小样本下未知内部威胁检测的方法研究

王一丰 郭渊博 李 涛 孔 菁
(信息工程大学密码工程学院 郑州 450001)

摘 要 极少量的内部威胁通常被淹没在海量的正常数据中,而传统的有监督检测方法在此很难发挥作用。此外,各类新形式内部威胁的出现使得传统需要大量同类标记样本数据学习特征的方法在实际中并不适用。针对检测未知内部威胁,文中提出了一种基于原型的分类检测方法。该方法使用长短期记忆网络提取用户行为数据的特征,通过在特征空间上比较与各类原型的距离(余弦相似度)来发现未知内部威胁,并采用元学习方法更新参数。最终通过基于 CMU-CERT 的合成数据集的实验也验证了该方法的有效性,在小样本条件下,对新出现的未知内部威胁的分类的准确率达到了 88%。

关键词 小样本学习,未知内部威胁,元学习,原型网络
中图法分类号 TP393 **文献标识码** A

Method for Unknown Insider Threat Detection with Small Samples

WANG Yi-feng GUO Yuan-bo LI Tao KONG Jing
(Cryptography Engineering Institute, Information Engineering University, Zhengzhou 450001, China)

Abstract Few insider threats are usually covered by a mass of normal data. It is difficult for traditional anomaly detection method based on machine learning to detect insider threats because of lacking in sufficient labeled data. To detect these unknown insider threats with small samples, this paper proposed a method based on prototypical networks witch used Long Short Term Memory networks to extract the features of user behavior data and updated parameters by meta learning. This method uses cosine similarity to classify new class samples which are not seen in training set. The experimental results with generated data based on CMU-CERT dataset finally show that the proposed method is effective, and the classification accuracy of detecting unknown insider threat is 88%.

Keywords Few-lhot learning, Unknown insider threat, Meta learning, Prototypical networks

1 引言

随着网络攻击的泛滥,研究者们越来越重视网络安全。组织网络不仅遭受着来自外部的攻击,更加伴随着内部人员带来的内部安全威胁。此外,随着大数据、物联网、人工智能等技术方案的发展与使用,不仅使得网络空间安全环境日趋复杂,也同时丰富了攻击的手段与路径。依靠网络安全专家分析和设计检测的传统方法在当今海量数据环境下效率低下,目前业内多采用伴随大数据一同流行的机器学习方法来解决现阶段诸多的网络安全问题。早在 20 世纪 80 年代就有学者将机器学习应用于入侵检测领域[1],但受限于当时的计算条件,该方法并未引起重视。近年来,随着技术的发展,愈发充足的计算条件使得机器学习方法逐渐成为解决现今诸多网络安全问题的主要手段,很多顶级安全领域会议上相关的文章也层出不穷[2]。

机器学习技术应用过程中有两个关键,即算法和数据。处理机器学习问题的关键之一是在特定应用问题上找到明显优于其他方法的合适算法[3],而目前元学习(Meta Learning)的研究正朝着自动化这一过程的方向探索,试图找到将数据集映射到算法的函数[4]。此外只有足够的高质量数据,才能充分训练机器学习模型且最终达到预期效果。然而,机器学习包括深度学习方法在实际应用中常常受到缺乏大量标记训练数据的阻碍[5-6]。而这其中深度神经网络(Deep Neural Network)模型的多层神经网络结构可以学习刻画数据本质属性的特征,对可视化和分类等任务有很大帮助[7]。尽管该模型在语音识别和图像视频识别等任务中显示出了很大的优势,但现有的人工神经网络结构还远远不及生物神经网络结构复杂[8],因其参数众多所以更加需要足量的数据来训练模型。

在网络安全领域,尤其在内部威胁检测中,实际中的很多时候可能没有足够多的标记数据来使用合适的机器学习方法,如 DNN。尽管组织内每天发生的安全事件不计其数,但实际的内部威胁检测中存在缺少数据的原因主要有以下 3 点:

1)所拥有的数据极度不平衡。负例数据(内部威胁数据),特别是几类发生频率较低的威胁在样本中所占数量极少。例如,针对用户的各类隐蔽的威胁行为建模时会发现,比起海量正常数据组织中此类内部威胁数据的数量很少。因此,目前的内部威胁检测方法主要是基于指标和基于异常的方法[9],但这些方法都无法关联特定的威胁类型或场景,且后

本文受国家自然科学基金项目(61602515,61501515)资助。
王一丰(1994—),男,硕士,主要研究方向为网络安全与深度学习,E-mail:wyfssr001@163.com(通信作者)。

续需要大量的人工分析。

2）缺乏足够的标签数据。组织所拥有的数据是没有标签的，对内部威胁来说这些攻击通常被淹没在海量正常数据中难以分辨。这不仅是因为内部人员可能比外人更了解组织的安全策略，而且更重要的是恶意活动通常只是用户使用其组织的信息系统整体活动的一小部分。单纯依靠领域专家从中区分攻击和正常数据显然成本太高，无法实行。

3）未知内部威胁没有对应样本数据。恶意的内部人员在完成攻击之前可以通过采用一些逃避检测的方法来隐藏他们的行为。因此有效的内部威胁检测系统必须不仅能检测已知的内部威胁，而且还要能检测训练时不曾见过的新类型的内部威胁，本文把这种威胁叫做未知内部威胁[10]。显然，这些新类型的未知内部威胁是没有对应的负例样本数据的。

因此，在内部威胁检测的应用中普遍存在小样本的情况，要运用机器学习方法不得不考虑如何解决标签样本稀缺这一困难。幸运的是，小样本问题不仅仅是在内部威胁检测中存在，在图像识别领域也存在，因此研究者们很早就开始了对小样本学习的研究，该方面的研究也叫做 Few-shot Learning，具体相关知识将在第 2 节中阐述。

然而，目前在内部威胁检测领域中基本没有看到结合小样本学习的例子。一方面很多学者不考虑实际内部威胁检测中存在小样本情况，另一方面很多研究者认为目前的小样本学习方法无法在该领域适用。对于前一种看法，前文提到对于内部威胁检测来说，实际中的种种原因都可能会造成小样本的情况。对于后一种看法主要有两种理由：1）目前小样本学习的分类效果对比传统的大数据范式来说很不理想，如表 1[11] 所列，各方法在泛化性更强的 miniImagenet 小样本数据集上的分类正确率普遍较低。2）目前，图片识别领域的小样本学习方法对内部威胁检测来说不能简单套用，跨领域应用难度大。综上，在小样本的情况下检测发现未见过的内部威胁困难重重，本文提出了一套可行的解决方案以供参考。

表 1　几种方法在 miniImagenet 上的分类准正确率表

（单位：%）

Method	5-way 1-shot	5-way 5-shot
Baseline fine tune	28.86±0.54	49.79±0.79
Baseline nearest neighbors	41.08±0.70	51.04±0.65
Matching network[14]	43.56±0.84	55.31±0.73
Prototypical network[13]	49.42±0.78	68.20±0.66
Graph neural network[20]	49.80±0.22	65.50±0.20
Meta-learning LSTM[18]	43.44±0.77	60.60±0.71
Model-Agnostic[11]	48.70±1.84	63.10±0.92

本文基于度量学习的思想，用原型网络来识别分类新类型的未知内部威胁，并采用元学习的方法来更新模型参数，所提出的方法具有灵活分类模型训练时从未见过的未知内部威胁的能力。因此，在实际中该方法可以依据少量样本快速识别新出现的未知内部威胁，而不需要再重新训练或设计模型，达到帮助组织迅速发现关联新威胁的效果。很多组织往往在新型威胁泛滥后才开始响应，在此之前组织内部是否已经遭受了攻击则无从得知，该方法有助于解决该问题。

本文第 2 节详细介绍了小样本学习的相关方法和未知内部威胁的研究背景；第 3 节设计了一个应用场景并详细描述了本文提出的检测方法；第 4 节是对本文实验的描述及对实验结果的分析；最后总结全文。

2　相关工作

人类可以轻松地从一张斑马图片学习到斑马这个概念，且下次见到实物时就能将其认出。人类拥有从少量样本中迅速学习一个新概念的能力，而机器学习中的算法则通常需要数百或更多的样本来执行类似的操作[12]。科学家们长期以来一直探索机器的小样本学习能力，以期望机器最终能够达到类似人脑的学习认知能力。目前，小样本学习的研究多应用在图片识别领域，因此大多方法都是在该领域分类效果最好的神经网络模型上改进的。目前，在小样本学习的研究中有几种成熟方法可供参考，其中采用较多的经典方法主要有 3 类，分别是 Fine Tune 方法、基于度量学习（Metric Learning）的方法和基于元学习（Meta Learning）的方法。本文采用了度量学习的思想来检测分类，并采用元学习方法来优化参数。

基于迁移学习的 Fine Tune 方法出现最早。该方法先在一个含有丰富标签数据的大规模数据集中训练一个基础网络，再用特定的小样本数据集来对该基础网络的某些参数进行调整，最终使得模型很好地适应特定的小样本数据集。训练时会固定基础网络的部分参数，使用很多训练技巧对特定的参数进行训练。例如，最常用的方法是通过通用数据集 Imagenet 训练基础的图片分类网络，再使用少量的特定数据（如医学领域中的某种罕见疾病的 CT 图片）来调整模型的部分参数。该方法原理简单，但使用时需要很多训练和调参技巧才能得到不错的效果。

第二种是基于度量学习的方法，该方法的思想是对样本间的距离分布进行建模，使得同类样本靠近异类样本远离，具有很好的泛化性。具体来说有不同的实现方法，例如 Snell 等提出了一种基于原型网络（Prototypical Network）的方法[13]，其思想简单且效果很好。它学习一个度量空间，通过计算各类的原型来进行分类。即每个类别都存在一个原型，该类的原型是少量支持集映射在特征空间中布雷格曼散度（Bregman Divergence）下的均值。分类新样本时分别计算与每个类别的原型的欧氏距离，其中距离最短的就识别为该类数据。同样的，Vinyals 等[14]给出了另一种经典方法，提出了基于注意力的匹配网络（Matching Network）模型。具体来说，先从支持集中训练一个分类网络，之后该网络对每个未见过的样本分别计算该样本到支持集中其他样本的余弦相似度，其中对支持集和测试集样本都使用了不同的特征提取函数。其创新体还体现在建模过程上，训练匹配网络模型在支持集下的测试集的训练误差最小，并且这与实际测试的过程是完全一致的。

最后，基于元学习方法的小样本学习方法是指机器每次的新任务都不是从零开始，是依据之前的任务经验快速学习新任务，而不是孤立地考虑新任务。元学习的目标是在各种不同的任务上学习出一个元模型，这样可以仅用少量样本就能解决一些新的学习任务。这种方法的挑战在于模型需要结合之前的经验和当前新任务的少量样本信息，并避免在新数据上过拟合。其中，元数据由以前任务学习的知识组成，是用于高效开发新任务时的有效假设。在元学习中训练了一个过程（meta learner）来生成一个分类器（learner），使得分类器在新任务的测试集上获得高精度。经典方法有递归记忆神经网

络模型、meta-learning LSTM 和模型无关自适应方法。

早在 2001 年，基于记忆神经网络的方法就被证明可用于元学习[15]。Santoro 等[16-17]利用递归记忆模型，通过权重更新来调节偏差，并将其缓存到记忆中来调节输出。文献[17]基于神经网络图灵机(Neural Turing Machines，NTM)，NTM 能通过外部存储进行短时记忆，还能通过缓慢权值更新进行长时记忆，因此可以学习记忆策略并用这些来进行分类，由此该方法可以快速准确地预测那些只出现过一次的数据。在 meta-learning LSTM 的方法中[18]研究了在少量数据下基于梯度的优化算法失败的原因。在小样本条件下的梯度优化算法无法在几步内完成优化，特别在非凸问题上多种超参数的选取无法保证其收敛速度。其次，不同任务随机初始化会影响任务收敛到好的解上，因此需要一种通用的初始化方法，和迁移学习不同的是，它初始化后能保证让模型从一个好的起点开始。文献[18]采用 LSTM 充当元模型，用其状态表达目标分类器的参数更新，最终学会如何在新分类任务上对分类器网络进行初始化和参数更新。文献[11]使用了模型自适应的方法，其用少量的迭代步骤就可以获得较好的泛化性，该方法可以学习任意标准模型的参数并快速适配该模型。而且这种方法无需关心模型的形式，也不需要为元学习增加新的参数，直接用梯度下降方法来训练。

此外，其他采用较多的还有贝叶斯网络、最大似然估计[19]和图神经网络[20]等方法来进行小样本学习，这里不再赘述。

另一方面，对于未知内部威胁的检测，目前效果最好的方法是文献[21]中的方法。文献[21]提出了一种基于集合的无监督技术检测方法，并在组织 8 个月的实际用户活动数据中进行检测未知内部威胁的实验；设计了一种多个不同分类器的集合来进行未知内部威胁的检测，其工作流程为首先从各个不同分类器中得出关于最异常点的共识，那么该共识应当被保留在最终的结果中，其次因为各个单独的分类器都受到算法模型、输入特征、超参数设置等偏差的影响，所以应该优先选择来自具有不相关偏差的分类器结果的组合，最终该方法在基于 CMU-CERT 的合成数据集中的实验取得了不错的检测效果。

3 小样本下的分类检测模型

目前几乎没有人将小样本学习中的技巧应用到网络安全中。研究者们通常集中于在 KDD[22]、DARPA Intrusion Detection Sets[23]等公开数据集上研究某类问题的解决方案，如入侵检测、身份认证等。这些研究固然取得了非常不俗的效果，但这些在充足数据下设计的方法在实际中可能会因缺乏数据而受阻。因此，本文提出了一种基于度量学习和元学习的未知内部威胁检测方法，该方法有两大特点：1)训练好的模型可以快速依据少量之前未见过的新样本学习识别该类样本的能力；2)下一个学习任务基于以前任务的经验快速学习，以前任务的经验可以用于下次的分类任务。

设计一个应用场景：一个中小型组织拥有一段时间未标记的用户历史行为数据、少量发现的各类实际威胁数据以及安全专家依据新出现类型的攻击在组织内模拟的少量未知内部威胁数据。这些数据经过标准化并处理后作为小样本学习模型的各类输入数据。然而组织中很多安全人员缺乏对新类型未知内部威胁的安全知识，现需要用这些数据设计并训练一个模型，该模型可以通过少量攻击数据快速发现组织中新出现的未知内部威胁，并且每次检测任务完成后可以保留经验并不断学习更新。

3.1 整体分类方法

基于文献[14]和文献[18]的思想，该文提出了一种小样本下检测未知内部威胁的方法。该方法使用计算与每类原型的余弦相似度来分类未见过的测试样本，并采用 LSTM 函数提取样本的深度特征。首先本文定义了如下几个概念，分别是包含 k 个训练样本的训练支持集 $S=\{(x_i,y_i)\}_{i=1}^{i}$ 和包含 n 个测试样本的训练测试集 $B=\{(x_i',y_i')\}_{i=1}^{n}$，其中 x_i' 是处理后的一段时间安全事件的多维特征向量，y_i' 是 x_i' 对应的标签，共 m 类，k 和 n 的数值都很小(小样本条件下)，S 和 B 是同类样本。训练过程的分类方法用式(1)的形式可以进行最简单的概括，其中 $\hat{y}_i'$ 是模型预测标签权重的线性组合，a 是下文的基于原型的分类函数，c 是各类的原型。具体流程如图 1 所示，依据度量学习的思想，最重要的是找到最适合的特征提取函数 g 来最大化不同类别样本之间的距离，此外模型的训练和测试过程都遵循此流程。

$$\hat{y}_i'=\sum_{j=1}^{n}a(x_i',c_j)\cdot y_j \tag{1}$$

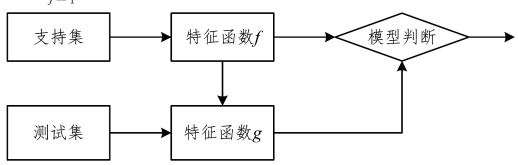

图 1 小样本下分类模型的工作流程图

3.2 特征提取函数

一般来说，实验直接将样本作为模型输入的效果不好，因此本文采用了特征提取函数 f 和 g 分别提取支持集样本 x_i 和测试集样本 x_i' 的深度特征并将其作为模型的输入。该文使用了双向 LSTM(Bidirectional Directional LSTM)[24]来提取支持集 S 的样本深度特征，即 f 是一个通过支持集 S 训练得到的双向 LSTM 模型，其输入在本质上是一个有时间信息的多维张量，因此选用在语言、音频的处理上效果较好的 LSTM 模型来进行特征提取。此外，由于样本数据量很少，为了充分利用支持集数据选择双向 LSTM 模型使得同类样本之间的信息互通。传统的单向 LSTM 模型只依据前面的若干输入决定当前的输出，而双向 LSTM 模型需要由前面若干输入和后面若干输入共同决定，这样最大程度地实现了同类样本之间信息的互通。最终支持集样本 x_i 的深度特征 $f(x_i)$ 的提取方法如式(2)所示：

$$\begin{aligned}&\overrightarrow{h_i},\overrightarrow{c_i}=\mathrm{LSTM}(x_i,\overrightarrow{h_{i-1}},\overrightarrow{c_{i-1}})\\&\overleftarrow{h_i},\overleftarrow{c_i}=\mathrm{LSTM}(x_i,\overleftarrow{h_{i-1}},\overleftarrow{c_{i-1}})\\&f(x_i)=x_i+\overrightarrow{h_i}+\overleftarrow{h_i}\end{aligned} \tag{2}$$

对于测试集样本来说，特征提取函数 g 是在支持集特征提取函数 f 的基础上得到的。由于支持集是作为一个整体输入到双向 LSTM 模型中的，而测试样本是每个单独输入进行预测的，因此不能直接套用。可以利用支持集上的特征提取函数 f 设计测试集特征提取函数 g。这里选择带有注意力机制的 LSTM 模型[25]来进行测试集的特征提取，g 基于之前的 f，如式(3)所示。对于每一个测试样本，通过 $q(q=1,2,\cdots,n)$

次迭代来得到每个测试样本的特征。

$$h_q', c_q = \mathrm{LSTM}(x_q', h_{q-1}, c_{q-1})$$
$$h_q = h_q' + x_q' \tag{3}$$
$$g(x_q', S) = \sum_{i=1}^{k} \mathrm{softmax}(h_{q-1}^T \cdot f(x_i)) \cdot f(x_i)$$

3.3 基于原型的分类函数

本文采用了一个基于原型网络的函数 a 作为分类函数。首先构建一个基于支持集 S 的各类原型集合 $C=\{c_i\}(i\in\{1,\cdots,m\})$，其中第 i 类样本原型 c_i 的计算方法如式(4)所示：

$$c_i = \frac{1}{|S_i|}\sum_{(x_i,y_i)\in S_i} f(x_i) \tag{4}$$

集合 $C=\{(c_i,y_i)\}_{i=1}^m$ 是 m 类原型 c_i 的集合。如式(5)所示，基于原型的分类函数 a，通过计算新样本与每类原型之间的余弦相似度得到对应的标签权重。这里之所以选用对相似点更为敏感的余弦相似度作为衡量标准而不采用对离群点更为敏感的欧氏距离，是因为威胁样本中同样包含大量正常行为数据，这些威胁数据与正常数据相比只有细微的差别。并且后文的实验结果表明，使用对相邻点更为敏感的余弦相似度比对离群点更为敏感的欧氏距离在分类效果上要好 12% 左右。

$$a(x_i', c_j) = \frac{\exp\left(\dfrac{g(x_i', S)\cdot c_j}{\| g(x_i', S)\| \times \| c_j \|}\right)}{\sum_{k=1}^{m}\exp\left(\dfrac{g(x_i', S)\cdot c_k}{\| g(x_i', S)\| \times \| c_k \|}\right)} \tag{5}$$

3.4 训练策略

本节讨论模型的训练策略，该方法中的训练和测试采用同样的策略进行。训练过程采用 5-way 5-shot 的方式，即每次训练集包含 5 类每类各 5 个样本。训练时迭代一次的流程如下：1)将样本划分为支持集和测试集；2)利用本次迭代的支持集，计算测试集的训练误差；3)计算梯度，更新参数。真正测试过程同样如此，特别的在完成训练之后，所有训练中用过的类别都不再出现在后续真正的测试中。换言之，训练集和测试集的类别是互不包含的。

该方法采用基于文献[11]的元学习方法更新优化参数。实际中，由于该文的训练过程中也有测试集 B，因此实际在此用测试误差代替了训练误差。将整个小样本分类任务定义为 T，构建其一次的采样或分解训练任务为 T_i。该方法会在新的采样任务上依据之前任务的学习经验训练局部的梯度下降参数，目标是找到对任务敏感的参数，使得任务的变化能较大影响模型的损失函数。全局参数代表了任务 T 的梯度下降参数，在参数下每次多分类任务中交叉熵损失函数 L 如式(6)所示：

$$L_{T_i}(\varphi(\theta)) = \sum_{x_i', y_i'\sim B} y_i' \cdot \log \hat{y_i}' + (1-y_i')\cdot \log(1-\hat{y_i}') \tag{6}$$

训练关于全局模型参数 θ 的具体任务的梯度下降参数 θ'，使其在 T 的各个采样任务上的误差和最小，过程如式(7)所示：

$$\min_{\theta} \sum_{T_i\sim T} L_{T_i}(\varphi(\theta')) = \sum_{T_i\sim T} L_{T_i}(\varphi(\theta) - \alpha \nabla_\theta L_{T_i}(\varphi(\theta))) \tag{7}$$

其中，步长 α 是超参数。

由于在小样本学习中仅有少量样本供训练，因此需要在几步之内就完成模型参数的优化。这要求我们找到最有效的梯度下降方法，使得每次任务仅通过几步梯度下降就可以最大程度地优化分类效果。跨任务的元学习任务采用随机梯度下降(Stochastic Gradient Descent，SGD)来更新模型参数。其他自适应的优化方法例如 Adam 方法尽管在深度学习的大部分情况下表现很好，但是其完全自动训练的参数不一定能适应本文的小样本情况且依赖大量训练数据。而 SGD 方法虽然存在速度较慢且容易陷入局部最优等问题，但由于本文实验中的 5-way 5-shot 方式中的训练样本很少，因此来通过人工调参和元学习方法来解决并适应小样本下的优化过程，其过程如式(8)所示：

$$\theta = \theta - \beta \cdot \nabla_\theta \sum_{T_i\sim T} L_{T_i}(\varphi(\theta)) \tag{8}$$

其中，β 是元步长。

元学习方法更新参数的过程如算法 1 所示。有内外两个循环，外循环是训练元学习的参数 θ，即一个全局的梯度下降的参数；内循环对每个采样任务 T_i 分别做梯度下降，进而在全局 θ 上做梯度下降。对于每次新任务，初始梯度下降参数为当前的 θ。其中模型学习的原型和 θ 不断更新，这使得模型有了记忆的特性。

算法 1　元学习更新参数算法

Input：小样本任务 T，步长超参数 α，β

Output：更新后的全局参数 θ

1. randomly initialize θ //＃随机初始化元参数
2. while not done do
3. Sample batch of task $T_i\sim T$ //任务采样
4. for all T_i do
5. Evaluate $\nabla_\theta L_{T_i}(\varphi(\theta))$ using (x_i', y_i') from T_i//计算当前任务的损失函数
6. Compute adapted parameters：$\theta_i' = \theta - \alpha\nabla_\theta L_{T_i}(\varphi(\theta))$ //计算当前梯度方向
7. meta update
8. end for
9. Update $\theta = \theta - \beta\nabla_\theta \sum_{T_i\sim T} L_{T_i}(\varphi(\theta_i'))$ using each L_{T_i}//更新元参数
10. end while

4 实验

为验证所提方法在内部威胁检测中的有效性，本文模拟前文的应用场景设计了实验。本节介绍了实验的数据、方法和结果。最终的实验结果表明，在当前实验环境下，本文提出的方法应用在未知内部威胁检测方面的效果很好，证明了该方法的有效性。

4.1 数据源

在实验数据方面，传统的小样本相关实验多在使用深度神经网络的图片识别领域中研究，并且图片识别领域中有一些公认的小样本数据集，如 Omniglot[26] 和 miniImagenet[27] 数据集。而在未知内部威胁检测领域中没有公认的小样本数据集使用，为了模拟应用场景，实验采用在 CMU-CERT 数据集[28]基础上的合成数据集。该合成数据集包括了正常数据、少量真实内部威胁数据以及安全专家模拟的攻击数据，这些数据代表了各种情景的实际内部威胁经验[29]。CMU-CERT 数据集包含了组织内部 4000 名用户在 516 天的实际活动数据，并由安全专家模拟了一些威胁数据。CMU-CERT 数据集包含用户登录和注销设备记录、用户邮箱活动相关的记录、

用户的文件活动记录、连接和断开可移动设备记录、用户使用互联网的相关活动记录和用户角色信息等，具体采用的信息如表 2 所列。此外，在 CMU-CERT 数据集的基础上，本实验依据数据集中已有的少量真实攻击数据进行模拟，合成了几类内部威胁数据，最终采用的数据包括了敏感数据窃取和共谋攻击等 10 类数据。此外，在划分观测序列时，不同的时间间隔会对序列长短、序列数量以及序列划分的有效性会产生不同的影响。为了使正常工作时的数据中同类样本尽量接近，实验中数据以 24 小时为间隔进行分割，且只采用了工作日的用户行为数据。

表 2 数据集中包含信息表

用户登录数据	时间	用户	机器	活动	（登陆/注销）
用户使用可移动存储设备记录数据	时间	用户	机器	文件名	活动(打开/写入/复制/删除)
用户收发邮件数据	时间	用户	机器	来源	目标 活动(查看/发送)
用户使用互联网数据	时间	用户	机器	网址	活动(上传/下载/访问)

4.2 实验结果

实验在 Docker[30] 中搭建 TensorFlow 框架的 GPU 版本，采用 python 语言来搭建原型测试系统。测试环境 CPU 为 Intel i5-9600@3.7 GHz，内存为 16 GB，硬盘为 1 TB 机械硬盘。实验采用上述的数据，针对 10 种不同类型的攻击（外部渗透敏感数据窃取攻击、内部单人敏感数据窃取、内部共谋数据窃取攻击、内部单人恶意破坏资源、内部共谋破坏资源、内部单人访问不安全站点、外部窃取身份攻击、内部单人权限滥用、内部共谋权限滥用），采用 5-way 5-shot 的方式进行，即每次任务选择 5 类数据作为训练集，再选择另外不同的 5 类数据作为测试集。采用训练集的 5 类每类各 5 个数据作为训练支持集，剩下的数据作为训练测试集。当模型在训练集 5 类样本的分类效果达到最优后完成本次训练，并使用测试集以同样的 5-way 5-shot 方式进行测试。也就是说，实验中模型训练时所用的 5 类数据与真正测试时所用的 5 类数据是互不包含的不同类别的数据。这说明测试的数据是该分类模型训练时从未见过的新类型的数据，因此可以用于检测未知内部威胁实验。整体实验流程如图 2 所示。

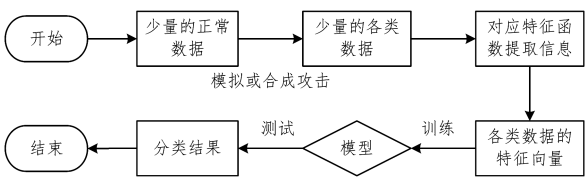

图 2 实验流程图

实验中除了本文提出的方法，我们还实现了其他的几种方法作为检测这些不同种类威胁的对比。实验分别采用了不用特征提取的方法、不用原型网络的方法、采用欧氏距离的方法和本文所采用的方法进行对比。几种方法的模型准确率如表 3 所列，分别包括不用特征提取函数、不用原型网络（匹配网络）、采用欧氏距离和本文方法的平均分类正确率（Accuracy）对比，分别是对选择不同的 5 类攻击样本作为训练集 S 的实验结果的平均。从中可以看出，不用特征函数而直接将标准化后的特征向量作为模型的输入效果很差，这主要是由于没有充分利用特征向量的时序信息。其次，文献[13]已经阐述了使用原型网络的好处，效果对比不采用原型网络逐个匹配的方式提升显著，实验中也证明了这一点。最后对比对离群点更为敏感的欧氏距离，采用对相似点更为敏感的余弦相似度的效果更好。其原因在于，内部威胁的隐蔽性和对抗性，本实验中所用的各类内部威胁数据相对于正常数据来说差异很小。

表 3 几种方法在本实验中的平均分类正确率

方法	5-way 5-shot 分类正确率
不用特征提取	0.44
不用原型网络	0.60
采用欧氏距离	0.76
本文方法	0.88

由于一般的深度学习模型如 CNN 和 DBN 对数据的依赖度较大，在本实验的 5-way 5-shot 环境下难以发挥其作用，因此最后选择与同样研究检测未知内部威胁的[21] 集成学习方法做比较。在基于集成模型的方法检测未知内部威胁方法中，该方法同样基于 CERT 数据集，在 8 个月的数据中利用各种不同的机器学习模型集合投票来发现未知内部威胁。对于未知内部威胁来说，虽然集成学习的方法不需要少量未知威胁样本，但相对的输出结果同样不包含检测威胁类型且需要长期训练。从表 4 中可以看出，本文方法与集成学习方法的效果差不多，但使用的数据和计算资源更少，且本文方法不需要长期数据的训练，能实现及时快速的发现。

最后基于元学习的参数更新方法能够在有限的几步内最有效地更新参数。实验对比了该算法和其他两种方法在 3 步内梯度下降方法性能的表现。图 3 中的结果显示了该文采用的方法对比预训练方法和随机初始化方法在小样本下的效果。实验结果表明，本文方法可以在仅学习少量样本的情况下在几个梯度更新中更快地适应模型。

表 4 本文方法与集成学习方法的比较

未知内部威胁检测方法	未知内部威胁的检测率
集成学习方法	0.89
本文方法	0.88

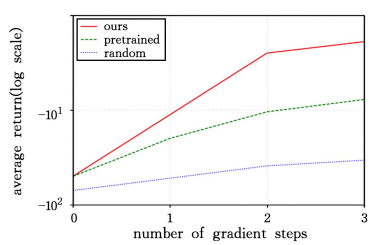

图 3 小样本下的几种参数更新方法的对比

该文利用 CMU-CERT 数据集为基础的小样本数据集对该方法检测未知内部威胁的有效性进行了验证，在 5-way 5-shot 的方式下，本文模型对于未见过威胁的平均正确分类准确率达到了 0.88，高于其他对比模型。本文提出的方法对企业安全人员在缺乏对新类型威胁的知识时快速发现组织内部此前未见过的未知内部威胁，及时有效检测隐蔽的恶意用户行为具有一定的帮助作用。

结束语　真正的内部威胁是复杂且具有对抗性的，这使得我们用于检测这些威胁的系统必须能够检测安全人员从未见过或预期的未知内部威胁。特别是在未来，随着软件定义网络(Software Defined Network，SDN)的发展，必然有着大量新类型的内部威胁出现。本文提出了一个基于小样本学习的模型，该方法结合并优化了此前在图像识别领域有不错表现的一些方法，所设计的模型能够仅通过少量新样本就快速识别此前从未见过的未知内部威胁。该方法对于组织在数据不足且安全人员对新威胁缺乏了解的情况下防御新出现种类的未知威胁有很大帮助，该方法不依赖于底层拓扑，并且可以通过分段式学习来积累学习经验。值得注意的是，由于该方法是在小样本的条件下设计的，如果有充足且合适的数据，其检测效果可能不如传统的大数据机器学习方法。

参 考 文 献

[1] MUKHERJEE B，HEBERLEIN L T，LEVITT K N，et al. Network intrusion detection[J]. IEEE Network，1994，8(3)：26-41.

[2] 张蕾，崔勇，刘静，等. 机器学习在网络空间安全研究中的应用[J]. 计算机学报，2018，9：1943-1975.

[3] KOTSIANTIS S B. Supervised machine learning：a review of classification techniques[J]. Informatica (lithuanian Academy of Sciences)，2007，31(3)：249-268.

[4] VILALTA R，DRISSI Y. A perspective view and survey of meta-learning[J]. Artificial Intelligence Review，2002，18(2)：77-95.

[5] KRIZHEVSKY A，SUTSKEVER I，HINTON G E，et al. ImageNet classification with deep convolutional neural networks[C]// Neural Information Processing Systems，2012：1097-1105.

[6] LECUN Y，BENGIO Y，HINTON G. Deep learning[J]. Nature，2015，521(7553)：436.

[7] HINTON G E，SALAKHUTDINOV R. Reducing the dimensionality of data with neural networks[J]. Science，2006，313(5786)：504-507.

[8] 焦李成，杨淑媛，刘芳，等. 神经网络七十年：回顾与展望[J]. 计算机学报，2016，39(8)：1697-1716.

[9] YOUNG W T，GOLDBERG H G，MEMORY A，et al. Use of domain knowledge to detect insider threats in computer activities[C]// IEEE Symposium on Security and Privacy. 2013：60-67.

[10] SENATOR T E，GOLDBERG H G，MEMORY A，et al. Detecting insider threats in a real corporate database of computer usage activity[C]// Knowledge Discovery and Data Mining. 2013：1393-1401.

[11] FINN C，ABBEEL P，LEVINE S，et al. Model-agnostic meta-learning for fast adaptation of deep networks[J]. International Conference on Machine Learning，2017：1126-1135.

[12] LAKE B M，SALAKHUTDINOV R，TENENBAUM J B，et al. Human-level concept learning through probabilistic program induction[J]. Science，2015，350(6266)：1332-1338.

[13] SNELL J，SWERSKY K，ZEMEL R S，et al. Prototypical Networks for Few-shot Learning[J]. Neural Information Processing Systems，2017：4077-4087.

[14] VINYALS O，BLUNDELL C，LILLICRAP T P，et al. Matching networks for one shot learning[J]. Neural Information Processing Systems，2016：3637-3645.

[15] HOCHREITER S，YOUNGER A S，CONWELL P R，et al. Learning to Learn Using Gradient Descent[J]. International Conference on Artificial Neural Networks，2001：87-94.

[16] SANTORO A，BARTUNOV S，BOTVINICK M M，et al. Meta-learning with memory-augmented neural networks[C]// International Conference on Machine Learning. 2016：1842-1850.

[17] SANTORO A，BARTUNOV S，BOTVINICK M M，et al. One-shot learning with memory-augmented neural networks[J]. arXiv：Learning，2016.

[18] RAVI S，LAROCHELLE H. Optimization as a model for few-shot learning[C]// International Conference on Learning Representations. 2017.

[19] LI F F，FERGUS R，PERONA P，et al. One-shot learning of object categories[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence，2006，28(4)：594-611.

[20] SATORRAS V G，ESTRACH J B. Few-shot learning with graph neural networks[C]// International Conference on Learning Representations. 2018.

[21] YOUNG W T，MEMORY A，GOLDBERG H G，et al. Detecting unknown insider threat scenarios[C]// IEEE Symposium on Security and Privacy. 2014：277-288.

[22] LI Y H，XIA J B，ZHANG S L，et al. An efficient intrusion detection system based on support vector machines and gradually feature removal method[J]. Expert Systems with Applications，2012，39(1)：424-430.

[23] LIPPMANN R P，CUNNINGHAM R K. Improving intrusion detection performance using keyword selection and neural networks[J]. Computer Networks，2000，34(4)：597-603.

[24] HOCHREITER S，SCHMIDHUBER J. Long short-term memory[J]. Neural Computation，1997，9(8)：1735-1780.

[25] VINYALS O，BENGIO S，KUDLUR M. Order matters：sequence to sequence for sets[C]// Trnational Conference on Learning Representations. 2016.

[26] LAKE B M，SALAKHUTDINOV R，GROSS J，et al. One shot learning of simple visual concepts[J]. Cognitive Science，2011，33(33).

[27] RUSSAKOVSKY O，DENG J，SU H，et al. ImageNet large scale visual recognition challenge[J]. International Journal of Computer Vision，2015，115(3)：211-252.

[28] LINDAUER B，GLASSER J，ROSEN M，et al. Generating test data for insider threat detectors[J]. Journal of Wireless Mobile Networks，Ubiquitous Computing，and Dependable Applications，2013，5(2)：80-94.

[29] CAPPELLI D M，MOORE A P，TRZECIAK R F. The CERT Guide to Insider Threats：How to Prevent，Detect，and Respond to Information Technology Crimes[M]. Hoboken：Addison-Wesley Professional，2012.

[30] MERKEL D. Docker：lightweight linux containers for consistent development and deployment[J]. Linux Journal，2014，2014(239)：2.