F ROM

ø PERATIONAL Ë 估价 小号ET ù SING Ç ONTAGIO Ç LASSIFIER 小号UPPLEMENTED W¯¯ ITH ö PERATIONAL Ť 汛期d ATA

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | 良性 | | 恶毒 | |
| 其他培训数据 | 训练集大小 |  | 不确定 | |  |
| 无（原Contagio） | 10000 | 0 | 0 | 19 | 254 |
| 随机子集2500 | 12500 | 0 | 14 | 4 | 255 |
| 随机子集5000 | 15000 | 0 | 14 | 4 | 255 |
| 随机子集7500 | 17500 | 0 | 14 | 4 | 255 |
| 不确定和恶意 | 10200 | 0 | 14 | 7 | 252 |
| 全面的训练分区 | 20000 | 0 | 14 | 4 | 255 |

tagio分类器。表五和表六列出了适用于运行评估集的这些分类器的性能。

这些结果表明，分类器的局部调整对提高分类器的准确性有很大的影响。请注意，如恶意观察所发生的，在整个得分中点以错误方向移动几个样本不会被视为有害的，因为这些样本已经处于不确定范围内（非常接近50％的投票标记），如图所示3.对于前两种再培训策略，良性区域观测值（某些真实负值）的比例从98.3％上升到99.8％，甚至超过普遍优于大学分类的准确度（99.5％）。假阳性的相应下降很重要，因为它与不确定观察值的下降一致。在这种情况下，如果操作员响应所有不确定或恶意的观察结果，

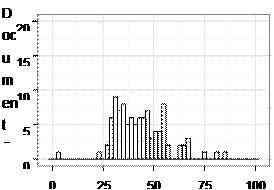
随机子集训练增加了直觉预期的结果。随着从训练集添加的随机样本的数量增加，分区评估数据的分类结果得到改善。将来自训练分区的高于不确定阈值的样本添加到结果中的分类器的准确度非常接近完整训练分区构建的分类器。因此，相互一致的分析对于识别分类器表现不佳的观察结果是有效的。由此可见，将这些样本添加到训练集中确实可以通过在靠近这些样本的区域提供支持来改善分类器。另一方面，增加相互高度一致的观测值可以提高分类器的效率。添加整个训练集和添加不确定样本的结果是相似的，但投入的努力是巨大的不同。获得基本事实的差异以及将10,000和200个观察值添加到训练集是不朽的。

*B. Mimicus*

为了证明相互协议分析在识别逃避检测的观察中的效用，我们转载

表VII。    PDF RATE ç ONTAGIO Ç LASSIFIER Ø UTCOMES ˚F 或中号IMICUS ê VASION 一个TTACKS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 良性 | | 恶毒 | |
| 脚本 |  | 不确定 | |  |
| 基线攻击 | 0 | 0 | 0 | 100 |
| F模仿 | 2 | 70 | 26 | 2 |
| FC模仿 | 7 | 78 | 15 | 0 |
| FT模仿 | 10 | 64 | 26 | 0 |
| FTC模仿 | 33 | 62 | 五 | 0 |
| F gdkde | 7 | 92 | 1 | 0 |
| FT gdkde | 4 | 95 | 0 | 1 |



PDFrate Contagio Classifer评分

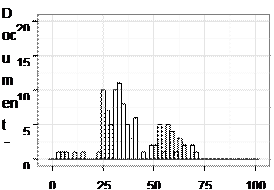
图4.      F模仿攻击的分数分布

Srndi c和Laskov [43]的工作，并对这些逃避尝试进行了相互同意的分析。我们使用Mimicus框架生成实施各种逃避攻击场景的PDF文档。我们使用了与Srndi c和Laskov出版物相同的数据集，并将生成的文档提交给pdfrate.com以获取分数。因为我们使用了相同的攻击数据，所以我们的结果仅限于每种攻击类型100个样本。我们能够取得与Mimicus研究中记录的结果非常接近的结果。

我们使用表七中各种攻击情景的相互协议来呈现分类结果。请注意，由于所有这些文档都是恶意的，正确的分类是恶意的。良性评级表示成功逃避。

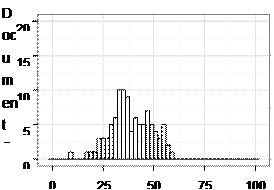
在图4至图7中显示了每个非GD-KDE情景下文档的PDFrate投票分数分布情况.DG-KDE攻击将在第VII节具体阐述。这些攻击的投票分数分布在很大程度上与典型的良性或恶意观察中看到的不相符。使用基于集合分类器多样性的方法，大多数这些攻击可以从良性观察中分离出来。因此，这些攻击应该被视为虚弱的模仿尝试。

当分类器的所有属性都已知时，33％的攻击是有效的。但是，当分类器或训练集的细节被扣留时，攻击成功率会降至10％或更低。除了对Contagio数据集进行评估之外，还针对使用大学数据集进行培训的分类器对模仿攻击数据进行了测试。这会导致备用FC攻击情形，因为攻击者不知道该训练集。图8显示了从FTC攻击应用恶意软件得分的分布情况，



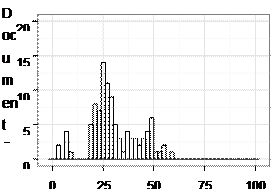
PDFrate Contagio Classifer评分

图5.       FT模仿攻击的分数分布



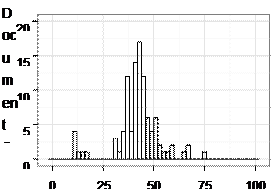
PDFrate Contagio Classifer评分

图6.      FC模仿攻击的分数分布



PDFrate Contagio Classifer Score 图7. FTC模仿攻击的得分分布

Nario将Contagio分类器与大学分类器进行对比。两个分类器的结果非常相似。在这两种情况下，100次逃避尝试中只有7次被归类为良性。仔细比较图6和图8得出的结论是，大学分类器在分歧区域中心附近提供了更紧密的分数聚类。Contagio分类器的结果与大学分类器的结果相似，因为Mimicus规避尝试将Contagio数据用于基线良性数据和攻击数据。



PDFrate大学分级分数

图8.        使用大学分类器进行FC模仿攻击的分数

表VIII。                            PDF格式RATE O TOMOS F 或R EVERSE M IMICRY A TTACKS

Contagio分类器

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 良性 | | 恶毒 | |
| 脚本 |  | 不确定 | |  |
| EXEembed | 77 | 22 | 1 | 0 |
| PDFembed | 93 | 7 | 0 | 0 |
| JSinject | 三十 | 67 | 3 | 0 |

大学分类

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 良性 | | 恶毒 | |
| 脚本 |  | 不确定 | |  |
| EXEembed | 0 | 4 | 16 | 80 |
| PDFembed | 81 | 19 | 0 | 0 |
| JSinject | 0 | 22 | 55 | 23 |

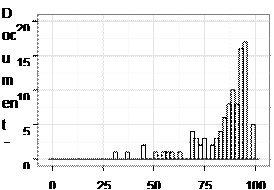
当使用相互协议时，大多数模仿攻击被标记为不确定，表明已知的分类器失败和可能的逃避。在最好的模拟攻击场景中，PDFrate的所有属性都是已知的，只有33％的模仿尝试被成功分类为良性。如果攻击者不知道分类器的某些细节（如确切的训练集），则模仿成功率低于10％。

*C.反向模仿*

我们还对Maiorca等人提出的Reverse Mimicry攻击进行了相互协议分析。[26]，[27]。没有公开记录复制这些攻击所需的确切程序。但是，Maiorca等人 为我们提供了他们学习中使用的文件。他们最近的攻击涉及每个逃避场景中的500个文件。为了与Mimicus攻击评估保持一致，我们为每个方案抽取了100个样本随机子集进行评估。

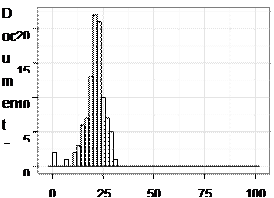
在表八中，我们展示了对反向模仿对Contago和University分类器的攻击进行双方同意分析的结果。这些针对大学分类器的攻击的分数分布如图9,10和11所示。尽管进行了相互协议分析，但67％的反向模仿攻击是对Contago分类器的成功规避（认为是良性）。

大学分类器比Contagio分类器的性能好得多。针对大学分类器的唯一方法是通过PDFembed攻击实现的。这种攻击非常成功，因为嵌入了完整的恶意PDF



PDFrate大学分级分数

图9.      EXEembed攻击的分数分布

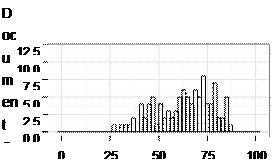


PDFrate大学分类器分数图10. PDFembed攻击的分数分布

在另一个良性文件中。这个嵌入式文档驻留在压缩数据流中，这意味着PDFrate的特征提取器无法观察到结构特征。这与其他情景EXEembed和JSinject相反，尽管尽量减少这种情况，但一些渎职指标仍然暴露。

PDFembed方案对pdfrate.com上的检测器有效，因为它不会执行递归解码和分析，这在操作系统中是必要的。这种故障类似于假设输入解压缩的可执行文件并在提供打包的可执行文件或特洛伊木马文件时失败的恶意软件分析系统。当在操作检测系统中部署PDFrate时，它通常在同时提供PDF流解码和从其他容器（例如电子邮件或zip文件）中提取PDF的框架内完成[3]。在所有的PDFembed攻击中，嵌入式文档都是相同的。Contagio和University分类器一旦提取出来就可以很容易地检测到该文档，分别返回97.6％和100％的分数。

对于单独的PDFrate实现，PDFembed场景代表了一个强大的逃避场景，其中分类器自省提供的好处很小，因为特征提取器被很好地避开了。即使Contagio基于



PDFrate大学分级分数

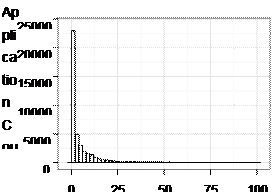
图11.     JSinject攻击的分数分布

分类器不适合EXEembed和JSinject中使用的恶意软件，但这些样本中的很多仍落在不确定的结果投票范围内。当使用更强大的大学分类器时，相互协议分析标志着这些可能会成功的逃避方案。

*D. Drebin*

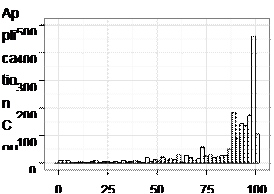
为了对Drebin Android恶意软件检测器进行相互协议分析，我们构建了一个集成分类器。我们采用了随机森林分类器，该分类器需要调整特征以确保计算效率并确保与原始线性SVM相当的结果。我们不使用所有字符串值作为要素，而是使用包含最耐用功能的891个功能子集。我们将所有功能用于受限类别，例如API调用和权限。对于任意命名的属性，如组件和意图，我们利用最多产的值，选择训练集中发生超过100次的值。最后，我们忽略了诸如URL和网络地址等高度易变项目的特定值，它们构成了Drebin使用的一半以上功能。最后，我们总结了每个功能类别的出现次数，并将这些次数用作功能。作为优化，我们在分类器训练期间（而不是在评估期间）去重复任何等价的特征向量。这种使用我们狭义特征集的重复数据删除从123,453个减少到63,379个独特的良性样本，从5,560个减少到2,185个独特的恶意样本。除了这些数据转换之外，我们在评估中使用了发布的Drebin数据集，包括数据集分区。

我们调整了基于随机森林的分类器，以提供与Drebin的线性SVM分类器相媲美的分类性能。我们调整的主要项目是训练每棵树时使用的良性与恶意样本的比率。这是必要的，因为各种训练集的良性与恶性比例极端不平衡。我们将个体树训练的比率调整为2.5良性至1恶意，以便匹配Arp等人选择的1％的期望假阳性率。我们将随机森林的其他可调参数设置为标准值：每个随机森林包含1000棵树，每个分割处尝试的变量数设置为要素数量的平方根。我们的随机森林分类器的平均误报率为1.06％，恶意软件检测率为92。使用没有不确定区域的传统阈值的已发布数据集分区上的3％。基于随机森林的分类器性能非常相似，尽管稍逊于Drebin线性SVM提供的性能。如图12所示



Drebin随机森林分数

图12.     良性样品的分数分布



Drebin随机森林分数

图13.     恶意样本的分数分布

表IX。     d 热病 ř ANDOM ˚F 奥列斯特Ç LASSIFIER ö UTCOMES AS 中号UTUAL 甲GREEMENT Ť HRESHOLD IS 甲DJUSTED

良性样品

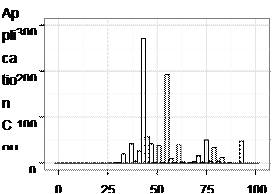
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 良性（％） | | 恶意（％） | |
| 相互协议阈值（％） |  | 不确定 | |  |
| 三十 | 97.46 | 1.49 | 0.54 | 0.52 |
| 40 | 96.49 | 2.45 | 0.63 | 0.43 |
| 50 | 95.12 | 3.82 | 0.71 | 0.35 |

恶意样本

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 三十 | 4.44 | 3.27 | 5.44 | 86.85 |
| 40 | 3.77 | 3.93 | 7.30 | 84.99 |
| 50 | 3.16 | 4.56 | 10.34 | 81.95 |

使用公布的数据集分区之一的良性样本的分数分布。图13显示了恶意样本的情况。正如预期的那样，得分分布的形状与PDFrate的形状相似，但由于分类器准确性较低，因此样本分布距离得分连续区的各个末端较远。表IX显示了典型相互协议阈值的分类器结果。

原始Drebin研究的一个重要方面是通过家庭对恶意软件进行分类，并对先前未知的恶意软件家族的分类器进行评估。这是通过扣留要从训练集中评估的家庭，然后将所得分类器应用于该家族中的恶意软件样本来实现的。这是由Arp等人注意到的。[4] Drebin先前提供的分类较差



Drebin随机森林分数

图14.      未知家庭的分数分布A

未知的恶意软件。我们将我们的基于随机森林的分类器和不确定的分数区域应用于相同的问题。图15比较了使用相互协议分析的线性SVM分类器和基于随机森林的分类器的检测率。

正如预期的那样，绝大多数未知恶意软件家族的评分分布均为弱逃避攻击，表明分类器认为这些观察既不类似于训练集中看到的良性或恶意样本。例如，图14显示了恶意软件家族A的得分。平均而言，每个家庭的75.2％被标记为不确定，另有8.2％被使用我们的随机森林分类器标记为恶意，而每个家庭的50.6％被Drebin线性SVM标记为恶意。家庭Q和R代表强烈的逃避行为。Arp等人 请注意，使用Drebin使用的功能集无法可靠地检测到R族。虽然Drebin使用的特征足以用于包含在训练集中的Family Q的检测，它与Drebin的特征空间中的其他家族完全不同，被标记为逃避。另一方面，家庭P与Drebin的特征空间中的其他恶意软件家族非常相似，因此不需要在训练集中拥有该家族的样本。除去这3个家庭，其余17个家庭中平均89.7％的样本被随机森林分类器识别为恶意或不确定，而线性SVM分类器检测到53.2％。将这些先前未知的样本标记为不确定应该被认为是有利的，以便操作员可以采取行动来改善分类器。尽管线性SVM分类器在这些场景中提供了抛硬币的平均分类准确度，

相互协议分析有效识别由新型攻击和有针对性的模仿所导致的PDFrate和Drebin恶意软件检测系统中的可能逃避。

          VI。              中号UTUAL 甲GREEMENT Ť HRESHOLD Ť UNING

对于我们的大多数评估，我们使用了50％的相互协议阈值，该阈值将分类器投票分数区域分成四个相等大小的象限。可以选择任意的相互协议阈值。表IX包含了Drebin对三个相互协议阈值水平的预测。

|  |
| --- |
| F:\PDFpy\God_with_me\2018Q2\paper\WhenATreeFallsUsingDiversityInEnsembleClassifiers_NDSS2016\paper.files\image016.gif  图15.     以前未知的（从训练集中排除）恶意软件家族的检测率比较 |

在表X中，我们介绍了适用于操作评估数据集的PDFrate大学分类器结果以及整个双方协议范围内的FC Mimicus攻击。

所选择的确切相互协议阈值在分类失败检测的改进和作为不确定性而抛出的分类器预测数量之间达成了平衡。希望获得较低不确定结果的运营商可以选择较低的阈值。以表X中的PDF率表现为例，如果选择30％作为阈值，则不确定区域包括35％至65％的集合分类器投票分数，而不是25％和75％以及50％阈值。对于运行数据集，良性样本的不确定率从0.456％下降到0.256％。然而，成功逃避企图的数量从7％上升到了12％。该阈值的最佳设置取决于操作员的偏好。通过调整相互协商阈值来调整不确定检测的灵敏度，

                   七。              GD-KDE 和E NSEMBLE SVM S

相互协议分析应适用于所有能够在单个分类器中提供足够多样性的整体分类器。为了验证这一点，我们研究了通过使用集合方法对SVMs进行相互协议分析来打击对SVM的规避的可行性。

Mimicus攻击框架利用SVM分类器对其PDFrate副本实施梯度下降和核密度估计（GD-KDE）攻击。这种攻击通过利用可微分分类器的已知决策边界进行[8]。

我们复制了Mimicus的GD-KDE回避攻击，并确认它们确实非常有效。使用依赖于libSVM [10]的R [2]的e1071包，我们计算了GD-KDE情景下8.9％恶意的平均概率（或91.1％良性），这些攻击正好在逃避区域内。Srndi c和Laskov使用'

表X.       PDF RATE ü 。高校Ç LASSIFIER P ERFORMANCE AS 中号UTUAL 甲GREEMENT Ť HRESHOLD IS 甲DJUSTED

良性运营评估

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 真正 | 假 |  |
| 相互 | 不确定 | 负 | 正 | 不确定 |
| 协议 | 得分了 | 率 | 率 | 率 |
| 阈 | 范围 | （TNR） | （FPR） | （UR） |
| 0％ | - | 99.8％ | 0.150％ | 0％ |
| 10％ | （45,55） | 99.8％ | 0.128％ | 0.0592％ |
| 20％ | （40,60） | 99.7％ | 0.103％ | 0.147％ |
| 30％ | （35,65） | 99.7％ | 0.0832％ | 0.256％ |
| 40％ | （30,70） | 99.6％ | 0.0712％ | 0.342％ |
| 50％ | （25,75） | 99.5％ | 0.0552％ | 0.456％ |
| 60％ | （20,80） | 99.3％ | 0.0331％ | 0.618％ |
| 70％ | （15,85） | 99.1％ | 0.0291％ | 0.825％ |
| 80％ | （10,90） | 98.7％ | 0.0261％ | 1.27％ |
| 90％ | （5,95） | 97.0％ | 0.0120％ | 3.01％ |
| 100％ | （0100） | 53.6％ | 0％ | 46.4％ |

恶意操作评估

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 阈 | 范围 | FNR | TPR | UR |
| 0％ | - | 0％ | 100％ | 0％ |
| 10％ | （45,55） | 0％ | 100％ | 0％ |
| 20％ | （40,60） | 0％ | 100％ | 0％ |
| 30％ | （35,65） | 0％ | 100％ | 0％ |
| 40％ | （30,70） | 0％ | 100％ | 0％ |
| 50％ | （25,75） | 0％ | 100％ | 0％ |
| 60％ | （20,80） | 0％ | 99.6％ | 0.366％ |
| 70％ | （15,85） | 0％ | 99.6％ | 0.366％ |
| 80％ | （10,90） | 0％ | 99.6％ | 0.366％ |
| 90％ | （5,95） | 0％ | 99.6％ | 0.366％ |
| 100％ | （0100） | 0％ | 95.6％ | 4.40％ |

Mimicus FC攻击

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 阈 | 范围 | FNR | TPR | UR |
| 0％ | - | 84％ | 16％ | 0％ |
| 10％ | （45,55） | 69％ | 8％ | 23％ |
| 20％ | （40,60） | 31％ | 4％ | 65％ |
| 30％ | （35,65） | 12％ | 4％ | 84％ |
| 40％ | （30,70） | 7％ | 1％ | 92％ |
| 50％ | （25,75） | 7％ | 1％ | 92％ |
| 60％ | （20,80） | 7％ | 0％ | 93％ |
| 70％ | （15,85） | 6％ | 0％ | 94％ |
| 80％ | （10,90） | 0％ | 0％ | 100％ |
| 90％ | （5,95） | 0％ | 0％ | 100％ |
| 100％ | （0100） | 0％ | 0％ | 100％ |

表十一。     Ñ 作者棕土d OCUMENTS PER GD-KDE甲TTACK W¯¯ 该处Ë NSEMBLE SVMÇ LASSIFIER P ROVIDES Ç ORRECT P REDICTION AS P 的钮用来˚F 功能设定 ü SED IS V ARIED 。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 特征子集 | |  |
| 攻击 | 5％ | 7.5％ | 10％ | 12.5％ |
| 基线恶意 | 100 | 99 | 98 | 98 |
| 基线良性 | 2 | 41 | 93 | 94 |
| F gdkde | 100 | 100 | 99 | 五 |
| FT gdkde | 99 | 100 | 92 | 1 |

表十二。    Ñ 作者棕土d OCUMENTS PER GD-KDE甲TTACK W¯¯ 该处Ë NSEMBLE SVMÇ LASSIFIER P ROVIDES Ç ORRECT P REDICTION AS P 的钮用来Ť 汛期 d ATA ü SED IS V ARIED 。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 训练数据子集 | | | |
| 攻击 | 12.5％ | 25％ | 50％ | 100％ |
| 基线恶意 | 86 | 87 | 92 | 98 |
| 基线良性 | 100 | 100 | 100 | 100 |
| F gdkde | 0 | 0 | 0 | 0 |
| FT gdkde | 0 | 0 | 0 | 0 |

从SVM实施的SVM决策边界缩放距离以提供类似的结果。GD-KDE攻击表明，不能依赖对SVM等单个分类器的反省来检测规避行为。

虽然对支持向量机分类器有效，但使用GD-KDE攻击的PDFrate随机森林分类器的结果与传统的同类产品（见表VII）大致相当。对随机森林进行类似的攻击是不实际的，因为随机森林具有极其复杂和随机的决策边界。

我们试图确定在多大程度上可以使SVM分类器通过多样性启用内省来识别可能的逃避。我们使用100个独立的SVM分类器实现了一个简单的基于SVM的集成分类器，分数是各个分类器投票的简单总和。为了确定对建筑物多样性重要的属性，我们改变了用于构建每个单独的SVM的特征子集和训练数据。我们进行了全面的网格搜索。表Ⅺ列出了最显着的结果，表Ⅺ列出了使用完整训练数据集的特征套袋，表II列出了使用全套特征集对训练数据进行套袋的表II。这些表格展示了匹配正确结果（恶意或不确定）的分类器结果的部分。

似乎收集训练数据并不是特别重要，因为在建立集成分类器时，相互协议分析是有用的。令我们惊讶的是，我们没有发现任何情况，除了完整的训练集提供了最好的结果。

然而，特征装袋对构建分类器至关重要，因为相互协议分析能够识别不确定的预测。对SVM分类器的特征进行装箱提供了推断中必要的多样性，这使得相互协商分析有意义。似乎基于完整特征集的子集的个体分类器比使用所有特征的单个分类器难以共同规避。虽然可以通过成功模仿一部分功能来避开单个分类器，但似乎基于少量功能的多个分类器的组合需要在整个功能集中进行更完整的模仿。特征套袋应用于许多人

表十三。   PDF格式速率 SVM E NSEMBLE C 分类器 O UTCOME F 或GD-KDE E VASION A TTACKS 。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 良性 | | 恶毒 | |
| 攻击 |  | 不确定 | |  |
| 基线恶意 | 0 | 0 | 2 | 98 |
| 基线良性 | 93 | 7 | 0 | 0 |
| F gdkde | 3 | 97 | 0 | 0 |
| FT gdkde | 8 | 91 | 1 | 0 |

独立的SVM使得GD-KDE风格的攻击不可行，因为不再有单一的可预测的决策边界来进行攻击。

结果还表明，当使用基于SVM的合奏时，仔细调整用于装袋的特征的部分是至关重要的。在通过在每个分类器中使用更少的特征来正确分类恶意观察（包括逃避尝试）的能力和通过使用更多特征的良性观察之间似乎存在权衡。使用更少的特征会导致更复杂的分类器分割更小，而更多特征更接近标准SVM，其具有单个超平面分割器。这些结果表明PDFrate中使用的特征为良性样品提供了更好的外推。看起来恶意样本在PDFrate的特征中有较高的变化，需要更多类似的训练样本才能成功分类。

表XIII显示了应用于Mimicus GD-KDE攻击和基线良性和恶意样本的SVM集成分类器的结果。结果表明，虽然逃避尝试成功地将分数从恶意范围中删除，但绝大多数逃避尝试都落在了不确定的范围内。在最佳情况下，只有8％的逃避尝试完全成功，而只有4.5％的已知数据位于不确定区域。这些结果与使用PDFrate的随​​机森林分类器获得的结果相当，其中GD-KDE攻击是不可能的。

因此，相互协议分析不仅适用于随机森林，而且似乎适用于所有具有充分多样性的乐团。对功能进行套袋看起来是该功能的核心。

                   八。             d 藏密坛城艺术和˚F UTURE W¯¯ ORK

集成分类器中的相互协议分析提供了分类器预测准确的置信度估计，而无需外部验证。许多分类器可以提供分数连续，例如与SVM中使用的决策边界的距离，但这些度量在面对模仿攻击时不准确。此外，传统的信心度量不适用于与已知基础事实的人口相背离的数据。

相互一致反映了分类器的内部一致性。假设足够的特征强度，这种内部一致性是分类器置信度的代表。针对PDFrate的攻击表明，模仿抗性特征对识别新型攻击至关重要。针对PDFrate的唯一强大规避攻击是成功的，因为它通过在另一个中嵌入恶意PDF完全回避了PDFrate的功能，使得特征提取器无法看到恶意PDF。其他攻击虽然试图欺骗特征提取器，但是还不够充分，因为有些特征仍然有效。如果特征提取器抗篡改并且特征是渎职的适当指标，则新的攻击将被检测为恶意或者被评定为不确定。然而，如果特征集（或特征提取机制）很弱，那么规避仍然是可能的。运营商必须保持警惕，以防止在恶意软件检测的特征提取阶段进行逃避。

在使用基本SVM分类器构建集合时，我们发现特征装袋对于生成必要的多样性至关重要，以使相互一致的测量有意义。不合格，装袋是指利用随机子集的训练数据。这种方法广泛用于机器学习技术。在我们的研究中，收集培训数据并未显示出对相互协议分析的重要性。这可能是由于该训练集缺乏多样性。进一步的研究可能表明，在什么条件下训练数据袋装提供了多样性，有助于促进相互协议分析 我们还观察到，调整SVM集合中使用的特征部分非常重要。我们观察到没有类似的需要调整基于随机森林的分类器的参数，但是这应该是未来研究的一个领域。每个节点尝试的特征数量（mtry）和树的深度可能会影响随机森林中的有用多样性。看起来很多特征，即使它们相互依存或分类价值低，都有助于逃避更加困难。

如果特征很强，那么训练集的相关性将决定个体观察的相互协议评级。如果测试观察结果与训练集中的样本类似，那么将发生高确定性预测。与特征空间中的训练集不同的测试观察值将给出分类器认为不确定的分类。在我们评估的一些情况下，训练集的质量对于发现逃避攻击显示出重要性。例如，高级PDFrate大学分类器比反向模仿攻击的Contagio分类器的回避少得多。对于Drebin评估，由于功能较弱，R族表示强回避，但如果包含在训练集中，则可以检测到家族Q. 因此，相互协议分析的有效性也取决于训练集中的充分覆盖。但是，训练集的有效性直接取决于特征的强度。弱特征集将需要比更紧密地模拟基本恶意属性的特征集更广泛的训练集。运营商应确保用于恶意软件检测的功能不仅能抵御欺骗，而且基于恶意软件造成的伪影，而不仅仅是与目前的攻击巧合。弱特征集将需要比更紧密地模拟基本恶意属性的特征集更广泛的训练集。运营商应确保用于恶意软件检测的功能不仅能抵御欺骗，而且基于恶意软件造成的伪影，而不仅仅是与目前的攻击巧合。弱特征集将需要比更紧密地模拟基本恶意属性的特征集更广泛的训练集。运营商应确保用于恶意软件检测的功能不仅能抵御欺骗，而且基于恶意软件造成的伪影，而不仅仅是与目前的攻击巧合。

如VB部分所示，保持分类器的训练集的秘密有助于提高针对有目标的逃避尝试的弹性。操作系统隐藏从分类器返回的确切分数可能是明智的，因为这些分数可帮助攻击者知道他们的变化是否会造成伤害或帮助他们逃避企图。这些信息可能削弱秘密训练集[18]提供的好处。

Mimicus的GD-KDE攻击表明，某些分类器可以使基于机器学习的检测器容易受到规避攻击。随机生成的集合分类器并未显示出容易受到类似攻击的影响，但可能会找到新的方法。测量集合分类器中相互一致性的能力几乎没有成本，但是提供了对实际分类器规避的检测。此功能是在分类器规避问题（如恶意软件检测器）的情况下使用集合的强有力原因。如果使用相互协议来优化分类器训练，则与使用随机选择相比，攻击者可以对训练集增加更多的知识。但是，这种知识如何被攻击者利用并不明显。

一些人主张使用简单的单片分类器，因为结果被认为更容易解释。例如，Drebin识别有助于分类的功能的能力受到称赞。然而，不清楚这些信息是否对最终用户真正有用。用户已经有机会查看权限并在提示时经常选择错误。由于URL和API调用可以进行社交设计，用户通常不会意识到这些元素，因此将这些项目作为上下文提供给用户不太可能帮助他们做出正确的决定。对于安全专业人员而言，集成分类器可提供帮助分析的机制，例如与现有已知恶意或良性样本的相似性。最重要的是，该功能集对训练有素的分析师会有帮助。

相互协议分析为操作员提供了对分类器准确性和对警报响应优先级的能力的更大信心。一些运营商将仅使用集成分类器内省来调整投票阈值。试图避免漏报（逃避攻击）的环境将使用较低的阈值并增加误报数量。另一方面，某些环境可能会使用高于正常的投票阈值来实现低误报率，但可能会导致更高的漏报率（例如防病毒引擎所实现的漏报率）。

当不确定观测值经过集中分析时，运营商从相互协议分析中获得最大收益。这些样本必须经过不同的补充分析或检测。由于表现良好的分类器的不确定观测数量较少，因此第二种观点可能相对昂贵，可能是手动驱动或涉及动态分析。基于集合多样性的信心估计对于希望识别新型攻击来执行附加分析的组织非常有用。尽管机器学习领域可能并不常规，但对于安全领域而言，增加不确定的结果是很直观的，许多系统只提供对已知观察的判断，无论是良性的还是恶意的。例如，SPAM过滤器对不能可靠分类的样品使用隔离是很常见的。很多时候，高保真警报优于每次观察的响应。

相互一致性分析对于识别那些与训练集中已知样本不相似的样本非常有效。由于将不确定样本添加到分类器中可显着提高分类器的准确性，因此对不确定观测值的分析可能会激励操作员而不是使敏感度降低。运营商有权以比随机添加到训练集更有效的方式改进分类器。

通过相互协议分析的应用可以提高基于机器学习的检测器的评估。一个简洁的度量指标是不确定率，或者一个分类器不适合提供预测的观测部分。分类器评估使用协议评分分布和方差的有效性可能是未来研究的主题。图12和图13中显示的分类器得分分布似乎表明回归可用于预测成功逃避的数量。然而，这种类型分析的困难在于，当没有提供外部地面真相时，将良性和恶意数据的弧线分开。

最重要的是，监测集合分类器中的相互协议，为以前看不见的攻击和有针对性的模仿攻击提供了回避的条件。当前的回避攻击已经引起了对基于学习的探测器的弹性的质疑，但它们表现出比以前假设的要弱。仅仅混淆攻击，使它们不再出现为已知的攻击是不够的。成功的模仿必须非常密切地反映良性样本。当然，未来研究相互协议分析可以提高攻击质量的程度似乎是值得的。

                                      IX。        ç 面上污染物

我们引入了一种新技术来检测恶意软件分类器的性能降低对个人观察。相互协议分析依赖于集成分类器的多样性来产生分类器置信度的估计。我们使用超过100,000个PDF文档和100,000个Android应用程序来评估我们的方法。将PDFrate应用于从真实网络中获取的文档，我们发现不确定结果的数量很小 - 仅为0.2％。如果排除这些不确定的例子，真正的阳性率从95％上升到100％，假阳性率从0.053％下降到0.0050％。此外，相互一致的分析对于识别要添加到训练集中的样本是有效的，导致分类器性能比随机抽样快得多的改进。在针对PDFrate的直接逃避攻击和针对Drebin的新型攻击中，大多数观察结果被指定为不确定的结果，通知操作员检测器失败。虽然躲避攻击仍然是可能的，但它们需要在整个功能集中进行更完整的模仿。

我们相信，协同分析一般可以应用于整体分类器。我们发现特征装袋对基于多样性的逃避检测非常重要。GD-KDE攻击在支持向量机方面取得了巨大成功，可以通过SVM集成来挫败。集成分类器相互协议分析提供了一种评估基于机器学习的检测器的准确性的关键机制，而无需使用外部验证。

一个CKNOWLEDGMENTS

作者要感谢所有评论者提出的宝贵意见和建议。这项工作得到了国家科学基金会批准号CNS 1421747和II-NEW 1205453的支持。本资料中表达的意见，发现，结论和建议均为作者的观点，并不一定反映NSF或美国政府的观点。

R EFERENCES

[1]        “Mimicus”，http：//github.com/srndic/mimicus。

[2]        “R项目”，http：//www.r-project.org/。

[3]        M. Arnao，C. Smutz，A. Zollman，A. Richardson和E. Hutchins，“Laika BOSS：可扩展的以文件为中心的恶意软件分析和入侵检测系统”https://github.com/lmco/ laikaboss，2015年7月。

[4]        D. Arp，M. Spreitzenbarth，M. Hubner，H. Gascon和K. Rieck，“第二*十一届年度网络和分布式系统安全研讨会（* Drebin：有效和可解释的检测你的口袋里的android恶意软件” *NDSS）*，2014年2月。

[5]        D. Barbara，C. Domeniconi和JP Rogers，“Detecting outliers using transduction and statistical testing，” *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*，ser。KDD '06。纽约，纽约州，美国：ACM，2006年，第55-64页。

[6]        M. Barreno，B. Nelson，R. Sears，AD Joseph和JD Tygar，“机器学习是安全的吗？” *，在2006年ACM信息，计算机和通信安全研讨会会刊上发表*。ASIACCS '06。纽约，纽约州，美国：ACM，2006年，第16-25页。

[7]        U. Bayer，PM Comparetti，C. Hlauschek，C. Kruegel和E. Kirda，“ *网络和分布式系统安全研讨会（NDSS）* 2009,2009”中的“可扩展，基于行为的恶意软件集群” 。

[8]        B. Biggio，一电晕，D. Maiorca，B.尼尔森，N. Srndic，P. Laskov，G.扎琴多，和F Roli，“对机器学习躲避攻击在测试时，”在*机器学习和数据库中的知识发现*，ser。计算机科学讲义，H. Blockeel，K. Kersting，S. Nijssen和F. Zelezn y，Eds.' Springer Berlin Heidelberg，2013，no。8190，第387-402页。

[9]        B.Biggio和P.Laskov在*2012年国际机器学习会议（ICML）上发表的题为* “中毒支持向量机攻击”的*文章*。

[10]     C.-C. Chang和C.-J. 林，“LIBSVM：支持向量库

机器“，http://www.csie.ntu.edu.tw/~cjlin/libsvm/，2011年5月。

[11]     D. Chinavle，P.科拉里，T奥茨和T. Finin，在“控辩式分类为垃圾邮件，合奏” *上的信息18 ACM会议论文集和知识管理*，SER。CIKM'09。纽约，纽约州，美国：ACM，2009年，第2015-2018页。

[12]     M. Cova，C. Kruegel和G. Vigna，“检测和分析驱动下载攻击和恶意JavaScript代码” *，第19届国际万维网会议文集*，系列丛书。WWW '10。纽约，纽约州，美国：ACM，2010，第281-290页。

[13]     G. Cretu，A. Stavrou，M. Locasto，S. Stolfo和A. Keromytis，“铸造恶魔：消除异常传感器的训练数据”，*安全和隐私，2008* SP。IEEE *Symposium on*，2008年，第81-95页。

[14]     M. Egele，G. Stringhini，C. Kruegel和G. Vigna，“COMPA：Detecting Compromised Accounts on Social Networks。”，*NDSS*，2013,2013。

[15]     P. Fogla和W. Lee，“Evading network anomaly detection systems：formal reasoning and practical techniques，” *Proceedings of the 13th ACM conference on Computer and communications security*。亚历山大，弗吉尼亚州，美国：ACM，2006年，第59-68页。

[16]     M. Handley，V. Paxson和C. Kreibich，“网络入侵检测：逃避，流量标准化和端到端协议语义”，*2001年USENIX安全研讨会*，2001年，第115-131页。

[17]     M. Heiderich，M. Niemietz，F.Schuster，T.Holz和J.Schwenk，“Scriptless Attacks：Stealing the Pie Without Touching Sill，” *Proceedings of the 2012 ACM Conference on Computer and Communications Security*， SER。CCS '12。纽约，纽约州，美国：ACM，2012年，第760-771页。

[18]     L. Huang，AD Joseph，B. Nelson，BI Rubinstein和JD Tygar，“Adversarial machine learning”，*Proceedings of the 4th ACM workshop on Security and artificial intelligence*，ser。AISec '11。纽约，纽约州，美国：ACM，2011年，第43-58页。

[19]     S. Jana和V. Shmatikov，*2012年5月IEEE安全与隐私专题讨论会（SP）*，2012年5月，第80-94页，“滥用恶意软件检测器中的文件处理以获得乐趣和利润” 。

[20]     J. Jang, D. Brumley, and S. Venkataraman, “BitShred: feature hashing malware for scalable triage and semantic analysis,” in *Proceedings of the 18th ACM conference on Computer and communications security*, ser. CCS ’11. New York, NY, USA: ACM, 2011, pp. 309–320.

[21]     G. Kakavelakis, R. Beverly, and J. Young, “Auto-learning of SMTP TCP Transport-Layer Features for Spam and Abusive Message Detection,” in *LISA 2011, 25th Large Installation System Administration Conference*. Boston, MA, USA: USENIX Association, Dec. 2011.

[22]     L. I. Kuncheva and C. J. Whitaker, “Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy,” in *Machine Learning*, vol. 51, May 2003, pp. 181–207.

[23]     P. Laskov and R. Lippmann, “Machine learning in adversarial environments,” in *Machine Learning*, vol. 81, Aug. 2010, pp. 115–119.

[24]     P. Laskov and N. Srndic, “Static detection of malicious JavaScriptbearing PDF documents,” in *Proceedings of the 27th Annual Computer Security Applications Conference*, ser. ACSAC ’11. New York, NY, USA: ACM, 2011, pp. 373–382.

[25]     W.-J. Li, S. Stolfo, A. Stavrou, E. Androulaki, and A. D. Keromytis, “A Study of Malcode-Bearing Documents,” in *Detection of Intrusions and Malware, and Vulnerability Assessment 2007*, B. Hammerli and¨ R. Sommer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4579, pp. 231–250.

[26]     D. Maiorca, D. Ariu, I. Corona, and G. Giacinto, “A Structural and Content-Based Approach for a Precise and Robust Detection of Malicious PDF Files,” in *Proceedings of the 1st International Conference on Information Systems Security and Privacy*. ScitePress Digital Library, 2015, pp. 27–36.

[27]     D. Maiorca, I. Corona, and G. Giacinto, “Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious PDF files detection,” in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, ser. ASIA CCS ’13. New York, NY, USA: ACM, 2013, pp. 119–130.

[28]     J. Mason, S. Small, F. Monrose, and G. MacManus, “English Shellcode,” in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, ser. CCS ’09. New York, NY, USA: ACM, 2009, pp. 524–533.

[29]     E. Menahem, A. Shabtai, L. Rokach, and Y. Elovici, “Improving malware detection by applying multi-inducer ensemble,” in *Computational Statistics & Data Analysis*, vol. 53, Feb. 2009, pp. 1483–1494.

[30]     B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. P. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia, “Exploiting machine learning to subvert your spam filter,” in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats 2008*. Berkeley, CA, USA: USENIX Association, 2008, pp. 7:1–7:9.

[31]     J. Neyman, “Outline of a Theory of Statistical Estimation Based on the Classical Theory of Probability,” *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 236, no. 767, pp. 333–380, 1937.

[32]     M. Parkour, “11,355+ Malicious documents archive for signature testing and research,”

http://contagiodump.blogspot.com/2010/08/malicious-documentsarchive-for.html, Apr. 2011.

[33]     R. Perdisci, D. Dagon, W. Lee, P. Fogla, and M. Sharif, “Misleading worm signature generators using deliberate noise injection,” in *Security and Privacy, 2006 IEEE Symposium on*, 2006, pp. 15 pp.–31.

[34]     M. A. Rajab, L. Ballard, N. Lutz, P. Mavrommatis, and N. Provos, “CAMP: Content-Agnostic Malware Protection.” in *NDSS 2013*, 2013.

[35]     M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, “A Bayesian Approach to Filtering Junk E-Mail,” in *AAAI 98 Workshop on Text Categorization*, Jul. 1998.

[36]     J. Schlumberger, C. Kruegel, and G. Vigna, “Jarhead analysis and detection of malicious Java applets,” in *Proceedings of the 28th Annual Computer Security Applications Conference*, ser. ACSAC ’12. New York, NY, USA: ACM, 2012, pp. 249–257.

[37]     H. Shacham, “The Geometry of Innocent Flesh on the Bone: Returninto-libc Without Function Calls (on the x86),” in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS ’07. New York, NY, USA: ACM, 2007, pp. 552–561.

[38]     G. Shafer and V. Vovk, “A Tutorial on Conformal Prediction,” *Journal of Machine Learning Research*, vol. 9, pp. 371–421, Jun. 2008.

[39]     M. Z. Shafiq, S. A. Khayam, and M. Farooq, “Embedded Malware Detection Using Markov n-Grams,” in *Proceedings of the 5th international conference on Detection of Intrusions and Malware,*

*and Vulnerability Assessment*, ser. DIMVA ’08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 88–107.

[40]     C. Smutz and A. Stavrou, “Malicious PDF Detection Using Metadata and Structural Features,” in *Proceedings of the 28th Annual Computer Security Applications Conference*, ser. ACSAC ’12. New York, NY, USA: ACM, 2012, pp. 239–248.

[41]     R. Sommer and V. Paxson, “Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,” in *Security and Privacy (SP), 2010 IEEE Symposium on*, May 2010, pp. 305 –316.

[42]     Y. Song, M. E. Locasto, A. Stavrou, A. D. Keromytis, and S. J. Stolfo, “On the Infeasibility of Modeling Polymorphic Shellcode,” in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS ’07. New York, NY, USA: ACM, 2007, pp. 541–551.

[43]     N. Srndic and P. Laskov, “Practical Evasion of a Learning-Based Classifier: A Case Study,” in *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, ser. SP ’14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 197–211.

[44]     G. Stringhini, C. Kruegel, and G. Vigna, “Shady Paths: Leveraging Surfing Crowds to Detect Malicious Web Pages,” in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, ser. CCS ’13. New York, NY, USA: ACM, 2013, pp.

133–144.

[45]     G. Varghese, J. A. Fingerhut, and F. Bonomi, “Detecting evasion attacks at high speeds without reassembly,” in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*. Pisa, Italy: ACM, 2006, pp. 327–338.

[46]     B. Waske, S. van der Linden, J. Benediktsson, A. Rabe, and P. Hostert, “Sensitivity of Support Vector Machines to Random Feature Selection in Classification of Hyperspectral Data,” in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, Jul. 2010, pp. 2880–2889.

[47]     Y. Ye, L. Chen, D. Wang, T. Li, Q. Jiang, and M. Zhao, “SBMDS: an interpretable string based malware detection system using SVM ensemble with bagging,” in *Journal in Computer Virology*, vol. 5, Nov.

2008, pp. 283–293.