软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2017,28(9):2354−2369 [doi: 10.13328/j.cnki.jos.005180]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 考虑信任可靠度的分布式动态信任管理模型*

游 静[1], 上官经伦[1], 徐守坤[1], 李千目[2], 王印海[3]

[1](常州大学 信息科学与工程学院,江苏 常州 213164)

[2](南京理工大学 计算机科学与技术学院,江苏 南京 210094)

[3](University of Washington, Smart Transportation Application and Research Laboratory, Seattle, WA 98195, USA)

通讯作者: 游静, E-mail: 278995265@qq.com

摘 要: 分布式动态信任模型作为适用于云计算环境下的访问管理机制已经得到广泛研究,然而现有的许多信任模型忽视了对信任数据可靠性的评估,导致推荐信任不可靠时出现模型失效.针对这一问题,提出了一种考虑信任可靠度的分布式动态信任管理模型 DDTM-TR.首先使用可靠度对信任进行评估,降低不可靠数据对直接信任、推荐信任、综合信任计算的影响;然后,选择多个待选节点计算它们的综合信任,并以计算出的综合信任为概率,随机选择待选节点进行交互;在交互结束后,根据交互满意度反馈修正节点的可靠度.仿真实验结果表明:DDTM-TR 模型在处理恶意服务、恶意推荐方面都优于对比模型,并且能够通过反馈算法进一步降低判断的失败率.

关键词: 动态信任模型;可靠度;DDTM-TR;信任管理;推荐信任

中图法分类号: TP316

中文引用格式: 游静,上官经伦,徐守坤,李千目,王印海.考虑信任可靠度的分布式动态信任管理模型.软件学报,2017,28(9):2354−2369. http://www.jos.org.cn/1000-9825/5180.htm

英文引用格式: You J, Shangguan JL, Xu SK, Li QM, Wang YH. Distributed dynamic trust management model based on trust reliability. Ruan Jian Xue Bao/Journal of Software, 2017,28(9):2354−2369 (in Chinese). http://www.jos.org.cn/1000-9825/5180.htm

## Distributed Dynamic Trust Management Model Based on Trust Reliability

YOU Jing[1], SHANG-GUAN Jing-Lun[1], XU Shou-Kun[1], LI Qian-Mu[2], WANG Yin-Hai[3]

[1](School of Information Science & Engineering, Changzhou University, Changzhou 213164, China)

[2](School of Computer Science and Engineering, Nanjing University of Science & Technology, Nanjing 210094, China)

[3](University of Washington, Smart Transportation Application and Research Laboratory, Seattle, WA 98195, USA)

**Abstract**: Distributed dynamic trust model, as a new access management mechanism which applies to cloud computing environment, has been studied extensively. However, many existing trust models ignore the reliability of trust data and lead to failure when facing malicious recommendation. To solve this problem, this article proposes a new model named DDTM-TR (distributed dynamic trust management model based on trust reliability). Firstly, in order to reduce the bad effects of unreliable data on direct trust, recommendation trust and integrated trust, the reliability of trust data is evaluated. Second, several candidate nodes are selected and their integrated trust values are calculated, and one of them is selected randomly according to the stochastic decision algorithm based on their integrated trust values. Finally, the node reliability is updated according to the feedback after the interaction. The experiments demonstrate that

* 基金项目: 国家自然科学基金(612724219); 江苏省自然科学基金(BK2009535); 江苏省高校优秀中青年教师境外研修计划

Foundation item: National Natural Science Foundation of China (612724219); Natural Science Foundation of JiangSu Provincial of China (BK2009535); Jiangsu Overseas Research & Training Program for University Prominent Young & Middle-Aged Teachers and Presidents

收稿时间: 2016-07-09; 修改时间: 2016-09-04, 2016-11-10; 采用时间: 2017-01-06; jos 在线出版时间: 2017-02-20

CNKI 网络优先出版: 2017-02-20 14:09:04, http://www.cnki.net/kcms/detail/11.2560.TP.20170220.1409.022.html

DDTM-TR model performs better than the compared models in resisting malicious service and malicious recommendation, and the failure rate can be reduced further by the feedback algorithm.

**Key words**: dynamic trust model; trust reliability; DDTM-TR; trust management; recommendation trust

近年来云计算迅速发展,但由于云计算环境的开放性、网络服务的多样性、网络实体的自主性等特点,服务交互往往发生在两个陌生的网络实体之间,因此,一些安全问题始终制约着云计算的发展.比如,如何确定网络实体身份的真实性?如何确定网络服务的真实性和可靠性?如何选择真实可靠的服务、避免恶意服务?这些问题最终会归结为实体间的信任管理问题.

研究发现,信任管理问题是各类分布式计算系统面临的共同问题.信任管理模型通过在服务交互之前计算信任度的方法,帮助服务请求者从众多云服务中选择安全、有效、质优的服务[1].信任管理模型研究阶段分为静态信任模型阶段和动态信任模型阶段.

- 静态信任模型为早期研究,用于保证单机系统的可信问题.静态信任管理模型首先构建基础的信任根,然后从信任根开始到硬件平台、操作系统、应用程序,一层认证一层,一层信任一层,把这种信任扩展到整个计算机系统,从而确保整个计算机系统的可信.但云计算的开放性使得层次信任过程难以实现,匿名性使得网络节点身份难以验证,自主性使得信任根难以确立,因此,静态信任管理模型不适用于云计算网络;
- 动态信任管理模型是参考了人类社会中的信任关系而提出的用于解决分布式网络的信任管理模型.动态信任管理模型认为信任关系随时间变化而变化,所以动态信任管理模型一般有以下目标:(1) 实时对信任关系进行评价;(2) 有效抑制恶意节点破坏信任评价;(3) 新加入节点不受原有节点“孤立”,同样可以累积信任.

动态信任模型从网络组织形式上分为集中式信任模型和分布式信任模型.集中式信任模型由一个中心节点统一管理全域范围内的实体信任信息.目前电子商务系统如 Amazon、eBay、阿里巴巴、京东等均采用的是集中式的信任管理.这类信任模型存在的问题有:没有对评价的可信度加以区分;缺乏时间适用性;存在单点失效;不易扩展等.分布式信任模型是依据人类社会当中的信任关系建立的模型,在这类模型中,网络节点独立维护自己的信任数据,无需中心节点进行管理.已有许多研究人员针对 P2P 网络、Ad hoc 网络提出了分布式、半分布式的信任管理策略,有些结合传统的加密技术保证用户的身份,取得较好的效果.分析认为,采用集中式的信任管理策略会成为云计算系统可扩展性的瓶颈,分布式信任模型的工作模式更适合云计算的不可预期性和易于扩展性.但是,已有的分布式信任管理模型还远不能满足云计算环境的要求,需要在已有信任管理研究成果的基础上探索新的信任管理模型来解决云计算中的信任问题.

目前,已有许多学者对分布式动态信任模型进行了研究.Melaye[2]、Wang[3]、孙玉星[4]等人分别提出了各自的基于贝叶斯的信任模型,这些模型使用贝叶斯公式计算信任传递,但未考虑时间对信任的影响,忽视了信任的动态性.Liu[5]提出了一种同时考虑了正面评价、负面评价和交互历史的推荐信任模型,但是存在 3 点不足[6]:

(1) 没有考虑节点动态进入和退出网络;

(2) 没有区分信任度和可靠度,模型反映出信任值越高越可靠;

(3) 没有考虑时间因素对反馈信任的影响.

EigenTrust[7]模型是目前为止最经典的信任管理模型之一,它通过节点间推荐度的迭代计算目标节点的全局信任,并由 Lu 等人[8,9]从运行速度、网络负载等方面对该模型进行了改进.EigenTrust++模型[10]对 EigenTrust 模型的信任计算方法进行了改进,提出了信任相似度的概念,使用信任相似度来区分正常推荐和恶意推荐,增强了模型对抗恶意推荐的能力.

信任计算时,上述模型大多考虑了其他节点的推荐信任来弥补分布式系统中单一节点直接数据不足的缺点.文献[2–4]使用贝叶斯概率公式处理推荐信任,贝叶斯概率公式是利用原有数据进行推理的常用手段,但在信任计算过程中,贝叶斯概率公式中的先验概率难以预估.文献[2–4]使用专家经验代替,导致结论的主观性较强,对专家经验的可靠性也要求较高,所以在实际的电子商务领域中,基于贝叶斯的信任模型应用受到较大的限制.

文献[5,7–9]考虑了可靠度对推荐信任的影响,但是这些模型默认高信任度的节点拥有高可靠度的推荐.显然,这一观点不总是成立[6],特别是当恶意节点开始协同欺骗时,这类模型就会出现更严重的问题.EigenTrust++模型使用相似度来代替可靠度,但相似度也存在一些问题,如相似与否的阈值选择、信任相似度对正常推荐和恶意推荐的区分度等.

本文就以上分析提出了新的分布式动态信任管理模型 DDTM-TR.为了保证云计算的易扩展性,新模型使用分布式信任模型的思想,每个网络节点负责管理自己的信任数据,不存在中心节点;针对信任动态性的特点,新模型使用动态信任模型的概念,信任会随着时间的衰减,节点也可以动态地加入和离开;针对恶意节点协同欺骗的问题,本文提出了信任可靠度的概念,将信任度和可靠度进行区分.在 DDTM-TR 模型中,时间久远或信任来源不可靠等情况会导致信任度评价的不可靠(可靠度降低),而不是待评价节点是不可信任的(信任度不变).经实验验证:新模型即使是在高质量服务进行恶意推荐时,仍然能够保证模型的可靠性.

## 1 考虑了信任可靠度的分布式动态信任管理模型

为了方便描述,本文将网络中提供服务的节点称为服务提供者,将网络中寻求服务的节点称为服务需求者,如果网络中某个节点既提供服务,又寻求服务,那么它既是服务提供者,又是服务需求者,相应地,这些节点属于服务提供者集合和服务需求者集合.

### 1.1 基本概念定义

从不同的角度对信任有不同的定义,本文对信任做出如下说明和定义.

定义 **1**. 信任是主体使用信任模型预测客体当前服务能力的行为.令 $US$ 表示用户集,$SP$ 表示服务集,若节点 $A\in US$,节点 $B\in SP$,则 $A$ 对 $B$ 的信任被描述为 $A\rightarrow B$.在本模型中,使用信任度 $T$ 和可靠度 $CT$ 来表达信任.

定义 **2**. 满意度是指在交互完成之后,主体节点根据本次交互的服务质量(服务响应时间、可用性等)做出的评价.根据信任计算时满意度数据的来源,可以将信任划分为直接信任和推荐信任.满意度取值范围为[0,1]:0 表示很不满意,1 表示很满意.

定义 **3**. 信任度描述主体节点对目标节点的服务能力预期判断,信任度只受满意度的影响,代表该节点对其他节点的服务能力的评价.信任度的取值范围为[0,1]:0 表示绝对不信任,1 表示绝对信任.

定义 **4**. 信任可靠度描述信任度的可靠程度.信任可靠度受时间、信任来源等因素的影响.随着时间的增加或信任来源不可靠,根据满意度得出的观点(信任度)不会发生变化,但可靠度会随之降低,表达这一观点不可靠.可靠度的取值范围为[0,1]:0 表示信任度观点绝对不可靠,1 表示信任度观点绝对可靠.

定义 **5**. 直接信任是节点基于本地的数据作出判断的行为,根据定义 1 中的描述,$A$ 对 $B$ 直接信任的信任度记为 $DT(A\rightarrow B)$,可靠度记为 $CT_{DT}(A\rightarrow B)$.

定义 **6**. 推荐信任是节点综合多个其他节点的直接信任作出判断的行为,根据定义 1 中的描述,$A$ 对 $B$ 推荐信任的信任度记为 $RT(A\rightarrow B)$,可靠度记为 $CT_{RT}(A\rightarrow B)$.

### 1.2 分布式动态信任管理的体系

在本文中,分布式动态信任管理体系如图 1 所示.

在本模型下,主体节点 $A$ 寻求目标服务经历以下几个步骤.

步骤 1. 信任评价.信任评价算法预测目标节点的信任度并最终做出选择.在 DDTM-TR 模型中,主体节点 $A$ 从服务节点集 $SP$ 中选取多个待选服务节点,分别计算出它们的直接信任、推荐信任和综合信任,并依据综合信任选择最终交互的服务节点.信任计算方法将在第 2 节介绍,信任评价算法将在第 3 节介绍;

步骤 2. 交互.主体节点 $A$ 依据信任评价结果与最终选择的目标节点进行交互,然后根据交互节点的服务状态做出满意度评价;

步骤 3. 信任反馈.信任模型依据上一步得到的满意度评价添加并修正本地信任数据,实现信任反馈.信

任反馈算法将在第 4 节介绍.

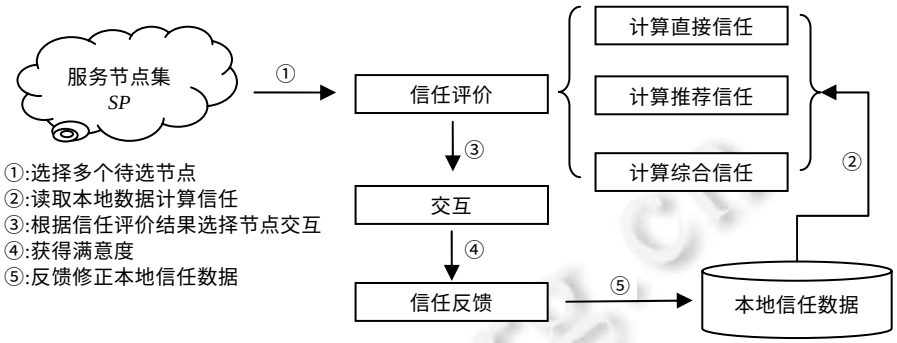



Fig.1 Schematic diagram of distributed dynamic trust management system

图 1 分布式动态信任管理的体系示意图

## 2 信任计算

### 2.1 本地数据存储

在 DDTM-TR 模型中,信任评价的过程无论是计算直接信任还是计算推荐信任,均需要本地数据的参与,主体节点需要存储以下 3 类数据.

1) 交互的历史记录 $H_{all}$,从中读取出直接信任计算所需的序列 $H$,包含交互目标节点信息、交互满意度数据、交互时间;
2) 节点推荐数据 $R_{all}$,用于存储从网络中获得的序列 $R_2$,包含着交互目标节点信息、推荐节点信息、交互的满意度、交互时间;
3) 节点可靠度序列 $CN_{all}$,从中读取出推荐信任计算所需的序列 $CN$,包含推荐节点信息、该节点的推荐可靠度、该节点推荐准确的次数 $ns$ 和推荐不准确的次数 $nf$.

以上 3 类数据由节点本地保存,并在第 4 节介绍的信任反馈算法得到更新.

### 2.2 直接信任计算

直接信任受本地信任数据的影响,时间因素也会影响信任数据的可靠度.

如图 2 所示,主体节点 $A$ 和待信任评价的目标节点 $B$ 之间的直接信任计算步骤如下.

- 步骤 1:读取节点 $A$ 本地存储的满意度.

节点 $A$ 从本地存储序列 $H_{all}$ 中读取关于目标节点 $B$ 的满意度,记为序列 $H$.序列 $H=\{h_1,h_2,\ldots,h_{dn}\}$,$dn$ 为交互次数.序列 $H$ 中,每个元素 $h_i$ 包含着当时的服务满意度 $sat_i$ 和交互时间 $t_i$.

- 步骤 2:计算本地满意度的可靠度.

信任具有时效性,随着时间的延长,满意度 $sat_i$ 结论的可靠度将从可靠变为不可靠.为了模拟这个过程,本文令 $cr_i=\theta(t-t_i)$,其中:$cr_i$ 表示历史服务满意度 $sat_i$ 的可靠度;$\theta(t-t_i)$为时间影响函数,$t$ 代表当前时间,$t_i$ 为记录 $h_i$ 发生的时间,时间影响函数的具体表达方式需要根据具体的应用场景进行选择.

- 步骤 3:计算直接信任的信任度 $DT(A\rightarrow B)$.

如果某一满意度有较高的可靠度,说明参考价值高,应该提高该数据对 $DT(A\rightarrow B)$结果的影响;反之,若本地满意度的可靠度较低,说明参考价值低,应该降低该数据对 $DT(A\rightarrow B)$的影响.

综上,$DT(A\rightarrow B)$的计算公式见公式(1).

$$DT(A\to B)=\begin{cases}\sum_{i=1}^{dn}\dfrac{cr_i}{\sum_{j=1}^{dn}cr_j}sat_i, & dn>0\\ 0.5, & dn=0\end{cases} \tag{1}$$

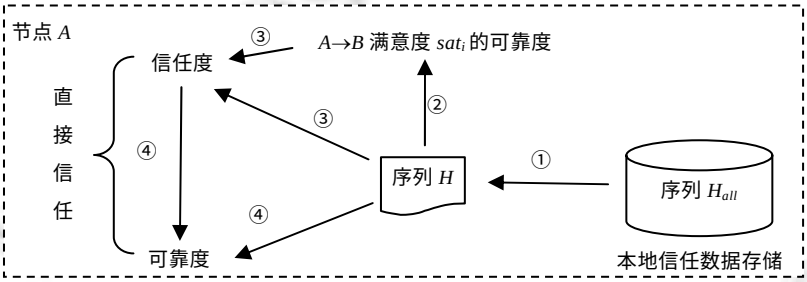
当没有历史记录时取 0.5,表达既非“信任”也非“不信任”.

- 步骤 4:计算直接信任的可靠度 $CT_{DT}(A\to B)$.

如果实体 $B$ 和主体节点 $A$ 满意度的离散度较低,说明节点服务稳定,$DT(A\to B)$和当前真实服务质量预计偏差较小,即 $DT(A\to B)$结论拥有较高的可靠度;反之,如果满意度的离散度较大,$DT(A\to B)$与当前真实服务质量可能存在较大的偏差,所以认为此时的 $DT(A\to B)$结论不可靠.根据上文描述,直接信任的可靠度与满意度的离散度有关,直接信任可靠度的计算方法见公式(2).

$$CT_{DT}(A\to B)=\begin{cases}\dfrac{1}{\sum_{i=1}^{dn}(DT(A\to B)-sat_i)^2+1}, & dn>0\\ 0, & dn=0\end{cases} \tag{2}$$

当没有历史记录时取值为 0,表示历史记录不可靠.




①:节点 $A$ 读取本地关于节点 $B$ 的满意度
②:计算满意度的可靠度
③:计算直接信任的信任度 $DT(A\to B)$
④:计算直接信任的可靠度 $CT_{DT}(A\to B)$

Fig.2 Computing process of direct trust

图 2 直接信任计算过程

### 2.3 推荐信任计算

推荐信任受两个方面影响:

- 一是推荐节点提供的信任信息,包含对目标节点的信任度和该信任度对应的可靠度;
- 二是主体节点眼中推荐节点的可靠度.

#### 2.3.1 推荐节点选择

本文推荐节点的选择采用前期研究[11]的方法,网络中的推荐节点来源于两个方面:

(1) 主体节点选择 $n$ 个推荐节点,满足两个条件:① 推荐节点与目标节点曾经发生交互;② 推荐节点为主体节点中节点可靠度最高的 $n$ 个节点之一;

(2) 当情形(1)中的节点数量不足 $n$ 时,目标节点向主体节点推荐其可信任的用户节点,主体节点随机选取这些节点中的一部分作为目标节点的自荐节点.

#### 2.3.2 推荐信任计算流程

如图 3 所示,主体节点 $A$ 和待信任评价的目标节点 $B$ 之间的推荐信任计算步骤如下.

- 步骤 1:获取并处理关于目标节点 $B$ 的推荐数据.

本文推荐节点来源有两种:一是推荐路径上的节点,来自于第 2.3.1 节中的描述,记这些节点的推荐信息为序列 $R_1$;二是反馈推荐的节点,推荐信息保存在主体节点 $A$ 的本地存储序列 $R_{all}$ 中,利用第 2.2 节直接信任的计

算方法计算出每个推荐节点的直接信任度与可靠度,将处理后的序列记为序列 $R_2$.若推荐节点数量为 $rn$,令序列 $R=\{r_1,r_2,\ldots,r_{rn}\}$表示这些推荐节点的推荐记录,有 $R=R_1\cup R_2$(若推荐节点同时存在与序列 $R_1$ 与序列 $R_2$,优先使用序列 $R_1$ 中的数据).$r_i$ 包含 $rdt_i$,$rcr_i$:$rdt_i$ 表示推荐节点 $i$ 对目标节点的直接信任的信任度;$rcr_i$ 表示推荐节点直接信任的可靠度,推荐节点通过 $rcr_i$ 向主体节点表达自己对 $rdt_i$ 的确定程度.

- 步骤 2:从本地 $CN_{all}$ 序列中读取对于推荐节点的可靠度评价.

在推荐信任中,主要使用的数据来自其他节点而非主体本身,所以主体节点 $A$ 需要根据推荐来源判断数据的可靠度.主体节点从本地信任存储 $CN_{all}$ 序列中读取序列 $CN$.序列 $CN=\{cn_1,cn_2,\ldots,cn_{rn}\}$表示节点的推荐可靠度,如果主体没有节点 $i$ 的推荐记录,$cn_i$ 取默认值 0.5.

- 步骤 3:计算节点推荐的综合可靠度 $rc_i$.

主体节点对 $rdt_i$ 需要从两个方面考虑它的可靠度:1) 推荐节点对 $rdt_i$ 的可靠度 $rcr_i$;2) 主体节点对推荐节点的可靠度 $cn_i$.综合以上两点,本文使用 $rc_i=rcr_i\times cn_i$ 表达 $rdt_i$ 对应的可靠度.

- 步骤 4:计算推荐信任的信任度 $RT(A\rightarrow B)$.

如果推荐节点 $i$ 直接信任的信任度 $rdt_i$ 对应的可靠度 $rc_i$ 较高,说明推荐节点有较大的把握且主体节点认为推荐节点可靠,所以 $rdt_i$ 具有较高的参考价值,需要提高 $rdt_i$ 对 $RT(A\rightarrow B)$计算结果的影响;反之,如果 $rdt_i$ 对应的可靠度 $rc_i$ 较低,说明推荐节点不确定其推荐或者主体节点认为推荐节点不可靠,则 $rdt_i$ 参考价值较低,需要降低其对 $RT(A\rightarrow B)$计算结果的影响.根据上文描述,本文推荐信任的信任度的计算公式见公式(3).

$$RT(A\rightarrow B)=\sum_{i=1}^{rn}\frac{rc_i}{\sum_{j=1}^{rn}rc_j}rdt_i \tag{3}$$

- 步骤 5:计算推荐信任的可靠度 $CT_{RT}(A\rightarrow B)$.

根据现实社会的信任习惯,人们觉得大家公认的观点更加可靠,同时认为包含着大量分歧的观点不可靠.当推荐节点对某一目标节点观点之间的离散度较低,说明 $RT(A\rightarrow B)$的结论是推荐节点公认的观点,拥有较高的可靠度;反之,如果推荐节点之间观点的离散度较高,说明推荐节点之间存在着大量的分歧,$RT(A\rightarrow B)$的结论比较不可靠.所以,推荐信任可靠度 $CT_{RT}(A\rightarrow B)$的计算方法见公式(4).

$$CT_{RT}(A\rightarrow B)=\frac{1}{\sum_{i=1}^{rn}(RT(A\rightarrow B)-rdt_i)^2+1} \tag{4}$$

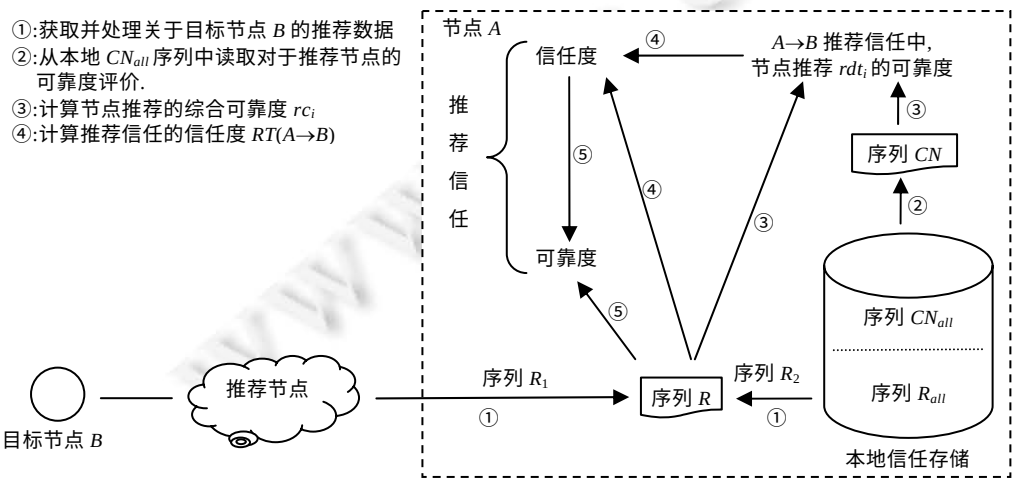



Fig.3 Computing process of recommandation trust

图 3 推荐信任计算过程

### 2.4 综合信任计算

本文中,节点的综合信任由直接信任和推荐信任构成.定义如下.

$$T(A\to B)=aDT(A\to B)+(1-a)RT(A\to B) \tag{5}$$

其中,$a$ 表示直接信任权重,由以下公式计算得到:

$$a=\frac{CT_{DT}(A\to B)}{CT_{DT}(A\to B)+CT_{RT}(A\to B)}.$$

一般来说,人们往往更容易相信自己的判断,认为直接信任比推荐信任更为可靠.但实际环境中,也可能出现直接经验过少导致直接信任比推荐信任更加不可靠的情况,因此往往很难确定 $a$ 的值.现有的方法往往采用专家打分、仿真实验结果等手段确定,缺乏科学性、合理性和自适应性.本文通过直接信任可靠度和推荐信任可靠度动态调节权重因子,如果直接信任更加可靠,则直接信任占有更大比重;如果推荐信任更加可靠,则推荐信任占有更大比重.

## 3 信任评价

### 3.1 决策算法

为了提高交互成功的几率,每次交互之前都会选择多个待选客体计算综合信任,根据综合信任作出决策.目前存在以下两种算法.

(1) 最高信任优先算法:从待选节点中选择信任度最高的节点进行访问;

(2) 基于信任度的随机算法:若$\{t_1,t_2,\ldots,t_{tn}\}$为 $t_n$ 个待选节点的综合信任度,选择节点时以 $t_i\Big/\sum_{j=1}^{t_n}t_j$ 的概率选择节点 $i$ 进行交互.

最高信任优先算法存在以下两点不足:① 信任度最高的节点容易陷入“被选择-获得信任-又被选中”的死循环中,而其他节点没有机会提高自己的信任度;② 信任度最高的节点承受较大的访问压力.

由于以上原因,本文选择基于信任度的随机算法决策信任.

### 3.2 信任评价算法

信任评价算法选取多个待选节点,计算待选节点的综合信任,并使用决策算法选择节点交互,信任评价算法如下.

*TrustDecided*(*A*) { //*A* 为主体节点

1. 从目标服务集合 $SP=\{sp_1,sp_2,\ldots,sp_{sn}\}$中选取 $m'$个节点作为待选节点,得到集合 $SP'=\{sp'_1,sp'_2,\ldots,sp'_{m'}\}$;
2. foreach $sp'_i$ in $SP'$ {
3. 从 $H_{all}$ 中读取节点 $A$ 对节点 $sp'_i$ 的序列 $H$ 并以此计算直接信任.
4. 记 $TR_1$ 表示节点 $A$ 关于目标节点 $sp'_i$ 的推荐节点集合(根据第 2.3.1 节中的描述选择).
5. 记 $TR_2$ 表示信任反馈中曾向节点 $u$ 反馈信任的节点集合.
6. 记 $TR_1$ 节点的推荐记录为序列 $R_1$,$TR_2$ 节点的推荐记录为序列 $R_2$,其中,$R_1$ 来自于网络,$R_2$ 来自于第 2.1 节描述的本地存储 $R_{all}$.令序列 $R=R_1\cup R_2$,并根据第 2.3.2 节描述的方法计算推荐信任.
7. 计算综合信任度 $T(A\to sp'_i)$.

}

8. 使用基于信任度的随机算法进行交互决策,根据决策结果选择交互对象.

## 4 信任反馈

因为信任具有动态性和模糊性,所以每次交互结束后都需要及时调整,使下次可以得出更加精确的结论.在本模型中,信任反馈算法需要更新本地存储历史记录 $H_{all}$、节点推荐数据 $R_{all}$ 和节点可靠度序列 $CN_{all}$.

### 4.1 更新历史记录$H_{all}$

设 $V$ 为本次交互后主体节点对被交互节点的交互满意度,为了简化起见,本文假设交互之后主体节点可以获知被交互节点真实的服务质量,并将其当作本次交互的满意度.随后,主体将交互满意度 $V$ 加入 $H_{all}$ 中对应位置,并填写交互目标、交互时间等信息.

### 4.2 更新节点推荐数据$R_{all}$

为了增加网络中的推荐合作,主体节点同样会将交互结果向推荐节点发送.同时,主体节点作为推荐节点为他人推荐时也会获得他人的回报,主体节点接收他人回报的满意度数据,并将其当作他人的推荐写入 $R_{all}$,填写推荐节点、交互时间等信息.

### 4.3 更新节点可靠度序列$CN_{all}$

当推荐失败时,为了防止欺诈行为,应当适当惩罚推荐节点.另外,当推荐成功时,也要适当奖励推荐节点.本文使用推荐 $rdt_i$ 和交互满意度 $V$ 的绝对值表示推荐误差,$\varepsilon$表示主体对推荐误差的容忍度,如果$||V-rdt_i||<\varepsilon$,认为节点进行了正确的推荐,对推荐节点进行可靠度奖励;反之,认为节点进行了错误推荐,对推荐节点进行可靠度惩罚.

假设需要对节点 $k$ 计算可靠度奖惩,使用公式(6)计算惩罚的情况,使用公式(7)计算奖励的情况.

$$cn_k^{new} = cn_k^{old} - M_f \times Q \times rcr_k \times p_f \times p_n \tag{6}$$

$$cn_k^{new} = cn_k^{old} + M_s \times Q \times rcr_k \times p_s \times p_n \tag{7}$$

其中,

- $M_f$ 与 $M_s$ 表示改变量的最大值.为保证 $cn_k^{new}$ 仍在可靠度的取值范围中,本文令:

$$M_f = cn_k^{old}, M_s = 1 - cn_k^{old};$$

- $Q$ 表示推荐误差对奖惩幅度的影响.如图 4 所示:当误差$||V-rdt_i||<\varepsilon$时,对推荐节点可靠度实行奖励,误差越大奖励越小;当误差$||V-rdt_i||>\varepsilon$时,对推荐节点可靠度实行惩罚,误差越大惩罚越大;
- $rcr_k$ 为节点进行推荐时其直接信任的可靠度,该数据对奖惩幅度存在影响.就恶意协同节点来说,它必须伪造一个较高的推荐可靠度来增加误导主体节点的可能性,因此,增加对这种误导行为的惩罚力度可以有效地抑制恶意协同.此外,正常节点也有可能进行不确定推荐,当正常节点表达了自己的不确定之后,主体节点根据 $rcr_k$ 的值降低惩罚力度;
- $p_f$和 $p_s$表示节点 $k$ 的失败推荐比例和成功推荐比例:如果节点推荐失败的概率很高,则该节点在以后的推荐中更加难以获得可靠度的提高;反之,如果节点对主体节点保持较高的推荐准确度,偶尔一次失误不会造成可靠度大幅度地降低;
- $p_n$ 表示推荐次数对奖惩力度的影响.随着推荐次数的增多,单次推荐对推荐节点可靠度的影响会逐渐降低.假设 $n$ 为某一节点向主体的推荐次数,本文使用 $p_n = 1/\sqrt{n}$ 表达推荐次数对奖惩力度的影响.

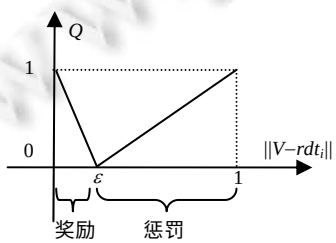



Fig.4 Relations between the recommendation errors and factor $Q$

图 4 推荐误差与因子 $Q$ 的关系

当计算完成之后,主体节点将 $cn_k^{new}$ 写入 $CN_{all}$ 中对应位置.

### 4.4 信任反馈算法

信任反馈算法实现向其他节点发送交互结果,更新本地信任数据两个功能,算法如下.

*TrustFeedback*($V$,$\varepsilon$) {

1. 将交互满意度 $V$ 及交互时间 $t$ 加入本地存储 $H_{all}$.
2. 令 $RN=TR_1\cup TR_2$.
3. foreach $rn_i$ in $RN$ {
4. 向 $rn_i$ 发送满意度 $V$ 及交互时间 $t$,$rn_i$ 节点接收后,将满意度 $V$ 和交互时间 $t$ 加入 $R_{all}$ 序列中.
5. 令 $n$=节点 $rn_i$ 向主体推荐的次数. //计算 $pn$(推荐次数对奖惩力度的影响), $ps$ 和 $pf$(成功/失败比例)的必要数据.
6. if $\Delta=\|V-rdt_{rn_i}\|>\varepsilon$ {
7. 使用公式(6)计算新的可靠度 $cn_{rn_i}$ .

} else {

8. 使用公式(7)计算新的可靠度 $cn_{rn_i}$ .

}

9. 将新的可靠度 $cn_{rn_i}$ 记录加入 $CN_{all}$.

}

}

## 5 实验分析

### 5.1 仿真系统

本文使用仿真实验验证模型性能,使用仿真系统模拟一个分布式网络,仿真系统基于 Java 实现.

#### 5.1.1 仿真节点

仿真系统将网络中的节点转化为抽象节点.抽象节点中包含节点编号、节点类别、节点真实信任度、邻居节点的位置等信息,节点可以通过递归询问邻居节点的方式访问目标节点,并且可以使用不同的信任模型进行决策.此外,节点还拥有独立的存储空间来存储本地数据.

#### 5.1.2 节点类型

仿真网络中包含着正常节点和恶意节点:正常节点提供正常服务并进行正常推荐;恶意节点则会进行恶意推荐,包括夸大、诋毁目标节点,还会与其他恶意节点进行协同欺骗,或者直接提供恶意服务.在本次实验中,节点分为以下 4 种,分别执行恶意服务、夸大攻击、诋毁攻击和正常服务.

- $A$ 类恶意节点:该类型恶意节点的真实服务质量为 0.1.若需要对其他 $A$ 类恶意节点进行推荐,采用夸大协同策略,将用于推荐的直接信任的信任度伪造为 $rdt'=1-rdt$;若需要对 $D$ 类正常节点进行推荐,采用恶意诋毁策略,将用于推荐的直接信任的信任度伪造为 $rdt'=1-rdt$;
- $B$ 类恶意节点:该类型恶意节点的真实服务质量在区间(0.7,1)之中.$B$ 类恶意节点会随机选择 50%的 $A$ 类恶意节点作为其合作伙伴.$B$ 类恶意节点对合作伙伴的 $A$ 类恶意节点提供夸大协同,将用于推荐的直接信任的信任度伪造为 $rdt'=1-rdt$;对于其他非合作伙伴的 $A$ 类恶意节点和 $D$ 类正常节点保持正常推荐来隐藏自己恶意协同的身份;
- $C$ 类恶意节点:该类型恶意节点的真实服务质量在区间(0.7,1)之中.$C$ 类恶意节点对 $D$ 类正常节点采用恶意诋毁策略,将用于推荐的直接信任的信任度伪造为 $rdt'=1-rdt$.对于 $A$ 类恶意节点保持正常推荐来隐藏自己恶意诋毁的身份;
- $D$ 类正常节点:该类型的节点的真实服务质量在区间(0.7,1)之中并保持着正常推荐,即:推荐时给出自

己直接信任的信任度,不进行数据伪造.

综上,仿真网络中,节点采取的策略见表 1.

**Table 1** Strategies of nodes

表 **1** 节点策略

| 节点类型 | 服务 | 推荐 |
|---|---|---|
| *A* 类恶意节点 | 恶意服务 | 对其他 *A* 类恶意节点进行夸大,对 *D* 类正常节点进行诋毁 |
| *B* 类恶意节点 | 正常服务 | 对 *A* 类恶意节点进行夸大 |
| *C* 类恶意节点 | 正常服务 | 对 *D* 类正常节点进行诋毁 |
| *D* 类正常节点 | 正常服务 | 正常推荐 |

5.1.3 判定规则

仿真系统随机选择抽象节点寻求交互服务,被选中的节点在仿真网络中寻找提供目标服务的节点,并使用仿真系统当前测试的信任模型计算信任.最后,根据信任模型得出的结论选择节点进行交互.

当节点作出选择之后,仿真系统会对信任选择的结果进行判定,判定规则见表 2.最后,仿真系统记录判定结果.多轮多次交互之后,仿真系统输出交互成功率作为当前测试信任模型的评价结论.

**Table 2** Decision rules of trust selection in the simulation system

表 **2** 仿真系统判定信任选择是否成功的规则

| 交互节点类型 | 判定结果 |
|---|---|
| *A* 类恶意节点 | 交互失败(节点提供恶意服务) |
| *B* 类恶意节点 | 交互成功(节点提供正常服务) |
| *C* 类恶意节点 | 交互成功(节点提供正常服务) |
| *D* 类正常节点 | 交互成功(节点提供正常服务) |

## 5.2 参数设置

DDTM-TR 算法、EigenTrust 算法和 EigenTrust++算法均需要设置一些外部参数才能正常运行,实验时,各个模型的参数设置见表 3.

**Table 3** Parameter settings

表 **3** 参数设置

| 参数设置 | DDTM-TR | EigenTrust++ | EigenTrust |
|---|---|---|---|
| 初始邻居节点数量 | 3 | 3 | 3 |
| 交互轮次数量 | 30 | 30 | 30 |
| 每轮交互数量 | 50 | 50 | 50 |
| 推荐误差容忍度$\varepsilon$ | 0.4 | 无 | 无 |
| 时间影响函数 | $e^{-0.005\Delta t}$ | 无 | 无 |
| 相似度影响参数$\beta$ | 无 | 0.85 | 无 |
| 实验次数 | 5 | 5 | 5 |

## 5.3 实验与分析

本文设计 6 组对比实验,分别从不同规模下的计算开销、节点存储复杂度、恶意服务攻击、夸大协同攻击、诋毁攻击、持续改进这 6 个方面分析验证 DDTM-TR 模型,并对冒名攻击进行了简要分析.

5.3.1 计算开销实验

查找信任计算所需的推荐数据是信任模型计算开销的主要因素[12],因此,实验采用所查找的节点数量来表示模型的计算开销.在本组实验中,当网络规模从 100 增长到 1 000,进行 30 轮,每轮 50 次的交互,DDTM-TR 模型、EigenTrust 模型和 EigenTrust++模型查找的节点数量如图 5 所示.

从图 5 中可以看出,EigenTrust 模型、EigenTrust++模型与 DDTM-TR 查找的节点数量差距很大.其中,EigenTrust 模型最多,EigenTrust++模型次之,DDTM-TR 模型最低.EigenTrust 模型迭代整个网络以计算全局信任,所以计算开销很大.EigenTrust++模型虽然也迭代了整个网络,但它去除了相似度较低的节点及该类节点的

朋友节点,所以计算开销相较 EigenTrust++模型低.由于云计算应用环境的服务多样性和用户分散性,使得计算全局信任不可行.在 DDTM-TR 模型中,只使用本地已存储的推荐信息和有限数量的推荐节点计算推荐信任极大地降低了计算开销,用户还可以通过调整参与计算的数据规模获得性能和可靠性的折衷.

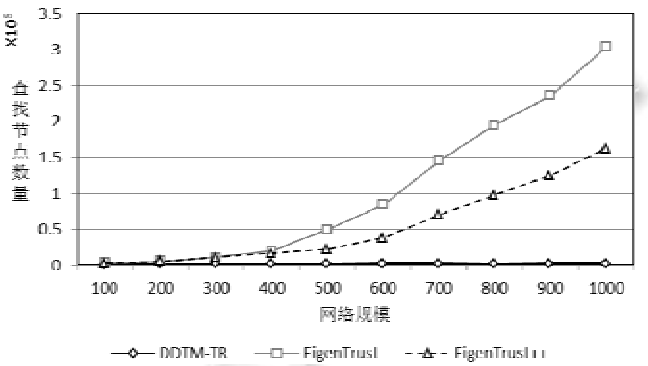



Fig.5 Computational overhead under different network size of each model

图 5 不同网络规模下各模型的计算开销

5.3.2 节点存储复杂度实验

在本组实验中,网络规模同样从 100 增长到 1 000,3 种模型分别进行 30 轮,每轮 50 次的交互,每个节点在本地平均存储的记录数如图 6 所示.

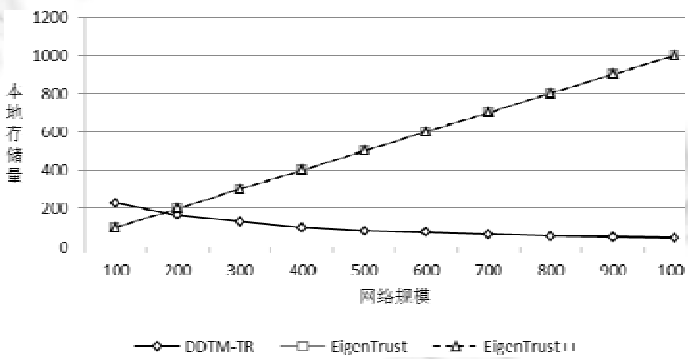



Fig.6 Average amount of local data storage of each model

图 6 各模型本地平均数据存储量对比

从图 6 中可以看出:DDTM-TR 模型本地平均数据存储量不断地减少,并趋于平缓.EigenTrust 模型与 EigenTrust++模型在计算节点 $k$ 的全局信任时,需要所有节点提供对于节点 $k$ 的局部信任以及自身的全局信任,其中,对于节点 $k$ 的局部信任来自于节点本地,自身的全局信任来自于信任档案节点.因此,随着网络规模的扩大,局部信任存储呈线性增长.在 DDTM-TR 模型中,本地数据存储量与交互次数相关.在交互总次数不变的情况下,随着网络规模的增大,交互逐渐稀疏,每个节点平均交互次数下降,导致本地平均数据存储量减少.

5.3.3 恶意服务攻击实验

在本组实验中,网络规模为 1 000 个节点,其中,$A$ 类恶意节点比例从 10%增长到 70%.3 种模型按照表 3 中设置进行交互后的交互成功率如图 7 所示.

从图 7 中可以看出,3 种模型的成功率明显好于无信任模型的情况.其中,DDTM-TR 模型略优于 EigenTrust 模型和 EigenTrust++模型.原因在于,EigenTrust 和 EigenTrust++均依赖于预信任节点.预信任节点在节点无法通过直接信任、推荐信任判断陌生节点时提供正确的判断.但在云计算网络中,预信任节点存在着以下的困难.

1) EigenTrust 将预信任节点定义为网络的构建者和最初的使用用户,认为它们会维护网络的安全性.这一观点主观性强烈,不能令人信服;
2) 在云计算网络中,用户节点扮演着预信任节点的角色,而预信任节点的身份会导致该节点需要面临更大的额外推荐开销;
3) 一旦无法保证预信任节点的匿名性,恶意节点可以对预信任节点展开针对性攻击,从而对整个网络造成巨大的破坏.

基于以上原因,本次实验中未加入预信任节点,导致 EigenTrust 和 EigenTrust++表现略不及 DDTM-TR 模型.

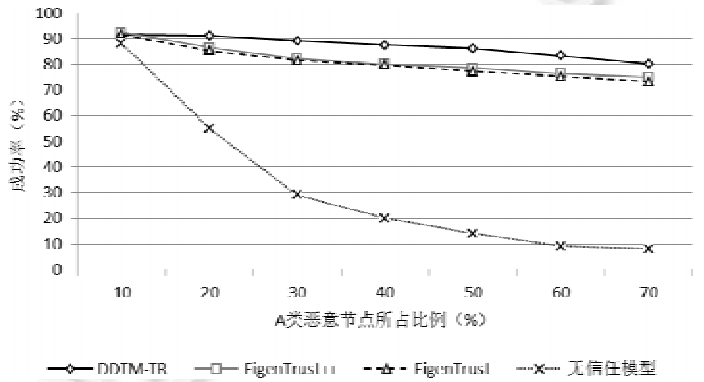



Fig.7 Success rate of interaction of each model only with malicious *A*

图 7 只存在 *A* 类恶意节点时各模型交互成功率

当 *A* 类恶意节点数量在 70%时,每一阶段轮次中,各个模型的交互成功率变化如图 8 所示.

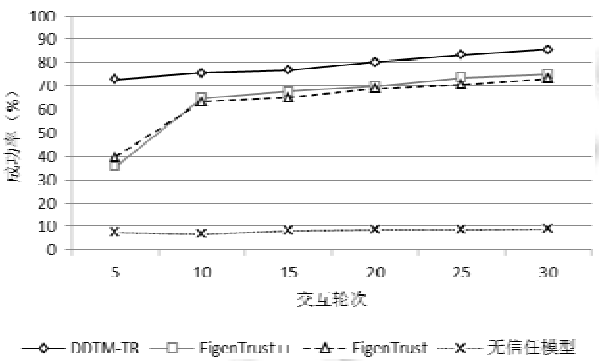



Fig.8 Success rate of interaction in different turns of each model with 70% malicious *A*

图 8 当 *A* 类恶意节点占 70%时,各模型在不同轮次的交互成功率

从图 8 可以看出:DDTM-TR 模型成功率增长较为平缓,在 25 轮~30 轮时成功率仍有缓缓提升;而 EigenTrust 模型与 EigenTrust++模型由于迭代整个网络,所以收敛迅速,并在 10 轮之后每轮的交互成功率趋平缓.同时,由于预信任节点的缺失,导致其在交互初期交互失败较多.

5.3.4 夸大协同攻击实验

在本组实验中,网络规模为 1 000,网络中加入 *A* 类和 *B* 类恶意节点,*A* 类恶意节点比例从 40%逐渐降低至 5%,*B* 类恶意节点数量从 0%逐渐增加至 35%,其他设置见表 3.这 3 种模型的交互成功率如图 9 所示.

从图 9 中可以看出:当 *B* 类恶意节点开始进行夸大协同时,EigenTrust 模型无法有效应对,甚至出现不及无信任模型的情况.原因在于:EigenTrust 模型将节点的服务质量与推荐能力等同,模型无条件信任服务质量高的

节点.在这种情况下,拥有高质量服务的 *B* 类恶意节点可以进行大量的夸大推荐而不会受到任何怀疑.

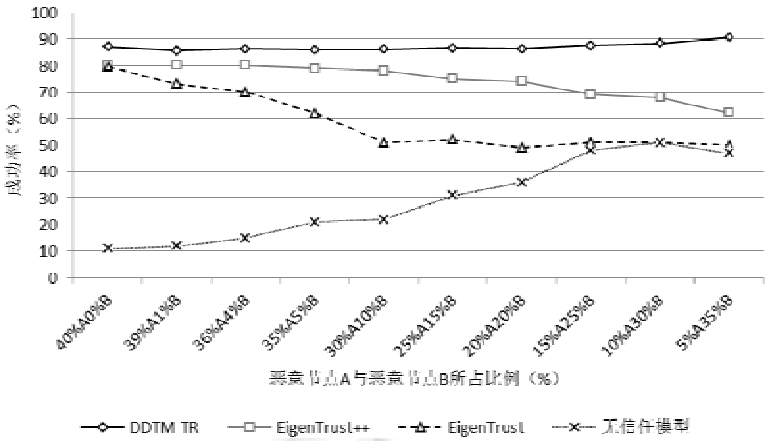



Fig.9 Success rate of interaction of each model with 40% malicious *A* and *B*

图 9 40%的恶意节点(*A* 类和 *B* 类)参与夸大协同攻击时各模型的交互成功率

EigenTrust++模型使用相似度来应对恶意推荐的问题,当 *B* 类恶意节点需要对较多的 *A* 类恶意节点进行夸大推荐时,因为此时 *B* 类恶意节点和 *D* 类正常节点相似度较低,所以算法可以轻易识别出 *B* 类恶意节点;但当 *B* 类恶意节点只需为少量 *A* 类恶意节点进行掩饰时,模型根据相似度难以确定是恶意协同还是误差累积.若敏感度较高,会导致大量 *D* 类正常节点因为相似度误差导致无法推荐;若敏感度较低,又会导致难以甄别恶意协同.

在云计算网络中,往往 *B* 类恶意节点只需为少量 *A* 类恶意节点进行服务,EigenTrust++模型的相似度更难发挥作用.

在 DDTM-TR 中,首先,服务质量和可靠度由不同的方法进行评价,解决了 EigenTrust 模型中高质量服务节点可以无条件进行恶意推荐的问题;其次,DDTM-TR 的反馈算法中存在对推荐偏差的奖惩机制,所以主体节点发现 *B* 类恶意节点夸大推荐后,主体节点惩罚其可靠度,不再信任其推荐能力,阻止 *B* 类恶意节点继续夸大协同.

图 10 展示了当 *A* 类恶意节点占 5%、*B* 类恶意节点占 35%时,各个模型在每一阶段交互的交互成功率.

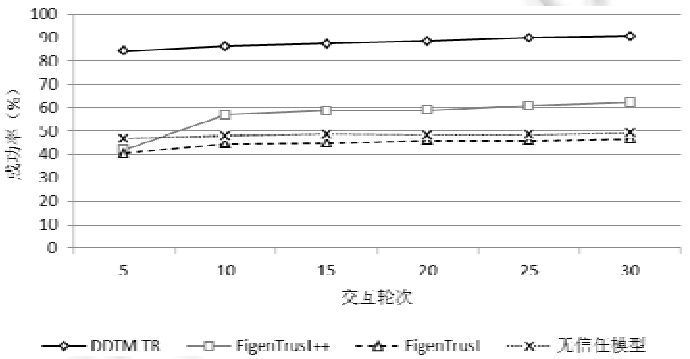



Fig.10 Success rate of interaction in different turns of each model with 5% malicious *A* and 35% malicious *B*

图 10 5%的 *A* 类恶意节点与 35%的 *B* 类恶意节点协同攻击时,各模型在不同轮次下的交互成功率

从图 10 中可以看出:由于 *B* 类恶意节点拥有较高的服务质量,EigenTrust 模型在各个轮次的交互中成功率都较低;EigenTrsut++模型虽然发挥了一定的作用,但由于 *B* 类恶意节点不针对所有的 *A* 类恶意节点进行推荐,相似度辨识较低,所以 EigeenTrust++模型的成功率不及图 8 中的表现.

5.3.5 诋毁攻击实验

在本组实验中,仿真网络规模为 1 000,仿真网络中加入 *A* 类恶意节点与 *C* 类恶意节点,*A* 类恶意节点比例从 40%逐渐降低至 5%,*C* 类恶意节点数量从 0%逐渐增加至 35%,其他设置见表 3.在这种情况下,3 种模型交互成功率如图 11 所示.

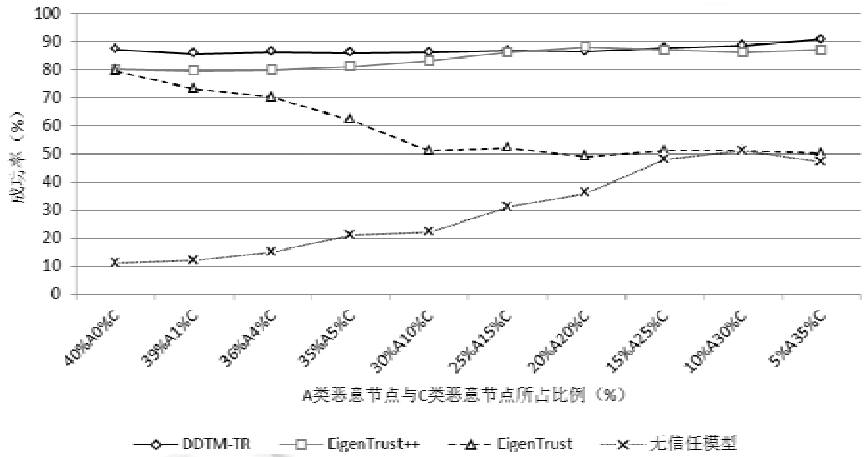



Fig.11 Success rate of interaction of each model with 40% malicious *A* and *C*

图 11 A 类恶意节点和 C 类恶意节点占 40%时,各模型的交互成功率

图 12 展示了当 *A* 类恶意节点占 5%、*C* 类恶意节点占 35%时,各个模型的处理恶意诋毁攻击的效率.

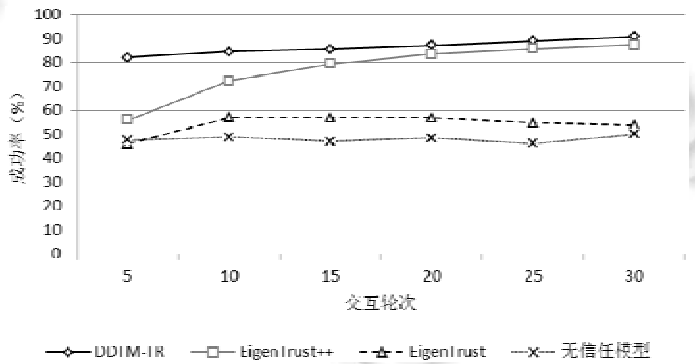



Fig.12 Success rate of interaction in different turns of each model with 5% malicious *A* and 35% malicious *C*

图 12 5%的 *A* 类恶意节点与 35%的 *C* 类恶意节点协同攻击时,各模型在不同轮次下的交互成功率

从图 11 与图 12 中可以看出:由于 *C* 类恶意节点对所有的 *D* 类正常节点进行诋毁,所以 *C* 类恶意节点与 *D* 类正常节点之间具有很大的差异,所以 EigenTrust++模型较第 5.3.4 节表现更好;EigenTrust 模型表现与图 10 差别不大,所以不再赘述;在 DDTM-TR 模型中,当 *C* 类恶意节点进行诋毁攻击时,反馈算法根据 *C* 类恶意节点的推荐与 *D* 类正常节点真实的满意度进行比较,从而对 *C* 类恶意节点的可靠度进行惩罚来抵御恶意诋毁攻击.

5.3.6 信任评价的持续改进实验

在本组实验中,将对比 90 轮交互中,不同阶段的 DDTM-TM 模型以验证反馈算法对 DDTM-TR 模型的作用.记第 0~30 轮交互的统计结果为 DDTM-TR30,记第 30~60 轮交互的统计结果为 DDTM-TR60,记第 60~90 轮交互的统计结果为 DDTM-TR90.每一阶段的交互统计开始时,上一阶段的成功次数记录都会清零.

仿真系统设置如下:*A* 类恶意节点比例从 10%增长到 70%,*B* 类恶意节点比例从 5%增长到 30%,每个 *B* 类恶

意节点会随机选择占 *A* 类恶意节点数量的 50%的 *A* 类恶意节点进行恶意协同.

实验结果如图 13 所示:DDTM-TR90 交互成功率最高,DDTM-TR60 其次,DDTM-TR30 最低.因为随着交互次数的增加,反馈算法起到了更大的作用.DDTM-TR 算法中,每一次交互之后,主体节点不仅获得直接交互经验,而且根据反馈算法修正其他节点的可靠度,使得节点可靠度更加接近真实值.在之后的信任判断时,由于节点可靠度得到改进修正,主体节点更加容易分别恶意推荐和正常推荐,从而作出正确的判断.

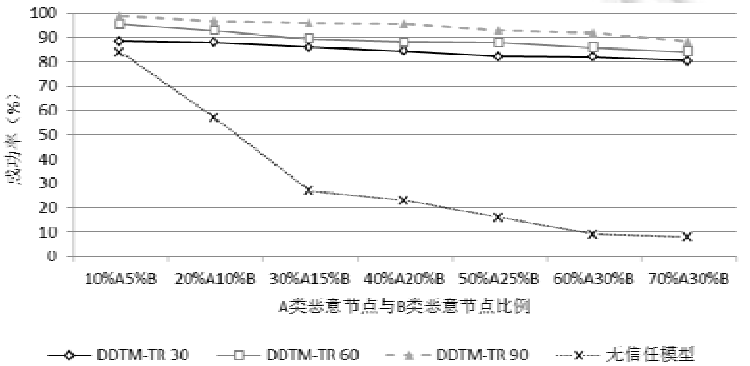



Fig.13 Effect of feedback algorithms in DDTM-TR model when the malicious *A* and malicious *B* exist

图 13 存在 *A* 类恶意节点和 *B* 类恶意节点时,交互反馈对 DDTM-TR 的影响

5.3.7 冒名攻击分析

冒名是指真实服务质量低的节点冒充其他服务质量高的节点,并企图:

(1) 降低其他用户节点对被冒名节点的信任度;

(2) 向其他节点提供恶意服务.

冒名问题最直接的解决办法是采用密码学密钥机制.DDTM-TR 模型要求在云计算系统中,通过密钥机制保证主体节点与真实目标节点的网络连接.另外,DDTM-TR 模型不同于全局信任模型,每个节点拥有自己的独立信任观点,并且不依赖于信任档案管理节点,因此,冒名节点进行冒名攻击还面临以下两点困难.

(1) 冒名节点难以获得大多数节点对某一节点的看法,所以在冒名目标的选择上存在困难;

(2) 即使冒名成功,冒名节点也只能影响冒名期间访问的用户节点的信任观点,而不能影响所有用户节点的信任观点.

## 6 结 语

本文提出了一种考虑了信任可靠度的分布式动态信任管理模型,采用信任度和可靠度作为信任评价的依据.信任度实现对目标节点服务质量的描述,可靠度实现对信任度可靠程度的描述.模型通过对可靠度的计算和分析,较为准确地过滤了恶意推荐或过时的直接经验.交互结束后,模型通过反馈算法对可靠度等数据进行调整和更新,以期进一步提高后续信任评价的可靠程度.实验分析表明:DDTM-TR 模型对典型的恶意攻击(恶意服务和冒名攻击)、恶意推荐(夸大和诋毁)均有较好的抵抗能力,而且在计算开销、节点存储复杂度方面也比以往的分布式模型更加适用于云计算环境.另外,DDTM-TR 还具备较强的自我调整能力,可以迅速提高信任评价的可靠程度.DDTM-TR 模型还存在着一定的局限性,比如:信任反馈算法有待继续优化,使模型对恶意节点有更高的敏感度,同时能正确处理正常节点的推荐失误.除此之外,恶意节点还可能采取更加复杂的攻击方式,需要进一步研究应对策略.

**References**:

[1] Singh A, Shrivastava M. Overview of security issues in cloud computing. Int'l Journal of Advanced Computer Research, 2012,2(1): 41−45.

[2] Melaye D, Demazeau Y. Bayesian dynamic trust model. In: Proc. of the Int'l Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2005), Berlin: Springer-Verlag, 2005. 480−489.

[3] Wang W, Zeng GS. Bayesian cognitive trust model based self-clustering algorithm for MANETs. Science China Information Sciences, 2010,53(3):494−505.

[4] Sun YX, Huang SH, Chen LJ, Xie L. Bayesian decision-making based recommendation trust revision model in ad hoc networks. Ruan Jian Xue Bao/Journal of Software, 2009,20(9):2574−2586. http://www.jos.org.cn/1000-9825/579.htm [doi: 10.3724/SP.J.1001.2009.00579]

[5] Li X, Ling L. Building trust in decentralized peer-to-peer electronic communities. In: Proc. of the Int'l Conf. on Electronic Commerce Research (ICECR-5). 2002. 1−15.

[6] Li HZ, Singhal M. Trust management in distributed systems. IEEE Computer Society, 2007,40(2):45−53.

[7] Kamvar SD, Schlosser MT, Garcia-Molina H. The eigentrust algorithm for reputation management in p2p networks. In: Proc. of the Int'l Conf. on World Wide Web. 2003. 640−651.

[8] Lu K, Wang JL, Li MC. An eigentrust dynamic evolutionary model in p2p file-sharing systems. In: Proc. of the Peer-to-Peer Networking and Applications. 2015. 1−14. [doi: 10.1007/s12083-015-0416-1]

[9] Nishikawa T, Fujita S. A reputation management scheme for peer-to-peer networks based on the eigentrust trust management algorithm. Journal of Information Processing, 2012,20(3):578−584.

[10] Fan XX, Liu L, Li MC, Su ZY. EigenTrust++: Attack resilient trust management. In: Proc. of the CollaborateCom 2012. 2012. 416−425.

[11] You J, Feng H, Sun YQ. Trust evaluation and service selection based on collaborative recommendation for cloud environment. Computer Science, 2016,43(5):140−145 (in Chinese with English abstract).

[12] Tian JF, Lu YZ, Li N. Trust chain management based on recommendation. Journal on Communications, 2011,32(10):1−9 (in Chinese with English abstract).

附中文参考文献:

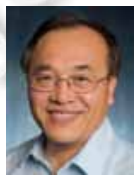
[11] 游静,冯辉,孙玉强.云环境下基于协同推荐的信任评估与服务选择.计算机科学,2016,43(5):140−145.

[12] 田俊峰,鲁玉臻,李宁.基于推荐的信任链管理模型.通信学报,2011,32(10):1−9.

游静(1975－),女,河北晋州人,博士,副教授,CCF 专业会员,主要研究领域为信任管理,软件抗衰,软件可靠性.

李千目(1979－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据处理,信息安全.

上官经伦(1992－),男,硕士,CCF 学生会员,主要研究领域为云计算,信任模型.



王印海(1965－),男,博士,教授,博士生导师,主要研究领域为智能交通系统.

徐守坤(1972－),男,博士,教授,CCF 高级会员,主要研究领域为数据库及信息系统,普适计算.