申煜铜,谈宇浩,夏文超. 基于联邦学习的物联网设备异常检测算法研究 [J]. 智能计算机与应用,2024,14(8):225-233.
DOI:10.20169/j.issn.2095-2163.240836

基于联邦学习的物联网设备异常检测算法研究

申煜铜, 谈宇浩, 夏文超
(南京邮电大学 通信与信息工程学院, 南京 210003)

摘 要: 异常检测技术通常用于检测物联网设备未经授权的活动,以保障网络和设备的安全性。联邦学习可以在保证用户数据隐私的情况下对多方数据进行统一模型训练,因此大多数异常检测算法都采用了联邦学习架构。然而,传统联邦学习存在训练开销大、本地模型表现不一致导致全局模型精度低等问题。针对于此,本文提出了一种基于联邦学习的轻量级设备异常检测算法。该算法在网络边缘节点使用变分自编码器和 LightGBM 对数据进行降维处理和特征提取,去除了冗余特征,降低了模型训练时间;在上传模型参数时采用了动态加权梯度更新算法,减少了训练过程中局部模型表现不佳对全局模型的影响。实验结果表明,本文所提算法相比对照算法查准率最高提升 7.46%;查全率最高提升 7.99%; *F*1 分数最高提升 0.077 3;模型训练耗时降低 63.08%。

关键词: 异常检测; 物联网; 联邦学习
中图分类号: TP309 **文献标志码**: A **文章编号**: 2095-2163(2024)08-0225-09

Research on anomaly detection algorithm of internet of things equipment based on federated learning

SHEN Yutong, TAN Yuhao, XIA Wenchao
(College of Telecommunications & Information Engineering, Nanjing University of Posts & Telecommunication, Nanjing 210003, China)

Abstract: Anomaly detection technology is usually used to detect unauthorized activities of IoT devices to ensure the security of networks and devices. Federated learning can conduct unified model training for multi-party data while ensuring user data privacy, so most anomaly detection algorithms adopt federated learning architecture. However, traditional federated learning has problems such as high training costs, inconsistent local model performance, and low global model accuracy. To address these issues, this paper proposes a lightweight device anomaly detection algorithm based on federated learning. The algorithm uses variational autoencoders and LightGBM to reduce dimensionality and extract features from the data at network edge nodes, removing redundant features and reducing the training time of the model; When uploading model parameters, a dynamic weighted gradient update algorithm was used to reduce the impact of poor local model performance on the global model during the training process. The experimental results show that compared with the control group, the algorithm proposed in this paper has a maximum improvement of 7.46% in accuracy; The highest recall rate increased by 7.99%; The highest improvement in *F*1 score is 0.077 3; The model training time is reduced by 63.08%.

Key words: anomaly detection; Internet of Things; federated learning

0 引 言

物联网是一个具有感知、监控、通信和智能处理的设备网络。物联网通过收集和处理来自各种传感器的传感数据,提供比互联网更加智能的环境和服务[1]。据 IHS Markit 估计[2],到 2030 年,将有1 250 亿台设备连接到物联网。由于具有提高客户参与度、减少资源浪费、增强数据收集能力并提供技术优化等诸多优点,物联网技术被越来越多地集成到各类系统中,例如工业控制系统、智慧交通系统、关键基础设施以及智慧城市等[3]。

工业制造是物联网技术的一个典型应用场景。截至 2022 年底,全球工业制造管理中使用了 100 多亿物联网设备。到 2025 年底,这一数字将接近 750

基金项目: 国家自然科学基金(92067201)。
作者简介: 申煜铜(1998-),男,硕士研究生,主要研究方向:机器学习和物联网安全; 谈宇浩(1998-),男,硕士研究生,主要研究方向:分布式学习。
通讯作者: 夏文超(1991-),男,博士,副教授,硕士生导师,主要研究方向:通感一体化,云边协同调度,感算融合等。Email:xiawc@ njupt.edu.cn
收稿日期: 2023-04-27

亿。由于近年来物联网设备的广泛部署以及物联网传感器在安全方面存在不足,物联网设备安全受到政府和企业的密切关注。目前,物联网设备仍不能有效的抵御黑客的攻击行为,并且已经造成了超过 1 万亿美元的经济损失[4]。因此,保证物联网数据安全对日常生活和经济活动具有重要意义,物联网安全挑战必须通过稳健有效的异常检测技术来解决。

近年来,利用机器学习或深度学习算法构建异常检测模型已成为主流。虽然相关研究表明,与常规方法相比,其可以实现显著的性能提升,但其中大部分模型都是基于足够的攻击实例数据建立的[5]。然而在实际场景中,工厂或企业的网络系统生成的用于训练的高质量攻击样本数据较少。基于这些数据,机器学习异常检测模型的检测能力有限。解决这一问题的有效方法是将不同企业的数据样本收集到云服务器上,并在此基础上训练检测模型[6]。事实上,绝大多数企业不愿意与其他企业分享他们的网络流量数据,因为这些数据中通常包含敏感信息并涉及内部隐私[7]。

联邦学习(Federated Learning,FL)是谷歌研究团队在 2016 年提出的分布式机器学习框架,可以同时实现训练数据的扩展以及用户隐私保护[8]。其将持有训练数据的公司、工厂或边缘设备视为参与联邦学习的边缘节点,并且每个边缘节点的训练数据都是私有的。边缘节点在其本地建立同样的机器学习模型,并使用其私有数据集进行训练。同时,在中心云服务器上建立一个与本地模型结构相同的全局机器学习模型。云服务器和多个边缘节点之间通过传输全局模型和本地模型参数来进行模型训练,并最终建立一个性能优异的全局模型,从而完成特定的检测任务[9]。与集中式学习架构不同,FL 在通信过程中传输的是模型参数,而不是每个边缘节点的本地训练数据。因此,FL 可以间接扩展数据,避免原始数据的泄露。

尽管 FL 架构解决了集中式学习的隐私问题,但由于物联网环境受到边缘设备资源的限制,在边缘设备和云服务器之间交换的大量模型参数可能会导致 FL 的通信开销过多,从而导致边缘设备性能受限,进而影响模型收敛,使得边缘设备无法快速检测到异常情况。

因此,针对上述问题,设计一种轻量级基于 FL 架构的异常检测算法,降低检测算法在训练模型时的时间消耗是本文研究的重点。

1 相关工作

物联网设备异常检测算法一般部署在网络层和应用层。检测算法首先从设备中收集一定数量的正常运行数据,以创建检测模型;然后收集物联网设备的运行数据作为模型的输入,从而识别出其中的异常流量,完成异常检测。目前,根据系统架构异常检测方案可以分为集中式和分布式,下面将从这两个方面介绍目前国内外研究的进展情况。

1.1 集中式设备异常检测算法

集中式设备异常检测算法需要边缘节点将数据发送到服务器进行统一的处理和训练。文献[10]提出了一种基于极限梯度提升(Extreme Gradient Boosting,XGBoost)算法的分类方法。作者将 XGBoost 和长短期记忆神经网络(Long Short-Term Memory,LSTM)算法结合,构建了一个新的检测模型来确保物联网设备安全,同时搭建了一个物联网系统来进行性能评估。其结果表明,该方法在检测系统安全漏洞、恶意软件感染、异常操作和内存泄漏等方面具有良好的性能,其结果准确率为 97%。文献[11]采用轻量级 C4.5 算法,通过直接分析设备中捕获的数据包并创建决策树来搜索拒绝服务(Denial of Service,DoS)攻击。尽管该方法达到了 100%的准确率,但在其测试结果中,只有18.15%的传输数据包被分析。

虽然文献[10-11]所提算法在准确率上有着比较好的表现,但在异常检测问题中,假阳率也是非常重要的指标,其反映了系统将正常数据误判为异常数据的概率。文献[3]利用一个模拟工业工厂的测试平台来训练几个异常检测模型。结果表明,基于随机森林(Random forest,RF)的检测模型具有最好的综合效果,真阳性率(True Positive Rate,TPR)为 97.44%。但是,只有支持向量机(Support Vector Machines,SVM)的假阳率(False Positive Rate,FPR)为 0.00,表示没有把正常数据误判为异常。

以上研究仅在一个数据集上进行了验证,无法反映算法的泛化性能。与之不同的是,Verma 等[12]在 CIDDS-001、UNSW-NB15、NSL-KDD 等数据集上进行了实验,并进行了 10 折交叉验证。实验结果表明,采用分类回归树(Classification and Regression Trees,CART)算法得到的平均准确率最高,为 96.74%;而 XGBoost 的准确率为 96.73%;但其真阳率最高,为0.973 1,综合性能最好。

集中式异常检测系统虽然取得了不错的效果,但随着物联网终端设备数量的迅速增长,设备和服

务器之间的海量数据传输所带来的开销成为了许多 IoT 应用时的主要瓶颈。同时，因为数据中通常包含敏感信息并涉及内部隐私，绝大多数企业不愿意与其他企业分享其网络流量数据。面对这些挑战，分布式异常检测框架受到了研究人员的关注。

1.2　分布式设备异常检测算法

分布式设备异常检测算法是指利用多个计算节点进行机器学习的算法。在该框架中，数据处理任务将放在边缘节点中进行，模型训练任务仍在云服务器中完成。由于边缘节点更接近用户终端设备，可以降低数据处理和发送速度，降低延迟。与集中式架构不同，云服务器只接受模型的训练参数，并不需要终端设备将数据上传到云中，解决了隐私问题。

Huong 等[13]提出了一种工业物联网系统的网络攻击异常检测系统。为了检测时间序列数据中的异常，该检测算法由一个编码器、一个 LSTM 单元和一个编码器组成。此外，利用核分位数估计器优化阈值，获得较高的异常识别精度。结果表明，该方法的 $F1$ 分数为 0.979，该指标综合了查全率和查准率，反映了系统的综合性能。此外，与集中式学习架构相比，带宽需求减少了 35%。Zhao 等[14]提出了一种智能异常检测系统来检测 UNIX 命令序列中的攻击。标记器用于将命令行文本转换为向量化的 shell 命令块；然后，使用命令块作为 LSTM 模型的输入数据。经与对照组的卷积神经网络（Convolutional Neural Network，CNN）模型相比，LSTM 模型在所有指标上都优于 CNN，且所提出的 FL-LSTM 系统 $F1$ 分数达到 0.992 1。Idris 等[15]改进了 FCNN 算法，提出了一种鲁棒、有效轻量级的算法 REDNN。该算法可以有效识别出对抗性攻击行为，并在 N-BaIOT 数据集上评估，证明了 REDNN 算法性能。在鲁棒性和资源消耗方面，其表现出比同类方法更高的性能。实验结果表明，该模型能够保持较高的精度和 $F1$ 分数，同时训练内存和时间消耗分别比 FCNN 减少 67.99%和 49.80%。

然而，以上研究忽略了边缘设备之间使用 FL 进行模型训练时边缘节点和云服务器之间频繁交换参数时所产生的通信开销。存在训练开销大、本地模型表现不一致导致全局模型精度低等问题。

针对这些问题，本文提出了一种基于联邦学习的物联网异常检测算法，通过对数据降维处理并结合动态加权梯度更新，从而减少训练开销的同时提升全局模型精度。

2　异常检测系统

本文考虑到物联网中典型的应用场景，该场景包括了云服务器、边缘节点以及数量众多的物联网设备。如图 1 所示，该系统主要包括云服务器、边缘节点和终端设备 3 个层。不同类型的终端设备通过 Wifi、zigbee、蓝牙以及 5G 等通讯方式，将传感器收集到的数据发送给边缘节点。边缘节点主要完成两个任务：一是收集终端设备发送的数据，并基于该数据训练本地检测模型；二是将训练得到的本地模型的参数上传至云服务器进行全局聚合。云服务器的主要工作包括对边缘节点上传的本地模型进行聚合，并将聚合得到的模型参数下发至所有边缘节点。

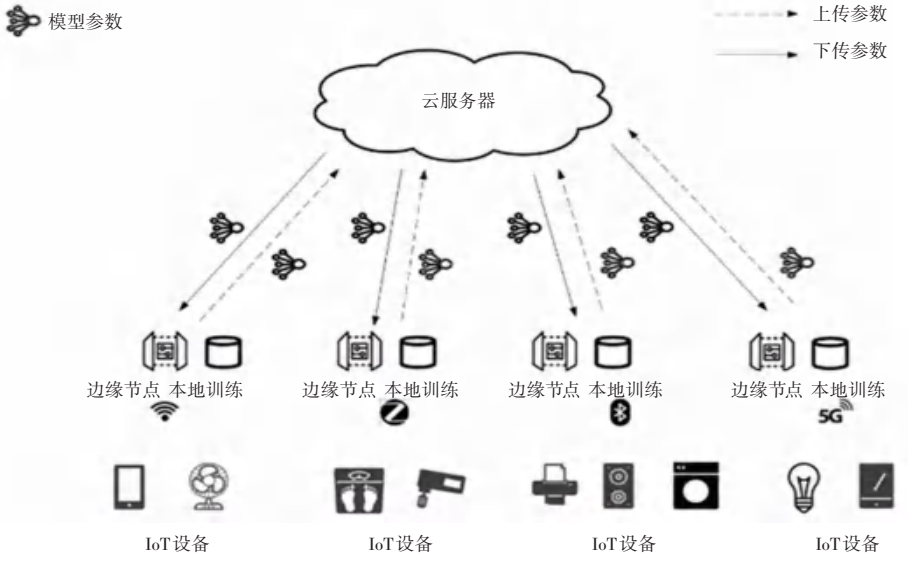


图 1　异常检测系统应用场景

Fig. 1　Application Scenarios of anomaly detection system

为确保系统中终端设备的数据安全,本文设计了一个联邦学习架构的异常检测系统,如图 2 所示。该检测系统主要包括数据预处理模块、聚类检测模块、数据降维模块、特征提取模块和模型训练模块等 5 个模块。前 4 个模块涉及到的数据处理过程是在边缘节点完成,本章将对在边缘节点处部署的 4 个模块进行具体介绍;模型训练模块是由边缘节点和云服务器协同完成的,将在第三章进行介绍。

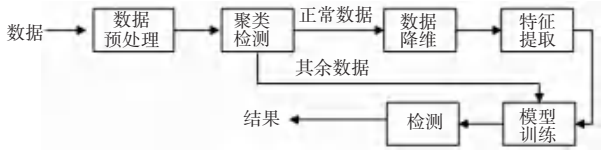


图 2 异常检测系统框图

Fig. 2 Block Diagram of anomaly detection system

2.1 数据预处理模块

数据预处理模块主要完成数据映射编码、数值归一化和数据集划分 3 个任务。

数据映射编码是指将数据中的字符型特征转化为数值型。如:本文所选用的 NSL-KDD[12] 数据集中包含 41 种特征,其中 38 个为数字特征,3 个为字符型特征。字符型特征有:协议特征 Protocol_type,其特征值为 TCP、UDP、ICMP,编码后其特征值为(1,0,0),(0,1,0),(0,0,1),1 维的特征经过编码处理后成为 3 维特征;特征 Flag 经过编码处理后为 11 维;特征 Service 经过编码处理后为 70 维。因此,该数据集的数据经过映射后成为 122 维。本文中选用的另一个数据集为 N-BaIoT[15],该数据集数据经过处理后成为 115 维。

通过数值归一化,数据集中的特征值 X 被线性映射到 0-1 区间,如公式(1)所示:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

为了对训练得到的检测模型的性能进行评估,需要把数据集划分为训练集和测试集。训练集用于模型训练,测试集用于在实际应用之前进行最后的验证评估。本文使用 80%的数据作为训练集,20%的数据作为测试集,同时本文采用了 5 折交叉验证,验证是否选取了合适的测试集。

2.2 聚类检测模块

数据预处理模块完成后得到的训练集 $D = \{X_1, X_2, \cdots, X_m\}$,该训练集中包含 m 个无标记样本,每个样本为 n 维向量,即 $X_i = (x_{i1}; x_{i2}; \cdots; x_{in})$。聚类检测模块的主要任务是对预处理后的数据进行聚类检测。检测完成后,数据将被分为正常簇和异常簇,异常簇中的数据将在下一步的分类检测模块进行检测。

本文采用 Birch 算法作为聚类检测方法,由于该算法基于聚类特征树(Cluster Feature Tree, CF Tree)来生成聚类结果,不需要设置任何初始类或初始质心,特别是对于非常大的数据集具有较高的聚类速度。这一优势可以完全满足物联网设备流量数据的时序要求。

采用 Birch 算法聚类处理后,边缘节点处的数据集 D 将被划分为 k 个不相交的簇 $\{C_l \mid l = 1,2,\cdots,k\}$。考虑到工业物联网的终端设备所产生的流量数据绝大多数为正常数据,因此将聚类后数据最多的簇作为正常簇数据(记为 C_{normal}),余下的 $k-1$ 个数据簇作为待检测数据,将在下一步的分类检测中进行判别。

2.3 数据降维模块

数据降维主要是对聚类检测模块中获得的正常数据簇 C_{normal} 进行降维处理。目前,绝大多数机器学习算法都涉及到距离计算,当数据的维度过高时,计算内积都需要消耗高额的计算资源;其次,尽管观测或收集到的数据是高维的,但很多时候和学习任务有关的仅是某些维度的特征,在高维空间训练模型容易将训练样本自身的特点看作所有样本共有的特点,使最终得到的检测模型过拟合,导致模型泛化性能下降。数据降维的目的就是减少冗余信息对模型的干扰,降低构建机器学习模型时的计算量。

本文选择变分自编码器(Variational Auto Encoder, VAE)算法进行数据降维,VAE 结构如图 3 所示。其中,编码器和解码器每个只有两个完全连接的隐藏层,实现了模型的简单性和轻量级。这种轻量级的 VAE 可以在硬件资源有限的边缘节点上进行训练,同时降低了在联邦学习环境中将模型参数发送到云服务器中所引起的通信成本,并提供了足够的检测性能。

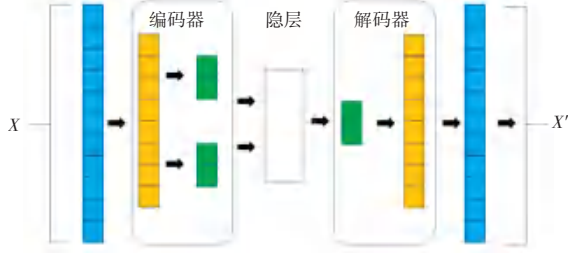


图 3 变分自编码器结构

Fig. 3 Variational autoencoder structure

本文在 VAE 算法训练完成后,将其编码器的输

出作为降维处理后的结果。即经过 VAE 算法处理后，n 维的数据 $X_i = (x_{i1}; x_{i2}; \cdots; x_{in})$ 将被转换为 d 维数据，此时，$X_i = (x_{i1}; x_{i2}; \cdots; x_{id})$，其中 $d < n$。本文所使用的 NSL-KDD[12] 数据集和 N-BaIoT[16] 数据集在降维处理前后数据维度的变化见表 1。

表 1　数据集降维前后对比

Table 1　Comparison of dataset before and after dimensionality reduction

数据集	降维前数据维度	降维后数据维度
NSL-KDD	121	18
N-BaIoT	115	15

2.4　特征提取模块

经过降维处理后，n 维的数据转变为 d 维数据 $X_i = (x_{i1}; x_{i2}; \cdots; x_{id})$，数据中某些特征对分类结果起到关键作用，而余下的特征为冗余特征，其中所包含的信息可以从关键特征中获得，因而对分类结果的作用很小。特征提取的目的就是将这些冗余特征去除，使用关键特征来构建异常检测模型。

边缘节点从云服务器中下载检测模型进行异常流量检测，检测完成后将模型参数发送给云服务器进行聚合。本文选择轻量级梯度提升算法（Light Gradient Boosting Machine，LightGBM）作为检测算法。

经过特征提取模块处理后，将选择出对结果影响最大的部分特征。LightGBM 算法本质上为一种决策树，基于所选特征可以对设备的流量信息进行判断，从而检测出异常流量数据。

由于本文采用了联邦学习架构，涉及到云服务器对检测模型的聚合更新过程，因此检测算法进行数据处理和模型训练的主要步骤如下：

步骤 1　边缘节点对收集到的传感器数据进行预处理后作为本地数据集；

步骤 2　边缘节点对数据集进行聚类检测，检测完成后，将正常数据分为一类，其余数据分为 N 类，在分类检测过程中检测；

步骤 3　边缘节点对步骤 2 中获得的正常数据进行数据降维处理；

步骤 4　云服务器向边缘节点发送初始检测模型；

步骤 5　边缘节点使用云服务器发送的检测模型对数据进行训练检测，在训练完成后向云服务器发送本地模型参数；

步骤 6　云服务器通过聚合边缘节点发送的模型参数来获得新的全局检测模型；

步骤 7　云服务器将全局检测模型发送到每个边缘节点。

重复步骤 5-步骤 7，直到全局模型收敛。该最优全局模型可以用来执行异常检测任务。

云服务器在聚合参数时通常采用联邦平均算法（Federated Averaging Algorithm，FedAvg），该算法计算设备参数的平均值，然后发送给边缘节点。考虑到不同边缘节点在训练完成后得到的最优特征并不完全相同，同时模型检测性能也有着较大差异。因此，本文提出了动态加权梯度更新算法。

3　动态加权梯度更新算法

文献[15]在云服务器对模型进行聚合更新时采用了 FedAvg 算法，而在联邦学习过程中，全局模型的性能会受到一些检测性能较差的模型或恶意模型的干扰，从而不能快速收敛，降低模型检测准确率。

因此，为了减少通信时间和一些局部模型对全局模型的不利影响，本文提出了动态加权更新算法。具体而言，根据局部模型在每轮训练中的准确率，引入局部模型过滤和动态加权聚合，得到动态加权梯度更新算法。在每一轮通信中，将执行边缘节点选择、动态权重计算和刷新操作。这种设置允许模型及时减少表现不佳的局部模型的影响。其具体流程如下：

（1）初始化：云服务器向所有参与训练的边缘节点发送 W^0（全局模型的初始参数）、ρ（学习率）、B（分批次训练时样本大小）、E（样本数量）、R（通信轮数总数）、β（检测精度阈值）、f（损失函数）。边缘节点将其通信状态设置为 True，允许将参数上传到中央云服务器。

（2）局部模型更新和训练：边缘节点收到上一轮全局模型参数 W^{r-1} 后，开始模型更新过程。即边缘节点将本地的模型参数 W_c^{r-1} 更新为 $\boldsymbol{W}^{r-1}$。然后使用自己的本地训练数据集 D 来训练本地检测模型，并得到本轮的模型参数 W_c^r。

（3）过滤上传局部模型参数：在这一阶段，主要通过阈值 β 过滤掉一些对全局模型无用甚至不利的局部模型。具体来说，在每一轮训练中，边缘节点首先计算本地异常检测模型的准确率 A_c^r 和选取特征在模型构建时的权重 N_c^r。如果本地模型中存在 $A_c^r < \beta$，则将其边缘节点通信状态设置为 $False$，这些边缘节点暂停与中心服务器的通信，并且不上传模型参数。当 $A_c^r \geqslant \beta$ 时，将参数集 $P_c^r = \{N_c, W_c^r, A_c^r\}$ 打包上传到中心服务器中，然后边缘节点结束更新进程。关于 β，本文参考多个局部模型经过少

数轮训练后,在测试集中检测准确率的浮动范围,并选择平均范围作为 β 值。如果 β 过低,模型过滤效果不明显,如果 β 过高,好的局部模型会被过滤掉。

(4)模型参数的动态加权聚合:中心服务器接收状态为 True 的边缘节点发送的参数,将 P_c^r 组成一个数组 S^r。对于 S^r 中的边缘节点,进行如下加权聚合操作:云服务器先计算平均准确率 A_r;然后根据公式(2)计算贡献率 μ_c^r;根据各边缘节点的准确率计算聚合权重 λ_c^r,见公式(3)。接下来,结合贡献率、聚合权重和模型参数,由公式(4)计算出不同的聚合参数合并后,得到的全局模型参数 W^r。最后,云服务器将 W^r 发送给所有边缘节点。接收到 W^r 后,各个边缘节点将通信状态更改为 True。

$$\mu_c^r = \frac{N_c^r}{\sum_{c \in S^r} N_c^r} \tag{2}$$

$$\lambda_c^r = \frac{e^{A_c^r}}{\sum_{c \in S^r} e^{A_c^r}} \tag{3}$$

$$W^r = \sum_{c \in S^r} \frac{\mu_c^r \times \lambda_c^r}{\sum_{c \in S^r} e\mu_c^r \times \lambda_c^r} \times w_c^r \tag{4}$$

(5)停止学习:在重复步骤(2)~(4)的 R 次后,得到最终的检测模型。

4 仿真结果与实验分析

4.1 实验环境

为了评估所提出的异常检测算法的性能,本文在 TensorFlow 平台上进行了仿真验证,台式机配置为 Intel Core 13400(6 核)CPU,运行频率为 2.50 GHz,安装内存为 128 GB。采用 Python 编程语言,并使用到 Numpy、Pandas 和 Scikit-learn 库。

4.2 数据集

实验采用 NSL-KDD 数据集和 N-BaIoT 数据集。NSL-KDD 数据集是开发 IDS 最常用的数据集之一。是 KDD'99 数据集的增强版,KDD'99 数据集在大量重复记录和数据集不平衡方面存在一些缺陷,而 NSL-KDD 数据集可以克服这些缺点[12]。数据集中每种类型样本的数量见表 2。N-BaIoT 数据集包含来自 9 个商业 IoT 设备的各种真实数据样本,这些数据中包括大量僵尸网络和良性网络流量。每个设备都受到 BASHLITE 或 Mirai 病毒的感染,并带有一些常规实例[16-17]。每个设备都包含足够的攻击记录和具有 115 个特征向量的常规实例,表 3 给出了每种设备的样本信息。

表 2 NSL-KDD 数据集

Table 2 NSL-KDD dataset

标签	训练集数据	测试集数据
DOS	41 334	4 592
Normal	60 608	6 734
Probe	10 490	1 165
R2L	895	99
U2R	46	5

表 3 N-BaIoT 数据集

Table 3 N-BaIoT dataset

设备 ID	设备名称	正常流量	异常流量	维数
i	Danmini 门铃	49548	968750	115
ii	Ecoobee 恒温器	13113	822763	115
iii	Ennio 门铃	39100	316400	115
iv	Philips 婴儿监视器	175240	923437	115
v	网络摄像头 PT-737E	62154	766106	115
vi	网络摄像头 PT-83	98514	729862	115
vii	Samsung 摄像头	52150	323072	115
viii	摄像头 XCS-1002	46585	816471	115
ix	摄像头 XCS-1003	19525	831298	115

4.3 评价指标

与其他方法类似,本文使用相同的指标来评估算法的性能。具体而言,使用真阳性(TP)、真阴性(TN)、假阳性(FP)和假阴性(FN)来分别表示正确分类为攻击、正确分类为正常、错误分类为攻击和错误分类为正常的样本数量[18-19]。使用这些数据,可以得到查准率 P、查全率 R、准确率 ACC,和 $F1$ 分数,以衡量算法在执行异常流量检测时的性能。计算公式如式(5)~式(8)所示:

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{8}$$

其中,ACC 表示在整个数据集上正确分类的样本数量;R 表示在数据集中攻击的总样本中被正确检测为攻击的样本数量;$F1$ 分数是精度和召回率的调和值。

4.4 结果分析

4.4.1 与集中式算法性能比较

为了评估本文所采用算法的性能,选取 DT[11]、SVM[3]、XGBoost[12]和 RF[12]算法作为对比算法,在相同的数据集 NSL-KDD 上进行了实验验证。实验结果见表 4。由此可见,本文提出的算法在所有

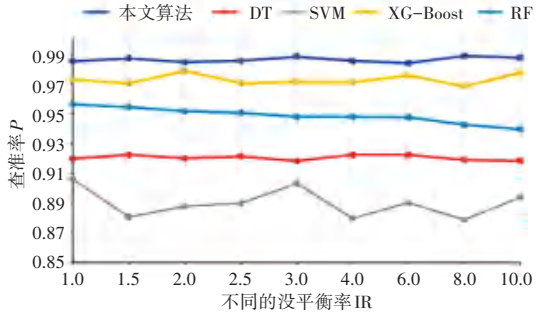
度量指标中均表现出比对照组检测方法更好的性能。查准率分别提高 6.57%、8.11%、0.86%、2.61%；查全率分别提高 1.46%、12.21%、6.9%、2.54%；准确率分别提高 3.68%、9.53%、2.82%、2.11%；*F*1 分数分别提高 0.041 2、0.101 9、0.039 4、0.025 8。

表 4　NSL-KDD 数据集上性能比较

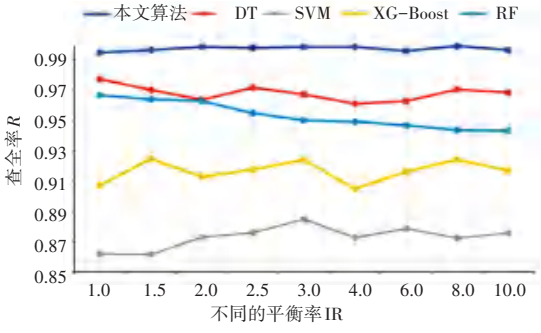
Table 4　Performance comparison on NSL-KDD dataset

算法	查准率 *P*	查全率 *R*	*ACC*	*F*1 分数
本文算法	0.983 9	0.994 7	0.989 2	0.989 3
DT	0.918 2	0.980 1	0.952 4	0.948 1
SVM	0.902 8	0.872 6	0.893 9	0.887 4
XG-Boost	0.975 3	0.925 7	0.961 0	0.949 9
RF	0.957 8	0.969 3	0.968 1	0.963 5

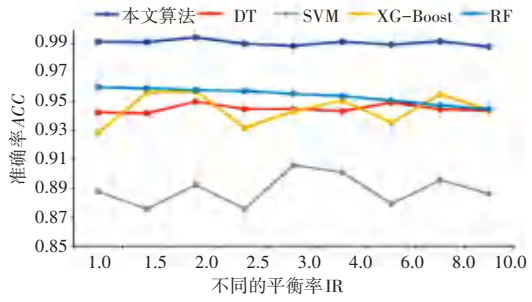
为了验证本文提出的算法的鲁棒性，通过改变原始数据集中的攻击样本数量，生成了具有不同不平衡率的数据集。这些数据集是从原始数据集中获得并随机生成的。接下来，将新的数据集输入到所提出的攻击检测方法中并进行评估，结果如图 4(a)～(d)所示。结果表明，所提出的攻击检测方法具有高准确率、低误报和较高的 *F*1 分数，优于对比算法。反映了本文所提方法通过分离攻击样本和正常样本，并在每个样本上运行 VAE 算法，缓解了异常检测中不平衡问题的挑战。该算法在不平衡数据集的性能评估中具有更好的鲁棒性。



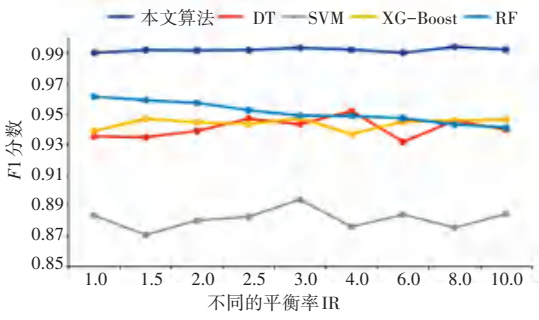
(a) 查准率对比



(b) 查全率对比



(c) 准确率对比



(d) *F*1 分数对比

图 4　不同的平衡率下 5 种算法的指标对比

Fig. 4　Comparison of indicators of five algorithms under different balance rates

4.4.2　与其他 FL 算法性能比较

本文主要与文献[15]中的对比算法(FCNN)及所提算法(RECNN)在相同的数据集 N-BaIoT 上进行了对比，其结果如图 5～图 8 所示。采用查准率 *P*、查全率 *R*、准确率 *ACC*、*F*1 分数 4 个方面评估了算法性能。结果表明，在 N-BaIoT 的 9 个设备中，FL-BVDL 算法的查准率分别提高了 2.51%、7.46%、3.17%、0.93%、2.88%、2.81%、2.31%、3.49%、2.81%；查全率分别提高 5.83%、7.99%、6.59%、7.92%、7.50%、7.08%、7.47%、6.99%、7.03%；准确率分别提高 2.99%、1.82%、4.89%、3.13%、3.38%、2.62%、2.23%、4.19%、3.85%；*F*1 分数分别提高 0.042 3、0.077 3、0.049 3、0.045 2、0.052 9、0.049 0、0.050 1、0.053 2、0.049 8。

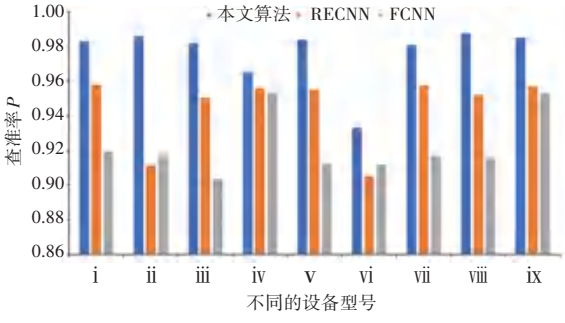


图 5　查准率 *P* 对比

Fig. 5　Comparison of precision

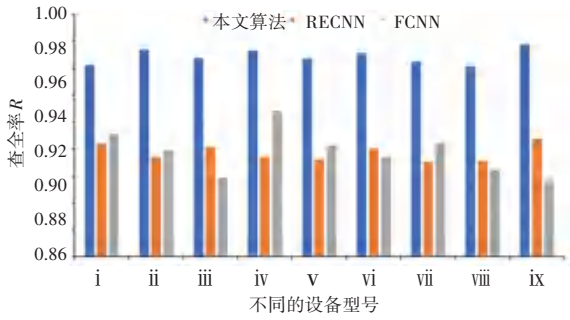


图 6 查全率 *R* 对比

Fig. 6 Comparison of recall rate

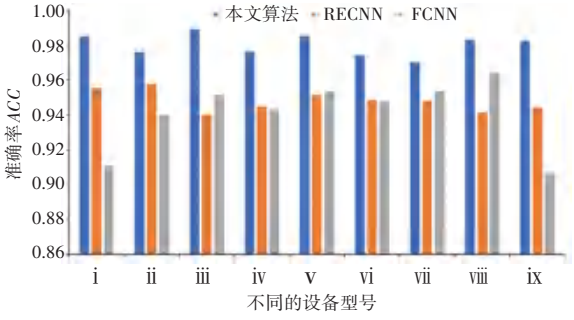


图 7 准确率 *ACC* 对比

Fig. 7 Comparison of accuracy

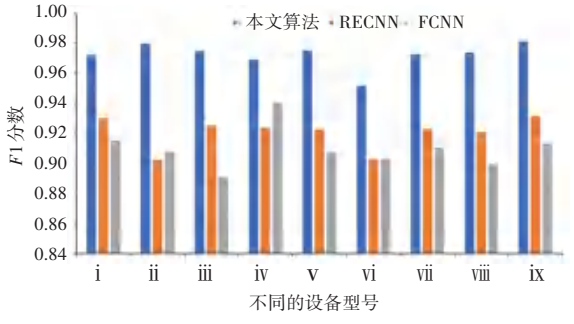


图 8 *F*1 分数对比

Fig. 8 Comparison of *F*1 score

4.4.3 模型训练时间分析

本文所提算法的模型训练时间与文献[15]中所提出的算法 RECNN 和 FCNN 进行了比较,结果如图 9 所示。在 N-BaIoT 数据集的 9 个设备中训练用时平均为 35.89 s。从图 9 中可以看出,所提出的模型与对比算法相比耗时更短,因为其在数据处理阶段采用了 Birch 聚类算法,降低了运算的数据量;同时,所提出的动态加权梯度更新算法减少了边缘节点和云服务器之间交换的大量梯度,在模型训练时能够更快收敛。

4.4.4 参数 β 值的选取

在本文提出的动态加权梯度更新算法中,主要通过阈值 β 过滤掉一些对全局模型无用甚至不利的局部模型。如图 10 所示,横坐标为 β 值,纵坐标为采用不同 β 值时对应的算法模型的查全率、查准率、准确率和 *F*1 分数指标。随着 β 值的增加,这 4 个指标继续增加,表明将一些性能差的局部节点去除后,改善了全局模型的性能,但 β 值继续增加导致算法准确率、*F*1 分数等指标变差,这是由于过滤掉大部分性能优越的本地模型,用于生成全局模型的高质量局部模型较少。基于这些数据得到的全局模型检测能力变差。从图中可以看到,当 β 值为 0.95 ~ 0.97 之间,模型的检测效果最好,本文最终选择 β 值为 0.96。

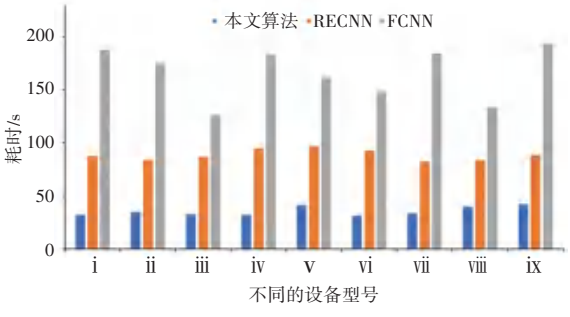


图 9 三种算法生成检测模型的训练时间对比

Fig. 9 Comparison of training time for generating detection models using three algorithms

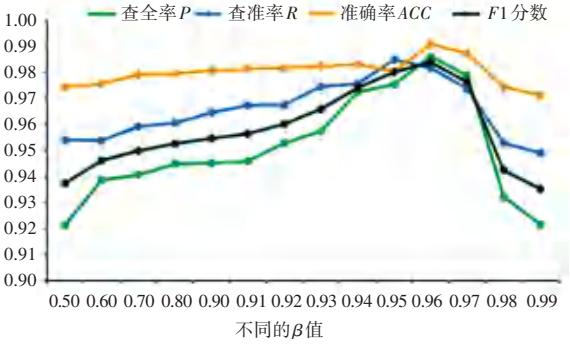


图 10 不同 β 值对模型性能影响

Fig. 10 Different β Value impact on model performance

5 结束语

本文提出了一种物联网设备异常检测算法,该算法使用变分自编码器来对特征进行变换和处理,并使用轻量级梯度提升算法来准确及时地检测异常。现有的联邦学习检测模型在边缘节点和云服务器节点之间频繁交换梯度,导致模型的通信开销过多,不适用于工业物联网环境。针对该问题,本文采用了动态加权梯度更新算法,减少了训练过程中局部模型表现不佳对全局模型的影响。同时,在数据预处理阶段采用 Birch 算法,通过减少数据训练量以及边缘设备和云聚合器之间交换的特征参数来减少计算开销和通信开销。在 NSL-KDD 和 N-BaIOT 两个数据集上对该算法进行了验证,并将其性能与

现有方法进行了比较。实验结果表明,本文所提算法在相同的性能下模型训练时间能够降低 63.08%。下一步工作将考虑异常检测系统容易受到对抗性攻击行为,利用联邦学习技术降低对抗样本对检测性能的干扰。

参考文献

[1] THAR B, MUHAMMAD A, ÁINE M, et al. A secure fog-based platform for SCADA-based IoT critical infrastructure[J]. Software: Practice and Experience, 2020, 50(5): 503-518.

[2] JAVIER N, AITOR B, JOSE A. A review of federated learning in intrusion detection systems for IoT[J]. IEEE Internet of Things Journal, 2022, 11(8): 2661-2674.

[3] ZOLANVARI M, TEIXEIRA A, GUPTA L, et al. Machine learning-based network vulnerability analysis of industrial internet of things[J]. IEEE Internet Things Journal, 2019, 6(4): 6822-6834.

[4] LIU J X, MICHELE N, JOHAN F, et al. Adversarial machine learning: A multilayer review of the state-of-the-art and challenges for wireless and mobile systems[J]. IEEE Communications Surveys Tutorials, 2022, 24(1): 123-159.

[5] LIU J, BAI J P, SUN B. Improved LSTM-based abnormal stream data detection and correction system for internet of things[J].IEEE Transactions on Industrial Informatics, 2022, 18(2):282-296.

[6] BADR L, HUSSEIN S. Optimized deep autoencoder model for internet of things intruder detection[J]. IEEE Access, 2022, 6(4): 1157-1180.

[7] MOHAMED A, OTHMANE F, DJALLEL H, et al. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT Applications: Centralized and federated learning[J]. IEEE Access, 2022, 10(4): 1281-1306.

[8] MUHAMMAD W, KAMLESH K, ASIF A, et al. Botnet attack detection in internet of things devices over cloud environment via machine learning[J]. Concurrency and Computation: Practice and Experience, 2021, 5:18-33.

[9] ZHANG C Z, CHEN Y L, YANG M, et al. A novel framework design of network intrusion detection based on machine learning techniques[J]. Security and Communication Networks, 2021, 19(1):58-73.

[10] SEBASTIAN G, AGUSTIN P. Iot-23: A labeled dataset with malicious and benign iot network traffic[J]. IEEE Internet of Things Journal, 2020, 3(4):1822-1834.

[11] BAKHTIAR F, PRAMUKANTORO E S, NIHRI H. A lightweight IDS based on j48 algorithm for detecting DoS attacks on IoT middleware[C]//Proceedings of 2019 IEEE 1st Global Conference on Life Sciences and Technologies (LifeTech). IEEE, 2019.DOI:10.1109/LifeTech.2019.8884057.

[12] VERMA A, RANGA V. Machine learning based intrusion detection systems for iot applications[J]. IEEE Internet Things Journal, 2021, 8(1): 2287-2310.

[13] HUONG T, BAC T, LONG D, et al. Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach[J]. Computers in Industry, 2021, 13: 103-132.

[14] ZHAO R, YIN Y, SHI Y, et al. Intelligent intrusion detection based on federated learning aided long short-term memory[J]. Physical Communication, 2020, 42: 101-157.

[15] IDRIS Z, HARSHA K, OMAR A. Effective and resource efficient deep neural network for intrusion detection in iot networks[C]// Proceedings of the 8th ACM on Cyber-Physical System Security Workshop. IEEE, 2022: 41-51.

[16] TAHER M, MOHAMMAD K, MUHAMMAD T, et al. IoT for smart cities: Machine learning approaches in smart healthcare: A review[J]. Future Internet, 2021, 13(8): 218-235.

[17] AL-MARRI N, BEKIR S, MOHAMED M. Federated mimic learning for privacy preserving intrusion detection[C]//Proceedings of International Black Sea Conference on Communications and Networking.Nanjing University. IEEE, 2020: 41-42.

[18] LI B, SONG J, LU R,et al. DeepFed: Federated Deep learning for intrusion detection in industrial cyber-physical systems[J]. Transactions on Industrial Informatics, 2021, 17(8): 5615-5624.

[19] MOTHUKURI V, KHARE P, PARIZI R M, et al. Federated-learning-based anomaly detection for IoT security attacks[J]. Internet of Things Journal, 2021, 9(4): 2545-2554..