文章编号:1671-4598(2021)07-0223-06 DOI:10.16526/j.cnki.11—4762/tp.2021.07.044 中图分类号:TP309 文献标识码:A

边缘计算中改进 ELM 的高效入侵检测算法

李忠成，高惠燕，张文祥
(浙江万里学院 智能控制技术研究所，浙江 宁波 315100)

摘要：边缘计算将云计算扩展到网络边缘，在解决了云计算时延高、移动性差和位置感知弱等缺陷的同时也带来了诸多安全问题；针对边缘计算网络开放性、异构型和节点资源受限等特点，研究设计具有6层结构的通用边缘计算入侵检测系统，并在此模型架构上提出了一个边缘计算入侵检测方案，基于该方案提出了一种适用于边缘计算部署的改进极限学习机的入侵检测算法TSS—ELM，TSS—ELM增加了云服务器训练样本筛选环节来优化机器学习中的外权，从而对边缘节点数据实现高效的入侵检测；仿真实验结果和分析表明，该算法在准确性、时间依赖性、鲁棒性和误报率方面与其他现有算法相比具有更优异的性能。

关键词：边缘计算；云计算；ELM；入侵检测；样本筛选

An Efficient Edge Computing Intrusion Detection Algorithm Based on Improved ELM

LI Zhongcheng, GAO Huiyan, ZHANG Wenxiang
(Institute of Intelligent Control Technology, Zhejiang Wanli University, Ningbo 315100, China)

Abstract: Edge computing extends the influence of cloud computing to network edge. It solves the problems of high delay, poor mobility and weak position perception in cloud computing, and it also brings many security problems. According to the characteristics of open, heterogeneous and limited node resources of edge computing network, this paper studies and designs an architecture of edge computing intrusion detection system with 6—layer structure, and an edge computing intrusion detection scheme is proposed. Based on this scheme, an intrusion detection algorithm TSS—ELM is proposed, which is suitable for edge computing deployment. This algorithm introduces the process of training sample selection and optimizes the external weight in machine learning, so as to achieve efficient intrusion detection for edge node data. Simulation results and analysis show that the algorithm has better performance in accuracy, time dependence, robustness and false alarm rate than other existing algorithms.

Keywords: edge computing; cloud computing; extreme learning machine; intrusion detection; training sample selection

0 引言

随着移动互联网、云计算和物联网技术的高速发展，笔记本、手机和可穿戴设备等智能移动终端不断涌现，海量数据和智能应用迅速填充着人们的生活，人们对快速实时稳定的通信要求越来越高[1]。在计算能力、能耗、实时性、数据安全和隐私等方面，集中式的云计算模式已经无法满足网络需求[2]。边缘计算应运而生，作为对云计算的延伸和补充，边缘计算将云计算扩展到了网络边缘，可以通过部署在不同位置的边缘服务器与终端设备直接连接，从而降低数据处理延迟，解决云计算时延高、移动性差和位置感知弱等问题[3]。但同时由于边缘计算网络通信的开放性也带来了边缘设备和用户数据的安全问题，边缘节点分布式部署的特点，使其很容易遭受到外部用户的非法入侵；边缘计算网络异构型的特点，使其受到入侵的情态极具多样化和复杂化；作为边缘节点集群管控中心的云服务器也会受其影响，面临更多的入侵威胁[4]。因此，构建一个面向边缘计算网络的入侵检测模型，并依此研究一种行之有效的入侵检测算法，在保障边缘计算网络安全、推进边缘计算的更广泛应用方面具有重要意义。本文聚焦边缘计算环境下的安全防御相关模型及算法展开研究，提出一种基于改进 ELM（极限学习机）算法的边缘计算入侵检测方案，贡献如下：

(1) 设计并架构一个具有6层结构的边缘计算入侵检测系统，系统功能包括边缘计算入侵检测到防御响应过程中的全部模块，通过监控边缘网络、节点状态和用户行为等，来实现对用户非法行为和外部入侵进行检测和防御。

(2) 以入侵检测系统架构为基础提出一个边缘计算入侵检测方案，并基于该方案提出一种适用于边缘计算部署的改进极限学习机的入侵检测算法，通过引入样本筛选的过程以优化学习过程中的外权，对边缘节点的数据实现高效的入侵检测。

(3) 通过与传统 ELM、BP 神经网络、SVM 和 CVM—ELM 算法比较准确率、训练时间、鲁棒性、误报率，证明

收稿日期:2020-12-17； **修回日期**:2021-01-08。
基金项目:国家自然科学基金(61073074);浙江省自然科学基金(LY19F020001);浙江省科技计划项目(LGG18F020001)。
作者简介:李忠成(1977-),男,吉林舒兰人,硕士,讲师,主要从事无线通信与网络安全方向的研究。
引用格式:李忠成,高惠燕,张文祥.边缘计算中改进 ELM 的高效入侵检测算法[J].计算机测量与控制,2021,29(7):223-228,234.

该算法是一种适用于边缘计算网络应用的入侵检测算法。

1 相关工作

边缘计算虽然能够有效降低云中心的计算负载，从而缓解带宽和延迟压力，但是随着边缘节点的不断增加和边缘设备智能化程度的不断提高，边缘计算也面临着包括虚假数据中心、数据篡改窃取以及拒绝服务攻击等一系列用户非法行为或外部入侵安全问题。面对这些问题，国内外研究人员提出了不同的解决方法。文献［5］提出了一种基于安全级别的防御方案，方案中不同的安全级别采用不同的数据包检查算法，并强制执行不同的插件，分析来自内部用户非法行为的攻击威胁[5]。文献［6］提出了零和随机博弈模型，模拟攻击者和用户之间的信息动态交互行为，通过推导模型的纳什均衡分析影响用户最优防御策略的主要因素[6]。文献［7］提出了一种分布式拒绝服务攻击分类算法来防御云数据中心的安全威胁，并对响应时间、负载开销等指标进行了实验分析[7]。文献［8］提出了一种利用虚拟机状态识别攻击方案，通过信息熵监视虚拟机状态，实验表明该方案能够识别拒绝恶意用户利用云资源发起的服务攻击，且具有良好的实时性和准确性[8]。文献［9］提出了一种分布式拒绝服务攻击缓解方案，该方案集成了一个可编程的网络监控，能够对恶意攻击进行检测，从而实现快速的入侵响应[9]。文献［10］将外部攻击者和内部用户之间的信息交互模拟为开环的微分博弈模型，该模型为后续抵御内部攻击的防御方案设计提供了一定的理论基础[10]。文献［11］通过构建入侵检测防御系统进行数据包综合分析，提出了一种基于嵌入式马尔可夫链模型，该模型在云网络安全服务的防御性和检测性方面取得了一定的成效[11]。文献［12］基于博弈理论提出了一种位置隐私系统，通过对网络中用户移动行为的分析研究，实现在最小化开销的同时最大化用户的位置隐私水平，从而提高云网络服务的安全性[12]。文献［13］提出了一种轻量级安全可持续负载均衡算法，利用该算法能够高效地找出低负载目标边缘数据中心，进而通过低负载中心的检测验证来避免外部攻击[13]。文献［14］提出了一种拒绝服务攻击的安全防御模型，该模型通过多接入边缘计算技术来处理来自终端设备的可疑流量，利用生成的本地化防御策略实现拒绝服务攻击[14]。文献［15］针对攻击者恶意破坏数据包转发问题提出了一种集中检测模型，利用分组丢弃概率的方法监控下行链路信道，该模型不需要对任何数据进行训练，能够有效缓解系统资源的额外损耗[15]。文献［16］提出了一种基于马尔可夫博弈的安全模型，利用马尔可夫链模拟数据交互的变化过程，并通过构造效用函数分析各类参数对数据用户选择最优保护策略的影响，从而提高网络中用户数据的安全性[16]。

综合上述研究，在边缘计算中确实存在着一系列安全入侵问题，并且现有研究大多基于传统算法，缺少对边缘节点资源受限和边缘网络复杂性高等特点的考虑，因此本文结合边缘计算网络和边缘数据中心的特点及其面临的安全威胁，设计一种高效的入侵检测模型和适用于边缘计算网络的入侵检测算法。

2 模型构建

与传统网络相比，边缘计算网络具有分布式、异构性和节点资源受限等特点，边缘计算网络中的边缘节点和云服务器都很容易受到大量异构设备接入带来的安全威胁。设计构建边缘计算入侵检测系统的目的是实时监控边缘计算网络、终端设备行为和边缘节点状态等，并检测分析边缘网络中外部攻击者的入侵企图、内部用户的越权行为和系统网络的安全缺陷，并根据检测结果进行合理的入侵响应和统计管理。系统功能特点包括数据采集分析、实时监控管理、智能动态响应、优化资源分配、操作简单和维护便捷等。针对边缘计算特殊的网络结构，为了确保边缘计算的安全性和网络服务的高效性，结合边缘计算的特点研究构建一种通用入侵防御系统模型，如图 1 所示，该模型根据边缘计算网络系统的入侵防御特点、功能及流程分为六层，按照边缘网络数据流向依次是物理层、网络层、数据层、检测层、分析层和管理层，系统功能模块包括捕获数据包、数据清洗过滤、入侵检测、信息分析与知识挖掘、入侵响应策略、存储与通信、数据统计等。

2.1 物理层

物理层包括接入边缘计算网络的各种用户智能设备或移动终端，这些具有多元性和异构性的用户设备只享受系统服务，它们之间并不互相提供服务。这些设备包括智能手机、掌上电脑、个人 PC、智能穿戴、智能家电、无人机、传感器和车联网中的车载设备等。由于边缘计算物理层的异构性特点，这些设备会通过各自不同的网络协议接入到系统中的边缘节点。物理层是边缘计算系统网络中一切数据的源头，而在物理层中的大量用户设备又无法区分其合法性，因此边缘计算网络所面临的安全威胁绝大部分都来自于物理层。

2.2 网络层

网络层的作用是为不同的边缘计算网络协议提供通信链接服务，该层负责接收从物理层传输过来的数据，并对数据进行封包和传输处理。网络层突出体现了边缘计算网络异构性和融合性的特点，边缘节点能够实现物理层异构的用户设备连接，原因就在于对多种通信协议的链接支持。边缘计算网络支持的协议有 Wifi、5 G、ZigBee、蓝牙以及 M2M 和以太网协议等。

2.3 数据层

数据层的作用是进行数据采集和处理。数据有来自物理层各类设备产生的数据，其中既有合法用户数据也有非法用户数据，还有网络层产生的数据，包括通信协议、设备连接信息和活动状态等。该层主要完成数据的采集、存储、清洗、过滤和预处理等。该层是入侵检测的首要层，各项功能主要在边缘节点中完成，边缘节点由资源受限的服务器或设备组成，本地能够存储和处理数据，是最接近

物理层的服务节点，在与云服务器通信时，边缘节点不用将采集数据全部传输给云，只要将本地处理后的数据按需上传即可。

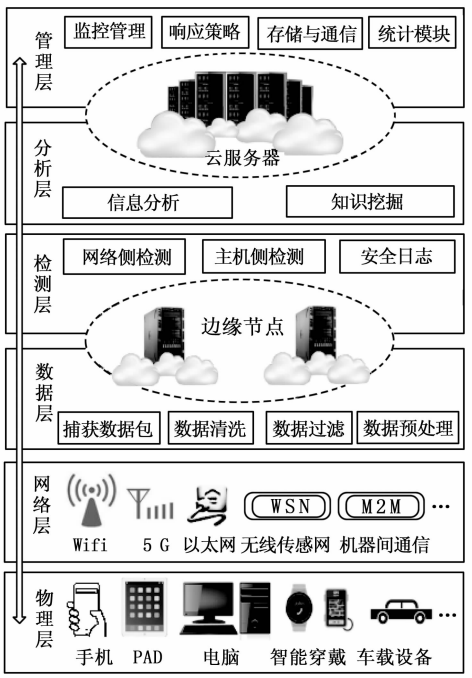

图 1　边缘计算入侵防御系统模型

2.4　检测层

检测层是入侵检测的核心层，检测层的任务也在边缘节点中完成的。数据层采集到的数据在系列处理之后数据量极大，而且这些海量数据多数属于正常数据，只有极少数数据是表征非法入侵行为发生的异常数据。检测层的功能就是通过数据分类检测从海量数据中找到这样的异常数据。在边缘节点中检测层会对预处理的物理层终端设备和网络层连接数据进行检测，利用入侵检测算法对数据进行分析判断；并对边缘节点的主机状态进行监控；对网络数据包协议和日志进行记录管理；定时向云服务器发送检测结果和日志记录。

2.5　分析层

分析层位于系统云服务器，任务是分析来自边缘节点检测层的处理结果和日志记录数据，将结果信息进行数据挖掘并生成应用服务。分析层对所有连接的边缘节点进行安全状态监控和异常状态分析，生成并存储每个边缘节点的安全/异常状态分析报告，一旦发现有异常状态的边缘节点，就对此类边缘节点数据展开追踪。

2.6　管理层

管理层也位于系统云服务器，该层的任务是对分析层生成的应用服务进行有效实施和对处于异常状态的边缘节点实时响应。该层的主要功能模块有监控管理、入侵响应策略、存储与通信管理、统计管理等，能够实现对边缘节点异常状态的统一监控和管理，计算边缘节点异常状态，根据分析结果实时自动响应追踪策略，对异常边缘节点信息和日志进行存储和管理，对非法入侵行为进行证据记录和追踪溯源。

3　算法设计

3.1　入侵检测方案

边缘计算中的入侵行为就是边缘计算网络中一切可能危害到数据机密性、真实性或系统可控性、可靠性的行为。结合上述入侵防御系统的六层架构，边缘计算中的非法入侵行为主要来自物理层，包括端口扫描、拒绝服务、本地未授权用户访问和远程未授权用户访问等入侵威胁。因此边缘计算中的入侵检测主要就是在边缘节点中对捕获的物理层数据进行处理、分析和检测。

入侵检测的核心问题是入侵检测分类算法，由于边缘节点的地理分布式、网络异构性和资源受限等特点，传统的入侵检测算法并不适合直接部署，为了合理利用边缘计算系统资源、有效地执行入侵检测任务，本节设计了一种云服务器与边缘节点协同合作的入侵检测方案，更好地提高入侵检测的可用性、准确性和高效性。

该方案考虑到边缘节点也具备一定的计算能力和存储能力，所以在边缘节点中部署非法入侵检测分类器，而数据量较大的总训练样本则存储在计算和存储能力强大的云服务器中，同时云服务器还存储了训练集数据的筛选规则，云服务器把样本按规则选择出的结果交给边缘节点训练。从而综合云服务器和边缘节点的计算存储协作完成对非法入侵的异常检测。这样能够确保边缘节点获得的数据更符合本地特性，能够更有效地提高系统检测能力。

该方案主要流程如图 2 所示。首先物理层的各种异构终端设备通过各自的网络协议接入到边缘计算网络；然后系统云服务器对总训练数据样本集进行存储管理，并按照样本筛选规则进行样本选择；之后云服务器将筛选后的训练样本集分发给边缘节点进行本地存储；各边缘节点再针对存储的训练集执行入侵检测的训练过程；最后在边缘节点上完成非法入侵行为的检测，并记录安全日志存储在该边缘节点中。

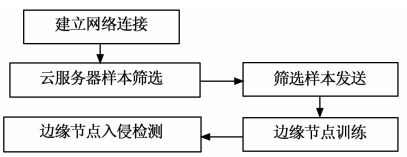

图 2　边缘计算入侵检测流程

该方案充分利用了边缘计算中云服务器和边缘节点资源，通过各自在入侵检测的训练过程和检测过程承担不同职能进行分工协作，能够有效提高系统资源利用率、降低网络中的通信开销、减少云服务器计算负载，通过安全日志分存也能有效降低系统风险。

3.2　入侵检测算法

在上述云服务器与边缘节点协同合作的入侵检测方案中，边缘节点需要在本地部署入侵检测算法以完成入侵检

测任务。考虑到边缘计算复杂的网络环境和大规模异构用户设备的动态威胁，必须通过灵活动态的训练数据集进行合理有效地实时数据训练。同时考虑到边缘节点资源受限的特点，现有的诸多入侵检测算法训练时间又过长，亟需设计一种适合边缘节点应用的准确率高、训练时间短、鲁棒性好的轻量级入侵检测算法[17]。ELM（极限学习机）算法求解直接，仅需求解输出权重，其学习过程易于在全局极小值收敛，该算法具有简单易用、训练参数少、学习速度快、泛化性能强等优点，非常适用于边缘计算中的入侵检测。

3.2.1 传统 ELM 算法

ELM 算法是 2004 年由南洋理工大学的 Guang－bin Huang 等提出的，最初 ELM 是为监督学习问题设计的[18]，但之后其应用范围不断推广，还包括以聚类为代表的非监督学习以及具有表征学习能力的变体和改进算法等。ELM 算法比传统 BP（反向传播）算法的学习模式更加简单有效，学习速度也更快，相比传统的 ANN（人工神经网络）也更方便和实用。

在 ELM 算法中，针对训练样本 $(\bar{x}, t)$，隐藏神经元数量为 K 的单隐层前馈神经网络输出函数为：

$$F_K(x_j) = \sum_{i=1}^{K}\beta_i G(a_i,b_i,x) = \sum_{i=1}^{K}\beta_i g(a_i \cdot x + b_i) \tag{1}$$

其中：a_i 为输入层和隐藏层第 i 个神经元之间的内权向量，b_i 为第 i 个神经元之间的偏置，β_i 为连接网络中第 j 个隐藏层和输出层第 i 个神经元之间的外权向量，$G(a_i, b_i, x)$ 为对应样本 x 第 j 个网络隐藏层输出的针对加法型的隐藏层节点，$g(a_i \cdot x+b_i)$ 为隐藏层的激活函数。

由于 N 个互异的数据样本满足 $\{(x_i, t_i)\}_{i=1}^{N} \subset R_n \times R_m$，那么假设有网络隐藏层神经元数量为 K，它能够以零误差逼近该数据样本，则存在 a_i，b_i 和 β_i（$i=1, 2, \cdots, K$），使得：

$$F_K(x_j) = \sum_{i=1}^{K}\beta_i G(a_i,b_i,x) = t_j, j = 1,2,\cdots,N \tag{2}$$

写成矩阵形式为：

$$\boldsymbol{H\beta} = \boldsymbol{T} \tag{3}$$

其中：$\boldsymbol{H}$ 是隐藏层输出矩阵，$\boldsymbol{H}(a_1, \cdots, a_K, b_1, \cdots, b_K, x_1, \cdots, x_K) = \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_K, b_K, x_1) \\ \vdots & \cdots & \vdots \\ G(a_1, b_1, x_N) & \cdots & G(a_K, b_K, x_N) \end{bmatrix}$；

$\boldsymbol{\beta}$ 是输出权重矩阵，$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_K^T \end{bmatrix}_{K\times m}$，$\boldsymbol{T}$ 是期望输出权重矩阵，$\boldsymbol{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N\times m}$。

因为大部分时候隐藏层节点个数 K 都远远小于训练数据样本的个数 N，所以该神经网络以零误差逼近训练样本很难实现，会造成 N 个互异数据样本的网络输出与实际输出之间产生误差。因此式（3）应该改为：

$$\boldsymbol{H\beta} = \boldsymbol{T} + \boldsymbol{E} \tag{4}$$

其中：$\boldsymbol{E} = \begin{bmatrix} e_1^T \\ \vdots \\ e_N^T \end{bmatrix}_{N\times m}$。再考虑到诸如训练样本维度降低或采集不全等情况，则可以定义平方损失函数为：

$$\boldsymbol{J} = \sum_{j=1}^{N}\beta_i G(a_i,b_i,x_j) - t_j = (\boldsymbol{H\beta} - \boldsymbol{T})^T(\boldsymbol{H\beta} - \boldsymbol{T}) \tag{5}$$

那么针对数据样本的训练问题就可以转化为求解平方损失函数的最小二乘解 β，利用 MoorePenrose 广义逆可以得出：

$$\beta = \arg\min_{\beta}\|\boldsymbol{H\beta} - \boldsymbol{T}\| = \boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{6}$$

式中，$\boldsymbol{H}^{\dagger}$ 表示隐藏层输出矩阵 $\boldsymbol{H}$ 的广义逆，可以用正交法计算求得。

3.2.2 改进 ELM 算法（TSS－ELM）

为了减少边缘节点的样本训练时间，更好地适应边缘计算的异构网络环境和动态训练过程，本节设计一种改进的训练样本筛选－极限学习机算法（TSS－ELM），该算法能够根据每个边缘节点的网络特性和训练特点进行数据样本的选择。TSS－ELM 算法的特点是在云服务器中增加训练数据样本的选择过程，并在边缘节点上通过算法部署进行异常检测和发现入侵。

TSS－ELM 算法把网络云服务器中存储的总数据样本集 S_n 分成两部分，分别是为边缘节点分配的训练样本集：

$$S_{nf}^{f} = \{s_j^f = (x_j^f, y_j^f)\}_{j=1}^{nf} \tag{7}$$

和备选样本集：

$$S_{nc}^{c} = \{s_j^c = (x_j^c, y_j^c)\}_{j=1}^{nc} \tag{8}$$

其中：$S_{nf}^{f} \cup S_{nc}^{c} = S_n$，$S_{nf}^{f} \cap S_{nc}^{c} = \varnothing$。进行样本筛选的目的是从 S_n 中筛选出 S_{nf}^{f}，使得边缘计算网络通过上述传统 ELM 算法利用 S_{nf}^{f} 学习以满足 $J(a_i, b_i, \beta) \leqslant \sigma$，其中 $\sigma \in (\min J(a_i,b_i,\beta))$ 是预先确定的网络系统性能指标上限。定义：

$$U^c = \{u_j^c \mid u_j^c = |t_j^c - F(x_j^c, a_i, b_i, \beta)|\}_{j=1}^{nc} \tag{9}$$

U^c 是 S_{nc}^{c} 中训练样本实际输出和网络输入输出差值的绝对值所组成的数据集合。再定义：

$$S_z^c = (x_m^c, y_m^c) \tag{10}$$

S_z^c 是 $S_n^c c$ 中对应 U^c 最大值元素的某元素。首先数据样本集初始化为：$S_{nf}^{f} = \varnothing$，$S_n^c c = S_n$，边缘计算网络参数初始化为：$a_i = \varnothing$，$b_i = \varnothing$，$\beta = \varnothing$，然后采用下面规则进行学习：

$$S_{nf}^{f} = S_{nf}^{f} \cup \{s_z^c\} \tag{11}$$

$$S_{nc}^{c} = S_{nc}^{c} - \{s_z^c\} \tag{12}$$

$$a_i = a_i + \frac{\{x_z^c\} - \{x_z^c\}_{\min})}{(\{x_z^c\}_{\max} - \{x_z^c\}_{\min}},$$

$$b_i = b_i + \frac{\{x_z^c\} - \{x_z^c\}_{\min}}{(\{x_z^c\}_{\max} - \{x_z^c\}_{\min}} \tag{13}$$

按照式（6）利用 S_{nf}^{f} 和 a_i、b_i 计算更新最优外权向量 β，使 $J(a_i, b_i, \beta) = \sigma$；再更新 S_{nf}^{f}、S_{nc}^{c}、a_i、b_i、β；通过若干次迭代后满足 $J(a_i, b_i, \beta) \leqslant \sigma$。

在 TSS－ELM 算法中网络隐藏层激活函数采用 Sig-

moid 变换函数：

$$G(a_i,b_i,x)=\frac{1}{(1+e^{(-a_i x+b_i)}}\tag{14}$$

TSS－ELM 算法的详细流程如图 3 所示。

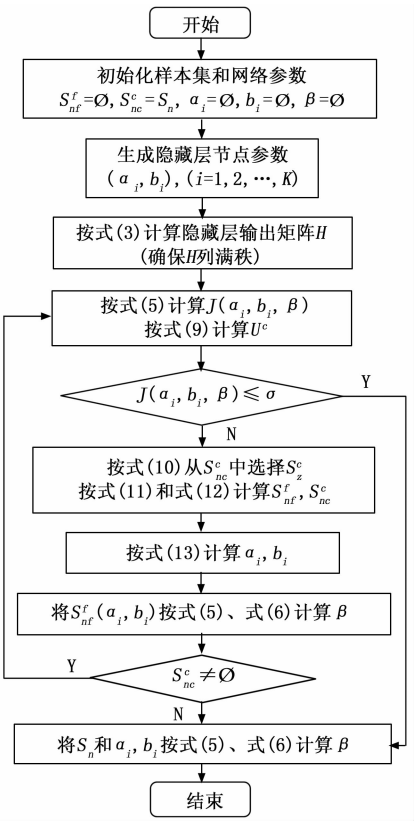

图 3　TSS－ELM 算法流程图

结合 TSS－ELM 算法流程能够发现，云服务器在为边缘节点选择训练样本的过程中，计算 $J(a_i, b_i, \beta)$ 占用了绝大部分时间，因此如果设 $t(J)$ 为求解 $J(a_i, b_i, \beta)$ 的平均时间，同时设云服务器与边缘节点数据传输延迟为 t'，t'远远小于 $t(J)$，则边缘节点学习时间 t 满足：

$$t\approx n_K\cdot t(J)\tag{15}$$

4　实验分析

4.1　实验环境及数据集

实验环境：操作系统采用 Windows10 64 位，CPU 采用 Intel core i7－77003（3.6 GHz），内存采用 16 GB DDR，算法实现程序采用 Matlab。

实验采用 KDD CUP99 数据集进行仿真测试，KDD CUP99 是模拟美国空军局域网的网络环境建立的网络测试数据集合，已经被广泛应用在各类网络入侵检测实验中[19]。KDD CUP99 异常类型分为四类，包括 39 种攻击类型，其中训练集中有 22 种，测试集中有 17 种。KDD CUP99 对每一条网络连接记录收集 41 个字段，包括 TCP 连接基本特征、TCP 连接内容特征、网络流量时间统计特征和网络流量主机统计特征等。每一条连接记录同时存在符号特征和离散特征，需要进行数据标准化预处理，即将类别特征转化为度量特征，再对度量特征进行归一化处理，本实验中所有记录的度量特征值统一在［0，1］范围，从而更利于各种有监督学习器对边缘网络数据进行学习和预测。

4.2　实验结果与对比分析

4.2.1　准确率分析

通过仿真实验针对本文提出的 TSS－ELM 算法与传统 ELM 算法、BP 神经网络、SVM 算法和 CVM－ELM 算法[20]分别作为部署在边缘计算环境中的入侵检测算法进行对比。设置 BP 学习速率 $lr=0.06$，动量项系数 $mc=0.9$，最大迭代次数 $epochs=5000$，$goal=0.0001$；SVM 核函数为 RBF，$gamma=0.005$，$C=10$。在 KDD CUP99 数据集中抽取数据，总训练数据样本 20 000 条，测试数据样本 10 000 条。

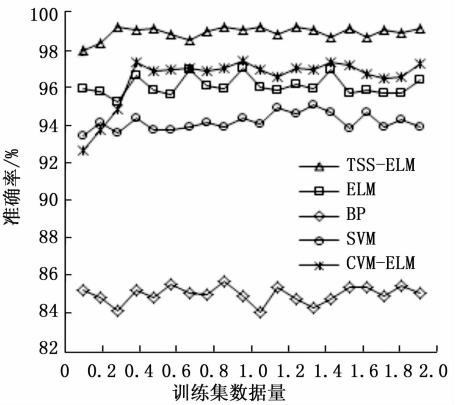

图 4　算法准确率比较

由图 4 的仿真实验结果可以看出，随着训练集数据量的增加，相比于其他 4 种入侵检测算法，TSS－ELM 表现出较高的准确率。分析其原因主要有 3 个方面：1）传统的机器学习算法不考虑输出权重大小，主要目的是达到最小训练误差，是基于梯度下降的，而 TSS－ELM 既达到最小训练误差，又达到最小权重范数；2）TSS－ELM 与传统机器学习算法相比不存在训练的过度拟合现象，也不存在求解过程的局部最小值问题，不存在学习重复率过高的问题；3）TSS－ELM 训练样本的筛选过程会使每个边缘节点获得更适于训练的最优外部权重，从而能够有效提高分类入侵检测的准确率。

4.2.2　训练时间分析

由于 BP 神经网络的训练时间过长，是其他算法的几十倍，因此在图 5 的仿真实验结果中不进行表示比较。由图 5 可以看出，本文提出的 TSS－ELM 的训练时间比 SVM 算法和 CVM－ELM 算法快得多，但比传统 ELM 算法略慢。究其原因主要是 TSS－ELM 比传统 ELM 多了一个的样本筛选的过程，但仔细对比图 5 所示训练过程，TSS－ELM 相比 ELM 的时间差异比较小，完全在可接受的范围内，因此 TSS－ELM 在入侵检测的训练时间方面也有较好的性能表现。

4.2.3　鲁棒性分析

考虑到边缘计算网络的高动态性，利用 KDD CUP99 数

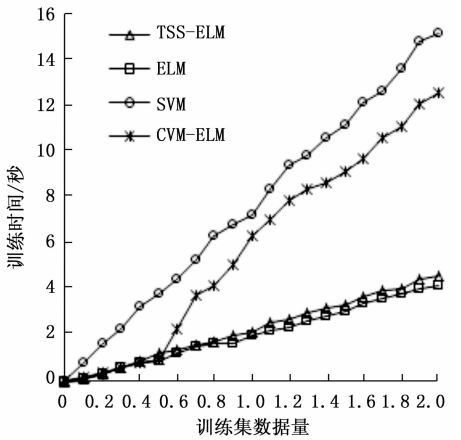

图 5 算法训练时间比较

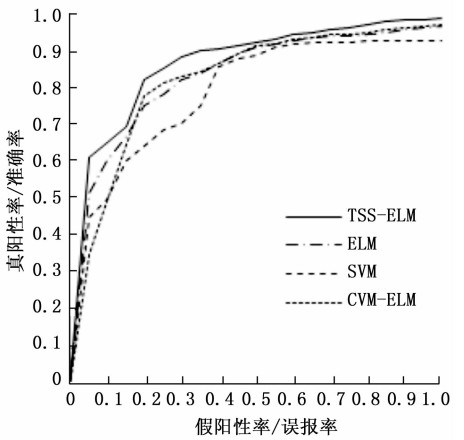

图 7 算法 ROC 曲线比较

据集模拟一个动态网络环境来分析 TSS－ELM 算法的鲁棒性，即分析 TSS－ELM 对时间统计的依赖性。因为系统物理层的原始数据未经处理是无法根据时间统计进行采集的，为了更准确地仿真 TSS－ELM 算法的有效实现，剔除 KDD CUP99 数据集中的 8 个基于时间的统计参数，即字段 24～31。由图 5 的仿真实验结果可以看出，剔除训练数据的时间统计参数后，TSS－ELM 算法的准确率几乎没有影响，而其他 4 种算法的检测精度都有不同程度的降低。因此 TSS－ELM 对时间统计属性的依赖性最小，适用于高实时性和高动态性的网络环境。

4.2.4 误报率分析

通过分析各种入侵检测算法的 ROC 曲线来对比误报率，在图 6 的仿真实验结果中，横坐标轴是假阳性率即误报率，纵坐标轴是真阳性率即准确率，由于 ROC 曲线向下与坐标轴围成的面积越大算法表现越优秀，因此，TSS－ELM 在入侵检测的误报率方面明显优于 ELM、SVM 和 CVM－ELM。

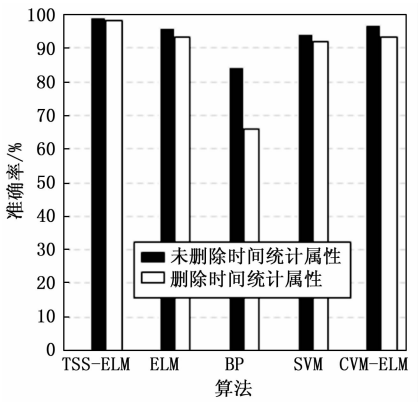

图 6 算法鲁棒性比较

5 结束语

针对边缘计算网络环境复杂、资源受限和高动态性等特点，分析了边缘计算中边缘节点、云服务器等易受到非法入侵的安全问题，完成了边缘计算入侵防御系统架构设计，并在此基础上提出了一种结合样本筛选改进极限学习机的 TSS－ELM 算法。仿真实验结果表明，与 BP、ELM、SVM 和 CVM－ELM 算法相比，TSS－ELM 算法在准确性、时间依赖性、鲁棒性和误报率等方面均表现出优异的性能，是一种适用于边缘计算环境下应用的入侵检测算法。下一步工作将在此方案的基础上研究入侵响应策略，使边缘计算入侵防御系统能够有效应对检测到的入侵行为。

参考文献：

[1] AL－FUQAHA A，GUIZANI M，MOHAMMADI，et al. Internet of things：A survey on enabling technologies，protocols，and applications [J]. IEEE Communications Surveys & Tutorials，2015，17 (4)：2347－2376.

[2] KLAS G. Edge computing and the role of cellular networks [J]. Compviter，2017，50 (10)：40－49.

[3] ZHANG J，CHEN B，ZHAO Y，et al. Data Security and Privacy－Preserving in Edge Computing Paradigm：Survey and Open Issues [J]. IEEE Access，2018，6：18209－18237.

[4] YOUSEFPOUR A，FUNG C，NGUYEN T，et al. All One Needs to Know about Fog Computing and Related Edge Computing Paradigms：A Complete Survey [J]. arXiv preprint arXiv：l 808. 2018 (5)：72－83.

[5] CHEN Z，DONG W，LI H，et al. Collaborative network security in multi－tenant data center for cloud computing [J]. Tsinghua Science and Technology，2014，19 (1)：82－94.

[6] WEI W，QIAN Z. A stochastic game for privacy preserving context sensing on mobile phone [C] //IEEE INFOCOM 2014－IEEE Conference on Computer Communications，IEEE，2014：2328－2336.

[7] SOMANI G，GAUR M S，SANGHI D. et al. DDoS attacks in cloud computing：Issues，taxonomy，and future directions [J]. Computer Communications，2015，107 (10)：30－48.

[8] CAO J，YU B，DONG F. et al. Entropy－based denial－of－service attack detection in cloud data center [J]. Concurrency and Computation：Practice and Experience，2015，27 (18)：5623－5639.

（下转第 234 页）