分布式协作频谱感知网络中恶意节点检测和定位方法研究

吴晓晓[1,2],李刚强[3],张胜利[1,2]
(1. 深圳大学电子与信息工程学院,广东深圳 518060; 2. 鹏城实验室,广东深圳 518055;
3. 黄淮学院信息工程学院,河南驻马店 463000)

摘 要: 认知无线电是解决无线通信能量有效性问题的关键技术,其中频谱感知对于提高频谱的利用效率有着重要意义. 针对基于共识的分布式协作频谱感知算法易受到恶意节点数据注入攻击,影响认知网络性能的问题,本文提出了两种基于神经网络的恶意节点检测和定位方法抵制网络内的恶意攻击行为,并采用基于Gossip Learning的联合学习策略进一步增强训练邻域检测和定位模型的鲁棒性. 本文在9个认知节点的曼哈顿网络上模拟了分布式频谱感知的过程,并验证所提出方法的有效性. 结果表明,所提方法具有良好的恶意节点检测和定位性能,联合学习策略能够使神经网络在样本局部有限的情况下学习到更多的攻击特征,提高本地检测和定位模型的可靠性.

关键词: 认知无线电; 协作频谱感知; 共识算法; 恶意节点; 神经网络; 联合学习
中图分类号: TN92 **文献标识码:** A **文章编号:** 0372-2112(2022)06-1370-11
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20210841

Detection and Localization of Malicious Nodes in Distributed Cooperative Spectrum Sensing Network

WU Xiao-xiao[1,2], LI Gang-qiang[3], ZHANG Sheng-li[1,2]
(1. *College of Electronics and Information Engineering, Shenzhen University, Shenzhen, Guangdong* 518060, *China*;
2. *Peng Cheng Laboratory, Shenzhen, Guangdong* 518055, *China*;
3. *College of Information Engineering, Huanghuai University, Zhumadian, Henan* 463000, *China*)

Abstract: Cognitive radio is a key technology to solve the problem of energy efficiency in wireless communication, and spectrum sensing is of great significance for improving the efficiency of spectrum utilization. To solve the problem that the consensus-based distributed cooperative spectrum sensing algorithm is vulnerable to malicious node data injection attacks, we propose two approaches for detecting and localizing malicious nodes based on neural networks. And a collaborative peer-to-peer machine learning protocol(Gossip Learning) is adopted to facilitate training these neural network models. We simulate the process of distributed cooperative spectrum sensing on a 9-node Manhattan network, and verify the effectiveness of the proposed approaches. Numerical results illustrate that the proposed neural network-based approaches can effectively improve the performance of detecting and localizing malicious nodes. The collaborative learning strategy can enable nodes to learn more attack characteristics, and thus make the network more robust to attacks.

Key words: cognitive radio; cooperative spectrum sensing; consensus algorithm; malicious node; neural network; collaborative learning

1 引言

频谱资源是有限的,在含有主要用户(Primary User,PU)和认知用户(Secondary User,SU)的无线通信网络中,提高频谱的利用率对于通信技术的发展具有重要的意义[1]. 传统固定分配模式下,PU节点的"独占"现象使得授权频段处于空闲模式时无法被网络中的SU用户有效利用,导致频谱利用效率低下. 因此,需要对现有的固定频谱分配策略,实施动态频谱管理[2]. 认知

收稿日期:2021-07-05;修回日期:2022-01-31;责任编辑:冯珂
基金项目:国家自然科学基金(No.62171291);广东省自然科学基金面上项目(No.2021A1515011915);河南省科技攻关项目(No.212102210142)

无线电(Cognitive Radio,CR)[3]由Joseph Mitola等于1999年提出,SU节点通过实时监测无线环境,利用感知结果在网络中可以自适应地检测频谱,实现自动进行频谱管理以及提高频谱利用率的目的.CR过程包含频谱感知、频谱管理和频谱共享3个主要阶段,其中频谱感知阶段是实现认知过程的关键[4].相较于单用户的独立感知,SU节点之间的协作频谱感知,能够通过信息交互克服多径效应、阴影衰落和隐藏终端等问题.

协作频谱感知根据信息共享方式的不同,可以简单分为集中式、中继式和分布式3种.其中,不同于依赖数据融合中心的集中式和中继式协作方法,分布式协作方法下的SU节点仅需要点对点通信的方式进行数据交换[5],大大提高了感知系统的自由度和鲁棒性.然而,分布式协作感知的扁平式结构增加了频谱感知的危险性,面临着恶意节点数据注入攻击的威胁,后者可能导致SU节点收敛至错误的感知结果,做出错误的决策,从而影响整体认知网络的性能.因此,针对分布式协作认知网络中存在的恶意节点的检测和定位尤为重要.

一般来讲,恶意节点对协作感知过程的攻击取决于特定的攻击方式,常见的攻击有主用户伪造攻击(Primary User Emulation,PUE)和拜占庭攻击(Spectrum Sensing Data Falsification,SSDF)[4].本文中考虑的攻击方式是拜占庭攻击的自然延伸,主要涉及能量检测值的协作感知,其中数据注入攻击的影响可以通过测量的误差值来衡量.本文对于基于共识的频谱感知算法脆弱性的见解在于认识到共识算法在一定条件下类似于DeGroot动态意见模型[6],恶意节点通过向其邻居节点发送包含恒定偏差的消息,明智地调整预期的收敛速度及加入随机噪声,来达到扰乱网络收敛的目的[7~9].文献[9,10]证明,网络总是收敛到等于偏置的最终状态,这将给基于共识的频谱感知算法带来严重的安全问题.因此,需要一种良好的恶意节点检测和定位方法来保护这些算法免受数据注入攻击.

在实际的应用中,一个好的攻击检测方法需要对问题进行不同的设置,并且通过观察多个独立的认知过程来确定网络中恶意节点的存在并进行定位.文献[8]中讨论了基于平均共识算法下的数据注入攻击场景,提出了一个时间差分的检测方法用于检测和定位恶意节点,该方法依赖攻击目标和最优收敛目标之间的差异.文献[9,11]提出了2种统计分数的方法用于恶意节点的检测和定位,通过多次运行算法来观察其邻居节点的异常.文献[10]讨论了投影次梯度算法下的内部攻击检测方法,设计了本地的度量函数,通过观察邻居节点的状态差分来统计各节点的得分,以检测和定位恶意节点.以上文献方法可以概括为分数方案,虽然具有合理的性能,但严重依赖专家来设计复杂的决策函数,存在论证复杂、阈值设置难度大、模型无学习能力以及无法有效给出网络出现攻击行为(或某个节点为恶意节点)的程度等问题.

针对此类问题,本文利用神经网络模型的自动学习能力实现复杂非线性函数的拟合,映射输出连续的概率来表示恶意的程度,从而更好地来检测和定位恶意节点,最终实现将恶意节点全部踢出分布式认知网络的目的.具体地,本文提出了2种基于神经网络的方法:(1)基于神经网络的时间差分方法,利用分布式网络节点收敛状态在攻击前后的不一致来检测网络的异常;(2)基于神经网络的空间差分方法,考虑了共识迭代过程中的过渡状态,通过计算认知节点与邻域节点在时间上的差分累积来检测和定位恶意节点.在此基础上,本文进而提出一种基于Gossip Learning的联合学习策略促进神经网络模型的训练,使节点在本地局部数据的基础之上能够学习到与全局模型接近的局部模型,以此缓解网络中部分节点存在训练数据不足和邻域攻击数据分布不一致的情形.最后,本文在9节点的曼哈顿网络上模拟了基于一致性共识分布式协作感知过程,并采集用于神经网络训练和测试所需的样本,仿真验证了所提方法的有效性.

2　分布式协作频谱感知与攻击模型

2.1　基于共识的频谱感知算法

分布式对等交互的频谱感知模型中,每个SU节点可以看作独立的融合中心,通过与邻居节点的信息交互做出PU节点是否占用频谱的判决.考虑一个由N个SU节点组成的认知网络$\mathcal{G}=(\mathcal{V},\mathcal{E})$,其中$V=\{1,2,\cdots,N\}$表示网络中SU节点的集合,如图1所示.$\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$表示网络中SU节点边的集合,节点$(i,j)\in\mathcal{E}$则表示网络中节点$i$和节点$j$可以直接通信,共享感知信息.定义$\mathcal{N}_i\in\mathcal{V}$表示节点$i$的邻居集合,即$\mathcal{N}_i=\{j\in\mathcal{V}:(i,j)\in\mathcal{E}\},\ \forall i\in\mathcal{V}$.

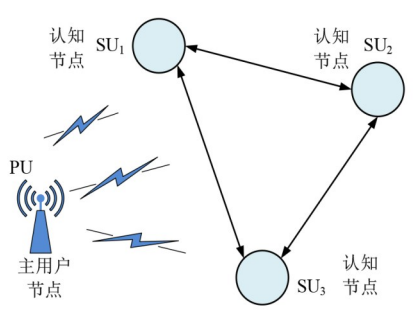


图1　分布式对等交互频谱感知模型

任意节点$i\in\mathcal{V}$独立运行能量检测算法得到关于PU节点的能量感知值$x_i(0)$,然后与邻居节点交互迭代直至所有参与协作的节点能量值趋于一致.其中,一致

性共识融合过程可以使用如下迭代更新规则来描述[12]：

$$x_i(t+1)=x_i(t)+\rho\sum_{j\in\mathcal{N}_i}\left(x_j(t)-x_i(t)\right) \tag{1}$$

其中，ρ 为迭代步长，满足 $0<\rho<1/\max\left(\left|\mathcal{N}_i\right|\right)$. $x_i(t)$ 和 $x_j(t)$ 分别为节点 i 和 j 在 t 时刻的状态值，若干次迭代更新后，整个认知网络将会收敛至所有节点的状态均值 $x_{\mathrm{av}}=x_i(\infty)=\frac{1}{N}\sum_{i\in\mathcal{V}}x_i(0)$. 图2给出了含有9个SU节点时分布式对等交互的频谱感知的数值模拟，使用均匀分布近似能量感知值的测量范围. 可以看到，各认知节点能够达成共识并收敛到状态均值 x_{av}.

事实上，以上的认知过程可以认为SU节点在共识过程中根据对邻居节点的状态加权达成最终的共识状态. 则式(1)可以写成如下形式[13]：

$$x_i(t+1)=\sum_{j\in\left\{\mathcal{N}_i\cup i\right\}}A_{ij}(t)x_j(t),\ t\in\{1,2,\cdots,T\} \tag{2}$$

其中，$A_{ij}(t)$ 为时间 t 上节点 i 分配给其邻居节点 j 的聚合权重. 将网络内SU节点的状态写成向量形式，令 $\boldsymbol{x}(t)=\left[x_1(t),x_2(t),\cdots,x_N(t)\right]\in\mathbb{R}^N$，式(2)可写成以下形式：

$$\boldsymbol{x}(t+1)=\boldsymbol{A}(t)\boldsymbol{x}(t),\ t\in\{1,2,\cdots,T\} \tag{3}$$

其中，$A_{ij}(t)=\left[\boldsymbol{A}(t)\right]_{ij}$ 表示权值矩阵中的第 (i,j) 个元素，$\boldsymbol{A}\in\mathbb{R}^{N\times N}$ 是一个 $N\times N$ 的对称矩阵. 通常情况下矩阵 $\boldsymbol{A}$ 是一个双随机矩阵，满足 $\mathbf{1}^{\mathrm{T}}\boldsymbol{A}(t)=\mathbf{1}^{\mathrm{T}},\boldsymbol{A}(t)\mathbf{1}=\mathbf{1},\boldsymbol{A}^{\mathrm{T}}(t)=\boldsymbol{A}(t)$，即每一行和列分别满足和为1.

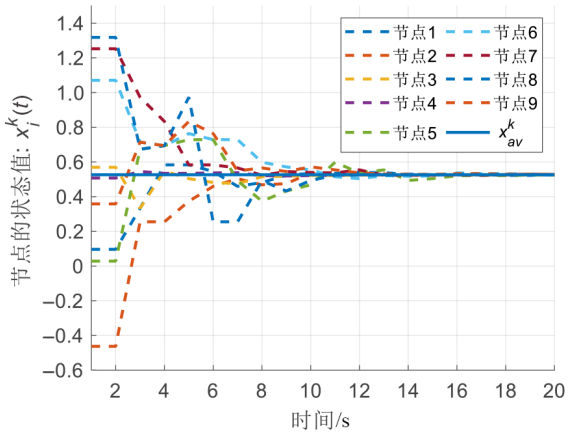


图2 分布式频谱感知过程中各SU节点的感知状态演化

引入概率矩阵 $\boldsymbol{P}\in\mathbb{R}^{N\times N}$，定义 $\left[\boldsymbol{P}\right]_{ij}=P_{ij}\geqslant 0$ 表示节点 i 选择与邻居节点 j 信息交换的概率. 则对于式(3)中的矩阵 $\boldsymbol{A}(t)$，其算法迭代过程中的期望形式如下：

$$\begin{aligned}\bar{\boldsymbol{A}}&=\mathbb{E}\left[\boldsymbol{A}(t)\right]\\&=\frac{1}{N}\sum_{i=1}^{N}P_{ij}A_{ij}\\&=\boldsymbol{I}-\frac{1}{2N}\boldsymbol{\Lambda}+\frac{\boldsymbol{P}+\boldsymbol{P}^{\mathrm{T}}}{2N}\end{aligned} \tag{4}$$

其中，$\boldsymbol{\Lambda}=\mathrm{diag}\left(\left[\Lambda_1,\Lambda_2,\cdots,\Lambda_N\right]\right)$ 是一个对角矩阵，对角元素 $\Lambda_i=\sum_{j=1}^{N}\left(P_{ij}+P_{ji}\right)$. 则式(4)中SU节点的期望状态可以表示为

$$\mathbb{E}\left[\boldsymbol{x}(t)|\boldsymbol{x}(0)\right]=\bar{\boldsymbol{A}}\mathbb{E}\left[\boldsymbol{x}(t-1)|\boldsymbol{x}(0)\right]=\bar{\boldsymbol{A}}^t\boldsymbol{x}(0) \tag{5}$$

其中，$\bar{\boldsymbol{A}}^t$ 表示 t 时刻的期望矩阵，$\bar{\boldsymbol{A}}$ 也是双随机矩阵且是对称的. 共识算法的收敛性在文献[14,15]中得到了有效分析，分布式协作的认知节点会收敛至相同的能量感知值. 对于有向的认知网络，各节点可以通过Push-Sum算法执行共识变量的更新[16].

2.2 恶意节点协同攻击方案

假设网络 $\mathcal{G}=(\mathcal{V},\mathcal{E})$ 中存在一些协同恶意的SU节点 $\mathcal{V}_s\subseteq\mathcal{V}$，恶意节点 $\mathcal{V}_s=\left\{1,2,\cdots,N_s\right\}$ 的数量远小于网络节点的数量 $N_s<N$. 正常的SU节点集合为 $\mathcal{V}_r=\mathcal{V}\backslash\mathcal{V}_s$，其个数为 $N_r=N-N_s$. 该攻击方案中，恶意节点的状态值不会受到正常节点的影响，且不会遵循协作感知的共识协议. 其攻击方案如下：

$$x_j(t)=\alpha+\beta_j(t),\ j\in\mathcal{V}_s \tag{6}$$

其中，$\beta_j(t)$ 为指数衰减的零均值噪声；α 为恶意节点的攻击目标值，在多个独立的认知过程中遵循一个特定的分布(本文假设恶意节点的攻击目标值分布 $\bar{\alpha}$ 与正常SU节点的初始能量感知值分布 $\bar{x}_i(0)$ 不同，即满足 $\bar{\alpha}\neq\bar{x}_i(0),\ i\in\mathcal{V}_r$). 假设其分布参数作为一个先验信息被SU节点群所获知，在此前提下，如果恶意节点保持它们的状态一直不变即 $x_j(t)=x_j(0),\ j\in\mathcal{V}_s$，其攻击行为会很容易地被正常的邻居发现，而在修改后的规则下恶意节点能够模拟正常节点的收敛行为，隐藏在网络中.

当 $N_s\neq 0$ 时，即网络存在恶意节点 $s\in\mathcal{V}_s$. 网络内各节点 t 时刻的能量感知状态可以重新写为

$$\boldsymbol{x}(t)=\left(\boldsymbol{x}_s(t)^{\mathrm{T}},\boldsymbol{x}_r(t)^{\mathrm{T}}\right)^{\mathrm{T}},\ s\in\mathcal{V}_s,r\in\mathcal{V}_r \tag{7}$$

其中，$\boldsymbol{x}_s(t)\in\mathbb{R}^{N_s}$ 和 $\boldsymbol{x}_r(t)\in\mathbb{R}^{N_r}$ 分别为恶意节点和正常节点在 t 时刻的状态. 在式(6)和式(3)条件下，如果 $s\in\mathcal{V}_s$，则 $A_{ss}(t)=1$；若 $s\in\mathcal{V}_s,r\in\mathcal{V}_r$，则 $A_{sr}(t)=0$. 式(4)中期望矩阵 $\bar{\boldsymbol{A}}$ 可以表示为

$$\bar{\boldsymbol{A}}=\begin{bmatrix}\boldsymbol{I}&\boldsymbol{0}\\\bar{\boldsymbol{B}}&\bar{\boldsymbol{D}}\end{bmatrix} \tag{8}$$

其中，$\bar{\boldsymbol{B}}\in\mathbb{R}^{N_r\times N_s}$ 和 $\bar{\boldsymbol{D}}\in\mathbb{R}^{N_r\times N_r}$ 分别为恶意节点与正常节点及正常节点与正常节点之间的子矩阵. 从 $\bar{\boldsymbol{A}}$ 矩阵可以看出，恶意节点不会受到正常节点的影响. 子矩阵 $\bar{\boldsymbol{D}}$

是次随机的,$\|\bar{\boldsymbol{D}}\|_2<1$成立. 如果恶意节点没有被网络隔离,则$\bar{\boldsymbol{B}}\neq 0$. 进一步地,随着时间$t$上的迭代,期望矩阵$\bar{\boldsymbol{A}}^t$可写为

$$\bar{\boldsymbol{A}}^t=\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \bar{\boldsymbol{B}} & \bar{\boldsymbol{D}}\end{bmatrix}^t=\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \left(\sum_{\tau=0}^{t-1}\bar{\boldsymbol{D}}^\tau\right)\bar{\boldsymbol{B}} & \bar{\boldsymbol{D}}^t\end{bmatrix} \tag{9}$$

则对于网络的正常节点$r\in\mathcal{V}_r$,提取相关权值$\begin{bmatrix}\left(\sum_{\tau=0}^{t-1}\bar{\boldsymbol{D}}^\tau\right)\bar{\boldsymbol{B}} & \bar{\boldsymbol{D}}^t\end{bmatrix}$,其状态随着时间$t$变化为

$$\begin{aligned}&\lim_{t\to\infty}\mathbb{E}\left[\boldsymbol{x}_r(t)|\boldsymbol{x}_r(0),\alpha\right]\\&=\begin{bmatrix}\left(\sum_{\tau=0}^{t-1}\bar{\boldsymbol{D}}^\tau\right)\bar{\boldsymbol{B}} & \bar{\boldsymbol{D}}^t\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_s^{\mathrm{T}}(t)\\ \boldsymbol{x}_r^{\mathrm{T}}(t)\end{bmatrix}\\&=\alpha\left(\sum_{\tau=0}^{t-1}\bar{\boldsymbol{D}}^\tau\right)\bar{\boldsymbol{B}}+\bar{\boldsymbol{D}}^t\boldsymbol{x}_r(0)\\&=\alpha\left(1-\bar{\boldsymbol{D}}^t1\right)+\bar{\boldsymbol{D}}^t\boldsymbol{x}_r(0)\end{aligned} \tag{10}$$

其中,在$\|\bar{\boldsymbol{D}}\|_2<1$成立时$\lim_{t\to\infty}\bar{\boldsymbol{D}}^t=0$. 随着时间$t$的增加,$\left(1-\bar{\boldsymbol{D}}^t1\right)=1$,$\bar{\boldsymbol{D}}^t\boldsymbol{x}_r(0)$逐渐趋近于0. 在$t\to\infty$时,网络内的正常节点会收敛到恶意节点的目标值$\alpha$. 图3显示了恶意协同攻击下认知节点的状态变化图,黑色实线为恶意节点,虚线为正常节点的状态随时间t的变化,蓝色实线为网络的最优收敛感知值x_{av}.

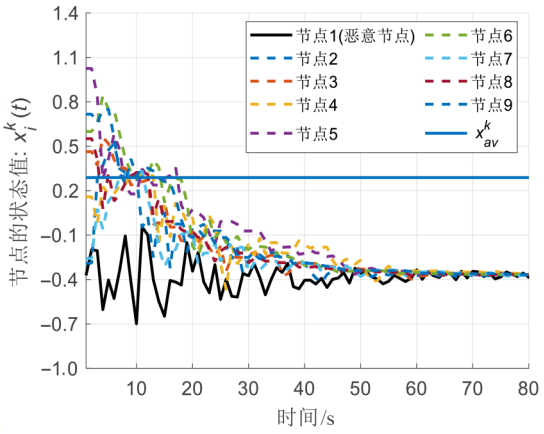


图3　恶意协同攻击下各SU节点的感知状态演化

3　面向恶意用户的检测和定位方案

3.1　面向恶意节点的邻域检测和定位任务

本文针对频谱感知过程中的恶意节点检测和定位问题,定义以下2个本地任务,正常节点$i\in\mathcal{V}_r$以分布式的方式独立地执行,具体如下.

(1)邻域检测任务-正常节点邻域内是否存在恶意节点:

$$\begin{cases}\mathcal{H}_0^i\colon\ \mathcal{N}_i\cap\mathcal{V}_s=\varnothing,\ \text{邻域内无恶意节点}\\ \mathcal{H}_1^i\colon\ \mathcal{N}_i\cap\mathcal{V}_s\neq\varnothing,\ \text{邻域内含有恶意节点}\end{cases} \tag{11}$$

(2)邻域定位任务-正常节点检查邻居节点是否为恶意节点:

$$\begin{cases}\mathcal{H}_0^{ij}\colon\ j\notin\mathcal{V}_s,\ \text{邻居节点}j\text{不是恶意节点},j\in\mathcal{N}_i\\ \mathcal{H}_1^{ij}\colon\ j\in\mathcal{V}_s,\ \text{邻居节点}j\text{是恶意节点},j\in\mathcal{N}_i\end{cases} \tag{12}$$

第一个任务是邻域检测任务,即正常节点检测邻域节点集合$\mathcal{N}_i$内是否存在恶意节点,式(11)中$\mathcal{H}_0^i$和$\mathcal{H}_1^i$分别为任务中的2个事件. 当$\mathcal{H}_1^i$为真时,即节点$i\in\mathcal{V}_r$检测到邻域内存在恶意节点. 然后节点会执行式(12)中的邻域定位任务,以检查邻居节点$j\in\mathcal{N}_i$是否属于恶意节点.

假设节点$i\in\mathcal{V}_r$考虑了K个实例的节点状态(并行或顺序运行),其中$k\in\{1,2,\cdots,K\}$节点i的状态可以表示为x_i^k,$x_j^k,j\in\mathcal{N}_i$为邻居节点的状态. 令$\tilde{\boldsymbol{x}}_i^k=\left[x_i^k,x_1^k,\cdots,x_j^k,\cdots,x_{|\mathcal{N}_i|}^k\right]^{\mathrm{T}}$,$j\in\mathcal{N}_i$表示节点自身获取的全部状态信息,如文献[8,9]提出了用于恶意节点检测和定位的时间差分(Temporal Difference,TD)和空间差分(Spatial Difference,SD)方法. 认知网络的异常可以根据历史的状态进行统计并设计相关统计函数进行检测.

3.2　基于神经网络的恶意用户检测和定位方法

TD和SD方法将节点$i\in\mathcal{V}_r$获得的状态向量$\tilde{\boldsymbol{x}}_i^k$融合成一个标量分数来检测和定位网络中的恶意节点. 而函数拟合是神经网络的一种自然应用,它将多层神经元连接起来,以完成复杂的功能. 本文将恶意节点检测和定位过程看作对节点$i\in\mathcal{V}_r$观察到的状态向量的分类问题,使用神经网络检测和定位网络内的恶意节点. 为方便表述,令$|\mathcal{N}_i|=M$表示神经网络模型的特征输入维度.

3.2.1　基于神经网络的时间差分方法

本节提出了一种基于神经网络的时间差分方法(Neural Network-based Time Difference,TDNN)来检测和定位网络中存在的恶意节点. 该方法的主要洞见来自恶意节点和正常节点的初始状态分布不同即$\bar{x}_i(0)\neq\bar{x}_j(0),\ i\in\mathcal{V}_r,j\in\mathcal{V}_s$,而在$t\to\infty$时整个分布式网络的收敛结果将被恶意节点引导至$\mathbb{E}\left[x_i^k(\infty)\right]=\alpha=\mathbb{E}\left[x_j^k(\infty)\right],\ i\in\mathcal{V}_r,j\in\mathcal{V}_s$. 这意味着,可以将节点之间的初始状态和收敛状态之间的差异作为神经网络的输入特征. 对于任意正常节点$i\in\mathcal{V}$,可以计算以下差分指标(实际上,每个节点独立地计算$\Delta_j^k(t+1)\triangleq x_j^k(t+1)-x_j^k(t)$,然后在时间$t$上求和得到$\zeta_{ij}$):

$$\zeta_{ij}=\frac{1}{K}\sum_{k=1}^{K}\left(x_j^k(T)-x_j^k(0)\right),\quad j\in\mathcal{N}_i \tag{13}$$

其中，$x_j^k(T)$和$x_j^k(0)$分别为节点的收敛状态值和初始状态值，其中$T<\infty$是一个足够大的值以确保网络收敛达成共识. 直觉地，在正常节点执行相关任务时其邻居节点$j\in\mathcal{V}_s$为恶意节点时，初始状态和收敛状态的差分值$\left|x_j^k(T)-x_j^k(0)\right|$接近于0，若差分值过大则显示有异常.

对于正常节点处的邻域检测和定位任务，神经网络的模型结构如图4所示. 该神经网络模型的输入为M维的向量. 根据式(13)中的度量指标，邻域检测和定位任务的输入如下：

$$\boldsymbol{a}^0=\hat{\boldsymbol{a}}^0=\left[\zeta_{i1},\zeta_{i2},\cdots,\zeta_{iM}\right]^{\mathrm{T}} \tag{14}$$

其中，$\boldsymbol{a}^0$和$\hat{\boldsymbol{a}}^0$分别为邻域检测和定位神经网络的输入特征向量. 对于邻域内的检测任务，输出y_i是一个标量，表示恶意节点是否出现在正常节点i的邻域内. 这可以视为简单的二分类问题，即如果邻域内含有恶意节点，则$y_i=1$，否则输出为$y_i=0$. 基于神经网络的检测任务对应数学式表示如下：

$$\boldsymbol{a}^h=f\left(\boldsymbol{W}^h\boldsymbol{a}^{h-1}+\boldsymbol{b}^h\right),\ h=1,2,\cdots,n-1 \tag{15}$$

$$y_i=g\left(\boldsymbol{W}^n\boldsymbol{a}^{n-1}+\boldsymbol{b}^n\right),\ y_i\underset{\mathcal{H}_0^i}{\overset{\mathcal{H}_1^i}{\gtrless}}\delta_{\mathrm{NN}} \tag{16}$$

其中，$\boldsymbol{W}^h\in\mathbb{R}^{L_h\times L_{h-1}}$为第$h-1$层和第$h$层之间的权值参数；$\boldsymbol{b}^h\in\mathbb{R}^{L_h}$为第$h$层的偏置向量；$\boldsymbol{a}^h$表示第$h$层神经单元的激活输出值；$L_h$为第$h$层含有的神经元个数，其中$L_0=M$，$L_n=1$；$f(\cdot)$为隐含层的激活函数，常用ReLU；输出层的激活函数为$g(\cdot)=1/(1+\mathrm{e}^{-x})$，能够将检测模型的输出映射到区间$[0,1]$，可用于表示事件发生的概率；$y_i$表示神经网络的期望输出；$\delta_{\mathrm{NN}}$为判断邻域任务中事件$\mathcal{H}_1^i$和$\mathcal{H}_0^i$发生的阈值，通常情况下可以设定$\delta_{\mathrm{NN}}=0.5$.

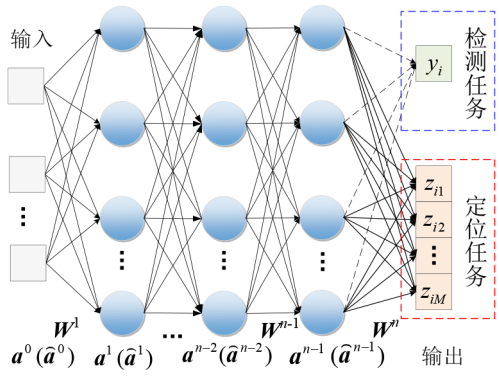


图4 基于神经网络的恶意节点检测和定位模型结构

当节点i在其邻域内检测到恶意节点存在($\mathcal{H}_1^i$为真)，则下一步是找到恶意节点在邻域内的位置. 为此，本文采用了与检测任务类似的神经网络结构执行定位任务，如图4所示. 定位神经网络的输入变量与检测模型相同，即输入维度也是M，而输出的大小等于邻居的个数M. 在这种情况下，可以采用类似于One-Hot编码的形式进行编码[17]，即当某个邻居为恶意节点时，则对应位置为1，否则为0. 例如，当对应于ζ_{i2}的邻居是攻击者时，则应训练该模型以生成输出$\boldsymbol{z}=\boldsymbol{e}_2$. 在数学上，邻域的定位模型可以描述为

$$\hat{\boldsymbol{a}}^h=f\left(\hat{\boldsymbol{W}}^h\hat{\boldsymbol{a}}^{h-1}+\hat{\boldsymbol{b}}^h\right),\ h=1,2,\cdots,n-1 \tag{17}$$

$$z_{ij}=g\left(\hat{\boldsymbol{W}}^n\hat{\boldsymbol{a}}^{n-1}+\hat{\boldsymbol{b}}^n\right),\ z_{ij}\underset{\mathcal{H}_0^{ij}}{\overset{\mathcal{H}_1^{ij}}{\gtrless}}\epsilon_{\mathrm{NN}} \tag{18}$$

其中，$\hat{\boldsymbol{W}}^h$，$\hat{\boldsymbol{a}}^h$和$\hat{\boldsymbol{b}}^h$分别是权重矩阵、隐藏层输出的激活状态和偏置向量；$\boldsymbol{z}_i\in\mathbb{R}^M$为定位神经网络的期望输出，其中$\boldsymbol{z}_i=\left\{z_{i1},z_{i2},\cdots,z_{iM}\right\}$；$\epsilon_{\mathrm{NN}}>0$是邻域定位任务的预定阈值. 在邻域检测和定位任务中，本文考虑的是回归问题，损失函数为真实标签与预测标签的平方误差之和，然后采用随机梯度下降进行神经网络参数学习. 在给定已标记的训练集，随机初始化参数$\boldsymbol{W}^h$和$\boldsymbol{b}^h$，然后通过反向传播方法将训练集上的回归误差最小化进行优化[18,19]. 对于定位任务下的神经网络模型，参数$\hat{\boldsymbol{W}}^h$和$\hat{\boldsymbol{b}}^h$以类似的方式进行初始化和更新.

3.2.2 基于神经网络的空间差分方法

TDNN仅考虑了邻居节点的初始状态和稳定状态. 如果能够充分地利用算法运行中的过渡状态信息，那么它有望探索更多的动态特性，从而进一步提高恶意节点的检测和定位性能. 因此，本文利用基于神经网络的空间差分(Neural Network-based Spatial Difference，SDNN)方法，通过观察邻居节点在时间上的差分累积获取用于神经网络的输入特征，模型结构与图4类似.

式(6)下，恶意节点一直尝试引导正常的邻居节点偏离认知网络的真实收敛值，随着时间t的增加，网络中正常节点的状态逐渐收敛至恶意节点的期望值α. 本小节对SDNN方法的洞见来自TDNN方法中ζ_{ij}的计算，式(13)重新写为

$$\zeta_{ij}=\frac{1}{K}\sum_{k=1}^{K}\left(\underbrace{x_j^k(T)-x_j^k(T-1)}+\cdots+\underbrace{x_j^k(1)-x_j^k(0)}\right),\ j\in\mathcal{N}_i \tag{19}$$

其中，过渡状态随着$t\to T$在计算中被抵消，未得到充分利用. 式(13)中考虑了节点之间的初始状态和收敛状态之间的差异，则可以考虑更多的过渡状态信息，计算过渡状态在时间t上的累积绝对差分来统计节点在不同事件$\mathcal{H}_1^i$和$\mathcal{H}_0^i$下的异常，生成SDNN的输入特征. 因此，定义以下度量用于邻域的检测任务：

$$s_{ij}^k(t)=\left|x_j^k(t)-x_j^k(t-1)\right|,\ j\in\mathcal{N}_i \tag{20}$$

$$\chi_{ij}=\frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{T}s_{ij}^{k}(t),\ j\in\mathcal{N}_i \tag{21}$$

其中,$x_j^k(t)$和$x_j^k(t-1)$分别为节点j在t时刻和$t-1$时刻的状态值.χ_{ij}为节点j在时间$t\in\{1,\cdots,T\}$和$k\in\{1,\cdots,K\}$上的特征值;则检测神经网络的输入特征为$\boldsymbol{a}^0=\left[\chi_{i1},\chi_{i2},\cdots,\chi_{iM}\right]^{\mathrm{T}},\ \forall i\in\mathcal{V}_r$.

在定位任务中,其主要目标就是准确地识别出恶意节点的位置,即区分出哪一个邻居是恶意节点.网络在事件$\mathcal{H}_1^i$下正常节点会收敛到恶意节点的目标值,即$\mathbb{E}\left[x_i^k(\infty)\right]=\alpha=\mathbb{E}\left[x_j^k(\infty)\right],\ i\in\mathcal{V}_r,j\in\mathcal{V}_s$.假设存在$T<\infty$使得共识算法收敛,在$0<t<T$的时间段内网络内的正常节点的行为总是与恶意节点表现不同,即

$$\mathbb{E}\left[x_i^k(t)-x_j^k(t)|\mathcal{H}_1^{ij}\right]\neq\mathbb{E}\left[x_i^k(t)-x_j^k(t)|\mathcal{H}_0^{ij}\right] \tag{22}$$

考虑类似于式(19)对于检测任务的洞见,通过观察节点i和其邻居节点j在时间t上的绝对差分累积来提取过渡状态信息包含的特征.因此,定义以下度量用于邻域的定位任务:

$$\hat{s}_{ij}^{k}(t)=\left|x_i^k(t)-x_j^k(t-1)\right|,\ j\in\mathcal{N}_i \tag{23}$$

$$\hat{\chi}_{ij}=\frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{T}\hat{s}_{ij}^{k}(t),\ j\in\mathcal{N}_i \tag{24}$$

其中,$\hat{s}_{ij}^k(t)$为节点i与邻居节点$j\in\mathcal{N}_i$在t时刻的差分的绝对值;$\hat{\chi}_{ij}$为节点i提取的对应于其邻居节点j的特征值,通过对邻居节点状态与观测节点i处的状态差分比较,有利于准确识别潜在的内部攻击者.则邻域定位神经网络的输入特征为$\hat{\boldsymbol{a}}^0=\left[\hat{\chi}_{i1},\hat{\chi}_{i2},\cdots,\hat{\chi}_{iM}\right]^{\mathrm{T}},\ \forall i\in\mathcal{V}_r$.

3.3　面向神经网络模型的分布式联合学习策略

前面主要介绍了如何使用神经网络来检测和定位网络中的恶意节点.通常训练数据$\tilde{\boldsymbol{x}}_i^k$由正常节点$i\in\mathcal{V}_r$进行收集,但实际上认知网络中的节点由于存储和通信带宽的限制,现实中很难将数据从分布式的节点收集到融合中心.在恶意节点存在的场景下,对于正常节点而言其收集到的攻击事件信息有2种情况:①恶意节点j不在节点i的邻域内$\mathcal{H}_1^i:j\notin\mathcal{N}_i$;②恶意节点$j$在节点$i$的邻域内$\mathcal{H}_1^i:j\in\mathcal{N}_i$.这就导致了节点收集的邻域信息会存在分布的不一致问题,节点使用局部的训练样本可能无法很好地表示网络中内部攻击的所有特征形式.为了缓解训练数据不足和数据分布不一致的问题,本节提出了一种基于分布式对等协议的联合学习策略(Gossip Learning)[20],以促进训练用于邻域检测和定位任务下的神经网络模型.

分布式联合学习模拟中,采用以下方式完成训练数据的收集.假设已经设置了一个训练数据收集过程包含P个可用于收集训练数据的网络$\mathcal{G}_p=(\mathcal{V}_p,\mathcal{E})$,其中$p\in\{1,\cdots,P\}$.对于每个网络$\mathcal{G}_p$,依据$\alpha$的先验信息,本文随机地选择网络中的节点充当恶意节点,然后独立运行K次频谱感知算法,节点则将本地记录的状态收集起来作为训练数据的样本来源,令$\tilde{\boldsymbol{x}}_i^k=\left[x_i^k,x_1^k,\cdots,x_j^k,\cdots,x_{\left|\mathcal{N}_i\right|}^k\right]^{\mathrm{T}},\ j\in\mathcal{N}_i$表示节点$i$的收集的本地状态.在此,将$\tilde{\boldsymbol{x}}_i^k$作为带有真实标签的数据,其在事件$\mathcal{H}_1^i$下其标签为"1",此事件包含$\mathcal{H}_1^i:j\in\mathcal{N}_i$和$\mathcal{H}_1^i:j\notin\mathcal{N}_i$两种情况.需要强调的是,$\tilde{\boldsymbol{x}}_i^k$是节点$i$本地的局部数据,节点之间的局部数据不能分享给其他节点.此外,$\mathcal{H}_0^i$下的局部数据标签为"0",可以通过运行无恶意节点的频谱感知算法进行模拟.需要指出的是,如何专门设置训练数据收集过程是一个具有挑战性的问题,在此仅假设每个节点都可以使用正确的标签获得训练数据.

3.3.1　神经网络联合学习框架

分布式联合学习的目标是让参与的节点充当本地的学习者,每个节点仅交换学习到的神经网络参数,而不分享可能包含其隐私信息的本地局部的训练数据.对于分布式的联合训练,机器学习(或者神经网络)使用的标准无约束经验风险最小化问题[21]可以描述为

$$\min_{\boldsymbol{W}}L(\boldsymbol{W})=\min\frac{1}{N}\sum_{i\in\mathcal{V}}L_i(\boldsymbol{W}) \tag{25}$$

其中,$\boldsymbol{W}$表示神经网络模型的参数;$L(\cdot)$是节点i的局部目标函数,定义为局部数据集的期望损失.本地节点i的目标是将其当地样本的预期损失降至最低,即

$$L_i(\boldsymbol{W})=\mathbb{E}_{\varsigma\sim\mathcal{I}_i}\left[\ell\left(\boldsymbol{W},\varsigma\right)\right] \tag{26}$$

其中,ς表示一对变量,由输入特征和对应标签组成,遵循未知概率分布$\mathcal{I}_i$,该概率分布依赖节点i的本地样本集;$\ell(\cdot)$是一个损失函数,用于量化ς上的预测误差.定义$\mathcal{D}_i=\left\{\varsigma_1,\varsigma_2,\cdots,\varsigma_q\right\}$表示任意节点$i\in\mathcal{V}$处的训练数据集,其中包含$q$个样本.对于全局的训练数据来说,完整数据集为$\mathcal{D}=\mathcal{D}_1\cup\mathcal{D}_2\cup\cdots\cup\mathcal{D}_N$,则式(26)中的优化目标可以写为

$$\min_{\boldsymbol{W}}L(\boldsymbol{W})=\min\frac{1}{N}\sum_{i\in\mathcal{V}}\left(\frac{1}{q}\sum_{\varsigma\in\mathcal{D}_i}\ell(\boldsymbol{W},\varsigma)\right) \tag{27}$$

其中,$L_i(\boldsymbol{W})=\frac{1}{q}\sum_{\varsigma\in\mathcal{D}_i}\ell(\boldsymbol{W},\varsigma)$,据此可以将式(26)的优化目标写成分布式的形式,每个节点可以独立地估计本地损失.神经网络的模型参数$\boldsymbol{W}$可以通过基于Gossip Learning的联合学习方式进行估计[22],下面将会描述该分布式的训练过程.

3.3.2　神经网络模型参数更新

Gossip Learning是一种无需控制中心即可从分布

式的数据学习模型参数的方法，各节点仅与其邻居节点交换模型参数而不分享其隐私的训练数据，其协议框架图如算法1所示.

算法1 基于对等协议的分布式联合学习算法流程

输入: 初始化各节点本地模型参数 $\boldsymbol{W}_i=\boldsymbol{W}$ 和学习率 η

输出: 联合学习神经网络参数 $\boldsymbol{W}_i$

REPEAT

· 模型聚合: 节点 $i\in\mathcal{V}$ 根据式(29)融合邻居模型参数

· 参数更新: 节点 $i\in\mathcal{V}$ 根据式(28)更新本地模型

· 参数发送: 节点 $i\in\mathcal{V}$ 随机选择邻居节点 $j\in\mathcal{N}_i$ 发送模型

END

在训练阶段，每个节点 i 具有结构相同的神经网络模型，各节点使用随机梯度下降(Stochastic Gradient Descent，SGD)算法来估计节点的局部模型参数 $\boldsymbol{W}_i$. 节点本地参数更新过程可以表示为

$$\boldsymbol{W}_i\leftarrow\boldsymbol{W}_i-\eta\hat{\nabla}L_i(\boldsymbol{W}_i),\quad i\in\mathcal{V} \tag{28}$$

其中，η 和 $\hat{\nabla}L_i(\cdot)$ 分别为节点 i 处神经网络模型训练使用的学习率和本地期望梯度. 通常在完成本地的迭代之后，节点会周期地将本地的模型参数发送给邻居节点，从而实现模型的聚合. 在分布式联合学习中，节点 i 仅与自身的邻居集合 $\mathcal{N}_i$ 交互，模型的聚合过程由节点独立在本地完成，即

$$\boldsymbol{W}_i\leftarrow\sum_{j\in\{\mathcal{N}_i\cup i\}}\mu_j\boldsymbol{W}_j,\quad\sum_{j\in\{\mathcal{N}_i\cup i\}}\mu_j=1 \tag{29}$$

其中，$0\leqslant\mu_j\leqslant1$ 为本地参与聚合过程模型的权重系数，本地节点 i 和邻居节点系数之和为1. 该方法不存在中央控制节点，可以从完全分布式的数据中学习可靠模型参数.

4 仿真分析

本节测试了所提TDNN和SDNN方法在分布式认知网络中恶意节点的检测和定位性能，并选择文献[8，9]中的TD和SD作为对比方法. 假设网络中仅有一个PU用户，本文在9个SU节点(N=9)的曼哈顿网络(文献[9]中图3)上模拟运行基于共识的分布式协作频谱感知算法，其中节点1为恶意节点. 在算法运行中，任意节点 $i\in\mathcal{V}$ 在协议运行时以 $P_{ij}=1\Big/\left|\mathcal{N}_i\right|$ 的概率选择邻居节点 $j\in\mathcal{N}_i$ 进行信息的交互，设置协议终止的时间为 $T=500$. 在模拟中，分布式网络每一个样本数据的收集需要运行 K 次算法，其中每次 $k\in\{1,2,\cdots,K\}$ 都从一个新的初始状态开始. 正常节点的感知值初始分布设置为 $x_r^k(0)\sim\mathcal{U}[-0.5,1.5]$，$r\in\mathcal{V}_r$，服从区间均值为0.5、区间长度为2的均匀分布；恶意节点的攻击目标值初始分布服从均值为0、方差为1的高斯分布，即 $x_s^k(0)=\alpha^k\sim\mathcal{N}(0,1)$. 恶意节点添加的人工噪声的衰减参数与期望矩阵的第二大特征值相关，即 $\beta_j^k(t)\sim\mathcal{U}\left[-\left(\lambda_2(\bar{\boldsymbol{A}})\right)^t,\left(\lambda_2(\bar{\boldsymbol{A}})\right)^t\right]$.

对于邻域检测和定位任务，可以通过设置不同的参考阈值 δ_{NN} 和 ϵ_{NN} 对检测和定位任务中的不同事件进行预测分类. 为了评估相关决策函数的性能，定义以下评估指标：

$$\begin{cases}P_{nd}^i:=P(\hat{\mathcal{H}}^i=\mathcal{H}_1^i|\mathcal{H}_1^i)\\P_{nf}^i:=P(\hat{\mathcal{H}}^i=\mathcal{H}_1^i|\mathcal{H}_0^i)\end{cases}\quad\begin{cases}P_{ld}^i:=P(\hat{\mathcal{H}}^{ij}=\mathcal{H}_1^{ij}|\mathcal{H}_1^{ij})\\P_{lf}^i:=P(\hat{\mathcal{H}}^{ij}=\mathcal{H}_1^{ij}|\mathcal{H}_0^{ij})\end{cases} \tag{30}$$

其中，$\hat{\mathcal{H}}^i$ 和 $\hat{\mathcal{H}}^{ij}$ 分别为邻域检测任务和定位任务的预测事件；$P_{nd}^i(P_{nf}^i)$ 和 $P_{ld}^i(P_{lf}^i)$ 分别是邻域的恶意节点检测成功(检测虚警)和定位成功(定位虚警)概率. 具体地，$P(\hat{\mathcal{H}}^i=\mathcal{H}_1^i|\mathcal{H}_1^i)$ 表示在事件 $\mathcal{H}_1^i$ 下经过相关决策函数能够对该事件预测正确($\hat{\mathcal{H}}^i=\mathcal{H}_1^i$)的概率；$P(\hat{\mathcal{H}}^{ij}=\mathcal{H}_1^{ij}|\mathcal{H}_1^{ij})$ 表示在事件 $\mathcal{H}_1^{ij}$ 下，节点执行定位任务能够预测准确($\hat{\mathcal{H}}^{ij}=\mathcal{H}_1^{ij}$)的概率. 同理，可得邻域检测和定位任务中的虚警概率. 基于 $P_{nd}^i(P_{nf}^i)$ 和 $P_{ld}^i(P_{lf}^i)$ 这2个概率，然后通过调整不同阈值得到ROC(Receiver Operating Characteristic)曲线，可以对不同决策函数建立的模型进行更全面的评价[23].

4.1 训练和测试参数设置

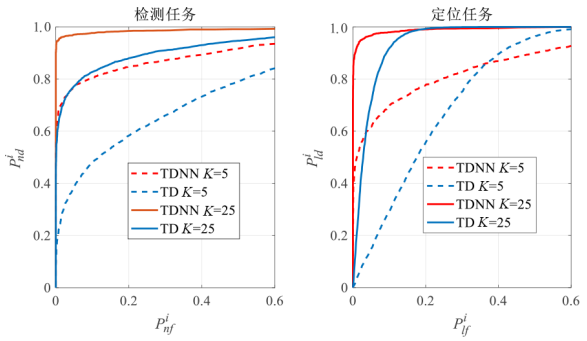
在邻域检测和定位任务中，本文神经网络模型含有3个隐含层，神经元数量分别为100，60和40. 采用小批量随机梯度下降(mini-batch SGD)法更新神经网络模型的参数，学习率为0.1，mini-batch的大小为50，最大训练轮数为200. 为了训练和测试所提方法，曼哈顿9节点网络在不同事件 $\mathcal{H}_1^i$ 和 $\mathcal{H}_0^i$ 下多次运行感知算法收集所需的样本信息. 对于每一个用于训练或者测试的样本，均由观察节点考虑 K 轮邻居节点的状态轨迹，然后在观察节点处进行融合生成，训练集、测试集及相应标签如表1所示. 用于定位任务的训练和测试样本是在已经确认网络中存在一个攻击者的情况下收集的，这样做的主要目的是让观察节点能够准确识别邻域内的邻居节点是恶意节点. 同时，为了防止神经网络模型出现退化或者出现走捷径的情况，在模型训练的过程中随机调整攻击者特征值在输入向量中的位置以及对应的标签的位置.

表1 神经网络的训练和测试数据集

邻域任务	任务事件	训练集	测试集	标签
检测任务	$\mathcal{H}_0^i$	4000	2000	0
	$\mathcal{H}_1^i;j\in\mathcal{N}_i\,\&\,j\in\mathcal{V}_s$	4000	2000	1
定位任务	$\mathcal{H}_1^i;j\in\mathcal{N}_i\,\&\,j\in\mathcal{V}_s$	4000	2000	$\boldsymbol{e}_j,j\in\mathcal{V}_s$

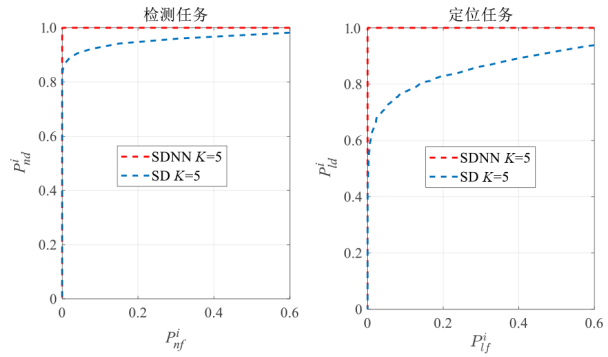
4.2　恶意节点检测和定位性能分析

本节给出了TDNN和SDNN的邻域检测和定位性能,使用TD方法作为TDNN方法的对比方法,SD方法作为SDNN的对比方法,因为TDNN和TD均只考虑了初始和稳定状态,而SD和SDNN考虑了过渡状态. 邻域内的检测性能和定位性能分别位于图5、图6中,图中P_{nf}^{i}和P_{nd}^{i}分别为检测任务下的虚警概率和检测成功概率,P_{lf}^{i}和P_{ld}^{i}分别为定位任务下的虚警概率和检测成功概率. 在邻域定位任务中,本文假设邻域检测任务全部正确,即能够完全区分事件$\mathcal{H}_1^i$和$\mathcal{H}_0^i$,定位任务中采用的样本均来自事件$\mathcal{H}_1^i$为真的情况下.



(a)　邻域检测任务的性能　　(b)　邻域定位任务的性能

图5　TDNN与TD方法的ROC曲线



(a)　邻域检测任务的性能　　(b)　邻域定位任务的性能

图6　SDNN与SD方法的ROC曲线

图5给出了TDNN和TD方法的邻域检测性能和定位性能,由图可知,随着观察次数K的增加,TDNN和TD方法的检测性能和定位性能均得到明显提高. 在K=5时,TDNN的检测性能和定位性能相比TD方法有着显著的提升,在虚警概率小于0.4时,TDNN方法有着较高的检测成功率和较低的虚警概率. 当K=25时,TDNN的检测和定位任务下的ROC曲线均位于TD方法的左上方,在虚警概率小于0.2时,TDNN已经可以提供可靠的检测和定位结果. 显然,在同样的特征提取方式下,使用神经网络来拟合邻域任务下的决策函数具有更好的优势.

图6给出了K=5时SDNN和SD方法的检测和定位ROC曲线. 可以看到,SDNN和SD方法均有着不错的检测和定位性能,在K=5时这2种方法的ROC曲线已经位于图5中TDNN和TD方法ROC曲线的左上方,例如在虚警概率为0.2时TDNN和TD方法的检测(定位)概率在0.8左右,而SDNN和SD检测(定位)概率均在0.9以上. 这表明,基于共识的感知算法迭代中的过渡状态确实为准确检测和定位恶意节点提供了更多的有效信息. 图6中,SDNN的ROC曲线完全位于SD方法的左上方,在任意虚警概率下,SDNN方法均有着较好的检测(定位)概率. 这表明本文所提出的SDNN方法可以显著改善邻域任务下的检测和定位性能.

4.3　分布式联合学习性能分析

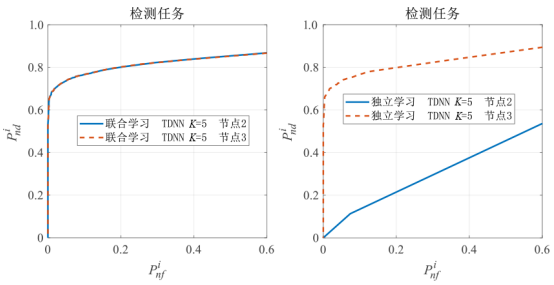
本节测试了神经网络模型在联合学习策略下的性能,参数设置与第4.1节相同,分别测试了TDNN和SDNN这2种方法的邻域检测性能. 图7~9中通过对比节点在独立学习和联合学习下的性能,验证联合学习方法的有效性. 独立学习指的是每个节点基于其本地数据训练来本地的模型参数,而不与邻居节点共享模型参数;联合学习指的是节点在训练神经网络模型的过程中,通过与邻居节点分享模型参数促进本地模型的训练.

图7给出了一个简单的联合学习测试场景,即节点存在训练数据不足无法训练出有意义的检测模型. 考虑了曼哈顿网络中的节点2和节点3,其中节点2含有充足的训练样本,节点3本地含有较少的训练数据,测试样本数量设置与表1相同. TDNN模型的参数设置与4.1节相同,观察次数设置为$K=5$. 图7中的(a)和(b)子图分别为联合学习和独立学习下的ROC曲线. 由图可知,在样本数据不足以完成有意义的训练时,独立学习方案下的TDNN在邻域检测任务中的性能表现很差,而联合学习方案下TDNN有着不错的检测性能.

图8给出了节点含有不同数量样本情况下的联合学习性能. 考虑了曼哈顿网络中的节点2、节点3和节点4. 训练样本数量的设置依次增加,分别为200,400和600. 由图8可知,独立学习方式下各节点由于样本数量设置的不同,其邻域检测性能有着明显的变化,节点4的性能最好(黄色曲线)、节点3次之(蓝色曲线),节点2最差(红色曲线). 联合学习方式下,虽然各节点的检测性能有着相同的趋势,如图8(a)小窗口所示,但节点2和节点3仍然有着良好的检测性能. 总的来说,联合学习使节点能够从其邻居节点那里学习到有效模型,在自身模型性能不足的情况下通过合并邻居节点的模型参数能够提高其检测性能.

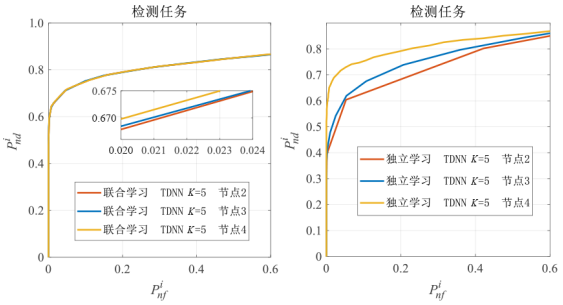
图9给出了SDNN方法在$K=1$的邻域检测性能,其中"next to"数据是指在恶意节点旁边的正常节点$\mathcal{H}_1^i$:$j\in\mathcal{N}_i$收集的样本,而"far from"数据是指在远离恶意节

点的正常节点$\mathcal{H}_1^i: j \notin \mathcal{N}_i$处收集的样本. 考虑了曼哈顿网络中的“next to”(2,3,4和7)节点以及“far from”(5,6,8和9)节点. 图9(a)为联合学习(实线)方式以及独立学习(虚线)方式下各节点在“next to”数据下的性能测试. 类似地,图9(b)为联合学习(实线)方式以及独立学习(虚线)方式下各节点在“far from”数据下的性能测试. 由图可知,各节点在联合学习方式下在“next to”或者“far from”均有着相近的检测性能,独立学习方式下各节点的检测性能有着不同的变化. 可以看出,不同类型的节点(“next to”和“far from”)在本地节点局部样本数据存在不一致的情形下也能通过联合学习获得较好的模型性能.



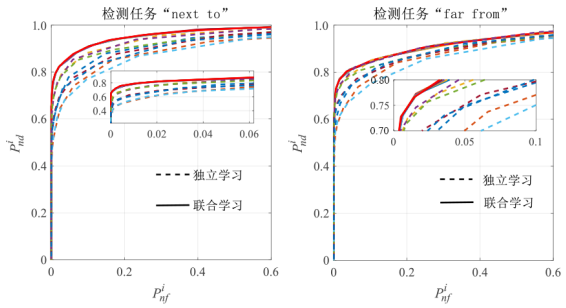
(a) 联合学习 (b) 独立学习

图7 本地训练数据不足时的联合学习和独立学习下的ROC曲线



(a) 联合学习 (b) 独立学习

图8 本地训练数据不同时的联合学习和独立学习下的ROC曲线



(a) 检测任务“next to” (b) 检测任务“far from”

图9 $K=1$时SDNN方法在联合学习和独立学习方式下的检测性能曲线

4.4 树莓派模拟下的性能分析

本节采用树莓派构建了含9个SU节点的曼哈顿网络,如图10所示. 各节点之间通过一个公用WIFI热点互相连接,通过给树莓派写入分布式共识算法运行的脚本文件,让树莓派模拟网络中的正常节点或者恶意节点. 节点状态的收集以及恶意节点的检测和定位均基于树莓派平台完成,考虑了基于共识协作感知算法下的恶意节点检测和定位,令设备1为恶意节点.

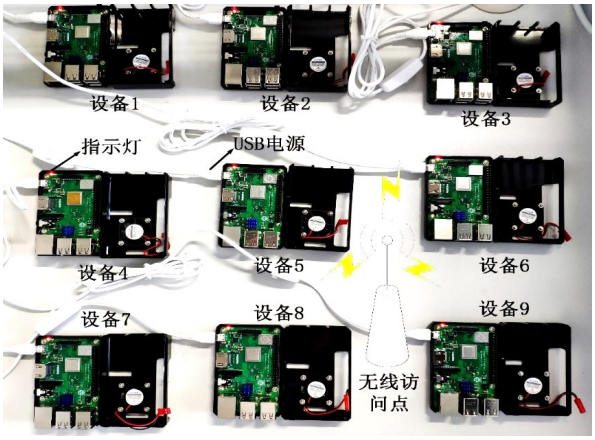
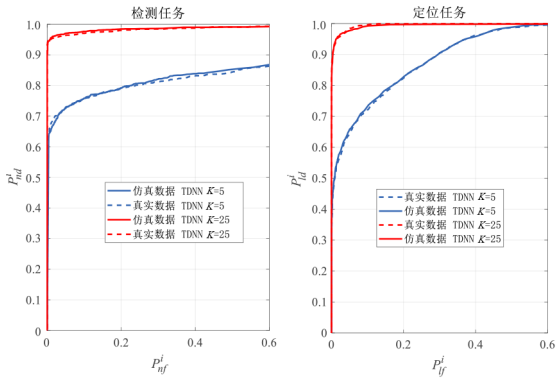


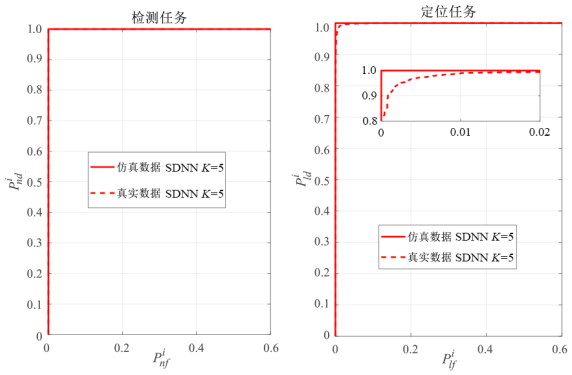
图10 基于树莓派模拟的9节点曼哈顿认知网络实物图

图11给出了TDNN在$K=5$和$K=25$的邻域检测性能和定位性能. 图中“真实数据”是神经网络模型在树莓派环境下获取样本数据进行训练和测试的性能.“仿真数据”是神经网络模型在MATLAB环境下模拟感知算法运行得到,然后进行训练和测试的性能. 可以看到,邻域检测任务和定位任务下TDNN方法在真实数据和仿真数据下ROC性能几乎没有差别. 图12给出了SDNN方法在$K=5$时的性能比较. 观察可知,SDNN方法在真实环境和仿真环境下的性能表现极为接近,在$K=25$时可以提供较好的检测定位性能. 总的来说,将神经网络模型部署在树莓派上也能很好地工作,并且与仿真环境下的性能表现非常接近.



(a) 检测任务 (b) 定位任务

图11 TDNN的真实性能和仿真性能对比



(a) 检测任务　　(b) 定位任务

图 12　SDNN的真实性能和仿真性能对比

5　结束语

本文研究了基于共识的分布式频谱感知算法受到数据注入攻击的情形,利用神经网络模型的自动学习能力实现复杂非线性函数的拟合,利用TDNN和SDNN这2种方法检测和定位网络中的恶意SU节点,从而帮助SU网络正确地感知频谱占用情况. 针对网络节点存在训练数据不足及数据分布不一致的问题,本文提出了一种基于Gossip Learning的联合学习策略促进本地神经网络模型的训练. 在曼哈顿9节点网络上模拟了分布式频谱感知过程,并对所提方法进行了仿真验证. 与传统的统计分数方法TD和SD相比,基于神经网络的方法能够显著提高攻击者的检测和定位性能. 同时,联合学习策略下各节点能够学习到与全局模型接近的局部模型,具有更好的鲁棒性.

参考文献

[1] 刘鑫, 仲伟志, 井庆丰. 认知无线电多时隙联合频谱感知方法及优化[J]. 电子学报, 2015, 43(5): 895-900.
LIU X, ZHONG W Z, JING Q F. Multi-slot cooperative spectrum sensing method with optimization in cognitive radio[J]. Acta Electronica Sinica, 2015, 43(5): 895-900. (in Chinese)

[2] 丁家昕, 方箭, 王坦. 实施动态频谱管理 提高资源利用效率[J]. 中国无线电, 2018, (1): 25-28.

[3] MITOLA J, MAGUIRE G Q. Cognitive radio: Making software radios more personal[J]. IEEE Personal Communications, 1999, 6(4): 13-18.

[4] 季薇, 李炳星, 郑宝玉. 基于信誉与共识的分布式智能入侵防御方案[J]. 系统工程与电子技术, 2018, 40(3): 665-670.
JI W, LI B X, ZHENG B Y. Distributed intelligent intrusion prevention scheme based on reputation and consensus [J]. Systems Engineering and Electronics, 2018, 40(3): 665-670. (in Chinese)

[5] YU G H, WU J, LONG C N. Contextual binary gossip: A fast cooperative spectrum sensing algorithm for cognitive radio networks[C]//Proceeding of the 11th World Congress on Intelligent Control and Automation. Piscataway: IEEE, 2014: 3013-3018.

[6] DEGROOT M H. Reaching a consensus[J]. Journal of the American Statistical Association, 1974, 69(345): 118-121.

[7] SUNDARAM S, GHARESIFARD B. Distributed optimization under adversarial nodes[J]. IEEE Transactions on Automatic Control, 2019, 64(3): 1063-1076.

[8] GENTZ R, WAI H T, SCAGLIONE A, et al. Detection of data injection attacks in decentralized learning[C]//2015 49th Asilomar Conference on Signals, Systems and Computers. Piscataway: IEEE, 2015: 350-354.

[9] GENTZ R, WU S X, WAI H T, et al. Data injection attacks in randomized gossiping[J]. IEEE Transactions on Signal and Information Processing Over Networks, 2016, 2 (4): 523-538.

[10] WU S X, WAI H T, SCAGLIONE A, et al. Data injection attack on decentralized optimization[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing. Piscataway: IEEE, 2018: 3644-3648.

[11] 黄豪杰, 吴晓晓, 李刚强. 基于区块链智能合约的物联网恶意节点检测和定位[J]. 物联网学报, 2020, 4(2): 58-69.
HUANG H J, WU X X, LI G Q. Anomaly detection and location of malicious node for IoT based on smart contract in blockchain network[J]. Chinese Journal on Internet of Things, 2020, 4(2): 58-69. (in Chinese)

[12] LI Z Q, YU F R, HUANG M Y. A distributed consensus-based cooperative spectrum-sensing scheme in cognitive radios[J]. IEEE Transactions on Vehicular Technology, 2010, 59(1): 383-393.

[13] CHAMLEY C, SCAGLIONE A, LI L. Models for the diffusion of beliefs in social networks: An overview[J]. IEEE Signal Processing Magazine, 2013, 30(3): 16-29.

[14] BOYD S, GHOSH A, PRABHAKAR B, et al. Randomized gossip algorithms[J]. IEEE Transactions on Information Theory, 2006, 52(6): 2508-2530.

[15] DIMAKIS A G, KAR S, MOURA J M F, et al. Gossip algorithms for distributed signal processing[J]. Proceedings of the IEEE, 2010, 98(11): 1847-1864.

[16] PATEL S, KHATANA V, SARASWAT G, et al. Distributed detection of malicious attacks on consensus algorithms with applications in power networks[C]//2020 7th International Conference on Control, Decision and Information Technologies(CoDIT). Piscataway: IEEE, 2020: 397-402.

[17] SCHÖLKOPF B, PLATT J, HOFMANN T. Multi-instance multi-label learning with application to scene clas-

sification[C]//Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference. Commonwealth: MIT Press, 2006: 1609-1616.
[18] SVOZIL D, KVASNICKA V, POSPICHAL J. Introduction to multi-layer feed-forward neural networks[J]. Chemometrics and Intelligent Laboratory Systems, 1997, 39(1): 43-62.
[19] WANG H Z, LI G Q, WANG G B, et al. Deep learning based ensemble approach for probabilistic wind power forecasting[J]. Applied Energy, 2017, 188: 56-70.
[20] GIARETTA L, GIRDZIJAUSKAS Š. Gossip learning: off the beaten path[C]//2019 IEEE International Conference on Big Data(Big Data). Piscataway: IEEE, 2019: 1117-1124.
[21] JIANG Z H, BALU A, HEGDE C, et al. Collaborative deep learning in fixed topology networks[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: MIT Press, 2017: 5906-5916.
[22] BLOT M, PICARD D, THOME N, et al. Distributed optimization for deep learning with gossip exchange[J]. Neurocomputing, 2019, 330: 287-296.
[23] FAWCETT T. An introduction to ROC analysis[J]. Pattern Recognition Letters, 2006, 27(8): 861-874.

作者简介

吴晓晓 女,1982年出生,湖北鄂州人. 深圳大学助理教授. 主要研究方向为社交网络中的数据挖掘算法、5G通信网络关键技术研究、信道编码理论等.

李刚强 男,1989年出生,河南驻马店人. 黄淮学院讲师. 主要研究方向为社交网络中的数据挖掘算法、分布式协议、机器学习等.

张胜利(通讯作者) 男,1978年出生,河北人. 深圳大学教授,博士生导师. 主要研究方向为无线通信、区块链关键技术、物理层网络编码等.
E-mail: zsl@szu.edu.cn