面向车联网隐私保护的深度确定性策略梯度缓存方法

申自浩[1],高永生[1],王 辉[2],刘沛骞[2],刘 琨[2]

(1.河南理工大学 计算机科学与技术学院,河南 焦作 454000;2.河南理工大学 软件学院,河南 焦作 454000)

摘 要:针对车联网隐私保护边缘节点缓存命中率低的问题,本文提出一种深度确定性策略梯度缓存(DDPGC)方法。首先,经可信机构认证的出租车作为二级缓存边缘节点获取热点数据并存储至本地缓存,然后将信息广播给周边服务请求车辆(SRV);SRV将收到的广播数据缓存至本地,发生服务请求时,按照本地缓存、出租车、云服务器的顺序依次查找。其次,在出租车和SRV部署神经网络,通过深度强化学习对其缓存数据决策替换,最大化缓存收益。最后,当SRV位于车辆稀疏处,无法从周边车辆获取请求数据时,结合k匿名与随机响应扰动机制产生匿名集,以匿名方式向云服务器发送请求,在保护用户位置隐私的前提下获取服务。仿真实验结果表明,DDPGC能够有效提高车辆缓存命中率,减少SRV与云服务器交互频次,有效保护车联网用户隐私安全。

关键词:计算机应用;车联网;隐私保护;深度强化学习;缓存替换

中图分类号:TP393 **文献标志码**:A **文章编号**:1671-5497(2025)05-1638-10

DOI:10.13229/j.cnki.jdxbgxb.20230908

Deep deterministic policy gradient caching method for privacy protection in Internet of Vehicles

SHEN Zi-hao[1],GAO Yong-sheng[1],WANG Hui[2],LIU Pei-qian[2],LIU Kun[2]

(1. *School of Computer Science and Technology*, *Henan Polytechnic University*, *Jiaozuo* 454000, *China*; 2. *School of Software*, *Henan Polytechnic University*, *Jiaozuo* 454000, *China*)

Abstract: To address the problem of low cache hit ratio in edge nodes for privacy-preserving in the Internet of Vehicles (IoV), a deep deterministic policy gradient caching (DDPGC) method was proposed. Firstly, a taxi certified by a trusted authority acted as a second-level caching edge node to acquire hotspot data and store it in the local cache. It then broadcasted this information to the neighboring service requesting vehicles (SRV). SRVs cached the broadcasted data locally and search for service requests in the order of priority of local cache, taxi, and cloud server when such requests arise. Secondly, a neural network was deployed in taxis and SRV to maximize the caching benefit through deep reinforcement learning for decision

收稿日期:2023-08-27.

基金项目:国家自然科学基金项目(61300216);河南省高等学校重点科研项目(23A520033);河南理工大学博士基金项目(B2022-16).

作者简介:申自浩(1980-),男,副教授,博士.研究方向:网络信息安全与智能信息处理.E-mail:szh@hpu.edu.cn

通信作者:王辉(1975-),男,教授,博士.研究方向:网络信息安全与智能信息处理.E-mail:wanghui_jsj@foxmail.com

replacement of their cached data. Finally, when SRV were located in vehicle sparsity and could not obtain request data from neighboring vehicles, a combination of k-anonymity and random response perturbation mechanisms generated anonymity sets to send requests to cloud servers in an anonymous manner to obtain services while protecting user location privacy. Simulation experimental results show that DDPGC can effectively improve the vehicle cache hit ratio, reduce the frequency of SRV interaction with the cloud server, and effectively protect user privacy security.

Key words: computer application; Internet of Vehicles; privacy protection; deep reinforcement learning; cache replacement

0　引　言

5G 技术的普及,为车联网的发展提供了动力。车辆在向云服务器请求数据时,需要提交关于车辆的相关隐私信息,但云服务器作为不可信实体,可能会导致请求车辆的隐私泄露[1]。因此,隐私问题逐渐成为车联网安全需要考虑的关键因素之一。

近年来,一些研究方案使用路边单元(Road-side unit,RSU)作为中间节点为车辆提供服务,但限于RSU的部署规模因地区、道路类型和交通流量等而异,覆盖城市的各个角落尚有难度。出租车作为城市的基础交通工具,具有位置遍布广泛、路线多样化等特点,可以更大程度地满足车联网用户位于不同地点的请求服务,且相比于大面积部署RSU具有更大的成本优势[2]。同时,通过缓存可减少车辆向云服务器请求的频次,提升车辆隐私保护效果。

目前,机器学习技术在车辆流量预测[3]和流量优化[4]方面获得良好的应用效果。深度强化学习(Deep reinforcement learning, DRL)[5]技术以其强大认知和决策能力被广泛应用于车联网隐私保护缓存领域。Dai 等[6]提出了基于许可区块链的深度强化学习缓存方法,使用基站维护许可区块链,但忽略了基站的缓存资源。宁兆龙等[7]采用强化学习结合长短期记忆网络的方法,提出一种协同缓存框架,但未考虑用户请求过程中的隐私泄露问题。以上方法均未能有效利用用户端的缓存资源,且在用户请求过程中,可能会遭受恶意攻击导致隐私泄露。

文献[8]提出了最近最少使用(Least recently used,LRU)缓存策略。当缓存存储满时,最近请求最少的缓存数据将被新数据替换。文献[9]提出了最不常使用(Least-frequently used,LFU)缓存策略,缓存满时,以请求次数为依据对缓存数据进行替换。文献[10]提出了先进先出(First in first out,FIFO)缓存策略,缓存满时,以缓存的先后时间为依据对缓存数据进行替换。Hu 等[11]提出 LPP-CACHE,其能够依次缓存流行度较高的请求数据,但并未考虑到车辆的实际偏好。以上方法虽然都能够对缓存数据进行替换,但均未考虑到车辆动态偏好的特征。

针对以上问题,为有效利用边缘车辆的缓存资源,本文提出了深度确定性策略梯度缓存(Deep deterministic policy gradient cache, DDPGC)方法。将出租车作为二级缓存节点缓存流行文件,行驶过程中向通信范围内的服务请求车辆(Service request vehicle,SRV)进行广播。在出租车和 SRV 上部署 DRL 神经网络,学习车辆的兴趣偏好,对车辆缓存文件决策替换,提升车辆缓存命中率,减少 SRV 与云服务器的直接交互。结合区块链与加密技术保障出租车与 SRV 的交互隐私安全,当 SRV 位于车辆稀疏处,且自身缓存未包含请求数据,使用 k 匿名结合随机响应扰动机制对自身位置进行扰动,在降低车辆位置隐私泄露的前提下获取服务。DDPGC 方法可以有效提高车辆的缓存命中率,减少 SRV 与云服务器交互的频次,实现在保护请求车辆位置隐私的前提下,获得有效的请求服务。

1　系统模型

本文提出的车联网系统模型如图 1 所示,主要包含 5 个实体:可信机构(Trusted authority, TA)、云服务器(Cloud server, CS)、出租车、SRV 和区块链。

(1)可信机构。一般地,TA 均由国家或交管部门维护,其具有强大的计算和存储能力,同时也可抵御外部攻击。

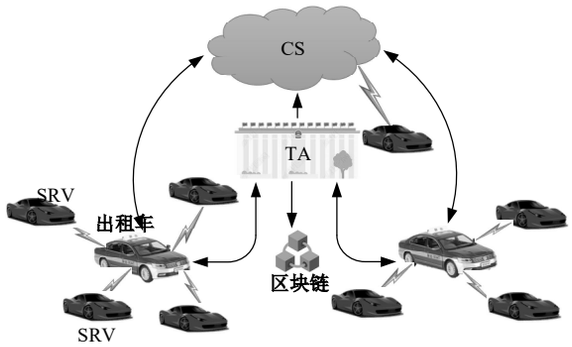

图1 DDPGC系统模型

Fig. 1 System model of DDPGC

(2)云服务器。CS拥有车辆的服务请求数据文件。当出租车缓存未命中SRV请求数据,则转发请求至CS,CS通过出租车响应SRV的请求结果。

(3)出租车。出租车具有基数大、位置遍布广泛等特性,且其路线及位置等均不代表出租车个体的兴趣点。为出租车配备具有AI计算能力的芯片组,扩充车辆的计算和存储资源,将其作为车联网系统二级缓存边缘节点,为其他车联网用户提供安全可靠的信息服务。

(4)服务请求车辆。SRV既可通过出租车及外界广播数据的缓存满足自身需求,又可使用车载单元与外界进行通信获取请求数据。

(5)区块链。区块链是一种分布式账本技术,其具有防篡改、去中心化、可追溯等特点。在本系统中,其主要工作为存储车辆交互过程中的数据,在车辆遭受恶意攻击时,能够实现对恶意车辆的追溯。

为了充分利用边缘节点的缓存资源,将出租车作为二级缓存节点,SRV在本地缓存未命中时,优先向通信范围内的出租车发送服务请求,出租车在本地缓存文件中进行查询;若命中,则直接返回请求数据,否则,转发请求至云服务器,云服务器通过出租车返回响应数据。

将包含SRV偏好的数据文件缓存至出租车可以有效提升缓存的命中率,主要依赖于出租车对覆盖范围内车辆偏好的了解程度。因此,为提高出租车和SRV的缓存命中率,分别在其上部署DRL网络模型,为车辆提供服务的同时,学习覆盖范围内车辆的请求偏好,对出租车和SRV的当前缓存文件进行决策更新,保证车辆的长期缓存命中率最大化。在DDPGC方法中演员网络和评论家网络共同构成了智能体。

定义1 动作选择函数

DDPGC的演员网络通过对输入的状态采用确定性策略的方式选取动作,该函数定义为:

$$A_{\mathrm{T}}=\pi(S_{\mathrm{T}}|\theta) \tag{1}$$

为增加学习过程的探索性,对演员网络选择的动作添加一定的噪声N,将动作选择函数转换为:

$$A_{\mathrm{T}}=\pi\left(S_{\mathrm{T}}|\theta\right)+\boldsymbol{N} \tag{2}$$

式中:$\pi(\cdot|\theta)$为演员网络的策略函数;S_{T}为演员网络输入的状态;θ为演员在线网络参数;$\boldsymbol{N}$为符合高斯分布的噪声向量。

定义2 缓存命中率

$C_{\mathrm{T}}(t)$表示出租车在t时刻的缓存文件,$R_{\mathrm{T}}(t)$表示在t时刻出租车接收到的服务请求。那么在出租车端的缓存命中率$H_{\mathrm{T}}(t)$为$C_{\mathrm{T}}(t)$和$R_{\mathrm{T}}(t)$的交集数量与请求$R_{\mathrm{T}}(t)$数量之比。即

$$H_{\mathrm{T}}(t)=\frac{\left|C_{\mathrm{T}}(t)\cap R_{\mathrm{T}}(t)\right|}{\left|R_{\mathrm{T}}(t)\right|} \tag{3}$$

定义3 累积奖励

由于智能体在与环境交互过程中,每个状态均产生一个即时奖励。然而,DDPGC旨在提高出租车长期的缓存命中率,因此将模型的奖励与训练中的命中率进行关联。此外,为使智能体能够考虑到这些行为决策的长期影响,设置累积奖励函数。

考虑到未来奖励存在不确定性,设置未来奖励的权重应小于当前奖励的权重。累积奖励函数$r(t)$可表示为:

$$r(t)=\sum_{i=0}^{M}\gamma^{i}H(t+i) \tag{4}$$

式中:$H(t+i)$为从t时刻开始第i个时刻的命中率;γ为折扣因子,且$\gamma\in[0,1]$。

定义4 目标Q值函数

目标Q值函数是一种估计当前策略下的累积奖励函数,目的是最小化当前策略下预测Q值和实际Q值之间的均方误差。目标Q值函数y_{T}可表示为:

$$y_{\mathrm{T}}=H_{\mathrm{T}}+\gamma\times Q\left(S_{\mathrm{T}}(t+1),\pi\left(S_{\mathrm{T}}(t+1)\right),\omega'\right) \tag{5}$$

式中:H_{T}为命中率;γ为折扣因子,$\gamma\in[0,1]$;$Q\left(S_{\mathrm{T}}(t+1),\pi\left(S_{\mathrm{T}}(t+1)\right),\omega'\right)$为在状态$S_{\mathrm{T}}(t+1)$下,采取动作的估计$Q$值,$\omega'$为评论家目标网络

参数。

定义 5　损失函数

通过最小化实际 Q 值与预测 Q 值之间的均方误差实现目标 Q 值函数的训练。评论家网络的训练损失函数 $L(\omega)$ 定义为:

$$L(\omega)=\frac{1}{n}\sum_{i=1}^{n}\left(y_{\mathrm{T}}^{i}-Q(S_{\mathrm{T}}(t),A_{\mathrm{T}}(t),\omega)\right)^{2} \quad (6)$$

式中:y_{T}^{i} 为第 i 个样本经过目标 Q 网络输出的目标 Q 值;$Q(S_{\mathrm{T}}(t),A_{\mathrm{T}}(t),\omega)$ 为在状态 $S_{\mathrm{T}}(t)$ 下,采取动作 $A_{\mathrm{T}}(t)$ 的 Q 值;ω 为评论家在线网络参数。

采用 n 次蒙特卡洛采样估计目标函数梯度,更新演员网络参数 θ:

$$\nabla_{\theta}J(\pi)\approx\frac{1}{n}\left(\nabla_{A_{\mathrm{T}}}Q_{\theta}\left(S_{\mathrm{T}}(i),A_{\mathrm{T}}(i)\right)\cdot\nabla_{\theta}\pi(S_{\mathrm{T}}(i)|\theta)\right) \quad (7)$$

式中:$\nabla_{A_{\mathrm{T}}}Q_{\theta}\left(S_{\mathrm{T}}(i),A_{\mathrm{T}}(i)\right)$ 为演员网络动作值函数的梯度;$\nabla_{\theta}\pi(S_{\mathrm{T}}(i)|\theta)$ 为演员网络在第 i 个状态下的策略梯度。

2　DDPGC 方法设计

2.1　车辆认证及缓存初始化

TA 选择两个大质数 p、q 及基点 P。随机产生一个大数作为自身的私钥 $sk_{\mathrm{TA}}\in Z_q^*$,并计算自身的公钥 $PK_{\mathrm{TA}}=sk_{\mathrm{TA}}\cdot P$。TA 将参数 $\{p,q,PK_{\mathrm{TA}},P\}$ 发送给系统中的所有车辆。

出租车和 SRV 分别向 TA 发送自身的识别码以及相关信息,TA 验证通过之后,选择随机数 $sk_{\mathrm{taxi}}\in Z_q^*$ 作为出租车的私钥,$sk_{\mathrm{SR}}\in Z_q^*$ 作为 SRV 的私钥,并计算出租车公钥 $PK_{\mathrm{taxi}}=sk_{\mathrm{taxi}}\cdot P$,SRV 公钥 $PK_{\mathrm{SR}}=sk_{\mathrm{SR}}\cdot P$。通过安全通道,TA 将 $\{PK_{\mathrm{taxi}},sk_{\mathrm{taxi}}\}$ 和 $\{PK_{\mathrm{SR}},sk_{\mathrm{SR}}\}$ 分别发送至出租车和 SRV。为保证系统的安全性,TA 备份车辆的公钥和私钥。

出租车作为系统中的服务提供车辆,其安全性应放在首位。在出租车出发前向 TA 进行认证,出租车的认证及缓存初始化主要步骤如下:

步骤 1　出租车选择一个随机数 $r\in Z_q^*$,计算 $V_1=r\cdot P$,$V_2=(r+sk_{\mathrm{taxi}})\cdot PK_{\mathrm{TA}}$,获取自身位置 L_{taxi} 及 ID,将 $\{V_1,V_2,L_{\mathrm{taxi}},\mathrm{ID}\}$ 使用 PK_{TA} 加密转发至 TA。

步骤 2　TA 收到认证信息后,使用自身私钥验证 $V_2=sk_{\mathrm{TA}}\cdot V_1+sk_{\mathrm{TA}}\cdot PK_{\mathrm{taxi}}$,若验证成功,转发 L_{taxi} 和 PK_{taxi} 至云服务器,云服务器根据每个出租车当前所在的不同位置,使用 PK_{taxi} 加密返回热点信息 HM;TA 使用 sk_{TA} 对出租车的身份识别码 ID 和验证结果 VAL 生成签名 ECDSA(ID, VAL),并将 $\{\mathrm{HM}\,|\,\Phi\}$ 通过 PK_{taxi} 加密返回至出租车,其中 $\Phi=(\mathrm{ID},\mathrm{VAL},\mathrm{ECDSA}(\mathrm{ID},\mathrm{VAL}))$,ECDSA 为椭圆曲线数字签名生成算法。

步骤 3　出租车对返回的消息解密之后,选取 HM 中前 n 个流行文件进行缓存,并将 Φ 保存至出租车,以便 SRV 进行安全认证。

为保证 SRV 与出租车之间的请求交互安全,SRV 需事先与出租车进行安全认证。SRV 认证与缓存初始化的主要步骤如下:

步骤 1　当 SRV 进入出租车覆盖范围内,首先向出租车获取公钥 PK_{taxi} 和 Φ,通过 PK_{TA} 验证签名的合法性。

步骤 2　若验证成功,SRV 向出租车发起请求加入信息,并使用 PK_{taxi} 对请求加密生成密文 VM。最后,将 $\{PK_{\mathrm{SR}}\,|\,\mathrm{VM}\}$ 发送至出租车。

步骤 3　出租车对 VM 解密,确认 SRV 的加入,返回当前出租车自身缓存的热点信息,并保存 PK_{SR},方便之后的通信。

当 SRV 在发起服务请求时,首先在本地进行查询,降低 SRV 对云服务器直接请求所面临的隐私泄露风险。

2.2　基于区块链的车辆交互信息管理

为保证车辆交互过程中的权益,在原有加密基础上,引入区块链作为第二层防护机制,为可能发生的车辆纠纷提供仲裁依据。

为保证车辆的安全性,同时在系统遭受恶意攻击的同时,能够实现追溯,SRV 在自身缓存未命中时,可分别使用 PK_{taxi} 和 PK_{TA} 对请求数据加密,并转发至出租车。若出租车在自身缓存中命中请求,则将响应数据发送给 SRV。否则,为获取请求数据,出租车转发未命中请求至云服务器。出租车是半可信的,因此为保证车辆安全,对出租车转发和接收到的数据通过 PK_{TA} 加密,保存至区块链。

将车辆的一个完整请求作为一个区块。每个区块包含区块头和区块体,其中区块头中主要包含请求完成的时间戳、出租车身份识别码、邻接区块的哈希地址和默克尔根;区块体中存储使用 PK_{TA} 加密的请求和响应信息的哈希值。

当系统中出现车辆数据被篡改或车辆遭受攻击时,通过区块链与TA结合,实现对恶意车辆的追溯。

2.3 车辆请求及决策更新

假设出租车B_i范围内具有j辆SRV,且均使用车载单元进行短程通信。当认证后的SRV位于出租车覆盖区域内时,若SRV_j在t时刻请求的信息未命中,则SRV_j可以连接到出租车并向其发送不带自身位置的服务请求。如果请求的数据在当前连接的出租车缓存中,出租车将包含该数据的文件返回给SRV_j;否则,转发未命中请求至云服务器。

每个时隙接收到SRV的未命中请求$R_T(t)$,出租车将检查自身缓存文件$C_T(t)$。当出租车在自身缓存中未命中时,可直接向云服务器发出请求$B_i^C=\{L_i|q_1,q_2,q_3,\cdots,q_j\}$,从而在降低用户位置隐私泄露的情况下获得请求内容。其中,B_i^C表示第i辆出租车向云服务器发起的请求;L_i表示第i辆出租车的位置;q_j表示R_i覆盖范围内的第j辆SRV在出租车未命中的请求,即覆盖范围内的j辆SRV在出租车B_i本地未命中时,均使用B_i的位置作为请求位置获取数据。

为了出租车和SRV在一段时间内缓存命中率最大化,将该问题转化为马尔可夫决策过程,该过程主要包含4个部分,即状态空间、动作空间、状态转移概率、奖励。出租车端主要表示为(S_T,A_T,P_T,r_T,γ),SRV端主要表示为(S_R,A_R,P_R,r_R,γ),其中折扣因子$\gamma\in[0,1]$。

(1)状态空间

若直接将请求文件$R_T(t)$作为状态的一部分输入,可能会导致智能体学习到的请求偏好效果不佳。因此,计算$R_T(t)$的请求偏好$\rho_G(t)$作为状态的一部分,以增强模型的决策能力。

云服务器端存储所有文件数据$F=\{f_0,f_1,\cdots,f_n\}$。出租车连接SRV个数为λ,且每个SRV在同一个时间内仅发出一个请求,则出租车每次接收到的所有请求文件的流行度集合为$\rho_G(t)=\{\rho_f^v(t)\}$,其中$v=(1,2,\cdots,\lambda)$,$\rho_f^v(t)=\dfrac{\sum_{k=1}^{\lambda}f^k}{\lambda}$,$f\in F$,$f^k=\begin{cases}1,\text{当第}k\text{个SRV在}t\text{时刻请求文件}f\\0,\text{其他}\end{cases}$。

由此出租车的状态空间可定义为:

$$S_T(t)=\{C_T(t),R_T(t),\rho_G(t)\} \tag{8}$$

同理,定义$\rho_H(t)=\{\rho_f^r(t)\}$,其中$r=(1,2,\cdots,\mu)$,$\rho_f^r(t)=\dfrac{\sum_{r=1}^{\mu}f^i}{\mu}$,$f\in F$,$f^i=\begin{cases}1,\text{当SRV的第}i\text{个历史请求为}f\\0,\text{其他}\end{cases}$。

$S_R(t)=\{C_R(t),R_H(t),\rho_H(t)\}$,$C_R(t)$表示SRV在$t$时刻的缓存文件,$R_H(t)$表示从$t$时刻起SRV的$n$次请求历史,$\rho_H(t)$表示在$t$时刻获取的请求历史偏好。

(2)动作空间

设出租车端智能体的动作空间为$A_T(t)=(a_T^I(t),a_T^o(t))$,当动作向量$a_T^I(t)=1$,在$t$时刻出租车请求云服务器获取的某个文件应被加入$C_T(t)$,当$a_T^I(t)=0$,则应丢弃。同理,当$a_T^o(t)=1$,$C_T(t)$中某个文件应该被删除,否则应被继续保留。因此,$a_T^I(t),a_T^o(t)\in\{0,1\}$。

同理,在SRV端也具有相似的缓存机制。设SRV端智能体的动作空间为$A_R(t)=(a_R^I(t),a_R^o(t))$,$a_R^I(t)$表示$SRV_i$获取的文件是否应该被缓存,当$a_R^I(t)=1$,当前文件被缓存,否则被丢弃;$a_R^o(t)$表示SRV当前缓存$C_R(t)$中某个文件是否应该被替换删除,当$a_R^o(t)=1$,当前文件被删除,否则当前文件保留。

(3)状态转移概率

$P_T(S_T(t),S_T(t+1))=\{S_T(t+1)|S_T(t),A_T(t)\}$,表示出租车在当前行为$A_T(t)$下,从当前状态$S_T(t)$转移到下一个状态$S_T(t+1)$的概率。

$P_R(S_R(t),S_R(t+1))=\{S_R(t+1)|S_R(t),A_R(t)\}$,表示$SRV_i$在当前行为$A_R(t)$下,从当前状态$S_R(t)$转移到下一个状态$S_R(t+1)$的概率。

(4)奖励函数

当在状态$S_T(t)$下执行动作$A_T(t)$,当前状态转变为$S_T(t+1)$,且返回当前的实时命中率,环境根据命中率进行奖励。

为了让智能体学习到最优策略,保证出租车和SRV缓存命中率最大化,且智能体不陷入局部最优解,设计出租车的累积奖励函数为:

$$r_T(t)=\sum_{i=0}^{M}\gamma^i H_T(t+i) \tag{9}$$

式中:$H_T(t+i)$为i时刻的命中率。由于未来的

奖励存在不确定性,设置$\gamma\in[0,1]$,随着i的增大,γ^i的值随之减小,这样可减小未来不确定性在短期内的影响,确保智能体学习到最优策略。

同理,设置SRV的累积奖励函数为:

$$r_{\mathrm{R}}(t)=\sum_{i=0}^{M}\gamma^i H_{\mathrm{R}}(t+i) \tag{10}$$

式中:$H_{\mathrm{R}}(t+i)$为i时刻的平均命中率。

经过上述分析,可以得到出租车端的缓存替换算法流程,如算法1所示。

算法1　出租车缓存替换算法

输入:演员和评论家在线网络参数θ、ω、演员和评论家目标网络参数$\theta'\leftarrow\theta$,$\omega'\leftarrow\omega$,更新系数τ,重放缓存RM;
输出:出租车最优缓存替换策略$\pi(\cdot|\theta)$;
(1) for k ←1,…,S
(2) 初始化缓存状态$C_{\mathrm{T}}(0)$;
(3) for t ←1,…,M
(4) 出租车接收SRV的消息VM;
(5) 使用$\mathrm{PK}_{\mathrm{taxi}}$对VM进行解密,获得请求$\mathrm{R_T(t)}$,计算车辆偏好$\rho_{\mathrm{G}}(\mathrm{t})$;
(6) 获取状态$\mathrm{S_T(t)}$;
(7) 计算动作$\mathrm{A_T(t)}$;
(8) 反馈实时命中率$\mathrm{H_T(t)}$;
(9) 更新出租车的缓存$\mathrm{C_T(t)}$为$\mathrm{C_T(t+1)}$;
(10) 将$\{\mathrm{S_T(t)},\mathrm{A_T(t)},\mathrm{S_T(t+1)},\mathrm{H_T(t)}\}$存储至重放缓存RM中;
(11) 从重放缓存中随机选择n个样本数据,计算目标Q值$\mathrm{y_T}$;
(12) 通过$\mathrm{L}(\omega)$和梯度反向传播更新ω;
(13) 基于梯度$\nabla_\theta \mathrm{J}(\pi)$更新$\theta$;
(14) end for
(15) 更新目标网络参数θ'和ω',
$\omega'=\tau\omega+(1-\tau)\omega'$,
$\theta'=\tau\theta+(1-\tau)\theta'$;
(16) end for

训练过程主要集中在出租车端,在出租车训练后,向SRV_i下发演员网络参数θ,返回SRV_i的请求文件,因此SRV_i端仅需部署演员网络即可。在计算$A_{\mathrm{R}}(t)$后,SRV_i根据出租车或云服务器的响应文件,将当前状态$C_{\mathrm{R}}(t)$更新为$C_{\mathrm{R}}(t+1)$。SRV_i端的缓存替换算法流程如算法2所示。

2.4　车辆稀疏处的直接请求服务

考虑到极端情况下,当SRV所在区域为车辆稀疏处,附近没有边缘出租车,且自身缓存未命中请求数据,此时,可通过算法3生成k匿名集,直接向云服务器发起服务请求,从而在降低用户位置隐私泄露概率的情况下获得请求数据。

算法2　SRV_i缓存替换算法

输入:出租车返回的请求文件和演员网络参数θ;
输出:SRV_i最优缓存替换策略$\pi(\cdot|\theta)$;
(1) 初始化缓存状态$\mathrm{C_R(0)}$,初始化请求$\mathrm{R_R(0)}$;
(2) for t ←1,…,M
(3) 获取$\mathrm{SRV_i}$的σ个请求历史
$\mathrm{R_H(t)}=\sum_{\mathrm{i}=1}^{\sigma}\mathrm{R_H(t+i)}$;
(4) 通过$R_{\mathrm{H}}(t)$计算SRV_i的偏好$\rho_{\mathrm{H}}(\mathrm{t})$;
(5) 获取状态$\mathrm{S_R(t)}$;
(6) 计算动作$\mathrm{A_R(t)}=\pi(\mathrm{S_R(t)}|\theta)$;
(7) 反馈平均缓存命中率$\mathrm{H_R(t)}$;
(8) 更新SRV_i当前缓存文件为$\mathrm{C_R(t+1)}$;
(9) SRV_i产生新的请求$\mathrm{R_R(t+1)}$;
(10) end for

k匿名位置生成算法的具体流程如算法3所示。输入SRV的历史兴趣点概率表T,真实位置L,计算与真实位置概率相似的兴趣点P,对P排序并选择前N个兴趣点加入候选区CZ,从CZ中选取$k-1$个位置与L组成k匿名集。为提升匿名集中位置的分散性,使用希尔伯特曲线(Hilbert curve,HC)对匿名位置划分,并存储至四叉树。通过对四叉树中的HC编码添加随机响应扰动,最终输出满足条件的k匿名集。当生成的k匿名集不包含L时,在获取服务请求之后,可对L的近邻位置点的请求结果取并集,作为请求结果。

算法3　k匿名位置生成算法

输入:SRV历史兴趣点概率表T,真实位置L;
输出:k匿名位置集Z;
(1) $\mathrm{L_p}\leftarrow\mathrm{get(T,L)}$;//根据T获取R的查询概率;
(2) $\mathrm{L'}\leftarrow\mathrm{sim(T,L_p)}$;
//获取与L查询概率相似的位置L′
(3) $\mathrm{CZ}\leftarrow\mathrm{sortDis(L',2k-2)}$;
//将L′根据欧几里得距离排序并选取候选区域
(4) $\mathrm{Z_k}\leftarrow\mathrm{selectL(CZ,k-1)+L}$;
//选取k−1个位置点与L组成匿名集
(5) for l in $\mathrm{Z_k}$
(6) QuadTreeInsert(l, root);
(7) end for
(8) $\mathrm{Z_k}\leftarrow\mathrm{TravelQuadTree(root)+Rappor}$;
//遍历四叉树,并添加随机响应扰动
(9) $\mathrm{Z}\leftarrow\mathrm{HCDecode(Z_k)}$;
(10) return Z

3　安全分析

3.1　隐私保护度分析

当SRV位于出租车覆盖范围内,且在本地缓

存未命中时,可直接向出租车请求所需数据。若请求命中,则无须转发该请求至云服务器,否则为未命中的SRV请求统一添加出租车自身位置作为请求位置,再转发至云服务器获取服务数据。

若SRV请求数据在自身缓存中,则不会对其隐私产生影响。因此,本方法对用户隐私产生影响的关键因素之一是车辆的缓存命中率。

本文主要考虑云服务器对SRV提出服务请求的推理攻击。假设出租车当前连接有n个SRV,每个SRV在一个时隙内最多发出一个请求,且出租车端命中个数为m,则为获取SRV_i的请求数据,出租车需要将剩余的$n-m$个未命中数据转发至云服务器。因此,在云服务器端SRV隐私泄露的概率可表示为云服务器精确匹配SRV请求数据的概率。

该问题可转化为排列组合问题。在n个SRV中找出$n-m$个SRV,且对选出的$n-m$个SRV进行排序,求得可能出现的排列组合个数即为云服务器精确匹配SRV请求数据的可能性。

因此,SRV请求数据隐私保护程度为:

$$\tau = 1 - \frac{1}{c_n^{n-m} \cdot A_{n-m}^{n-m}} \tag{11}$$

式中:c_n^{n-m}为从n个SRV中找出$n-m$个SRV的组合数;A_{n-m}^{n-m}为对$n-m$个SRV进行全排列。

3.2 基于区块链的车辆安全追溯

(1)区块链的存在,保证了数据的不变性,可为用户提供追溯功能。当用户接收到假数据或受到恶意服务提供者的攻击,通过出租车身份识别码及假数据的时间戳,即可找到对应区块,结合TA即可确认请求及响应内容,从而追溯恶意车辆。主要追溯流程为:首先,获取SRV请求连接的出租车的公钥,通过TA获得对应出租车的身份识别码;其次,按照SRV请求的时间序列对存储相应身份识别码的区块进行检索;最后,对相应区块的哈希值进行匹配并形成证据链,完成车辆的追溯。依据SRV请求信息的哈希值以及出租车转发存储至区块链的数据哈希值完成请求信息的确认。同理,依据SRV接收到的响应信息哈希值与出租车接收并转发存储至区块链的信息哈希值完成响应信息的确认。若哈希匹配失败,则溯源至对应车辆。

在车辆的交互过程中,每生成一个区块,均需要构造相应的默克尔树,其叶子节点为对应请求及响应数据的哈希值。在构造过程中,随着叶子节点的增加,树的高度也随之增加。此时,对于默克尔树的生成以及基于默克尔树的查找匹配,其时间复杂度趋于$O(\log_2 n)$。在DDPGC方法中,随着请求车辆数的增加,生成区块的数量也随之增加,生成区块数与请求车辆数成正比,因此区块链生成的时间复杂度趋于$O(n\log_2 n)$。

(2)由于区块链中默克尔树的存在,确保区块中交易数据的完整性,如果有恶意攻击者试图篡改其中某一条数据,那么该条数据的哈希值将会发生改变,从而导致默克尔根的改变。当一个区块的默克尔根值发生改变,将会被其他的节点检测到,因此必须同时修改后续区块的区块头,以维持区块链的一致性。然而,这种攻击是非常困难和耗费资源的。

因此,引入区块链实现恶意车辆的追踪机制,可有效增强系统的安全性,确保车辆交互过程中的权益。

4 仿真结果

在仿真实验中,基于Python和tensorflow搭建深度神经网络。为防止训练过程的剧烈波动,保证模型训练的稳定性,设置演员网络参数为0.000 1,评论家网络参数为0.000 1,使用Adam优化器对演员和评论家网络进行训练。训练的数据集基于文件流行度生成,且遵循Zipf分布,设置云服务器端可获取的总文件数为30,出租车可连接SRV个数为[5,30]。由于状态转移概率是未知的,在仿真实验中,随机生成状态转移矩阵$\boldsymbol{P}$。

由图2可知,当出租车覆盖范围内SRV个数为8时,在训练中,智能体逐渐学习到出租车覆盖区域内SRV的兴趣偏好,其平均缓存命中率逐渐稳定在0.63左右。

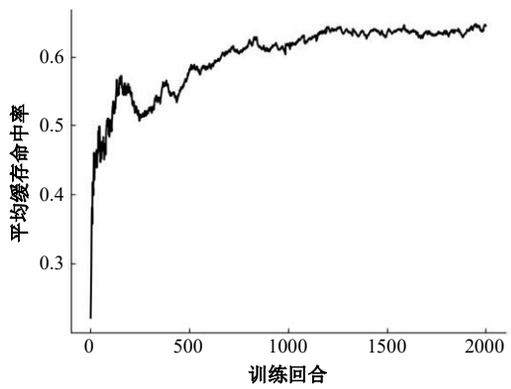

图2 出租车端平均缓存命中率

Fig. 2 Taxi side average cache hit ratio

为了验证DDPGC方法的有效性,将其与LRU[8]、LFU[9]、FIFO[10]和LPP-CACHE[11]算法进行对比。假设出租车端的文件缓存率为100%,即可以存储云服务器中所有的文件数据,此时SRV的每次请求在出租车端均可命中,即命中率为100%,但这种情况是不可能发生的。因此,设置出租车端的文件缓存率为[0.25,0.55],SRV端的文件缓存率为[0.1,0.4]。

由图3可以看出,在出租车和SRV端,随着出租车缓存利用率逐渐增大,各方法在缓存命中率均呈现上升趋势,但DDPGC方法始终优于对比方法。在出租车端,随着缓存利用率的减小,DDPGC方法的优势更为明显。这也表明在缓存容量有限的情况下,DDPGC方法相对其他方法更有优势。在SRV端,随着缓存利用率逐渐增大,DDPGC方法始终优于对比方法,因为LFU、LRU和FIFO遵循静态替换规则,未考虑SRV请求动态变化的特性。在LPP-CACHE中,RSU能够依次缓存流行内容,但并未完全考虑SRV的动态需求。同时,LPP-CACHE也浪费了SRV的缓存资源;DDPGC能够适应用户偏好不断变化的场景,可通过请求预测SRV的偏好,以优化车辆的缓存资源。因此,采用DDPGC方法进行缓存替换可以在缓存容量有限的情况下,有效提高车辆缓存的命中率。

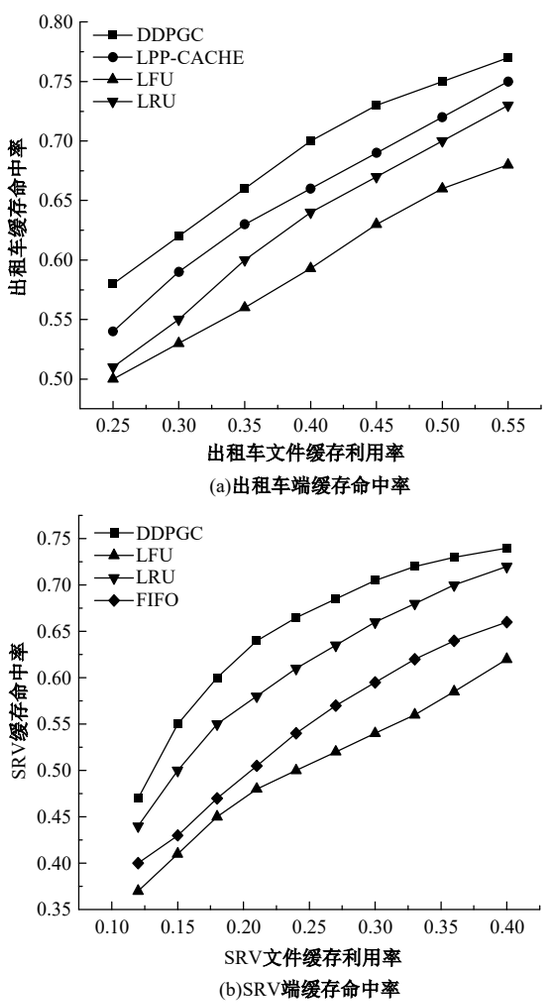

图3　不同缓存替换策略下缓存命中率

Fig. 3　Cache hit ratio with different cache replacement policies

由图4可以看到,随着出租车连接的SRV数量增多,车辆完成请求所需的时间相应增加。这是因为每个SRV在当前时刻最多发起一次请求,随着请求数量的增多,出租车端在查找、计算、替换方面所消耗的资源相对增加,从而引起请求时延增加。然而,与其他方法相比,DDPGC具有良好的缓存命中率,可以更有效地利用自身缓存资源,减少回程流量,因此在请求时延方面仍具有一定的优势。

由图5可看出,在DDPGC方法中,当缓存命中率较低时,隐私保护程度较其他方法仍具有一定的优势。对于文献[11],在RSU未命中请求数据时,转发的请求包含SRV真实位置。文献[2]推断出用户真实位置的概率为$1/k$,本文将选取的位置随机响应扰动后生成k匿名集,在扰动过程中,偶尔出现车辆的真实位置不存在于匿名集的情况。因此,云服务器仅从单次查询中推断出SRV真实位置的概率理论上小于等于$1/k$。由式(11)和出租车端平均缓存命中率可知,随着出租车连接SRV数量的增加,本方法中用户的隐私保

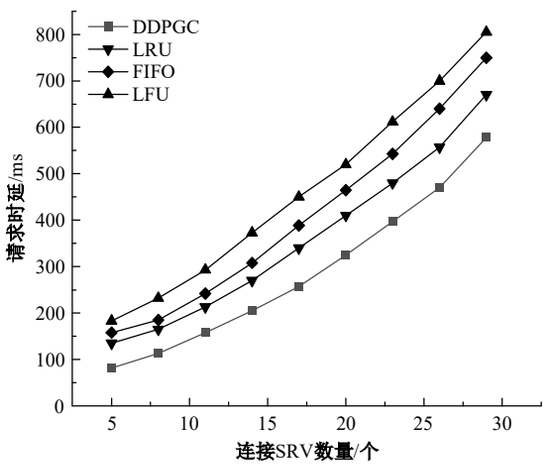

图4　连接不同SRV下的请求时延

Fig. 4　Request latency when connecting to different SRV

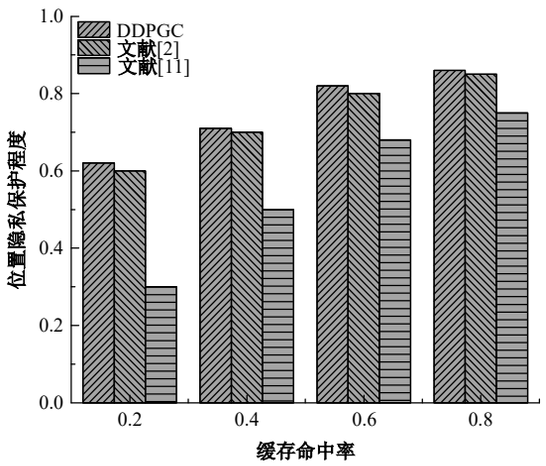

图5 缓存命中率与隐私保护程度之间的关系

Fig. 5 Relationship between cache hit ratio and level of privacy protection

护程度也逐渐趋近于1。

图6给出了出租车连接SRV数量与隐私保护度之间的关系。DDPGC-H1和DDPGC-H2分别表示出租车端平均缓存命中率为0.63和0.77时的隐私保护度。GCCG[12]为基于聚类的k-匿名策略,结合聚类的方法生成k-匿名集,每次均需向服务器发起请求,服务器端对SRV的可识别概率为$1/k$。

图6实验结果表明,与GCCG相比,DDPGC方法考虑了缓存机制,出租车端未命中的数据请求在转发给云服务器时,统一使用出租车的标识向云服务器发起请求,受命中率的影响,服务器端精确匹配多个SRV的难度随之增加。因此,DDPGC方法在减少车辆与非完全可信实体通信

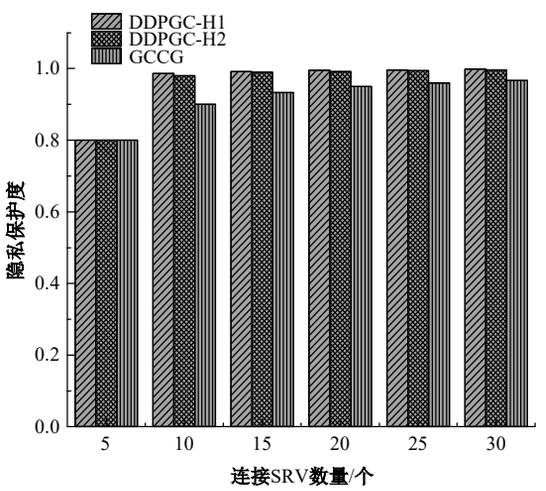

图6 连接SRV数量与隐私保护度

Fig. 6 Number of SRV connected and privacy protection degree

频率的同时,保证了车辆请求的匿名性,有效增强了用户的隐私保护度。然而,由图6中DDPGC-H1和DDPGC-H2的实验结果可以看出,随着缓存命中率的提升,DDPGC方法的隐私保护度会略有降低。这是因为,随着到达服务器端未命中请求数量的减少,服务器端精确匹配SRV的难度相对降低,但即使在该情况下,DDPGC方法仍然具有一定的隐私保护优势。

5 结束语

本文提出了一种保护隐私的深度确定性策略梯度缓存方法。该方法将出租车作为二级缓存边缘节点,为SRV提供请求服务,当SRV请求在自身缓存未命中时,可通过二级缓存节点出租车获取请求数据。在出租车和SRV端部署深度强化学习网络,提升缓存的命中率,减少SRV与云服务器直接通信的频次,同时也降低了SRV隐私泄露的可能性。当SRV位于车辆稀疏处时,利用k匿名技术向云服务器发起请求。通过隐私保护度分析可知,DDPGC方法在隐私保护方面具有一定的优势。仿真实验表明,DDPGC能够有效提高车辆缓存命中率,减少SRV与云服务器交互频次,在保护用户隐私的前提下,获得有效的请求服务。下一步将在本文方法的基础上优化算法的性能,考虑出租车之间的协同训练及协作缓存,进一步提升车辆的缓存命中率。

参考文献:

[1] Li Y, Tao X, Zhang X, et al. Break the data barriers while keeping privacy: a graph differential privacy method[J]. IEEE Internet of Things Journal, 2023, 10(5): 3840-3850.

[2] 崔杰, 陈学峰, 张静, 等. 基于公交车缓存的车联网位置隐私保护方案[J]. 通信学报, 2021, 42(7): 150-161.
Cui Jie, Chen Xue-feng, Zhang Jing, et al. Bus cache-based location privacy protection scheme in the Internet of vehicles[J]. Journal on Communications, 2021, 42(7): 150-161.

[3] Liu T, Zhang J. An adaptive traffic flow prediction model based on spatiotemporal graph neural network [J]. The Journal of Supercomputing, 2023, 79(14): 15245-15269.

[4] Zhang K, Liu Y, Zhang J, et al. TDCA: improved optimization algorithm with degree distribution and

communication traffic for the deployment of software components based on autosar architecture[J]. Soft Computing, 2023, 27(12): 7999-8012.

[5] 张健, 李青扬, 李丹, 等. 基于深度强化学习的自动驾驶车辆专用道汇入引导[J]. 吉林大学学报: 工学版, 2023, 53(9): 2508-2518.
Zhang Jian, Li Qing-yang, Li Dan, et al. Merging guidance of exclusive lanes for connected and autonomous vehicles based on deep reinforcement learning [J]. Journal of Jilin University (Engineering and Technology Edition), 2023, 53(9): 2508-2518.

[6] Dai Y, Xu D, Zhang K, et al. Deep reinforcement learning and permissioned blockchain for content caching in vehicular edge computing and networks[J]. IEEE Transactions on Vehicular Technology, 2020, 69(4): 4312-4324.

[7] 宁兆龙, 张凯源, 王小洁, 等. 基于多智能体元强化学习的车联网协同服务缓存和计算卸载[J]. 通信学报, 2021, 42(6): 118-130.
Ning Zhao-long, Zhang Kai-yuan, Wang Xiao-jie, et al. Cooperative service caching and peer offloading in Internet of vehicles based on multi-agent meta-reinforcement learning[J]. Journal on Communications, 2021, 42(6): 118-130.

[8] Sabnis A, Salem T S, Neglia G, et al. GRADES: gradient descent for similarity caching[J]. IEEE/ACM Transactions on Networking, 2023, 31(1): 30-41.

[9] Zong T, Li C, Lei Y, et al. Cocktail edge caching: ride dynamic trends of content popularity with ensemble learning[C]//IEEE INFOCOM 2021 - IEEE Conference on Computer Communications, Vancouver, Canada, 2021: 1-10.

[10] Yang J, Song Z, He P, et al. Social-aware caching strategy based on joint action deep reinforcement learning[J]. Wireless Communications and Mobile Computing, 2021, 2021: 1-15.

[11] Hu L, Qian Y, Chen M, et al. Proactive cache-based location privacy preserving for vehicle networks [J]. IEEE Wireless Communications, 2018, 25(6): 77-83.

[12] Ni S, Xie M, Qian Q. Clustering based k-anonymity algorithm for privacy preservation[J]. International Journal of Network Security, 2017, 19(6): 1062-1071.