软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007698] [CSTR: 32375.14.jos.007698]

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

基于图与超图的以太坊智能合约庞氏骗局实时识别框架*

陈伟利[1], 竹 超[1], 肖桂姣[2], 唐明董[1]

[1](广东外语外贸大学 信息科学与技术学院, 广东 广州 510006)

[2](广东外语外贸大学南国商学院 计算机学院, 广东 广州 510545)

通信作者: 陈伟利, E-mail: mathutopia@163.com

摘　要: 随着区块链技术的快速发展, 智能合约逐渐成为金融欺诈的典型工具, 其中智能合约形式的庞氏骗局已成为区块链生态中一种典型且高风险的诈骗形式, 导致了大量的经济损失. 目前基于合约源代码、账户交易数据等抽取特征并构建机器学习模型的庞氏骗局合约识别研究取得了显著进展, 但区块链的匿名性以及交易的滞后性, 让这类方法在应用上存在明显的不足. 为突破这一瓶颈, 提出一种基于图与超图的智能合约庞氏骗局实时识别框架. 该框架基于合约部署时刻强制公开的字节码, 在合约无任何外部交易发生前, 通过提取操作码序列, 构建图与超图结构, 并结合图卷积神经网络与超图卷积神经网络, 实现“部署即检测”的实时识别能力. 在最新的数据集上的实验结果表明, 该框架在检测准确率和 $F1$ 分数方面显著优于传统方法. 这表明该框架能在不依赖源代码和交互数据的情况下, 实现对庞氏骗局智能合约的深层次特征抽取, 为欺诈合约的“事后追溯”转为“事前审计”提供了有效支持.

关键词: 区块链; 智能合约; 庞氏骗局; 以太坊; 图卷积神经网络; 超图卷积神经网络

中图法分类号: TP311

中文引用格式: 陈伟利, 竹超, 肖桂姣, 唐明董. 基于图与超图的以太坊智能合约庞氏骗局实时识别框架. 软件学报. http://www.jos.org.cn/1000-9825/7698.htm

英文引用格式: Chen WL, Zhu C, Xiao GJ, Tang MD. Real-time Identification Framework for Ethereum Smart Contract Ponzi Schemes Based on Graphs and Hypergraphs. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7698.htm

Real-time Identification Framework for Ethereum Smart Contract Ponzi Schemes Based on Graphs and Hypergraphs

CHEN Wei-Li[1], ZHU Chao[1], XIAO Gui-Jiao[2], TANG Ming-Dong[1]

[1](School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510006, China)

[2](School of Computer, South China Business College Guangdong University of Foreign Studies, Guangzhou 510545, China)

Abstract: With the rapid development of blockchain technology, smart contracts have gradually become a common vehicle for financial fraud. Among them, Ponzi schemes in the form of smart contracts have become a typical and high-risk form of fraud in the blockchain ecosystem, resulting in significant economic losses. Recent research has made significant progress in identifying Ponzi scheme smart contracts by extracting features from contract source code and account transaction data and using them to build machine learning models. However, blockchain anonymity and transaction latency significantly limit the practical applicability of these methods. To address these issues, this study proposes a real-time identification framework for smart contract Ponzi schemes based on graphs and hypergraphs. This framework is based on the bytecode that is required to be disclosed at the time of contract deployment. Before any external transaction involving the contract occurs, the proposed framework extracts opcode sequences, constructs graph and hypergraph structures, and combines graph convolutional networks with hypergraph convolutional networks to enable real-time “detection upon deployment”. Experimental results on the latest dataset show that this framework significantly outperforms traditional methods in terms of detection

* 基金项目: 广东省基础与应用基础研究基金 (2025A1515012982)
收稿时间: 2025-07-16; 修改时间: 2025-11-25, 2026-04-09; 采用时间: 2026-05-14; jos 在线出版时间: 2026-08-19

accuracy and *F*1-score. This demonstrates that the framework can perform deep feature extraction for Ponzi scheme smart contracts without relying on source code or interaction data, providing effective support for transforming the identification of fraudulent contracts from “post-event tracing” to “pre-event auditing”.

Key words: blockchain; smart contract; Ponzi scheme; Ethereum; graph convolutional network (GCN); hypergraph convolutional network (HGCN)

1 引 言

区块链[1]是一种基于分布式账本的创新技术, 它通过去中心化的网络架构和密码学技术[2], 确保数据的透明性、安全性及不可篡改性. 每一个区块都包含一组交易数据, 并通过哈希链接与前一个区块形成链式结构, 从而使得每个区块的数据很难被篡改或删除. 区块链的去中心化特点意味着它不依赖于单一的中央权威或中介, 而是通过全网节点的共识机制达成一致, 从而确保数据的可靠性和防篡改性. 最初, 区块链技术应用于比特币[3]等加密货币的交易中, 通过去中心化的验证机制解决了传统货币体系中的信任问题. 然而, 随着技术的发展, 区块链的应用已扩展至供应链管理、医疗健康[4]、金融服务、知识产权保护[5]等多个领域, 展现了广泛的商业潜力.

在区块链中, 每个参与节点都拥有一份账本副本, 所有的交易记录都在网络中的各个节点上同步更新. 这种分布式存储和共识机制不仅消除了传统金融体系中对中心化机构的依赖, 还降低了交易成本和时间延迟, 提高了效率. 此外, 区块链引入了智能合约技术[6], 它是一种在区块链上自动执行的程序代码, 能够在满足特定条件时自动执行合约条款, 极大地增强了金融和法律协议的自动化和透明度. 智能合约的出现, 不仅简化了合同执行流程, 还为各类应用场景 (如金融衍生品、数字身份验证、物联网等) 提供了创新解决方案.

以太坊 (Ethereum)[7]是一个去中心化的区块链平台, 专注于为开发者提供构建和部署去中心化应用的基础架构. 与比特币不同, 以太坊不仅限于货币转账, 它通过其图灵完备的虚拟机——以太坊虚拟机 (Ethereum virtual machine, EVM)——支持更复杂的操作和业务逻辑. 智能合约 (smart contract) 是以太坊平台的核心组件, 作为自动执行的代码脚本, 能够在满足预定条件后自动执行交易或其他任务. 智能合约运行在 EVM 中, 允许用户预定义规则和条件, 无需中介机构的参与就可以确保协议的自动执行. 这使得以太坊成为一种去信任的协议工具, 适用于从金融交易、资产管理到供应链、身份验证等各类场景. 智能合约的部署过程透明且不可篡改, 确保了执行的公正性和可靠性. 然而, 一旦合约部署在区块链上, 代码无法修改, 这也带来了相应的安全挑战, 如合约中可能存在的漏洞或恶意逻辑[6].

以太坊的智能合约推动了去中心化金融 (DeFi)[8]的快速发展, 但也成为欺诈者的温床, 其中最典型的便是庞氏骗局[9]. 这类合约以高额回报为诱饵, 实质上利用新投资者的资金向早期参与者支付收益, 一旦资金链断裂, 整个骗局便会崩溃, 导致投资者遭受巨大损失. 智能合约的去中心化和不可篡改特性不仅提高了金融交易的效率, 也为欺诈行为提供了掩护, 使得追踪和监管更加困难. 庞氏骗局智能合约[10]通常通过隐藏或混淆源代码增加审计难度, 并利用资金流动的不透明性制造虚假收益假象, 而区块链的匿名性进一步加大了身份验证和资金追踪的难度, 使欺诈行为得以隐蔽扩张. 相较于传统庞氏骗局, 这类智能合约因自动化执行、匿名化运作和不可篡改的特性, 使投资者更难察觉风险, 同时欺诈者能够借助区块链技术逃避监管. 因此, 在合约部署阶段构建有效的检测技术, 对保护投资者利益、维护区块链生态安全至关重要.

以往的庞氏骗局检测研究主要依赖合约执行后的交易数据[11]、合约源代码[12]等构建特征来建立识别模型. 然而, 这些方法存在明显的局限性: 1) 交易数据滞后性. 交易数据通常是在智能合约运行一段时间后积累出来的. 这样一来, 投资者往往在庞氏骗局的初期就面临较大的风险, 尤其是在欺诈行为尚未暴露时, 参与交易的大部分投资者的资金已遭受损失; 2) 源代码不可见性: 在以太坊等去中心化平台上, 绝大多数智能合约的源代码并不公开. 这使得基于源代码的静态分析方法、特征抽取方法无法应用, 传统的通过源代码特征检测庞氏骗局的方法变得无效或难以实现.

为了克服上述局限性, 本文提出智能合约庞氏骗局实时识别问题, 即在智能合约上线时, 仅依赖必须提供的字

节码[13]、不依赖源代码和滞后的交易数据提取关键特征, 在合约发生交易前进行庞氏骗局合约识别. 这一问题的解决将在合约部署的初期阶段实现潜在的欺诈行为识别, 将欺诈合约的“事后追溯”式的检测问题, 转化为“事前审计”的范式, 从而有效提升对投资者的保护. 然而, 字节码的特征抽取与识别建模仍然面临诸多挑战. 首先, 字节码高度抽象, 缺乏清晰的结构划分, 导致其难以直接解读. 同时, 与源代码相比, 字节码并不包含函数或类等高级结构, 使得直接理解其功能逻辑变得更加复杂. 此外, 字节码往往包含大量与合约功能无关的信息. 如何提取有效特征, 并构建性能良好的识别模型, 是本研究需要解决的重要问题.

为应对这些挑战, 本文提出首先将字节码转化为操作码, 进而利用操作码语义构建各类图结构实现特征抽取. 具体而言, 我们首先将智能合约字节码转化为操作码序列, 并构建 3 种不同的图结构: 顺序图、控制流图和超图. 通过这些图结构, 全面刻画了智能合约中操作码的执行顺序、控制流跳转关系及合约内部的复杂逻辑依赖, 为后续的深度学习模型提供了丰富的结构化输入特征, 使得我们能够从更高维度、更深层次地理解和分析合约的功能. 在此基础上, 我们结合图卷积神经网络 (graph convolutional network, GCN)[14]和超图卷积神经网络 (hypergraph convolutional network, HGCN) 进一步提取操作码图中的深层特征, 挖掘合约中的潜在风险信号. 图和超图网络能够充分利用图结构的邻接关系, 捕捉到合约内部节点之间的复杂依赖关系, 从而提升庞氏骗局检测的准确性和鲁棒性. 本文的主要贡献如下.

(1) 面向部署阶段的仅字节码实时检测范式: 针对现有方法依赖源代码或交易数据的局限, 明确将庞氏骗局识别任务限定于合约部署时刻, 仅利用强制公开的字节码进行风险判定, 实现了在无任何外部交互发生前的准实时预警, 显著增强了方法的适用性与事前监管能力.

(2) 基于 3 层图结构的语义化字节码表征方法: 提出从操作码序列出发, 依次构建顺序图、控制流图和超图的结构化建模流程, 分别刻画指令时序、程序控制流和高阶语义关联, 为深度学习提供更丰富的结构化输入.

(3) 融合 GCN 与 HGCN 的集成识别模型 MG-DAF: 设计一种多图协同学习框架, 联合利用 GCN 和 HGCN 分别处理不同层级的图结构, 并通过集成策略融合预测结果, 在真实大规模数据集上显著提升了庞氏骗局检测的准确率、召回率与鲁棒性.

本文第 2 节总结相关工作. 第 3 节详细描述模型的设计和构造. 第 4 节展示实验研究和结果. 第 5 节进行讨论, 总结并提出未来的研究方向.

2 相关工作

在区块链领域, 欺诈检测[15]一直是研究的热点问题. 其中, 庞氏骗局与智能合约的结合, 为以太坊生态系统带来了新的欺诈形式, 并对其安全性和可信度产生了严重威胁. 当前的智能合约庞氏骗局欺诈检测研究主要集中在智能合约源代码、账户交易数据以及字节码的分析上, 旨在通过不同的技术手段识别和预防庞氏骗局.

第 1 类研究基于源代码检查, 基于源代码检查的研究方法主要通过分析智能合约的控制逻辑来识别潜在的庞氏骗局. 早期研究依赖于人工检查合约源代码, 例如, Chen 等人[16]通过分析智能合约源代码逻辑, 研究了 4 种庞氏骗局合约的源代码级别模式, 提出了基于源代码特征的检测方法. Bartoletti 等人[17]则通过手动分析开源代码, 建立了将合约分类为庞氏骗局的标准模型. 这类方法的优点在于能够直接从源代码层面评估智能合约的逻辑, 但其局限性在于需要大量人力资源, 且只能识别与已知庞氏骗局模式相似的合约. 为克服人工检查的低效性, 自动化工具逐渐被引入. Lu 等人[18]提出的 SourceP 方法是一种基于预训练模型和数据流分析的检测技术, 仅依赖智能合约的源代码特征即可高效识别庞氏骗局行为, 显著减少了人工干预. 尽管如此, 这类方法仍面临一个关键问题: 大多数智能合约的源代码并未公开, 限制了其实际应用范围.

第 2 类研究基于特征工程, 随着机器学习技术的发展, 基于特征工程的庞氏骗局检测方法逐渐成为主流. 这类方法通过从智能合约的源代码、交易数据和字节码中提取特征, 并利用机器学习模型进行分类, 从而识别潜在的庞氏骗局行为. Chen 等人[19,20]从交易数据和智能合约字节码中提取了账户特征和操作码特征, 构建了基于机器学习的分类模型, 能够较为准确地识别庞氏骗局. Wang 等人[21,22]则提出了综合考虑账户特征和代码特征的方案, 结

合合成少数类过采样技术 (SMOTE) 和长短期记忆 (LSTM) 网络模型, 有效解决了数据样本不平衡问题, 提高了检测的准确性和鲁棒性. Zhang 等人[23]进一步创新性地引入了字节码特征, 并利用改进的 LightGBM (轻量级梯度提升机) 算法, 显著提升了庞氏骗局检测的效率和精度. 此外, Wang 等人[24]通过操作码上下文分析和自适应提升 (AdaBoost) 算法, 提出了一种新的以太坊庞氏骗局检测方案. 该方案通过对操作码上下文关系的细致分析, 进一步提高了对智能合约中潜在庞氏骗局的识别能力. Ji 等人[25]则提出了一种基于控制流图 (CFG) 的机器学习方法, 通过从智能合约中提取控制流特征来识别庞氏骗局, 为检测方法提供了新的视角. Zheng 等人[26]在 ACM TOSEM 上系统研究了仅依赖字节码的庞氏骗局检测框架, 不仅融合 *n*-gram 操作码序列的 TF-IDF 与 Word2Vec 语义嵌入, 还创新性地引入“创作者向量 (creator vector)”建模开发者历史行为, 在无需源代码和交易数据的前提下实现了部署阶段的早期预警, 进一步推动了纯字节码检测范式的发展.

第 3 类研究基于网络嵌入, 随着图嵌入技术和图神经网络 (graph neural network, GNN) 的发展, 基于网络嵌入的检测方法逐渐崭露头角. 这类方法通过捕捉交易中账户之间的相互作用模式, 利用图结构分析和机器学习技术实现更高效的欺诈检测. Wu 等人[27] 提出的 trans2vec 算法, 通过结合以太坊交易图中的交易金额和时间戳信息, 对交易图进行嵌入表示, 从而有效捕捉交易中的动态特征. Tan 等人[28]则采用 node2vec 算法, 通过调整随机游走的概率计算节点嵌入, 并将所得嵌入应用于下游的庞氏骗局检测任务中. 这些方法通过图嵌入技术, 能够更好地表征账户之间的交易行为, 为检测庞氏骗局提供了新的视角. 随着图神经网络的引入, 基于图的检测算法取得了显著进展. 例如, Yu 等人[29]采用图卷积神经网络对以太坊交易图进行分析, 通过捕捉账户之间的高阶关系, 实现了对庞氏骗局的高效识别. 图神经网络能够自动学习图结构中的特征表示, 避免了传统方法中依赖手工设计特征的局限性, 从而提升了检测的准确性和鲁棒性.

近年来, 也有研究尝试从图结构建模的角度出发, 通过多视图分析与控制流建模进一步提升庞氏骗局检测的性能. 一方面, Jiang 等人[30]提出了一种多视图对比图神经网络模型, 基于元路径生成多种语义视图, 通过强化学习机制动态分配视图权重, 并结合多尺度对比学习增强图表示的区分性, 在以太坊庞氏合约检测任务中表现出优异的检测效果, 说明融合图结构信息与跨视图语义关系对于识别复杂欺诈行为具有显著优势. 另一方面, 黄静等人[31]提出了一种基于代码控制流图的庞氏骗局合约检测模型, 将 Solidity 合约源代码解析为控制流图结构, 并结合图卷积网络与 Transformer 多头注意力机制, 深入挖掘合约中关键交易路径与控制依赖关系. 实验表明, 该方法相比传统机器学习与序列建模方法在准确率与 *F*1 分数方面具有明显优势, 验证了代码控制逻辑在庞氏骗局检测中的有效性, 并拓展了图结构方法在智能合约分析中的应用边界. Chen 等人[32]提出 PonziHunter 框架, 进一步将图建模从源代码层面拓展至仅依赖链上字节码的场景. 该方法通过反编译字节码构建融合跨函数控制流与状态变量依赖关系的图结构, 并引入代码切片思想识别与庞氏行为相关的关键基本模块; 同时设计了面向控制流图的结构感知数据增强策略, 并结合对比预训练实现对对抗扰动的抵抗提升模型鲁棒性. PonziHunter 不仅在真实数据集上显著优于现有工具, 还具备定位欺诈关键路径的能力, 体现了从字节码出发进行可解释图建模的有效路径.

尽管上述方法在庞氏骗局检测中取得了一定成效, 但仍存在显著缺陷. 首先, 大多数合约的源代码是未公开的, 这限制了基于源代码分析方法的适用性. 其次, 基于交易特征的方法通常需要在大量交易完成后才能提取有效特征, 而此时投资者可能已经造成了严重的经济损失. 为解决这些问题, 本文提出仅依赖智能合约字节码识别潜在欺诈行为的智能合约实时审计方法. 与现有方法相比, 本文的创新点在于能够在智能合约部署阶段即实现安全性评估, 从而为早期发现和预防庞氏骗局提供了一种有效的解决方案. 通过结合图结构分析和图神经网络技术, 本文不仅提升了检测的效率和精度, 还为智能合约的安全性研究开辟了新的方向.

3 方　法

本文所提出的庞氏骗局实时检测算法框架如图 1 所示. 首先根据现有的带标签的数据集, 获取所有合约的字节码表示文件. 在此基础上, 通过字节码转操作码算法将智能合约的字节码转化为操作码序列. 接下来, 基于操作码序列, 以操作码为节点, 操作码之间的不同关系为边构建 3 种不同的图结构: 操作码顺序图、控制流图和控制流超图. 最后, 利用图卷积神经网络以及超图卷积神经网络分别从每种图结构中提取深层次特征, 并构建模型识别潜

在的庞氏骗局合约. 本节将依次介绍框架中各个相关的模块.

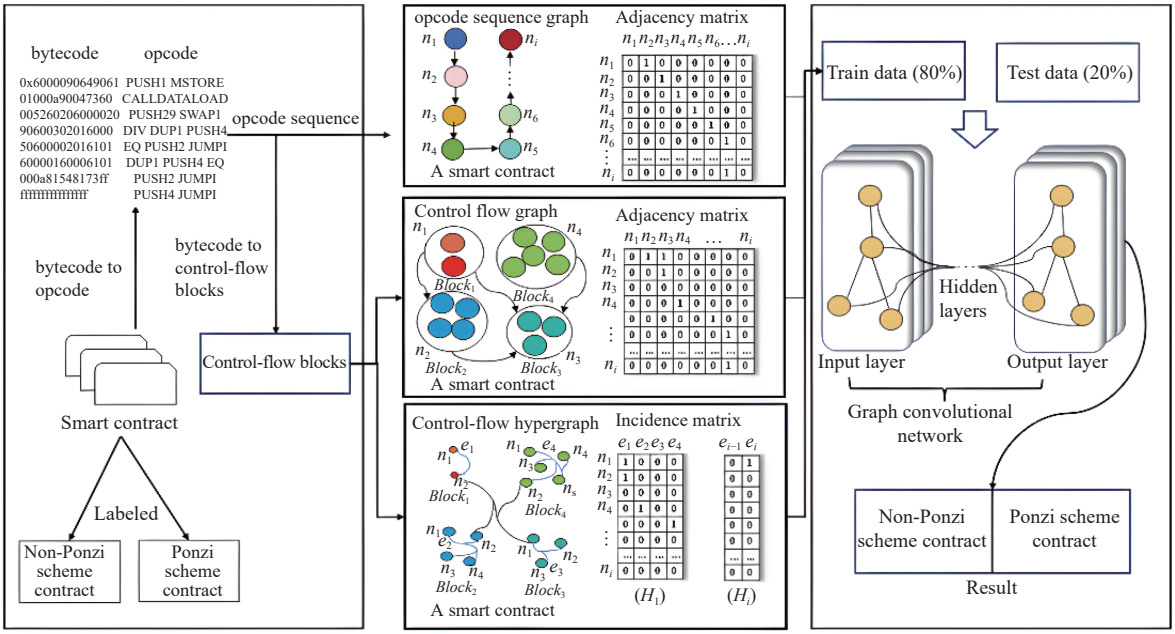


图 1　基于操作码图结构的实验框架图

3.1　字节码转操作码

以太坊智能合约的字节码是一种由一系列十六进制数字组成的低级指令序列, 由以太坊虚拟机 (EVM) 逐条解析并执行. 根据以太坊黄皮书[33]的定义, EVM 是一个基于栈的状态机, 其运行过程严格受控于操作码的结构与状态转换逻辑. 每条指令不仅控制栈的基本操作, 还可能引发内存的读写以及控制流的跳转等, 从而构建出完整的合约执行路径.

将字节码 (bytecode) 转化为操作码 (opcode) 的过程, 实质上是对这些十六进制数据进行逐字节解析, 并映射为 EVM 可识别并执行的具体指令. 每种操作码对应一种特定的行为, 例如 PUSH 用于将数据压入栈中, DUP 用于复制栈顶元素, 而 SWAP 则实现栈中数据的交换操作. 在实际解析过程中, 字节码中的每个字节会被依次读取, 并根据 EVM 的操作码规范映射为相应的操作码. 该映射关系具有明确且固定的规则: 例如, 字节值 0x60–0x7f 分别代表 PUSH1–PUSH32, 0x80–0x8f 表示 DUP1–DUP16, 而 0x90–0x9f 则对应 SWAP1–SWAP16 等操作. 操作码往往并不仅由一个字节组成. 例如, PUSH 操作后面会紧跟一段数据, 用作其操作数. 因此, 在生成操作码序列时, 需同时提取与操作码相关联的操作数信息. 每条操作指令通常会被标准化为一个操作码三元组: (位置, 操作码名称, 操作数). 其中, 每条操作指令的“位置”表示其在字节码序列中的起始偏移. 大多数非 PUSH 操作码不含操作数, 而 PUSH1–PUSH32 类指令则分别携带 1–32 字节不等的操作数. 因此, 每条操作指令的长度从 1 字节到最多 32 字节不等 (后文图 2 所示为截取的一段字节码转换为操作码三元组的实例). 通过这种结构化表示方式, 原始的字节码被转换为操作码三元组序列, 为后续的顺序图构建, 控制流分析等推理提供了坚实基础.

3.2　操作码顺序图构建

操作码顺序图反映的是操作码的先后顺序关系. 我们将操作码三元组中不同的操作码视为不同节点, 通过遍历操作码三元组序列, 依据操作码元组的先后顺序添加操作码之间的有向边, 构建出表示操作码顺序关系的有向图, 并将生成的有向图转换为图对象. 图 3 展示了一个智能合约操作码的部分顺序图. 我们对每个样本执行上述操作, 最终构建出一个数据集, 其中每个元素都是一个图对象, 表示相应的有向图及其标签.

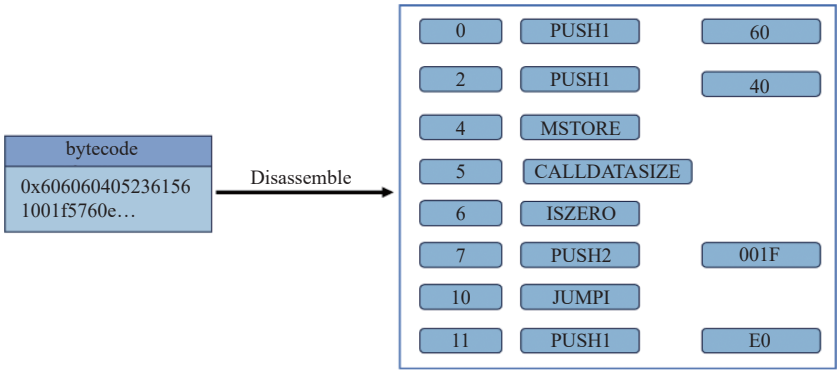


图 2　智能合约字节码转换为操作码

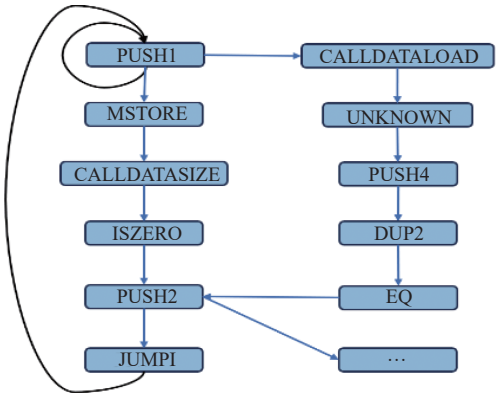


图 3　操作码顺序图

3.3　操作码控制流图构建

3.3.1　操作码控制流代码块划分

在智能合约执行过程中, 操作码通常按顺序执行, 但部分操作码具有特殊的控制流语义, 如 JUMP、JUMPI、RETURN、REVERT、SELFDESTRUCT 等, 会导致程序跳转或终止. 为了对字节码进行结构化建模, 需将操作码序列划分为控制流基本块 (basic block), 并构建控制流图 (control flow graph, CFG). 具体而言, 本文定义了两类操作码集合用于基本块划分.

blkend 集合表示导致基本块结束的指令, 包括: ① STOP: 停止执行; ② RETURN: 从合约返回数据; ③ REVERT: 回滚状态并返回错误信息; ④ INVALID: 无效操作, 导致异常终止; ⑤ SELFDESTRUCT: 销毁合约并转移余额; ⑥ JUMP: 无条件跳转; ⑦ JUMPI: 条件跳转.

blkbg 集合表示基本块开始的指令, 包括: ① 所有 JUMPDEST 指令: 表示合法跳转目标位置; ② 所有紧跟在 JUMP 或 JUMPI 等跳转指令后的操作码: 表示跳转后可能的执行入口.

基于上述规则, 系统地将字节码划分为基本块, 并构建控制流图 (CFG), 为后续的静态分析和图神经网络建模提供结构化输入. 该流程已在近期研究中得到验证, 如 EtherSolve 提出通过符号堆栈执行分析动态跳转, 准确恢复闭源智能合约的控制流结构, 显著提升了字节码级漏洞检测的可行性与准确率. 准确的基本块划分是理解合约执行逻辑、发现潜在风险的关键步骤. 算法 1 详细描述了本文采用的智能合约字节码基本块划分流程.

算法 1. 操作码序列划分为基本块.

输入: 操作码三元组列表 optris;

输出: 基本块列表 blocks.

1. 初始化空列表 tris 用于暂存当前块指令
2. 初始化空列表 blocks 用于存储所有基本块
3. 定义集合 blkend 和 blkbg 来表示结束和开始操作
4. 对于 optris 中的每个三元组 tri 执行
5. 如果 tri.opcode 在 blkend 中:
6. 将 tri 添加到 tris
7. 将 tris 添加到 blocks
8. 清空 tris
9. 如果 tri.opcode 在 blkbg 中:
10. 如果 tris 非空:
11. 将 tris 添加到 blocks 中
12. 清空 tris
13. 将 tris 添加到 blocks
14. 否则 (默认情况):
15. 将 tris 添加到 blocks
16. 如果遍历结束后 tris 非空:
17. 将 tris 添加到 blocks
18. 断言: blocks 中的所有操作码数量之和等于原始 optris 数量
19. 返回 blocks

算法 1 输入一个名为 optrits 的参数, 该参数是一个操作码三元组的列表. 算法的目标是根据以太坊智能合约的控制流结构, 将这些操作码划分为若干基本块, 最终返回一个列表, 列表中的每个元素都是一个代码块, 表示一段连续执行的指令序列.

这样, 通过调用 blockoptris 函数, 可以将智能合约的控制流程清晰地划分为多个代码块, 为后续的合约分析提供了便利.

3.3.2 控制流图构建

基于上述操作码块的划分结果, 下面介绍控制流图 $G(V,E,W)$ 的定义和构建. 在控制流图 $G(V,E,W)$ 中, 一个节点 V_i 表示一个操作码代码块, 边 $e=(V_i,V_j)$ 和相应的权重 W 表示合约在执行过程中节点 V_i 存在概率 W 跳转到 V_j. 一个操作码代码块可能由多个操作码三元组按顺序组成. 图 4 展示了一个智能合约部分操作码构建的部分控制流图, 图中每一个方框表示一个节点, 每个节点包含数量不同的操作码三元组.

算法 2 给出了控制流图构建的方法. 我们根据每个代码块中包含的控制操作码情况, 构建代码块之间的有向连接, 并设置相应的概率. 其核心步骤如下.

(1) 单一操作码节点: 若一个节点仅包含一个操作码

• 若不是最后一个块, 则以概率 1 连接到下一个基本块.

• 若为最后一个块, 则等概率连接到其他基本块.

(2) 最后是 JUMP 操作码的节点

• 若前一个操作码是 PUSH, 则将该节点以概率 1 连接到 PUSH 操作数给定的地址所在的基本块.

• 若无法确定跳转地址, 则等概率地跳转到其他基本块.

(3) JUMPI 操作码节点

• 若前一个操作码是 PUSH, 则:

• 以 0.5 的概率跳转到 PUSH 操作数给定的地址所在的基本块.
• 若不是最后一个块, 另外 0.5 的概率跳转到下一个基本块.
• 若是最后一个块, 剩余 0.5 的概率等分连接到其他基本块.
• 若无法确定跳转地址, 则等概率地跳转到其他基本块.

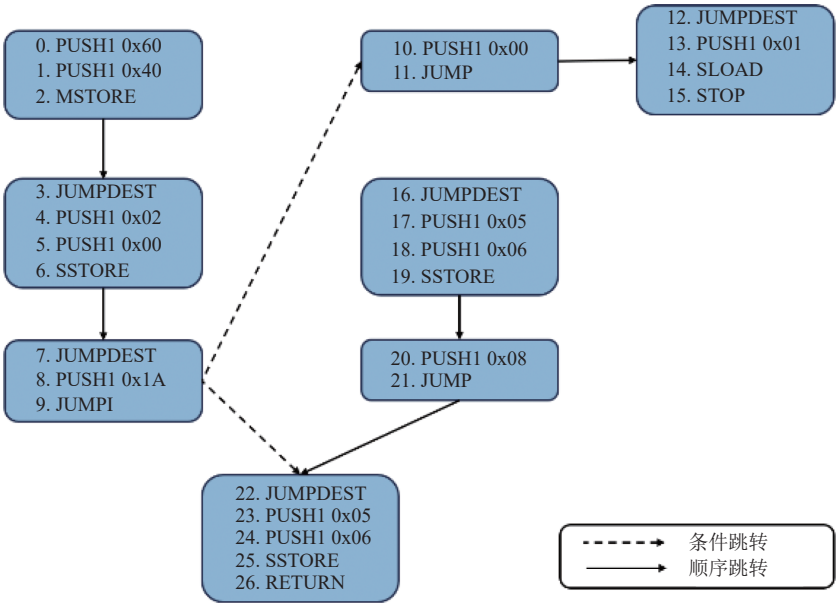


图 4　操作码控制流图

(4) 结束操作码节点

• 认为该块为控制流终止点, 等概率连接到其他基本块.

(5) 默认情况

• 若不是最后一个块, 另外 0.5 的概率连接到下一个基本块.
• 若是最后一个块, 剩余 0.5 的概率等分连接到其他基本块.

智能合约在执行过程中, 可能根据不同的情况有不同的执行路径. 由于我们不可能模拟出所有的情况, 因此, 在上述构建代码块连接的过程中, 我们引入了概率, 表示各种连接的可能性. 尽管这种表示方式并不是真实的跳转概率, 但在一定程度上模拟了可能的跳转逻辑. 控制流图构建的方法如算法 2 所示.

算法 2. 控制流图构建.

输入: 智能合约操作码基本块列表 blocks;
输出: 控制流图 $G(V, E, W)$.

1. blocks 的长度记为 n, p 记为 $1/(n-1)$
2. 初始化空列表 source, target, edge_weight
3. 初始化 block_offsets: 每个 blocks[i] 的起始 offset
4. 对 i 从 0 到 $n-1$ 执行:
5.　若 blocks[i] 仅包含一个操作:
6.　　若 i 是最后一个块:
7.　　　对 j != i 添加边 $i \rightarrow j$, 权重为 p
8.　　否则添加边 $i \rightarrow i+1$, 权重为 1

9. 若 blocks[i] 的最后一个操作是"JUMP":
10. 若前一个操作是 PUSHx:
11. 查找目标 offset 对应的块 j, 添加边 $i \to j$, 权重为 1
12. 否则:
13. 对 i != j 添加边 $i \to j$, 权重为 p
14. 若最后操作是"JUMPI":
15. 若前一个是 PUSHx:
16. 查找目标 offset 对应的块 j, 添加边 $i \to j$, 权重为 0.5
17. 若 $i < n-1$, 添加边 $i \to i+1$, 权重为 0.5
18. 否则对 j != i 添加边, 权重为 $0.5p$
19. 否则:
20. 对 j != i 添加边 $i \to j$, 权重为 p
21. 否则 (普通顺序执行):
22. 若 $i < n-1$, 添加边 $i \to i+1$, 权重为 0.5
23. 否则对 j != i 添加边, 权重为 $0.5p$
24. 返回图 $G(V, E, W)$

3.4 操作码控制流超图构建

在深入建模智能合约的执行逻辑时, 传统控制流图虽能揭示合约中基本块之间的跳转关系, 但难以表达操作码之间的高阶语义关联. 为此, 我们引入控制流超图结构, 用以建模多个代码块之间的结构共性及语义关联, 从而提升图结构对智能合约行为模式的表达能力.

具体地, 首先从智能合约的字节码中提取出基本块, 过滤掉仅包含 STOP 指令的冗余块, 获得清洗后的基本块集合. 随后基于该集合构建操作码控制流超图 $G(V,E)$. 在该超图中, 以所有操作码作为核心节点 V_i, 用于表示合约中的基础操作单元. 针对每个基本块, 构建一个超边 $e = (V_i, V_j, \ldots, V_k)$, 将该基本块中包含的所有操作码作为超边的关联节点, 从而显式捕捉块内操作码的组合语义和局部依赖模式. 由于现有 GNN 框架多基于简单图, 为兼容训练过程, 本文采用二部图形式对超图进行编码. 对每个基本块引入一个超边虚拟节点, 并通过边将该虚拟节点与所属操作码节点连接, 等效表达超边的多元关系. 此外, 对于仅包含单个操作码的基本块, 本文采用滑动窗口策略, 将连续两个单操作码组合构成超边, 进一步增强对局部操作码依赖模式的捕捉能力. 最终, 所构建的控制流超图不仅保留了块内的顺序特性, 还刻画了操作码之间的高阶结构关联, 极大丰富了模型对智能合约执行逻辑和语义模式的表示能力. 该过程的具体实现流程如算法 3 所示.

算法 3. 控制流超图构建.

输入: 智能合约操作码基本块列表 blocks;
输出: 控制流超图 $G(V, E)$, 其中, V 表示操作码节点和超边虚拟节点, E 表示超边虚拟节点指向操作码节点的边.

1. 初始化一个空列表 filtered_blocks
2. 遍历每个 block ∈ blocks:
3. 如果 block 中任意一条指令不是"STOP":
4. 将 block 添加到 filtered_blocks
5. 初始化: edge_index, superedge_dict, current_superedge, current_superedge_id (从 145 开始编号)
6. 遍历每个 block ∈ filtered_blocks:
7. 若 block 仅含一个操作码:

```
8.      将其编号减 1 后加入 current_superedge
9.      若 current_superedge 长度为 2:
10.         构造超边元组: superedge ← tuple(current_superedge)
11.         若 superedge 不在 superedge_dict:
12.             分配新 ID: superedge_dict[superedge] ← current_superedge_id
13.             current_superedge_id += 1
14.         对每个 opcode 节点 i ∈ superedge:
15.             添加边连接: edge_index.append((superedge_dict[superedge], i))
16.         清空 current_superedge
17. 若 block 含多个操作码:
18.     提取所有 opcode 的编号减 1, 构造超边: superedge ← tuple([…])
19.     若 superedge 不在 superedge_dict:
20.         分配新 ID: superedge_dict[superedge] ← current_superedge_id
21.         current_superedge_id += 1
22.     对每个 opcode 节点 i ∈ superedge:
23.         添加边连接: edge_index.append((superedge_dict[superedge], i))
24. 返回超图 G(V, E)
```

图 5 展示了一个合约的部分控制流超图示意图, 其中节点编号 1–13 分别对应不同的操作码, 图中椭圆表示超边, 椭圆中的节点来自同一个基本块, 体现其内部操作顺序与语义关联.

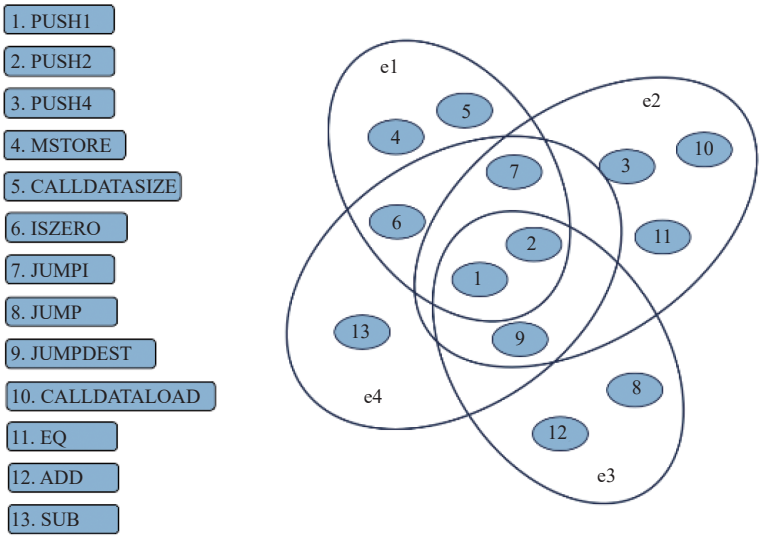


图 5 操作码超图

3.5 图卷积神经网络

图卷积神经网络[14]是一种专为处理图结构数据而设计的深度学习模型. 在图结构中, 节点之间具有不规则的拓扑关系, 传统神经网络难以直接建模. 因此, 传统的神经网络方法并不适合直接应用于图数据. GCN 通过引入图卷积操作, 有效整合了节点自身与其邻居的信息, 实现了对图中节点特征的提取与更新.

考虑一个图 $G = (V, E)$, 其中 V 表示节点集合, E 表示边集合. 节点间的连接关系可表示为邻接矩阵 $A \in \mathbb{R}^{n \times n}$, 其中 $n = |V|$. 为了实现节点的自信息保留, 引入单位矩阵 I, 定义包含自环的邻接矩阵为:

$$\hat{A} = A + I \tag{1}$$

并令 $\hat{D}$ 为对应的度矩阵, 表示为:

$$\hat{D}_{ii} = \sum_j \hat{A}_{ij} \tag{2}$$

则单层 GCN 的传播规则为:

$$H^{(1)} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}XW\right) \tag{3}$$

其中, X 是节点初始特征矩阵, W 为可训练权重, σ 是非线性激活函数, $H^{(1)}$ 是第 1 层输出的节点特征表示. 在我们的模型中 GCN 被堆叠为 3 层, 每一层都在不断扩大感受野, 实现更高阶邻域的特征整合. 节点特征经过 3 层传播后, 采用全局平均池化 (global mean pooling) 将节点级特征整合为图级表示:

$$h_G = \frac{1}{|V|}\sum_{v_i \in V} h_i^{(3)} \tag{4}$$

其中, $h_i^{(3)} \in \mathbb{R}^d$ 表示第 3 层中第 i 个节点的输出, $h_G \in \mathbb{R}^d$ 是图的整体向量表达.

该图级向量随后输入至线性全连接层进行分类预测:

$$z = h_G W^{(fc)} + b \tag{5}$$

$$\hat{y}_j = Softmax(z) = \frac{\exp(z_j)}{\sum_{k=1}^{c}\exp(z_k)},\ j = 1,2,\ldots,c \tag{6}$$

其中, $W^{(fc)}$ 和 b 为分类器的权重与偏置项. 该模型通过堆叠图卷积层、激活函数和池化操作, 实现了对图中结构和特征的有效建模, 并完成分类任务.

3.6 超图卷积神经网络

超图卷积神经网络是传统图卷积神经网络在超图域的重要扩展[34]. 相较于普通图结构中边仅能连接两个节点的限制, 超图中的超边可以连接任意数量的节点集合, 这种高阶连接特性使其能够更灵活地表征现实世界中复杂的关系网络.

考虑一个图 $H_G = (V, E)$, V 是节点集合, E 是超边集合, 定义关联矩阵 $H \in \mathbb{R}^{n\times m}$:

$$H(i,j) = \begin{cases} 1, \text{ if } v_i \in e_i \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

定义节点度矩阵 D_v 与超边度矩阵 D_e 为:

$$D_v(i,i) = \sum_{j=1}^{m} w(e_j)H(i,j) \tag{8}$$

$$D_e(j,j) = \sum_{i=1}^{n} H(i,j) \tag{9}$$

HGCN 在此结构上定义了如下的节点更新传播机制:

$$H^{(k+1)} = \sigma\left(\hat{D}_v^{-\frac{1}{2}}HW_e\hat{D}_e^{-1}H^{\mathrm{T}}D_v^{-\frac{1}{2}}H^{(k)}W^{(k)}\right) \tag{10}$$

其中, $H^{(k)}$ 是第 k 层的节点表示, $W^{(k)}$ 是可学习权重, W_e 是超边权重矩阵, σ 是非线性激活函数, $\hat{D}_v^{-\frac{1}{2}}HW_e\hat{D}_e^{-1}H^{\mathrm{T}}D_v^{-\frac{1}{2}}$ 相当于一个归一化后的超图拉普拉斯操作符.

公式 (10) 揭示了 HGCN 的三阶段计算流程: 首先完成超边归一化邻接矩阵构建, 确保消息传递的数值稳定性, 其次执行特征空间的线性变换, 最终经过激活函数实现非线性映射. 这种设计使得节点特征能够沿着超边定义的高阶拓扑结构进行扩散, 同时保持对超边权重的敏感性.

3.7 动态注意力融合模块

不同的图表示 (SG、CFG、CFH) 从不同维度刻画了智能合约的异质特征. 在多模型集成中, 传统的硬投票机制或静态加权策略通常假设各子模型的可靠性是固定的. 然而, 在面对高度复杂的庞氏合约变种时, 静态假设往往会导致“少数服从多数”的误判, 甚至被表现较差的子模型拖累最终性能.

为突破这一局限, 受注意力机制在序列建模中展现出的自适应特征提取能力[35]以及异构图网络中语义级权重分配思想[36]的启发, 本文设计了动态注意力融合 (dynamic attention fusion, DAF) 模块.

该模块借鉴了注意力机制中“自适应聚焦”的核心逻辑, 并针对智能合约异构图特征的融合需求进行了改进. 其核心机制是: 由各子模型在当前样本上的预测“置信度”实时决定其在最终决策中的权重. 通过引入这种数据驱动的决策方式, 系统能够自动赋予判断最笃定的子模型更高的权重 (即话语权), 从而动态化解模型间的信息冲突, 提高模型在面对未知变种时的鲁棒性. 具体推导过程如下:

首先, 对于给定的智能合约样本, 设 3 个子模型 $m \in \{\mathrm{SG}, \mathrm{CFG}, \mathrm{CFH}\}$ 预测其属于庞氏骗局的概率为 p_m. 为了衡量模型的判定把握, 我们构建每个模型输出的完整二分类概率分布 $P_m = [p_m, 1-p_m]$ (分别对应庞氏骗局与正常合约的概率), 并提取该分布中的最大值作为模型的实时置信度得分 C_m:

$$C_m = \max(P_m) \tag{11}$$

其中, $C_m \in [0.5, 1]$, 数值越接近 1, 表示模型 m 对当前分类结果越笃定. 其次, 考虑到不同图模型在全局任务中的基础特征提取能力存在差异, 我们引入了一组可学习的基础权重参数 $W_{\text{base}} = \{w_{\text{SG}}^{\text{base}}, w_{\text{CFG}}^{\text{base}}, w_{\text{CFH}}^{\text{base}}\}$. 结合实时置信度, 计算每个模型的动态注意力得分 e_m:

$$e_m = w_m^{\text{base}} \cdot C_m \tag{12}$$

其中, 基础权重参数 W_{base} 在网络初始化时均设为 1, 随后在反向传播过程中通过梯度下降联合更新, 无需人工干预超参数设定.

随后, 利用 *Softmax* 函数对得分进行归一化, 生成各模型针对该特定样本的动态权重 w_m:

$$w_m = \frac{\exp(e_m)}{\exp(e_{\text{SG}}) + \exp(e_{\text{CFG}}) + \exp(e_{\text{CFH}})} \tag{13}$$

最终的集成预测概率 P_{final} 通过对各模型的输出概率进行动态加权求和获得:

$$P_{\text{final}} = \sum_m w_m \cdot P_m \tag{14}$$

综上所述, 本节提出的 DAF 模块将“全局可学习的基础权重”与“局部样本的实时置信度”进行了有机结合. 这种“数据驱动、动态聚焦”的软融合机制, 使得模型在面对具有高隐蔽性、特征分布极度不均的新型庞氏骗局样本时, 能够自适应地筛选最优特征分支, 显著增强了系统决策的鲁棒性与解释性.

3.8 多图集成学习方法

为了进一步提升庞氏骗局检测的精度与鲁棒性, 本文提出了一种名为 MG-DAF (multi-graph dynamic attention fusion) 的增强型多图集成学习框架. 如图 6 所示, 该框架协同利用序列、控制流及异构关联这 3 类图模型, 从多维特征空间深度捕获智能合约的欺诈表征. 引入了动态注意力融合模块, 该模块摒弃了传统的静态加权方式, 通过实时计算各子模型输出概率的置信度度量, 实现决策权重的自适应分配. 实验证明, 这种“高置信度优先”的决策机制能够有效识别复杂多变的攻击模式, 显著增强了模型对新型庞氏骗局的检测召回率.

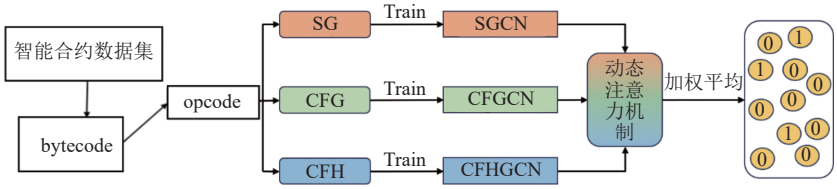


图 6 MG-DAF 集成方法框架

4 实　验

4.1 数据集与实验环境

为了全面评估 MG-DAF 框架的检测性能与泛化能力, 本实验构建了两个严格物理隔离的数据集合: 用于模型训练与内部评估的基础数据集, 以及用于外部压力测试的泛化数据集.

(1) 基础数据集 (内部训练与测试): 本文基础数据集源自 Xblock (https://xblock.pro/), 原始包含 6 498 个以太坊智能合约 (314 个庞氏样本, 6 184 个正常样本), 标签均经由人工交叉验证得到[26]. 为严谨验证模型的泛化性能, 本研究在实验初期实施了“样本预隔离”策略: 从 6 184 个正常合约中随机抽取 200 个样本作为预留负类背景, 并将其从模型训练、验证及内部测试的所有流程中完全剔除. 最终, 用于模型构建的基础实验池包含 6 298 个合约样本.

(2) 外部泛化数据集 (Test_ponzi): 为评估模型在未知变种下的鲁棒性, 本研究构建了模拟真实链上环境的非平衡外部数据集 Test_ponzi. 该数据集的正类样本源自 Bartoletti 等人[17]提供的经典庞氏骗局地址库. 通过与基础数据集进行严格的代码去重比对, 剔除重叠样本后, 最终选出 62 个全新的庞氏骗局合约. 负类样本则采用前述预先隔离的 200 个常规合约. 该集合共计 262 个样本, 构成了完全独立的外部压力测试场景.

基于上述数据源, 本文按照第 3 节所述的建模方法, 统一生成了操作码顺序图、控制流图及控制流超图. 所有图结构的原始特征均统一编码为 145 维的操作码统计向量, 作为 MG-DAF 框架的多模态输入.

所有实验在以下硬件与软件环境下进行: Intel(R) Core(TM) i9-13900K CPU @ 3.00 GHz, 64 GB 内存, NVIDIA GeForce RTX 4090 GPU (24 GB 显存). 图构建与预处理使用 Python 3.9 实现, 模型训练与推理基于 PyTorch Geometric 框架, 在 CUDA 12.1 环境下运行.

4.2 评估指标与评估方法

在本文中, 我们采用与其他有代表性的庞氏骗局检测方法同样的评价指标 (*Precision*、*Recall* 和 *F*1 分数) 来全面评估模型的性能. 以下是对这些评价指标的详细解释以及相应的数学公式.

(1) 精确度 (*Precision*): 表示模型正确预测为正例的样本数量在所有预测为正例的样本中的比例. 其计算公式为:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}.$$

(2) 召回率 (*Recall*): 表示模型正确预测为正例的样本数量在所有实际正例样本中的比例. 计算公式如下:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}.$$

(3) *F*1 分数: 是精确度和召回率的调和平均数, 提供了综合考虑模型性能的指标. 计算公式为:

$$F1\text{-score} = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

为了更精确地评估模型, 我们采用 5 折交叉验证评估模型结果, 即将数据集划分为 5 个互斥子集, 每轮以 4 份为训练集、1 份为测试集. 同时考虑到类别极度不平衡 (庞氏合约样本仅 314 个), 通过 SMOTE 过采样方法在每轮训练集上执行上采样, 使训练阶段正负样本比达到 1:1, 而测试集保持原始分布以反映真实场景性能. 实验最终结果取 5 轮指标的平均值.

4.3 模型参数设置

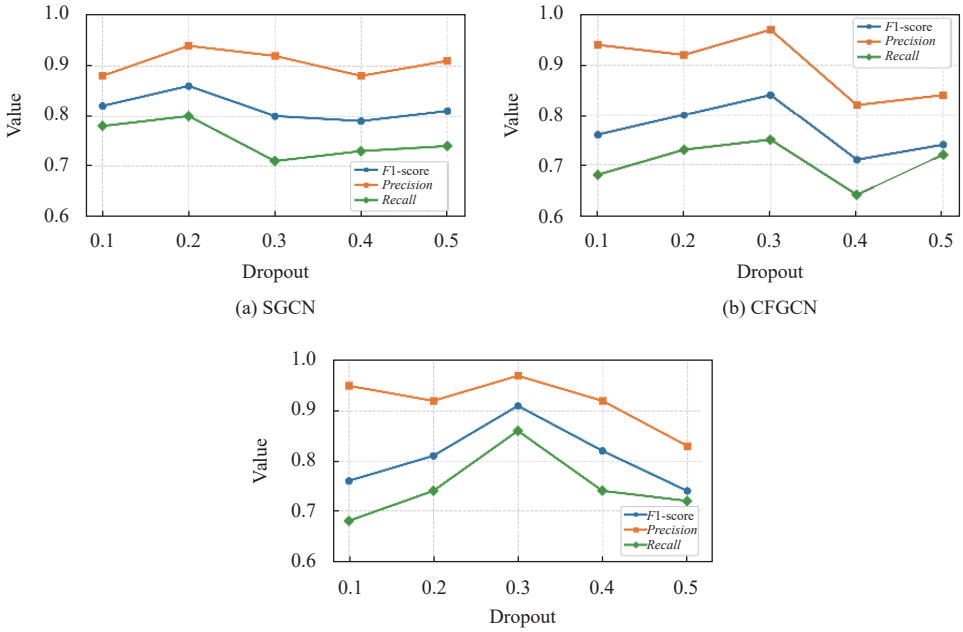
由于模型中涉及多个重要超参数, 下面首先探索超参数的设置问题.

4.3.1 随机失活 (Dropout)

随机失活是控制神经网络正则化强度的关键参数, 表示训练过程中隐藏层神经元被随机屏蔽的概率. 通过在训练时以设定比率将神经元输出置零, 迫使网络不依赖于特定神经元的激活模式, 从而提升泛化能力. 随机失活比率并不是越大越好, 过大的比率会导致模型较难收敛, 难以在训练集上取得令人满意的正确率.

在实验中, 我们分别将 Dropout 设置为 (0.1, 0.2, 0.3, 0.4, 0.5), 实验结果如图 7 所示. 可以看出随着 Dropout 的

增加, 模型效果总体上都是先上升再下降, 根据实验结果最终将 SGCN、CFGGCN、CFHGCN 这 3 个模型的 Dropout 值分别设置为 0.2、0.3、0.3.



(a) SGCN (b) CFGCN

(c) CFHGCN

图 7 Dropout 对实验结果的影响

4.3.2 批量大小 (Batch size)

批大小决定了单次输入模型的样本数量, 直接影响训练过程的稳定性和效率. 较大的批次能充分利用 GPU 并行计算能力加速训练, 例如批量处理 128 张图像可显著减少迭代次数, 但同时会降低梯度更新的随机性, 可能导致模型收敛至泛化性较差的局部最优解; 较小的批次虽然增加了训练波动, 却因其噪声注入效应有助于探索更优的参数空间.

我们使用 16、32、64、128、256 这 5 个参数分别进行实验. 实验结果如后文图 8 所示. 随着批大小的增加, SGCN 和 CFGCN 的检测性能逐渐提高, 而在 CFHGCN 中, 当批大小提高到 256 时, 性能开始下降. 最终我们选取 256、256、128 分别作为 SGCN、CFGCN、CFHGCN 这 3 个模型的 Batch size 取值.

4.3.3 隐藏通道数 (Hidden_channels)

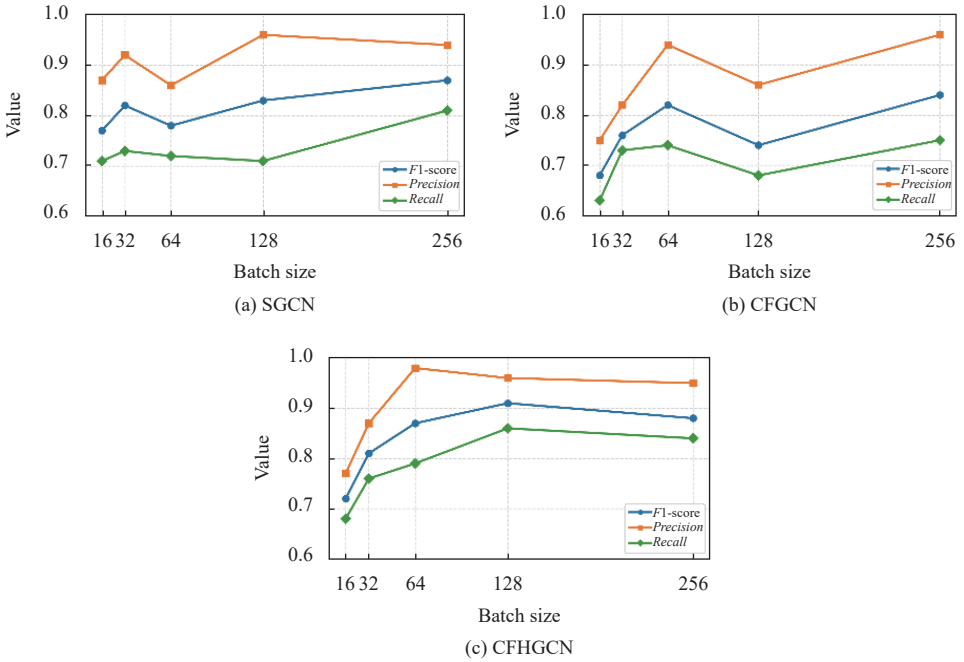
隐藏层通道数是神经网络中控制特征表达能力的核心参数, 常见于卷积层或图神经网络层中, 用于定义每层输出的特征维度数量. 通过增加通道数, 网络能够捕获更复杂的空间或语义模式. 需注意的是, 通道数并非越多越好, 过高的通道数会导致模型参数冗余, 不仅容易引发过拟合, 还可能因显存不足导致训练中断; 而过低的通道数则会使模型难以捕捉数据中的关键特征, 造成欠拟合.

我们分别采用 16、32、64、128、256 这 5 个参数进行试验, 实验结果如后文图 9 所示, 随着 Hidden_channels 的增加, 3 个模型的实验效果产生了不同的变化, 最终我们分别使用 128、64、256 作为 SGCN、CFGCN、CFHGCN 这 3 个模型的 Hidden_channels 取值.

4.4 模型结果与消融实验

通过上述对超参数的探索, 我们选取了最佳参数, 并分别基于顺序图、控制流图和控制流超图训练 3 个子模型, 最终通过动态注意力融合机制集成预测结果, 构成完整框架 MG-DAF. 结果如图 10 所示. MG-DAF 在庞氏骗局检测任务中取得显著优势, 精确率、召回率与 $F1$ 分数分别达到 0.9872、0.8954 和 0.9391, 充分验证了多图协

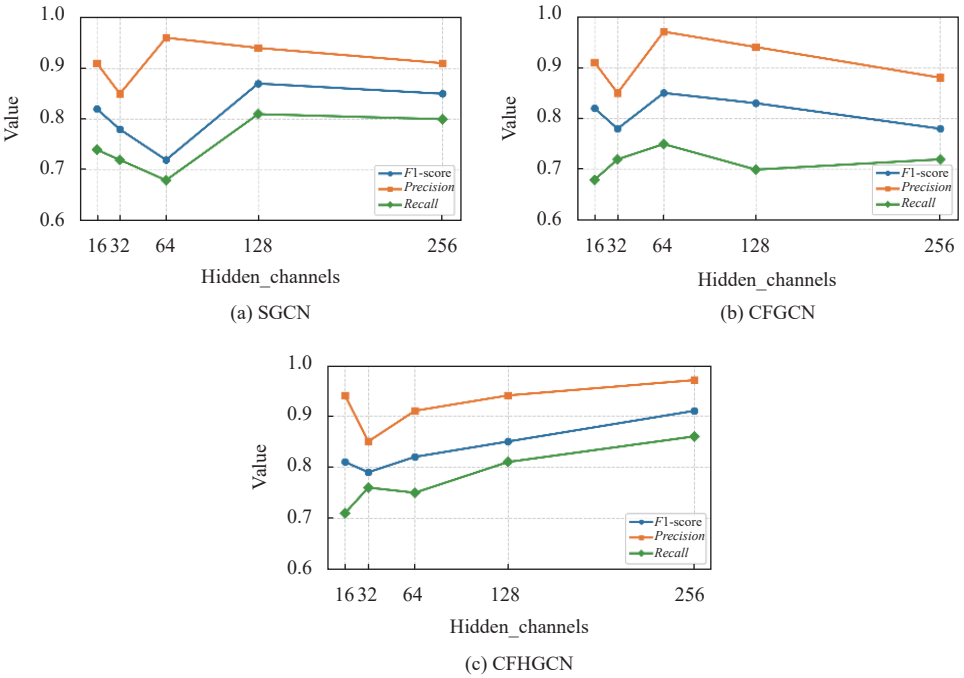
同建模的有效性.



(a) SGCN (b) CFGCN

(c) CFHGCN

图 8 Batch size 对实验结果的影响



(a) SGCN (b) CFGCN

(c) CFHGCN

图 9 Hidden_channels 对实验结果的影响

为了深入理解各图结构对检测性能的独立贡献, 我们进一步开展了消融实验, 分别评估了这 3 个单图模型及其集成效果. 图 10 展示了不同模型配置下的实验结果.

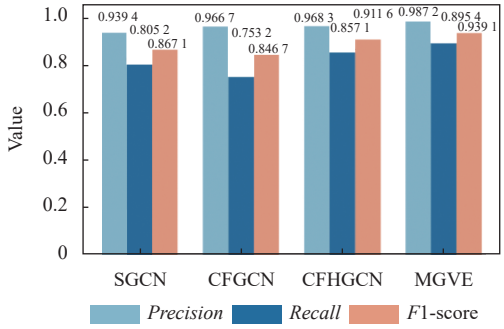


图 10 不同模型配置下的实验结果

导致 3 个模型性能差异的核心原因在于各模型对合约字节码的图结构建模方式不同. SGCN 基于顺序图, 仅关注操作码的线性执行顺序, 因此在分析简单合约时表现较好. 然而, 庞氏骗局通常包含复杂的控制逻辑, 顺序图无法捕捉代码块间的隐式依赖关系, 导致对复杂骗局的召回率较低. CFGCN 采用控制流图, 显式建模代码块间的跳转关系, 其精确率提升至 0.966 7, 表明对固定跳转模式的检测能力更强. 但 CFG 仅支持二元跳转建模, 无法在一个结构单元内自然地表示多个操作码间的复杂、协同关系, 导致召回率降至 0.753 2. CFHGCN 通过引入控制流超图解决了这一瓶颈. 其核心在于每条超边直接对应一个完整的基本块, 并将该块内所有操作码节点聚合为一个高阶单元. 这种建模方式完整保留了块内操作码的执行顺序和相互依赖关系. 这避免了 CFG 将基本块视为原子节点而丢失的内部细节. 同时图神经网络在超图上进行消息传递时, 可以将一个超边内部的复杂信息作为一个更丰富的“信号”传递给与其相连 (通过控制流跳转关系) 的其他超边. 这比 CFG 中简单的块间跳转边能传递更丰富的上下文信息, 有助于理解跨基本块的复合逻辑流程. 这使得其召回率提升至 0.857 1, $F1$ 分数达到 0.911 6.

MG-DAF 模型通过集成顺序图、控制流图和超图的建模结果, 并采用动态注意力融合机制综合决策, 实现了性能的进一步优化, 得到如下结论.

(1) 特征互补性: 顺序图捕捉高频操作序列, 控制流图定位异常跳转, 超图解析复杂交互, 三者覆盖了庞氏骗局的多样性特征.

(2) 置信度感知与校准: 引入注意力机制, 模型能够实现预测置信度的实时感知, 并依据决策风险自动调节各支路的贡献度. 这种机制通过压制高不确定性的噪声输出, 确保了融合结果在处理复杂边界样本时具备高度的判别一致性.

(3) 鲁棒性增强: 集成策略平衡了模型的精度与泛化能力, 尤其对缺乏明确模式的骗局具有更强的适应性.

这一实验结果验证了多图集成方法在智能合约安全分析中的有效性, 为应对复杂金融欺诈场景提供了可靠的技术路径.

4.5 实验效率与实时性分析

为了评估本研究提出的方法在实际应用中的效率, 我们特别针对从原始字节码构建顺序图、控制流图以及超图, 并基于这 3 个图进行模型推理直至输出最终分类结果的全过程进行了时间开销测量. 这一过程包括两个主要阶段.

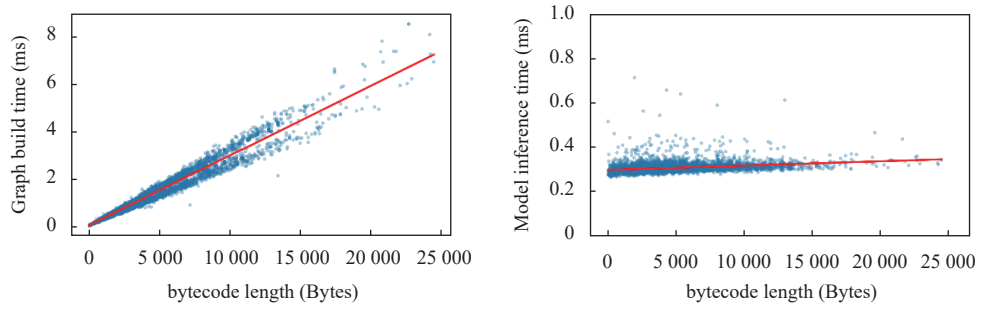
首先是图构建阶段, 即从以太坊智能合约的原始字节码出发, 依次完成以下 3 个子步骤.

(1) 解析字节码并提取操作码序列, 构建反映指令线性执行顺序的顺序图.

(2) 在顺序图基础上识别基本块与跳转逻辑, 生成结构化的控制流图.

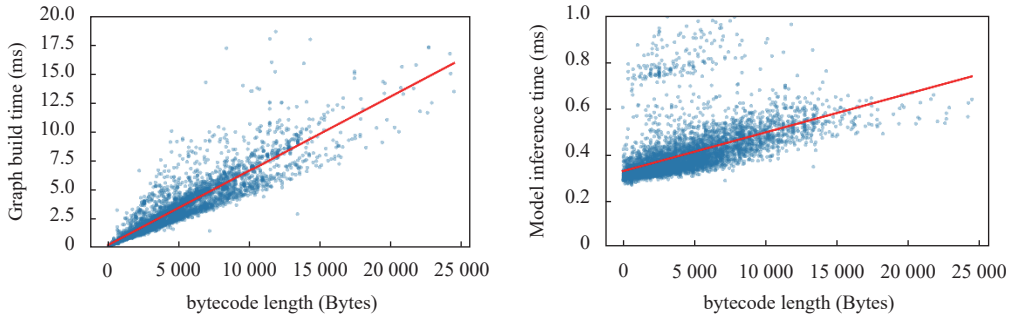
(3) 进一步基于控制流图, 通过滑动窗口策略聚合多跳语义依赖, 构造高阶关联的超图.

其次是模型推理阶段, 将上述构建完成的 3 种图结构分别作为输入, 送入预先训练好的图神经网络分类器, 生成关于该合约是否属于庞氏骗局的预测结果. 为量化各阶段性能, 我们对 6 498 个智能合约样本进行了实验, 其字节码长度覆盖 128–25 000 字节. 如图 11–图 13 所示, 顺序图、控制流图和超图的构建时间均随字节码长度呈近似线性增长, 表明系统具备良好的可扩展性. 具体如下.



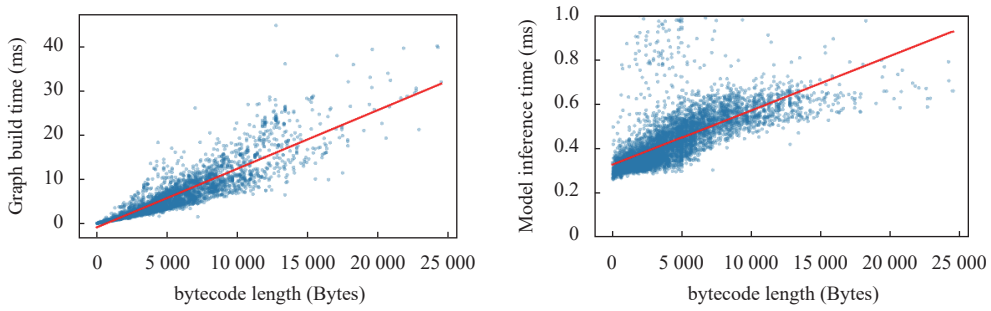
(a) Sequence graph construction time vs. bytecode length (b) Model inference time vs. bytecode length

图 11 顺序图构建及模型推理时间



(a) CFG construction time vs. bytecode length (b) Model inference time vs. bytecode length

图 12 控制流图构建及模型推理时间



(a) Hypergraph construction time vs. bytecode length (b) Model inference time vs. bytecode length

图 13 超图构建及模型推理时间

(1) 顺序图构建 (图 11): 平均耗时约为 0.000 3 ms/Byte, 在 20 KB 合约上约需 6 ms.

(2) 控制流图构建 (图 12): 平均耗时约为 0.000 6 ms/Byte, 在 20 KB 合约上约需 12 ms.

(3) 超图构建 (图 13): 平均耗时约为 0.001 2 ms/Byte, 在 20 KB 合约上约需 24 ms.

此外, 模型推理时间 (model inference time) 几乎不受字节码长度影响, 在绝大多数情况下稳定在 0.3–0.6 ms 范围内, 最大不超过 1 ms. 这表明一旦图结构准备就绪, 分类任务可近乎实时完成. 综合来看, 端到端处理总耗时在典型合约 (≤ 20 KB) 上平均仅为 43 ms. 这一效率对于实现“部署即检测”的实时安全审计具有关键意义. Nikolić 等人[37]对近百万以太坊合约的实证分析显示, 绝大多数漏洞合约在创建后 10 个区块内 (约 2–3 min) 即被首次调用, 表明攻击者会迅速识别并利用新部署的脆弱合约. 因此, 有效的检测机制必须在数分钟内完成分析, 方能在资金损

失发生前发出预警. 相比之下, 本文方法的毫秒级响应速度 (43 ms) 远快于典型的攻击窗口, 完全有能力在任何外部交互或资金注入前完成风险判定, 真正实现事前审计. 该效率水平充分满足以太坊等主流区块链平台对新部署智能合约进行准实时安全筛查的工程需求, 验证了本方法在兼顾高精度与高吞吐能力方面的实用价值.

4.6 对比实验

本研究聚焦于仅依赖字节码的实时庞氏骗局识别问题, 因此在对比实验中, 我们选取了 6 篇代表性工作作为基线方法, 涵盖传统机器学习与最新深度学习两类范式. 所有方法均在相同的合约数据集上运行, 并采用一致的数据划分策略与评估指标, 以确保公平比较. 具体而言, 所选基线包括以下 4 类基于字节码的传统机器学习方法.

文献 [20]: 提取操作码频率特征, 利用 XGBoost 分类器对庞氏骗局合约进行识别.

文献 [23]: 提取智能合约字节码相似性特征, 通过计算两个字节码字符串之间的相似度, 使用改进的 LightGBM 算法对庞氏骗局合约进行识别.

文献 [24]: 利用 *n*-gram 算法直接提取合约操作码特征, 同时引入 ADASYN 来处理类不平衡数据, 利用改进的 AdaBoost 分类器对庞氏骗局合约进行识别.

文献 [25]: 基于智能合约的控制流图 (CFG), 消除图中的无意义元素, 提取了 *n*-gram TF-IDF 特征, 并利用这些特征构建了一个随机森林 (RF) 模型, 用于庞氏骗局的检测.

为进一步验证本方法在深度学习范式下的先进性, 我们还引入两项近期基于图神经网络的工作, 并对其图构建阶段进行了字节码适配 (因原始方法依赖源代码或交易数据, 无法直接应用于本任务).

文献 [30]: 原方法基于交易行为图与合约交互图构建多视图. 我们将其适配为: 从字节码提取操作码序列, 分别构建顺序图与控制流图作为两个语义视图, 保留其多视图对比学习与强化权重分配机制;

文献 [31]: 原方法依赖 Solidity 源代码生成精确的控制流图. 我们将其适配为: 基于字节码反汇编结果自动构建 CFG, 并采用其提出的 GCN 与 Transformer 多头注意力融合架构进行端到端分类.

实验结果如表 1 所示, 其中加粗数值表示各指标下的最优性能. 可以清晰地看出, 本文提出的 MG-DAF 方法在 *Precision*、*Recall* 与 *F*1-score 这 3 项指标上均优于 6 种对比方法, 充分验证了本框架在检测准确性与全面性方面的优势.

表 1 不同方法在基础数据集上的检测性能对比

Methods	*Precision*	*Recall*	*F*1-score
文献[20]	0.9017	0.8035	0.8469
文献[23]	0.9183	0.8576	0.8789
文献[24]	0.9231	0.7792	0.8451
文献[25]	0.9242	0.7922	0.8531
文献[30]	0.9366	0.8369	0.9116
文献[31]	0.8853	0.8698	0.8748
MG-DAF	**0.9872**	**0.8954**	**0.9391**

实验结果表明, 通过深入分析智能合约的字节码特征和图结构信息, 我们的模型能够更精准地识别庞氏骗局的特征模式, 从而实现了检测性能的显著提升. 这些实证数据不仅验证了所提方法的有效性和可靠性, 更凸显了其实际应用价值.

4.7 模型泛化能力测试

为了进一步验证 MG-DAF 框架在处理未知样本及现实场景中的鲁棒性, 本节基于前文 (第 4.1 节) 构建的完全独立的外部数据集 Test_ponzi 开展了泛化性能测试. 该非平衡数据集包含 62 个全新庞氏变种和 200 个未参与训练的隔离正常合约, 最大程度模拟了真实以太坊主网的背景噪声与未知威胁.

本实验将已在基础数据集上训练收敛的 MG-DAF 模型直接应用于 Test_ponzi 集合. 在不进行任何二次微调的前提下, 模型对外部未知样本的检测效能如表 2 所示.

表 2　MG-DAF 在外部泛化数据集上的检测性能

指标	数值
Precision	0.952 4
Recall	0.822 6
*F*1-score	0.883 1

实验数据表明, 即便在面对全新的样本时, MG-DAF 依然展现出了较好的判别能力.

高精确率 (*Precision* 0.952 4): 模型在复杂背景噪声中能够精准锁定欺诈合约, 误报率极低, 证明了多图特征表征在区分正常逻辑与欺诈行为上的严谨性.

稳定的查全率 (*Recall* 0.822 6): 对于 62 个从未见过的庞氏合约, 模型成功识别了其中的绝大部分. 虽然相比内部测试集指标略有波动, 但这一表现足以证明模型捕捉核心欺诈属性的能力.

MG-DAF 能够展现出强泛化性的核心原因在于其内置的动态注意力融合 (DAF) 机制. 在检测这批外部新型样本时, 由于部分合约在顺序维度 (sequence) 或控制流维度 (CFG) 可能进行了伪装, 导致单一模型的判定信心下降. 此时, DAF 模块能够实时感知各分支的置信度变化, 自动调高在该样本上表现更为稳健的图模型的决策权重. 这种“自适应决策”逻辑使得框架在面对未知的庞氏骗局时, 具备了极强的适应能力和实战价值, 为复杂金融场景下的智能合约安全分析提供了可靠的技术路径.

5 总　结

为了克服当前智能合约庞氏骗局识别存在的局限性, 本文提出基于字节码的庞氏骗局实时识别框架. 通过构建智能合约操作码的 3 种图结构并进行特征提取, 成功实现了对智能合约庞氏骗局的高效、实时检测. 具体而言, 我们首先基于操作码的出现顺序构建了操作码顺序图, 直观地展示了合约中操作码之间的序列连接关系; 其次, 引入软件工程中的控制流代码块概念, 将操作码划分为多个控制流单元, 并根据其跳转逻辑构建了控制流图; 最后, 利用超图结构综合表示控制流代码块之间更高层次的语义关系. 在分类模型构建方面, 我们分别采用了图卷积神经网络与超图卷积神经网络对不同图结构进行学习, 并构建集成学习框架 MG-DAF. 实验结果表明, 所提出的 MG-DAF 方法在 *Precision* (0.987 2)、*Recall* (0.895 4) 和 *F*1-score (0.939 1) 等指标上均取得了显著提升. 这一结果表明, 3 层次图结构能够从多维角度全面捕捉智能合约的内在特征, 为庞氏骗局的有效识别提供了强有力的支持.

在未来工作中, 我们将从 3 个方面进一步完善本框架, 增强模型可解释性. 通过图结构可视化与关键路径溯源, 揭示模型识别庞氏行为的具体依据; 优化图表示学习能力, 探索动态基本块划分、上下文感知的超边生成等策略, 并引入注意力机制等先进图神经网络架构, 以更精准地建模高阶欺诈模式; 提升泛化能力, 计划收集 BSC、Polygon 等多链智能合约数据, 构建跨链评估基准, 验证方法在不同区块链环境下的适用性与鲁棒性.

References

[1] Tiwari N, Ranjan P, Biswal PK, Barde C, Sinha N. Survey and analysis of blockchain technologies with respect to: Properties, algorithms, architecture, models, evolution and framework. Multimedia Tools and Applications, 2025, 84(14): 13571–13615. [doi: 10.1007/s11042-024-19580-3]

[2] Vijayakumar G, Singh K, SK K. Privacy preserving decentralized swap derivative with deep learning based oracles leveraging blockchain technology and cryptographic primitives. Computers and Electrical Engineering, 2024, 119: 109510. [doi: 10.1016/j.compeleceng.2024.109510]

[3] Panda SK, Sathya AR, Das S. Bitcoin: Beginning of the cryptocurrency era. In: Panda SK, Mishra V, Dash SP, Pani AK, eds. Recent Advances in Blockchain Technology: Real-world Applications. Cham: Springer, 2023. 25–58. [doi: 10.1007/978-3-031-22835-3_2]

[4] Anand VR, Alagiri I, Jayalakshmi P, Brahmam MG, Abdullah AB. Lattice homomorphic assisted privacy preserving electronic health records data transmission in Internet of medical things using blockchain. Trans. on Emerging Telecommunications Technologies, 2025, 36(2): e70070. [doi: 10.1002/ett.70070]

[5] Yuan SQ, Yang WZ, Tian XD, Tang WJ. A blockchain-based privacy preserving intellectual property authentication method. Symmetry, 2024, 16(5): 622. [doi: 10.3390/sym16050622]

[6] Hejazi N, Lashkari AH. A comprehensive survey of smart contracts vulnerability detection tools: Techniques and methodologies. Journal of Network and Computer Applications, 2025, 237: 104142. [doi: 10.1016/j.jnca.2025.104142]

[7] Crisostomo J, Bacao F, Lobo V. Machine learning methods for detecting smart contracts vulnerabilities within Ethereum blockchain—A review. Expert Systems With Applications, 2025, 268: 126353. [doi: 10.1016/j.eswa.2024.126353]

[8] Bongini PA, Mattassoglio F, Pedrazzoli A, Vismara S. Crypto ecosystem: Navigating the past, present, and future of decentralized finance. The Journal of Technology Transfer, 2025, 50(5): 2054–2075. [doi: 10.1007/s10961-025-10186-x]

[9] Bhadra S, Singh KN. Ponzi scheme like investment schemes in India—Causes, impact and solution. Journal of Money Laundering Control, 2024, 27(2): 348–362. [doi: 10.1108/JMLC-02-2023-0040]

[10] Feng X, Shi QC, Li XY, Liu HY, Wang LM. IDPonzi: An interpretable detection model for identifying smart Ponzi schemes. Engineering Applications of Artificial Intelligence, 2024, 136: 108868. [doi: 10.1016/j.engappai.2024.108868]

[11] Chen DJ, Liao ZY, Chen RD, Wang H, Yu C, Zhang K, Zhang N, Shen XM. Privacy-preserving anomaly detection of encrypted smart contract for blockchain-based data trading. IEEE Trans. on Dependable and Secure Computing, 2024, 21(5): 4510–4525. [doi: 10.1109/TDSC.2024.3353827]

[12] Lei G, Zhang DH, Xiao JM, Fan GD, Cao YL, Feng ZY. FMCF: A fusing multiple code features approach based on Transformer for Solidity smart contracts source code summarization. Applied Soft Computing, 2024, 166: 112238. [doi: 10.1016/j.asoc.2024.112238]

[13] Chen JC, Xia X, Lo D, Grundy J, Luo XP, Chen T. DefectChecker: Automated smart contract defect detection by analyzing EVM bytecode. IEEE Trans. on Software Engineering, 2022, 48(7): 2189–2207. [doi: 10.1109/TSE.2021.3054928]

[14] Aljabri A, Jemili F, Korbaa O. Convolutional neural network for intrusion detection using blockchain technology. Int'l Journal of Computers and Applications, 2024, 46(2): 67–77. [doi: 10.1080/1206212X.2023.2284443]

[15] Ashfaq T, Khalid R, Yahaya AS, Aslam S, Azar AT, Alsafari S, Hameed IA. A machine learning and blockchain based efficient fraud detection mechanism. Sensors, 2022, 22(19): 7162. [doi: 10.3390/s22197162]

[16] Chen WL, Zheng ZB, Ngai ECH, Zheng PL, Zhou YR. Exploiting blockchain data to detect smart Ponzi schemes on Ethereum. IEEE Access, 2019, 7: 37575–37586. [doi: 10.1109/ACCESS.2019.2905769]

[17] Bartoletti M, Carta S, Cimoli T, Saia R. Dissecting Ponzi schemes on Ethereum: Identification, analysis, and impact. Future Generation Computer Systems, 2020, 102: 259–277. [doi: 10.1016/j.future.2019.08.014]

[18] Lu PC, Cai L, Yin KT. SourceP: Detecting Ponzi schemes on Ethereum with source code. In: Proc. of the 2024 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP). Seoul: IEEE, 2024. 4465–4469. [doi: 10.1109/ICASSP48485.2024.10448439]

[19] Chen WM, Li XR, Sui Y, He NY, Wang HY, Wu L, Luo XP. SADPonzi: Detecting and characterizing Ponzi schemes in Ethereum smart contracts. Proc. of the ACM on Measurement and Analysis of Computing Systems, 2021, 5(2): 26. [doi: 10.1145/3460093]

[20] Chen WL, Zheng ZB, Cui JH, Ngai E, Zheng PL, Zhou YR. Detecting Ponzi schemes on Ethereum: Towards healthier blockchain technology. In: Proc. of the 2018 World Wide Web Conf. Lyon: Int'l World Wide Web Conf. Steering Committee, 2018. 1409–1418. [doi: 10.1145/3178876.3186046]

[21] Wang L, Cheng H, Zheng ZB, Yang AJ, Xu M. Temporal transaction information-aware Ponzi scheme detection for Ethereum smart contracts. Engineering Applications of Artificial Intelligence, 2023, 126: 107022. [doi: 10.1016/j.engappai.2023.107022]

[22] Wang L, Cheng H, Zheng ZB, Yang AJ, Zhu XH. Ponzi scheme detection via oversampling-based long short-term memory for smart contracts. Knowledge-based Systems, 2021, 228: 107312. [doi: 10.1016/j.knosys.2021.107312]

[23] Zhang YM, Yu WQ, Li ZY, Raza S, Cao HH. Detecting Ethereum Ponzi schemes based on improved LightGBM algorithm. IEEE Trans. on Computational Social Systems, 2022, 9(2): 624–637. [doi: 10.1109/TCSS.2021.3088145]

[24] Wang MX, Huang J. Detecting Ethereum Ponzi schemes through opcode context analysis and oversampling-based AdaBoost algorithm. Computer Systems Science and Engineering, 2023, 47(1): 1023–1042. [doi: 10.32604/csse.2023.039569]

[25] Ji SH, Huang CX, Zhang PC, Dong H, Xiao Y. Ponzi scheme detection based on control flow graph feature extraction. In: Proc. of the 2023 IEEE Int'l Conf. on Web Services (ICWS). Chicago: IEEE, 2023. 585–594. [doi: 10.1109/ICWS60048.2023.00077]

[26] Zheng ZB, Chen WL, Zhong ZJ, Chen ZG, Lu YT. Securing the Ethereum from smart Ponzi schemes: Identification using static features. ACM Trans. on Software Engineering and Methodology, 2023, 32(5): 130. [doi: 10.1145/3571847]

[27] Wu JJ, Yuan Q, Lin D, You W, Chen WL, Chen C, Zheng ZB. Who are the phishers? Phishing scam detection on Ethereum via network embedding. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(2): 1156–1166. [doi: 10.1109/TSMC.2020.3016821]

[28] Tan RN, Tan QF, Zhang Q, Zhang P, Xie YS, Li Z. Ethereum fraud behavior detection based on graph neural networks. Computing, 2023, 105(10): 2143–2170. [doi: 10.1007/s00607-023-01177-7]

[29] Yu T, Chen XM, Xu Z, Xu JL. MP-GCN: A phishing nodes detection approach via graph convolution network for Ethereum. Applied

Sciences, 2022, 12(14): 7294. [doi: 10.3390/app12147294]

[30] Jiang XF, Tsai WT. MVCG-SPS: A multi-view contrastive graph neural network for smart Ponzi scheme detection. Applied Sciences, 2025, 15(6): 3281. [doi: 10.3390/app15063281]

[31] Huang J, Wang MX, Han HG. Ponzi scheme contract detection based on code control flow graph. Ruan Jian Xue Bao/Journal of Software, 2025, 36(10): 4628–4644 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7318.htm [doi: 10.13328/j.cnki.jos.007318]

[32] Chen JZ, Liu JL, Wu JL, Lin D, Wu JJ, Zheng ZB. PonziHunter: Hunting Ethereum Ponzi contract via static analysis and contrastive learning on the bytecode level. ACM Trans. on Software Engineering and Methodology, 2026, 35(2): 37. [doi: 10.1145/3735971]

[33] Wood G. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 2014. 1–32.

[34] Feng YF, You HX, Zhang ZZ, Ji RR, Gao Y. Hypergraph neural networks. In: Proc. of the 2019 AAAI Conf. on Artificial Intelligence. Honolulu: AAAI, 2019. 3558–3565. [doi: 10.1609/aaai.v33i01.33013558]

[35] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

[36] Wang X, Ji HY, Shi C, Wang B, Ye YF, Cui P, Yu PS. Heterogeneous graph attention network. In: Proc. of the 2019 World Wide Web Conf. San Francisco: ACM, 2019. 2022–2032. [doi: 10.1145/3308558.3313562]

[37] Nikolić I, Kolluri A, Sergey I, Saxena P, Hobor A. Finding the greedy, prodigal, and suicidal contracts at scale. In: Proc. of the 34th Annual Computer Security Applications Conf. San Juan: ACM, 2018. 653–663. [doi: 10.1145/3274694.3274743]

附中文参考文献

[31] 黄静, 王梦晓, 韩红桂. 基于代码控制流图的庞氏骗局合约检测. 软件学报, 2025, 36(10): 4628–4644. http://www.jos.org.cn/1000-9825/7318.htm [doi: 10.13328/j.cnki.jos.007318]

作者简介

陈伟利, 博士, 副教授, 主要研究领域为区块链, 数据挖掘, 人工智能.

竹超, 硕士, 主要研究领域为机器学习, 庞氏骗局智能合约检测.

肖桂姣, 讲师, 主要研究领域为加密数字货币, 数字经济.

唐明董, 博士, 教授, CCF 高级会员, 主要研究领域为服务计算, 区块链, 数据挖掘.