基于机器学习的物联网应用动态安全卸载策略

居晓琴
（江苏航运职业技术学院智能制造与信息学院，江苏南通 226010）

摘 要：针对当前物联网任务卸载算法在延迟、能耗和安全方面存在的缺点，文章提出了一种基于机器学习的物联网应用动态安全卸载策略，通过使用机器学习策略，可以在雾-物联网（Fog-IoT）环境中实现高效、安全的卸载。首先，采用Neuro-Fuzzy模型在智能网关上保护数据；其次，使用粒子群优化为IoT设备选择一个最佳Fog节点；然后，通过智能网关将任务卸载到雾节点上，如果雾节点无法处理工作负载，则将其转发到云中，敏感数据保存在私有云，非敏感数据实施动态卸载策略进行卸载。实验结果表明，提出的动态安全卸载策略最大程度地减少了延迟和能耗，比其他现有算法更具优势。

关键词：物联网；雾计算；动态安全卸载；强化学习；Neuro-Fuzzy模型
中图分类号： TP393 **文献标识码：** A

Dynamic security offloading strategy of Internet of things application based on machine learning

Ju Xiaoqin
(*School of intelligent manufacturing and information, Jiangsu Shipping College, JiangsuNantong 226010*)

Abstract: Aiming at the shortcomings of current IoT task offloading algorithms in terms of delay, energy consumption and security, a dynamic security offloading strategy for Internet of things applications based on machine learning is proposed. By using machine learning strategies, it can be used to offload efficiently and securely in Fog and Internet of Things (Fog-IoT) environment. First, the Neuro-Fuzzy model is used to protect data on the intelligent gateway. Secondly, particle swarm optimization is used to select an optimal Fog node for the IoT device, and then the task is offloaded to the fog node through the intelligent gateway. If the fog node cannot handle the workload, then Forward it to the cloud, sensitive data is stored in the private cloud, and non-sensitive data is dynamically uninstalled. The experimental results show that the proposed dynamic security offloading strategy minimizes delay and energy consumption and is superior to other existing algorithms.

Key words: Internet of Things; fog computing; dynamic security offloading; reinforcement learning; Neuro-Fuzzy model

1 引言

雾计算(Fog Computing，FC)被认为是监控物联网应用的理想平台，可用于智能城市、可穿戴传感器、医疗保健和车辆监控等多个领域[1,2]，用于减少计算的延迟和功耗。雾计算在云计算和物联网(Internet of Things，IoT)之间形成了分布式网络环境的中间层，可以提供一个连续

体来桥接丢失的链接，这些数据可以在更靠近边缘的终端处理或者推送至云上[3]。该模式可以集成在同构和异构无线网络中，充分利用资源，提高整体网络效率，减少网络流量[4]。雾计算是物联网和云之间的一个层，包括智能门、路由器、交换机和接入点等组件。

在当前社会中，由于移动设备数量众多，移动计算起着至关重要的作用。移动雾计算（Mobile Fog Computing，MFC）作为三类移动计算中的一种，有着不可替代的作用。MFC旨在减少需要转发到云端进行处理和存储的数据量。当需要实时进行大量数据处理、存储和分析时，MFC可提高系统效率，而且每个物联网设备可以将计算任务卸载到雾设备中而不是发送到云计算数据中心，从而明显地减少了传输延迟。

2 粒子群优化

PSO初始化为一群随机粒子，然后通过迭代找到最优解。在每一次迭代中，粒子通过跟踪粒子本身所找到的最优解和整个种群目前找到的最优解两个极值来更新自己的位置和速度。假设在D维的目标搜索空间中，粒子群包含有N个粒子，其中第i个粒子的位置是一个D维向量，可以表示为：

$$Y_i=\left(y_i^1,y_i^2,\ldots,y_i^D\right) \tag{1}$$

第i个粒子的速度可以表示为：

$$V_i=\left(v_i^1,v_i^2,\ldots,v_i^D\right) \tag{2}$$

第i个粒子在飞行过程中离目标函数最优解最近的位置可以表示为：

$$P_i=\left(p_i^1,p_i^2,\ldots,p_i^D\right) \tag{3}$$

整个粒子群当前时刻搜索到的最优位置可以表示为：

$$P_g=\left(p_g^1,p_g^2,\ldots,p_g^D\right) \tag{4}$$

在找到粒子最优和粒子群最优值后，PSO中的粒子根据公式(5)(6)来更新自己的速度和位置：

$$v_i^d\left(k+1\right)=\omega v_i^d\left(k\right)+c_1r_1\left(p_i^d\left(k\right)-y_i^d\left(k\right)\right)+ c_2r_2\left(p_g^d\left(k\right)-y_i^d\left(k\right)\right) \tag{5}$$

$$y_i^d\left(k+1\right)=y_i^d\left(k\right)+\alpha v_i^d\left(k+1\right) \tag{6}$$

其中，$i=1,2,\ldots,N$，$d=1,2,\ldots,D$，ω为惯性因子，c_1,c_2表示学习因子，$r_1,r_2\in[0,1]$是随机数，α为约束因子，用来控制速度的权重。

3 物联网应用动态安全卸载策略

考虑一个物联网Γ，网络节点为$\Gamma=\{1,2,\ldots,n\}$。此网络中的每个物联网设备可能包含计算密集型或延迟敏感型计算任务。这些物联网设备部署在一个网络中，该网络通过智能网关分别连接到雾节点和云，从而创建一个分层网络。雾节点形成了云的网络连续体。给定一个任务，物联网评估该任务，看它是否可以使用常驻资源在本地执行该任务。如果物联网发现它无法执行任务，它会将任务卸载到雾中。雾要么执行任务，要么将其发送到云。本文的目的是执行动态卸载，同时在任务卸载期间将用户敏感的任务保持在雾中，同时在吞吐量、延迟、能耗、资源利用率和响应时间方面获得高性能。

3.1 系统建模

当前使用的雾-云-物联网系统框架是由物联网层的物联网移动设备、网络层的网络设备、雾层的雾设备和云层的云基础设施组成，如图2所示。最底层的物联网设备负责采集、监控和测量数据，然后向雾层发送、接收数据。物联网设备的特点是计算能力低，受电池寿命和小尺寸因素的限制，内存相当低。网络层由交换机、路由器和网关等网络设备组成，它们可以采用小范围雾的功能。在此框架中，通过对物联网设备数据的评估，采用智能网关对网络进行安全保护。雾层由雾节点组成，雾节点是一种高性能的分布式系统，可以沿连续过程向云和物联网层报告处理结果。云计算层分布在顶层，由可以存储和处理海量数据的云服务器组成，具有无限的能力来实现安全且繁重的计算，云层架构可以分为私有、公共或混合的。

3.2 基于粒子群优化算法的雾节点选择

随着雾节点当前工作负载的变化，采用PSO算法更新用于选择最佳雾节点的信息。雾节点的作用为降低IoT移动设备与雾之间的总处理延

迟。最佳雾化节点的选择由两个指标决定：可用处理能力（Available Processing Capacity，APC）和剩余节点能量（Remaining Node Energy，RNE）。每个节点都将使用这两个指标来计算其适合度。当用户设备发出请求时，将选择具有较高APC和RNE的雾节点。

对于任务i的每个粒子x_i，采用APC和RNE计算适应度值：

$$f(x_i)=\omega_1 APC(x_i)+\omega_2 RNE \tag{7}$$

其中，ω_1和ω_2分别表示APC和RNE的权重因子，$\omega_1\in[0.1,0.9]$和$\omega_2=1-\omega_1$。

考虑任务i的特征sz_i、cx_i、μ_i，其中sz_i、cx_i分别表示任务i的大小复杂度和平均等待时间。令bs和F为雾节点的当前缓冲区大小和CPU频率的缓冲区，则可以计算任务x_i的等待时间$L(x_i)$：

$$L(x_i)=\frac{sz_i\times cx_i\times\mu_i+bs(x_i)}{F(x_i)} \tag{8}$$

用户选择最佳节点后，请求将发送到雾节点。在物联网级别，采用下列提议的方案用于选择可以对其进行卸载的最佳雾节点。

3.3 用于安全评估的模糊神经模型

在物联网雾架构中，物联网设备通过网关与上层通信。网关负责物联网设备、雾、云和用户设备之间的桥接。网关提供通信链接，对IoT设备进行实时控制，并提供离线服务。网关可用于保护往返于上层的数据，通过隔离异常行为的资源来实现安全性。由于智能网关具有可观的处理能力和存储能力，本文通过在智能网关上使用模糊神经网络（Nuero-Fuzzy Network）来保护网络安全。

采用Nuero-Fuzzy模型来评估智能网关上来自物联网设备的数据，通过考虑探测值Sv和响应时间Ts两个因素进行安全评估。从这两个值可以推导出预测值Pv。如果Pv大于1，认为资源为有效读数，否则该读数无效，从而保证了资源与事务隔离。

假设Nuero-Fuzzy模型由N个设备组成$(d_1,d_2,\ldots,d_N)$，每个输入具有两个参数Sv和Ts，输出为有效和无效的值。探测值可以分为小、中和大三级，响应时间对应低、中和高。在提出的模型中，如果数据大小在100至350bit之间，则探测值适中；小于100bit，则Sv较小，大于350bit，则Sv较大。同样，Ts在100至1000ms之间时视为中等级别，低于100ms或者高于1000ms则视为低或高级别。对于每个情节，生成的Sv、Ts、Pv和输出都会被存储在知识库内，然后根据构造的的模糊规则对Neuro-Fuzzy网络的培训，以适应来自IoT设备的传入数据。构造的模糊规则如表1所示。

表1 构造的模糊规则

探测值S_v	时间T_s	预测值P_v
小	高	无效
大	低	
小	中/低	有效
中	高/中/低	
大	高/中	

由于探测值和响应时间不正确，IoT设备的数据被检测为无效。根据获得的有效或无效预测值，保留受信任设备的数据。

3.4 动态卸载任务

当雾节点无法在延迟约束内处理所有接收到的任务时，雾节点会将任务卸载到云服务器中。针对此情况，提出了基于Q-学习的动态任务卸载方案。Q-学习是一种无模型的强化学习机制，通过体验一个行为的后果来学习最佳的行为过程，而不必构建域映射。该方法通过代理在特定状态下尝试来实现，代理根据所收到的即时奖励和对其所采取的状态和行为的未来奖励的估计来评估行为的后果。通过反复尝试所有状态，获得最佳状态。

Q-学习包括状态空间、行为空间和奖励函数。每个状态s和行为a的配对（s，a）都有一个Q值。如果位于状态s中的代理选择一个行为，则使用式(10)根据获得的奖励数量更新状态-行为对的Q值。选择行为时，利用贪婪策略考虑后续状态的最高Q值。

在雾层给定的任务i，行为a_i表示从所有现有的VM中选择满足卸载任务i的虚拟机VM_i。任务要求包括服务器的类型(私有或公共)、可用于

在任务限制内执行任务的VM。行为空间表示为$a=\{a_1,a_2,\ldots,a_i\}$，云服务器中可用VM定义状态空间$S_m=\{VM_1,VM_2,\ldots,VM_i\}$，每个VM可以用CPU和内存量MEM进行表征，行为对表示为：

$$s=\begin{bmatrix} VM_1,a_1 & \ldots & VM_1,a_i \\ \ldots & \ldots & \ldots \\ VM_m,a_1 & \ldots & VM_m,a_i \end{bmatrix}\in(S,a) \tag{9}$$

将任务分配给满足延迟和资源限制的任何虚拟机。为了确定对服务器和任务需求的当前观测的最佳操作，雾节点根据当前状态和从环境接收到的回报选择适当的云。该系统的目标是最大化接收到的回报和最小化等待时间。将任务调度问题中的动态任务卸载看作马尔可夫决策过程，行为空间由用于每个任务i的二进制向量来描述。当可用VM接收到当前任务i时用1表示，否则用0表示。然后计算状态-行为对的奖励函数，获得的奖励表示云服务器的当前状态（运行、等待、忙碌等）。状态-行为对规则如(10)所示：

$$Q(s,a)\leftarrow Q(s,a)+\alpha\left[r+\gamma\max_{\hat{a}}Q(\hat{s},\hat{a})-Q(s,a)\right] \tag{10}$$

其中，$\alpha\in(0,1)$是学习率，r是在s状态下采取行为a所获得的奖励，$\gamma\in[0.1,0.9]$是折扣因子。使用任务分类器将输入任务分为敏感S_i或非敏感NS_i任务，敏感任务卸载到私有云服务器，非敏感任务卸载到公共云。

4 实验结果与分析

为了验证提出的基于机器学习的物联网应用动态安全卸载策略的有效性，采用吞吐量、延迟、能耗、资源利用率和响应时间等几种指标评估其性能，并在相同的环境下与FCFS[12]、DTO-SO[13]、CMS-ACO[14]和LOTEC[15]几种卸载方案进行对比。

4.1 实验环境和评估指标

本文的实验环境是在Pentium (R) Dual-Core CPU E570 0@3.0 0 GHz和RAM 2 GB计算机上进行测试，实验环境中在开源网络模拟器NS3.26上进行Java编程。创建一个雾-云物联网网络，该网络由1个智能网关，5~10个物联网移动设备，5个雾节点和1个混合云服务器组成。在实验中，所有模拟参数均已设置为遵循均匀分布。每个设备均由CPU供电，其时钟频率范围为1~1.5 GHz，时钟频率是随机设置的。将移动设备之间的可用带宽设置为100 ~1000Kb/s之间。计算卸载要求CPU周期和任务以位的形式卸载，计算任务分为复杂和非复杂任务。为了表征卸载任务的复杂性，使用负载输入数据比率（Load-input Data Ratio，LDR）。当LDR较高时，该任务被分类为计算密集型任务，否则为非密集型。非计算密集型任务可以在本地设备或边缘执行。在测试过程中，采用吞吐量、延迟、能耗、资源利用率和响应时间等几种指标用于比较分析。

吞吐量R_T表示为单位时间T卸载的任务数TO：

$$R_T=TO/T \tag{11}$$

延迟时间表示提交申请任务并获得其结果的持续时间。计算方法为：

$$T_d=T_{pro}+T_q+T_t+T_p \tag{12}$$

其中，T_{pro}、T_q、T_t、T_p分别表示处理延迟、排队延迟、传输延迟和传播延迟。

能耗E_c是指IoT移动设备执行任务消耗的能量：

$$E_c=E_p+E_t \tag{13}$$

其中，E_p、E_t分别表示处理任务卸载过程中消耗的能量和在传输和接收任务结果期间消耗。

资源利用率RU表示为移动设备在24小时内利用资源的时间百分比：

$$RU=\left(\frac{N(i)}{24}\right)\times 100 \tag{14}$$

响应时间T_R是指用户请求与操作接收之间的时间间隔。

4.2 结果分析

图1给出了请求数量对所提出方法的吞吐量影响，并与DTO-SO方法对比。将敏感任务的延迟要求设置为1s，将非敏感任务的延迟要求设置为1.5s，设置任务数$n\in\{10,20,30,40,50\}$。从图中可以看到，吞吐量随着物联网设备请求数量的增加而增加。当n=10时，本文方法实现的吞吐量敏感任务为30KB/s，非敏感任务为23KB/s。对比方法在

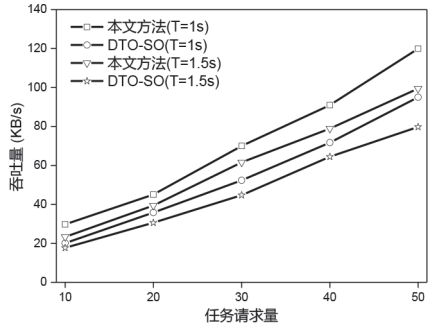

图1 不同方法在吞吐量的测试结果

敏感和非敏感任务分别为20 KB/s 和18 KB/s；在n=50时，本文方法在敏感任务和非敏感任务的吞吐量为120 KB /s和100 KB /s，而DTO-SO的吞吐量分别为95 KB /s和80 KB /s。与DTO-SO相比，提出的安全卸载方案提高了23.2%的吞吐量。

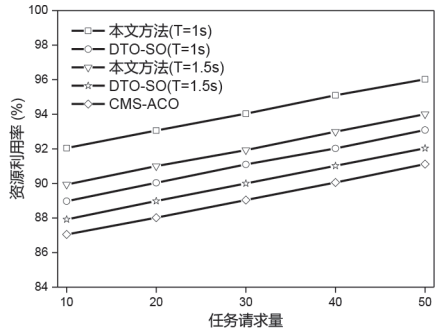

图2 不同方法在资源利用率的测试结果

图2给出了资源利用率的测试结果对比。从图中可以看出，所提出的卸载方案在n=10时，敏感任务和非敏感任务的资源利用率为92%和90%，DTO-SO和CMS-ACO则为89%和87%。

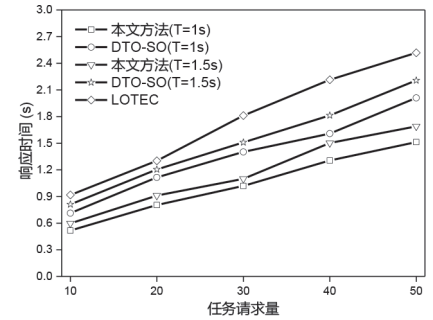

图3 不同方法在响应时间的测试结果

图3给出了不同卸载方案的响应时间的结果对比。从图中可以看出，随着请求任务数量的增加，响应时间逐渐增多。在n=10时，与LOTEC中的0.9s和DTO-SO中的0.7s相比，本文方法只需0.5s来完成卸载任务。

5 结束语

本文提出了一种基于机器学习的物联网动态应用安全卸载策略，解决了当前物联网任务卸载算法在延迟、能耗和安全方面的问题。该方案利用Neuro-Fuzzy模型消除无效资源，保护敏感数据，采用PSO选择最佳卸载雾节点，并通过调度程序动态平衡雾节点和云资源的负载以及数据安全。实验结果表明，所提出的方法用于分层系统架构中的任务卸载是安全有效的，性能明显优于其他卸载算法。

基金项目：

2018年度江苏高校哲学社会科学研究基金项目“基于虚拟现实技术的教学方法研究”（项目编号：2018SJA1239）。

参考文献

[1] Li C,Xue Y,Wang J,et al.Edge-oriented computing paradigms: A survey on architecture design and system management[J]. ACM Computing Surveys (CSUR), 2018, 51(2): 1-34.

[2] Roman R,Lopez J,Mambo M.Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges[J].Future Generation Computer Systems,2018, 78: 680-698.

[3] Wang D, Ding W, Ma X, et al.MiFo: A novel edge network integration framework for fog computing[J]. Peer-to-peer Networking and Applications, 2019, 12(1): 269-279.

[4] Naeem R Z,Bashir S, Amjad M F,et al.Fog computing in internet of things: Practical applications and future directions[J]. Peer-to-Peer Networking and Applications,2019,12: 1236－1262.

作者简介：

居晓琴（1979-），女，汉族，江苏南通人，扬州大学，硕士，江苏航运职业技术学院智能制造与信息学院，副教授；主要研究方向和关注领域：计算机应用。