计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330298
61(12): 3005−3026, 2024

Web 3.0 数字经济风险感知技术综述

贾金萍[1,2] 肖诗涵[1,2] 钱 堃[1,2] 杨艳琴[1,3] 张 召[1,2]

[1](区块链数据管理教育部工程研究中心(华东师范大学) 上海 200062)

[2](华东师范大学数据科学与工程学院 上海 200062)

[3](华东师范大学软件工程学院 上海 200062)

(jpjia22@stu.ecnu.edu.cn)

Survey of Risk Perception Technology for Web 3.0 Digital Economic

Jia Jinping[1,2], Xiao Shihan[1,2], Qian Kun[1,2], Yang Yanqin[1,3], and Zhang Zhao[1,2]

[1](*Engineering Research Center of Blockchain Data Management* (*East China Normal University*), *Ministry of Education*, *Shanghai* 200062)

[2](*School of Data Science and Engineering*, *East China Normal University*, *Shanghai* 200062)

[3](*Software Engineering Institute*, *East China Normal University*, *Shanghai* 200062)

Abstract The Web 3.0 digital economic system takes the blockchain platform as its infrastructure, and revolves around the digital assets such as cryptocurrencies, NFTs, digital collectibles, and decentralized applications (DApps) like decentralized finance (DeFi) and gaming finance (GameFi) to conduct various socio-economic activities. Smart contracts are the core of DApps on the public blockchains and the public permissioned blockchains, such as Ethereum, Solana, EOSIO, Findora, Antchain, ChainMaker, et al. Smart contracts can be deployed by any individual or organization, and are visible and accessible to all blockchain users. This openness brings new opportunities for economic development while also harboring numerous financial risks. We analyze potential risks for Web 3.0 digital economic by focusing on smart contract and summarize the current research on risk perception technology from three aspects: encoding, functionality, and application of smart contracts. We first introduce the research challenges, security vulnerability types, and four categories of vulnerability detection methods in smart contract vulnerability detection technology. Next, we analyze common types of smart contract scams and summarize existing scam recognition techniques based on different classifications of training data. Subsequently, we introduce the current state of technology for detecting four types of illicit transactions behaviors based on blockchain transaction records. Lastly, by analyzing the limitations of existing work, we envision the future research directions.

Key words Web 3.0; blockchain; smart contract; risk perception technology; digital economy

摘 要 Web 3.0 数字经济体系以区块链平台为基础设施,围绕加密货币、NFT、数字藏品等数字资产和去中心化金融(DeFi)、游戏金融(GameFi)等去中心化应用(DApp)开展各项社会经济活动.在公有链和开放联盟链下,作为 DApp 内核的智能合约可以由任何个人或组织予以部署,并对全体用户可见及可调用.这种开放性给经济发展带来了新的机遇,同时也蕴含了许多金融风险.以智能合约为中心分析了 Web 3.0 数字经济潜在的风险,并从智能合约的编码、功能、应用 3 个层面总结了风险感知技术的研究现状.首先介绍了智能合约漏洞检测技术的研究挑战、安全漏洞类型和 4 类漏洞检测方法;其次分析了常见的智能

收稿日期: 2023-04-06; **修回日期**: 2023-08-15

基金项目: 国家自然科学基金项目(61972152);上海市优秀学术/技术带头人项目(23XD1401100)

This work was supported by the National Natural Science Foundation of China (61972152) and the Program of Shanghai Leading Talent Program of Eastern Talent Plan (23XD1401100).

通信作者: 张召(zhzhang@dase.ecnu.edu.cn)

合约骗局类型,并根据训练数据的不同分类总结了现有的智能合约骗局识别技术;接着介绍了基于区块链交易记录对4种非法交易行为进行检测的技术现状;最后结合对现有工作局限性的分析,展望了未来的研究方向.

关键词 Web 3.0;区块链;智能合约;风险感知技术;数字经济

中图法分类号 TP391

互联网自出现以来经历了从 Web 1.0 到 Web 3.0 的发展变革. 在 Web 1.0 阶段,用户可以浏览雅虎、搜狐等门户网站中的文本、图片和视频,对互联网数据只具有读的权限. 在 Web 2.0 阶段,用户不仅可以获取信息,而且可以基于微博、微信、YouTube 等社交媒体平台在线生成内容并与其他用户交互协作,也就是说,用户对互联网数据具有了读和写的权限. 然而, Web 2.0 下用户的数字身份由平台方认证和管理,数据的所有权掌握在平台方手中,其价值的分配也由平台方制定的协议决定,因而存在平台间可移植性差、隐私泄露、数据篡改、资源垄断等问题. 随着区块链技术的出现与发展,互联网开始进入 Web 3.0 阶段. 文献 [1] 提到 Web 3.0 一词最早由 HTTP 协议的发明者 Tim Berners Lee 于 1998 年提出. 在之后的十几年间, Web 3.0 得到了学术界和工业界的积极探讨,提出了包括语义网(semantic Web)[2]、沉浸式互联网(immersive Web)[3]在内的诸多设想. 然而,所有设计都停留在实验层面,并没有得到大规模的应用. 2014 年,以太坊联合创始人 Wood[4] 将 Web 3.0 重新定义为区块链技术支撑下的一种“无需信任的交互系统”,该系统利用区块链技术以去中心化的方式保证了 Web 3.0 用户的数字身份和个人数据具有不可篡改、高可用和可移植等特性,在一定程度上实现了用户对其数字身份和个人数据的所有权与支配权.

区块链技术起源于 2008 年由 Nakamoto[5](中本聪)发明的比特币,它通过哈希算法和链式结构保证了数据的不可篡改性,通过 P2P 网络和共识机制实现了去中心化的数据管理. 2013 年, Buterin[6] 提出了一个可编程的区块链平台以太坊,使区块链平台可以支撑更广泛的应用. 近年来,为了服务于更多应用领域,国内外也出现了不少新的高性能区块链平台,比如国外的 Solana, EOSIO, Findora 和国内的蚂蚁链、长安链等. 随着底层基础平台的发展,基于区块链平台开发的去中心化应用(decentralized application, DApp)开始大量涌现,其应用领域也从数字货币扩展到金融、游戏、保险、物联网等各个领域. 相比于中心化的应用与服务, DApp 中用户的身份数据和数字资产由区块链系统维护,因而技术上不受单一平台的管控. 对于用户而言, DApp 避免了平台方对其身份权限的控制和对个人数据的滥用;对于开发者而言,区块链系统将所有 DApp 的数据流打通,实现了安全的信息共享和价值流通;对于市场而言, DApp 避免了龙头企业对生产要素的垄断,有助于形成以消费者为中心的健康发展局面. 因此,稳步推进 Web 3.0 的到来是非常有意义的.

数字经济是继农业经济、工业经济后的新经济形态, Web 3.0 下的数字经济主要表现为以加密数字货币、NFT、数字藏品为代表的数字资产和用于开展各类社会经济活动的去中心化应用. 目前,国外及我国香港地区的 Web 3.0 应用主要运行在以太坊等公链系统中,我国内地的 Web 3.0 应用则主要运行在开放联盟链[7] 中. 基于公链系统形成了庞大的虚拟货币交易市场,以 MakerDAO[8] 为代表的 DeFi 项目正在探索利用智能合约技术构建低成本无国界的数字金融体系. 与公链系统相比,开放联盟链中节点的加入需要经过授权,且使用人民币作为唯一结算建立收益分配机制. 但开放联盟链在用户的注册使用、项目部署、智能合约访问权限等方面与公链系统是一致的,均具有较大的开放性. 目前,开放联盟链上的 DApp 开发主要集中在区块链游戏、供应链金融、公益溯源、版权合同、票据民生等领域,并逐渐发展出了具有一定规模的数字藏品交易市场.

无论是基于公有链还是开放联盟链, Web 3.0 数字经济存在发展机遇的同时也蕴含着风险. 一方面, DApp、区块链交易所和用户钱包遭黑客攻击的事件频发,用户因此损失的财产不计其数. 例如, 2016 年以太坊去中心化自治组织 DAO 的众筹项目因存在智能合约漏洞被黑客窃取了 360 万以太币[9];同年,香港比特币交易所 Bitfinex 遭黑客攻击,致使近 12 万比特币被盗[10]. 另一方面, DApp 的商业模式存在很大的不确定性,区块链平台易被用于洗钱、诈骗、传销、赌博等违法犯罪活动. 例如,混币器 Tornado. Cash 自创建以来已被用于清洗价值超过 70 亿美元

的虚拟货币[11]；以“加密猫”为代表的链游通过特殊的游戏规则实现对游戏币或道具的炒作升值，当不再有后来者接盘时资金盘崩溃，游戏内剩余的道具资产持有人则沦为受害者；去中心化智能合约平台 Forsage 募集散户投资者资金超 3 亿美元，其运作方式却是典型的庞氏骗局，即通过新投资者的资产来支付早期投资者的收益，严重损害了投资者的利益[12]. 因此，构建 Web 3.0 下数字经济监管体系，完善金融风险感知技术，对促进 Web 3.0 数字经济健康发展至关重要.

Web 3.0 数字经济风险广义上包含底层的系统风险和上层的应用风险，前者主要是指区块链系统在交易的共识、执行和存储过程中潜在的风险，后者则主要包括去中心化应用及其数字资产交易所面临的风险. 本文探讨的 Web 3.0 数字经济风险是指狭义的应用层面的风险. 由于去中心化应用是 Web 3.0 数字经济区别于传统数字经济的典型特征，因此本文主要对去中心化应用项目从开发到运行的各个环节进行风险分析和风险感知. 去中心化应用的底层逻辑是部署在区块链平台上的一组智能合约[13]，数字资产所有权的转移通常也通过智能合约的调用来完成. 可以说，智能合约是 Web 3.0 数字经济体系的核心，因此本文的 Web 3.0 数字经济风险感知技术最终落脚到智能合约的编码、功能和调用 3 个层面. 如图 1 所示，从编码层面上看，智能合约潜在的漏洞存在被黑客攻击的风险；从功能层面上看，智能合约可能有实施诈骗、赌博等违法犯罪活动的风险；从调用层面上看，智能合约的调用过程可能存在非法交易行为. 因此，围绕 Web 3.0 的数字资产和去中心化应用，本文探讨的数字经济风险感知技术主要包括智能合约漏洞检测、智能合约骗局识别、非法交易行为检测 3 类.

学术界对此开展了大量研究工作，目前也存在一些与上述技术相关的文献综述. Chen 等人[14] 梳理了 Web 3.0 下数字经济的概念与分类，并总结了现阶段的技术生态和发展挑战. Huang 等人[15] 在区块链技术最新研究进展方面综述了现有的区块链网络分析模型. Wu 等人[16] 以元宇宙为背景分析了 Web 3.0 中可能出现的金融犯罪形态及其分类框架，并分别从学术研究和相关政策的角度概述了 Web 3.0 元宇宙金融监管方案. 文献 [14–16] 在整体框架上对 Web 3.0、区块链和元宇宙相关技术的研究现状进行了总结，但对 Web 3.0 数字经济风险感知技术只是概括性地做了介绍. 陈伟利等人[17] 关注区块链数据分析技术的研究进展，总结了区块链数据分析的研究问题和技术现状. 魏松杰等人[18] 对区块链系统在应用中存在的安全问题并进行了分类总结. Tolmach 等人[19] 对区块链智能合约形式化规范及验证技术进行了综述. 钱鹏等人[20] 对智能合约安全漏洞检测技术进行了综述. 然而，这些综述只针对某一项核心技术进行了深入的调研与总结，并没有从 Web 3.0 数字经济的视角对其风险感知技术的研究现状进行介绍.

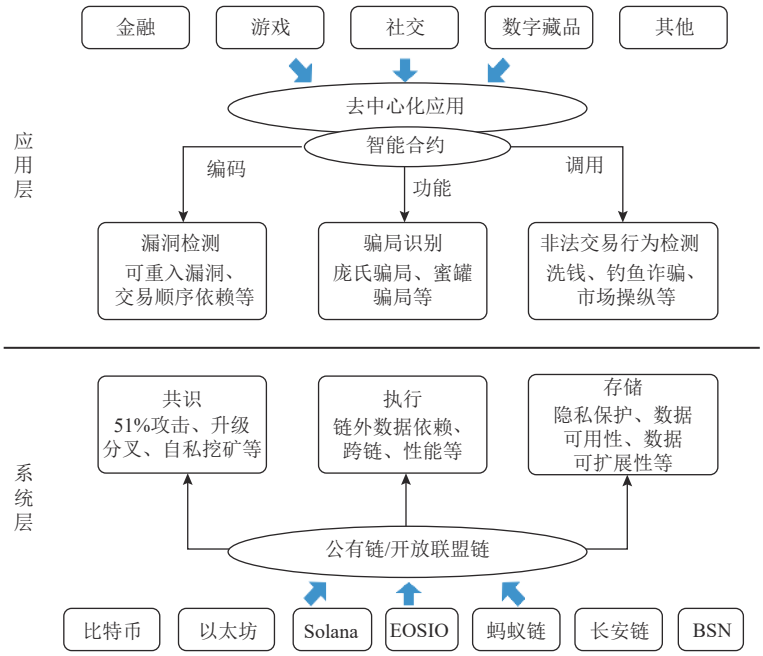

Fig. 1 Overview of the risk system in Web3.0 digital economy

图 1 Web3.0 数字经济风险体系总览

1 智能合约漏洞检测技术

智能合约本质上是运行在以太坊等区块链系统中的软件程序,一般使用图灵完备的高级语言编写.智能合约被部署到区块链平台后,其代码将被区块链虚拟机,例如以太坊虚拟机 EVM 或 EOSIO 系统的虚拟机 WASM 编译为字节码存储.此时,智能合约被识别为一个 160 b 的十六进制地址,任何用户都可以通过向该地址发送交易来调用智能合约.智能合约因具有不可篡改性和透明性成为以太坊等区块链系统的基石,在去中心化的金融交易、数字认证和供应链管理等问题上获得了广泛的应用.

然而,由于智能合约编程语言和编程工具成熟度不够,以及合约开发人员考虑不周,导致编写出的智能合约可能会存在安全漏洞.而智能合约通常会持有一定价值的数字资产,因而非常容易招致黑客攻击,从而造成巨大的经济损失.更令人担忧的是,智能合约被部署至区块链后是不可篡改的,其在运行过程中产生的安全问题因而难以被修复.因此,在智能合约部署前做好安全漏洞的检测工作,避免因存在智能合约漏洞而带来的经济损失,是一件非常有必要的任务.

与面向传统高级程序语言(如 C++, Java 等)的漏洞检测技术相比,智能合约的代码分析与漏洞检测显然具有更大的挑战.首先,智能合约通常由特定的语言(例如 Solidity)开发,具有非标准的语义行为和面向事务的机制,使得其形式化验证非常复杂;其次,现有区块链平台上的智能合约大部分并不开源,只能基于编译后的字节码进行分析,然而由于现有的智能合约字节码通常是基于堆栈的低级代码,使得对其执行逻辑的跟踪非常困难;最后,智能合约在区块链上运行时使用的数据并不是静态已知的,这对分析技术的抽象能力要求极高.

针对上述挑战,学术界已经提出了不少智能合约漏洞检测方法,本文将它们归纳为符号执行、深度学习、形式化规范与验证、混合方法 4 类,将分别在 2.2~2.5 节中详细介绍.

1.1 安全漏洞分类

通过对现有工作的分析,本文总结常见的智能合约安全漏洞类型有 9 个:

1)可重入性(reentrancy)漏洞.当一个合约(调用者)调用另一个合约(被调者)时,调用者处于等待调用完成状态,被调者可以重复回调调用者的函数.可重入性漏洞通常存在于转账函数内,发送方调用函数 *call.value*(*amount*) 触发接收方的回退函数实现转账.若调用者的余额变量在调用外部函数之后才被修改,则给了被调者窃取其资产的机会.具体做法为:被调者在回退函数中回调该转账函数,由于此时账户余额还未发生修改,因而能顺利通过余额检查再次实现转账,于是被调者可以利用多次回调来盗取调用者中的所有金额.The DAO 事件中攻击者就是利用了可重入性漏洞窃取了项目中的大量资金.

2)区块信息依赖(block information dependency)漏洞.区块链系统提供了一些接口用于智能合约获取当前区块的信息,例如区块数量(*block.number*)、区块 gas 限制(*block.gaslimit*)、区块时间戳(*block.timestamp*)等.如果一些合约使用这些区块信息作为触发条件来执行一些关键操作,则给恶意矿工提供了操纵合约执行的机会,因为这些区块信息完全由矿工决定.其中,智能合约使用区块时间戳作为触发条件而引发的漏洞被称为时间戳依赖(timestamp dependency, TD)漏洞,目前存在较多针对 TD 漏洞的检测工作.

3)交易顺序依赖(transaction-ordering dependency)漏洞.用户可以发送交易来调用智能合约,但是交易的发送时间和执行时间并没有严格的关联,矿工可以根据自己的需要调整交易的打包顺序.如果智能合约的业务逻辑以交易顺序作为决策条件,或者交易执行顺序会影响合约的执行结果,则恶意矿工可以通过操纵交易打包顺序来对合约进行攻击.

4)未处理异常(mishandled exceptions)漏洞.区块链中有多种合约调用方式,在某些调用方式下,如果被调合约发生异常(例如没有足够的 gas 或堆栈溢出等),则被调合约中止运行并回滚已执行的结果.但在通过指令发起的调用中,被调合约中的异常可能不会显式地传播回调用合约,而是以返回值的形式返回给调用合约,此时调用者需要显式地验证这些返回值以确定是否正确执行.智能合约中调用方式和异常处理方式的不一致会导致许多没有正确处理的异常存在.

5)算术(arithmetic)漏洞.智能合约在处理数值计算时可能发生错误,未经检查的数值可能会被恶意利用.例如恶意用户利用整数溢出(integer overflow)窃取合约资产.编程语言对整数类型的存储空间有特定的长度限制,一旦整数操作结果超过了这个长度则发生溢出.在使用智能合约进行转账时,假设在完成发送者(*from*)到接收者(*to*)资金(*value*)转移的同时,合约账户(*sender*)会收取一笔手续费(*fee*),则

即使转账函数对 *balance*[*from*] <*value*+*fee* 进行了检查，但当(*value*+*fee*)发生溢出时(即 *value*+*fee*=0)，就算 *value*+*fee*>*balance*[*from*] 也可以通过检查并完成转账. 这种情况下，超过账户 *from* 余额的资金被分配给账户 *to* 和 *sender*，相当于攻击者凭空产生了资金.

6)无限循环(infinite loop)漏洞. 这是智能合约中的常见漏洞，其原因主要包括，函数内部存在没有退出条件或退出条件无法达到的循环体，以及函数和回退函数之间或函数自身的循环调用. 攻击者可以利用无限循环漏洞使智能合约拒绝服务(deny-of-service, DoS)合约中存在的资金则会被永久锁定(locked ether)而无法取出.

7)短地址攻击(short address attack)漏洞. 智能合约通常对交易中地址类型的数据有特定的长度要求，例如必须为 20 B，当小于此长度时会自动用后续数据或 0 补齐. 当攻击者恶意打包小长度地址数据时，若合约缺乏对输入数据长度的检查，则会导致交易数据解析错误.

8)坏随机性(bad randomness)漏洞. 若智能合约的执行条件为一些随机数，并且随机数的生成以一些可预测的值(例如时间)作为种子，则使得随机性不再随机. 攻击者在理解了合约逻辑后，便可恶意利用这些可预测的随机值非法获利.

9)不受限操作(unrestricted action)漏洞. 泛指一切不受约束的操作，包括未受保护的所有权(unprotected ownership)、不合理的访问控制(improper access control)、未经检查的调用(unchecked-calls)，以及危险委托调用(dangerous delegate call)等. 若普通用户可以设置或更改合约所有权，便可以执行合约的一切操作，包括对自毁(self-destruct)函数的调用. 委托调用(delegate call)接口允许智能合约在自己的上下文中执行外部合约的代码，外部合约代码可以修改原合约的状态，缺乏对代码调用的检查给恶意代码注入提供了可乘之机.

上述 9 种漏洞类型基本涵盖了目前在以以太坊为主的区块链系统中发现的大部分智能合约漏洞. 此外，还存在一些针对其他区块链的智能合约漏洞. Cui 等人[21]针对 Solana 区块链的 Rust 智能合约研究了 6 类漏洞的检测，包括缺少对签名者、拥有者和密钥的检查(missing signer/owner/key check)、整数溢出/下溢(integer overflow/underflow)、跨合约调用(cross-program invocation)、数值精度错误(numerical precision error)、碰撞种子规范化(bump seed canonicalization)、类型混淆(type confusion). He 等人[22]对 EOSIO 区块链(一个很受 DApp 开发者关注的区块链平台)中由 C++编写的智能合约进行漏洞检测，主要检测 Fake EOS 代币、Fake Receipt、Rollback、Missing Permission Check 这 4 类漏洞. 这 4 类漏洞并不常见，针对它们的检测工作也相对较少，本文涉及的智能合约漏洞检测技术主要与前面 9 类安全漏洞相关.

1.2 符号执行方法

符号执行是一种经典的程序分析方法，也是最早用于区块链智能合约漏洞检测的方法[23]. 符号执行将所有输入视为符号变量以探索所有可能的执行路径，这些符号变量可以覆盖程序的整个输入域. 直观地说，对于函数 $f(x)$，符号执行考虑的不是一个具体的执行轨迹，例如 $f(10)$，而是一个符号输入 φ，即 $f(\varphi)$，其中 φ 为整个输入域. 当执行到一个分支时，例如 if 语句，执行过程将分叉以探索多条可能的路径，每一个分叉都是程序可能的一种执行状态. 为了降低状态空间，符号执行器将程序的当前状态以及路径条件(例如 $\varphi\leqslant 3$)编码为一阶逻辑公式，并使用 SMT(satisfiability modulo theory)求解器来检查程序路径是否可行，从而避免进一步探索不可能的路径.

符号执行的结果是将智能合约代码构建成为符号执行树或控制流图，它代表了所有可能的执行路径. 基于符号执行树或控制流图，可以通过模式匹配来检测程序漏洞. 具体而言，如果存在满足某些特定约束的智能合约执行路径，就可以认为智能合约存在某种特定的漏洞. 如图 2 所示，若智能合约的控制流图中 *call*. *value* (*amount*) 指令执行后，外部合约的回退函数回调了当前函数，则该环形执行路径将绕过 *call*. *value* (*amount*) 指令之后的余额修改指令 *balance*−= *amount*，从而说明当前合约存在可重入性漏洞. 此外，比较不同执行路径交错的结果，可以验证是否存在交易顺序依赖漏洞.

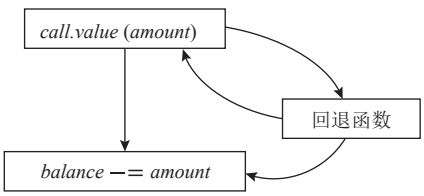

Fig. 2 Illustration of the execution path of reentrancy vulnerability

图 2 可重入漏洞执行路径示意图

Oyente[23]是采用符号执行方法进行漏洞检测的代表性工具，它由 Luu 等人[23]于 2016 年提出，并在当时的 19 266 份以太坊智能合约中标记了 8 833 份错误合约. Oyente 将 EVM 指令集映射到约束条件，由

以太坊全局状态提供合约变量的初始化值，消息调用的数据被视为输入符号，通过符号执行构造合约字节码的控制流图，可以检测交易顺序依赖、时间戳依赖、未处理异常等漏洞. 随后，Krupp 等人[24]从操作码的角度给出了合约漏洞的通用定义，他们针对 CALL, SELFDESTRUCT, CALLCODE, DELEGATECALL 这 4 个具有较高概率引发风险的指令建立了漏洞模式. Bose 等人[25]提出了一个“Explore-Refine”的漏洞检测框架：Explore 阶段借助存储依赖图（storage dependency graph）表示合约执行对存储变量的影响，减少了需要推理的相关指令的数量，提高了漏洞检测效率；Refine 阶段对找出的漏洞进一步验证，提高了漏洞检测的准确率.

与上述子图匹配的方式不同，Ye 等人[26]分别从漏洞合约和安全合约中提取结构化程序特征作为漏洞签名，提出基于漏洞签名的检测规则和检测方法. 此外，Liao 等人[27]关注跨合约漏洞检测的有效性和效率提升问题，提出一个面向字节码的跨合约漏洞检测框架 SmartDagger，通过从字节码中恢复属性信息，有选择地分析功能子集和重用控制流数据流图等策略提高性能. Zheng 等人[28]针对单线程符号执行效率低的问题提出并行漏洞检测算法，关注那些涉及全局变量的漏洞检测.

最后我们发现，还有一些对漏洞进行修复的工作. Rodler 等人[29]为以太坊提供了一个字节码重写引擎，可以对易受整数溢出/下溢和访问控制错误影响的智能合约进行自动强化. Nguyen 等人[30]基于符号执行识别 4 种常见漏洞的执行路径，并采用指令替换的方式实现合约代码的重写，从而完成漏洞的修复.

1.3 深度学习方法

基于符号执行的方法虽然在检测智能合约漏洞方面是有效的，但这种依赖于专家定义的规则进行智能合约漏洞检测的方法存在 2 个问题：1）人工定义的漏洞模式相对简单，难以覆盖一些复杂的漏洞模式，因而会导致较高的误判率；2）专家定义的规则的可伸缩性有限，随着以太坊智能合约数量不断增加，专家不可能针对所有智能合约来设计精确的模式.

研究人员提出了各种基于深度学习的智能合约漏洞自动检测方法，以避免依赖专家定义的模式和降低构建新类型漏洞检查器的人工成本. 该方法大致分为 2 类：一类是将深度学习模型直接用于智能合约源代码或字节码，通过人工标注合约漏洞类型，对模型进行监督学习训练；通过计算与不同漏洞类型的智能合约的相似度，以确定合约的漏洞类型. 另一类则是将深度学习模型用于智能合约源代码转换后的合约图，如图 3 所示，包括合约图生成阶段，从智能合约源代码中提取控制流和数据流，将功能代码转换为代码语义图以构建合约图；图归一化阶段，通过合并节点的方式来规范化图，保留与关键信息相关的控制流和数据流，将代码语义图进行归一化；图模型训练和检测阶段，采用不同深度学习方法对智能合约漏洞检测模型进行训练，将训练好的模型用于漏洞检测.

Rossini 等人[31]基于字节码采用不同深度学习模型对智能合约进行检测，比较了 LSTM, ResNet 1D CNN, 2D ResNet-18 CNN, 2D Inception v3 CNN 这 4 种不同类型的神经网络，发现 ResNet 1D CNN 在合约分类方面表现最优. Gao 等人[32]通过计算与存在漏洞合约的相似度来确定检测的合约是否存在漏洞，基于代码嵌入和相似性检查技术，通过比较以太坊中现有智能合约代码嵌入向量与已知漏洞的相似度，能够高效地识别代码克隆及相关错误. Qian 等人[33]提出基于注意力机制的 BLSTM-ATT 模型以精确检测可重入错误，通过智能合约的片段表示来捕获基本

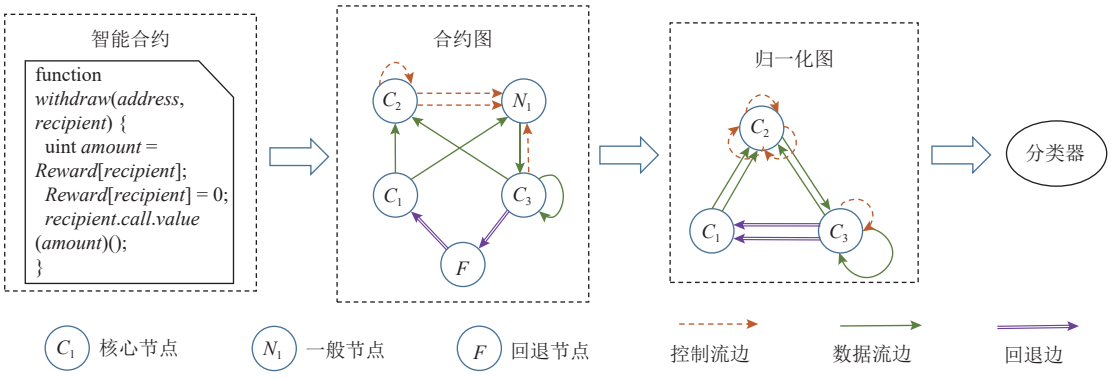

Fig. 3 Smart contract vulnerability classification model based on contract graph

图 3 基于合约图的智能合约漏洞分类模型

的语义信息和控制流依赖性. 将基于匹配的检测方法直接应用于智能合约会受到 2 个问题阻碍, 即编译器的快速演变导致字节码生成的多样性, 以及同构业务逻辑容易造成噪声代码干扰. 为此, Huang 等人[34] 提出了面向智能合约字节码的规范化和切片技术增强字节码匹配. 首先, 对字节码进行预处理并构建控制流图, 基于控制流图确定切片标准, 并在字节码执行基础上提取对应的切片; 然后, 对切片进行归一化处理, 利用图嵌入网络对面向切片的控制流图进行矢量化; 最后, 通过成对比较相似性, 检验表示目标合约切片的向量是否可以匹配任何易受攻击切片的向量.

MANDO-GURU[35] 基于学习将控制流图和调用图结合嵌入到注意力神经图网络中, 生成节点级和图形级的智能合约嵌入, 并训练分类器来识别智能合约中粗粒度合约级和细粒度代码行级的各种类型的漏洞. Zhuang 等人[36] 将图神经网络用于智能合约中的漏洞检测, 可以检测出重入漏洞、时间戳依赖漏洞和无线循环漏洞. 依据函数中不同程序元素的重要性提取了 3 类节点, 为突出重要节点的核心作用, 通过合并节点对图进行规范化, 将图卷积网络(graph convolutional network, GCN)扩展为无度 GCN(DR-GCN), 并提出新的时态消息传播网络, 这 2 种网络将智能合约函数归一化图作为输入以及将该函数是否存在某种类型漏洞的标签作为输出. Wu 等人[37] 注意到文献 [36] 方法采用的图形表示对于程序来说是复杂的, 不仅考虑 3 种类型的节点, 还涉及复杂边缘信息, 使得模型无法在不同合约中推广, 为此他们提出了一个预训练模型来自动检测智能合约的漏洞, 在模型训练过程中提供了大量的智能合约源代码及其控制流图以及真实标签, 再使用训练好的模型对新输入的源代码和控制流图产生对应的漏洞标签. 首先, 该方法只捕获关键数据流信息, 保留程序足够的特征, 同时使模型能实现跨智能合约的泛化. 其次, 在该方法中嵌入预训练的模型, 并使用考虑代码内在结构, 该方法在自然语言代码搜索、克隆检测、代码精化方面表现出更好的性能. Zhang 等人[38] 提出了一种多目标检测神经网络(MODNN), 该模型不仅可以利用已有的特征识别已知模式中的漏洞, 还可以从扩展的特征中学习未知模式中的漏洞和支持多个漏洞的并行检测, 具有高度可扩展性, 无需为每种类型漏洞分别训练模型, 大大减少了时间和人力成本. Cai 等人[39] 结合抽象语法树(AST)、控制流图(CFG)和程序依赖图(PDG), 构造具有句法和语义特征的智能合约函数的图表示. 为进一步增强该方法表示能力, 执行程序切片来规范控制流图, 消除与漏洞无关的冗余信息, 在源代码中保留了足够的句法和语义信息, 使用双向图表示进行上下文特征学习, 并使用混合注意力池化层提取图特征用于漏洞检测.

1.4 形式化规范与验证方法

基于符号执行的方法和基于深度学习的方法在本质上都是对已有漏洞针对性地进行检测, 这种方法的主要缺点是很难考虑到程序所有可能的执行情况, 因而很可能会漏掉一些已知和未知的漏洞. 形式化规范和验证方法则对程序的规范进行描述, 进而对其正确性进行验证, 因而能够避免漏洞的遗漏.

智能合约的形式化规范与验证包括形式化规范和形式化验证 2 个阶段. 形式化规范采用数学模型将智能合约的行为描述成一个数学公式或逻辑公式; 形式化验证通过自动化工具或手工证明验证该数学或逻辑公式的正确性, 从而证明智能合约的正确性. 相比传统的测试方法, 形式化规范和形式化验证方法具有更高的精度和可靠性, 能够发现更多的安全漏洞和错误.

智能合约进行形式化规范与验证的挑战主要来自 2 个方面: 一方面, 没有足够的文档来定义或描述 Solidity 及其字节码的全部特性, 难以解决出现歧义和未定义行为时的语义建模问题; 另一方面, Solidity 及其字节码中与区块链相关的特性, 例如与账户相关的映射、回退函数、与交易相关的断言等, 增加了合约语义分析与设计的难度, Solidity 支持多种方式的函数调用和异常处理等功能, 也需要一个更加通用的语义构造机制.

现有的工作主要包括对 Solidity 源码、Solidity 字节码和由 Solidity 编译成的中间语言的语义进行规范与验证. Jiao 等人[40] 面向 Solidity 语言开发了一个语义规范模型, 该模型围绕内存操作、实例创建和函数调用 3 个 Solidity 的核心特性, 提供智能合约的形式化规范以验证智能合约的执行行为是否安全. 此外, 该形式化规范的语义从通用的角度设计, 以适应智能合约语言特性的演变, 而且所设计的语义可以完全覆盖官方文档中提及的核心语言特性. 与语义级别的形式化规范不同, Permenev 等人[41] 对智能合约的功能属性进行形式化规范与验证, 提出了第一个智能合约功能属性自动验证器 VerX. VerX 专注于验证外部回调不会引发新执行路径的合约, 其语言规范将合约的执行形式转化为时间安全属性, 进而将安全验证问题抽象为可达性检查问题. Schneidewind

等人[42]设计了一个面向EVM字节码的静态分析器，它基于horn子句的可达性分析来验证合约的安全性，该研究更侧重于理论证明. 与之类似，Frank等人[43]通过对智能合约在Keccak256函数、内存模型、跨合约分析、验证4个方面建模，将合约的正确执行路径建立为约束系统，通过检查合约是否存在可达且被约束系统否定的状态判定合约是否存在漏洞. So等人[44]专注于智能合约的算术安全验证，提出了一种能够自动发现和利用合约不变量对智能合约进行高精度分析的算法，该算法可以检测所有的算术错误，同样也可以推广至其他安全属性的验证中.

1.5 混合方法

各种检测方法采用不同的思路来实现漏洞的检测，在检测效率和精度方面有各自的优缺点，如表1所示. 具体来说，基于符号执行的漏洞检测方法是在智能合约控制流图和数据流图上匹配具有特定子图结构的漏洞模式，具有较高的精度，然而这类方法比较依赖漏洞模式的定义，而且控制流图和数据流图的构建往往使该方法效率不高；基于深度学习的漏洞检测方法则将带有漏洞类型标注的合约源码或字节码及其控制流图用于深度神经网络模型的训练，以实现对漏洞的分类，这类方法在处理大规模智能合约的漏洞检测时效率较高，而其检测结果的准确度和精度往往差强人意，检测结果往往也不具有可解释性；基于形式化规范与验证的漏洞检测方法是对合约的正确性规范做出建模，然后基于模型约束实现合约安全性的检验，这类方法对合约审查有着较高的安全性，但在不同区块链系统及智能合约编程语言之间的迁移性和普适性较差，而且其识别出的漏洞合约通常具有较高的假阳性. 由此看来，不存在一种方法能够兼顾智能合约审查的安全性、漏洞检测的效率和漏洞识别的准确率.

Table 1 Comparison of Advantages and Disadvantages of Different Vulnerability Detection Methods

表1 各类型漏洞检测方法优缺点对比

漏洞检测方法	效率	准确率	标注	定义漏洞	可解释性	可迁移性	可靠性
符号执行	低	高	不需要	需要	好	差	低
深度学习	高	低	需要	不需要	差	好	低
形式化规范与验证	高	低	不需要	不需要	好	差	高
混合方法	低	高	需要	需要	好	好	高

随着各项漏洞检测方法的发展成熟，部分专家学者发现结合不同的漏洞检测技术能够有效提高智能合约漏洞识别的效率和精度，由此产生了一些用于漏洞检测的混合方法. Samreen等人[45]结合静态分析和动态分析方法提出SmartScan分析器，以用于检测由于意外恢复而导致的智能合约拒绝服务攻击(DoS caused by unexpected revert)漏洞. SmartScan的第1阶段采用基于模式识别的静态分析技术并执行自动化的合约测试，以有效地识别被测合约中所有可能的漏洞；第2阶段根据测试发现的漏洞信息自动生成攻击合约，并将被测合约和攻击合约部署至以太坊的测试网络中进行动态分析，以确认该漏洞的存在. 同时，Liu等人[46]将深度学习方法与传统的专家模式相结合，以一种可解释的方式进行智能合约漏洞检测，其中神经网络用于处理复杂的代码语义理解，专家模式为精确分析提供有价值的局部信息. 他们的方法由3部分组成：局部专家模式提取、合约代码语义图构造、结合局部专家模式和全局图特征用于漏洞检测及其权重输出的多编码网络. 随后，Liu等人[47]再次探索了将图神经网络和专家知识相结合进行智能合约漏洞检测，不同的是此处的专家知识用于提取安全模式特征，而不是漏洞模式. 他们将源代码转换为控制流图和数据流图，并通过节点消除方法规范图结构以突出关键的图节点；然后将归一化的图输入到时域消息传播网络中进行特征提取，同时也利用专家知识从源代码中提取安全模式特征；最后结合图特征和安全模式特征得到最终的漏洞检测结果.

2 智能合约骗局识别技术

除了潜在的漏洞会被恶意攻击外，智能合约本身在功能上也可能是恶意的，这类合约通常会伪装在一个Web 3.0项目中以吸引用户投资，用户因缺乏对项目的深入了解很容易上当受骗. 此外，在公链系统或开放联盟链系统下，用户可以随意加入并部署智能合约，这也大大提高了骗局合约部署的可能性. 为了维护Web 3.0数字经济市场健康发展，研究和开发对智能合约骗局的识别技术以降低用户的经济损失是至关重要的. 本文总结了现有研究工作中发现

的智能合约骗局，分析其检测方法后发现，大部分骗局识别技术的本质都是神经网络分类模型. 根据输入数据的类型，我们将这些方法分为基于源码的、基于交易的、基于字节码的、基于操作码的和其他类型的智能合约骗局识别技术.

2.1 智能合约骗局类型

通过对现有工作的分析，本文总结常见的智能合约骗局类型有 3 种：

1）庞氏骗局. 庞氏骗局是传统金融投资领域中一种古老但久经考验的投资欺诈手段. 它往往以投资项目的形式出现，但其背后并没有真正盈利的项目，给先前投资者的回报来自于后来者的资金投入，这种情况一直持续到模式难以为继. 加密货币因其安全性而受到公众的信任，由于以太智能合约的透明性和不变性，使得许多投资者相信智能合约不会构成庞氏骗局的风险，这使得隐藏在智能合约中的庞氏骗局更具欺骗性，许多智能合约庞氏骗局的投资者因为不了解智能合约的源代码而蒙受了巨大损失.

2）蜜罐骗局. 蜜罐骗局是一种新型恶意陷阱合约，是指攻击者通过故意部署带有漏洞的智能合约来引诱受害者受骗，以窃取受害者的加密资产. 此类合约在设计上似乎存在一个明显的漏洞，例如允许用户从合约中获取数字资产，但前提是用户需要向合约中转入一定数额的资金. 然而，一旦用户试图利用这一明显的漏洞并存入了一笔资金，合约中潜在的陷阱就会被触发以阻止用户取出. 由于受害者只关注明显的漏洞，而不考虑合约中可能隐藏的陷阱，因而这种合约很容易引诱用户上当受骗.

3）勒索软件. 勒索软件是一种恶意软件，它通常会恶意加密用户的重要文件，并要求用户支付一定的金钱来执行解密. 勒索软件是网络犯罪中常见的手段，也存在于 Web 3.0 去中心化应用中. 用户的数据被勒索软件加密后，通常只能支付赎金赎回. 勒索软件因其恶意性被认为是最危险的威胁之一.

2.2 基于源码的识别技术

常用的对智能合约进行分类的方法是使用自然语言处理（natural language processing, NLP）技术分析合约源码识别智能合约骗局. 长短期记忆（long short term memory, LSTM）是一种循环神经网络，可以有效传递和表达长时间序列中的信息并且不会导致长时间前的有用信息被忽略，能很好地刻画具有时空关联的序列数据. 自然语言处理领域常用 LSTM 对语言建模，即用 LSTM 提取文本的语义语法信息. 借鉴自然语言处理的思想，智能合约中包含源代码和注释，可以通过基于注意力的 LSTM 网络[48] 从源代码和注释中捕获语义特征. 但仅依靠智能合约的源代码和注释语义，分类效果并不理想. 正如之前提到的，基于神经网络的文本分类方法主要使用 LSTM 对文本进行分类，但智能合约代码区别于传统文本，智能合约包含源代码和代码注释，二者均包含语义信息，并且代码注释通常较为简短，存在语义稀疏问题. Tian 等人[49] 提出了一种新的分类模型 SCC-BiLSTM，采用高斯 LDA（GLDA）模型和注意机制来提高分类器的性能. 该模型使用注意力机制捕获重要的代码特征，解决了源代码中注释的语义稀疏问题.

以太坊上目前部署了数以百万计的智能合约，其中不乏是不法分子为牟利而部署的欺诈合约，其中又以庞氏骗局合约、蜜罐骗局合约为代表，给投资者带来了巨大损失，因而，正确识别骗局合约是必要的. 现有的庞氏骗局合约检测的主要思路是人工标注骗局合约后通过训练分类器来检测更多的庞氏骗局合约. 但人工标注无法捕获源代码的结构和语义特征，其检测效果并不理想. 为了保留源代码的结构信息，Chen 等人[50] 提出了一种基于深度学习的庞氏骗局检测方法，即 MTCformer. 首先，使用基于结构遍历（SBT）的方法将智能合约代码的抽象语法树转化为特殊格式的代码标记序列以保留结构信息；然后，利用多通道的文本卷积神经网络（TextCNN）和 Transformer 从源代码中自动提取结构和语义特征，并学习代码标记序列之间的长期依赖关系；最后，使用具有成本敏感的损失函数的全连接神经网络进行分类. He 等人[51] 提出 CTRF（code and transaction）方法来检测以太坊上的庞氏骗局合约，通过提取智能合约代码的文字特征、序列特征以及交易特征形成一个数据集，对训练集进行过采样，以处理正样本和负样本不平衡的问题，该方法提高了模型的召回率. Ishimaki 等人[52] 研究了 8 种类型蜜罐骗局合约，并衡量了每种蜜罐骗局合约造成的危害，通过对资金转移过程的论证和对合约代码的分析，他们还发现了一种新型蜜罐骗局合约. 在基于合约名称和参数的分析中，提取了可以在 EtherScan 上确认的蜜罐类名称的特征，以及在识别蜜罐骗局合约的过程中需要关注的函数 *msg. sender. transfer*() 参数的特征，分析表明在转账过程中使用 *this. Balance* 的合约可能是蜜罐骗局合约.

2.3 基于交易的识别技术

基于源码进行智能合约骗局识别具有较高的准确率，然而在区块链平台中，大部分合约都不是开源的，为此部分学者考虑通过分析交易记录识别合约

骗局. 在以太坊中，交易是从一个账户地址发送到另一个账户地址的消息，以太坊的主要活动是由交易触发的，包括以太币转账、创建和调用智能合约. 区块链具有不可篡改的特性，自以太坊创建以来，智能合约的每一笔交易记录都被完整保存在以太坊中. 此外，不同类型的智能合约具有不同的交易行为，我们可以通过收集智能合约及其交易数据，对智能合约的交易行为进行分析，实现对智能合约的分类，进而利用已知的骗局合约发现更多智能合约骗局.

Hu 等人[53]提出一种基于交易的分类和检测方法来识别智能合约骗局，重点研究以太坊中的交易数据，从以太坊收集并筛选出 10 000 多个包含一定交易量的智能合约，基于对这些合约的分析，发现交易行为在不同类型的合约中具有不同特征，最终找到了 4 种交易行为模式. 此外，他们提出了一种数据切片方法，对收集的智能合约进行切片以解决数据集不足的问题，再使用切片后的数据集来训练 LSTM 模型，并利用模型对智能合约进行分类. 黄步添等人[54]提出了一种基于词嵌入模型的智能合约分类方法，该方法在提取智能合约源代码和注释的语义特征的同时，也提取智能合约账户交易特征，通过 LSTM 网络捕获合约源代码的语义，并获得词向量. 最后，将词向量和账户特征输入 Bi-LSTM 获得输出类别标签的概率分布. 该方法将账户信息和源代码信息相结合，可以有效提高分类效果.

此外还存在通过构建交易图来进行骗局合约识别的研究工作. 现有的基于交易图的异常行为检测方法通常侧重构造同构交易图，未区分节点和边的异构性，导致部分交易信息丢失. 现有的异构建模方法可以通过元路径描述更为丰富的信息，但忽略了实体之间的时间依赖性. Jin 等人[55]引入了时间感知元路径增强(TMFAug)作为即插即用模块，以捕获庞氏骗局中基于元路径的交易模式，该模块可以与现有的基于交易图的庞氏骗局的检测方法结合. 基于交易的检测智能合约的方法除了可以用于检测庞氏骗局，也可以用于识别蜜罐骗局. Camino 等人[56]提出了一种基于合约交易行为的检测蜜罐合约的方法，在合约创建者、合约、交易发送者和其他参与者之间创建了所有可能的资金流动案例的分区，添加了交易聚合特征以及其他合约特征，该方法能帮助检测到之前未被检测的蜜罐骗局合约.

以往基于交易识别智能合约骗局的方法在识别代码高度相似的合约时表现不尽如人意，因此，研究者逐渐倾向于将账户特征、合约代码特征和交易特征结合考虑. Fan 等人[57]提出名为 TTG-SCSD 的框架，将账户动态交互信息和 TDA 方法引入智能合约骗局检测，通过挖掘动态演变账户交互信息的拓扑特征，利用该特征和拓扑数据分析进行智能合约骗局检测. TTG-SCSD 为每个合约构建离散的动态交互图，并设计建模表征账户行为的交互特征，结合拓扑量化机制来捕获交易中的合约意图. Wang 等人[58]提出了一种基于 LSTM 的庞氏骗局的检测方法 PSD-OL，将合约账户特征和合约代码特征结合考虑，利用过采样技术以提取有效特征数据用于检测，使用 LSTM 训练检测模型，用于检测以太坊中的庞氏骗局合约. 但现有大多数庞氏骗局检测技术存在 2 大制约：缺乏时间预警以及无法融合多元信息，导致庞氏骗局检测信息滞后且性能不及预期. Jin 等人[59]提出一种用于识别以太坊庞氏骗局的双通道预警框架，该框架从原始数据中提取并融合智能合约代码级和交易级特征. 此外，提出了一种用于生成交易图序列的时间进化增强策略，在一定程度上缓解了数据不平衡问题.

2.4 基于字节码的识别技术

现有的基于 NLP 和合约代码的模型识别智能合约骗局存在 2 个主要问题：1)基于 NLP 的模型其应用场景有限，只能对开源的智能合约进行分类，然而，在以太坊上，只有不到 6% 的智能合约是开源的，这意味着基于 NLP 的模型无法对超过 94% 的智能合约进行分类. 2)基于 NLP 的模型容易受到源代码上下文特性的影响，在不更改代码逻辑的条件下，开发人员可以修改源代码的上下文特性，如注释、变量名和函数名，也可以采用不同方式编写源代码以执行相同的功能. 但是，任何对源代码的添加、删除或修改操作都有可能欺骗基于 NLP 的分类器. 为了解决上述问题，需要能够在无法获取智能合约源代码的情况下对智能合约进行分类的模型.

基于交易的识别技术需要使用历史交易数据特征，但通常庞氏骗局智能合约的生命周期都比较短，这种方法不能及时检测出庞氏骗局，只能检测出已经积累足够交易数据的庞氏骗局，当这些模型检测到欺诈时，欺诈已经广泛传播，危害已经造成. 受字节码特性在智能合约其他领域广泛使用的启发，如合约漏洞和欺诈检测，我们发现字节码可以从逻辑方面反映功能特性，因此将智能合约的字节码特征应用于智能合约分类.

Bartoletti 等人[60]通过阅读开源智能合约的源代码和搜集互联网上的相关资料，确定了以太坊上 100 个多庞氏骗局合约. 基于 Levenshtein 测量合约字节

码的相似度，识别了部分隐藏源代码的庞氏骗局. 在此基础上，将庞氏骗局划分为 4 种不同类型，深入分析庞氏骗局的生存周期以及给参与者带来的问题. Zheng 等人[61] 建立了更大的庞氏骗局智能合约数据集，从多个视图中提取包括字节码、代码语义和开发人员相关信息特征，利用机器学习的方法构建多视图级联集成模型 MulCas，可以在智能合约创建时识别庞氏骗局. Lou 等人[62] 提出一种基于改进的卷积神经网络识别智能合约中庞氏骗局的方法，该方法首先引入了空间金字塔池化方法对卷积神经网络进行改进，避免了现有方法中复杂的特征选择和统计过程，解决了智能合约字节码长度不一的问题. 改进后的卷积神经网络模型，其准确率和召回率都比当前最优的随机森林方法有所提高. Hara 等人[63] 提出了基于字节码检测蜜罐合约的模型，使用逆文档频率提取字节码特征，并使用 word2vec 学习 Solidity 的字节码分布表示，该模型能事先预测蜜罐合约以减少受害者的数量.

Shi 等人[64] 提出一种基于字节码对智能合约进行分类的方法，该方法与基于源码的识别技术相比具有更好的抵抗攻击的能力，通过特征选择与集成学习解决了智能合约中数据不平衡问题，与代码特征相比，账户特征对智能合约分类的影响更小. 林丹等人[65] 也提出了一种基于字节码对以太坊智能合约进行分类的方法，该方法首先收集了不同类型智能合约字节码以及对应类别标签，包括交易所、金融、赌博、游戏和高风险 5 种不同类别. 然后，基于收集的合约字节码，提取了能够丰富智能合约语义信息和逻辑结构的控制流图(CFG)特征和操作码特征. 最后，基于提取的智能合约特征，使用 XGBoost 等分类算法对智能合约进行分类. Hu 等人[66] 设计了 SCSGuard 框架为不同类型的骗局提供了统一的解决方案，SCSGuard 是一个基于 N 元语法特征和注意力神经网络的框架，适用于所有智能合约字节码. 与以往通过基于静态和动态分析识别智能合约漏洞的方法不同，该方法无需人工定义任何规则，大大降低了人力成本. SCSGuard 利用具有关注层的序列学习神经网络 GRU 以捕获智能合约字节码中的隐藏信息，在庞氏骗局、蜜罐合约的检测中有较好的性能表现.

2.5 基于操作码的识别技术

除了字节码外，基于操作码的自动检测模型也能够及时检测出庞氏骗局. 目前也存在一些工作根据操作码序列特点，采用系统化的方法逐步构建庞氏骗局的检测模型.

Peng 等人[67] 提出了一种有效检测智能庞氏骗局全生命周期的模型，该模型仅使用智能合约的操作码特征，能够及时、全面、准确地检测智能合约中的庞氏骗局，在庞氏骗局的生命周期中的任何时刻都能保证其性能. 此外，他们提出一种检测庞氏骗局的系统建模策略，考虑了包括特征类型、特征值的计算方法、过采样比例、关键特征和分类器等因素，该策略可以在较少的迭代次数中得到一个有效的模型. Chen 等人[68] 提出一种利用数据挖掘和机器学习的方法来检测区块链上的庞氏骗局的方法，该方法首先下载了 3 071 份智能合约，包括交易信息和源代码信息；然后，将源代码编译为字节码，并将字节码分解为操作码，再从交易中提取账户特征，从操作码中提取代码特征；最后提出一个基于账户特征和代码特征的回归树模型 XGBoost 的分类模型用于检测潜在的庞氏骗局. 虽然目前已有解决方案关注智能合约庞氏骗局检测，但这些方法仍然存在 2 个问题：提取的合约特征不够完整，以及用于检测庞氏骗局的算法不够高效. Zhang 等人[69] 提出了一种基于改进的 LightGBM 算法的智能合约庞氏骗局识别方法，该算法用 Smote_Tomek 混合采样代替 LightGBM 的权值分配，创新性地提取字节码特征，并将其与已有的用户交易特征和操作码特征相结合来识别庞氏骗局.

基于操作码的识别技术在识别蜜罐合约和勒索软件方面也取得一定研究成果. Torres 等人[70] 提出了一种自动检测以太坊蜜罐智能合约的工具 HONEY-BADGER，该工具使用符号执行和精确启发式算法的组合来自动检测蜜罐合约，对以太坊上的蜜罐智能合约进行系统分析，通过该工具确认了 151 935 份智能合约中至少有 282 个蜜罐合约. Nalinipriya 等人[71] 提出基于 WMFO 和 deep RNN 来识别勒索软件的方法，该方法首先提取操作码的 N 元语法特征，使用 TF-IDF 来帮助确定特定单词在上下文中对数据的重要性，采用概率主成分分析来选择适用于 deep RNN 进行分类的显著特征. Sun 等人[72] 提出一种不依赖交易数据的 PonziDetector 检测技术，该技术引入了行为森林来捕获智能合约在交互过程的动态行为，帮助尽早识别庞氏骗局，PonziDetector 的准确率和召回率达到了 94.6% 和 93.0%. Chen 等人[73] 提出一种语义感知的检测方法 SADPonzi，该方法的符号执行技术从运行时的合约代码中构造控制流图，并生成重要的操作码路径及其符号上下文，综合这些信息来识别与投资者相关的交易行为和所采取的分配策略，可以达到 100% 的准确率和召回率.

2.6 其他类型的识别技术

除了上述提到的识别智能合约骗局的技术以外，还存在一些其他类型的识别技术，包括基于 ABI 分析的智能合约分类方法和基于有序目标统计与有序增强的庞氏骗局检测方法.

智能合约的分类方法大多基于源代码或字节码进行分析，然而作为智能合约的接口，abi 中存在关于智能合约功能和行为的关键信息. 为了解 abi 对智能合约分类的影响，Sun 等人[74] 提出一种基于 abi 的智能合约自动分类方法. 他们采用 *MCC* 作为分类模型的评价指标，*MCC* 的计算过程如式(1)所示：

$$MCC=\frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}},\tag{1}$$

其中 *TP*, *FP*, *TN*, *FN* 分别表示测试数据集中分类结果为真阳性、假阳性、真阴性、假阴性的样本个数. *MCC* 的取值范围为 −1~1, *MCC*=−1 表示完全错误的分类结果，*MCC*=0 表示随机分类结果，*MCC*=1 表示完美分类结果. 与传统的基于源代码分类方法相比，Sun 等人[74] 的方法将 *MCC* 评分提高了 0.1 以上.

现有的基于梯度 Boost 的算法在处理类别特征和计算估计梯度时，存在由于目标泄露，即使用当前模型所基于的目标值来估计每个步骤中使用的梯度导致的预测偏移问题，即训练数据的条件分布偏移了测试数据. 并且这些算法没有考虑到智能合约中庞氏骗局数据集分布不平衡的问题，即所有合约中庞氏骗局合约占比小，导致模型过拟合和泛化能力较弱. Fan 等人[75] 提出一种新的基于智能合约平台的庞氏骗局检测方法，即采用有序目标统计方法对模型进行训练，可以直接处理类别特征. 使用数据扩充方法以解决数据集不平衡的问题，模型的 *F-score* 达到了 98%. Fan 等人[76] 提出一种基于有序增强思想的防泄露庞氏骗局检测模型 AI-SPSD，该模型克服了基于有序目标统计的预测偏移问题，可以从运行时操作码级别及时检测新部署的智能合约庞氏骗局. 该模型通过分析开源 DApp 以进一步扩展庞氏骗局的数据集，通过数据增强消除了数据集不平衡问题，提高了生成模型的质量和性能，模型的 *F-score* 达到了 96%.

3 非法交易行为检测技术

Web 3.0 项目发展的同时，吸引了许多网络犯罪分子的注意，他们企图利用区块链平台的匿名性进行洗钱、钓鱼诈骗、勒索赎金收取等非法交易行为，并试图通过一些链上、链下的手段操纵数字资产价格从中获利. 这些非法行为危害了社会安全和公共秩序，导致了金融市场的不稳定，因此监管机构和市场参与者需要实时监控区块链平台中的交易动态，及时发现这些非法交易行为. 然而，区块链系统中交易的匿名性使得对非法交易行为的追踪变得十分困难，区块链系统巨大的交易数据增长量也给实时分析带来挑战. 区块链系统中存在的非法交易行为种类多样，学术界针对每种类型的非法交易行为提出了不同的检测技术，3.1~3.4 节将分别介绍洗钱、钓鱼诈骗、市场操纵、勒索软件的赎金支付 4 类非法交易行为的检测方法及研究现状.

3.1 洗钱模式检测

洗钱在全球范围内是一个影响很大的问题，它是指犯罪分子从犯罪活动中非法获取钱财后将其转换成合法的资金. 在比特币的场景下，不法分子盗取了大量比特币后意图将盗取的比特币通过交易所直接兑现. 然而，在开展业务之前，交易所通常会实施“了解你的客户”(know your customer, KYC)流程以验证用户的身份，审查他们的金融活动，并确定他们可能构成的风险，这使得不法分子的洗钱行为容易暴露. 为了躲避 KYC 的检查，不法分子通过使用 Bit-Laundry, Helix Light, Bitcoin Fog, Tornado. cash 等比特币混币器服务增加交易被追踪的难度，从而掩盖自己的犯罪行为. 比特币混币服务的目的是为了增强交易的匿名性、隐蔽资金的来源使其不可追踪，混币器的原理如图 4 所示. 比特币地址 *addr*1, *addr*2, *addr*3 的用户分别向混币器的 3 个地址 *mix*1, *mix*2, *mix*3 发送 1 个比特币，并提供地址 *addr*4, *addr*5, *addr*6 用于比特币的接收. 然后混币器每次从 *mix*1, *mix*2, *mix*3 中随机选择 1 个地址，将比特币分别转账给 *addr*4, *addr*5, *addr*6. 这样的结果是混淆了比特币系统内交

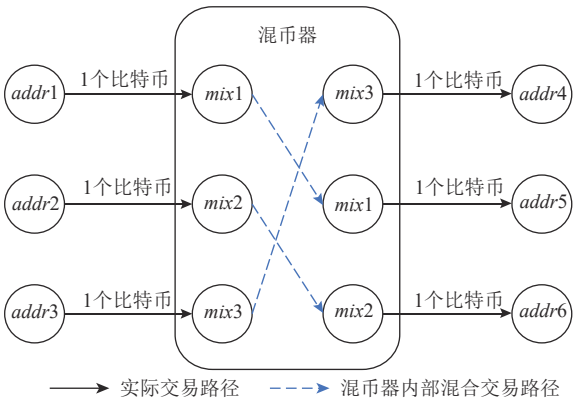

Fig. 4 Illustration of mixer principle

图 4 混币器原理示意图

易的发送者和接收者之间的关系，从而增加了追踪资金来源和分析用户交易行为的难度. 近年来，学术界对于区块链反洗钱问题进行了深入研究，研究方法包括逆向工程、时序图、机器学习以及深度学习等.

Möser 等人[77]对比特币反洗钱(AML)工作进行了系统研究，他们基于从区块链中提取的交易图，系统分析了 Bitcoin Fog, Blockchain. info, BitLanudry 这 3 种流行的交易匿名器，使用逆向工程方法来了解操作模式，并尝试将匿名交易追溯到所探测的目标账户，该工作概述了在不了解 KYC 的情况下仅利用交易图中的公共信息的“反洗钱”策略. 结果表明该策略成功探查到了 BitLaundry 的输入和输出账户之间的关联. 然而，此方法仅能进行简单的混币器洗钱检测，对于具有庞大用户基础的混币器其检测效果欠佳.

Wu 等人[78]针对混币服务关联地址的检测问题提出了基于时序图的方法，该方法利用比特币交易记录构建了 2 个时间有序图，包括同构地址交互网络(AAIN)和异构地址交互网络(TAIN). 具体来说，提出了一种新的 ATH motif 概念来整合边缘属性信息与高阶结构，开发了基于 AAIN 中的时间基序和 TAIN 中的 ATH 基序的混合基序作为混合检测的关键特征，利用设计的这些特征建立了一个基于 PU (learning from positive and unlabeled examples)学习的检测模型来处理混合检测中标签极度不平衡的问题. 在 3 个真实的比特币数据集上的实验证明了此方法的有效性，但由于该检测模型依赖于先验信息，对于未知的复杂混合策略可能存在检测遗漏的问题.

除了传统的统计方法外，随着人工智能的发展，一些学者采用机器学习以及深度学习的方式对涉嫌洗钱的交易和账户进行聚类和分类. Meiklejohn 等人[79]使用启发式聚类方法基于共享权限的证据对比特币钱包进行分组和识别，总结出 3 种典型疑似洗钱的交易模式，分别是汇聚、折叠和分割. Ketenci 等人[80]使用时频分析作为特征提取方法，实现了 2 维时频特征，其使用随机森林方法作为机器学习方法，并采用模拟退火进行超参数整定，在相关测试集上获得的良好的表现. Lorenz 等人[81]探讨了多类深度学习和机器学习技术在加密货币反洗钱中的适用性，包括深度神经网络(DNN)、随机森林、K 近邻(KNN)和朴素贝叶斯(NB)，结果表明与其他分类器相比，DNN 和随机森林分类器在减少误报方面取得了最高的准确率. 除了 DNN 外，图卷积神经网络(GCN)也常被用于解决分类问题，Alarab 等人[82]开发了一种将 LSTM 与 GCN 结合起来的分类模型 temporal-GCN，其仅使用交易特征对比特币椭圆数据交易图的非法交易进行分类. 具体来说，LSTM 不仅考虑了比特币交易图的时间序列，而且考虑了图中最具影响力的前 k 个节点的图结构数据，整体模型的准确率达到了 97.7%.

3.2 钓鱼诈骗检测

网络钓鱼诈骗也是区块链场景下最为猖獗的犯罪之一，给区块链平台和用户造成了巨大的经济损失. 网络钓鱼通常是通过伪装成通信网络中受信任的参与者来获取敏感信息，如账户名、密码和金融账户号码等的欺诈行为. 在区块链中，钓鱼者则是通过邮箱或者社交平台发布包含以太坊地址的钓鱼链接，诱导用户向钓鱼地址投资转账以骗取投资者的资产，其流程如图 5 所示. 钓鱼诈骗事件通常发生于以太坊这个全球第 2 大区块链平台，近年来多起以太坊上的非同质化代币(non-fungible token, NFT)失窃事件与钓鱼诈骗有关，攻击者伪造网站、发送邮件或者使用不安全的 Discord 机器人在某些官方的 Discord 服务器上发布钓鱼链接，从而获取受害者的隐私数据甚至控制受害者的以太坊钱包，造成财产的损失. 例如 2022 年 5 月 NFT 艺术家 Beeple 的推特账户被黑客入侵，攻击者利用推特这一网络媒体散播钓鱼链接获取受害者信息，窃取了共计 43.8 万美元的加密货币和 NFT. 钓鱼诈骗攻击占据了超过半数的以太坊网络犯罪攻击，造成的损失占比也最高，因此目前迫切需要有高效的方法来检测钓鱼诈骗账户. 学术界近年来针对此问题有一系列研究，检测方法大致分为机器学习技术以及深度学习技术.

用于检测钓鱼诈骗账户的机器学习方法包括 lightGBM、SVM、决策树、Naïve Bayes 等. Chen 等人[83]提出了一种基于区块链交易检测钓鱼账户的系统方法，并以以太坊为例验证了其有效性. 具体来说，他

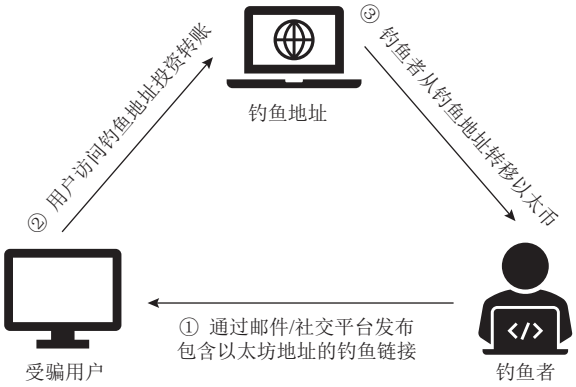

Fig. 5 Illustration of phishing scam process

图 5 钓鱼诈骗流程示意图

们提出了一种基于交易记录的图级联特征提取方法和一种基于 lightGBM 的双采样集成算法来构建识别模型. Wu 等人[84]提出了一种通过挖掘交易记录来检测以太坊网络钓鱼诈骗的方法. 具体来说, 首先从 2 个授权网站爬取标记的钓鱼地址, 并根据收集到的交易记录重建交易网络; 然后, 通过考虑交易金额和时间戳, 用 trans2vec 提取地址特征, 用于后续钓鱼识别; 最后, 采用一类支持向量机(SVM)将节点分为正常节点和网络钓鱼节点. 这项工作通过网络嵌入的方式对以太坊网络钓鱼检测进行调查, 并提供了对如何嵌入大规模交易网络特征的见解. Kabla 等人[85]提出了一种称为以太坊网络钓鱼骗局检测(Eth-PSD)的检测机制, 该机制试图使用一种基于机器学习的新方法来检测网络钓鱼骗局相关的交易, 其使用了包括决策树、Naïve Bayes 等 8 种不同的机器学习方法进行综合检测, 并选取最佳方法作为 Eth-PSD 的主分类器, 其主要贡献点在于降低了数据集的维度, 弥补了之前工作中数据集不平衡、特征工程复杂和检测精度低等不足之处.

由于经典机器学习模型只适用于浅层的线性分类任务, 对于复杂非线性关系的拟合效果欠佳, 随着区块链交易网络的复杂化, 其难以检测某些隐藏在深处的违法账户. 为了更深层次识别出违法账户以及非法交易, 学者们引入了深度学习方法. 现有的基于深度学习的钓鱼诈骗检测方法通常采用 2 种方式对交易数据建模以作为模型的输入:

1)建模方式是结构化数据, 传统神经网络直接对结构化数据进行建模分析. Wen 等人[86]提出了一种基于混合深度神经网络检测钓鱼诈骗账户的模型 LBPS, 该模型提供了一种新颖的交易记录分析方法, 采用 BP 神经网络获取交易记录特征之间的隐式关系, 且 LSTM-FCN 神经网络从目标账户的所有交易记录中获取时间特征. 实验结果表明 LBPS 能有效识别网络钓鱼诈骗账户.

2)建模方式是采用交易图的形式, 使用图神经网络等方法对图数据进行分析. Ron 等人[87]和 Zhao 等人[88]用图的形式分别对比特币和以太坊的交易演化过程进行了建模, Ron 等人[87]从交易图中发现大型交易往往与多个小型交易有强关联性, Zhao 等人[88]从时间图的角度研究了以太交互网络的演化本质, 整理了 4 种以太区块链网络的增长率和模型, 分析了高阶顶点的活跃寿命和更新率, 获取了相关的图特征. Chen 等人[89]使用类似方法将账户和交易视为节点和边, 将网络钓鱼帐户的检测转化为节点分类问题, 提出了一种基于图卷积网络和自动编码器的检测方法以精确区分网络钓鱼账户. Li 等人[90]提出了一种时间事务聚合图网络(TTAGN), 具体来说, 其在时间边缘表示模块中对节点之间的历史交易记录的时间关系进行建模, 以构建以太坊交易网络的边缘表示, 并在 edge2node 模块中把节点周围的边聚合和将拓扑交互关系融合得到交易特征, 最后进一步将交易特征与图神经网络获得的公共统计和结构特征相结合, 以识别网络钓鱼地址. CT-GCN[91]和 Trans-DetectionNet[92]则使用了基于图卷积网络的检测方法, 前者使用了双层图卷积网络进行特征学习, 后者提出了边缘采样到节点向量(Esmp2NVec)的节点嵌入算法, 都得到了较好的准确率及 $F1$ 分数.

3.3 市场操纵行为检测

加密货币是一种主要基于区块链技术的新型数字资产, 近年来受到了极大的关注. 推动加密货币发展的关键事件是 2008 年由中本聪提出的比特币, 15 年之后, 根据 CoinMarketCap 数据统计, 截至 2023 年 1 月 1 日, 全球加密货币市场共有加密货币 22 163 种, 总市值共计约 7 986.88 亿美元, 与传统货币体系相比加密货币具有更低的交易成本、更高的交易安全性和透明度等优势, 受到了大众的关注. 然而, 随着加密货币的火热, 其弊端也渐渐展现出来, 最大的问题在于其缺乏监管, 导致了市场操纵行为的增加. 市场操纵是指通过不公平的措施影响市场的定价, 从而获取不正当利益.

Eigelshoven 等人[93]对当前加密货币市场操纵方案进行了全面地描述和分类, 该方案使用以概念为中心的方法, 将加密货币市场操纵的方法总结为哄抬抛售、订单操纵、洗牌交易、抢先交易、内幕交易、分布式拒绝服务攻击(distributed denial of service attack, DDoS)以及稳定币发行等 7 种策略, 如表 2 所示. 哄抬抛售策略通过协调购买力来推高区块链发行币的价格, 这些计划多是通过 Twitter, Reddit, Telegram, Discord 等社交媒体和平台组织的, 尤其是具有高度匿名性的 Telegram. 哄抬抛售的目标往往是较新的市值低、发行量小的币种, 并且整个哄抬抛售的过程相对较短. 洗牌交易指的是自己执行交易来创造人工交易量, 形成循环交易, 夸大特定资产的活动和价值吸引新的买家. 订单操纵指的是交易所通过短时间内伪造大量买入或卖出订单(并不实际执行这些订单, 或者通过发起大量订单, 然后立即取消这些订单), 来操纵市场对某种货币供需的感知, 从而影响该货币的价格. 抢先交易主要发生在以太坊中, 由于

Table 2 Summary of Cryptocurrency Market Manipulation Methods

表 2 加密货币市场操纵方法总结

策略	描述	角色/平台
哄抬抛售	协调购买力来推高区块链发行币的价格.	社交媒体
洗牌交易	执行交易来创造人工交易量，形成循环交易，夸大特定资产的价值.	交易所
订单操纵	通过短时间内伪造大量订单来操纵市场对某种货币供需的感知，但这些订单生命期短，或实际并不执行.	交易所
抢先交易	设置更高 gas 费用抢先完成攻击交易.	区块链
内幕交易	滥用可预测未来交易的内幕信息来制定交易策略.	交易所
分布式拒绝服务攻击	通过反复发送大量的服务请求，试图使一个网站或网络提供的服务失效.	互联网
稳定币发行	大量发行稳定币用于购买某种数字货币，此时该数字货币流通量减少，从而导致其价格上涨.	交易所

以太坊区块链中用户以 gas 的形式向矿工支付少量金额以确认交易，此时抢跑者在一笔正常交易等待打包的过程中通过设置更高 gas 费用抢先完成攻击交易，以此攫取用户的利益. 内幕交易是指滥用可以预测未来交易的内幕信息来制定交易策略，加密货币的内幕交易主要发生在交易所的硬币上市背景下，这可能导致突然的价格波动. 分布式拒绝服务攻击是指通过反复发送大量的服务请求，试图使一个网站或网络提供的服务失效. 与简单的拒绝服务攻击相比，DDoS 攻击由几个不同的来源，如远程控制的僵尸网络进行. DDoS 的目标往往是加密货币交易所，使得交易所服务中止，冻结交易所内的交易，并利用现有的数量来压低目标加密货币的价格. 稳定币是一种特殊加密货币，其价格由现实的货币政策控制，例如 Tether 是一种固定在 1 美元价值上的稳定货币. 具有稳定币发行权的组织或个人可以通过大量发行稳定币用于购买某种数字货币，致使该数字货币的流通量减少，从而导致其价格上涨. 若在此之前发行者手中持有大量此种数字货币，则他将从此次稳定币的发行中获利. Eigelshoven 等人[93] 随后对这 7 种策略进行了特征化，并确定了市场存在的 6 个漏洞，包括交易所标准和复杂程度的差异、加密货币生态系统的匿名性、市场监管的缺失、低市场壁垒及投资者经验的缺乏、社交媒体的利用，以及区块链生态系统和共识协议中的设计漏洞和利用.

学术界对如何发现存在的市场操纵问题进行了深入的研究. Chen 等人[94] 提出了一种挖掘交换交易网络来发现市场操纵及其操纵模式，从而实现对比特币的监管. 以比特币交易所 Mt. Gox 所泄露的比特币交易历史为样本，Chen 等人[94] 首先根据其特征将账户分为 3 类，然后将交易历史构建为 EHG, ELG, NMG 这 3 张图，通过对图的度量分析得到了市场被操纵的证据，随后将图序列重塑为矩阵，通过对矩阵进行 SVD(singular value decomposition)，识别出了一些异常的基本网络. Pereira 等人[95] 同样对 Mt. Gox 活跃时期的比特币区块链进行了图分析，重点在于评估了每个节点的度中心性的特征，分析了具有最高中心性值的节点排名随时间的变化趋势，提供了使用节点排名更改来检测恶意活动的思路，实验证明，使用这个排名方法可以预测网络中的异常行为.

3.4 勒索赎金支付交易追踪

勒索软件是一种恶意软件，它通过电子邮件或基于网络的漏洞进行传播，对受害者的文件、数据等资料进行加密，要求受害用户交付赎金以解密文件. 在勒索软件诞生之初，交付赎金的方式主要是真实货币，例如银行卡支付等. 这种支付方式需要经由银行等中心化机构，整个交易流程容易追溯，银行可冻结赃款及追踪勒索者的地址. 为了更好地隐藏自己的犯罪行为，在比特币等虚拟货币支付形式出现后，勒索者们将比特币作为勒索赎金的支付方式，这给追踪赃款带来了困难. 如何识别涉及勒索交易的区块链地址是追踪勒索行为资金流动的关键，学术界对此进行了研究.

Huang 等人[96] 创建了一个测量框架，用于执行勒索软件支付、受害者和运营商的大规模的端到端测量. 该框架通过结合勒索软件二进制文件、种子赎金支付、感染受害者遥测和带有所有者信息的比特币地址数据库等一系列数据源，从受害者支付比特币的时刻追踪金融交易，对勒索软件运营商套现策略进行了分析，发现了比特币交易所 BTC-e 容易成为勒索赎金套现的中介.

Akcora 等人[97] 提出了一个高效且易于处理的数据分析框架用来提取与比特币交易相关的特征，并使用基于拓扑数据分析的方法和新的区块链图相关特征，可以在仅提供有限的先前交易记录的情况下，自动检测与勒索软件有关的新恶意地址. 与现有的基于启发式的方法相比，Akcora 等人[97] 提出的方法在勒索软件交易检测的精度和召回率方面有显著的改进，并且可以用于自动化勒索软件检测. Raheem 等人[98] 调查收集了 10 个近期使用的比特币作为赎金支付方式的勒索软件家族，包括 CryptoLocker, CryptoDefense, CryptoWall, DMA Locker, WannaCry, CryptoTorLocker2015, TeslaCrypt, Jigsaw, ZCrypto, VenusLocker，结合聚类模型和区块链的信息识别，收集和分析了用户的比特币地址.

同样采用聚类方法的还有 Han 等人[99] 和 Wang 等人[100] 的工作，他们都使用了一种细粒度的地址聚类的方法挖掘地址与其所有者之间的关系，缓解了普通聚合方式存在的过度聚合的问题. Han 等人参考宏观经济学中的活动分类对比特币各类行业进行了定义，他们将比特币活动分为 5 个行业：1)暗网走私. 通过比特币进行走私或非法服务交易，例如 SilkRoad. 2)交易所. 在比特币和其他货币之间进行交易，例如 Mt. Gox. 3)赌博. 即使用比特币进行赌博，例如 Satoshi-Dice. 4)投资. 提供比特币回报和管理服务，包括比特币借贷款(如 Nexo)和钱包管理(如 Trezor). 5)矿工. 参加矿池通过生成新的区块获得奖励. 根据比特币用户在行业中的活动目的和模式，行业成员分别被标记为暗网供应商/客户、交易所卖家/买家、博彩发起者/赌徒、商人/投资者和矿池组织者/矿工. Han 等人根据聚类结果基于时间网络训练了一个多标签分类模型来识别动态变化的用户行业身份，从而识别某些违法行为. Wang 等人对 2012—2021 年比特币的赎金支付、赎金转移和受害者迁移进行了大规模分析，分析了赎金在不同行业之间转移的轨迹并跟踪受害者在不同行业的迁移，发现勒索软件犯罪分子常将赎金分散到多个行业的事实.

4 未来研究方向

Web 3.0 被认为是下一代的互联网，区块链技术的发展使 Web 3.0 的构想成为现实. 以区块链为基础设施的数字资产和去中心化应用不断拓宽应用领域的边界，促进了社会经济方方面面的发展. 可以预见，未来将有更多的 DApp 被部署上链，越来越多的用户会参与其中，也将发展形成全新的商业模式. 保障 Web 3.0 生态下数字经济的健康发展，就需要不断强化风险感知技术，突破现有研究的局限. 本文将未来研究方向总结为 4 个方面：

1)研究复杂场景下的智能合约语义理解和交易行为分析方法. 随着区块链技术在更广泛的应用场景中得到采用，智能合约的复杂性和交易规模将进一步增加，因此需要更加精准、高效的分析方法来确保智能合约的正确性、安全性和可靠性. Web 3.0 项目的底层逻辑是部署在区块链平台上的一组智能合约，数字资产的所有权转移通过智能合约的调用完成. 理解智能合约的功能和语义，可以有效识别 Web 3.0 应用中的经济活动，并进一步分析出其中可能存在的风险. 然而，现有的部署在公链上的智能合约，只有少数合约的源代码是公开的，大部分合约仅以字节码的形式存在，这给合约语义理解和功能分类带来挑战. 现有的对智能合约字节码的分析工作更多地聚焦于智能合约安全漏洞的检测，但是安全漏洞的模式比较单一，只会涉及到一些跟转账相关的关键代码片段，并不关心智能合约所描述的功能和业务逻辑. 而现有的一些侧重于智能合约功能理解的工作直接以原生的合约字节码作为输入进行特征提取和分类，但由于原生的字节码无法准确反映程序执行逻辑和实际执行状况，这种以字节码直接作为输入的方式存在分类精确度低、解释困难和鲁棒性不高等问题.

在带有时序特征和跨链交易的复杂场景下进行 Web 3.0 交易行为分析将是未来的重要研究方向. 现有的区块链交易行为分析技术通常受限于传统的机器学习方法，无法充分利用丰富的结构化时序信息，如合约代码中的函数、变量、参数，以及金融风险特有的一些语义和时序特征信息，因而在复杂场景和海量智能合约场景下往往会导致高误报率和漏报率. 此外，在真实的应用场景中还存在更为复杂的跨链交易行为，这给区块链数字金融风险监测带来了更大的挑战. 一方面，不同链上的交易数据结构、加密算法、智能合约虚拟机等均有差异，异构的数据与代码在处理和存储上往往存在较大差异，从而给上层分析模型带来较大的数据访问开销；另一方面，跨链交易涉及多个区块链系统，经过跨链桥对多条链上的交易进行连接并完成有效的账户识别和追踪是一项非常有挑战的任务.

2)利用多模态链上、链下数据提升风险感知技术的性能. 区块链作为 Web 3.0 的基础设施，承载了大部分 Web 3.0 项目的代码与数据，因而对 Web 3.0 数字经济体系的风险感知主要着力于对区块链平台进行数据分析和实时监测. 随着 Web 3.0 相关话题在社交媒体平台的热议，项目官网、个人媒体等产生了大量与数字资产、DApp 相关的链下数据，其中蕴含了大量信息可以辅助 Web 3.0 数字经济的风险感知. 甚至从某种程度上可以认为，丰富的链下信息对链上数字资产和经济活动产生了可预测的影响. 因此，结合链上链下、数据进行风险感知是提高监管能力的一种有效手段，目前还未见有相关的研究工作.

链下的数据往往是多模态的，包括文字、数字、代码块、图片等，这些数据来源众多、形式各异，给数据分析工作带来一些挑战. 首先，多模态数据的获取、清洗、标注、融合等数据预处理操作是一个耗时

耗力的大工程，特别是在多模态数据的标注方面，Web 3.0 数字经济作为一种新兴的经济形态，在去中心化的业务部署模式下缺少权威的领域专家，这导致高质量的标注数据集很少，标签的准确性和精确程度对后续数据分析也有较大影响. 其次，多模态异构的数据给风险模式识别与检测模型训练也带来不小的挑战，训练过程中多模态数据表征学习模型的泛化能力弱，对标签数据平衡性较为敏感，且对多模态数据的访问也影响模型训练的效率. 最后，如何在缺少信任的网络环境下，利用链上链下多模态信息对数据分析结果的有效性进行验证，也是一个需要考虑的问题. 这是因为，对于涉及到金融欺诈的项目，链下多媒体数据所反映的信息和链上智能合约的实际执行逻辑可能不一致，甚至是矛盾的，这种情况下链下信息对于智能合约功能的准确理解反倒起到了反作用. 综上所述，实现对多模态链下数据的有效利用是一件重要且有挑战的事情.

3）设计面向 Web 3.0 生态的高效数据存储引擎以支撑大规模异构数据分析. 为区块链分析任务设计和开发高效的存储引擎并提供高效的数据查询服务一直是区块链数据管理领域研究的热点，目前学术界关注可验证、高效、链上链下混合的区块链查询和分析技术，工业界开始研发链下数据管理方式以拓展面向区块链数据的查询功能. 但是，不管哪种方式，数据的底层存储均采用关系型的数据库模型，存在数据解析速度慢、实体之间关系抽取效率低、数据表达能力有限的问题. 而方便进行高效表达的图数据库，不管是采用基于节点的文件存储、基于 B+树的存储，还是基于 LSM-树的存储，也都存在检索多版本的区块链时序数据效率不高，且不支持任意时间窗口的即席（ad-hoc）数据查询的问题.

对于包含百万级以上智能合约的 Web 3.0 生态，其所涉及的数据形态多样、数据量大、数据之间的关系复杂，目前还没有一款存储引擎可以高效地支撑此类大规模异构数据的分析. 具体来说，现有的对智能合约的代码逻辑的抽取主要是通过从字节码或操作码中抽取程序块及其逻辑关系从而构建智能合约的控制流图（control flow graph, CFG）. 对于部署在区块链上的百万级以上的智能合约，实时生成 CFG 并进行分析是不现实的，而在代码量具有一定规模的合约中，构建出的 CFG 往往节点数量规模大且跳转关系复杂，给 CFG 的存储和查询造成巨大负担. 为了降低 CFG 的存储代价，需要研究高效的 CFG 存储机制. 为了快速地找到智能合约所在的 CFG，以及 CFG 对应的代码块，也需要设计针对操作码的高效索引机制. 除了需要分析智能合约代码和链下推广信息以外，还需要追踪数字资产流向并识别数字资产的转移模式，因而对交易执行时序图的分析是必不可少的. 然而，对交易执行时序图的高效访问会直接影响到分析的实时性反馈. 区块链交易数据是以区块的方式成批追加的，要求存储引擎在满足比较好的写性能的基础上也要有比较好的读性能，并且对基于时间窗口的时序数据访问友好，而现有的基于 LSM-tree 和 B-tree 的存储引擎的图数据库均不能同时满足上述需求.

4）针对开放联盟链建立适合的激励机制以保证国内 Web 3.0 数字经济的活力. 开放联盟链是在联盟链节点准入机制的基础上允许用户自由部署或使用去中心化项目的区块链技术. 与公链系统相比，开放联盟链无代币激励，而是采用由人民币兑换的燃料计价方式，而且其节点的接入仍然受到准入机制的限制，这在一定程度上保证了基础设施层的安全性；与传统的联盟链相比，开放联盟链的用户不再局限于联盟内部，任何用户都可通过区块链服务网络平台创建链账户，参与部署调用智能合约，因此具有更大的开放性.

目前开放联盟链下数字经济发展的最大问题是未能建立完善的商业模式以实现经济发展的闭环. 这个问题导致了去中心化应用在短期内只能消耗投入的资金，不能获得期望的收益. 在这种情况下，数字经济发展缺乏活力，去中心化应用发展的可持续性面临很大的挑战. 例如，截至目前，国内运行在区块链服务网络（blockchain-based service network）上的去中心化应用项目仅剩十几个，较多的项目已经关停. 与国外的去中心化应用项目相比，国内的开放联盟链缺乏赋能去中心化应用的激励机制. 在充分考虑到不同参与者的需求和利益后，建立合理的激励方式来促进他们的参与和贡献是促进去中心化应用可持续发展的重要途径. 然而，国内的法律和监管环境对于激励机制的设计和实施可能存在限制，联盟的不同参与方之间的动机和利益可能存在差异，性能瓶颈、数据隐私、公平性等方面的问题仍然备受关注，这些都是开放联盟链中 Web 3.0 数字经济发展所面临的挑战.

5 总　　结

Web 3.0 数字经济体系主要包括数字资产和去中

心化应用(DApp),其中数字资产的所有权及其变更由区块链维护,DApp的核心是运行在区块链上的智能合约,区块链因此被认为是Web 3.0数字经济的基础设施.尽管区块链技术降低了数字世界中人们价值交换的信任成本,但仍然存在智能合约漏洞、骗局、非法交易等风险因素,直接或间接地影响了Web 3.0数字经济的健康发展.本文从智能合约的编码、功能、应用3个层面总结了Web 3.0中数字经济风险的感知技术,主要包括:1)总结了现有工作中发现并研究的安全漏洞分类,并分4个类别介绍了现有的智能合约漏洞检测方法;2)概述了现有的智能合约骗局,根据模型训练数据的不同,分类介绍了现有的智能合约骗局识别技术;3)针对Web 3.0下4类主要的非法交易行为,分别总结了现有的检测方法及其研究现状.最后,本文在上述分析与总结的基础上,展望了Web 3.0数字经济风险感知技术未来的研究方向和可能面临的挑战.

作者贡献声明:贾金萍负责收集、整理文献,设计文章写作框架并撰写论文的主要内容;肖诗涵撰写1.3节和第2节部分;钱堃撰写第3节;杨艳琴针对第3节提出指导意见并修改论文;张召针对全文提出指导意见并修改论文.

参 考 文 献

[1] Shannon V. A more revolutionary Web[EB/OL]. (2006-03-01)[2023-03-21].https://www.nytimes.com/2006/05/23/technology/23iht-web.html

[2] Berners-Lee T, Hendler J, Lassila O. The semantic web[J]. Scientific American, 2001, 284(5): 34−43

[3] MacIntyre B, Smith T F. Thoughts on the future of WebXR and the immersive Web[C]//Proc of the 2018 IEEE Int Symp on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct). Piscataway, NJ: IEEE, 2018: 338−342

[4] Wood G. Dapps: What Web 3.0 looks like[EB/OL]. (2014-04-17)[2023-03-15].https://gavwood.com/dappsweb3.html

[5] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[EB/OL].[2023-03-20]. http://www.bitcoin.org/bitcoin.pdf

[6] Buterin V. A next-generation smart contract and decentralized application platform[EB/OL]. (2018-02-08)[2023-03-15].https://ethereum.org/en/whitepaper/

[7] The State Information Center. BSN open permissioned blockchain[EB/OL].[2023-03-18]. https://opb.bsnbase.com/main/index (in Chinese)
(国家信息中心. BSN开放联盟链[EB/OL].[2023-03-18]. https://opb.bsnbase.com/main/index)

[8] The Maker Foundation. MakerDAO[EB/OL].[2023-03-18].https://makerdao.com/zh-CN/ (in Chinese)
(Maker基金会. MakerDAO[EB/OL].[2023-03-18].https://makerdao.com/zh-CN/)

[9] Siegel D. The DAO attack: Understanding what happened[EB/OL].(2016-06-25)[2023-03-26].https://www.coindesk.com/learn/understanding-the-dao-attac/

[10] Reuters. Bitcoin exchange of Hong Kong suffers from hacker theft and losses of approximately $72 million[EB/OL]. (2016-08-03)[2023-03-26].https://www.reuters.com/article/bitcoin-hacking-updates-0803-idCNKCS10E10I

[11] United States Department of the Treasury. U. S. treasury sanctions notorious virtual currency mixer Tornado. Cash[EB/OL]. (2022-08-08)[2023-03-28].https://home.treasury.gov/news/press-releases/jy0916

[12] United States Securities and Exchange Commission. SEC charges eleven individuals in $300 million crypto pyramid scheme[EB/OL].(2022-08-01)[2023-03-28].https://www.sec.gov/news/press-release/2022-134

[13] The Ethereum Foundation. Decentralized application (DAPPS)[EB/OL].[2023-02-28].https://ethereum.org/zh/dapps/#what-are-dapps (in Chinese)
(以太坊基金会. 去中心化应用(DAPPS)[EB/OL].[2023-02-28].https://ethereum.org/zh/dapps/#what-are-dapps)

[14] Chen Chuan, Zhang Lei, Li Yihao, et al. When digital economy meets web 3.0: Applications and challenges[J]. IEEE Open Journal of the Computer Society, 2022, 3: 233−245

[15] Huang Huawei, Kong Wei, Zhou Sicong, et al. A survey of state-of-the-art on blockchains: Theories, modelings, and tools[J]. ACM Computing Surveys, 2021, 54(2): 1−42

[16] Wu Jiajing, Lin Kaixin, Lin Dan, et al. Financial crimes in Web3-empowered metaverse: Taxonomy, countermeasures, and opportunities[J]. IEEE Open Journal of the Computer Society, 2023, 4: 37−49

[17] Chen Weili, Zheng Zibin. Blockchain data analysis: A review of status, trends and challenges[J]. Journal of Computer Research and Development, 2018, 55(9): 1853−1870 (in Chinese)
(陈伟利,郑子彬. 区块链数据分析:现状、趋势与挑战[J]. 计算机研究与发展, 2018, 55(9):1853−1870)

[18] Wei Songjie, Lü Weilong, Li Shasha. Overview on typical security problems in public blockchain applications[J]. Journal of Software, 2022, 33(1): 324−355 (in Chinese)
(魏松杰,吕伟龙,李莎莎. 区块链公链应用的典型安全问题综述[J]. 软件学报, 2022, 33(1):324−355)

[19] Tolmach P, Li Yi, Lin Shangwei, et al. A survey of smart contract formal specification and verification[J]. ACM Computing Surveys, 2021, 54(7): 1−38

[20] Qian Peng, Liu Zhenguang, He Qinming, et al. Smart contract vulnerability detection technique: A survey[J]. Journal of Software, 2022, 33(8): 3059−3085 (in Chinese)
(钱鹏,刘振广,何钦铭,等. 智能合约安全漏洞检测技术研究综

述[J]. 软件学报, 2022, 33(8): 3059−3085)

[21] Cui Siwei, Zhao Gang, Gao Yifei, et al. VRust: Automated vulnerability detection for Solana smart contracts[C]//Proc of the 2022 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2022: 639−652

[22] He Ningyu, Zhang Ruiyi, Wang Haoyu, et al. EOSAFE: Security analysis of EOSIO smart contracts[C]//Proc of the 30th USENIX Security Symp. Berkeley, CA: USENIX Association, 2021: 1271−1288

[23] Luu L, Chu D H, Olickel H, et al. Making smart contracts smarter[C]//Proc of the 2016 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2016: 254−269

[24] Krupp J, Rossow C. TEEHER: Gnawing at Ethereum to automatically exploit smart contracts[C]//Proc of the 27th USENIX Security Symp (USENIX Security 18). Berkeley, CA: USENIX Association, 2018: 1317−1333

[25] Bose P, Das D, Chen Yanju, et al. Sailfish: Vetting smart contract state-inconsistency bugs in seconds[C]//Proc of the 2022 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2022: 161−178

[26] Ye Jiaming, Ma Mingliang, Lin Yun, et al. Vulpedia: Detecting vulnerable Ethereum smart contracts via abstracted vulnerability signatures[J]. Journal of Systems and Software, 2022, 192: 111410

[27] Liao Zeqin, Zheng Zibin, Chen Xiao, et al. SmartDagger: A bytecode-based static analysis approach for detecting cross-contract vulnerability[C]//Proc of the 31st ACM SIGSOFT Int Symp on Software Testing and Analysis. New York: ACM, 2022: 752−764

[28] Zheng Peilin, Zheng Zibin, Luo Xiapu. Park: Accelerating smart contract vulnerability detection via parallel-fork symbolic execution[C]//Proc of the 31st ACM SIGSOFT Int Symp on Software Testing and Analysis. New York: ACM, 2022: 740−751

[29] Rodler M, Li Wenting, Karame G O, et al. EVMPatch: Timely and automated patching of Ethereum smart contracts[C]//Proc of the 30th USENIX Security Symp. Berkeley, CA: USENIX Association, 2021: 1289−1306

[30] Nguyen T D, Pham L H, Sun Jun. SGUARD: Towards fixing vulnerable smart contracts automatically[C]//Proc of the 2021 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2021: 1215−1229

[31] Rossini M, Zichichi M, Ferretti S. Smart contracts vulnerability classification through deep learning[C]//Proc of the 20th ACM Conf on Embedded Networked Sensor Systems. New York: ACM, 2022: 1229−1230

[32] Gao Zhipeng, Jayasundara V, Jiang Lingxiao, et al. Smartembed: A tool for clone and bug detection in smart contracts through structural code embedding[C]//Proc of the 2019 IEEE Int Conf on Software Maintenance and Evolution (ICSME). Piscataway, NJ: IEEE, 2019: 394−397

[33] Qian Peng, Liu Zhenguang, He Qinming, et al. Towards automated reentrancy detection for smart contracts based on sequential models[J]. IEEE Access, 2020, 8: 19685−19695

[34] Huang Jianjun, Han Songming, You Wei, et al. Hunting vulnerable smart contracts via graph embedding based bytecode matching[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 2144−2156

[35] Nguyen H H, Nguyen N M, Doan H P, et al. MANDO-GURU: Vulnerability detection for smart contract source code by heterogeneous graph embeddings[C]//Proc of the 30th ACM Joint European Software Engineering Conf and Symp on the Foundations of Software Engineering. New York: ACM, 2022: 1736−1740

[36] Zhuang Yuan, Liu Zhenguang, Qian Peng, et al. Smart contract vulnerability detection using graph neural network[C]//Proc of the 29th Int Joint Conf on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 2020: 3283−3290

[37] Wu Hongjun, Zhang Zhuo, Wang Shangwen, et al. Peculiar: Smart contract vulnerability detection based on crucial data flow graph and pre-training techniques[C]//Proc of the 32nd IEEE Int Symp on Software Reliability Engineering (ISSRE). Piscataway, NJ: IEEE, 2021: 378−389

[38] Zhang Lejun, Wang Jinlong, Wang Weizheng, et al. Smart contract vulnerability detection combined with multi-objective detection[J]. Computer Networks, 2022, 217: 109289

[39] Cai Jie, Li Bin, Zhang Jiale, et al. Combine sliced joint graph with graph neural networks for smart contract vulnerability detection[J]. Journal of Systems and Software, 2023, 195: 111550

[40] Jiao Jiao, Kan Shuanglong, Lin Shangwei, et al. Semantic understanding of smart contracts: Executable operational semantics of solidity[C]//Proc of the 2020 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2020: 1695−1712

[41] Permenev A, Dimitrov D, Tsankov P, et al. VerX: Safety verification of smart contracts[C]//Proc of the 2020 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2020: 1661−1677

[42] Schneidewind C, Grishchenko I, Scherer M, et al. eThor: Practical and provably sound static analysis of Ethereum smart contracts[C]// Proc of the 2020 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2020: 621−640

[43] Frank J, Aschermann C, Holz T. ETHBMC: A bounded model checker for smart contracts[C]//Proc of the 29th USENIX Conf on Security Symp. Berkeley, CA: USENIX Association, 2020: 2757−2774

[44] So S, Lee M, Park J, et al. VeriSmart: A highly precise safety verifier for Ethereum smart contracts[C]//Proc of the 2020 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2020: 1678−1694

[45] Samreen N F, Alalfi M H. SmartScan: An approach to detect denial of service vulnerability in Ethereum smart contracts[C]//Proc of the 4th IEEE/ACM Int Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). Piscataway, NJ: IEEE, 2021: 17−26

[46] Liu Zhenguang, Qian Peng, Qang Xiang, et al. Smart contract vulnerability detection: From pure neural network to interpretable graph feature and expert pattern fusion[C]//Proc of the 30th Int Joint Conf on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 2021: 2751−2759

[47] Liu Zhenguang, Qian Peng, Wang Xiaoyang, et al. Combining graph neural networks with expert knowledge for smart contract vulnerability detection[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 35(2): 1296−1310

[48] Tang D, Qin B, Feng X, et al. Effective LSTMs for target-dependent sentiment classification[J]. arXiv preprint, arXiv: 1512.01100, 2015

[49] Tian Gang, Wang Qibo, Zhao Yi, et al. Smart contract classification with a Bi-LSTM based approach[J]. IEEE Access, 2020, 8: 43806−43816

[50] Chen Yizhou, Dai Heng, Yu Xiao, et al. Improving Ponzi scheme contract detection using multi-channel TextCNN and transformer[J]. Sensors, 2021, 21(19): 6417

[51] He Xuezhi, Yang Tan, Chen Liping. CTRF: Ethereum-based Ponzi contract identification[EB/OL].[2023-03-28].https://www.hindawi.com/journals/scn/2022/1554752/

[52] Ishimaki M, Omote K. Ethereum contract honeypot risk analysis[C]// Proc of the 5th Int Conf on Frontiers in Cyber Security (FCS). Berlin: Springer, 2022: 226−240

[53] Hu Teng, Liu Xiaolei, Chen Ting, et al. Transaction-based classification and detection approach for Ethereum smart contract[J]. Information Processing & Management, 2021, 58(2): 102462

[54] Huang Butian, Liu Qi, He Qinming, et al. Towards automatic smart-contract codes classification by means of word embedding model and transaction information[J]. Acta Automatica Sinica, 2017, 43(9): 1532−1543 (in Chinese)
(黄步添, 刘琦, 何钦铭, 等. 基于语义嵌入模型与交易信息的智能合约自动分类系统[J]. 自动化学报, 2017, 43(9): 1532−1543)

[55] Jin Chengxiang, Zhou Jiajun, Jin Jie, et al. Time-aware metapath feature augmentation for Ponzi detection in Ethereum[J]. arXiv preprint, arXiv: 2210.16863, 2022

[56] Camino R, Torres C F, Baden M, et al. A data science approach for honeypot detection in Ethereum[J]. arXiv preprint, arXiv: 1910.01449, 2019

[57] Fan Shuhui, Fu Shaojing, Luo Yuchuan, et al. Smart contract scams detection with topological data analysis on account interaction[C]// Proc of the 31st ACM Int Conf on Information & Knowledge Management. New York: ACM, 2022: 468−477

[58] Wang Lei, Cheng Hao, Zheng Zibin, et al. Ponzi scheme detection via oversampling-based long short-term memory for smart contracts[J]. Knowledge-Based Systems, 2021, 228: 107312

[59] Jin Jie, Zhou Jiajun, Jin Chengxiang, et al. Dual-channel early warning framework for Ethereum Ponzi schemes[C]//Proc of the 7th Big Data and Social Computing China National Conf. Berlin: Springer, 2022: 260−274

[60] Bartoletti M, Carta S, Cimoli T, et al. Dissecting Ponzi schemes on Ethereum: Identification, analysis, and impact[J]. Future Generation Computer Systems, 2020, 102: 259−277

[61] Zheng Zibing, Chen Weili, Zhong Zhijie, et al. Securing the Ethereum from smart Ponzi schemes: Identification using static features[EB/OL].[2023-03-18]. https://dl.acm.org/doi/abs/10.1145/3571847

[62] Lou Yincheng, Zhang Yanmei, Chen Shiping. Ponzi contracts detection based on improved convolutional neural network[C]//Proc of the 2020 IEEE Int Conf on Services Computing (SCC). Piscataway, NJ: IEEE, 2020: 353−360

[63] Hara K, Takahashi T, Ishimaki M, et al. Machine-learning approach using solidity bytecode for smart-contract honeypot detection in the Ethereum[C]//Proc of the 21st IEEE Int Conf on Software Quality, Reliability and Security Companion (QRS-C). Piscataway, NJ: IEEE, 2021: 652−659

[64] Shi Chaochen, Xiang Yong, Yu Jiangshan, et al. A bytecode-based approach for smart contract classification[C]//Proc of the 2022 IEEE Int Conf on Software Analysis, Evolution and Reengineering (SANER). Piscataway, NJ: IEEE, 2022: 1046−1054

[65] Lin Dan, Lin Kaixin, Wu Jiajing, et al. Bytecode-based approach for Ethereum smart contract classification[J]. Chinese Journal of Network and Information Security, 2022, 8(5): 111−120 (in Chinese)
(林丹, 林凯欣, 吴嘉婧, 等. 基于字节码的以太坊智能合约分类方法[J]. 网络与信息安全学报, 2022, 8(5): 111−120)

[66] Hu Huiwen, Bai Qianlan, Xu Yuedong. SCSGuard: Deep scam detection for Ethereum smart contracts[EB/OL].[2023-03-28]. https://ieeexplore.ieee.org/abstract/document/9798296

[67] Peng Jianxi, Xiao Guijiao. Detection of smart Ponzi schemes using opcode[C]//Proc of the 2nd Blockchain and Trustworthy Systems Int Conf. Berlin: Springer, 2020: 192−204

[68] Chen Weili, Zheng Zibin, Cui Jiahui, et al. Detecting Ponzi schemes on ethereum: Towards healthier blockchain technology[C]//Proc of the 2018 World Wide Web Conf. New York: ACM, 2018: 1409−1418

[69] Zhang Yanmei, Yu Wenqiang, Li Ziyu, et al. Detecting Ethereum Ponzi schemes based on improved LightGBM algorithm[J]. IEEE Transactions on Computational Social Systems, 2021, 9(2): 624−637

[70] Torres C F, Steichen M, State R. The art of the scam: Demystifying honeypots in Ethereum smart contracts[J]. arXiv preprint, arXiv: 1902.06976, 2019

[71] Nalinipriya G, Balajee M, Priya C, et al. Ransomware recognition in blockchain network using water moth flame optimization-aware DRNN[J]. Concurrency and Computation: Practice and Experience, 2022, 34(19): e7047

[72] Sun Weisong, Xu Guangyao, Yang Zijiang, et al. Early detection of smart Ponzi scheme contracts based on behavior forest similarity[C]//Proc of the 20th IEEE Int Conf on Software Quality, Reliability and Security (QRS). Piscataway, NJ: IEEE, 2020: 297−309

[73] Chen Weimin, Li Xinran, Sui Yuting, et al. SADPonzi: Detecting and characterizing Ponzi schemes in Ethereum smart contracts[J]. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 2021, 5(2): 1−30

[74] Sun Xun, Lin Xingwei, Liao Zhou. An ABI-based classification approach for Ethereum smart contracts[C]//Proc of the 2021 IEEE Int Conf on Dependable, Autonomic and Secure Computing, Int Conf on Pervasive Intelligence and Computing, Int Conf on Cloud and Big Data Computing, Int Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). Piscataway, NJ:

IEEE, 2021: 99−104

[75] Fan Shuhui, Fu Shaojing, Xu Haoran, et al. Expose your mask: Smart Ponzi schemes detection on blockchain[EB/OL].[2023-03-24]. https://ieeexplore.ieee.org/abstract/document/9207143

[76] Fan Shuhui, Fu Shaojing, Xu Haoran, et al. Al-SPSD: Anti-leakage smart Ponzi schemes detection in blockchain[J]. Information Processing & Management, 2021, 58(4): 102587

[77] Möser M, Böhme R, Breuker D. An inquiry into money laundering tools in the bitcoin ecosystem[C]//Proc of the 2013 Symp on Electronic Crime Research (2013 APWG eCrime). Piscataway, NJ: IEEE, 2013: 1−14

[78] Wu Jiajing, Liu Jieli, Chen Weili, et al. Detecting mixing services via mining bitcoin transaction network with hybrid motifs[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 52(4): 2237−2249

[79] Meiklejohn S, Pomarole M, Jordan G, et al. A fistful of bitcoins: Characterizing payments among men with no names[C]//Proc of the 2013 Conf on Internet Measurement. New York: ACM, 2013: 127−140

[80] Ketenci U G, Kurt T, Önal S, et al. A time-frequency based suspicious activity detection for anti-money laundering[J]. IEEE Access, 2021, 9: 59957−59967

[81] Lorenz J, Silva M I, Aparício D, et al. Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity[C]//Proc of the 1st ACM Int Conf on AI in Finance. New York: ACM, 2020: 23: 1−23: 8

[82] Alarab I, Prakoonwit S. Graph-based LSTM for anti-money laundering: Experimenting temporal graph convolutional network with bitcoin data[J]. Neural Processing Letters, 2022, 55(1): 689−707

[83] Chen Weili, Guo Xiongfen, Chen Zhiguang, et al. Phishing scam detection on Ethereum: Towards financial security for blockchain ecosystem[C]//Proc of the 29th Int Joint Conf on Artificial Intelligence (IJCAI). San Francisco: Margan Kaufmann, 2020: 4506−4512

[84] Wu Jiajing, Yuan Qi, Lin Dan, et al. Who are the phishers? Phishing scam detection on Ethereum via network embedding[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 52(2): 1156−1166

[85] Kabla A H H, Anbar M, Manickam S, et al. Eth-PSD: A machine learning-based phishing scam detection approach in Ethereum[J]. IEEE Access, 2022, 10: 118043−118057

[86] Wen Tingke, Xiao Yuanxing, Wang Anqi, et al. A novel hybrid feature fusion model for detecting phishing scam on Ethereum using deep neural network[J]. Expert Systems with Applications, Amsterdam: Elsevier, 2023, 211: 118463

[87] Ron D, Shamir A. Quantitative analysis of the full bitcoin transaction graph[C]//Proc of the 17th Financial Cryptography and Data Security Int Conf (FC 2013). Berlin: Springer, 2013: 6−24

[88] Zhao Lin, Sen G S, Khan A, et al. Temporal analysis of the entire Ethereum blockchain network[C]//Proc of the 2021 Web Conf. New York: ACM, 2021: 2258−2269

[89] Chen Liang, Peng Jiaying, Liu Yang, et al. Phishing scams detection in Ethereum transaction network[J]. ACM Transactions on Internet Technology, 2020, 21(1): 1−16

[90] Li Sijia, Gou Gaopeng, Liu Chang, et al. TTAGN: Temporal transaction aggregation graph network for ethereum phishing scams detection[C]//Proc of the 2022 ACM Web Conf. New York: ACM, 2022: 661−669

[91] Fu Bingxue, Yu Xing, Feng Tao. CT-GCN: A phishing identification model for blockchain cryptocurrency transactions[J]. International Journal of Information Security, 2022, 21(6): 1223−1232

[92] Duan Xincheng, Yan Biwei, Dong Anming, et al. Phishing frauds detection based on graph neural network on Ethereum[C]//Proc of the 17th Int Conf on Wireless Algorithms, Systems, and Applications (WASA 2022). Berlin: Springer, 2022: 351−363

[93] Eigelshoven F, Ullrich A, Parry D A. Cryptocurrency market manipulation: A systematic literature review[EB/OL].[2023-03-28].https://aisel.aisnet.org/icis2021/fintech/fintech/1/

[94] Chen Weili, Wu Jun, Zheng Zibin, et al. Market manipulation of bitcoin: Evidence from mining the Mt. Gox transaction network[C]// Proc of the 2019 IEEE INFOCOM Conf on Computer Communications. Piscataway, NJ: IEEE, 2019: 964−972

[95] Pereira D M, Couto R S. Using degree centrality to identify market manipulation on bitcoin[C]//Proc of the 2021 Data Privacy Management, Cryptocurrencies and Blockchain Technology: ESORICS Int Workshops (DPM 2021 and CBT 2021). Berlin: Springer, 2022: 208−223

[96] Huang D Y, Aliapoulios M M, Li V G, et al. Tracking ransomware end-to-end[C]//Proc of the 2018 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2018: 618−631

[97] Akcora C G, Li Yitao, Gel Y R, et al. Bitcoinheist: Topological data analysis for ransomware detection on the bitcoin blockchain[J]. arXiv preprint, arXiv: 1906.07852, 2019

[98] Raheem A, Raheem R, Chen T M, et al. Estimation of ransomware payments in bitcoin ecosystem[C]//Proc of the 2021 IEEE Int Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/ SustainCom). Piscataway, NJ: IEEE, 2021: 1667−1674

[99] Han Weili, Chen Dingjie, Pang Jun, et al. Temporal networks based industry identification for bitcoin users[C]//Proc of the 16th Int Conf on Wireless Algorithms, Systems, and Applications (WASA 2021). Berlin: Springer, 2021: 108−120

[100] Wang Kai, Pang Jun, Chen Dingjie, et al. A large-scale empirical analysis of ransomware activities in bitcoin[J]. ACM Transactions on the Web, 2021, 16(2): 1−29

Jia Jinping, born in 1996. PhD candidate. Her main research interests include blockchain, distributed databases, and location-based service.

贾金萍, 1996 年生. 博士研究生. 主要研究方向为区块链、分布式数据库、基于位置的服务.

Xiao Shihan, born in 2000. Master candidate. Her main research interests include blockchain and distributed databases.

肖诗涵, 2000 年生. 硕士研究生. 主要研究方向为区块链、分布式数据库.

Qian Kun, born in 1999. Master candidate. His main research interests include blockchain and distributed databases.

钱　堃, 1999 年生. 硕士研究生. 主要研究方向为区块链、分布式数据库.

Yang Yanqin, born in 1977. PhD, associate professor. Senior member of CCF. Her main research interests include compilation optimization, embedded system, and blockchain technology.

杨艳琴, 1977 年生. 博士，副教授. CCF 高级会员. 主要研究方向为编译优化、嵌入式系统、区块链技术.

Zhang Zhao, born in 1977. PhD, professor, PhD supervisor. Senior member of CCF. Her main research interests include distributed databases and blockchain data management.

张　召, 1977 年生. 博士，教授，博士生导师，CCF 高级会员. 主要研究方向为分布式数据库、区块链数据管理.