计算机科学
COMPUTER SCIENCE

基于节点影响力的区块链匿名交易追踪方法

李致远，徐丙磊，周颖仪

引用本文

李致远，徐丙磊，周颖仪. 基于节点影响力的区块链匿名交易追踪方法[J]. 计算机科学，2024，51(7): 422-429.

LI Zhiyuan, XU Binglei, ZHOU Yingyi. Blockchain Anonymous Transaction Tracking Method Based on Node Influence [J]. Computer Science, 2024, 51(7): 422-429.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230400177

基于节点影响力的区块链匿名交易追踪方法

李致远[1,2,3]　徐丙磊[1]　周颖仪[1]
1 江苏大学计算机科学与通信工程学院　江苏 镇江 212013
2 江苏省工业网络安全技术重点实验室　江苏 镇江 212013
3 江苏省泛在数据智能感知与分析应用工程研究中心　江苏 镇江 212013

摘　要　随着区块链技术的快速发展,借助虚拟货币进行非法交易的行为越来越普遍,且数量仍在快速增长。为打击该类犯罪行为,目前主要从网络分析技术和图数据挖掘等角度研究区块链交易数据,以进行区块链交易追踪。然而,现有的研究在有效性、普适性以及效率等方面存在不足,且无法对新注册地址进行有效追踪。针对上述问题,文中提出了一种基于节点影响力的账户余额模型区块链交易追踪方法 NITT,旨在追踪特定目标账户模型地址的主要资金流向。相比传统方法,该方法引入时间策略,降低了图数据规模,同时采用多重权重分配策略,筛选出了更有影响力的重要账户地址。在真实数据集上进行实验,结果表明,所提方法在有效性、普适性和效率等方面具有较大的优势。

关键词:区块链;匿名交易追踪;账户余额模型;节点影响力
中图分类号　TP301

Blockchain Anonymous Transaction Tracking Method Based on Node Influence

LI Zhiyuan[1,2,3],XU Binglei[1] and ZHOU Yingyi[1]
1 School of Computer Science and Communication Engineering,Jiangsu University,Zhenjiang,Jiangsu 212013,China
2 Jiangsu Provincial Key Laboratory of Industrial Network Security Technology,Zhenjiang,Jiangsu 212013,China
3 Jiangsu Province Ubiquitous Data Intelligent Perception and Analysis Application Engineering Research Center,Zhenjiang,Jiangsu 212013,China

Abstract　With the rapid development of blockchain technology,illegal transactions with the help of virtual currencies are becoming increasingly common and still growing rapidly. In order to combat such crimes,blockchain transaction data are currently studied mainly from the perspectives of network analysis technology and graph data mining for blockchain transaction tracking. However,the existing studies are deficient in terms of effectiveness,generalizability,and efficiency,and cannot effectively track newly registered addresses. To address the above issues,a node-influence-based blockchain transaction tracking method NITT for account balance models is proposed in the paper,aiming to track the main fund flow of a specific target account model address. Compared with traditional methods,the proposed method introduces a temporal strategy to reduce the graph data size. It also filters out more influential and important account addresses by using a multiple weight assignment strategy. Experimental results on real datasets show that the proposed method has greater advantages in terms of effectiveness,generalizability and efficiency.

Keywords　Blockchain,Anonymous transaction tracking,Account balance model,Node influence

1　引言

区块链[1]技术的核心思想是使用密码学方法将数据块连接起来,形成不可篡改的数据链,这使得区块链上的数据具有不可逆性、不可篡改性和去中心化等特点。

近年来,随着区块链技术的快速发展,其匿名性吸引了越来越多的非法交易,如金融诈骗、黑客盗币以及赃款清洗等[2-7]。著名区块链安全公司 Chainalysis[8] 2022 年的报告显示,2021 年加密货币相关业务的非法交易活动造成的损失已超过 140 亿美元并且仍处于极速增长中。

区块链账户可以被视作节点,每一条交易记录可以被视作边,因此区块链交易数据可以自然而然地建模为网络或图结构。也正因为此特点,目前学术界主要集中于从网络分析技术或图数据挖掘等角度对区块链交易数据进行研究分析。主流的研究方法是基于规则的启发式方法和基于污染/染色的追踪分析方法,但此类方法都是基于特定场景的研究,在有效性、普适性和效率等方面都存在一定的局限性。与此同时,目前的研究都主要集中于以比特币为代表的未消费交易输出模型(Unspent Transaction Output Model,UTXO)区块链,对以太坊为代表的账户余额模型区块链的相关研究较少。

综上所述,针对现有研究的缺陷和不足,本文提出了一种基于节点影响力的账户模型区块链交易追踪方法 NITT(Node Influence Transaction Tracing),可以针对特定目标账户模型地址追踪后续主要资金流向。本文的主要工作如下:

到稿日期:2023-04-25　返修日期:2023-09-23
基金项目:国家重点研发计划(2020YFB1005503);江苏省自然科学基金面上项目(BK20201415)
This work was supported by the National Key Research and Development Program of China(2020YFB1005503) and Jiangsu Provincial Natural Science Foundation Project(BK20201415).
通信作者:李致远(lizhiyuan@ujs.edu.cn)

1）针对大规模图数据算法运行效率的问题，在目标交易子图构建过程中引入了时间策略，根据设置的时间区间抽取相关交易数据构建子图，使图数据的规模急剧减小，算法运行效率与同类算法相比有较大的提升。

2）针对资金流向所涉及地址节点的影响力筛选问题，改进了 LeaderRank 算法，提出了引入多权重分配策略对 LeaderRank 值（简称 LR 值）进行按权分配的 ABW-LeaderRank 算法，筛选出更有影响力的重要账户地址。

3）在真实数据集上进行实验，在召回率、输出节点数、追踪深度、时间消耗等方面对追踪效果进行评估。通过实验，验证了该方法在追踪效果上的优势。最后通过权重平衡系数 β 和交易子图构建时间间隔，对召回率影响进行分析，得到了在合理时间消耗的算法最佳性能。

2 相关工作

目前，学术界主要从启发式方法、污染/染色分析方法以及图分析方法这 3 个方面对区块链交易溯源跟踪展开研究。Zheng 等[9]采用一种新的启发式聚类方法来获取比特币地址之间的关联关系，并采用改进的 Louvain 聚类算法进一步获取用户之间的关联关系。Möser 等[10]提出了一种基于污染分析的比特币追踪方法，但该方法会导致参与交易的“干净的钱”将被错误地归类为“脏钱”，且“脏钱”的数量将呈指数级增长。考虑到上述问题，Möser 等[10]同时提出了基于 haircut 的方法，考虑了脏输入的金额值，交易中的每个输出都包含脏输入和干净输入的比例。Anderson[11]提出了先进先出（First-In First-Out，FIFO）染色方法，它不同于 haircut 染色，FIFO 染色可以做到污染不扩散。另外 FIFO 染色使用资产库存管理中类似的概念来对无法明确识别的污染交易进行排序，仍不能从根本上解决精度问题。Tovanich 等[12]提出了一种基于污点分析的动态网络方法来提取污点流，用来分类来源参与者和描述不同的资金流动模式。另外，广度优先搜索（Breadth-First Search，BFS）算法[13]及其变体[14]很早便已被应用于区块链交易追踪的相关研究中。文献[14]中的算法被应用于 2015 年 Ashley Madison 勒索骗局的分析，并进一步估计了该骗局中可疑地址控制的金额。Yousaf 等[15]通过时间和金额相近的交易来识别跨链的资金流动，并总结了 3 种跨链交易模式，以及这些行为的识别方法。Haslhofer 等[16]提出了名为 GraphSense 的比特币交易数据分析框架，用来跟踪比特币的流动。Zhu 等[17]提出了一种基于以太坊的可追溯性方法，该方法使用图分析来跟踪攻击者。

上述研究除了 Yousaf 等[15]提出的交叉 sledger 交易跟踪方法外，其他方法仅适用于 UTXO 模型区块链的交易追踪问题，但并不适用于以以太坊为典型代表的账户余额模型区块链。Chen 等[18]根据节点和边的不同语义，提出了 3 种用于以太坊交易数据分析的网络，即资金流图、合约创建图和合约调用图。除了资金流图和合约调用图之外，Zhao 等[19]还引入了另外两种网络建模方法，即账户创建图和账户投票图。Lin 等[20]提出将以太坊交易记录建模为时间加权多向图，并从网络的角度[21]结合不同交易因素研究以太坊交易跟踪问题。Motamed 等[22]提出了月交易图和累计月交易图的概念，分别表示一个月的交易数据和累计交易数据。Chen 等[23]通过图形分析，描述了以太坊 ERC20 令牌生态系统上的令牌创建者、令牌持有人和令牌转移活动。Wu 等[24]提出了一个用于账户的区块链可扩展交易跟踪工具 TRacer，并开发了改进的个性化 PageRank 方法，用于推断账户之间的相关性。近年来，对区块链交易跟踪的方法变得更加多样化，如基于语义的方法[25]、神经网络方法[26-27]、基于深度学习的方法[28]等。

3 区块链交易追踪方法

本文提出了基于节点影响力的账户模型区块链交易追踪方法 NITT，其模型框架结构如图 1 所示，主要分为 3 个执行模块，分别为交易子图抽取、节点影响力计算、资金追踪图构建。

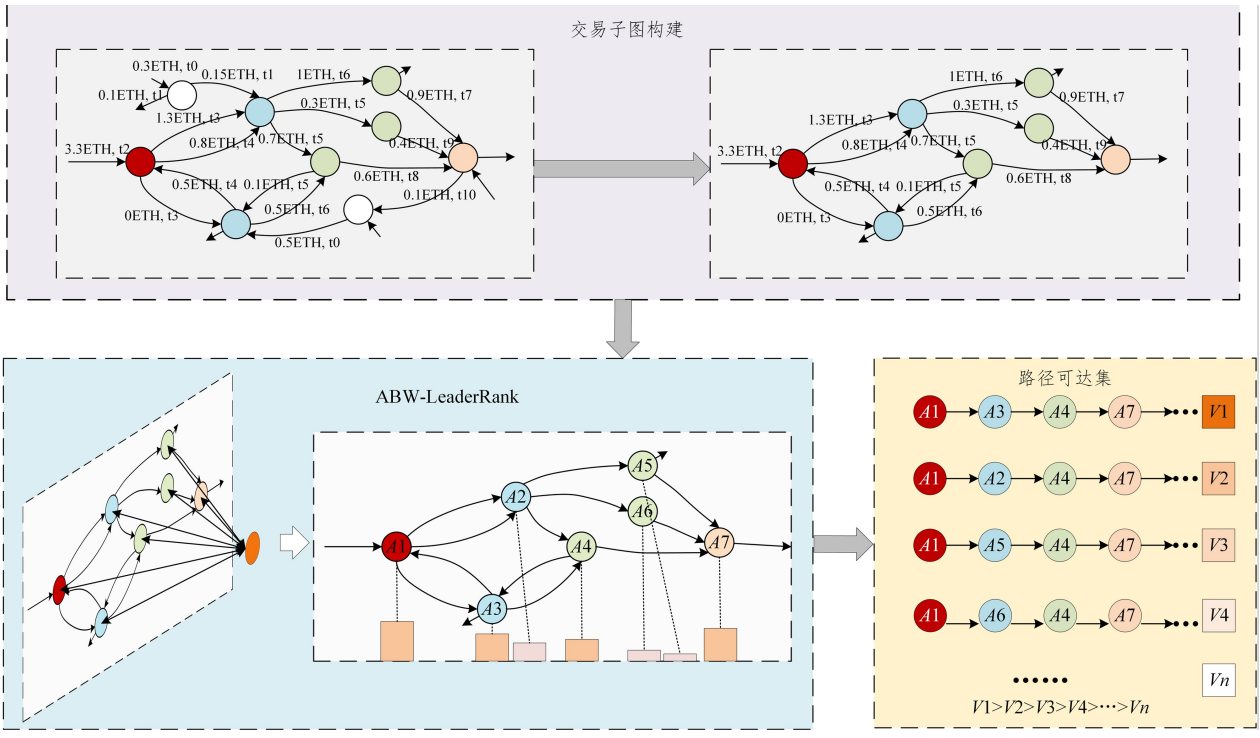

图 1 NITT 模型框架

Fig. 1 NITT model framework

3.1 本文模型的流程

1)根据设置的初始地址 *address* 执行交易子图 $G(V,E)$ 构建过程,抽取时间戳位于起始交易时间 t_{start} 及终止交易时间 t_{end} 内的交易数据,同时构建各地址间的交易次数字典 T_{ij}、各地址间的交易金额字典 A_{ij} 以及各交易时间戳字典 TS_{ij}。

2)ABW-LeaderRank 算法初始化。为流程 1 中构建的交易子图中的每个地址节点分配一个初始权重,$LR_i^{(0)}=1$,其中 $i=0,1,\cdots,n$。

3)更新权重:对于每个地址节点 i,按照式(1)更新其权重。

$$LR_i^{(t)}=\sum_{j\in\mathcal{N}_{in}(i)}\frac{w_{ij}\cdot LR_i^{(t-1)}}{\sum_{k\in\mathcal{N}_{out}(j)}w_{jk}} \tag{1}$$

其中,$\mathcal{N}_{in}(i)$表示指向地址节点 i 的节点集合,$\mathcal{N}_{out}(j)$表示地址节点 j 指向的节点集合,w_{ij} 表示边(i,j)的权重。

4)收敛条件:重复步骤 3),直到满足收敛条件(见式(2)),或达到预定的最大迭代次数。

$$\sum_{i=1}^{n}|LR_i^{(t)}-LR_i^{(t-1)}|<\epsilon \tag{2}$$

5)分配背景节点的 LR 值:将背景节点 g 的 LR 值根据入度与出度关系分配给其他节点。

6)根据地址节点的 LR 值进行排序,以确定以太坊交易图中地址节点的重要性和影响力。

7)从初始地址出发,沿出度方向搜索所有可达路径。停止搜索条件为搜索到出度为零的节点或达到指定跳数。在构建可达集的同时,生成一个存储对应各条可达路径的 LR 值和的集合。

8)根据各条路径的 LR 值和给出前 N 条资金去向,构成交易子图并进行可视化展示。

3.2 交易子图构建

由于以太坊全量交易数据过于庞大,且在交易追踪过程中绝大部分的交易数据均属于无效数据,在区块链交易追踪问题中,基本思想是从起始地址开始,沿着出度方向追踪资金流向。但账户模型区块链因其地址的复用性往往记录了多笔历史交易数据,但只有在特定时间之后的数据可以为本文的交易追踪提供分析依据。因此,为了提高算法的运行效率以及降低时间复杂度,本文提出了根据特定时间片段的交易子图抽取方法,如图 2 所示,从全量数据中抽取用于构建以太坊交易子图的特定交易数据,并分别记录相关交易数据中各个地址间的交易次数、交易金额以及对应的时间戳。图 2 中,假设以 t_2 为起始时间戳、t_9 为截止时间戳来追踪交易流向,通过遍历相关地址节点的历史交易数据并剔除其他交易,可以构建图 2 中下半部分图所示的交易子图。

交易子图构建过程主要通过算法 1 实现。

算法 1 交易子图构建算法

输入:address,t_{start},t_{end},G(V,E),T_{ij},A_{ij},TS_{ij},a_k

输出:G(V,E),T_{ij},A_{ij},TS_{ij}

```
1. 初始化 Tij,Aij,TSij,V,E
2. /* 对每个地址提取交易 k,并将它们按时间排序 */
3. for each address
4.     search(k,address)
5.     tk←sort k by time
6.     /* 对符合要求的交易进行相应操作 */
7.     for each tk
8.       if tstart<= tk<= tend
9.         V. append(i,j)
10.        E. append(eij)
11.        Tij=Tij+1
12.        Aij=Aij+ak
13.        TSij. append(tk)
14.      endif
15.    endfor
16. endfor
```

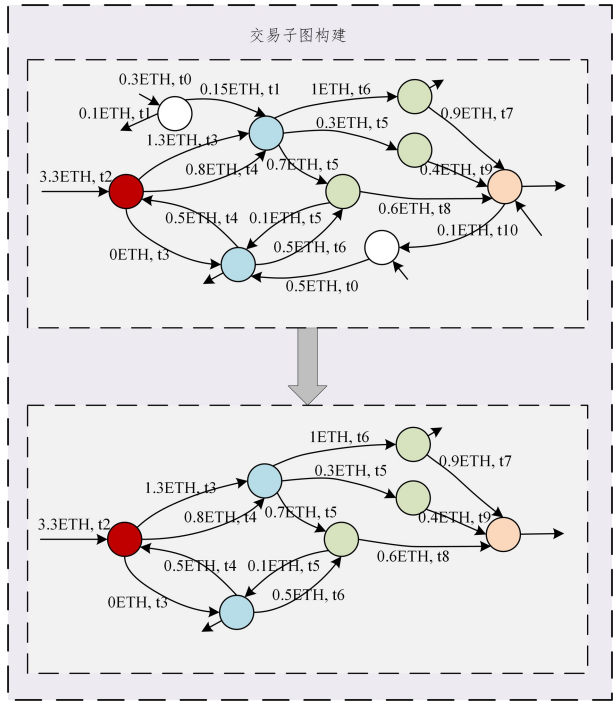

图 2 交易子图构建

Fig. 2 Transaction subgraph construction

算法 1 中,输入参数 *address* 表示初始地址,t_{start} 表示起始时间戳,t_{end} 表示终止时间戳,$G(V,E)$ 表示交易子图,T_{ij} 表示交易次数字典,A_{ij} 表示交易金额字典,TS_{ij} 表示交易时间戳字典,算法的输出为参数更新后的值。首先初始化各个参数;算法 1 中的第 2-5 行表示从本地数据库查询地址 address 的所有交易数据,将查询到的交易数据按照时间顺序排序;算法 1 中的第 7-15 行针对每个交易 t_k,检查交易时间 t_k 是否位于 V 和 V 之间,如果是,则执行以下操作:1)将交易发送方地址 V 和接收方地址 V 添加到顶点集 V 中(如果尚未添加);2)在边集 E 中添加一条边 e_{ij},表示从地址 a_k 到地址 a_k 的交易关系;3)更新交易次数字典:a_k;4)更新交易金额字典,a_k,其中 a_k 为交易 TS_{ij} 的金额;5)更新交易时间戳字典,将时间戳 TS_{ij} 添加到 TS_{ij} 的列表中。

通过上述算法,可以构建交易子图 $G(V,E)$,以及用于描述交易次数的字典 T_{ij}、交易金额的字典 A_{ij} 以及交易时间戳字典 TS_{ij},用于运行后续的 ABW-LeaderRank 算法。

3.3 ABW-LeaderRank 算法

通过 3.2 节的交易子图构建过程,已经将所涉及的地址节点特定时间段内的交易数据建模为带权交易子图。本节

提出了一种基于 ABW-LeaderRank 的算法，用于在以太坊交易追踪问题中计算所涉及地址的节点影响力，进而可以通过地址节点在追踪过程中的影响力构建交易追踪子图。

在 ABW-LeaderRank 算法中，第 1 步是初始化阶段。在这个阶段，首先需要为上述交易子图添加一个背景节点，并且假设这个背景节点与交易图中所有地址节点双向连接，使得该交易网络变为一个强连通网络，从而保证算法收敛性以及解决悬挂节点的问题。接着，需要为图中的所有地址节点分配一个初始的 LR 值，通常设置为 1。这意味着每个地址节点 i 的初始 LR 值为：

$$LR_i^{(0)}=1,\forall i=0,1,\cdots,n \tag{3}$$

其中，$i=0$ 表示背景节点 g，$i=1,2,\cdots,n$ 表示网络中的其他地址节点。

第 2 步是进行 LR 值的迭代更新。在这个阶段，根据每个地址节点的出度和邻居节点的权重来更新其 LR 值。节点 i 的 LR 值更新如下：

$$LR_i^{(t+1)}=\sum_{j\in N_i}\frac{w_{ij}\cdot LR_i^{(t)}}{\sum\limits_{k\in N_j}w_{jk}} \tag{4}$$

其中，$LR_i^{(t)}$ 表示第 t 次迭代后地址节点 i 的 LR 值，N_i 表示地址节点 i 的邻居地址节点集合，w_{ij} 表示地址节点 j 到地址节点 i 的边的权重，t 表示当前迭代次数。

在以太坊资金追踪问题中，将权重 w_{ij} 设置为地址间的交易次数和交易金额的组合。为了实现这一目标，使用式(5)来计算权重：

$$w_{ij}=\beta\cdot T_{ij}+(1-\beta)\cdot A_{ij} \tag{5}$$

其中，T_{ij} 表示地址 i 和地址 j 之间的交易次数，A_{ij} 表示地址 i 和地址 j 之间的总交易金额。β 是介于 0 和 1 之间的参数，用于平衡交易次数和金额在权重计算中的相对重要性。当 β 接近 1 时，交易次数对权重的贡献更大；当 β 接近 0 时，交易金额对权重的贡献更大。

在计算 w_{ij} 之前，需要对 T_{ij} 和 A_{ij} 进行归一化处理，以使它们的数值范围相同，计算式如下：

$$\hat{T}_{ij}=\frac{T_{ij}-\min(T)}{\max(T)-\min(T)} \tag{6}$$

$$\hat{A}_{ij}=\frac{A_{ij}-\min(A)}{\max(A)-\min(A)} \tag{7}$$

其中，$\min(T)$和 $\max(T)$分别表示图中交易次数的最小值和最大值；$\min(A)$和 $\max(A)$分别表示图中交易金额的最小值和最大值。将归一化后的 $\hat{T}_{ij}$ 和 $\hat{A}_{ij}$ 代入权重计算公式，得到：

$$w_{ij}=\beta\cdot\hat{T}_{ij}+(1-\beta)\cdot\hat{A}_{ij} \tag{8}$$

使用该权重计算式，可以在 ABW-LeaderRank 算法中更好地捕捉交易次数和金额对资金流向的影响，并且可以根据实际情况进行适当的调整。

第 3 步是检查收敛条件。收敛条件的目的是确保算法在地址节点的权重稳定下来之后停止迭代。收敛条件可以通过式(9)表示：

$$\sum_{i=1}^{n}|LR_i^{(t)}-LR_i^{(t-1)}|<\epsilon \tag{9}$$

其中，n 表示地址节点的数量，ϵ是一个正数，用于控制算法的收敛精度。当所有地址节点在当前迭代与上一次迭代之间的 LR 值变化之和小于ϵ时，则认为算法已经收敛，停止迭代。ϵ 的设置取决于具体应用场景对算法精度以及时间的要求。较小的ϵ值将使结果更加精确，但需要更多的迭代次数，消耗更多的时间。较大的ϵ值将使得算法较快地收敛，但可能降低结果的精度。通常，ϵ值可以设置为 $10^{-3}\sim10^{-6}$，具体取决于实际应用场景和对算法精度的需求，也可以通过尝试不同的ϵ值，观察其对结果和计算时间的影响，从而找到最佳设置。此外，为了避免算法无休止地运行，本文还设置一个最大迭代次数 ω，一旦达到最大迭代次数，即使收敛条件尚未满足，算法也将终止。最大迭代次数 ω 的设置取决于计算资源和时间限制，通常设置为 100～1 000 次，也可以根据实际需求调整 ω。

第 4 步是分配背景节点的 LR 值。背景节点是为了保证算法的收敛性而人为添加的节点，在算法收敛后需要将该节点的 LR 值分配给其他节点，否则整体交易子图的 LR 值将失衡。与此同时，在账户模型区块链交易追踪问题中，对于入度远大于出度或出度为零的地址应给予更大的关注，因为这意味着资金进入了冷钱包不再流动。因此，区别于 LeaderRank 算法，ABW-LeaderRank 算法更倾向于将背景节点的 LR 值分配给入度远大于出度的节点以及悬挂节点，对于入度-出度为负数的节点不分配 LR 值，具体的分配规则如下。

首先，计算入度-出度为正数的节点之和。

$$S=\sum_{i=1}^{n}\max(k_i^{\text{in}}-k_i^{\text{out}},0) \tag{10}$$

其中，$i=1,2,\cdots,n$，k_i^{in} 表示节点 i 的入度，k_i^{out} 表示节点 i 的出度。

然后，对于每个节点 i，在仅考虑入度-出度为正数的情况下，计算其入度-出度的比例 P_i。

$$P_i=\frac{\max(k_i^{\text{in}}-k_i^{\text{out}},0)}{S} \tag{11}$$

最后，根据比例 P_i 分配背景节点的 LR 值。

$$LR_i=LR_i^{(t)}+LR_0^{(t)}\cdot P_i \tag{12}$$

通过这种方式，可以根据地址节点入度和出度的关系为每个节点分配适当的 LR 值。特别地，对于入度远大于出度的节点，这种方法将分配更多的 LR 值，以强调这些节点在资金流向中的影响力。根据上文的描述，我们得到 ABW-LeaderRank 算法，具体如算法 2 所示。

算法 2 ABW-LeaderRank 算法

输入：G(V,E)，A，n，T_{ij}，A_{ij}，ò，ω，k_i^{in}，k_i^{out}，β

输出：LR_i

1. /* 初始化阶段 */
2. for i=1:n
3. $\quad LR_i^{(0)}=1$
4. endfor
5. $LR_0^{(0)}=1$
6. /* LR 值迭代更新，并检查收敛条件 */
7. for i=0:n,j=0:n
8. $\quad \hat{A}_{ij}$←compute $\hat{A}_{ij}$ according to Eq(6)
9. $\quad \hat{T}_{ij}$←compute $\hat{T}_{ij}$ according to Eq(7)
10. $\quad w_{ij}$←compute w_{ij} according to Eq(8)
11. endfor
12. do

```
13.  LR_i^(t) ← computeLR_i^(t) according to Eq(4)
14. until satisfy Eq(2) ort=ω
15. /* 分配背景节点的 LR 值 */
16. S← compute S according to Eq(9)
17. for i=1:n
18.  P_i ← compute P_i according toEq(10)
19.  LR_i ← compute LR_i according to Eq(11)
20. endfor
```

在算法 2 中，输入参数 $G(V,E)$是由算法 1 更新后的交易子图，$\boldsymbol{A}$ 表示图 G 的邻接矩阵，n 表示图 G 的节点个数，T_{ij} 表示交易次数，A_{ij} 表示交易金额，ϵ为一个正数，用于控制算法的收敛精度，ω 表示最大迭代次数，k_i^{in} 表示节点 i 的入度，k_i^{out} 表示节点 i 的出度。输出参数为 LR_i，为每个节点的 LR 值。

算法 2 中，第 1－5 行为每个节点赋予一个 LR 值，初始值设置为 1，也可以根据需求调整背景节点的值。第 7－11 行为计算边的权重，由归一化后的交易次数和交易金额进行组合，引入平衡因子 β来平衡交易次数和交易金额的重要性大小。第 12－14 行为对每个节点利用式(4)迭代更新其 LR 值，第 14 行为收敛条件，当达到式(2)的要求时或者迭代次数达到上限 ω 时，停止迭代。最后，分配背景节点的 LR 值，如第 15－20 行，首先计算所有入度减出度为正数的节点之和 S，然后计算每个节点的入度与出度之差占 S 的比例 P_i，根据 P_i 分配背景节点的 LR 值，得到除背景节点外所有节点的 LR 值。

最后根据计算得到的 LR 值对地址节点进行排序，排序后的地址节点列表反映了每个地址在以太坊交易网络中的影响力，越靠前的节点越重要，由此找到前 N 个重要节点(见算法 3)。这有助于识别资金流向的关键地址节点，例如冷钱包地址等。

算法 3 地址节点 LR 值排序算法

```
1. create a empty list
2. for each v_i
3.   List. append(i, LR_i)
4. endfor
5. sort(LR_i)
6. select(top_N)
```

在算法 3 中，首先创建一个空列表 *list*，然后将网络中的每个节点 i 及其对应的 LeaderRank 值 LR_i 作为一个元组(i, LR_i) 添加到列表 *list* 中，将列表 *list* 按照 LR 值降序排序，最后选取前 N 个重要节点。

3.4 交易追踪子图构建

在获得以太坊交易网络所有地址节点的影响力后，可以沿着初始地址节点的出度方向搜索所有可达路径 p，构成一个路径可达集 P 以及路径对应的地址节点的 LR 值合集 *SUM*。在此过程中，可以使用深度优先搜索(Depth-First Search，DFS)或广度优先搜索(BFS)实现。搜索停止的条件可以是搜索到出度为零的地址或达到指定跳数。两种算法的区别在于访问顺序和搜索策略。

1)广度优先搜索(BFS)。BFS 是一种按层次遍历图的方法。从初始节点开始，先访问所有与初始节点直接相邻的节点，然后访问这些邻居节点的邻居节点，依次执行。在遍历过程中，BFS 会将访问过的节点添加到一个队列中，以保证按层次顺序访问。BFS 算法如算法 4 所示。

算法 4 BFS 算法

```
1. create a empty listall_paths for storing all paths and LR_i
2. let Q be queue
3. Q. append((initial_node,[],0))
4. definedepth_limit / * 限制搜索深度 * /
5. while Q is not empty
6.   node,path,lr_sum=Q. pop(0)
7.   path. append(node)
8.   lr_sum += LR[node] / * LR 值求和 * /
9.   if k_i^out=0 or depth_limit=t / * 如果当前节点的出度为零或已达
     到搜索深度限制 t * /
10.     all_paths. append((path[:],lr_sum))
11.  else
12.     queue. append((neighbor,path[:],lr_sum))
13.  endif
14. end while
```

2)深度优先搜索(DFS)。DFS 是一种先沿着一个分支尽可能深入搜索，直到无法继续为止，然后回溯并沿着其他分支继续搜索的方法。在遍历过程中，DFS 使用递归或栈来实现。DFS 算法如算法 5 所示。

算法 5 DFS 算法

```
1. create a empty listall_paths for storing all paths and LR_i
2. create a empty listpath for storing current paths
3. define lr_sum a variable for storing Σ_{i=0}^{n} LR_i
4. initialize Σ_{i=0}^{n} LR_i=0
5. define a functiondfs(node,path,lr_sum,depth_limit)
6.    path. append(node)
7.    lr_sum += LR[node] / * LR 值求和 * /
8.    if k_i^out=0 or depth_limit=t / * 如果当前节点的出度为零或已
      达到搜索深度限制 t * /
9.       all_paths. append((path[:],lr_sum))
10.   else
11.      dfs(neighbor,path,lr_sum,depth_limit)
12.   endif
13. endfunction
14. path. pop() / * 将当前节点从路径中移除 * /
15. lr_sum -= LR[node]/ * LR 值减去 * /
16. dfs(initial_node,path,lr_sum,depth_limit)
```

使用 BFS 和 DFS 的区别主要在于遍历顺序、搜索策略、内存占用以及计算时间。如果需要找到最短路径或节省计算时间，可以选择 BFS。如果需要发现所有可能的路径并且内存有限，可以选择 DFS。在实际应用中，可以根据具体情况来选择算法达到不同的追踪要求。

4 性能评估

4.1 数据集

本节通过整合网络公开的区块链交易追踪数据集，构建了一个专用于账户余额模型区块链(以太坊)的交易追踪数据集，涵盖了近 3 年包含黑客盗币、集资诈骗以及非法金融交易

在内的 8 个交易追踪案例。这些案例均从 peckschild，Chainalysis 等著名区块链安全公司公开报告中获得，并通过 Google BigQuery 平台的 crypto_ ethereum 交易数据集以及以太坊浏览器 Etherscan 获取到相关地址的原始交易数据。该数据集的一些统计数据如表 1 所列。

表 1 交易追踪数据集统计数据

Table 1 Transaction tracking dataset statistics

字段	描述	数量/个
源地址	交易追踪案例源地址	8
目标地址	在相应交易追踪案例中已知的重要地址节点	2358
总地址量	所有交易追踪案例中涉及的总地址量	108451
总交易量	所有交易追踪案例中涉及的总交易量	5412573

4.2 实验评估

为了评估所提出的 NITT 方法在实际交易追踪实验中的效果，本文使用了以下度量，即召回率、输出节点数、追踪深度、时间消耗，详细信息如下。

1）召回率（Recall）：该指标评估某方法可以追踪到多少个目标地址节点，该指标的定义为 $Recall=\frac{V_t}{V_n}$，其中 V_t 代表追踪到的目标地址的数量，V_n 是案例中所有目标节点的数量。

2）输出节点数（Output Nodes）：该指标评估某方法输出的交易追踪结果中涉及的重要地址节点数，输出图中较少的地址节点可以保证更高效的人工验证或审计效率。

3）追踪深度（Tracing Depth）：该指标评估某方法可以从源地址节点遍历交易子图的深度，即检测从源地址节点最多多少跳到达邻居地址节点。

4）时间消耗（Time Consumption）：该指标评估某方法整体运行所消耗的时间。

本节实验在 Ubuntu 18.1 64 位操作系统上进行，其中 CPU 为 32 核心 AMD EPYC 75F3，GPU 为 NVIDIA GeForce RTX 3090，8T 三星 M.2 SSD 固态硬盘，416TB 7200RPM SATA，Python 版本为 3.8。为了与同类型区块链交易追踪算法进行比较，本文使用了 Poison[5]，TRacer[16]，BFS[19] 算法进行对比实验。对于实验参数，为防止运行时间过长，设置 BFS 算法和 Poison 算法最大追踪深度为 4；设置 TRacer 算法 $\varphi=10^{-3}$，$\beta=0.7$；设置 NITT 方法 $t_{end}-t_{start}=48$ h，$\beta=0.4$，$\epsilon=10^{-4}$，$\omega=100$。所有算法在 8 个交易追踪案例重复进行 30 次实验，取 30 次实验结果的均值作为各算法的平均检测结果，如表 2 所列。

表 2 以太坊交易追踪各算法平均检测结果对比

Table 2 Comparison of average detection results of various Ethereum transaction tracking algorithms

Models	Recall	Output nodes/pcs	Tracing depth/hop	Time consumption/s
BFS	0.7869	35260	4	352
Poison	0.7523	31850	4	541
Tracer	0.8835	3480	42	213
NITT	0.9238	2560	56	176

由表 2 可以看到，NITT 方法的召回率最高，达到 0.9238，其次是 TRacer 算法，达到了 0.8835。BFS 和 Poison 算法的召回率相对较低，分别为 0.7869 和 0.7523。从输出地址节点数来看，NITT 和 TRacer 算法的输出节点数较少，分别为 2560 个和 3480 个，可以保证更高的人工验证或审计效率。BFS 和 Poison 算法的输出节点数相对较多，分别为 35260 和 31850。同时 NITT 和 TRacer 的追踪深度较高，分别为 56 跳和 42 跳。BFS 和 Poison 算法的追踪深度相对较浅，都为 4 跳。最后在时间消耗上，NITT 的时间消耗最少，为 176 s，TRacer 算法的时间消耗稍微多一些，为 213 s，BFS 算法为 352 s，而 Poison 算法的时间消耗为 541 s。

为了对比各算法的运行效率，本文计算了各算法的平均召回率增长曲线，如图 3 所示。从图中可以看到，随着时间的增长，NITT 方法最快达到收敛状态，平均召回率为 0.9238，该算法前期召回率较低是因其前期需要先进行交易子图的重构；TRacer 算法效率仅次于 NITT 方法，平均消耗 213 s，达到了 0.8835 的召回率；BFS 和 Poison 算法达到收敛状态所消耗的时间较长，分别为 352 s 和 541 s，平均召回率分别为 0.7869 和 0.7523。由此可见 Poison 算法的效果最差，消耗时间最长且召回率最低。

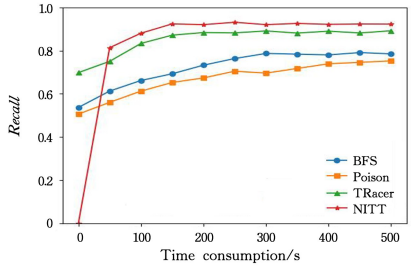

图 3 各算法的平均召回率

Fig. 3 Average recall rate of each algorithm

4.3 性能分析

为了分析本文所提出的 NITT 方法在不同参数设置下对交易追踪效果的影响，本节分别从交易子图构建时间间隔 $t_{end}-t_{start}$、权重 w_{ij} 平衡系数 β 进行分析。

由图 4 可以看到，随着时间间隔的增加，NITT 方法的召回率稳步增长，从追踪起始时间 t_{start} 开始的最初 12 h 内召回率增长率最高，在 12 h 时召回率已达到 0.6512，12 h 到 48 h 期间召回率从 0.6512 稳步增长至 0.9201，此后召回率增长较为缓慢。

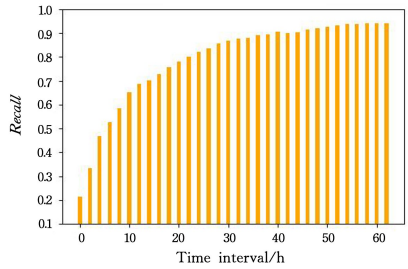

图 4 交易子图构建时间间隔对召回率的影响

Fig. 4 Impact of transaction sub graph construction time interval on recall rate

图 5 给出了权重 w_{ij} 平衡系数 β 对算法召回率的影响曲线。由上文可知，β 是一个介于 0 和 1 之间的参数，用于平衡交易次数和交易金额在权重计算中的相对影响力。当 β 接近 1 时，交易次数对权重的贡献更大；当 β 接近 0 时，交易金额对权重的贡献更大。从图中可以看到，随着 β 的增长召回率从最低的 0.7812 开始增长，在 β 为 0.4 时召回率最高，平均

为 0.9218，此后召回率开始下降，在 β 为 1 时召回率降低至 0.7515。

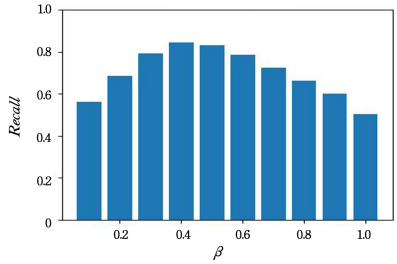

图 5 权重平衡系数 β 对召回率的影响

Fig. 5 Impact of weight balance coefficient β on recall rate

结束语 本文主要对账户余额模型区块链的交易追踪问题进行了研究，提出了一种基于节点影响力的账户模型区块链交易追踪方法 NITT。首先，针对以太坊数据规模庞大的问题，提出了基于时间片段的交易子图抽取策略，仅抽取特定时间段内的交易数据构成用于交易追踪的子图，极大地提高了算法效率。然后，提出了 ABW-LeaderRank 算法，用于计算追踪过程中所涉及地址节点的影响力。接着，根据地址的影响力确定最有可能的资金流向路径，构成交易追踪子图，可为交易追踪提供辅助参考。最后，通过实验评估和性能分析，验证了该方法的有效性和可行性。

本文仅考虑了交易次数和金额对交易追踪的影响，在未来，我们将更多的因素（如资金流动方向等）考虑在内，并结合深度学习、神经网络等知识，探索更精确、更具有普适性的交易追踪方法。

参 考 文 献

[1] TREIBLMAIER H, GARAUS M. Using blockchain to signal quality in the food supply chain: The impact on consumer purchase intentions and the moderating effect of brand familiarity[J]. International Journal of Information Management, 2023, 68: 102514.

[2] ZHOU X, YANG W, TIAN X. Detecting Phishing Accounts on Ethereum Based on Transaction Records and EGAT[J]. Electronics, 2023, 12(4): 993.

[3] XIA Y, LIU J, WU J. Phishing detection on ethereum via attributed ego-graph embedding[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2022, 69(5): 2538-2542.

[4] GU Z, LIN D, WU J. On-chain analysis-based detection of abnormal transaction amount on cryptocurrency exchanges[J]. Physica A: Statistical Mechanics and its Applications, 2022, 604: 127799.

[5] LIAO Z, ZHENG Z, CHEN X, et al. SmartDagger: a bytecode-based static analysis approach for detecting cross-contract vulnerability[C]// Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis. 2022: 752-764.

[6] ZHENG P, ZHENG Z, LUO X. Park: Accelerating smart contract vulnerability detection via parallel-fork symbolic execution [C]// Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis. 2022: 740-751.

[7] Chainalysis. The 2022 Crypto Crime Report [EB/OL]. https://www.chainalysis.com/blog/2024-crypto-crime-report-introduction/.

[8] WU J, LIU J, ZHAO Y, et al. Analysis of cryptocurrency transactions from a network perspective: An overview[J]. Journal of Network and Computer Applications, 2021, 190: 103139.

[9] ZHENG B, ZHU L, SHEN M, et al. Malicious bitcoin transaction tracing using incidence relation clustering[C]// Mobile Networks and Management: 9th International Conference, MONAMI 2017. 2018: 313-323.

[10] MÖSER M, BÖHME R, BREUKER D. Towards risk scoring of Bitcoin transactions[C]// Financial Cryptography and Data Security: FC 2014 Workshops, BITCOIN and WAHC 2014. Christ Church, Barbados, 2014: 16-32.

[11] ANDERSON R. Making Bitcoin Legal(Transcript of Discussion)[C]// Security Protocols XXVI: 26th International Workshop. Cambridge, UK, 2018: 254-265.

[12] TOVANICH N, CAZABET R. Pattern Analysis of Money Flows in the Bitcoin Blockchain[C]// International Conference on Complex Networks and Their Applications. Cham: Springer International Publishing, 2022: 443-455.

[13] ZHAO C, GUAN Y. A graph-based investigation of bitcoin transactions[C]// Advances in Digital Forensics XI: 11th IFIP WG 11.9 International Conference. Orlando, FL, USA, 2015: 79-95.

[14] PHETSOUVANH S, OGGIER F, DATTA A. Egret: Extortion graph exploration techniques in the bitcoin network[C]// 2018 IEEE International Conference on Data Mining Workshops(ICDMW). 2018: 244-251.

[15] YOUSAF H, KAPPOS G, MEIKLEJOHN S. Tracing transactions across cryptocurrency ledgers[C]// 28th USENIX Security Symposium(USENIX Security 19). 2019: 837-850.

[16] HASLHOFER B, KARL R, FILTZ E. O Bitcoin Where Art Thou? Insight into Large-Scale Transaction Graphs[C]// SEMANTiCS(Posters, Demos, SuCCESS). 2016.

[17] ZHU H, NIU W, LIAO X, et al. Attacker Traceability on Ethereum through Graph Analysis[J/OL]. Security and Communication Networks, 2022. https://doi.org/10.1155/2022/3448950.

[18] CHEN T, LI Z, ZHU Y, et al. Understanding ethereum via graph analysis[J]. ACM Transactions on Internet Technology (TOIT), 2020, 20(2): 1-32.

[19] ZHAO Y, LIU J, HAN Q, et al. Exploring EOSIO via graph characterization[C]// Blockchain and Trustworthy Systems: Second International Conference(BlockSys 2020). Dali, China, 2020: 475-488.

[20] LIN D, WU J, YUAN Q, et al. Modeling and understanding ethereum transaction records via a complex network approach [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2020, 67(11): 2737-2741.

[21] SONG W, ZHANG W, WANG J, et al. Blockchain data analysis from the perspective of complex networks: Overview[J]. Tsinghua Science and Technology, 2022, 28(1): 176-206.

[22] MOTAMED A P, BAHRAK B. Quantitative analysis of cryptocurrencies transaction graph[J]. Applied Network Science, 2019, 4(1): 1-21.

[23] CHEN W，ZHANG T，CHEN Z，et al. Traveling the token world：A graph analysis of ethereum erc20 token ecosystem [C]// Proceedings of The Web Conference 2020. 2020：1411-1421.

[24] WU Z，LIU J，WU J，et al. TRacer：Scalable Graph-based Transaction Tracing for Account-based Blockchain Trading Systems [J]. IEEE Transactions on Information Forensics and Security，2022，18：2609-2621.

[25] WU Z，LIU J，WU J，et al. Know Your Transactions：Real-time and Generic Transaction Semantic Representation on Blockchain & Web3 Ecosystem [C]// Proceedings of the ACM Web Conference 2023. 2023：1918-1927.

[26] LIU J，ZHENG J，WU J，et al. FA-GNN：Filter and Augment Graph Neural Networks for Account Classification in Ethereum [J]. IEEE Transactions on Network Science and Engineering，2022，9：2579-2588.

[27] HUANG T，LIN D，WU J. Ethereum Account Classification Based on Graph Convolutional Network [J]. IEEE Transactions on Circuits and Systems II：Express Briefs，2022，69(5)：2528-2532.

[28] GU Z，LIN D，WU J. On-chain analysis-based detection of abnormal transaction amount on cryptocurrency exchanges [J]. Physica A：Statistical Mechanics and its Applications，2022，604：127799.

LI Zhiyuan，born in 1981，Ph. D，postdoctor，associate professor，is a senior member of CCF(No. 11049S). His main research interests include mobile social networks，Internet of Things，and software defined networks and cybersecurity.

（责任编辑：喻藜）

CCF 高性能计算博士学位论文激励计划提名征集通知！

2024 年，CCF 高性能计算专业委员会推出“CCF 高性能计算博士学位论文激励计划”，该激励计划旨在进一步推动高性能计算领域高水平创新人才培养工作。每年评选出不超过 3 篇，在高性能计算领域做出杰出创新研究工作的博士学位论文，对论文完成人给予表彰，年度最多不超过 3 人。在 CCF 全国高性能计算学术年会上颁发。即日起正式开通征集，现将 2024 年度推荐有关事项通知如下：

1. 参评候选论文同时满足以下条件：

(1)2022 年 1 月 1 日至 2024 年 6 月 10 日期间在中国内地高校或研究机构获得博士学位；

(2)未获得过 CCF 或其他一级学会的博士学位论文激励计划或提名。

2. 提名条件：

该激励计划评选采用推荐－评审制，由单位推荐或 CCF 高性能计算专委执委联名推荐候选博士学位论文，由专家评审小组组织评审，评选过程具体包括推荐、资格审查、初评、终评 4 个阶段。

(1)单位推荐：每个具有计算机类学科博士授予权的单位可推荐参评博士学位论文不超过两篇，其他单位可推荐参评博士学位论文不超过一篇。

(2)专家推荐：要求不少于 3 名 CCF 高性能计算专委会执行委员联名推荐，其中一位推荐人为主推荐人。每位推荐人可推荐不超过两名候选人。

3. 参评推荐材料：

推荐材料应于 2024 年 7 月 10 日 17：00 前提交至 CCF 高性能计算博士学位论文激励计划秘书处(lixidai@ict. ac. cn)，邮件主题请注明【CCF 高性能计算博士学位论文激励计划】。

(1)采用单位推荐形式，提交单位推荐表；

(2)采用专家推荐形式，提交专家推荐表；

(3)博士学位论文电子版；

(4)其他相关证明材料等。

4. 评选流程及时间安排(拟定)：

(1)受理材料：2024 年 7 月 10 日 24：00

(2)资格审查：2024 年 7 月 15 日 24：00

(3)初评：2024 年 8 月 10 日前，由专家评审小组组织初评，从中选出不超过 10 篇论文入围提名；

(4)初评公示：2024 年 8 月 12 日－16 日

(5)终评：2024 年 9 月 10 日前，由专家评审小组组织终评，最终评选出不超过 3 篇论文入选 CCF 高性能计算博士学位论文激励计划，每篇获得税前 1 万元，并获得中国计算机学会高性能计算专业委员会“年度高性能计算博士学位论文激励计划”；进入终评的未入选论文，获得中国计算机学会高性能计算专业委员会“年度高性能计算博士学位论文激励计划提名”；

(6)颁发：2024 年 9 月 24 日－26 日，在武汉 CCF 全国高性能计算学术年会上颁发。

据 CCF 微信公众号