title

abstract

# Introduction

在现实场景中，要获取大量的标注数据往往需要投入大量的人力，物力，在大多数情况下，只有少量数据是带标签的，能大量轻易获得的是无标签数据。

在大多数应用场景中，存在容易获取的大规模未标记数据，这些数据可能包含与已知异常相同类别的异常，或新的异常类别，即未知异常。因此，除了已标记的异常样本之外，利用这些未标记的数据来检测已知和未知异常同样至关重要。无监督[1]方法不受样本标签的限制，但由于缺乏带标签异常的先验知识，精度欠佳。半监督[2]方法，其利用标记的异常来构建异常检测模型，但这些方法忽略了未标记数据中可能异常的有效信息。一个可能的解决方案是使用当前的异常检测方法从未标记数据中检测一些伪异常或者说生成伪标签，然后合并伪标签和真实标签来训练，以构建更广义的异常检测模型。然而，伪标签可能有很多误报，合并后会增加标签的噪声。

考虑到这些问题，在所提出的强化学习方法中采用两种奖励机制，自动和交互地拟合给定的异常示例，并同时检测无标签数据中的已知和未知异常。通过引导智能体有效地利用少量带标签的异常，同时有意识地探索大规模的未标记数据以查找任何可能的异常，实现对带标签异常学习和无标签异常探索的平衡。

[1]Liu F T, Ting K M, Zhou Z H. Isolation-based anomaly detection[J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2012, 6(1): 1-39.

[2]Alsoufi M A, Razak S, Siraj M M, et al. Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review[J]. Applied sciences, 2021, 11(18): 8383.

# background

## Basic theory

异常检测的目标为识别离群对象，其与一般分布不一致，例如机械中缺陷，图片或者视频中的特定对象，垃圾邮件等都可以划分为异常检测的范畴。异常检测已被证明在许多领域至关重要，例如网络安全，医疗检测等。本文侧重研究异常检测中的网络入侵检测这一场景。网络流量通常指的是一段时间内的流量，具有一定的时序性，在这样时序的场景下，异常一般可以分为点，上下文和集合异常。点异常即与全局大多数点都不一样的某些点，也就是离群点，这些点周围的点密度一般都比较低。上下文异常一般是指某个时间点所表现出来的特征与其前后的时刻有较大差异。在集合异常的情况下某一个点可能不表现出异常，但是这个点所在的集合一起表现出异常，在时序的场景则可能是连续一段时间内的所有点一起表现出异常。网络攻击针对计算机或者其他基础设施采取各种形式的入侵，这些异常行为会造成各种破坏，导致软件，服务等失效，或者隐私数据被窃取。

常见的异常有拒绝服务攻击，僵尸网络，侦察攻击等。拒绝服务攻击(Denial of service, DoS)是一种攻击者持续向受害者发送大量数据包，使受害者无法正常提供服务的网络攻击。攻击者通过占用受害者的关键系统资源，阻止受害者接收和处理合法的请求。分布式拒绝服务攻击(Distributed Denial of Service, DDoS)是一种基于拒绝服务攻击的大规模、广泛的攻击方式，该攻击通常涉及数千台感染了来自不同网络的恶意软件的主机，其对网络的破坏更为严重。许多僵尸网络被用来启动分布式拒绝服务攻击，例如Bashlite、Mirai和Torii。Bashlite利用Linux系统和物联网设备中的漏洞发起多种不同类型的分布式拒绝服务攻击。Bashlite可以向受害者的TCP或UDP端口发送随机字符串，或重复发送具有指定标志位的TCP数据包。Mirai是一种恶意软件，它可以将使用Linux操作系统的网络设备变成远程控制的机器人，使它们成为僵尸网络的一部分，然后可以发起大规模的分布式拒绝服务攻击。Mirai专门攻击物联网设备，如IP摄像头和家庭路由器。Mirai使用一张包含常见出厂默认用户名和密码的表来识别易受攻击的物联网设备。受mirai感染的网络设备还会不断扫描物联网中的其他设备，从而发起攻击。Torii是一个复杂的物联网僵尸网络，隐藏了多种Mirai变种。Torii具有强大的数据泄露能力，可以从被感染的物联网设备中窃取数据并长期保持感染状态。受感染的物联网设备可能被用于挖矿或其他非法在线活动。

典型的侦察攻击有操作系统指纹和业务扫描。攻击者在未经授权的情况下收集受害者的信息，目的是寻找受害者的安全漏洞。攻击者通过启动操作系统指纹来确定受害者运行的操作系统类型。业务扫描用于检查受害者开放端口上运行的业务。盗窃攻击的方式有键盘记录和数据泄露。键盘记录通过记录键盘输入来窃取受害者的数据。数据泄露通过非法复制受害者的信息来达到目的。暴力破解是一种密码分析攻击。攻击者试图通过系统地检查所有可能的密码和密码短语来找到受害者的正确密码。随着密码规模的增大，穷举所需的时间也呈指数级增长，因此这种穷举攻击对于简单任务更加有效。

根据从行为，日志，审计数据等各种可以获取的网络信息中提取到的特征，通过各种方法进行异常检测，可以实现威胁的预防，损失的估计以及对可能攻击的预测等。在进行异常检测时，可以采用无监督算法，也可以采用有监督算法，可以采用机器学习方法，也可以利用神经网络，使用深度学习方法。孤立森林是其中比较有代表性的算法，其具有线性复杂度并且易于解释，在设计本文的异常检测方法时也有借助到该算法。

孤立森林是一种无监督的决策树算法，通过分类出不同节点所需的深度来确定异常。在一个数据平面上用一个超平面进行切割，将该平面分割成两个子空间，不断循环这个过程直到切割出来的子空间只包含一个点。异常点周围的密度较低，往往需要较少的分割次数就能将其分割在一个子空间内，反之，正常点周围的密度较高，往往需要更多的分割次数。这就是孤立森林的基本思想，反应在决策树上就是异常点被分离为叶子节点时需要的次数更少，离根节点更近。在训练的过程中记录从根节点到叶子节点的路径数，根据式(2-1)计算异常得分：

其中为在个样本中，样本的异常得分。在孤立森林的训练过程中会训练t棵树，为样本在t棵决策树上由根节点到叶子节点的路径长度的均值。为训练过程中由n个样本构造的t棵决策树的平均路径长度，是对树整体路径长度的一个平均度量。分离出样本所需的路径长度越小，即该样本能更容易的被切分到一个孤立的空间，那么就会越小，就会越大。根据公式可知一个点的异常得分越高，则越有可能是异常。

与监督学习，无监督学习，半监督学习一样，强化学习是一种学习方式。强化学习从环境与智能体不断的交互过程中获取带有奖励的各种信息，从而学会在特定场景下采取最佳动作，获取最大奖励，最终达成特定的目标。不同于其他三种学习方式，强化学习并非直接依赖数据的标签或者利用数据间的相似度进行聚类，强化学习根据环境反馈的奖励，即可以量化的激励或者惩罚，不断调整训练对象的行为模式，达成训练目标，强化学习的基本流程如图所示。

图2-1 强化学习基本流程

强化学习的主体主要由环境和智能体构成，环境和智能体之间通过状态，动作和奖励不断的交互。

⚫环境：整个训练和测试过程的大背景。环境会向智能体反馈状态和奖励。

⚫智能体：强化学习的训练对象，即强化学习算法或模型。

⚫状态：对环境整体的一个观测值，即环境和智能体在当前时刻的一个总体情况，环境会向智能体反馈状态用以判断应该执行什么动作。

⚫动作：智能体可以采取一系列动作，根据当前状态，依据特定的算法或者策略，智能体执行该状态下的最佳动作以获取最大奖励，同时动作会改变环境的状态。

⚫奖励：智能体在特定状态下采取特定动作后，环境会向智能体反馈奖励，或正向或惩罚。智能体会倾向于获取正向的奖励，从而可以设计合理的奖励机制引导智能体学会在某一状态下做出正确的动作。

理论上认为强化学习的训练过程是马尔可夫决策过程。强化学习中的马尔可夫决策过程为智能体采取动作改变状态并获取奖励这一与环境交互的循环过程。马尔可夫决策的核心思想在于下一步的状态只与当前状态和当前采取的动作有关，与更早的状态和动作无关，马尔可夫过程如式(2-2)所示。

⚫：为某一特定状态，为状态集合。

⚫：为某一特定动作，为动作集合。

⚫：状态下采取动作的状态转移概率矩阵，或者可以理解成，即根据当前状态采取动作后转移到下一状态的概率。

⚫：也可以理解为，即智能体在状态下采取动作之后所获得的即时奖励。

同时在整个过程中还存在着策略，用于表示智能体在状态下执行动作的概率。

强化学习算法用于指导智能体的行为，按照智能体选择动作的标准可以把算法分为基于价值和基于概率两类。基于价值的方法输出的是各个动作的价值，最终会选择价值最大的动作执行。基于概率的方法会输出下一步可能采取的各个动作的概率，根据概率来选择最终要执行的动作，一般会选择概率最大的动作执行。其中比较有代表性的强化学习算法有DQN和PPO2。

在最早的强化学习算法例如Q-learning和Sarsa中并未利用到神经网络，由表2-1可知，Q-learning需要不断维护，当有很多状态和动作时，状态和动作之间会产生大量的组合，要维护这些组合需要极大的开销，这时采用神经网络可以很好的解决这些问题。例如图2-2所示，可以输入状态或者状态加动作，输出价值用于下一步要执行的动作的判断，而无需记录维护每一个状态动作对的价值。

表2-1 Q-learning伪代码

|  |
| --- |
| 初始化集 |
| 每一轮重复训练过程： |
| 初始化状态 |
| 迭代过程： |
| 在状态下根据策略，如贪心策略，选择动作执行 |
| 执行动作，得到奖励和下一状态 |
|  |
|  |
| 直到达到终止状态 |

图2-2 强化学习神经网络使用方法

DQN就是将神经网络与强化学习进行融合的算法，Q-learning以表格的形式存储所有状态动作对的动作价值，并在训练过程中不断的更新这些动作值，也就是Q值：，而DQN则是利用神经网络拟合这些Q值，即值网络化，其总体结构如图2-3所示：

图2-3 DQN总体结构

DQN利用类似Q-learning的公式来进行训练，只不过Q-learning是利用公式直接更新的Q值，DQN更新的是神经网络的参数，即公式中更新的变成了神经网络的参数。DQN采用两个神经网络分别拟合公式中的和，假设这两个网络一开始分别被初始化为Q和Q1，这两个神经网络在一开始是一样的。在训练过程中，模型抽取形如的数据进行训练，DQN是离线模型，可以通过过往经验学习。Q1拟合的的参数为目标值，Q的参数会随着训练过程的每一步而不断的改变，去逼近Q1所拟合的目标值，因此DQN神经网络的损失函数被定义为。Q1的参数在一定步长内会固定不变，当达到设定的步长值后，Q1的参数会同步为此时Q的参数，然后继续训练过程。DQN主要采用三个技巧来优化训练过程：

1.探索机制：如果某一个动作取得了正向奖励，那么智能体有可能一直采取这个动作而不去尝试其他动作，但是可能存在其他可以取得更大奖励的动作，解决这一问题可以采用探索机制。常用的探索方法为贪心算法，即每一步以的概率挑选价值最大的动作，以的概率随机选择要执行的动作。

2.固定目标网络：目标网络为上文提到的Q1，该网络为Q网络提供需要拟合的目标值。目标网络也包含Q函数，如果目标一直变化得话会对训练造成困难，将目标网络固定住，经过一定步长后再更新为最新的Q函数。

3.经验回放：在训练过程中会有经验池存放过往的训练数据，存储形式为。强化学习中比较耗时的是环境与智能体的交互过程，将过往的交互过程作为经验存储下来可以节省交互时间，也能提高以往经验的利用率。同时，训练样本来自不同训练步长时的不同策略，增加了数据的多样性。

## Related work

# Methology

## Overall framework

强化学习算法总体架构(如果不用可迁移的代码图里不加距离矩阵这四个字)

在训练过程中，智能体通过动作来改变环境、环境反馈状态和奖励用于智能体学习。使用距离矩阵和孤立森林计算状态，距离矩阵反映待检测点和已知点之间的欧氏距离，孤立森林计算出的得分可以反映各个点的异常程度。通过强化学习的PPO2策略，智能体从一系列动作中选择概率最高的动作。智能体基于奖励更改参数以优化策略，从而能够根据环境反馈的状态选择更好的动作并获得更大的奖励。测试过程的目的是评估模型的质量。智能体只需要根据状态选择动作，不需要更改其参数，因此不需要奖励。但是，由于智能体执行动作，环境会发生变化，因此状态仍会不断更新。在测试过程中，初始的已知正常和异常被初始化为训练集的已知正常或异常。在之后的判断中，通过设置阈值来改变已知点，如果动作概率超过阈值，则将其添加到已知点中。此外，为了提高测试的准确性，可以在早期添加人工标注来替换阈值，类似于主动学习。针对少标签场景，为带标签部分和无标签部分分别设计奖励函数，最后综合两部分奖励实现联合优化。对于带标签部分，环境依据标签反馈严格的正向或者负向惩罚，对于无标签部分，环境通过无监督的形式生成奖励，鼓励智能体对未知部分进行探索。

## Detailed Design

### 状态

状态是对环境的观测量，它必须包含足够的有效信息，以帮助智能体判断并执行正确的动作。为了提高模型整体的可迁移性，状态必须足够通用，而不是针对某一特定数据集设计，时刻t的状态包括四个部分：

首先，在环境初始化期间由孤立森林计算的异常得分是状态之一。第二，已选择过的节点为已知的异常类或正常类，这是一种有效的信息。由此计算待检测点与已知异常点的距离和已知正常点之间的距离。此外，如果待检测点的邻居包含已知异常点，则是帮助智能体做出判断的重要信息，其反映的是待检测点的邻居中是否包含异常点。异常得分越高，该点越有可能是异常，同时，一个点离已知异常越近，已知正常越远，或者已知异常是其邻居，则越有可能是异常。这些不是某一数据集独有的，而是所有数据集都可根据特征计算出的状态值，状态的通用性可以有效提高框架的可迁移性。

### 动作

动作是实现强化学习模型中异常检测目标的一种手段，因此，这里的动作为在最少错误的情况下挑选出异常点。在二分类的情况下，智能体只需要判断待检测点是否为异常。动作0表示该点正常，而动作1表示该点异常。当进行多分类任务时，智能体必须识别异常点并判断其异常类别。此时的动作为优先选择异常点并将其分类到正确的类别中。动作0表示该点正常，而动作n表示该点属于第n类异常。在二分类的基础上，多分类不仅应该确定它是异常还是正常，还应该确定它属于哪种异常。多分类使该模型能够应用于更细粒度的异常检测场景，在实际场景中有各种类型的异常值，有时需要区分异常点是哪种类型的异常。

### 奖励

不同于大多数强化学习模型单一的奖励机制，本文针对带标签部分和无标签部分分别设计了不同的奖励机制。

针对带标签部分，采用有监督的模式给予奖励，奖励的作用是指导智能体采取正确措施以实现设定的目标。这里的目标是引导智能体优先识别异常点，因此，当智能体正确判别出异常点时，它将获得正向的奖励，否则，它将获得负向惩罚。如果智能体判断某一点为正常，并且该点的标签确实为正常，则奖励为0，对智能体不奖励也不惩罚。在二分类中，当一个点被判断为异常，并且其标签确实为异常时，奖励为1，否则为-0.1。负向奖励是对智能体错误行为的惩罚或容忍度，负向奖励越大，对错误的容忍度越小。如果模型在数据集上没有任何错误，则很容易导致过拟合，然而，如果惩罚太小，则对错误的容忍度太高，从而导致欠拟合。根据经验和实验，本文最终选择了-0.1，奖励函数如式3-4所示。

在多分类任务中，正常样本的正确识别获得0的奖励。如果判断为异常，并且模型将其归为正确的类别，则奖励为1，否则为-0.1，如式3-5所示。

针对无标签部分，采取无监督的方式反馈奖励，使用孤立森林算法输出奖励。首先利用智能体神经网络的一部分对待检测点的特征进行处理，神经网络输入为待检测点的特征，输出结果作为孤立森林的输入。这样做的目的是保证孤立森林的输入也随着智能体的参数变化而不断变化，如果直接输入特征，那么在整个训练过程中，孤立森林的输入都不会变化，输出自然也不会变化。孤立森林输出的异常得分做归一化处理作为最终的奖励，奖励函数如式3-6所示。

状态的通用性以及智能体与环境之间的不断交互使得奖励能够有效地引导智能体完成异常检测的学习。模型从状态中学习到环境相关知识，使用已知正常点和异常点计算状态，反映包括正常点和异常点的数据总体分布。因此，该模型从数据的总体分布中学习，而不是仅从异常值分布中学习。奖励机制是引导模型实现优先挑选异常的一种手段。

带标签部分的奖励函数采取严格的奖惩机制，引导智能体去充分学习带标记点的特征，在遇到类似特征时，能够做出正确的判断。无标签部分的奖励函数为异常得分，一个点越可能是异常那么这个点的异常得分也就越高，奖励也就越高，从而鼓励智能体去充分探索无标签部分可能的异常点。结合两种奖励函数，对有标签部分严格学习，无标签部分充分探索，实现少标签场景下的异常检测。

### 策略

策略指导智能体在特定状态下执行正确的动作，这里使用PPO2算法。本文使用数据特征和标签来完成模型训练。智能体通过与环境的持续交互来改变其参数，从而获得正向的奖励，环境也会根据智能体的动作而变化，伪代码如表3-1所示。

表3-1 策略伪代码

|  |
| --- |
| 输入：特征，标签，总训练步长 |
| 输出：模型 |
| 计算初始化环境 |
| 初始化策略，网络参数， |
| 训练过程： |
| 当前训练步数为n，当n达到时训练结束 |
| 如果n=1或者一个回合结束： |
| 从和中随机采样 |
| 根据策略输出动作 |
| 环境输出新的状态 |
| 获得无标签和带标签奖励 |
| 更新参数 |
| 当损失值在100内不再明显变化，例如变化范围小于0.0001： |
| 结束训练 |

首先，使用特征，根据公式2-1，3-1和3-2计算和并获得。然后，初始化策略，它代表智能体的参数。这样，环境和智能体的初始化就完成了。在训练过程中，智能体根据策略执行动作。在接收到动作后，环境更新式3-3中的状态，并基于等式3-4或3-5以及3-6返回奖励。收到来自环境的反馈后，智能体通过式2-5计算损失值，并根据损失函数更新网络参数。当训练步长达到设定的最大步长或损失值在特定步长内没有显著变化时，训练终止。测试过程的不同之处在于不需要初始化策略，而是使用经过训练的模型参数。在测试过程中，模型根据状态连续选择动作，而不基于奖励优化参数。

# Experiments and Results

## Data Set

## Experimental Result

针对少标签场景，训练集中包含少量带标签的异常，少量无标签的异常数据，即未知异常，以及大量无标签的正常数据，模拟现实场景从只有少量数据带标签，存在大量无标签数据的场景，同时与无监督，半监督，有监督方法进行对比。

为了验证提出所提的两种奖励函数在少量带标签，大量无标签数据场景下的可适应性，本文进行了相关实验并与多种无监督，半监督，有监督算法进行对比，实验结果如表3-4所示。数据行的0到8分别对应八种不同的异常，依次为Fuzzers, Analysis, Backdoors, DOS, Exploits, Generic, Reconnaissance, Shellcode和Worms。模型列中Paper为本文方法，Paper\_a和Paper\_u为消融实验，Paper\_a只考虑带标签异常，Paper\_u只考虑不带标签的数据。实验结果表明所提方法充分利用了带标签和无标签数据，并且在少标签场景下有稳定的精度。此处的训练集只包含少量带标签异常，其余数据不论正常还是异常均为无标签数据，带标签数据在训练集中占比不到百分之一。表3-4所示验证了方法的基本性能，训练集只包含一类异常，测试集也只包含了对应那一类的异常。在这样的测试环境下待检测数据集相对于训练集未包含未知异常，可以充分检验方法本身的有效性。由实验结果可知方法在每一类异常上都取得了不错的效果，证明该方法在少标签场景下具备较好的稳定性。同时在少标签场景下，所提方法相较于其他方法取得了更好的效果。可以看到，无监督方法由于缺乏带标签的先验知识，仅通过数据特征进行学习，所获取的知识有限，在所有方法中取得的效果最差。半监督方法大多基于带标签数据扩展标签，即生成伪标签，这样对无标签数据的探索也依赖于有标签部分的精度，具有一定的局限性，生成的伪标签也带有噪声，对精度的提升有限。在无监督，半监督和有监督算法中，有监督在各个异常类上取得的综合效果最好，其充分利用了带标签的先验知识，但是在少标签场景下，能够利用的先验知识有限，这也限制了有监督算法的上限。

表3-4 无未知异常少标签分类结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据  模型 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Paper | 0.96 | 0.93 | 0.96 | 0.82 | 0.94 | 0.93 | 0.97 | 0.95 | 0.96 |
| Paper\_a | 0.91 | 0.84 | 0.73 | 0.75 | 0.86 | 0.86 | 0.97 | 0.75 | 0.93 |
| Paper\_u | 0.61 | 0.56 | 0.65 | 0.68 | 0.61 | 0.51 | 0.81 | 0.70 | 0.85 |
| Isolation Forest | 0.55 | 0.51 | 0.55 | 0.53 | 0.51 | 0.52 | 0.60 | 0.54 | 0.52 |
| OneClassSVM | 0.55 | 0.53 | 0.54 | 0.55 | 0.56 | 0.53 | 0.52 | 0.53 | 0.54 |
| LocalOutlierFactor | 0.55 | 0.68 | 0.67 | 0.51 | 0.56 | 0.56 | 0.60 | 0.57 | 0.62 |
| Random Forest | 0.91 | 0.89 | 0.89 | 0.76 | 0.86 | 0.85 | 0.96 | 0.78 | 0.92 |
| ANN | 0.85 | 0.86 | 0.78 | 0.74 | 0.84 | 0.80 | 0.98 | 0.76 | 0.84 |
| SVM | 0.86 | 0.89 | 0.91 | 0.81 | 0.83 | 0.84 | 0.86 | 0.83 | 0.81 |
| LabelPropagation | 0.81 | 0.89 | 0.92 | 0.79 | 0.83 | 0.78 | 0.79 | 0.85 | 0.80 |
| LabelSpreading | 0.82 | 0.89 | 0.88 | 0.82 | 0.84 | 0.80 | 0.82 | 0.84 | 0.82 |
| SelfTrainingClassifier | 0.79 | 0.88 | 0.89 | 0.73 | 0.78 | 0.61 | 0.78 | 0.80 | 0.70 |

此外，在实际环境中，测试集除了包含训练集中带标签的异常类别，还会包含其他的未知异常类别，因此为了更加充分的验证所提方法在真实场景下的适应性，在训练集不变的前提下向测试集中加入了更多的未知异常类别，实验结果如表3-5所示。由实验结果可知，在加入了更多的未知异常的情况下，所提方法的精度有所下降，但是仍然保持了一个不错的结果，并且在整体效果上优于无监督，半监督和有监督算法。相较于本文所提算法，其余对比算法在遇到未知异常时，如果未知异常与已经学习过的异常相似，则还能保持不错的效果，反之则对其是完全陌生的异常，效果就会很差。所提方法对带标签数据充分学习，在遇到相似未知异常时能正确判断。同时方法对无标签数据进行探索，测试集中存在的未知异常对于智能体也是一种探索，当智能体学会这种探索模式后，在未知异常与带标签数据相差较大时，也能进行一定的探索。综上，即使测试集存在训练集中不包含的未知异常，所提方法依然能够取得不错效果。

表3-5 含未知异常少标签分类结果

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据  模型 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Paper | 0.81 | 0.83 | 0.86 | 0.85 | 0.71 | 0.84 | 0.66 | 0.87 | 0.72 |
| Paper\_a | 0.76 | 0.79 | 0.80 | 0.67 | 0.70 | 0.81 | 0.61 | 0.71 | 0.73 |
| Paper\_u | 0.60 | 0.56 | 0.72 | 0.56 | 0.52 | 0.54 | 0.74 | 0.60 | 0.60 |
| Isolation Forest | 0.52 | 0.53 | 0.52 | 0.55 | 0.50 | 0.51 | 0.51 | 0.51 | 0.52 |
| OneClassSVM | 0.55 | 0.54 | 0.54 | 0.55 | 0.55 | 0.53 | 0.54 | 0.55 | 0.54 |
| LocalOutlierFactor | 0.58 | 0.56 | 0.55 | 0.59 | 0.61 | 0.58 | 0.56 | 0.51 | 0.57 |
| Random Forest | 0.79 | 0.77 | 0.79 | 0.72 | 0.77 | 0.80 | 0.59 | 0.72 | 0.58 |
| ANN | 0.74 | 0.77 | 0.73 | 0.72 | 0.67 | 0.73 | 0.61 | 0.70 | 0.62 |
| SVM | 0.77 | 0.75 | 0.79 | 0.75 | 0.68 | 0.78 | 0.56 | 0.72 | 0.58 |
| LabelPropagation | 0.77 | 0.78 | 0.78 | 0.73 | 0.72 | 0.68 | 0.54 | 0.72 | 0.69 |
| LabelSpreading | 0.78 | 0.77 | 0.78 | 0.76 | 0.70 | 0.72 | 0.58 | 0.70 | 0.69 |
| SelfTrainingClassifier | 0.71 | 0.71 | 0.71 | 0.70 | 0.70 | 0.71 | 0.53 | 0.71 | 0.51 |

同时本文对所提方法进行了消融实验，分别测试了方法在只考虑带标签数据和只考虑无标签数据下的检测效果，即将Paper与Paper\_a，Paper\_u对比。可以看到只考虑带标签数据时效果略有下降，在少标签场景下模型可以获得的带标签先验知识有限，失去了对无标签数据的探索，所以精度会下降。在只考虑无标签数据的场景下，缺乏带标签数据的引导，仅靠无监督的探索，效果很差。由消融实验可知，针对带标签和无标签数据分别设计奖励函数是有效的。

综上，所提方法在少标签场景下通过两种不同的奖励函数充分利用了带标签数据和无标签数据。针对带标签数据严格的奖励机制使得模型充分学习了带标签的先验知识。针对无标签数据的无监督奖励则鼓励模型去探索无标签数据中的未知异常。所提方法对带标签的学习和无标签的探索进行了折中，可以适应少标签的场景。

# Conclusion

# 参考文献