

面向雾计算的个性化轻量级分布式网络入侵检测系统

叶天鹏, 林祥, 李建华, 张轩凯, 许力文

(上海交通大学网络空间安全研究院, 上海 200240)

摘要: 随着物联网技术不断发展, 低时延高动态大带宽的新型物联网应用不断出现。这些需求导致海量设备和信息广泛聚集在网络边缘, 因而推动了雾计算架构的出现和深入发展。而随着雾计算架构的广泛深入应用, 为了保障其安全所部署的分布式网络安全架构也面临着雾计算本身所带来的挑战, 如雾计算节点计算和网络通信资源的局限性以及雾计算应用的高动态性限制了复杂网络入侵检测算法的边缘化部署。为了解决上述问题, 提出了一个面向雾计算架构的个性化轻量级分布式网络入侵检测系统(PLD-NIDS)。该系统基于卷积神经网络架构训练大规模复杂网络流入侵检测模型, 同时进一步采集各雾计算节点的网络流量类型分布情况, 提出个性化模型蒸馏算法和基于加权一阶泰勒近似剪枝算法对复杂模型进行快速个性化压缩, 突破了传统模型压缩算法在面对大量个性化节点时由于压缩计算开销过大而只能提供单一压缩模型用于边缘节点部署的局限性。根据实验结果, 所提的PLD-NIDS架构能够实现边缘入侵检测模型的快速个性化压缩。与传统模型剪枝算法相比, 所提出的架构在计算损耗和模型精度上取得了较好平衡。在模型精度上, 所提的加权一阶泰勒近似剪枝算法与传统一阶泰勒近似剪枝算法相比, 在同样的0.2%模型精度损失条件下能够提升约4%的模型压缩比。

关键词: 入侵检测; 雾计算; 模型压缩; 分布式系统

中图分类号: TP393

文献标志码: A

DOI: 10.11959/j.issn.2096-109x.2023035

Personalized lightweight distributed network intrusion detection system in fog computing

YE Tianpeng, LIN Xiang, LI Jianhua, ZHANG Xuankai, XU Liwen

Institute of Cyber Science and Technology in Shanghai Jiao Tong University, Shanghai 200240, China

Abstract: With the continuous development of Internet of Things (IoT) technology, there is a constant emergency of

收稿日期: 2022-08-26; 修回日期: 2023-02-16

通信作者: 林祥, lionel@sjtu.edu.cn

基金项目: 中央高校基本科研业务费专项资金(23X010200978); 信息网络安全公安部重点实验室(公安部第三研究所)开放课题(C20608)

Foundation Items: The Fundamental Research Funds for the Central Universities” (23X010200978), Open Subject of the Ministry of Public Security Key Laboratory of Information Network Security (The Third Re-search Institute of the Ministry of Public Security) (C20608)

引用格式: 叶天鹏, 林祥, 李建华, 等. 面向雾计算的个性化轻量级分布式网络入侵检测系统[J]. 网络与信息安全学报, 2023, 9(3): 28-37.

Citation Format: YE T P, LIN X, LI J H, et al. Personalized lightweight distributed network intrusion detection system in fog computing[J]. Chinese Journal of Network and Information Security, 2023, 9(3): 28-37.

new IoT applications with low latency, high dynamics, and large bandwidth requirements. This has led to the widespread aggregation of massive devices and information at the network edge, promoting the emergence and deep development of fog computing architecture. However, with the widespread and in-depth application of fog computing architecture, the distributed network security architecture deployed to ensure its security is facing critical challenges brought by fog computing itself, such as the limitations of fog computing node computing and network communication resources, and the high dynamics of fog computing applications, which limit the edge deployment of complex network intrusion detection algorithms. To effectively solve the above problems, a personalized lightweight distributed network intrusion detection system (PLD-NIDS) was proposed based on the fog computing architecture. A large-scale complex network flow intrusion detection model was trained based on the convolutional neural network architecture, and furthermore the network traffic type distribution of each fog computing node was collected. The personalized model distillation algorithm and the weighted first-order Taylor approximation pruning algorithm were proposed to quickly compress the complex model, breaking through the limitation of traditional model compression algorithms that can only provide single compressed models for edge node deployment due to the high compression calculation overhead when facing a large number of personalized nodes. According to experimental results, the proposed PLD-NIDS architecture can achieve fast personalized compression of edge intrusion detection models. Compared with traditional model pruning algorithms, the proposed architecture achieves a good balance between computational loss and model accuracy. In terms of model accuracy, the proposed weighted first-order Taylor approximation pruning algorithm can achieve about 4% model compression ratio improvement under the same 0.2% model accuracy loss condition compared with the traditional first-order Taylor approximation pruning algorithm.

Keywords: intrusion detection, fog computing, model compression, distributed system

0 引言

随着物联网技术的快速发展,大量传感设备被部署在人们的生活环境中,这意味着海量的数据需要被低时延传输和处理。而被广泛应用的云计算架构因其中心化的特点,会导致数据的处理过程远离用户,从而产生巨大的网络传输开销和传输时延。因此,雾计算架构被研究者提出,其通过将数据处理能力分布式部署在网络边缘靠近用户的网络节点上,从而有效利用空闲的计算资源,同时降低网络传输开销和传输时延。雾计算架构被认为是促进物联网体系高速发展的有力选择。

但是,网络架构的分布式演进给传统网络安全防护体系带来了新的挑战。由于大量的业务处理设备被部署至网络边缘,传统的系统边界变得模糊,而传统的基于网络边界实现网络入侵检测的网络安全防护体系变得不再适用。随着用户将业务和设备部署至网络边缘,分布式的网络入侵检测系统被研究者重视。通过将网络流量检测模型部署至网络边缘节点,从而监控关键节点及其相邻节点的网络流量安全态势,并在处理后直接

进行处置响应,或将情况回传至网络中心节点,全局判断后进行处置响应。这类分布式网络入侵检测系统(NIDS, network intrusion detection system)能够初步满足雾计算架构下的网络安全防护需求,但存在以下不足:雾计算架构下网络边缘节点的硬件资源受到成本、能耗、体积等约束,导致无法支持原有中心化设备条件下部署的超大规模网络流量分析模型^[1];网络边缘节点数量众多,所面临的流量环境存在显著不同。传统中心化流量检测算法面对的是整个业务网络的流量,在统计层面流量的类型和分布较为稳定。而每个边缘节点的小型流量检测算法需要满足特定节点流量环境的需求,同时存在高度动态的特点,需要算法具备敏捷高效的能力。

随着人工智能技术的快速发展,基于深度学习技术的入侵检测模型被研究者提出,并广泛应用至入侵检测系统^[2-3]。为了实现对基于深度学习的分布式网络入侵检测模型的轻量化,本文提出了分布式入侵检测模型压缩架构,通过分层压缩的方法,将原有的部署于中心节点的复杂入侵检测模型进行模型蒸馏和剪枝,压缩为适合部署在

资源受限的雾计算节点上的轻量级模型。同时,本文还提出了一种根据检测模型部署节点上流量分布进行个性化快速剪枝的算法,其能够对完成初步结构压缩的恶意流量识别模型进行进一步剪枝,通过对深度学习模型中神经元的重要性进行快速评估,从而将对识别精度影响不大的神经元去除,进而降低模型部署所需的内存和计算资源。相较于传统的模型轻量化剪枝算法,本文提出的算法具有推理复杂度低的优势,能够满足雾计算场景中边缘节点高度动态化的需求,同时能够在相同的剪枝比例下在部署节点上获得更高的识别精度,或在相同的识别精度条件下将模型的大小和运算资源需求进一步降低。

1 相关工作

1.1 分布式网络入侵检测系统

目前已有的关于物联网中的分布式网络入侵检测系统的部署模式的区别主要是,NIDS 模块被部署在本地的物联网设备、雾计算节点、云计算节点等其中一类或多类节点上。但是,从网络流量监控的便利性出发选择部署方式,将 NIDS 模块部署在边缘路由节点上是广泛被接受的选择,在这种情况下,各个不同的研究区别在于该分布式架构是否存在一个中心化的统筹节点或是否使用共识算法完成统合。文献[4]提出了一种基础的分布式的异常检测系统架构,每个节点都将监测其邻近的设备是否发生异常,当发生异常后将阻断其传输的数据包。文献[5]在分布式 NIDS 中定义了本地节点和协调节点,本地节点用于执行,协调节点用于模型发布,在云上进行模型训练。基于长短记忆(LSTM, long short-term memory)算法对网络攻击进行了识别,取得了一定的效果。文献[6]将分布式深度学习引入分布式网络入侵检测系统中,通过在边缘节点将本地的网络流量作为训练样本输入模型进行训练获得梯度,然后将该梯度回传至中心节点进行权重网络更新,随后同步至所有的边缘节点。这种方法充分利用了整个边缘网络的设备资源,但是同时带来了巨大的网络开销。文献[7]提出了融合基于签名的攻击检测算法和基于极端梯度提升的异常检测算法的网络入侵检测算法,并将其部署在雾计

算节点上,在测试中获得了较好的效果。通过对以往分布式网络入侵检测架构的回顾,作者认为利用边缘和中心节点的部署模式能够综合各种情况的优点,具有较好的灵活性。但是,针对分布式入侵检测算法与分布式架构进行深入研究并不多,与传统的中心化入侵检测算法并无太大区别。

对于在物联网系统中轻量级入侵检测系统的设计,研究者也进行了探索。早些年,研究者更多倾向于使用轻量级的分类判别算法实现入侵检测。例如,文献[8]与文献[9]分别使用了支持向量机(SVM)和极限学习机(ELM)来实现分布式网络入侵检测。而随着深度学习技术的发展,对深度学习模型的轻量化研究不断推进。文献[10]提出了一种基于堆叠自编码器的概率特征提取方法,并结合网络剪枝技术,通过对特征提取网络进行剪枝且复用,并单独训练针对不同攻击的分类器。这一方法有效降低了针对不同攻击类型的识别模型训练成本,但未考虑剪枝算法的成本,并且最终所形成的入侵检测深度学习模型依旧较为复杂。文献[11]提出了一种基于轻量级深度神经网络(DNN)的网络入侵检测模型,通过压缩特征提取范围,有效降低了该模型的部署资源需求。文献[12]提出了一种基于一维卷积神经网络(CNN)的轻量级物联网入侵检测算法模型,在该模型检测效果与基础模型相当的情况下,将模型的文件大小压缩了 52.4%,推理时间降低了 55.7%。文献[13]提出了基于稀疏进化训练的新型网络威胁检测模型,该模型在工业 4.0 的物联网安全的真实场景中平均测试时间为 2.29 ms,同时达到了 99%的平均检测精度。文献[14]提出了一种基于轻量级深度神经网络的网络威胁检测新方法,该方法通过扩展和压缩特征图来减少计算负担,并使用反残差结构和通道洗牌操作来实现更有效的训练。过去的相关研究更多地围绕算法本身的轻量化,未结合部署模式进行综合设计,同时未对边缘节点网络状态的复杂性进行考虑,没有针对每个节点的特点进行算法设计,无法满足边缘节点入侵检测动态变化的需求。

1.2 深度学习模型压缩

尽管在分布式网络入侵检测领域,对轻量级入侵检测模型的研究目前并不深入,但由于深度

学习算法的快速发展,研究者正在不断研究深度学习模型压缩算法从而降低其应用局限性^[15]。

深度神经网络压缩算法可以被分为4个主要类别:参数量化、模型剪枝、轻量型卷积核设计和知识蒸馏。在参数量化方面,Gong等^[16]通过使用K-means聚类算法对模型的权重参数进行聚类,从而在模型的存储过程中只需要存储K个质心,从而有效降低模型的内存开销。Gupta等^[17]将CNN模型中的参数精度从浮点精度改为16位定点精度,从而在大幅度降低模型存储内存需求的同时保证模型精度几乎不受影响。Vanhoucke等^[18]进一步证实了在模型权重参数的精度被优化至8位时,精度损失依旧可控。在模型剪枝方面,早期研究者通过非结构化剪枝,实现对已有模型权重矩阵的稀疏化,从而降低模型的内存需求和加速运行速率。Hassibi等^[19]提出基于Hessian矩阵减少权重网络的连接数量。Han等^[20]提出了模型剪枝的标准三步骤:权重训练、模型剪枝、重训练。但是为了克服非结构剪枝技术所带来的稀疏矩阵对硬件加速的影响,后期研究者重点研究结构化剪枝。研究者通过对权重网络参数重要性的判别,实现对非重要参数的批量化删除^[21]。在轻量型卷积核设计方面,研究者通过使用小尺寸卷积核代替原有卷积网络中的标准卷积核,从而有效降低参数数量和提升模型推理速度。文献[22]提出SqueezeNet,使用1×1的卷积核代替原有的3×3的卷积核,从而降低了模型的参数数量。Howard等^[23]提出MobileNet,使用深度可分离卷积的方式代替原有的标准卷积,在降低运算量的同时保证了原有的特征提取能力。在知识蒸馏方面,其核心思想是通过将大型神经网络中训练获得的数据分类知识传递至小型神经网络,从而提升小型神经网络的分类准确度。Hinton等^[24]提出了完整的知识蒸馏思想,利用大型神经网络输出层的软标签对小型神经网络训练过程进行校正,从而能够有效提升小型神经网络的分类准确度。Romero等^[25]同样利用知识蒸馏的思想,提出了FitNet,通过将宽而浅的教师网络压缩为窄而深的学生网络。

根据上述对深度学习模型压缩技术的分析,有许多值得在分布式网络入侵检测技术中应用的深度学习模型压缩所借鉴的技术,但同时需要针

对应用场景进行进一步优化。

2 系统与算法建模

本节阐述提出的个性化轻量级分布式网络入侵检测系统(PLD-NIDS, personalized lightweight distributed-NIDS)架构,并建模分析在这种架构下能够针对其分布式边缘节点的特性进行灵活快速低消耗的深度学习模型轻量化算法。

2.1 PLD-NIDS 架构

本文提出的个性化轻量级分布式网络入侵检测系统架构如图1所示,该系统利用了云计算中心算力丰富的节点作为模型训练的节点,而雾计算节点,作为分布在物联网设备边缘的边界路由节点,能够有效获取邻近设备的网络流量,适合作为流量检测模型的部署和应用节点。但是,由于每个雾计算节点面临的物联网设备种类和应用不同,且存在动态变化的需求特点,本文提出的PLD-NIDS将采集各个边缘节点所需检测流量类型的概率分布作为先验知识输入模型轻量化过程,从而针对各个雾计算节点进行个性化压缩。

本文将某个待部署的边缘计算节点 k 的各个类型流量的先验概率分布情况记为 $P^k = \{p_1^k, \dots, p_n^k\}$,其中 n 为该节点先验流量类型数量, p_i 表示第 i 种流量的概率分布大小。云计算中心针对已有的海量数据进行复杂网络的教师模型的训练,该模型记为 Model_l ,其中所有数据的类别记为集合 C ,数量记为 N 。随后,云中心利用收集获得的不同部署节点包含的流量类别的先验知识进行个性化知识蒸馏,针对节点 k 的先验概率分布 $P^k = \{p_1^k, \dots, p_n^k\}$,其中将节点 k 存在的流量类型集合记为 c_k ,类别数量为 n ,且 $n \leq N$ 。本文通过蒸馏训练获得学生模型 Model_s^k , Model_s^k 与 Model_l 的结构可以异构,该过程可以降低模型剪枝算法的目标模型大小,有效降低剪枝算法所需的计算开销。针对 Model_s^k ,使用本文提出的分布个性化剪枝算法进行剪枝,获得基于节点 k 流量先验分布 P^k 的个性化剪枝模型 Model_p^k 。

2.2 模型个性化轻量化算法

本文提出的深度学习模型个性化轻量化算法包含两个主要部分。第一部分是个性化模型蒸馏算法。本文将教师模型 Model_l 在输出层

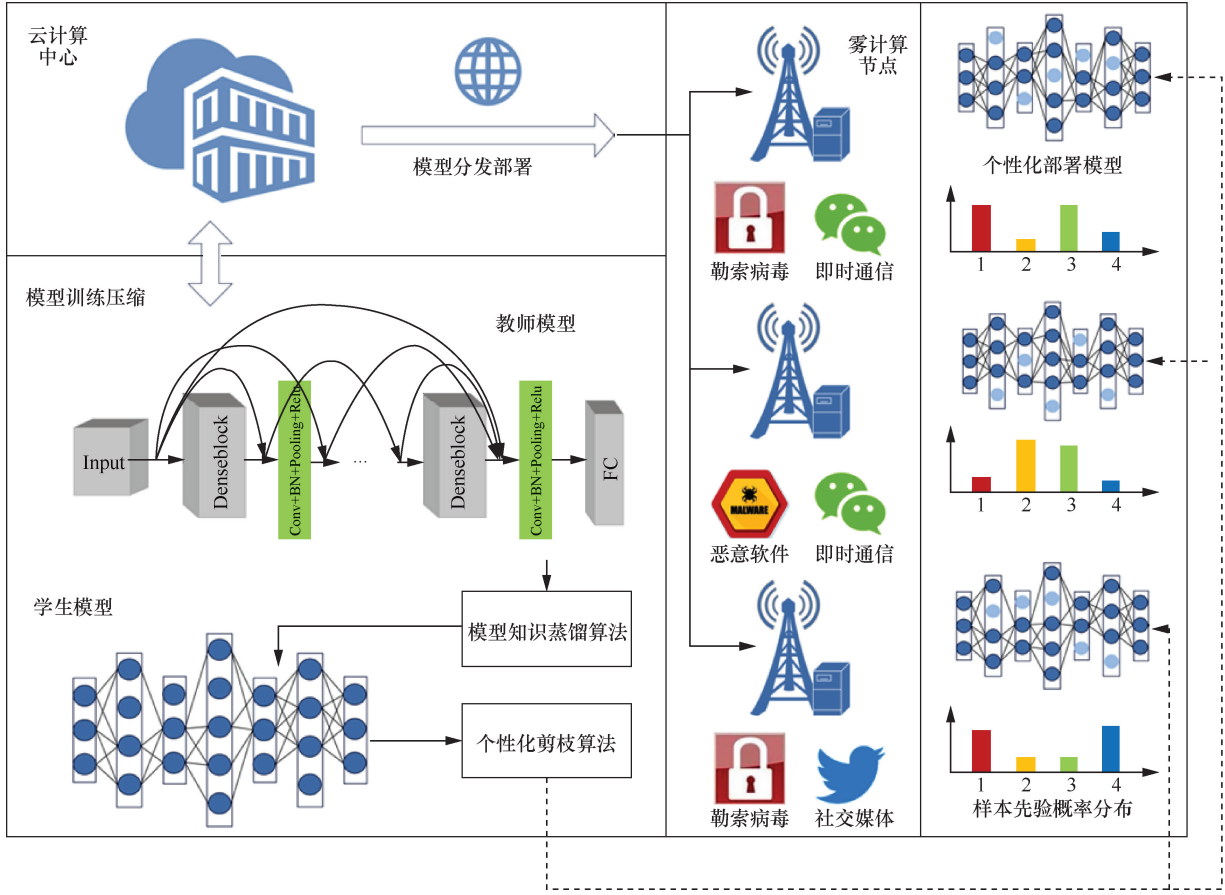


图1 个性化轻量级分布式网络入侵检测系统架构
Figure 1 PLD-NIDS architecture

Softmax 函数前所给出的输出向量记为 z_i ，即 $\hat{y}_i = \text{Softmax}(z_i)$ 为基于模型所推理的输入样本类别概率分布，同理学生模型的输出向量记为 z_s ，预测概率分布为 $\hat{y}_s$ 。因此，部署于节点 k 的学生模型 Model_s^k 的训练过程中所使用的损失函数的定义如下。

$$L_{\text{student}} = \lambda L_{\text{SL}} + (1 - \lambda) L_{\text{KD}} \quad (1)$$

$$L_{\text{SL}} = H(\max((y_s), y_r)) \quad (2)$$

$$L_{\text{KD}} = \tau^2 \text{KL}(y_s, S_k(y_i)) \quad (3)$$

其中， L_{SL} 为学生模型正常训练过程中预测结果 $\hat{y}_i$ 与真实结果 y_r 产生的损失， $H(*)$ 表示交叉熵函数。 L_{KD} 为教师模型输出与学生模型输出的差别所带来的损失函数，其中 τ 为蒸馏温度超参数，应当根据训练效果进行调整优化， $y_s = \text{softmax}(z_s / \tau)$ ， $y_i = \text{Softmax}(z_i / \tau)$ ， $\text{KL}(*)$ 表示 KL 散度函数， $S_k(*)$ 是一个基于节点 k 中流量先验分布的映射函数，由于 $y_i = \{p_i^1, \dots, p_i^N\}$ 共

有 N 个元素，与 k 节点中流量先验分布 $P^k = \{p_1^k, \dots, p_n^k\}$ 的 n 个元素无法对应。因此，根据 k 节点中所包含的类别实现对 y_i 的映射筛选，使得

$$S_k(y_i) = \{p_i^j\}, j \in c_k \quad (4)$$

从而能够实现两个模型输出的 KL 散度计算。

第二部分是个性化模型剪枝算法。为了使模型剪枝算法具备较高的通用性和易用性，本文重点关注结构化剪枝算法，即针对模型中的通道或层进行剪枝，而非针对参数进行非结构化剪枝。非结构化剪枝算法的本质是将模型参数部分置零，无法改变该模型的结构，尽管可以利用稀疏矩阵加速技术提升模型的处理效率，但提升较不明显且对硬件有较高要求，同时对模型的体积没有压缩。本文将一个待压缩的深度学习模型的权重参数定义为 $W = \{w_1, w_2, \dots, w_m\}$ 。其中， m 对应神经网络的层数，而对于一个输入该模型的数据集 $D = \{(x_0, y_0), \dots, (x_r, y_r)\}$ ，该模型训练的目标函数可以被定义为

$$\min_W E(D, W) = \min_W E(y|x, W) \quad (5)$$

因此, 当将该模型权重参数 W 中的某一个特定神经元剪枝时, 在运算过程中等价于令 $w_{a,b} = 0$, 其中, a 表示参数所在层数, b 表示神经元编号, 这种操作会使得模型的目标函数产生损失, 基于该损失可以对该神经元的重要性进行评估, 目标函数损失越大意味着该权重对模型越重要^[26], 评估函数如下。

$$I_{a,b} = \left(E(D, W) - E(D, W | w_{a,b} = 0) \right)^2 \quad (6)$$

但是, 与其他基于权重数据的 L2 范数等进行剪枝排序的算法相比, 该算法在计算复杂度上存在严重不足, 需要对神经网络中的每一个待剪枝参数进行一次模型正向推理, 这种计算开销是不可接受的。因此, 对式(6)进行泰勒展开, 从而进行近似计算成为一种有效的加速方法, 针对函数 $f(x)$ 在 $x = a$ 处的展开公式如下。

$$f(x) = \sum_{p=0}^P \frac{f^{(p)}(a)}{p!} (x-a)^p + R_p(x) \quad (7)$$

其中, $f^{(p)}(a)$ 为函数在 a 处的 p 阶导数, $R_p(x)$ 为 p 阶剩余项。因此, 可以得到目标函数在 $w_{a,b} = 0$ 处的一阶泰勒展开公式。

$$E(D, W | w_{a,b} = 0) = E(D, w_{a,b}) - \frac{\partial E}{\partial w_{a,b}} w_{a,b} + R_1(w_{a,b} = 0) \quad (8)$$

其中, 在实际应用过程中将一阶剩余项忽略对评估精度的影响很小。因此, 式(8)的一阶泰勒快速算法为

$$I_{a,b}^{(1)} \triangleq \left(\frac{\partial E}{\partial w_{a,b}} w_{a,b} \right)^2 = (g_{a,b} w_{a,b})^2 \quad (9)$$

其中, $g_{a,b} = \frac{\partial E}{\partial w_{a,b}}$ 表示该神经元所具备的梯度,

而在神经网络的误差反向传播过程中, 令反向传播的学习率 $\varepsilon \rightarrow 0$, 则能够保持权重网络不变的情况下计算获得梯度, 因此只需直接调用并将神经元 $w_{a,b}$ 包含的所有 w_m 对应的梯度求和, 即可得到式(9)的估算值, 如下。

$$I_{a,b}^{(1)} \triangleq \sum_{w_m \in w_{a,b}} (g_{w_m} w_m)^2 \quad (10)$$

而本文希望该重要性函数对不同类别的样本的剪枝损失进行加权评价, 即个性化剪枝完成后的模型在特定边缘节点 k 上部署时能够获得更高的真实准确率。因此, 为了获得在具备先验流量分布知识 $P^k = \{p_1^k, \dots, p_n^k\}$ 的节点 k 所部署的个性化剪枝模型 Model_s^k , 本文提出计算加权重要性评估函数 $I_{a,b|D}^{(1)}$, 由基于输入样本类别的概率权重加权平均值计算获得, 计算公式如下。

$$I_{a,b|D}^{(1)} = \frac{1}{r} \sum_{(x_i, y_i) \in D} p_{y_i}^k I_{a,b}^{(1)}(x_i, y_i), p_{y_i}^k \in P^K \quad (11)$$

其中, r 为输入样本的总数, 根据输入样本 (x_i, y_i) 的真实标签 y_i 确定其归属流量类别并找到对应的 $p_{y_i}^k \in P^K$, 从而在计算加权重要性评估函数 $I_{a,b|D}^{(1)}$ 时完成加权计算。

3 实验

3.1 实验环境与准备

为了验证所提系统的有效性, 本文实验的模型训练与测试均基于 NVIDIA A40 显卡硬件平台, 使用 USTC-TFC2016 数据集进行测试, 该数据集所包含的网络流量名称及样本数量如表 1 所示。

该数据集包含 10 种正常网络流量以及 10 种恶意网络流量。其中, 恶意流量数据是引用自 CTU 研究人员从 2011 年到 2015 年真实网络环境中收集的 10 种恶意软件流量。而正常网络流量样本则为数据集建立者基于 IXIA 网络仿真工具在测试环境下生成的模拟仿真流量。本文选择 DenseNet121 结构作为教师模型的神经网络架构, AlexNe 结构作为学生模型的神经网络架构。两者均是以高维张量作为输入的, 因此需要对数据集的流量 PCAP 包样本进行预处理: 将同一五元组的流量作为同一会话分组, 记为 $S^R = s_1^R, \dots, s_v^R, \dots, s_V^R$, 即该会话共有 V 条流量。本文将压缩流量数据结构设置为 32×64 像素结构的一张图, 其中第 v 行表示该会话中的第 v 条流量。在该图中每个像素由 256 位灰度构成, 可由网络流量中每 8 位二进制转换而来。因此, 在该图中第 v 行包含 64 个像素, 用于表示第 v 条流量的前 256 位二进制内容。当第 v 条网络流量的长度不足 256 位时, 则后尾补零填满。当网络对话

中流量数量 $V < 32$ 时, 同样后尾补零填满。最终构成一个形状为 $[1, 32, 64]$ 的输入张量。

表1 USTC-TFC2016 网络流量名称及样本数量
Table 1 Network flow names and amounts of USTC-TFC2016

流量名称	训练样本	剪枝及测试样本	流量类型
BitTorrent	4 800	1 200-Zipf(1,20)	正常
FTP	4 800	1 200-Zipf(1,20)	正常
MySQL	4 800	1 200-Zipf(1,20)	正常
Skype	4 800	1 200-Zipf(1,20)	正常
Weibo	4 800	1 200-Zipf(1,20)	正常
Facetime	4 800	1 200-Zipf(1,20)	正常
Gmail	4 800	1 200-Zipf(1,20)	正常
Outlook	4 800	1 200-Zipf(1,20)	正常
SMB	4 800	1 200-Zipf(1,20)	正常
WOW	4 800	1 200-Zipf(1,20)	正常
P2P Cridex	4 800	1 200-Zipf(1,20)	恶意
Htbot	4 800	1 200-Zipf(1,20)	恶意
Neris	4 800	1 200-Zipf(1,20)	恶意
Chat/IM Shifu	4 800	1 200-Zipf(1,20)	恶意
Virut	4 800	1 200-Zipf(1,20)	恶意
Geodo	4 800	1 200-Zipf(1,20)	恶意
Email Miuref	4 800	1 200-Zipf(1,20)	恶意
Email Nsis-ay	4 800	1 200-Zipf(1,20)	恶意
Tinba	4 800	1 200-Zipf(1,20)	恶意
Game Zeus	4 800	1 200-Zipf(1,20)	恶意

现有数据集中所包含的网络流量类别一般较少, 基本在 10 个以下。而 USTC-TFC2016 包含的 20 种不同网络流量还不足以体现出真实场景中的网络流量类型的复杂性。因此, 本文实验使用 CIFAR100 数据集作为测试对象, 模拟测试在面对更多种类输入样本的情况下, 本文提出的 PL-NIDS 架构是否具备有效性和优越性。CIFAR100 数据集包含 100 种不同类别的样本, 每类样本包含 500 个训练样本和 100 个测试样本。

为了在测试过程中体现出 PL-NIDS 架构的有效性和优越性, 分别针对模型蒸馏阶段与模型剪枝阶段提出了对照消融实验算法。

在模型蒸馏阶段, 本文实验使用 CIFAR100 数据集中前 50 种类别的图片和前 20 种类别的图片测试在大量类别样本条件下获取局部类别分类模型的蒸馏算法的性能。在模型训练过程

中, 超参数 λ 设为 0.1, τ 设为 5, 学习率设为 0.000 1, dropout 概率设为 0.5。云中心已经基于 50 种类样本数据完成了大规模教师模型的训练, 记为 DenseNet-50, 该模型作为未被知识蒸馏压缩的基线模型。本文提出的算法目标为获得其中特定 20 种类别样本的子学生模型, 记为 AlexNet-20。本文使用的对照算法包括: 使用传统无加权算法的知识蒸馏算法 (消融组), 获得全类别的学生模型, 记为 AlexNet-50, 并以其对样本的分类结果为结果; 直接使用特定类别的训练样本直接训练子模型, 记为 AlexNet-20-Origin; 重新使用特定类别的训练样本训练教师模型 DenseNet-20, 并基于该模型进行知识蒸馏训练子学生模型, 记为 AlexNet-20-20。

在模型剪枝阶段, 本文实验基于同一个子学生模型 AlexNet-20 进行剪枝对比, 该部分实验基于 USTC-TFC2016 数据集进行, 在测试过程中, 由于需要设置雾计算节点中流量的先验分布, 本文提出以 $\alpha=1$ 为超参数的 zipf 分布, 随机生成流量分布类型, 并进行 10 次实验取结果的平均值作为最终的结果。在对照算法方面, 本文提出的基于权重化的一阶泰勒近似的个性化模型剪枝算法记为 Taylor-FO-Weighted。本文使用的对照算法包括: 不包含个性化加权的一阶泰勒近似的模型剪枝算法 (消融组), 记为 Taylor-FO- Origin^[27]; 基于 L2 范数结构化剪枝的算法, 该模型剪枝算法将模型每层的通道根据其模型参数的 L2 范数进行比例剪枝, 记为 L2 Structure^[28]; 基于随机结构化剪枝的算法, 该模型剪枝算法将模型的每层的通道进行随机按比例剪枝, 记为 Random Structure。

3.2 评价方法

为实现个性化的轻量级分布式网络入侵检测系统, 本文提出了针对基于不同算法进行压缩后的模型大小、模型推理复杂度、模型训练损耗以及模型精度这 4 个维度来进行对照分析。在模型大小方面, 每个被部署于雾计算节点的网络入侵检测模型由于节点存储空间的限制, 必须具备更小的存储空间。在模型推理复杂度方面, 同样受节点算力的限制, 部署模型在进行流量分析的过程中, 需要使用更低的计算量, 从而降低模型的算力消耗, 该项目标准的评估单位为百万次乘加累积操作数 (MMAC, million multiply-accum-

mulate)。在模型训练损耗方面,本文实验通过在同一个设备上完成各个不同的压缩算法,基于训练时长对各个算法模型训练损耗进行评估。在模型精度方面,本文实验对实验结果进行如下定义:TP为被模型预测为正类的正样本;TN为被模型预测为负类的负样本;FP为被模型预测为正类的负样本;FN为被模型预测为负类的正样本。

因此,模型的查准率(P)、召回率(R)以及 $F1$ 值可以基于式(12)进行定义,本实验基于模型查准率和 $F1$ 值对模型准确度进行综合评估。

$$\begin{cases} P = \frac{TP}{TP+FP} \\ R = \frac{TP}{TP+FN} \\ F1 = \frac{2PR}{P+R} \end{cases} \quad (12)$$

3.3 实验结果与分析

知识蒸馏模型测试结果如表2所示。从表2可以看出,基于大规模复杂模型训练的DenseNet-50模型能够获得最好的模型查准率 P 和 $F1$ 值,分别为0.752 2和0.748 9,与其他基于AlexNet架构的模型相比提升明显。但是该模型存在模型偏大、推理复杂度高的缺陷,尤其是在训练过程中占用大量训练资源,训练时间过长,无法满足海量边缘节点的轻量化和个性化的部署需求。而其他对照算法均采用了AlexNet架构,该架构模型复杂度更低、结构更简洁、模型更小,因此更适合轻量化部署于边缘节点。而在与其他对照算法的比较中,可以发现直接训练获得20分类的AlexNet模型,即AlexNet-20-Origin算法的训练损耗最低,但其模型准确度最低,而基于重新训练的20分类的DenseNet教师网络进行蒸馏获得

的AlexNet-20-20模型算法尽管较AlexNet-20-Origin算法在模型准确性方面有所提升,但重训DenseNet所带来的巨大训练开销无法满足场景需求。而直接基于原有的50分类DenseNet网络知识蒸馏获得50分类的AlexNet-50算法在模型查准率上有着较大的提升,为0.734 1,但其模型输出有着较多的误分类,召回率存在较大不足,导致其最终的 $F1$ 值仅有0.606 4,提升并不明显。同时其分类类别较多,导致在计算KL损失部分时带来较大的计算开销,其训练损耗也产生较大的增加。与这些对照算法相比,本文提出的AlexNet-20算法能够在各个方面取得较好的效果,与AlexNet-20-Origin算法相比,在模型查准率 P 和 $F1$ 值上分别有0.034 1和0.034 8的提升,同时在模型训练损耗方面,AlexNet-20尽管在模型训练损耗上较AlexNet-20-Origin有所增加,但与AlexNet-50和AlexNet-20-20相比增长幅度相当有限。

此外,为了验证不同蒸馏温度超参数 τ 对算法性能所产生的影响,本文进行了针对不同蒸馏系数 τ 的对照实验。在控制其他变量的同时,分别针对 $\tau = \{1, 3, 5, 10, 20\}$ 对模型蒸馏算法进行了结果测试,其结果如表3所示。

表3 不同蒸馏温度条件下 AlexNet-20 模型测试结果
Table 3 Results of AlexNet-20 model in different distillation parameters

对照参数 τ	模型查准率 P	$F1$ 值
1	0.610 4	0.604 7
3	0.631 4	0.630 1
5	0.640 0	0.637 3
10	0.624 4	0.624 7
20	0.619 8	0.609 7

表2 知识蒸馏模型测试结果
Table 2 Results of knowledge distillation models

算法名称	模型大小/MB	模型推理复杂度/MMAC	模型训练损耗/s	模型查准率 P	$F1$ 值
DeneNet-50	32.34	213.55	10 154.18	0.752 2	0.748 9
AlexNet-20	27.32	121.12	860.85	0.640 0	0.637 3
AlexNet-50	27.44	121.24	2833.68	0.734 1	0.606 4
AlexNet-20-Origin	27.32	121.12	249.96	0.605 9	0.602 5
AlexNet-20-20	27.32	121.12	11 934.31	0.613 9	0.611 0

根据上述实验结果,在一定的模型参数条件下,存在一个最优的模型蒸馏系数使得学生模型可以取得最优的性能结果。

在个性化模型剪枝部分相关实验中,40%权重剪枝测试结果如表4所示。从表4可以看出,通过模型剪枝算法,雾计算节点所需部署的模型大小和推理复杂度均有较为明显的降低。与对照剪枝算法 L2 Structure 和 Random Structure 相比,尽管这两类算法的剪枝训练损耗较低,但在剪枝比例较高的情况下,模型的精度受到了较大损失。而从与消融组的对照实验中可以发现,尽管参数重要性评估算法中对输入样本类别有特定要求导致显卡的内存加载开销增加,但在整体剪枝训练开销增加不大的情况下,基于先验流量分布知识的个性化模型剪枝(Taylor-FO-Weighted)算法在相同剪枝比例下较均匀剪枝(Taylor-FO-Origin)算法有着更高的模型准确度,在模型查准率 P 和 $F1$ 值上分别有 0.069 7 和 0.217 2 的提升。

在不同剪枝比例下模型查准率对比如图2所示。

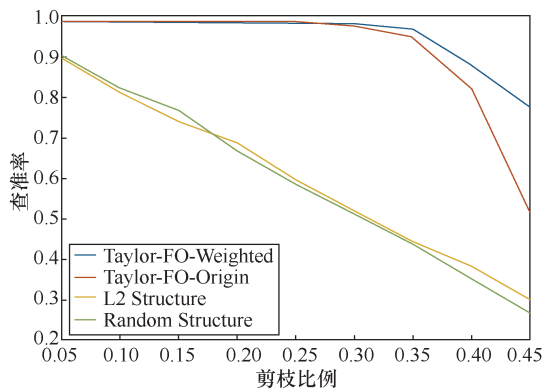


图2 不同剪枝比例下模型查准率对比

Figure 2 Comparison of model precision under different pruning ratios

根据图2可以看出,本文提出的个性化模型剪枝算法能够在较高的剪枝比例下获得相同的模型精度。与原有的一阶泰勒近似剪枝算法相比,

在两者均承受约 0.2% 的模型查准率损失的条件 下,本文提出的加权一阶泰勒近似剪枝算法能够提升约 4% 的模型压缩比。

4 结束语

由于云边协同网络架构的不断演进,雾计算架构作为边缘计算的一种形式已经被大量用于业务部署场景。因此,本文提出了一种个性化轻量级分布式网络入侵检测系统,通过对雾计算节点的流量分布情况进行先验检测,随后根据不同的节点情况进行个性化模型压缩,获得更为轻量级的部署模型,从而能够有效应对雾计算节点计算和存储资源受限的问题。在实验中,本文提出的算法经过测试均展现出了其在轻量化和个性化上的优势。同时,较低的模型压缩开销意味着该架构能够满足较高频率的动态模型部署。由于雾计算架构下网络流量分布的动态性变化,PLD-NIDS 可以满足雾计算节点所部署的个性化网络入侵检测模型快速响应的需求。

参考文献:

- [1] MANIMURUGAN S. IoT-fog-cloud model for anomaly detection using improved Naïve Bayes and principal component analysis[J]. Journal of Ambient Intelligence and Humanized Computing, 2021: 1-10.
- [2] LIU H Y, LANG B. Machine learning and deep learning methods for intrusion detection systems: a survey[J]. Applied Sciences, 2019, 9(20): 4396.
- [3] BUCZAK A L, GUVEN E. A survey of data mining and machine learning methods for cyber security intrusion detection[J]. IEEE Communications Surveys & Tutorials, 2015, 18(2): 1153-1176.
- [4] THANIGAIVELAN N K, NIGUSSIE E, KANTH R K, et al. Distributed internal anomaly detection system for Internet-of-Things[C]//Proceedings of 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC). 2016: 319-320.
- [5] DIRO A, CHILAMKURTI N. Leveraging LSTM networks for attack detection in fog-to-things communications[C]//Proceedings of IEEE Communications Magazine. 2018: 124-130.
- [6] DIRO A A, CHILAMKURTI N. Distributed attack detection scheme using deep learning approach for Internet of things[J]. Future Generation Computer Systems, 2018, 82: 761-768.

表4 40%权重剪枝测试结果
Table 4 Results of model pruning with 40% threshold

算法名称	模型大小/MB	模型推理复杂度/MMAC	剪枝损耗/s	模型查准率 P	$F1$ 值
AlexNet-20	27.32	222.91	—	0.985 7	0.985 6
Taylor-FO-Weighted	15.87	192.1	37.21	0.891 9	0.840 7
Taylor-FO-Origin	15.87	192.1	8.72	0.822 2	0.623 5
L2 Structure	16.39	133.7	4.03	0.382 9	0.390 0
Random Structure	16.39	133.7	0.098	0.374 8	0.286 4

- [7] LAWAL M A, SHAIKH R A, HASSAN S R. An anomaly mitigation framework for IoT using fog computing[J]. *Electronics*, 2020, 9(10): 1565.
- [8] MALEH Y, ABDELLAHEZZATI W. Lightweight intrusion detection scheme for wireless sensor networks[J]. *International Journal of Computer Science*, 2015, 42: 1-8.
- [9] AN X S, ZHOU X W, LYU X, et al. Sample selected extreme learning machine based intrusion detection in fog computing and MEC[J]. *Wireless Communications and Mobile Computing*, 2018, 2018: 1-10.
- [10] DAO T N, LEE H. Stacked autoencoder-based probabilistic feature extraction for on-device network intrusion detection[J]. *IEEE Internet of Things Journal*, 2022, 9(16): 14438-14451.
- [11] NGUYEN X H, NGUYEN X D, HUYNH H H, et al. Realguard: a lightweight network intrusion detection system for IoT gateways[J]. *Sensors (Basel, Switzerland)*, 2022, 22(2): 432.
- [12] 彭媛媛. 基于一维卷积神经网络的轻量级物联网入侵检测方法研究[D]. 成都: 电子科技大学, 2022.
- PENG Y Y. A lightweight IoT intrusion detection method based on one-dimensional convolutional neural network[D]. Chengdu: University of Electronic Science and Technology of China, 2022.
- [13] THAKKAR A, LOHIYA R. A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges[J]. *Archives of Computational Methods in Engineering*, 2021, 28(4): 3211-3243.
- [14] ZHAO R J, LI Z J, XUE Z, et al. A novel approach based on lightweight deep neural network for network intrusion detection[C]//*Proceedings of 2021 IEEE Wireless Communications and Networking Conference (WCNC)*. 2021: 1-6.
- [15] CHOUDHARY T, MISHRA V, GOSWAMI A, et al. A comprehensive survey on model compression and acceleration[J]. *Artificial Intelligence Review*, 2020, 53(7): 5113-5155.
- [16] GONG Y C, LIU L, YANG M, et al. Compressing deep convolutional networks using vector quantization[J]. *arXiv Preprint arXiv: 1412.6115*, 2014.
- [17] GUPTA S, AGRAWAL A, GOPALAKRISHNAN K, et al. Deep learning with limited numerical precision[C]//*Proceedings of the 32nd International Conference on Machine Learning*. 2015: 1737-1746.
- [18] VANHOUCKE V, SENIOR A, MAO M. Improving the speed of neural networks on CPUs[C]//*Proceedings of Deep Learning and Unsupervised Feature Learning Workshop*. 2011: 1-8.
- [19] HASSIBI B, STORK D G. Second order derivatives for network pruning: optimal brain surgeon[C]//*Proceedings of Advances in Neural Information Processing Systems*. 1992: 164-171.
- [20] HAN S, POOL J, TRAN J, et al. Learning both weights and connections for efficient neural networks[C]//*Proceedings of the 28th International Conference on Neural Information Processing Systems*. 2015: 1135-1143.
- [21] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient ConvNets[J]. *arXiv Preprint arXiv: 1608.08710*, 2016.
- [22] IANDOLA F N, HAN Song, MOSKEWICZ M W, et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size[J]. *arXiv Preprint arXiv:1602.07360*, 2016.
- [23] HOWARD A G, ZHU M L, CHEN B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications[J]. *arXiv Preprint arXiv: 1704.04861*, 2017.
- [24] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. *arXiv Preprint arXiv: 1503.02531*, 2015.
- [25] ROMERO A, BALLAS N, KAHOU S E, et al. FitNets: Hints for thin deep nets[J]. *arXiv Preprint arXiv:1412.6550*, 2014.
- [26] MOLCHANOV P, TYREE S, KARRAS T, et al. Pruning convolutional neural networks for resource efficient inference[C]//*Proceedings of 5th International Conference on Learning Representations (ICLR)*. 2019.
- [27] MOLCHANOV P, MALLYA A, TYREE S, et al. Importance estimation for neural network pruning[C]//*Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020: 11256-11264.
- [28] WANG W, ZHU L Q, GUO B Q. Reliable identification of redundant kernels for convolutional neural network compression[J]. *Journal of Visual Communication and Image Representation*, 2019, 63: 102582.

[作者简介]



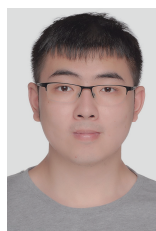
叶天鹏(1993—)，男，江苏无锡人，上海交通大学博士生，主要研究方向为车辆云计算、人工智能技术在网络安全领域的应用、自然语言处理。



林祥(1979—)，男，福建闽侯人，上海交通大学工程师，主要研究方向为网络空间安全、自然语言处理、网络安全大数据挖掘。



李建华(1965—)，男，江西九江人，上海交通大学教授、博士生导师，主要研究方向为网络空间、个人隐私保护、数据安全。



张轩凯(1999—)，男，四川广安人，上海交通大学硕士生，主要研究方向为二进制代码分析、物联网安全设备漏洞挖掘以及人工智能与安全。



许力文(1999—)，男，山东泰安人，上海交通大学硕士生，主要研究方向为网络安全态势感知。