**基于半监督深度学习的木马流量检测方法**

谷勇浩1,2 黄博琪1,2 王继刚3 田甜3 刘焱4 吴月升4

1（北京邮电大学计算机学院 北京 100876）

2（北京邮电大学智能通信软件与多媒体北京市重点实验室 北京 100876）

3（中兴通讯股份有限公司 南京 210012）

4（百度安全 北京 100080）

（guyonghao@bupt.edu.cn）

**Trojan Traffic Detection Method Based on Semi-supervised Deep Learning**

Gu Yonghao1,2, Huang Boqi1,2 , Wang Jigang3, Tian Tian3, Liu Yan4, and Wu Yuesheng4

1（*School of Computer Science*, *Beijing University of Posts and Telecommunications*, *Beijing* 100876）

2（*Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia*，*Beijing University of Posts and Telecommunications*, *Beijing* 100876）

3（*ZTE Corporation*，*Nanjing* 210012）

4（*Baidu security*，*Beijing* 100080）

**Abstract** The existing Trojan detection technology has some problems, such as the inaccuracy of manual feature extraction, the difficulty of obtaining a large number of labeled samples, the insufficient utilization of unlabeled samples, and the low recognition rate of unknown samples. In this paper, we use semi-supervised deep learning to detect Trojan traffic, and make full use of a large number of unlabeled network traffic for model training. Firstly, the detection method based on mean teacher model is used to improve the detection accuracy. Then, in order to solve the problem that the model generalization ability is not enough due to the random noise in the mean teacher model, a detection method based on virtual adversarial mean teacher is proposed. At last, the experimental results show that the proposed semi-supervised deep learning detection method has higher accuracy in the task of two classifications, multi classification and unknown sample detection under the condition of less labeled samples; at the same time, the detection method based on virtual adversarial mean teacher model has stronger generalization performance than the original mean teacher model in the task of multi classification.

**Key words**  trojan traffic detection; semi-supervised deep learning; virtual adversarial training; mean teacher

**摘要** 现有木马检测技术存在人工提取特征不够准确，大量标记样本获取困难，无标记样本没有充分利用、模型对于未知样本识别率较低等问题。本文采用基于半监督深度学习的木马流量检测方法，充分利用大量未标记网络流量用于模型训练。首先，采用基于mean teacher模型的检测方法提高检测准确率；然后，为解决mean teacher模型中采用随机噪声导致模型泛化能力不足的问题，提出基于虚拟对抗mean teacher模型的检测方法。最后，通过实验验证本文所提半监督深度学习检测方法在少标记样本下的二分类、多分类以及未知样本检测任务中具有更高的准确率；同时，基于虚拟对抗mean teacher模型的检测方法在多分类任务中比原始mean teacher模型表现出更强的泛化性能。

关键词 木马流量检测；半监督深度学习；虚拟对抗训练；mean teacher

中图法分类号 TP391

木马，别称木马病毒，英文名为Trojan。木马通过攻击者传播感染另一台宿主计算机来完成破坏行为。其攻击行为就如同特洛伊木马典故一般，也因此得名木马。木马是如今较为流行的计算机病毒，按照病毒分类法，将不会自我复制，也不“主动”感染其他文件的病毒分类为木马。木马一般通过伪装吸引用户点击、下载和执行，为攻击者提供打开宿主机的接口，使攻击者可以任意毁坏、窃取宿主机的文件，甚至远程操控宿主机。

国内外学者对木马病毒的检测早有研究，检测方法主要有三种：1，基于程序特征码的木马检测方法。2，基于主机行为的木马检测方法。这两种基于木马样本本身的检测方法由于木马病毒的种类增多和变异较快使得检测效果下降。3，基于网络行为的木马检测方法。基于网络行为的木马检测方法因硬件捕获流量能力的提升，可以实时地识别未知流量，弥补了前两种方法的缺陷，已成为学术界的研究热点。

基于网络行为的木马检测方法的有效性主要取决于特征的提取和分类算法。传统机器学习方法依靠人工设计的特征，如基于网络协议的特征、基于进程、api等统计特征，以及支持向量机、决策树、k-means聚类和决策树等分类算法，取得了较好的检测效果。但依然存在特征设计不够准确、识别未知样本能力不足的问题。

近年来得益于人工智能领域的发展，深度学习迅速在图像识别、自然语言处理等领域取得成功[1-2]。传统的机器学习相对于深度学习而言都为浅层结构算法，这些算法在解决简单的分类问题时足够优秀，但在拟合非线性函数时，其泛化能力会下降。而深度学习利用自身多层的神经网络结构，能更好的表示事物的特征，很好的弥补了这些算法在特征选择和未知样本分类的不足。

为了解决现有木马检测方法中人工提取特征不够准确、大量标记样本难以获取、无标记样本没有充分利用、模型对于未知样本识别率较低以及半监督深度学习模型mean teacher中随机噪声引起的泛化能力不足等问题，本文采用对抗训练来改进半监督深度学习的方法，将虚拟对抗训练和mean teacher模型相结合，主要贡献包括：

（1）引入半监督深度学习模型mean teacher，充分利用未标记数据来改善模型检测性能。在USTC-TFC 2016数据集中，模型仅需要10%的标记数据就能在二分类、多分类以及未知样本分类任务中达到98%的准确率。同时，随着标记样本数量的减少，半监督深度学习模型检测性能的下降程度明显小于全监督模型。

（2）引入虚拟对抗训练来改进半监督深度学习中的随机噪声，提升模型的泛化性能。相比mean teacher模型，虚拟对抗mean teacher模型在各对比实验中具有更好的检测效果，在多分类任务中提升明显。

**1 相关工作**

**1.1**木马检测方法

基于程序特征码的检测技术是最早用于检测恶意软件的方法，因其简单和实用，很多反病毒软件和木马检测软件都使用该方法。所谓程序特征码即为一串二进制数据，用于唯一标识一种木马病毒。研究人员通过分析可疑程序样本，从中提取程序特征码，并写入木马特征库。基于程序特征码的检测技术虽然简单易行，却存在很大的滞后性。随着恶意样本变异速度的加快，安全研究人员无法分析所有的恶意样本，往往要等损害实际发生后才能更新特征库。文献[3]在此基础上，将样本的二进制特征码转换为灰度图，再利用机器学习的方法进行分类。

基于主机行为的木马检测方法是通过监视木马攻击主机的行为，从中总结非法行为的方法。这种方法一般需要实时监测主机关键系统文件与注册表值的修改、非法驱动程序的加载、远程通讯进程的植入以及特定API接口的的调用等行为来判断该程序是否为木马。基于主机行为的木马检测方法主要有以下缺点：1，可能会将未知软件的操作视为非法操作，判别为木马病毒而引起误报。2，需要实时监控主机的行为，木马病毒越复杂，需要监视的行为就越多，系统资源的消耗也就越大，影响主机的正常工作。文献[4]通过训练一个基于主机行为特征的SVM分类器，解决了基于特征码的检测方法存在的特征码提取错误和滞后问题。基于该分类器的恶意木马检测系统对60多种真实的木马都具有很高的检测率。

基于网络行为的木马检测方法又可以细分为基于网络通信负载特征匹配方法与基于通信行为的木马检测方法。

基于网络通信负载特征匹配方法原理上与流量分类所使用的深度包检测技术（DPI）相同，都是通过截获流量的负载提取并分析特征，其检测流程与基于程序特征码的检测方法相似，都需要建立一个特征数据库，当需要检测未知流量时将其特征与数据库中的特征比对，如果相匹配则发出报警。文献[5]利用DPI技术进行网络恶意软件的分类。文献[6]将这种方法用于应用层协议的恶意流量检测，如利用HTTP隧道技术的木马在通信负载中提取HTTP头部的URL、GET、和POST等参数特征，可以进行有效检测。

基于通信行为的木马检测方法主要考虑木马通信的整体行为，而不是截取流量的载荷来分析特征。这类检测方法的有效性取决于提取的通信行为特征，传统的检测方法采用k-means等机器学习方法，对木马流量进行聚类识别，特征的选取大多来自于研究者的实际经验，人工设计特征虽然有一定效果，但不够完整，具有一定的局限性。文献[7]提出通过检测木马建立通信的早期阶段的网络行为来识别木马。文献[8]基于网络行为进行木马的检测。通过检测传输层和网络层的通信行为，提取6个特征建立检测模型。该方法不受应用层流量的影响，也不受木马结构和通信协议的影响。文献[9]通过对网络流量的分析，提出一种基于端到端监控的恶意软件检测系统。该方法设计了972个不同协议和网络层的行为特征，使用事务、会话、流和会话窗口等不同流量表示形式处理数据，然后通过特征选择方法选出最有意义的特征，成功提高了检测准确率。文献[10]比较了四种最先进的机器学习算法，随机森林，序列最小优化，多层感知器和Logistic回归模型的性能结果，寻找最佳超参数和特征集选择来优化分类器。随机森林分类器通过查看网络包标识符，如包长度、包和标志计数以及到达间隔时间，识别出与TrickBot相关的流，达到了99%的准确率。文献[11]提出一种面向HTTPS隐蔽隧道的加密流量检测方法，并将木马通信划分为元数据交互阶段与加密应用数据交互阶段，提出基于时间序列模式的木马流量检测方法，测试结果表明该方法能够有效检测基于 HTTPS隐蔽隧道的加密木马。

本文所提方法是基于网络行为的检测方法，但基于网络行为的检测方法的有效性取决于人工提取的特征，而受到木马变异速度的加快，特征越来越难以设计，这就需要深度学习中的表征学习进行特征的提取。

**1.2**木马流量的表征方法

基于机器学习算法进行分类的核心在于“特征”，即通过分析原始数据并提取能反映原始对象特征的数据。如分析木马源文件得到程序特征码、基于HTTP协议的木马中提取的协议特征和基于不同通信阶段的木马流量特征等。这些特征，都是依赖专家经验人工设计的特征，而这些特征设计的准确与否直接影响分类的效果。本文使用原始流量可以很好地利用深度学习进行特征提取，从而代替人工提取特征，这就是表征学习。表征学习按照表征方法的不同，可以分为表征为图像、表征为时间序列和表征为无向图。

**a.**表征为图像

深度学习网络要求输入的原始数据为向量形式，而将其他形式转化为向量的行为称作表征学习，其中最为典型是表征为图像。在图像领域，深度学习最先得到应用，也最早用于恶意流量分类。文献[12]没有提取原始pcap格式流量的特征，而是将其处理转化为图像，即二维向量，然后使用擅长图像分类的卷积神经网络进行训练分类，成功识别恶意流量。文献[13]在该方法的训练阶段，从原始流量提取流量字段，如tcp字段信息等，作为特征序列，然后将学习到的特征序列转换为图像输入到CNN中。文献[14]通过将流量表征为图像，利用多个序列层的集成森林，交替生成类向量和特征向量，对恶意软件进行分类，提出了一种混合分层感知方法。该方法具有自动设置参数的自适应特性，因此对小尺度和大尺度数据都能很好地工作，并且具有很好的计算效率。

**b.**表征为时间序列

除了将木马通信流量表征为图像，还可以表征为一段时间序列。从数据结构的角度来看，恶意软件通信类似于自然语言，也是按顺序获取的，流量数据的顺序和依赖关系等产生了类似于“句子”的语义，具有潜在的含义。此外，由于恶意软件通信由不同目的地和通信内容组成，和自然语言中不同单词组成的不同句子可能具有相同的语义一样，前者可能具有相同的潜在功能。文献[15]根据恶意软件通信的特征与自然语言的特征之间的相似性，分析其动态行为，通过循环神经网络(RNN)寻找特征并预测未知的恶意行为。文献[16]提出了一种基于时空分层特征的神经网络检测模型（HSTF模型），并将专家设计的特征和统计特征相结合，提高了模型的自学习能力，检测基于HTTP的木马准确率达到99.4%。

**c.**表征为无向图

使用图嵌入将图像映射到一个低维特征空间，在该空间中，图像结构信息和图像属性被最大限度地保留。该方法广泛应用于节点分类、节点推荐、链路预测等。现有的图嵌入工作主要集中在无向图上。例如，struc2vec算法使用层次结构来测量不同尺度下节点相似性，并构造多层图来编码结构相似性并生成节点的结构上下文。文献[17]采用两个叠加去噪自动编码器（SDAs）进行表征学习，并考虑计算机程序的函数调用图和Windows应用程序编程接口（API）调用。给定一个程序，首先使用一种图嵌入技术，将程序的函数调用图映射到低维特征空间的向量。

以上表征方法各有优劣：将流量表征为图像的二维结构可以最大程度的保留原始信息，利于深度学习网络进行特征提取，缺点是数据预处理较为麻烦；将流量表征为时间序列有利于学习时间关系的特征，但直接将流量输入会造成维度过大，需要先将原始流量提取出协议字段，损失一定的信息。为了更好的利用以及体现深度学习模型的优势，尽可能保证输入数据的原始性，本论文拟采用表征为图像的方法。

**2基于半监督深度学习的木马流量检测**

**2.1**木马流量检测系统架构

木马流量检测系统架构如图1所示，该系统由数据准备模块，数据预处理模块，模型训练模块，以及检测和结果输出模块组成。其中输入数据是实时网络流量或静态pcap文件，然后进行数据预处理，详细处理流程在第3节描述。待测数据经过数据过滤用于模型检测，训练数据经过数据过滤和格式化用于模型训练，其中训练模型为半监督深度学习模型（包括mean teacher和虚拟对抗mean teacher两种），分别在2.2与2.3详细说明。训练好的模型提供给模型检测模块，模型检测模块接收待测数据并输出检测结果。

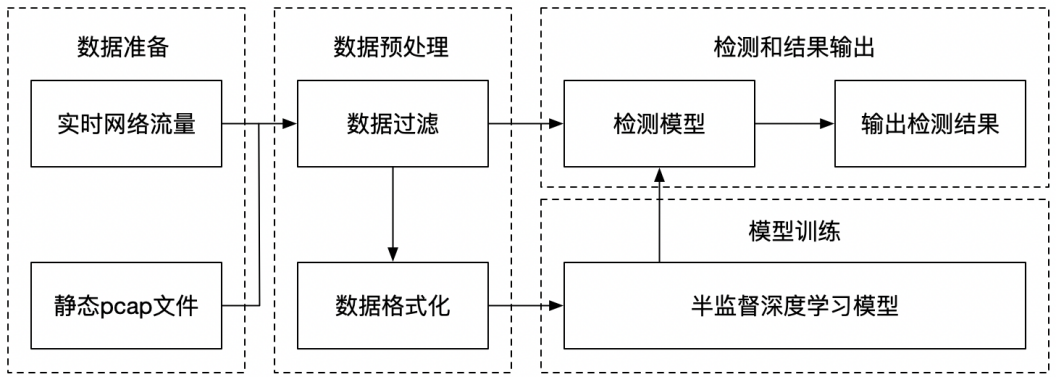


Fig.1 Trojan traffic detection system architecture

图1 木马流量检测系统架构图

## 2.2 Mean teacher模型

半监督学习是不借助外界人工交互，自动利用未标记样本来训练分类器的学习方法。网络本身的半监督学习通过将深度学习网络用于预测未标记样本，并将其预测值作为未标记样本的伪标签，随着训练的进行，网络对未标记样本的预测也将趋于正确，也就更能利用未标记样本提升模型。

相比于全监督深度学习，半监督学习由于未标记样本的分类损失函数未定义，噪声正则化本身不利于半监督学习。为了克服这一点，Mean teacher模型[18]的前身temporal ensembling模型[19]评估每个有噪声和无噪声的数据点，然后在两个预测之间应用连续损失函数。即对于已标记的样本，经过随机数据增强和dropout正则网络，输出预测值z，计算交叉熵和与前一次预测值z的平方差，作为连续损失函数；对于未标记的样本因为没有标签则不需要计算交叉熵。模型训练步骤如图2所示。

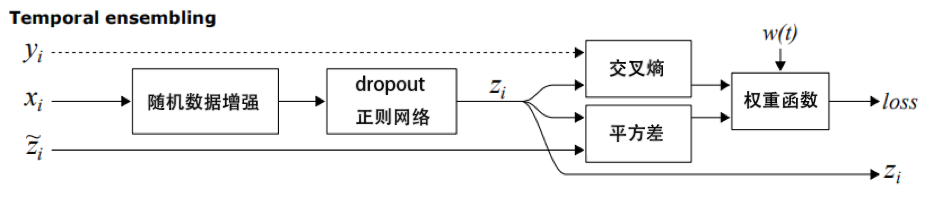


Fig.2 Temporal ensembling model training steps

图2 Temporal ensembling模型训练步骤

Mean teacher的目标是从学生网络中形成一个更好的教师网络，而不需要额外的训练。设定更好的教师网络有两种方法：第一种，在推理时，向模型中添加噪声来部分缓解softmax输出不能在训练数据之外提供准确预测的问题，因此有噪声的教师可以产生更精确的目标。这种方法被证明在半监督图像分类方面很有效[18]。第二种，模型保持对每个训练样本的指数移动平均（EMA）预测。公式表达为：

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

其中μt为学生网络的网络参数，μ\*t为教师网络的网络参数，t代表本次训练步骤，t-1为上一次训练步骤，α为超参数，本文实验中训练次数为6000 步，α从第0步的0线性增长到第4000步的1，在第一次训练步骤两个网络的μ将使用高斯算法初始化，从均值为0，方差为1的高斯分布中采样，作为初始权值。在每个训练步骤中，该批样本的所有EMA预测都将根据新的预测进行更新。因此，每个样本的EMA预测是由学生网络的当前预测和教师网络的早期预测集合形成的。这种组合提高了预测质量，并将其用作教师网络的训练目标来提高预测效果。

EMA预测参考了temporal ensembling的思想，即将上次训练的网络参数加入本次训练的预测和网络参数调整中，使得网络参数变化更为平滑。Mean teacher的EMA预测则更进一步，通过本次训练的学生网络和上次教师网络的参数来优化调整本次的教师网络，并指导学生网络的参数调整，获得更好的训练目标以及更平滑的预测结果。

Mean teacher模型除了给教师网络添加噪声外，还定义教师网络的值为学生网络的平均权重而不是平均预测值。由于教师网络是连续学生网络的平均值，我们称之为mean teacher方法（如图3所示）。与直接使用最终权重相比，在训练步骤中使用平均模型权重能产生更准确的预测效果。我们在训练中利用这一点来构建更好的目标。教师网络使用学生网络的EMA权重，而不是与学生网络共享权重。现在它可以在每一步而不是每一个训练周期之后聚合信息。此外，由于权重平均值提高了所有层的输出，而不仅仅是顶部输出，因此目标网络具有更好的抽象表示。这带来了两个实际优势：第一，目标标签越精确，学生和教师网络之间的反馈回路就越快，从而获得更好的分类准确度。第二，该方法适用于大数据集和在线学习。

图3中的连续损失函数Lm定义为学生网络预测值（权重μ和噪声γ）与教师网络预测值（权重μ\*和噪声γ′）之间的欧几里得距离与学生网络对有标记样本的交叉熵分类损失之和：

|  |  |
| --- | --- |
|  | (2) |

式（2）的为学生网络输出的预测值，为教师网络输出的预测值，为输入的样本，M为类别总数，c为类别编号，为样本是否属于c类的指示变量，相同为1，不同为0，为样本样本属于c类的预测概率。其中μ与μ\*定义由公式（1）所得，交叉熵只有有标记样本需要计算，无标记样本不需计算。

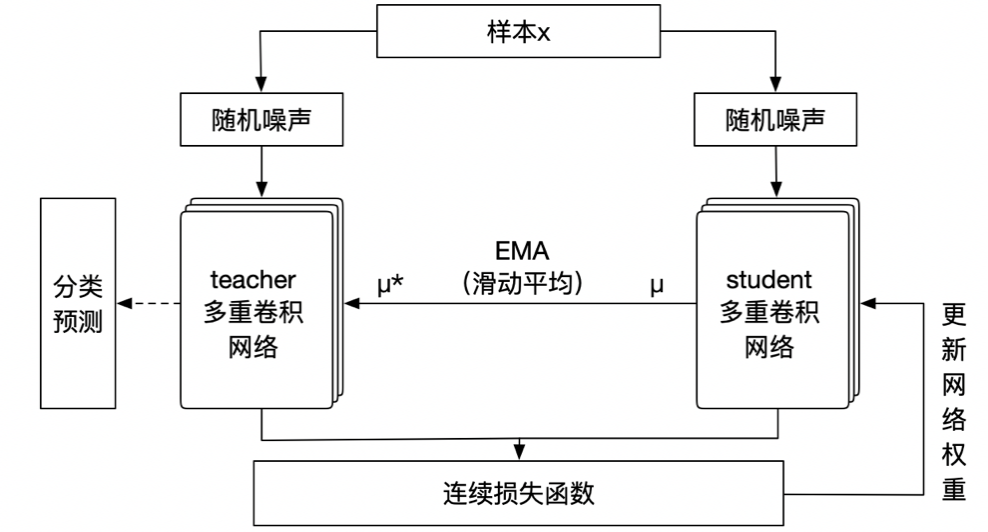


Fig.3 Trojan traffic detection model based on mean teacher

图3 基于mean teacher 的木马流量检测模型

模型中教师网络和学生网络的结构一致，参考了手写网络lenet-5的结构，如图4所示。

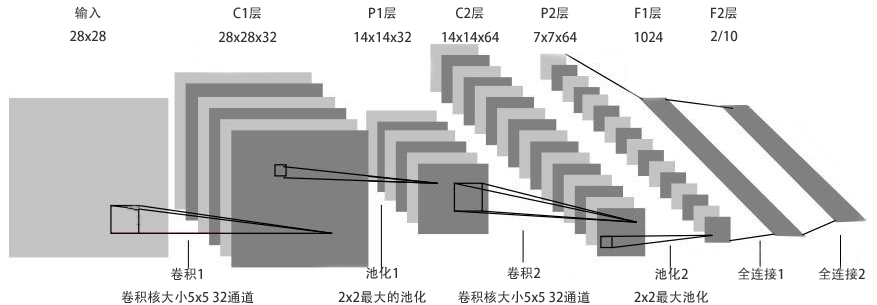


Fig.4 Network structure of teacher and student

图4 教师和学生的网络结构图

mean teacher训练流程算法：

|  |
| --- |
| **算法1：mean teacher训练流程** |
| **输入**：为输入的样本，为有标签样本, 为第t批的输入样本  为学生网络的参数，为教师网络的参数，与是采用高斯分布初始化的参数  为超参数  为随机噪声生成函数  表示含参数的学生网络模型，表示含参数的教师网络模型  epochs为训练步数，取值6000  batch为一批训练数，取值50  **输出**：网络参数为µ\*的教师网络  1 **for** i **in** [1, epochs] **do**  2 **for** t **in** [1, batch] **do**  3  *=*  /\*学生网络预测值\*/  4  5  *=*  /\*教师网络预测值\*/  6 /\*描述见公式(2)\*/  7 update µt using Gradient Descent |
| 8 **End for**  9 **End for**  10output µ\*/\*最终训练得到的网络参数\*/ |

**2.3**虚拟对抗mean teacher模型

由于未标记样本的存在，随机噪声的数据增强方式本身不利于半监督学习。因为随机噪声只是将决策边界从该数据点推离，而推离的方向是随机的，有可能有利于分类训练也可能相反。Goodfellow等人[20]提到：由于图像的输入特征的精度限制，当样本中每个元素值添加的噪声值小于样本输入特征精度时，分类器无法将样本区分开。如果图像的精度为8bit，那么样本中低于1/255的信息再输入时将会被丢弃，当输入样本有足够多时，噪声的扰动足以使得模型做出错误分类。又由于mean teacher中teacher网络为student网络的EMA预测，并生成训练目标，随机噪声可能导致错误分类，从而阻止新信息的学习。

为了克服随机噪声对模型的扰动，Goodfellow等提出对抗训练Adversarial Training，即根据输入快速生成对抗样本，并将其用于模型训练中，名为Fast Gradient Sign Method(FGSM)方法。该方法根据模型的参数、输入的样本以及样本的标签生成噪声，损失函数定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (3) |
|  |  | (4) |

其中D[p, q]为度量分布p，q之间的距离，比如交叉熵函数,p为模型的预测分布函数，q为实际的分布函数。*y*为标签，为标记样本，*r*为扰动，*θ*为模型参数，为超参数，控制扰动的大小。其中可以近似为：

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

其中*g*为模型梯度下降值：

|  |  |
| --- | --- |
|  | (6) |

常规的神经网络分类模型中，要使模型的分类准确率提升，需要不断减小损失函数的值，即在更新网络参数时将损失函数减去计算得到的梯度g。但对抗训练的方法与之相反。这种方法的核心思想是训练一个对抗扰动r，该扰动需要使得模型作出错误的分类，那么就需要沿着梯度的下降方向添加扰动r，即拉大对抗样本和原始样本的距离，这和原来的参数更新目的正好相反，因此需要在训练参数中加上而不是减去计算得到的梯度。该算法即为FGSM算法。

对抗训练除了能使模型抵御对抗样本外，在引入扰动后，模型的泛化能力也得到提升，原因在于添加的扰动与模型的权重向量高度一致,即模型会拟合产生的对抗样本，将原样本可能的盲区覆盖，达到比随机噪声更好的正则化效果，最终改善mean teacher模型的性能。

对抗训练对于有监督学习模型的泛化能力已经得到证明，但公式（3）和（4）中的样本为有标记样本，无法输入未标记的样本，故并不适用于半监督学习。所以在此之上，又提出虚拟对抗训练Virtual Adversarial Training(VAT)[21]。与Goodfellow提出的有监督学习对抗方法不同的是，VAT找的扰动方向是使预测的输出分布偏离模型预测标签的方向，而adversarial training找的是使模型预测最大地偏离正确标签的方向，这也是为什么后者只适用于有监督训练。我们可以称基于目前模型预测的标签是虚拟标签（virtual label），然后VAT根据虚拟标签计算噪声扰动，这使得VAT可以用于有监督和半监督学习。VAT算法损失函数：定义如下：

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

(8)

不同之处在于样本分为标记样本和未标记样本,前者适用公式(3)，而对于未标记样本的，则用近似，其中代表上次训练步骤的模型参数，即未标记样本的标签用模型预测出的虚拟标签y代替，其损失函数为：

(9)

(10)

其中可以近似为：

|  |  |  |
| --- | --- | --- |
|  |  | (11) |
|  | | (12) | |

虚拟对抗和mean teacher方法的结合关键在于mean teacher模型中随机噪声的替换以及在训练mean teacher网络的同时计算对抗噪声，这意味着要在mean teacher模型的损失函数上添加虚拟对抗的损失函数。由于虚拟对抗是针对单个网络进行训练的，且mean teacher中教师网络作为学生网络参数的指导者，更适合与虚拟对抗结合，使其不仅利用原模型中的EMA权重，还可以结合对抗噪声的损失函数更新学生网络的参数。那么改进模型在每个训练步骤进行梯度下降时，不止要使连续损失函数LDS下降到最小，还要同时使对抗损失函数也下降到最小，这样才能保证教师网络的目标函数是准确的，而不能单独令LDS或最小，故最终损失函数为两者之和。

Mean teacher的损失函数见公式（2），虚拟对抗的损失函数见公式（9），将两者相结合可得最终的损失函数L：

|  |  |  |
| --- | --- | --- |
|  |  | (13) |

基于虚拟对抗mean teacher方法的木马流量检测模型如图5所示。

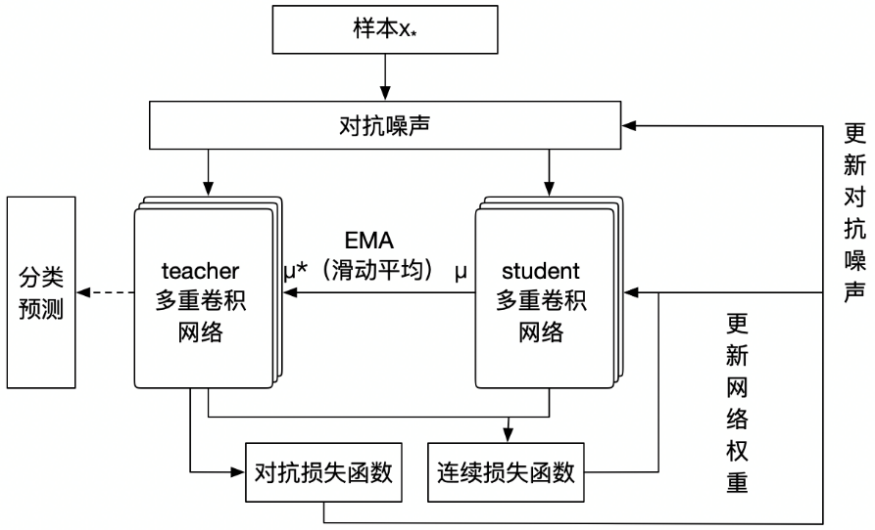


Fig.5 Trojan traffic detection model based on virtual adversarial mean teacher

图5 基于虚拟对抗mean teacher的木马流量检测模型

|  |
| --- |
| **算法2：虚拟对抗mean teacher训练流程** |
| **输入**：为输入样本，包含有标签样本和无标签样本，为第t批的输入样本  为学生网络的参数，为教师网络的参数，与是采用高斯分布初始化的参数  为超参数  为对抗噪声生成函数  表示含参数的学生网络模型，表示含参数的教师网络模型  epochs为训练步数，取值6000  batch为一批训练数，取值50  **输出**：网络参数为µ\*的teacher网络  1 **for** i **in** [1, epochs] **do**  2 **for** t **in** [1, batch] **do**  3  *=*  /\*学生网络预测值\*/  4  5  *=*  /\*教师网络预测值\*/  6  7  8 update using formula(5) & using formula(11)  9 update µt using Gradient Descent |
| 10 **End for**  11 **End for**  12output µ\* /\*最终训练得到的网络参数\*/ |

**3实验分析**

**3.1**数据集介绍

木马流量分类中普遍存在的一个问题就是缺少可用的公开数据集。经典机器学习所采用的数据集多是围绕特征展开，一般都经过人工筛选或统计工程，如常见的KDD99和NSL-KDD所提供的41个流量特征。本文希望避免繁杂的人工特征提取，利用深度学习的特性提取特征，故输入的数据越原始越好，现采用USTC-TFC2016[12]数据集作为输入。

USTC-TFC2016数据集包含恶意流量与正常流量两部分。恶意流量数据集包含Cridex、Geodo、Hitbot、Miuref、Neris、Nsis、Shifu、Tinba、Virut和Zeus等10种流量。它们截取自CTU数据集，分为原文件、截取文件和合并文件。该流量由CTU大学的研究人员于2011至2015年从真实环境采集。正常流量数据集包含BitTorrent、Facetime、FTP、Gmail、MySQL、Outlook、Skype、SMB、World Of Warcraft和Weibo等10种流量。该流量使用IXIA公司的专业仿真设备IXIABPS进行采集，涵盖了8类常用的网络应用。该数据集的文件格式为pcap，大小为3.71GB，数据集中流量采集服务器的ip和mac已用随机地址代替。

**3.2**对比模型及流量预处理

本文将对比半监督深度学习Mean Teacher模型（下称MT）、虚拟对抗Mean Teacher模型（下称VMT）、CNN模型[12]、HTFS模型[16]、机器学习模型（DT、RF[10]、GBDT），各模型流量预处理过程如下。

**a.** Mean Teacher和CNN模型预处理

深度学习模型设计之初是为了识别和分类图像，要将其用于木马流量检测需要把木马流量表征为图像。图6是流量预处理流程：

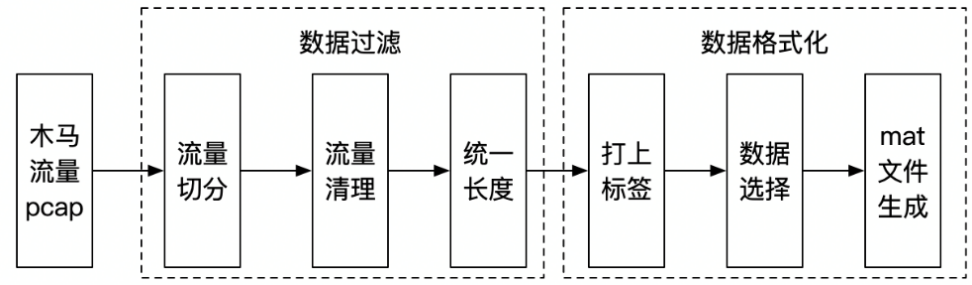


Fig.6 Network flow preprocessing

图6 流量预处理流程

**步骤1 (流量切分):** 按照一定的粒度将原始流量切分为多个离散单元，切分规则为按照会话或流的方式对数据进行切分，输入和输出数据的格式都是pcap。同一个pcap文件存储同一会话（源ip，源端口，目的ip，目的端口，协议等）的所有内容。

**步骤2 (流量清理):** 对于步骤1中应用层信息为空的会话，需要清理删除，以及对于完全相同的会话内容，仅保留一份。

**步骤3 (统一长度):**由于深度学习网络的输入要求统一大小的输入向量，必须将清理过的会话文件统一长度。本文选择统一为784字节，大于784字节则截取前784字节，小于784字节的文件在后面补充Ox00直到784字节 。之所以选取前784个字节主要基于三个因素考虑：（1）对于木马流量，基于传统机器学习在检测恶意流量时选取特征的经验，前部一般为连接建立阶段，更能体现流量的行为特点，而后续数据传输阶段因长度过长难以反映类型特征[12]。（2）实践中USTC-TFC2016数据集经过前两步骤处理后，90%的流量会话都小于784个字节，仅有少部分会话超过784字节，考虑到数据实时处理速度和检测效果的平衡，选取的长度应大于80%-90%的会话大小。（3）借鉴了经典图像识别网络lenet-5的输入向量大小(28x28)，由于该网络结构存在两个2x2大小的池化层，需要保证输入向量可以被4整除为7x7的大小。

**步骤4 (打上标签):**按照前文步骤将数据集切分得到共17921个样本，其中为满足半监督训练的要求，按比例对数据打标签，实验一中分别标注10%，5%,1%,0.1%,实验二为100%标注，实验三为10%标注。

**步骤5 (数据选择):**在实验二（检测未知木马样本实验）中需要从10种木马流量中选出一种作为未知木马，其他实验无需该步骤。

**步骤6 (mat文件生成):**将图像转换为MAT格式文件，它是CNN的输入文件标准格式之一，也是matlab数据存储的标准格式，主要作用是将大量样本压缩存储，并输入到深度学习模型中用于训练。

**b.** 机器学习模型预处理

本文使用决策树（下称DT），随机森林（下称RF）和梯度下降决策树（下称GBDT），这些模型需要从数据集中提取特征进行训练，预处理流程如下：

1. Mean Teacher和CNN模型预处理的步骤1~步骤2；
2. 提取包含数据包总数、流持续时间在内的35维流级特征[9-10]，如表1所示；
3. 将其存储为csv格式的文件。

**Table 1 Flow level features**

**表1 流级特征表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| pack\_num | 数据包总数 | std\_time\_src | 均差源时间间隔 | max\_packetsize\_src | 最大源包大小 |
| time | 流持续时间 | max\_time\_dst | 最大目的时间间隔 | mean\_packetsize\_src | 平均源包大小 |
| flow\_num | 流数目 | min\_time\_dst | 最小目的时间间隔 | std\_packetsize\_src | 均差源包大小 |
| cipher\_num | 加密组件长度 | mean\_time\_dst | 平均目的时间间隔 | max\_packetsize\_dst | 最大目的包大小 |
| packetsize\_size | 平均包大小 | std\_time\_dst | 均差目的时间间隔 | mean\_packetsize\_dst | 平均目的包大小 |
| max\_time | 最大间隔时间 | max\_time\_flow | 最大流持续时间 | std\_packetsize\_dst | 均差目的包大小 |
| min\_time | 最小间隔时间 | min\_time\_flow | 最小流持续时间 | max\_packetsize\_flow | 最大流大小 |
| mean\_time | 平均间隔时间 | mean\_time\_flow | 平均流持续时间 | min\_packetsize\_flow | 最小流大小 |
| std\_time | 均差间隔时间 | std\_time\_flow | 均差流持续时间 | mean\_packetsize\_flow | 平均流大小 |
| max\_time\_src | 最大源时间间隔 | max\_packetsize\_packet | 最大包大小 | std\_packetsize\_flow | 均差流大小 |
| min\_time\_src | 最小源时间间隔 | mean\_packetsize\_packet | 平均包大小 | cipher\_content\_ratio | 加密位为0次数 |
| mean\_time\_src | 平均源时间间隔 | std\_packetsize\_packet | 均差包大小 |  |  |

**c.** HSTF模型预处理

HSTF模型[16]采用了CNN和LSTM深度学习模型，输入的流量为原始图像数据、包级特征和流级特征。

**原始数据特征**：同Mean Teacher和CNN模型预处理的步骤1~步骤3，将其存储为npy格式文件。

**包级特征：**同Mean Teacher和CNN模型预处理的步骤1~步骤2；根据原始数据特征提取时设置的784字节计算小于等于784字节的样本比例对应地选择6个数据包进行包级特征提取，不足补0，提取的12维包级特征如表2所示，将其存储为npy格式的文件。

**流级特征：**同机器学习模型预处理流程。

**Table 2 Package level features**

**表2包级特征表**

|  |  |
| --- | --- |
| ip\_flags\_MF | IP首部标志位：分段 |
| ip\_flags\_DF | IP首部标志位：不分段 |
| ip\_ttl | 生存时间 |
| tcp\_flags\_FIN | TCP首部标志位 |
| tcp\_flags\_SYN | TCP首部标志位 |
| tcp\_flags\_RST | TCP首部标志位 |
| tcp\_flags\_PSH | TCP首部标志位 |
| tcp\_flags\_ACK | TCP首部标志位 |
| tcp\_flags\_URG | TCP首部标志位 |
| tcp\_flags\_ECE | TCP首部标志位 |
| tcp\_flags\_CWR | TCP首部标志位 |
| load\_len | 数据负载长度 |

**3.3**评价指标

实验使用评价指标包括:

|  |  |
| --- | --- |
|  | (14) |
|  | (15) |
|  | (16) |
|  | (17) |

准确率(Accuracy)用来判断模型的整体效果,精确率(Precision)用于判断模型的误报程度，召回率(Recall)用于判断模型的漏报率。其中,TP表示正确分类的目标类数目,TN表示正确分类的其它类数目,FP表示错误分类的目标类数目,FN表示错误分类的其它类数目。

**3.4**对比实验

本节设计了三组实验来验证前述模型的对比效果。包括少标记样本下的二分类实验（实验一），未知样本检测的实验（实验二）和少标记样本下的多分类实验（实验三）。

实验一：前述模型在少标记样本下（10%,5%,1%,0.1%标记样本）的二分类（木马流量，正常流量）效果对比，主要体现现实中有标记样本难以获取，而未标记样本大量存在的状况。其中MT和VMT模型的神经网络结构与CNN模型一致，网络参数一致，实验结果如图7所示。

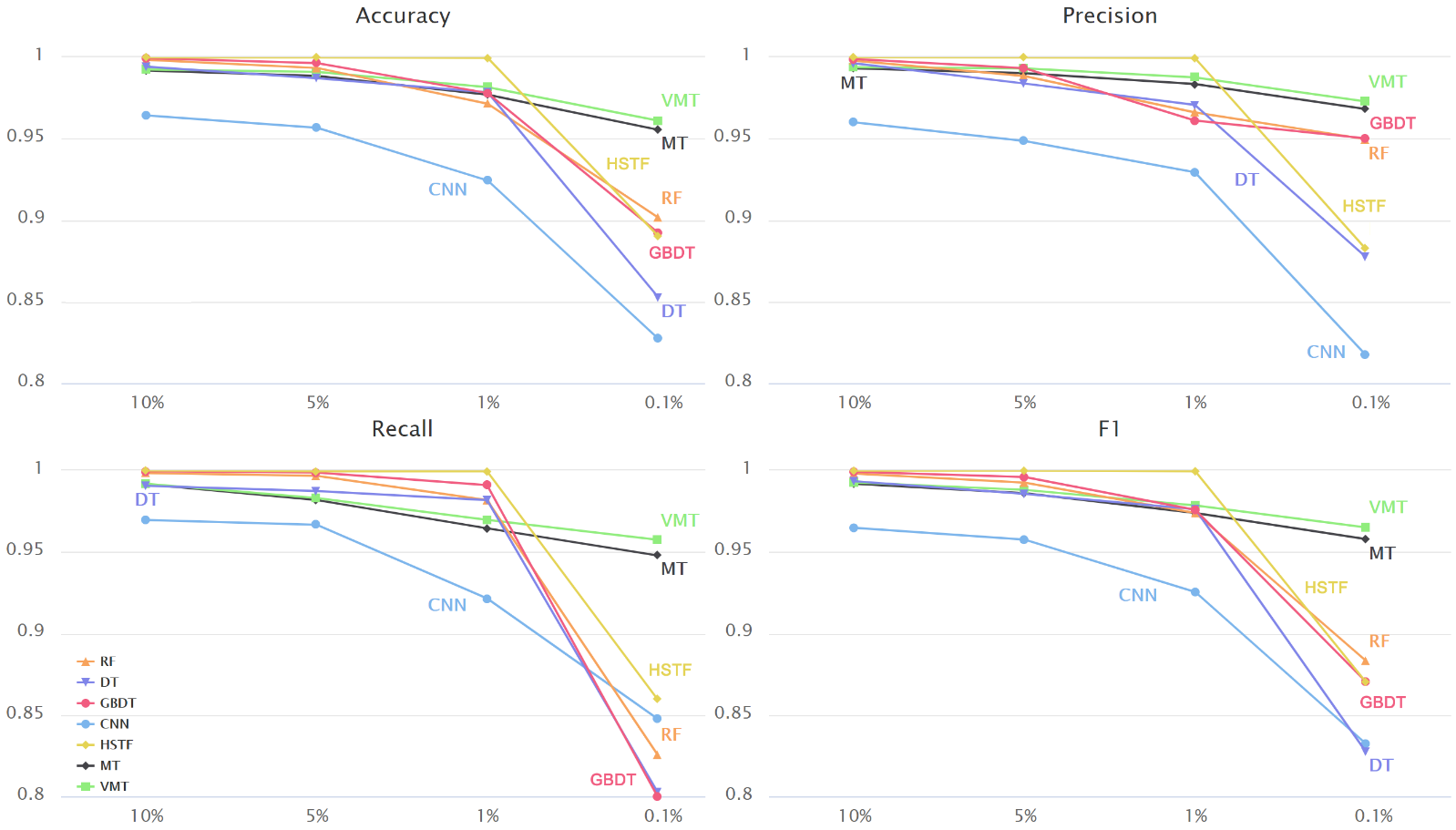
****

Fig. 7 Classification results under different proportions of labeled samples

图7 不同标记样本占比下的二分类结果图

实验二：前述模型检测未知样本的分类效果对比，即从十类样本中选取九类作为训练集，另一类作为测试集，总共重复十次，参与训练的样本为100%标记样本，主要体现现实中当模型遇到未知木马时的检测能力。该实验是为了验证半监督深度学习（MT和VMT）在检测未知样本下分类效果比有监督学习好，解决有监督学习模型对于不在训练集里的未知样本检测能力不足的问题。实验结果见图8。

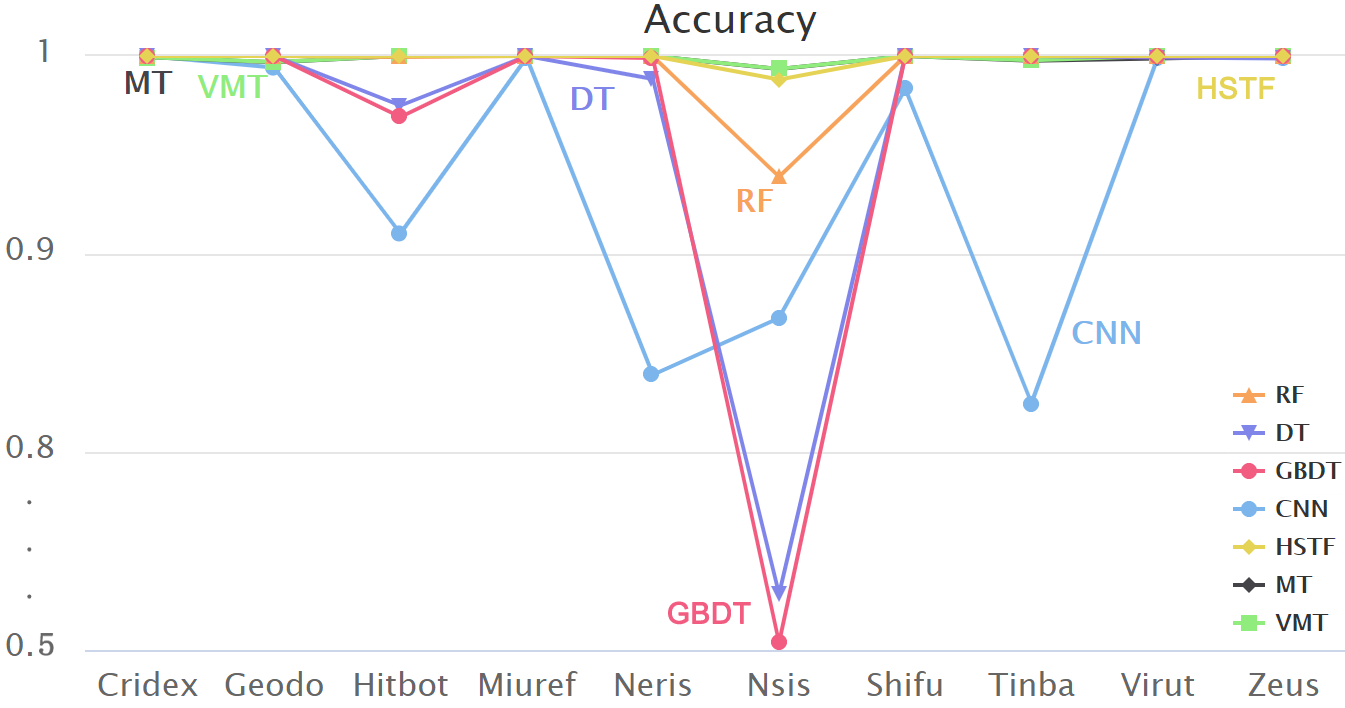
****

Fig.8 Classification accuracy rate of unknown samples

图8 未知样本的分类准确率图

实验三：前述模型在少标记样本下的多分类效果。分别设置10类木马样本以及包含所有类的全集木马样本（ALL，将每类样本的准确率按照样本量加权平均获得），标记样本占比为10%。现实中检测恶意木马除了识别木马与正常流量外，也需要进一步识别木马的种类与家族，本实验设计对比半监督深度学习（MT和VMT）与有监督学习在少标记样本下的多分类效果。多分类任务要比二分类任务困难，更能展现模型的分类检测能力，特别是在少标记样本的条件下。深度学习的网络结构与参数与少标记样本下的二分类实验一致，除了模型输出的分类数不同。实验结果见图9。

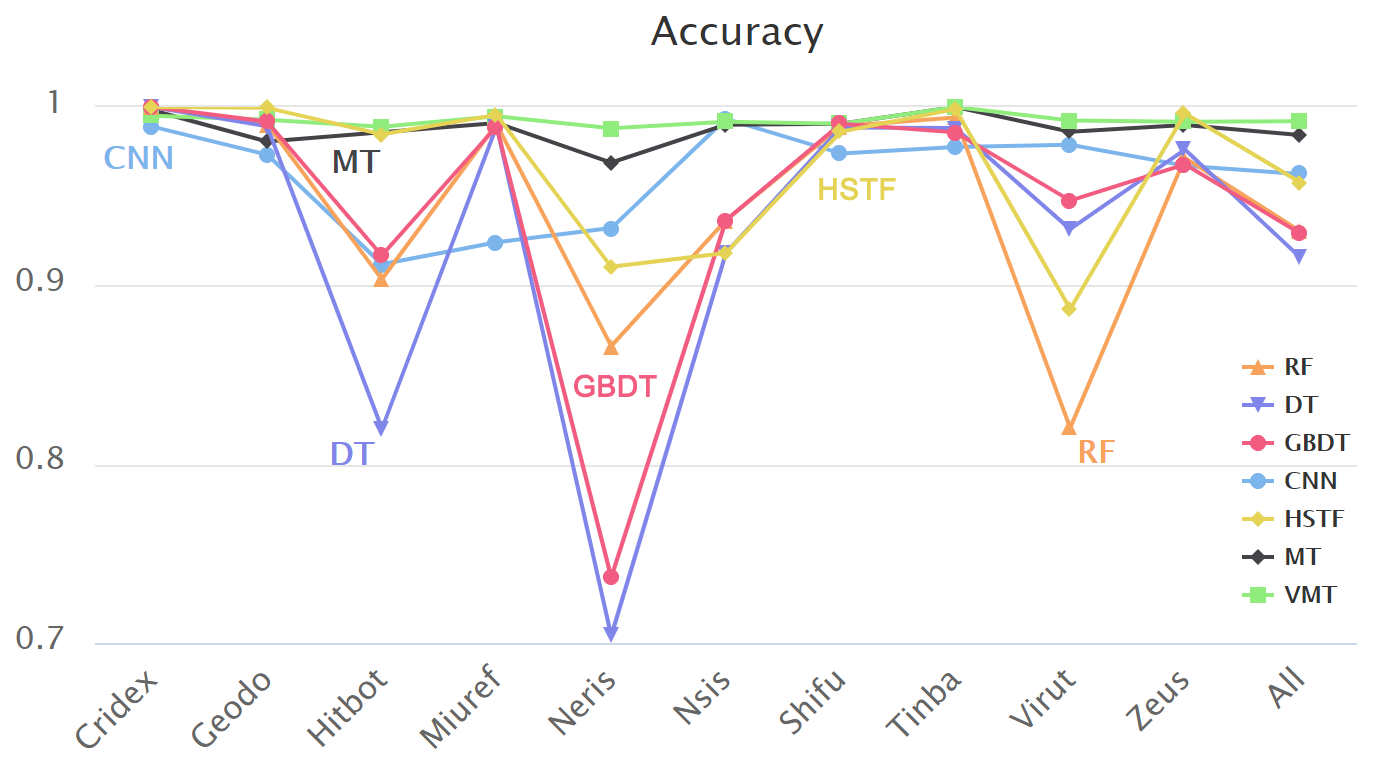


Fig.9 Ten classification accuracy rate with less labeled samples

图9 少标记样本下木马流量十分类准确率对比图

**3.5**结果分析

由图7可知，在10%标记样本下，半监督和有监督学习除CNN外，各指标都达到0.99以上的水平，说明该数据集二分类难度并不高，靠10%的标记数据集，大部分模型可以准确识别恶意流量。随着标记样本量的减少和未标记样本占比的上升，半监督模型逐渐显示出更好的分类效果。当标记样本占比下降到1%时，除HSTF模型依然保持0.99的水平外其他模型指标都有下降，但半监督模型整体效果要优于其他有监督模型。当标记样本占比下降到0.1%时，有监督模型指标出现明显下降(降到0.8至0.9区间)，而半监督依然保持在0.95以上。该实验说明：(1) HSTF模型采用了丰富的特征集，使其在1%标记样本下仍有较好的二分类效果；(2)标记样本占比逐渐减少给半监督深度学习模型造成的影响相对较小，尤其是标记样本占比下降到0.1%时，半监督深度学习模型优势更加明显。

由图8可知，半监督深度学习和有监督学习模型都具有一定的未知样本识别能力。半监督深度学习（MT和VMT）模型在10类未知样本识别准确率上达到0.99，HSTF和RF模型在9类未知样本(除Nsis)识别准确率上达到0.99，GBDT模型在8类未知样本(除Nsis，Hitbot)识别准确率上达到0.99，DT模型在7类未知样本(除Nsis，Hitbot，Neris)识别准确率上达到0.99，CNN模型在6类未知样本(除Nsis，Hitbot，Neris，Tinba)识别准确率上达到0.99。总体上看，半监督深度学习模型有较好的未知样本识别能力。

由图9可知，在标记样本占比10%的条件下，半监督和有监督学习的多分类指标相比二分类稍有下降，表明多分类任务要比二分类任务更难。图9中，半监督深度学习模型对所有类别识别准确率均在0.96以上，其中VMT模型准确率在0.98以上，优于MT模型。HSTF模型对7个类别（除Neris，Nsis，Virut）识别准确率在0.96以上，HSTF模型对7个类别（除Hitbot、Miuref、Neris）识别准确率在0.96以上，RF、GBDT、DT模型对6个类别（除Hitbot，Neris，Nsis，Virut）识别准确率在0.96以上。该实验说明：（1）半监督深度学习模型在多分类任务上的识别准确率明显优于有监督学习模型（2）在多分类检测任务中，VMT模型比MT模型表现出更强的泛化性能。

此外，综合图7和图9可知，在标记样本占比继续减少，尤其是标记样本占比降低到0.1%时，半监督深度学习模型在多分类任务中识别准确率的优势将更加明显。

**3.6**模型复杂度

除了检测性能外，本文还考虑到模型的复杂度。神经网络的网络层数和训练样本量是影响模型训练时间复杂度的主要因素。

对于CNN模型，时间复杂度来自卷积层和全连接层的计算，卷积层(Convolution Layer, 简称CL)的计算时间取决于输入样本量和卷积核大小，公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (18) |

其中为卷积层的层数，l为本层，M为特征图的大小，K为卷积核大小，为上层输入的大小，为本层输出的大小。对于全连接层(Fully Connected Layer, 简称FC)，层之间的神经元互相连接，是每层神经元状态维数的乘积，公式如下：

|  |  |  |
| --- | --- | --- |
|  |  | (19) |

其中为全连接层的层数，l为本层，为本层的神经元数量。

对于半监督深度学习模型，其时间复杂度和CNN模型类似，学生网络和教师网络在训练中为滑动平均关系，实际进行卷积计算的为学生网络,而损失函数只定义了参数的更新方向，使用的都为梯度下降，所以MT和VMT的时间复杂度相同。公式（18）（19）中的参数在训练前就已确定，属于常量，因此模型的时间复杂度主要取决于训练样本量N，即O（N）。

对于HSTF模型，除了卷积层和全连接层外，还需要LSTM层的计算，以及包级特征和流级特征的提取，因此HTFS模型的时间复杂度相比其他模型更高。LSTM时间主要由输入的序列长度决定，如式（20）所示，L是序列长度，CLSTM是单元状态维数。

|  |  |  |
| --- | --- | --- |
|  |  | (20) |

对于其他机器学习模型，DT的训练时间复杂度是O（N\*M\*D），其中N是训练样本量，M是特征数量，D是树的深度。RF和GBDT是在DT的基础上分别采用bagging和bootsting方法提升分类性能。这些模型都需要提取流级特征，需要完整扫描一遍流量，而MT和VMT仅需流量的前784个字节，因此前者样本量要远大于后者。

**3.7**模型参数

**表3模型超参数表**

|  |  |  |
| --- | --- | --- |
| CNN | epochs训练步数 | 6000 |
| batch一批训练数 | 50 |
| Learning rate学习率 | 0.003 |
| MT  VMT | epochs训练步数 | 6000 |
| batch一批训练数 | 50 |
| Learning rate学习率 | 0.003 |
| EMA超参数 | 从0线性增长到1 |
| Input noise噪声超参数 | 0.15 |
| 扰动超参数 | 5 |
| HSTF | epochs训练步数 | 200 |
| batch一批训练数 | 128 |
| Learning rate学习率 | 0.003 |
| DT | max\_leaf\_nodes最大叶节点 | 150 |
| RF | max\_leaf\_nodes最大叶节点 | 120 |
| n\_estimators估计量 | 260 |
| GBDT | max\_depth最大深度 | 4 |
| max\_leaf\_nodes最大叶节点 | 210 |
| n\_estimators估计量 | 240 |
| min\_samples\_leaf最小叶 | 7 |

**4总结**

总结全文，本文主要做了以下贡献：

（1）将半监督深度学习与木马流量检测相结合，把图像识别领域的mean teacher模型进入木马流量检测，将木马流量数据转化为图像向量数据输入到深度学习网络中，实现了自动提取木马流量特征以及未标记样本的利用，避免了人工特征选择，与全监督深度学习相比在二分类、多分类以及未知样本分类上效果更好。同时，随着标记样本数量的减少，半监督深度学习模型检测性能的下降程度明显小于全监督模型。

（2）将虚拟对抗训练与mean teacher模型相结合，用虚拟对抗训练获得的对抗噪声代替mean teacher模型中的随机噪声，引入对抗训练的思想提高模型泛化能力，达到识别未知样本的目的。

尽管本文提出的方法和模型改善了木马流量检测性能，但依然存在以下不足：

（1）系统模型识别速度不够快，对于实时的网络流量检测需要大数据平台的支持，瓶颈主要在流量预处理阶段，还存在改进空间。预处理阶段的步骤1需扫描一遍pcap文件，时间复杂度为o(n)，n为pcap文件数量，步骤2和3可同时进行，时间复杂度为o(m)，m为五元组会话数量，步骤4和5的时间复杂度为o(k)，k为样本类的数量，步骤6的时间复杂度为o(n)，n为pcap文件数量。

（2）本文使用到的模型为半监督深度学习模型，依然需要一定数量和类别的木马流量标记样本保证检测的准确率。

**参 考 文 献**

1. J. S. Bai, M. H. Zhang, and J. Bi. Survey of attacks and countermeasures for SDN[J]. ZTE Communications, 2018, 16(4):3–8.
2. S. H. Wang, Q. Z. Zhuo, H. Yan, et al. A network traffic prediction method based on LSTM[J]. ZTE Communications, 2019, 17(2):19–25.
3. Liu L, Wang B. Malware classification using gray-scale images and ensemble learning[C]// 2016 3rd International Conference on Systems and Informatics (ICSAI). IEEE, 2016:1018-1022.
4. Wang P, Wang Y S. Malware behavioural detection and vaccine development by using a support vector model classifier[J]. Journal of Computer & System Sciences, 2015, 81(6):1012-1026.
5. Boukhtouta A, Mokhov S A, Lakhdari N E, et al. Network malware classification comparison using DPI and flow packet headers[J]. Journal of Computer Virology and Hacking Techniques, 2016, 12(2):69-100.
6. Liming Wang, Haoying Mu, Lin Xu, Jinglin Chen, Xiyang Liu and Ping Chen. TrojanURLDetector: A Statistical Analysis Based Trojan Detection Mechanism[J]. Information Technology Journal, 2010, 9(6): 1124-1132.
7. Jiang D, & Omote K. A RAT Detection Method Based on Network Behavior of the Communication’s Early Stage[J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2016, 99(1):145-153.
8. Li S, Yun X, Zhang Y, et al. A Novel Approach of Detecting Trojan Based on Network Behavior Analysis[C]// International Conference on Communication Technology Proceedings(ICCT). IEEE, 2012:513-518.
9. Bekerman D, Shapira B, Rokach L, et al. Unknown malware detection using network traffic classification[C]// 2015 IEEE Conference on Communications and Network Security (CNS). IEEE, 2015:134-142.
10. Ali Gezer, Gary Warner, Cliffrd, et al. A flow-based approach for Trickbot banking trojan detection[J]. Computers & Security, 2019, 84:179-192.
11. JIA Fangfang, CHEN Shi, WU Shuang, et al. Encrypted Trojan Detection Method Based on HTTPS Hidden Tunnels[J]. Journal of Information Engineering University, 2019, 020(004):461-466,479.
12. Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, & Yiqiang Sheng. Malware traffic classification using convolutional neural network for representation learning[C]// 2017 International Conference on Information Networking (ICOIN). IEEE,2017:712-717.
13. Tobiyama S, Yamaguchi Y, Shimada H, et al. Malware Detection with Deep Neural Network Using Process Behavior[C]// 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC). IEEE, 2016:577-582.
14. Roseline S A , Sasisri A D , Geetha S , et al. Towards Efficient Malware Detection and Classification using Multilayered Random Forest Ensemble Technique[C]// 2019 International Carnahan Conference on Security Technology (ICCST). 2019:1-6.
15. Shibahara T, Yagi T, Akiyama M, et al. Efficient Dynamic Malware Analysis Based on Network Behavior Using Deep Learning[C]// Global Communications Conference. IEEE, 2017:1-7.
16. Xie J , Li S , Yun X , et al. HSTF-Model: an HTTP-based Trojan Detection Model via the Hierarchical Spatio-Temporal Features of Traffics[J]. Computers & Security, 2020:101923.
17. Jiang, H., Turki, T., & Wang, J. T. L. DLGraph: Malware Detection Using Deep Learning and Graph Embedding[C]// 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE,2018:1029-1033.
18. Tarvainen A, Valpola H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results[C]// Annual Conference on Neural Information Processing Systems (NIPS). 2017:1195-1204.
19. Laine S, Aila T. Temporal Ensembling for Semi-Supervised Learning[C]// International Conference on Learning Representations (ICLR). 2016:1-13.
20. Goodfellow, Ian J, Shlens, Jonathon, Szegedy, Christian. Explaining and Harnessing Adversarial Examples[C]// International Conference on Learning Representations(ICLR). 2015:1-10.
21. Takeru M, Shin-Ichi M, Shin I, et al. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2018, 41(8):1979-1998



**Gu Yonghao**, born in 1980. PhD, lecture, master supervisor. Member of China Computer Federation. His main research interests include network security, and machine learning.



**Huang Boqi**, born in 1994. Master's degree. He main research interests include network security.



**Liu** **Yan,** born in 1985. Head of AI security team of Baidu X-lab. His main research interests include Web security, AI security, and machine learning.



**Wu** **Yuesheng,** born in 1987. Senior research fellow of Baidu X-lab. His main research interests include Web security, AI security, and machine learning.



**Wang Jigang**, born in 1978, PhD, researcher, Senior member of CCF. His main research interests include network security, operating system and artificial intelligence.



**Tian Tian**, graduated from TU Dortmund in Germany majoring in Electronic Information Engineering, and worked at Infineon Technologies AG in Germany as a wireless sensor network R&D engineer before joining ZTE Corporation in 2009.