# 以太坊庞氏骗局智能合约的早期检测方法研究

张艳梅[1]，郭思颖[1,2]，贾恒越[1]，姜茸[3]
（1.中央财经大学信息学院，北京 100081；2.四川农商联合银行信息科技部，四川 成都 610041；
3.云南财经大学云南省服务计算重点实验室，云南 昆明 650221）

**摘　要：**以太坊是区块链的典型应用代表，它允许开发者创建和执行智能合约。以太坊技术的迅猛发展在推动智能合约普及的同时，也引发链上安全风险剧增，其中算法驱动的智能庞氏骗局给区块链应用带来了新的安全挑战。为了实现对智能合约庞氏骗局的早期检测，提出了一种基于图卷积网络（GCN）的检测方法PonziGCN。该方法融合了智能合约的语义特征和控制流图特征，通过提取字节码相似度、操作码频率等语义特征，以及控制流图的基本特征和结构特征，构建了多特征融合的检测框架。实验结果表明，所提方法在精确率、召回率、F值和AUC值等关键性能指标上均表现优异，精确率达到0.982，召回率为0.987，F值为0.978，AUC值为0.983，显著优于现有的算法。特征重要性分析表明，图结构特征和代码中与交易功能相关的操作码频率特征在模型中具有最高的重要性。

**关键词：**以太坊；智能合约；庞氏骗局；图卷积神经网络；控制流图
**中图分类号：**TP393
**文献标志码：**A
**DOI：**10.11959/j.issn.1000–436x.2025156

# Detection of smart contract Ponzi schemes based on graph convolutional neural networks

ZHANG Yanmei[1], GUO Siying[1,2], JIA Hengyue[1], JIANG Rong[3]
1. School of Information, Central University of Finance and Economics, Beijing 100081, China
2. Information Technology Department, Sichuan Rural Commercial Bank, Chengdu 610041, China
3. Yunnan Key Laboratory of Service Computing, Yunnan University of Finance and Economics, Kunming 650221, China

**Abstract:** Ethereum is an application of blockchain that allows developers to create and execute smart contracts. With the rapid development of Ethereum technology, the widespread application of smart contracts has introduced new security challenges, particularly the proliferation of fraudulent contracts such as Ponzi schemes. To achieve early detection of Ponzi schemes in smart contracts, a detection model named PonziGCN based on graph convolutional network (GCN) was proposed. The model integrated semantic features and control flow graph features of smart contracts. By extracting semantic features such as bytecode similarity and opcode frequency, as well as basic and structural features of control flow graphs, a multi-feature fusion detection framework was constructed. Experimental results demonstrate that the PonziGCN model performed excellently in key performance metrics, achieving an accuracy of 0.982, a recall of 0.987, an F-score of 0.978, and an AUC value of 0.983, significantly outperforming existing advanced algorithms. Feature importance analysis indicates that graph structural features and opcode frequency features related to transaction functions in the code hold the highest importance in the model.

**Keywords:** Ethereum, smart contract, Ponzi scheme, graph convolutional neural network, control flow graph

**收稿日期：**2025–05–12；**修回日期：**2025–08–13
**通信作者：**张艳梅，Zhangym@cufe.edu.cn
**基金项目：**国家重点研发计划基金资助项目（No.2024YFC3308100）；国家自然科学基金资助项目（No.62372493）
**Foundation Items：**The National Key Research and Development Program of China (No.2024YFC3308100), The National Natural Science Foundation of China (No.62372493)

## 0　引言

随着区块链技术的兴起，智能合约作为其核心应用之一，已经在金融、法律、物联网等多个领域展现出巨大的应用潜力。区块链的典型应用代表以太坊是建立在区块链技术之上的一种去中心化平台。作为区块链的典型应用代表，以太坊不仅支持加密货币交易，还允许开发者创建和执行智能合约。智能合约[1]是一种自动执行、自我验证的程序，它能够在预定条件得到满足时自动执行合约条款，无需中介机构的介入。根据 DappRadar 的统计，目前已有约4 450个数据应用程序和超191 000个智能合同。然而，正如任何新兴技术一样，智能合约的普及也伴随着一系列新的挑战和风险，尤其是智能合约漏洞和欺诈合约，为区块链犯罪提供了可乘之机。

欺诈性合约，如庞氏骗局和蜜罐合约，利用智能合约漏洞对投资者和用户的资产安全构成威胁。庞氏骗局合约是欺诈性合约的一个典型例子，它们利用智能合约的自动化执行特性和交易数据的披露来误导参与者获取信任。这类合约的执行并不产生实际价值，而是通过新投资者的资金来支付旧投资者的回报。随着时间的推移，这种模式难以持续吸引新投资者，最终导致合约崩溃，给大多数投资者带来损失。根据Chainalysis发布的加密犯罪报告，庞氏骗局已经造成了高达数百亿美元的损失，其中一些骗局的涉案金额甚至达到了数十亿美元。因此，建立准确高效的智能合约庞氏骗局早期识别模型显得尤为重要，能够实现对潜在庞氏骗局的快速检测和预警，为投资者提供风险评估信息，提高区块链技术的公信力，促进其在更广泛领域的健康发展。

智能合约发展初期，部分学者使用字节码特征或操作码特征计算待测合约与已知庞氏骗局合约的字节码或操作码相似度[2-3]。随后，有学者从字节码中提取合约运行时图信息，将庞氏合约的检测制定为图分类任务进行识别[4-5]，从而提升了检测效果。在合约执行过程中，产生了大量的交易信息，部分学者利用比特币地址、交易和合约调用信息等交易信息，捕捉庞氏骗局交易账户和模式的特征[6-7]。由于仅利用交易信息的庞氏骗局检测准确度并不高，因此不少学者提出了多特征融合的方法，结合了代码特征和交易信息特征[8-9]。但是，目前的研究方法尚存在如下问题：1) 一些相关研究用到了交易特征，而交易特征是合约执行后产生的，资金已经开始流动，识别骗局并采取措施无法挽回已经造成的损失，因此如何仅使用代码特征在交易发生之前进行庞氏骗局早期检测是研究重点；2) 使用字节码、操作码挖掘代码语义信息的相关研究忽略了程序执行的控制流与数据流信息；3) 利用图相似度等方法提取图特征的相关研究忽略了字节码、操作码中的语义信息，准确度有待提升。

因此，本文提出一种充分利用并融合了智能合约代码的语义信息特征与控制流图特征的检测方法。本文的主要贡献如下。

1)提出一种智能合约控制流图特征的提取方法。该方法通过反编译字节码构建控制流图（CFG），并从中提取图的基本特征和结构特征，从而准确刻画了合约的执行路径和逻辑结构特性。

2)提出融合字节码的语义特征和控制流图特征的庞氏骗局合约检测方法。对智能合约的代码特征进行了全面的捕捉与深度融合，增强特征的表达力，以提高对庞氏骗局的早期检测能力。

3)提出基于图卷积网络（GCN）的检测模型，该模型既能从智能合约字节码中提取复杂语义特征，又能有效处理图结构数据，将多特征融合以有效识别庞氏骗局合约。实验结果表明，本文方法在精确率、召回率、F值、和AUC值等方面均优于现有的相关算法。

## 1　相关工作

### 1.1　智能合约代码特征研究

智能合约是一套以数字形式定义的承诺，包括合约参与方可以在上面执行这些承诺的协议[10]，本质是运行在区块链上的计算机程序。因此，目前对区块链智能合约安全的代码特征研究主要分为2类：代码的安全性和隐藏于代码中的欺诈行为。

智能合约作为一段程序代码，在设计和开发的过程中难免会面临代码安全方面的挑战。早期智能合约漏洞检测工作主要采用形式化验证、符号执行、模糊检测等方法检测智能合约漏洞。随着深度学习的发展，逐渐有学者将深度学习的方法运用于漏洞检测。文献[11]利用具有注意力机制的双向长短期记忆（BiLSTM-ATT）模型，检测了可重入漏洞，还提出了智能合约的合约片段表示，有助于捕

获基本的语义信息和控制流依赖关系。文献[12]将源代码转换为合约图，使用一种新的时序消息传播（TMP）网络[13]提取图特征，结合专家模式进行智能合约漏洞检测。随着人工智能技术的发展，也有学者将大模型方法运用到漏洞检测中。文献[14]通过实证研究，评估了ChatGPT在智能合约漏洞检测中的应用，为理解大型语言模型在智能合约漏洞检测中的潜力和挑战提供了宝贵见解。

隐藏于代码中的欺诈行为则是智能合约拥有者在合约代码里设计了欺诈性策略，具体而言就是庞氏骗局和蜜罐合约。文献[15]抓取了比特币论坛的帖子，分析了庞氏骗局的特征和受害者的特征。文献[16]首次对蜜罐智能合约进行了系统分析，研究了蜜罐智能合约的流行程度、行为和对以太坊区块链的影响。文献[17]使用字节码为特征，提出了基于N-gram特征和LightGBM的机器学习模型来检测蜜罐合约。文献[18]在使用字节码特征的同时还加入操作码特征和交易特征，使用SMOTE + Tomek方法改进的LightGBM模型进行智能合约庞氏骗局识别。智能合约运行过程中存在丰富的语义和数据控制流关系，文献[5]则使用字节码特征，提出合约运行时行为图（CRBG）将庞氏合约的检测制定为图分类任务，赋能图神经网络（GNN）进行GRBG分析，可以在交易发生之前预先识别庞氏合约。

综上，代码本身的安全漏洞问题和隐藏于代码间的欺诈识别都是智能合约代码特征研究的重点。新的漏洞层出不穷，如何找到更为准确的漏洞检测方法识别新型漏洞将是未来的主要研究方向。隐藏于代码的欺诈合约并非漏洞合约，而是主观故意为之，通常有庞氏骗局合约和蜜罐合约之分。目前针对庞氏骗局合约的研究较多，检测方法主要基于字节码特征和交易特征。然而，庞氏骗局合约目前开源较少，大多研究只能使用字节码和操作码进行匹配，忽略了智能合约运行过程中丰富的控制和数据流语义。探索智能合约中的图特征，挖掘庞氏骗局中的控制流和数据流关系特征是值得深入的研究方向。

### 1.2 智能合约代码溯源研究

智能合约代码溯源是指通过反编译技术，将字节码转换回控制流图或类似源代码的伪代码，从而分析合约的功能和逻辑。以太坊智能合约涉及用户的安全性与隐私性，因此智能合约源代码少有开源，而欺诈智能合约开源代码更为稀少，对于欺诈的审计缺少现成可用源代码，同时字节码和操作码可提供的信息有限，使智能合约代码溯源较为困难；但也并非无计可施，有不少学者从逆向工程领域得到启发，从智能合约字节码中还原出了高级伪代码或控制流结构信息等更适合分析的信息。

文献[19]开发了一种智能合约逆向工程工具，将智能合约字节码恢复为适合手动分析的高级伪代码，从而可以降低以太坊智能合约生态系统中的整体不透明度。文献[20]利用以太坊虚拟机处理函数的方式，从合约字节码中自动恢复函数签名，达到了98.7%的精确率。在此基础上，文献[21]使用深度学习方法进行了改进，对函数签名和返回进行了恢复，并得到了更高的精确率和更快的执行速度。

智能合约欺诈行为的溯源涉及区块链账户的隐私与安全，较多关键代码并未开源，因此对于审计者来说，智能合约的不透明性较高。但是相较于单一的智能合约字节码，智能合约溯源可提供给审计者的信息更为丰富，有助于更准确快速地识别出欺诈的智能合约，是非常具有潜力的研究方向。

### 1.3 庞氏骗局检测方法研究

庞氏骗局是一种欺诈模式，依靠新投资者资金支付旧投资者回报，最终导致金字塔崩溃，一旦执行，资金无法追回。因此，开发准确高效的庞氏骗局检测技术是区块链安全领域的关键研究方向。目前，庞氏骗局的检测方法主要分为以下3种：基于代码检测、基于交易信息检测和融合检测。

目前庞氏骗局检测大多使用字节码特征或操作码特征，少部分文献使用源代码特征。文献[2]使用字节码为特征，构建了一个庞氏骗局的数据集，使用蒙特卡罗算法来估计以太坊区块链上以太坊虚拟机（EVM）合约与已知庞氏骗局合约之间的字节码相似度。文献[3]将字节码视为字符串，借鉴自然语言处理文本特征提取的思想，测量待测合约与庞氏骗局合约集合字节码的相似性，构建了基于改进的CatBoost算法的庞氏骗局合约分类模型。

由于区块链的匿名性，可获取的交易信息不多，仅基于交易信息的庞氏骗局检测准确度并不高。文献[6]检索与庞氏骗局相关的比特币地址集合，通过多输入启发式的地址聚类方法，设计了一组可用于描述庞氏骗局的特征，提出了基于监督学习算法的比特币庞氏骗局的自动分析。文献[7]提出了一

种通用的异构特征增强模块，在包含协调交易和合约调用信息的辅助异构图上提取基于元路径的特征，将与行为模式相关的异构特征聚合到同构图中的相应账户节点，通过特征增强来提高现有庞氏检测方法的性能。但该方法单独使用可行性不高，需要与其他检测方法结合。

在庞氏骗局检测中，多特征融合的方法更为适用，大多数研究都结合了代码特征和交易信息特征。文献[9]从交易历史和操作码中提取 2 种特征，使用随机森林检测智能合约庞氏骗局。实验结果还表明交易账户特征无法单独使用但有助于提升模型精度。文献[8]在文献[9]的基础上，从字节码、语义、开发者等多个视图中提取大量特征，使用多视图级联集成方法，在以太坊上实时检测智能庞氏骗局而不依赖合约交互信息，该模型在 F 值和鲁棒性方面优于最先进的方法和许多其他基线方法。

综上，目前已有许多庞氏骗局检测方法的相关研究，但如何更加全面地提取字节码、操作码、交易特征进行庞氏骗局早期检测，还有待深入研究，精确率有待提高。

## 2　基于图卷积神经网络的庞氏骗局检测

本研究使用智能合约字节码作为原始数据，在特征选取方面分为 2 个部分。第 1 部分关注代码本身的语义特征，包括字节码的相似度特征和操作码频率特征。智能合约的字节码包含了合约逻辑的直接映射，比较不同合约的字节码相似度，可以识别出合约之间的逻辑相似性，揭示庞氏骗局合约之间可能的复制或模仿行为，对于检测那些试图通过微小修改来逃避检测的欺诈合约尤其有效。操作码是由字节码反编译而来，构成合约逻辑的基本指令集，庞氏骗局合约通常会使用特定的操作码模式来实现其欺诈机制，如频繁使用发送以太币的操作码及使用条件分支操作码来控制资金流向等。通过分析操作码的使用频率，可以识别出这些模式，从而为检测欺诈合约提供依据。第 2 部分关注代码的控制流图特征，包括控制流图基本特征和结构特征。受到软件工程领域逆向工程方法的启发，将其应用于智能合约庞氏骗局的识别。逆向工程是一种分析技术，通过对软件二进制文件表示进行分析，从而提取出系统的功能、结构和逻辑[22]。在智能合约的背景下，将这一概念应用于字节码文件，通过反编译，构建 CFG，还原合约代码的执行路径和逻辑结构。基于 CFG，一方面提取了图的基本特征，包括节点数、边数和平均度数，帮助评估合约的规模和复杂度，而平均度数则可以指示节点间的平均连接密度，这与合约的安全性和稳定性相关。另一方面，进一步深入分析 CFG 的结构特征，采用了 Node2vec 图嵌入算法，模拟随机游走来捕捉节点的上下文信息，生成代表节点特征的向量，输入 GCN 进行平均池化处理聚合每个节点的信息，得到图结构特征向量。这种图结构特征不仅包含了节点的局部邻域信息，还通过 GCN 的层次化结构捕捉到了更全局的结构特征，能够提供关于智能合约 CFG 全局结构的深入洞察。因此本研究使用智能合约代码特征进行庞氏骗局识别，融合语义特征和控制流图特征，提出一种基于 GCN 的庞氏骗局检测模型 PonziGCN。其中语义特征包含操作码频率特征和字节码相似度特征；控制流图特征包含图基本特征和图结构特征。

PonziGCN 检测流程如图 1 所示，整个流程可以分为 3 个阶段：数据收集、特征提取以及模型搭建与预测分类。在数据收集阶段，使用 etherscan.io 平台作为数据源，收集智能合约字节码和操作码数据，然后将收集到的字节码转换为 CFG。在特征提取阶段，提取语义特征和图特征。语义特征包括操作码频率特征和字节码相似度特征，对操作码进行频率统计与特征筛选提取操作码频率特征；对字节码采用 Levenshtein 距离算法计算字节码之间的距离分数，以此作为字节码相似度特征。图特征包括控制流图基本特征和结构特征，基本特征包括 CFG 图节点数量、边数量、平均度量等图基本特征；结构特征利用 Node2Vec 算法对 CFG 图进行随机游走序列生成，并通过 GCN 平均池化，生成图结构特征。在模型搭建与预测分类阶段，首先使用上采样对数据集中多数类和少数类样本进行分离，以解决类别不平衡问题。然后将提取的语义特征和图特征进行线性融合，转换为 GNN 数据格式，形成综合特征矩阵。最后，将预处理后的数据集分为 2 个部分，即 80% 的数据作为分类模型中的训练集，20% 的数据作为测试集，使用 train_test_split 保证划分的随机性。在所有实验中，训练集和测试集来自相互独立的样本以避免数据泄露，训练后预测测试集以获得最优分类结果。特征提取过程和本文方

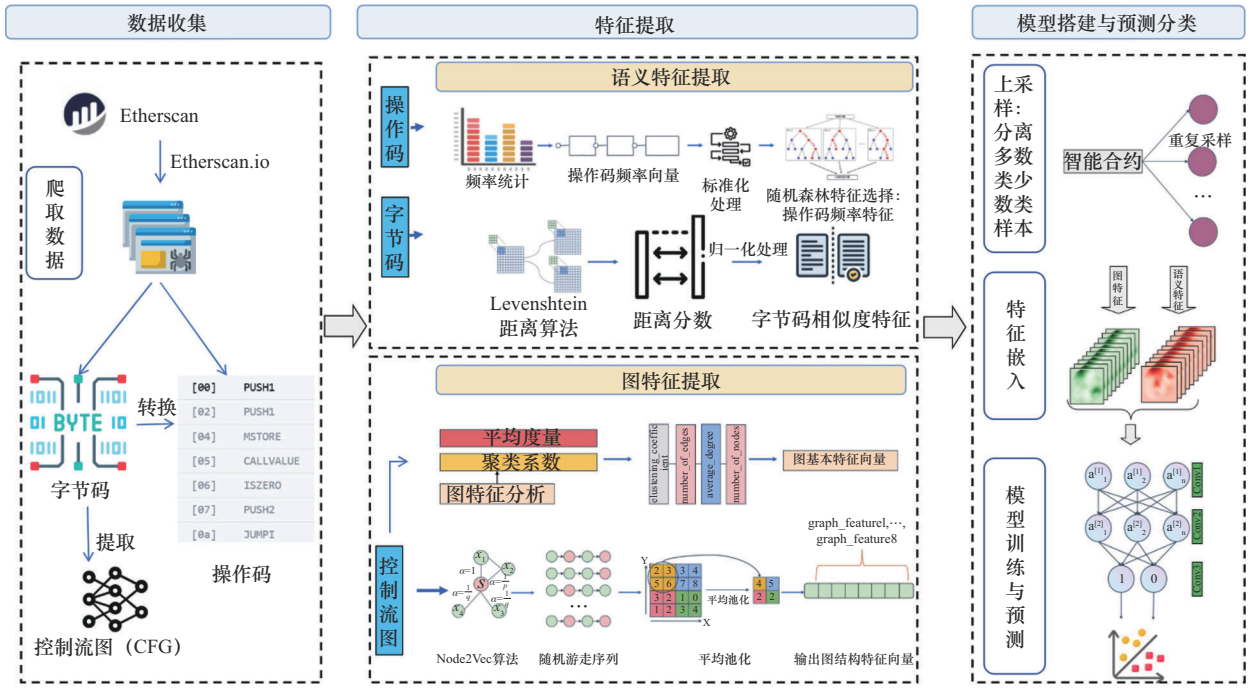


图 1　PonziGCN 检测流程

法将在下文中详细描述。

## 2.1　数据收集

### 2.1.1　爬取数据

为了保证数据集的实时性，使用爬虫从 etherscan.io 网站爬取相关的公开信息（包括智能合约的字节码、操作码）。为了确保数据集标签的可靠性，使用了与文献[18]相同的数据集标签爬取数据，同时结合了文献中以太坊智能合约的地址和标签，形成了一个新的数据集。数据集包括来自 etherchain.org 和 etherscan.io 的 3 774 份智能合约的字节码、操作码信息。

### 2.1.2　数据收集与转换

在区块链领域，智能合约的字节码（bytecode）和操作码（opcode）是 2 个核心概念。智能合约的源代码首先被编写在高级语言如 Solidity 中，然后通过以太坊编译器编译成字节码，每个字节码指令都对应着以太坊虚拟机中的一个操作。

操作码是构成字节码的基本指令，它们是 EVM 可以理解和执行的原子操作。不同的字节码值都对应了特定的操作码，每种操作码都有其特定的功能，它们共同定义了智能合约的行为。字节码和操作码之间可以通过以太坊区块链提供的工具和 API 进行转换。

在智能合约漏洞检测领域，CFG 广泛应用于智能合约审计，因为智能合约源代码样本量庞大，使用源代码构建 CFG 以分析代码中丰富的控制和数据流语义。而在庞氏骗局检测领域，由于严重缺少源代码样本，选择使用字节码文件，参考操作码信息与逆向工程方法，进行字节码的 CFG 还原构建。具体步骤如下。

第 1 步，基本块分区，将十六进制字符串分解为 EVM 指令，然后将这些指令划分为基本块。第 2 步，从基本块恢复代码图。第 3 步，消除冗余代码。第 4 步，聚合表达，替换许多指令进一步简化生成的中间表示。最后生成的 CFG 部分结构如图 2 所示。在下一部分将会对 CFG 进行进一步的分析与特征提取。

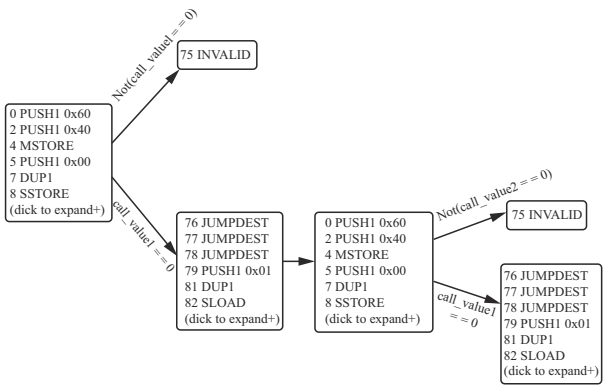


图 2　CFG 图示例

## 2.2　特征提取

### 2.2.1　语义特征

本节深入分析智能合约的执行逻辑和结构特性，以提取 2 种关键特征，即操作码频率特征和字节码相似度特征，进行多维度线性融合，得到语义特征。操作码频率特征反映了合约中各操作码出现的频次；而字节码相似度特征则量化了合约字节码之间的相似性，这有助于揭示合约间的潜在联系和功能相似性。具体操作方法如下。

1)操作码频率特征

操作码指令的类型是人为规定且有限的，因此对每个操作指令在合同操作码文件中出现的次数进行计算，得到每个合约的操作码频率向量，向量的长度等于操作码的种类数，每个元素代表相应操作码的出现频率。庞氏骗局和非庞氏骗局合约操作码频率如图 3 所示，其中，“PUSH”在 2 类合约中出现的频率最高，占比几乎重叠，“PUSH”用于将数据压入虚拟机的栈上，因此在任何合约上使用频率都非常高。DUP、SWAP、 JUMP、JUMPDEST 是与智能合约执行流程相关的操作码，决定合约下一步的跳转位置，这在庞氏骗局和非庞氏骗局合约之间出现了较大的差异性，而其后的很多操作码都显示出了庞氏骗局独有的特征。

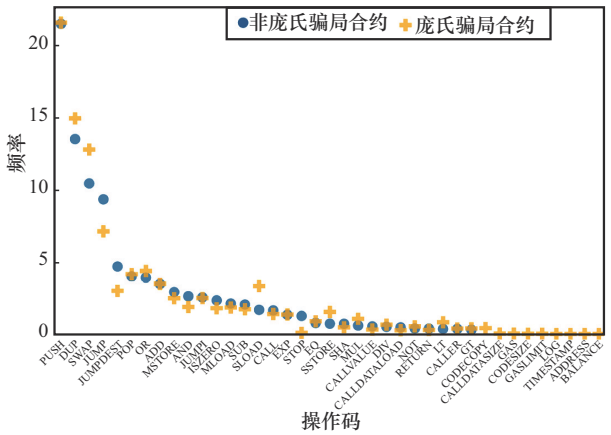


图 3　庞氏骗局和非庞氏骗局合约操作码频率

由于数据集并不平衡，庞氏骗局合约只占少数，无法用同一量纲对操作码频率进行衡量，因此需要对特征向量进行标准化处理，以消除不同操作码频率之间的量纲差异。同时，目前有超 140 个不同的操作码，将这些操作码全部输入模型维度过大，因此需要使用随机森林模型对操作码频率特征进行选择，去除那些对模型贡献较小的特征，最后只保留了 64 维操作码特征，以提高模型的效率和准确性。

2)字节码相似度特征

字节码是智能合约源代码编译后的字符串，难以直观地展示智能合约的功能，因此，传统方法手动分析很难从字节码中提取特征。然而，字节码是源代码编译的产物，如果源代码相似，则合约内函数将相似，对于庞氏骗局合约的识别具有重要作用。由此可以将智能合约字节码作为字符串进行处理，计算 2 个字节码字符串之间的相似性。

使用 Levenshtein 距离（NLD）算法来计算 2 个字节码字符串之间的相似度。Levenshtein 距离通过计算从一个字符串转换到另一个字符串所需的最少单字符编辑操作（插入、删除或替换）来衡量相似度。给定 2 个字节码字符串 $a$ 和 $b$，长度分别为 $m$ 和 $n$，创建一个二维数组 $\boldsymbol{D}$ 来存储中间结果。$\boldsymbol{D}[i][j]$ 表示将源字符串的前 $i$ 个字符转换为目标字符串的前 $j$ 个字符所需的最小编辑距离。对于任意 $i$，$\boldsymbol{D}[i][0]=i$ 表示将源字符串的前 $i$ 个字符变成空字符串所需的操作数是 $i$ 次删除操作。对于任意 $j$，$\boldsymbol{D}[0][j]=j$ 表示将源字符串的前 $j$ 个字符变成空字符串所需的操作数是 $j$ 次插入操作。递推公式为

$$\boldsymbol{D}[i][j]=\min\begin{cases}\boldsymbol{D}[i-1][j]+1\\ \boldsymbol{D}[i][j-1]+1\\ \boldsymbol{D}[i-1][j-1]+\operatorname{cost}\left(a_i,b_j\right)\end{cases}\tag{1}$$

其中，第 1 个算式是删除操作，代表从源字符串中删除一个字符；第 2 个算式是插入操作，代表在目标字符串中插入一个字符；第 3 个算式是替换操作，代表如果字符不相等，则替换为目标字符；cost 是一个判定字符是否相等的函数，如果 $a_i=b_j$，则 $\operatorname{cost}\left(a_i,b_j\right)=0$ 表示不需要替换，如果 $a_i\neq b_j$，则 $\operatorname{cost}\left(a_i,b_j\right)=1$ 表示需要替换。最后，矩阵的右下角 $\boldsymbol{D}[m][n]$ 即源字符串 $a$ 和目标字符串 $b$ 的 Levenshtein 距离，表示将源字符串转换为目标字符串所需的最少编辑操作数。

将 Levenshtein 距离标准化，使距离值与字节码字符串的长度无关，将距离值规范化到[0, 1]的范围，从而能够更准确地反映字节码间的相似度。对于字节码字符串 $a$ 和 $b$，计算式为

$$\mathrm{NLD}(a,b)=\frac{\boldsymbol{D}(a,b)}{\max(|a|,|b|)} \tag{2}$$

其中，$\boldsymbol{D}(a,b)$是$a$和$b$之间的Levenshtein距离，$|a|$和$|b|$分别是字节码$a$和$b$的长度。这样就得到了字节码相似度分数，以此作为字节码相似度特征。

**2.2.2** 图特征

本节将对智能合约的CFG进行深入分析，以提取2类图特征：图的基本特征和图的结构特征。一方面，从CFG的节点信息入手，提取了包括但不限于节点的度量、聚类系数、图密度等基本图论特征，提供了合约结构的初步理解。另一方面，使用图嵌入算法和神经网络模型提取图结构特征，将图的节点和边映射到低维空间，捕捉节点间的复杂关系和图的全局属性。最后通过特征融合技术将这些信息整合到一个统一的特征向量中，具体操作方法如下。

1) 图基本特征

基于图论的基础理论，对合约CFG的基本特征进行分析，2类合约的节点分布有着显著差异，非庞氏骗局合约的节点数量分布广泛，节点数量庞大、图结构复杂的合约也比较多；庞氏骗局合约普遍节点较少，集中在100个节点以内，边的分布同样如此。

为了平衡不同合约间节点数量级的差异，引入了图的平均度数作为指标，具体计算式为

$$\text{average\_degree}=\frac{1}{n}\sum_{i=1}^{n}\deg(v_i) \tag{3}$$

其中，$n$代表CFG有$n$个节点，$\deg(v_i)$是第$i$个节点的度数。平均度数可以用来描述图中所有节点的连接程度的平均水平，反映图的密集程度。根据统计，非庞氏骗局合约的密集程度远高于庞氏骗局合约，且呈现较为分散的分布特征，而庞氏骗局合约的平均度数集中在1附近。

最后，提取了节点数量“number_of_nodes”、边数量“number_of_edges”、平均度数“average_degree”作为图基本特征。

2)图结构特征

图结构特征提取流程如图4所示。首先，使用图嵌入Node2vec算法提取控制流图每个节点的特征向量，得到一个特征矩阵，然后，将特征矩阵与代表图结构的邻接矩阵输入GCN进行图卷积得到全局特征，最后，使用平均池化操作得到图级别的特征向量作为图结构特征。

Node2Vec是一种用于图节点嵌入（node embedding）的算法，旨在通过学习图中节点的低维向量表示，将图的结构信息有效地捕捉到低维空间，使用基于随机游走的策略来生成节点的邻居序列，并通过优化目标函数来学习节点嵌入。Node2Vec采用了Skip-Gram模型来训练节点的嵌入向量。Skip-Gram模型的目标是给定一个节点的“中心”节点，预测它的“上下文”节点。在Node2Vec中，中心节点是当前节点，而上下文节点是通过随机游走生成的邻居节点。Node2vec节点跳转过程如图5所示，假设第$i$-1步走到s，要探索第$i$步该游走到哪一个节点，对s下一步跳转到各个点的跳转概率计算为

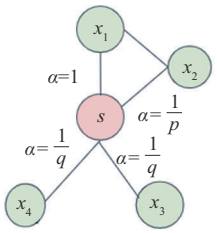


图5 Node2vec节点跳转过程

$$\boldsymbol{P}(c_i=x|c_{i-1}=s)=\begin{cases}\dfrac{\boldsymbol{w}_{sx}\alpha_{pq}(s,x)}{Z}, (s,x)\in E\\ 0,\text{其他}\end{cases} \tag{4}$$

其中，$E$为边集，$\boldsymbol{w}_{sx}$为相邻边上的权重，$Z$为权重之和，$\alpha_{pq}(s,x)$为修正系数，控制游走趋势，其计算式为

$$\alpha_{pq}(s,x)=\begin{cases}\dfrac{1}{p}, d_{sx}=0\\ 1, d_{sx}=1\\ \dfrac{1}{q}, d_{sx}=2\end{cases} \tag{5}$$

其中，$d_{sx}$表示节点s到x的最短距离，只能取{0,1,2}中的数值；如果2个节点是同一节点，则跳转返回上一节点的概率为$\frac{1}{p}$；如果2个节点直接相连，则$d_{sx}=1$，则跳转到与s节点相邻的节点概率为1；如果2个节点不相邻，则$d_{sx}$=2，跳转到不相连的节点概率为$\frac{1}{q}$。基于此策略，$p$取值较小时，容易跳转回上一个节点，图的遍历越倾向于广度优先搜

索；$q$取值较小时，容易跳转到更远的节点，图的遍历越倾向于深度优先搜索。由此得到了每个节点的随机游走序列。

然而，对于每个图而言，得到的图特征仍是一个$M \times N$的特征矩阵，$M$为节点数量，$N$为每个节点的特征维度，为了得到一个图级别的$1 \times N$维的特征向量，将每个图的特征矩阵进行图卷积操作与池化处理。在图卷积操作中，每个节点的特征会根据其邻居节点的特征加权聚合，并进行非线性变换。在池化操作中，使用平均池化，目的是将节点的特征合并成图级别的特征向量，它对所有节点的特征进行均值计算。平均池化操作如图6所示，输入特征矩阵，对于每个池化窗口，计算其包含的所有数值的平均值。如高亮的窗口包含数值2、3、5、6，它们的平均值是$\frac{2+3+5+6}{4}=4$，再将这些平均值组成新的矩阵，即输出矩阵，多次重复后，得到了一个维度为8的图结构特征向量。

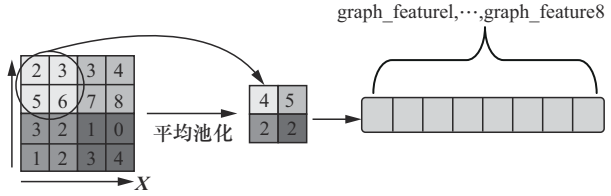


图6　平均池化操作

## 2.3　模型

### 2.3.1　数据不平衡问题

在真实世界中，庞氏骗局合约只占海量的智能合约很小一部分，数据集中大部分都是正常合约，庞氏骗局识别问题可以被认定为一种不平衡二分类问题，因此使用上采样的方法以平衡多数类和少数类样本，来解决数据不平衡问题。首先，从合并后的数据集中分离出标记为0（非庞氏骗局）的多数类样本和标记为1（庞氏骗局）的少数类样本。然后，使用sklearn.utils.resample函数对少数类样本进行上采样，以使其数量与多数类样本相同，设置replace=True参数表示允许重复采样，即同一个样本可以被多次选中。最后，将上采样后的少数类样本与原始的多数类样本合并，形成一个新的平衡数据集。测试集保持原始分布，确保测试阶段不会受到训练集操作的干扰。

### 2.3.2　模型

2.2节提取了语义特征和图特征，并将其标准化处理后线性嵌入，得到一个新的融合多特征的特征向量。定义了一个适合智能合约庞氏骗局识别的GCN模型。Kipf等[23]将CNN应用于图结构数据，从而开发了GCN，其核心在于图卷积操作，目的是利用节点的邻居信息来更新节点的表示，从而更好地反映节点的上下文。给定节点$i$的特征表示，GCN的节点更新规则为

$$\boldsymbol{h}_i^{(l+1)}=\sigma\left(\sum_{j\in N(i)}\frac{1}{c_{ij}}\boldsymbol{W}^{(l)}\boldsymbol{x}_j^{(l)}\right) \tag{6}$$

其中，$N_{(i)}$表示节点$i$的邻居集合，$c_{ij}$是归一化因子，$\sigma$是激活函数，$\boldsymbol{W}^{(l)}$是权重矩阵。GCN通常由多层卷积操作构成，每一层的输出节点表示都将包含该节点及其邻居的特征信息。每层图卷积都将进行节点特征的更新，同时将邻接信息传播到下一层，在每一层，GCN将节点的特征矩阵与权重矩阵进行线性变换，并通过激活函数（如ReLU）进行非线性映射。本文使用了3层卷积神经网络模型，其中，每一层GCNConv都是为了捕获图的拓扑特征。输入$\boldsymbol{X}$为提取的特征矩阵，隐藏层的第1层conv1为GCNConv层，用于将节点特征从原始维度转换到一个更高的维度，并使用RELU激活函数进行非线性变换；然后通过Dropout层，防止过拟合，提高模型的泛化能力。第2层conv2进行了同样的操作来到conv3，应用softmax函数，将特征维度降低

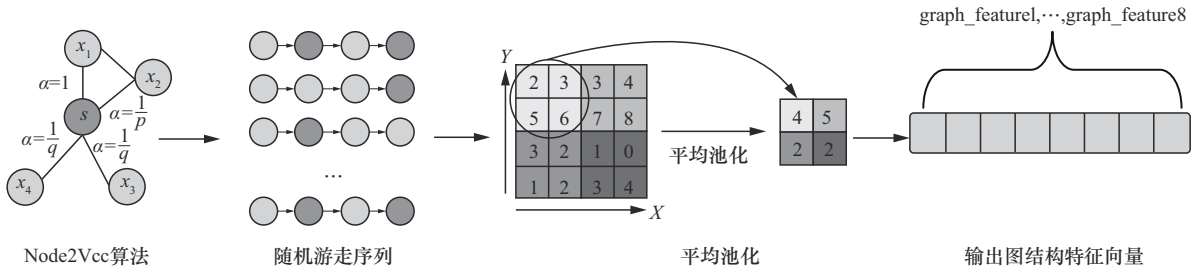


图4　图结构特征提取流程

到二维，表示二分类问题，进行最终的类别预测，输出预测值$\hat{y} \in \{0,1\}$。

### 2.4 复杂度分析

本节分3个阶段对PonziGCN方法的复杂度进行分析，分别为数据预处理、特征提取、模型搭建与预测分类。

1) 数据预处理阶段

在这一阶段，要从字节码构建控制流图，复杂度为$O(N \times l^2)$，其中，$N$为合约数量，$l$为字节码长度。

2) 特征提取阶段

在语义特征提取部分，首先，计算操作码频率特征，时间复杂度为$O(N \times k \times l)$，其中，$k$为操作码种类数。其次，计算字节码之间的Levenshtein距离，其时间复杂度为$O(m \times n)$，其中，$m$和$n$为2个字节码字符串分别的长度；因此对于$N$个合约，假设所有字节码的长度接近$l$，则总体复杂度为$O(N^2 \times l^2)$。

图特征提取部分包含2个步骤：图特征提取和图嵌入。对于每个图，计算基本特征的时间复杂度是$O(N'+E)$，其中，$N'$是节点数，$E$是边数；对于$N$个合约，图特征提取的时间复杂度为$O(N \times (N'+E))$。在图嵌入部分使用Node2Vec算法，复杂度为$O(d \times N' \times T)$，$d$是嵌入维度，$T$是每个节点的随机游走次数。

3)模型搭建与预测分类阶段

训练1个GCN模型，每一层GCN的复杂度是$O(N'+E)$，使用3层GCN结构，因此图卷积部分的整体复杂度为$O(3 \times (N'+E)) = O(N'+E)$。

根据以上分析，PonziGCN模型的总体复杂度可以表示为

$$O(N \times (m \times n + k \times l + l^2) + N \times (N'+E) + N \times d \times N' \times T + N' + E)$$

因此，随着合约数量$N$和图的规模增大，模型的计算开销会相应增加。

现有的先进方法中，基于规则的方法、机器学习方法都较为简单，基于操作码的频率或其他简单特征来判断，因此复杂度较低。分析对比同样使用图特征与神经网络模型的方案PonziGuard，PonziGuard是基于合约运行时行为图（CRBG）的Ponzi合约检测方法，利用GNN进行建模，在后文对比实验部分会详细介绍。PonziGuard控制流图的构建和特征提取复杂度为$O(N \times (N'+E))$，其中，$N$为合约数量，$N'$为节点数量，$E$为图的边数，使用GNN进行训练，训练的时间复杂度为$O(L \times (N'+E))$其中，$L$是图卷积层数，$N'$是图节点数，$E$是边数。因此，理论上简略计算PonziGuard模型的总体复杂度为$O(N \times (N'+E) + N' + E)$。

综合上述分析，PonziGCN提供了更精细的特征提取过程，包括字节码相似度计算和操作码频率等，因此其计算开销也相对较高。但是，相比采用简化的特征提取方式的相关方法，PonziGCN因为充分捕捉字节码中的重要细节信息，其精准度也会更高。

## 3 实验

本节使用真实数据集进行实验，验证本文模型的有效性，回答了以下问题。

问题1：本文将特征分为语义特征、图特征与交易特征，其中语义特征和图特征属于代码特征，使用何种特征组合可以使模型达到最优效果？

问题2：与现有的先进方法和模型相比，本文模型准确性如何？

问题3：本文模型的各个特征重要性如何？

### 3.1 实验环境

实验采用了文献[18]中的数据集，包含3 774个合约的字节码和操作码数据，其中132个是庞氏骗局合约。该数据集包含合约字节码、64-D操作码以及是否涉及庞氏骗局的标签。

实验环境是Windows 11操作系统、Pycharm平台和Python编程语言。

### 3.2 评价指标

本文采用的评价指标有精确率（Precision）、召回率（Recall）、F值（F-score）和AUC值。精确率是所有被判定为庞氏骗局的合约中真实庞氏骗局的比例；召回率是模型检测到的庞氏骗局在所有庞氏骗局中的比例；F值是综合评价精确率和召回率的混合量度；AUC值是接收者操作特征（ROC）曲线下的面积，它衡量模型在不同阈值下区分正负类别的能力，ROC曲线通过计算不同阈值下的真正率（TPR, true positive rate）和假阳性率（FPR, false positive rate）得到，AUC值越接近1，分类效果越好。这些指标的计算式为

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{7}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{8}$$

$$\text{F\_score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Recision} + \text{Recall}} \tag{9}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{10}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{11}$$

其中，TP 和 FP 分别代表被判断为庞氏骗局合约的庞氏骗局合约数量和普通合约数量，TN 和 FN 分别代表被判断为普通合约的数量和庞氏骗局合约的数量。

同时，为了验证本文方法的高效性，还选择了程序运行时间 run_time（以 s 为单位）作为参考指标，对比不同模型的运行效率。

### 3.3　实验结果分析

#### 3.3.1　特征组合效果对比实验

针对问题 1，本文将特征集划分为 3 个主要类别：语义特征、图特征和交易特征。本文研究的核心目标是实现庞氏骗局智能合约的早期检测，因此，特别关注语义特征和图特征，而将交易特征作为辅助对比研究特征。为了深入分析这些特征的有效性，设计了一系列对比实验，以评估不同特征组合的性能。

在语义特征方面，本文提取了字节码相似度特征和 64 维操作码词频特征。对于图特征，提取了控制流图的基本特征和 8 维图结构特征。对于交易特征，则提取了 7 维与合约相关的交易特征。

本文定义了 7 种不同的特征组合类型，并采用统一参数设置和数据平衡处理的 GCN 模型。数据集被随机划分为 80% 的训练集和 20% 的测试集，以进行对比实验。实验结果通过精确率、召回率、F 值和 AUC 值 4 个评价指标进行评估，其值越高，表明模型的检测效果越佳。为了全面评估提出的 PonziGCN 模型在智能合约庞氏骗局检测中的表现，避免实验结果的偶然性或过拟合，对每个特征组合模型进行五轮次实验，每轮实验都采用了相同的超参数设置，最后性能指标取平均结果。不同特征组合模型性能如表 1 所示，当使用单一特征时，语义特征展现出最佳性能，其召回率和 F 值分别达到了 0.985 和 0.950。紧随其后的是图特征，其 F 值达到了 0.885。在组合特征的使用中，语义特征与图特征的组合表现最为出色，精确率、召回率和 F 值分别达到了 0.982、0.987 和 0.978，AUC 值也高达 0.983；这表明，该特征组合在准确度和模型性能方面均达到了最优，且未涉及交易特征，与本文探索智能合约庞氏骗局早期识别模型的重点相契合。

**表 1**　不同特征组合模型性能

| 特征 | 精确率 | 召回率 | F 值 | AUC |
|---|---|---|---|---|
| 语义特征 | 0.917 | 0.985 | 0.950 | 0.974 |
| 图特征 | 0.805 | 0.941 | 0.885 | 0.946 |
| 交易特征 | 0.706 | 0.922 | 0.800 | 0.795 |
| 语义特征和交易特征 | 0.918 | 0.984 | 0.950 | 0.965 |
| 图特征和交易特征 | 0.799 | 0.892 | 0.843 | 0.928 |
| 语义特征、图特征和交易特征 | 0.9111 | 0.944 | 0.927 | 0.906 |
| 语义特征和图特征 | 0.982 | 0.987 | **0.978** | **0.983** |

进一步地，本文在语义特征和图特征的基础上分别或联合添加交易特征，发现这些操作对模型性能产生了一定程度的负面影响。这一结果从侧面表明，交易特征对于提升模型性能并无显著帮助，反而可能由于特征输入的复杂性和高维度而干扰了模型的原有效果。

#### 3.3.2　模型对比实验

针对问题 2，对比实验将从以下 2 个方面进行：一方面，将本文方法与目前先进算法进行对比实验；另一方面，对本文 GCN 模型调整参数和结构进行消融实验。

目前主流的检测算法包括机器学习算法和深度学习算法，分别选取了机器学习领域的 Random-Forest 算法、XGBoost 算法和 SMOTETomek_LGBM 算法[18]以及深度学习领域的 SPP-CNN 算法[24]、PonziGuard 算法[5]与本文 PonziGCN 算法进行对比实验，实验细节如下。

1) 基于规则的检测方法：该方法通过分析操作码频率，制定规则，针对与庞氏骗局合约高度相关的操作码（如 CREATE、NUMBER 等）频率阈值，实现资源受限环境下庞氏骗局合约的轻量级检测。

2) RandomForest[25]：该算法使用随机森林模型，是一个包含多个决策树的分类器，其输出类别由单个树的模式决定，算法使用 3 种特征: 交易特征、操作码频率和字节码相似性。

3) XGBoost[26]：XGBoost 是一种增强算法，将许多树模型集成到一个强分类器中。该算法采用 CART 回归树模型，使用的特性是交易特征、操作码频率和字节码相似性。

4) SMOTETomek_LGBM[18]：该算法使用 SMOTE_Tomek 方法改进后的 LightGBM 模型来平衡二分类问题，并考虑 3 个特征：交易特征、操作码频率和字节码相似性。

5) SPP-CNN[24]：该算法使用了引入空间金字塔池化方法作为改进的 CNN 模型，使用的特征为字节码生成的图像特征。

6) PonziGuard[5]：该算法提取合约运行时行为图特征，赋能 GNN 进行建模分析。

不同算法性能对比如表 2 所示。由表 2 可知，基于规则的检测方法各项指标表现较差，但运行时间非常短；改进的机器学习算法 SMOTE-Tomek_LGBM 效果比传统的机器学习算法更好，精确率可达 0.966；深度学习算法比机器学习算法效果有提升，SPP-CNN 算法和 PonziGuard 算法 F 值分别达到了 0.969 和 0.974。本文算法相较于目前已有的先进算法，有了一定程度的提升，精确率、召回率和 F 值分别达到了 0.982、0.987 和 0.978。从运行效率来看，对每个算法都设置了迭代 100 次，最后发现机器学习算法的效率最高，XGBoost 算法仅用了 0.841 s 就完成了任务；SPP-CNN 算法由于涉及图像识别，运行时间最长，用了 307.026 s 完成任务。本文模型使用了 55.287 s 完成任务，在保证精确率的情况下效率较高，但仍存在提升空间。

为了避免过拟合和数据泄露，提供更可靠的性能评估，对 PonziGCN 方法使用了五折交叉验证并计算标准误差。PonziGCN 模型在五折交叉验证下的性能表现如表 3 所示。由表 3 可知，模型在各项指标上均表现优异，其中，精确率为 0.973 ± 0.011，说明模型在识别庞氏骗局时大多判断正确，误报率较低；召回率达到 0.987 ± 0.012，说明模型漏报极少；F 值为 0.977 ± 0.014，在精确率和召回率之间取得了良好的平衡。最后，AUC 值为 0.982 ± 0.015，进一步反映出模型具有良好的整体判别能力。

**表 2　不同算法性能对比**

| 算法 | 精确率 | 召回率 | F 值 | 运行时间/s |
|---|---|---|---|---|
| 基于规则的检测方法 | 0.183 | 0.288 | 0.241 | 0.100 |
| 随机森林 | 0.906 | 0.966 | 0.935 | 16.436 |
| XGBoost | 0.933 | 0.949 | 0.965 | 0.841 |
| SMOTETomek_LGBM | 0.966 | 0.966 | 0.966 | 3.812 |
| SPP-CNN | 0.964 | 0.938 | 0.969 | 307.026 |
| PonziGuard | 0.955 | 0.968 | 0.974 | 47.738 |
| **PonziGCN** | **0.982** | **0.987** | **0.978** | **55.287** |

**表 3　PonziGCN 模型在五折交叉验证下的性能表现**

| 指标 | 均值±标准误差 |
|---|---|
| 精确率 | 0.973 ± 0.011 |
| 召回率 | 0.987 ± 0.012 |
| F 值 | 0.977 ± 0.014 |
| AUC | 0.982 ± 0.015 |

同时，为了进一步验证未出现过拟合的情况，对比了模型在训练集与测试集上的性能表现，如表 4 所示。由表 4 可知，PonziGCN 模型在训练集与测试机上的性能表现差异较小，这也进一步表明模型并未出现过拟合的情况。

**表 4　PonziGCN 模型在训练集与测试集上的性能表现**

| 数据集 | 精确率 | 召回率 | F 值 | AUC |
|---|---|---|---|---|
| 训练集 | 0.980 | 0.987 | 0.977 | 0.983 |
| 测试集 | 0.965 | 0.979 | 0.969 | 0.972 |

本文算法使用 GCN 模型，为了探究更为有效适配的模型算法，对 GCN 模型进行了参数与结构的改进对比实验，同时对比了不同的神经网络模型的效果，以证明 PonziGCN 模型的有效性。实验模型细节如下。

GCN-NoDropout：基于 GCN 的无 Dropout 网络模型（GCN_nzoDropout），Dropout 是一种常见的正则化技术，它的主要作用是防止模型的过拟合，设置该模型可以探究 PonziGCN 模型的普适性与泛化能力。

多层感知机（MLP）[27]：将 GCN 模型替换为

MLP 模型，它是一种前馈神经网络，对数据的局部结构和空间关系没有内建的理解机制，比图卷积层的计算复杂度低。设置该模型可以探究使用图神经网络模型是否有必要，以及在不同复杂程度下模型的效率情况。

GAT-2GCN：结合 GCN 和图注意力网络（GAT）[28]的新型模型，使用 GAT 作为第 1 层进行节点特征聚合，采用自注意力机制来计算不同邻居节点对目标节点的影响力，使用 GCN 作为第 2、3 层来进一步处理节点的特征，进行卷积操作，能够有效地捕捉结构信息。

使用相同的数据集，并对不平衡数据集进行处理，使用代码语义特征和控制流图特征输入上述模型，得到的结果如图 7~图 11 所示。PonziGCN 在 GCN 的基础上进行了增强，以提高模型的泛化能力和性能，其在各个性能指标上都表现优异，且在整个训练过程中表现出极高的稳定性，在迭代 100 次左右达到稳定，表明 PonziGCN 模型在处理图数据方面具有优越的性能和泛化能力。但在运行时间方面，PonziGCN 在模型设计上为了提高检测准确性而采用了更复杂的结构，运行时间相对较长，后续可以进一步探究性能提升的途径。GCN-NoDropout 模型去除了 Dropout 层，在 AUC、精确度、召回率和 F 值上表现出较大的波动，尤其是在训练周期的中后期，这可能表明模型在没有 Dropout 正则化的情况下容易过拟合。MLP 模型在所有性能指标上都表现出稳定的趋势，整体性能略低于 GCN 模型，但胜在结构简单，运行时间最短，效率最高。GAT-2GCN 模型在 AUC、精确度、召回率和 F 值上都表现出较好的稳定性，但准确度较低，运行时间最长。证明了结合注意力机制的 GCN 能够有效地捕捉图数据的结构信息，但由于模型复杂，综合能力较差，后续还需进一步研究提升模型性能的途径。

### 3.3.3　特征重要性分析

针对问题 3，本文基于扰动法，通过评估特征值的小幅变化（0.01）对模型预测输出的影响，来衡量每个特征的重要性。实验使用了语义特征和图特征共 78 维特征，使用性能最好的 GCN 模型，随机取 80% 的数据作为训练集，20% 的数据作为测试集进行实验，实验结果显示了特征重要性前 10 的特征，如图 12 所示。

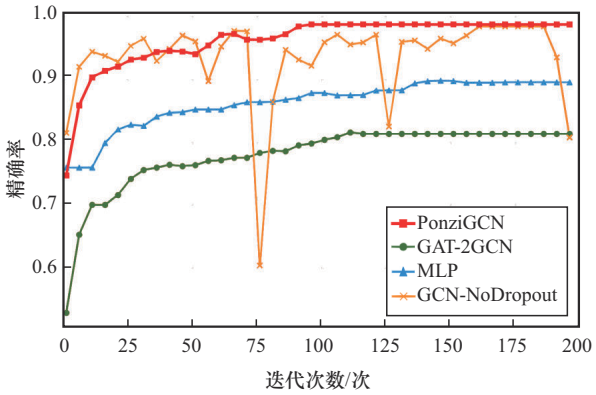

图 7　不同模型精确率随迭代次数变化

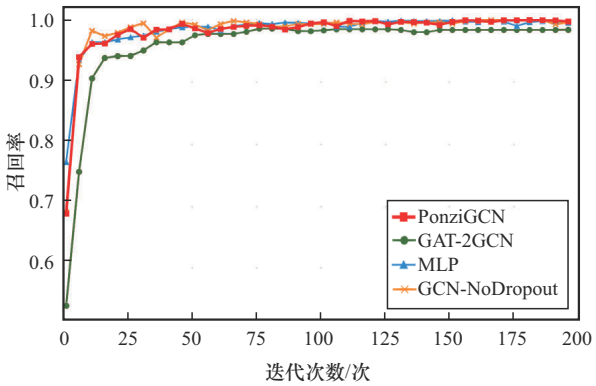

图 8　不同模型召回率随迭代次数变化

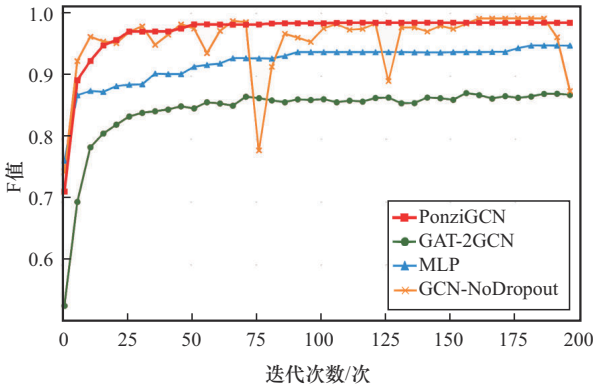

图 9　不同模型 F 值随迭代次数变化

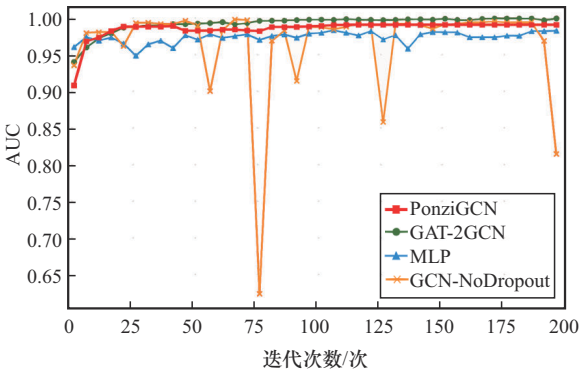

图 10　不同模型 AUC 值随迭代次数变化

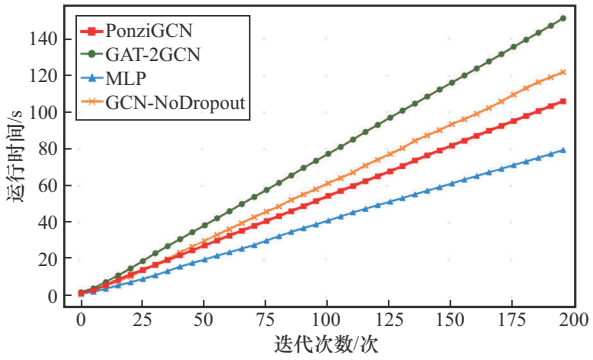


图 11　不同模型运行时间随迭代次数变化

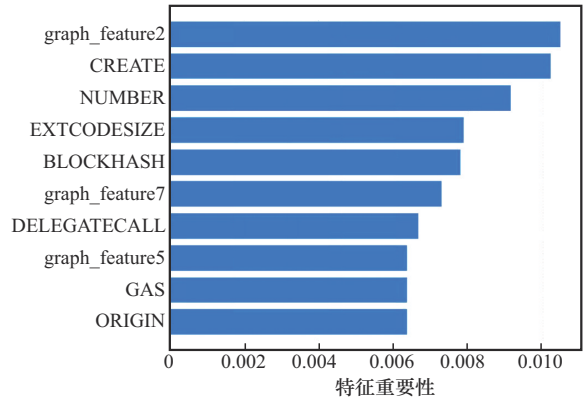


图 12　重要性前 10 的特征

图结构特征包括 graph_feature2、graph_feature5 和 graph_feature7，它们代表了智能合约代码的控制流图结构特征。操作码频率特征则包括 CREATE、NUMBER、EXTCODESIZE、DELEGATECALL、BLOCKHASH、GAS 和 ORIGIN，这些特征反映了智能合约中特定操作码的出现频率。

庞氏骗局合约通常具有以下特点。1)资金流动不透明，通过频繁的合约调用和复杂的资金链条隐藏真实的资金来源。2)频繁创建新合约，通过创建新合约来不断吸引新的投资者。3)高度依赖控制流，合约的控制流图结构可能非常复杂，包含大量的条件判断和资金流转路径。结合这些特点进行分析，本文的实验结果揭示了一个显著的现象：在所有考察的特征中，图结构特征（如 graph_feature2 和 graph_feature7）具有较高的重要性，表明图结构特征能有效捕捉到合约内部函数调用和控制流的异常模式，反映合约内函数的调用顺序和依赖关系。庞氏骗局合约通常包含多个循环调用和复杂的资金流动结构，因此，这些图结构特征能够帮助模型识别复杂的欺诈行为，对于提升模型的整体性能起到了关键作用。此外，操作码频率反映了合约代码中常见的操作模式。CREATE 操作码频繁出现通常意味着合约会创建新的合约地址，而在庞氏骗局合约中，创建新合约的操作往往被用来隐藏资金流动和非法资金链条，因此这是庞氏骗局合约中的一个典型特征。NUMBER 操作码与合约中的函数调用次数相关，而 EXTCODESIZE 操作码则表示合约代码的大小，这 2 个特征能反映合约的复杂性与是否涉及多次合约调用，频繁的合约调用和合约大小的异常增长可能指示出合约存在潜在的恶意行为。被标记为高重要性的操作码，均与触发回扣交易功能紧密相关，通过这些操作码频率特征，模型能够识别出潜在的庞氏骗局合约。因此，代码中与交易功能密切相关的操作码频率特征，对于模型预测庞氏骗局合约也具有极高的指示价值。

### 3.4　实验小结

本实验对本文 PonziGCN 模型特征选择与模型构建进行了全面的评估，并与其他先进的检测算法进行了对比。结果表明，PonziGCN 模型在精确率、召回率、F 值和 AUC 值等关键性能指标上均表现优异，其中精确率达到 0.982，召回率为 0.987，F 值为 0.978，AUC 值为 0.983。结合语义特征和图特征的组合在所有特征组合中表现最佳，这强调了在庞氏骗局检测中同时考虑合约代码的语义信息和结构信息的重要性。此外，特征重要性分析揭示了图结构特征和与交易功能密切相关的操作码频率特征在模型中具有最高的重要性。

模型也存在不足之处，尽管 PonziGCN 模型在性能上优于其他模型，但其运行时间也相对较长。表明在实际应用中可能需要在模型性能和计算效率之间进行权衡，未来的工作可以探索优化模型结构，以提高其运行效率。总体而言，PonziGCN 模型能为区块链技术的健康发展提供有力的技术支持，并为智能合约庞氏骗局的早期检测提供一种有效的解决方案。

## 4　结束语

本文提出了一种基于 GCN 的智能合约庞氏骗局检测方法 PonziGCN，该方法融合了智能合约代码的语义特征和控制流图特征，以实现对潜在庞氏骗局的早期检测和预警。本研究使用智能合约

字节码作为原始数据，通过特征工程分为语义特征和图特征两大部分。语义特征包括字节码相似度和操作码频率特征；而图特征则涵盖控制流图的基本特征和结构特征，采用Node2vec算法和图卷积神经网络技术，以捕捉智能合约的代码特征和全局结构特征。实验结果表明，PonziGCN模型在各项性能指标上均表现优异，证明了同时考虑合约代码的语义信息和结构信息对于庞氏骗局检测的重要性。

未来可以从以下几个方面进行改进本文的研究工作：1)针对大规模图数据，可以研究更加高效的图卷积计算方法，例如图采样技术或分布式图计算方法；2)可以对智能合约字节码进行更为深入的溯源研究，恢复其伪代码表现形式，研究其特征表现；3)进一步优化模型的可解释性，使GCN能够提供更具可解释性的决策依据，这对于庞氏骗局检测甚至智能合约漏洞检测都具有重要的意义。

此外，以下相关问题也值得未来深入思考和研究。1) 由于PonziGCN方法结合了字节码反编译、控制流图的构建以及基于GCN的模型训练与推理，这些过程需要大量的计算资源。在计算资源受限的区块链环境中，计算复杂性和高时延仍然是一个不可忽视的挑战。因此，如何优化模型计算效率、提高推理速度以适用于边缘设备值得进一步探索。2) 尽管PonziGCN方法在现有数据集上取得了良好的效果，但考虑到攻击者可能通过代码混淆或加密手段规避检测，因此，引入抗混淆的表示学习机制、构建混淆样本数据集并采用对抗训练方法，提升模型在面对加密或混淆合约时的鲁棒性和泛化能力也值得探究。3) 面对新型或变种庞氏骗局，攻击者可能通过规避已知模式进行隐匿。因此，如何结合专家经验来预测这些变种的潜在特征，并通过无监督学习、迁移学习和增量学习等方法，使模型能够快速适应新型骗局的变化也值得深入研究。

## 参考文献：


[1] SZABO N. Smart contracts: building blocks for digital markets[J]. EXTROPY: The Journal of Transhumanist Thought, 1996, 18(2): 28.

[2] 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.

[3] ZHANG Y M, KANG S Q, DAI W, et al. Code will speak: early detection of ponzi smart contracts on ethereum[C]//Proceedings of the 2021 IEEE International Conference on Services Computing (SCC). Piscataway: IEEE Press, 2021: 301-308.

[4] WU J J, LIU J L, CHEN J Z, et al. ContraPonzi: smart ponzi scheme detection for ethereum via contrastive learning[C]//Proceedings of the 2023 4th Asia Service Sciences and Software Engineering Conference. New York: ACM Press, 2023: 155-162.

[5] LIANG R C, CHEN J, HE K, et al. Ponziguard: detecting ponzi schemes on ethereum with contract runtime behavior graph (CRBG)[C]//Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering (ICSE). Piscataway: IEEE Press, 2024: 766-777.

[6] BARTOLETTI M, PES B, SERUSI S. Data mining for detecting Bitcoin ponzi schemes[C]//Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). Piscataway: IEEE Press, 2018: 75-84.

[7] JIN C X, JIN J, ZHOU J J, et al. Heterogeneous feature augmentation for ponzi detection in ethereum[J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2022, 69(9): 3919-3923.

[8] ZHENG Z B, CHEN W L, ZHONG Z J, et al. Securing the ethereum from smart ponzi schemes: identification using static features[J]. ACM Transactions on Software Engineering and Methodology, 2023, 32(5): 1-28.

[9] CHEN W L, ZHENG Z B, NGAI E C H, et al. Exploiting blockchain data to detect smart ponzi schemes on ethereum[J]. IEEE Access, 2019, 7: 37575-37586.

[10] LINNHOFF-POPIEN C, SCHNEIDER R, ZADDACH M. Digital marketplaces unleashed[M]. Berlin: Springer, 2018.

[11] QIAN P, LIU Z G, HE Q M, et al. Towards automated reentrancy detection for smart contracts based on sequential models[J]. IEEE Access, 2020, 8: 19685-19695.

[12] LIU Z G, QIAN P, WANG X Y, et al. Combining graph neural networks with expert knowledge for smart contract vulnerability detection[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(2): 1296-1310.

[13] ZHUANG Y, LIU Z G, QIAN P, et al. Smart contract vulnerability detection using graph neural network[C]//Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. Piscataway: IEEE Press, 2020: 3283-3290.

[14] CHEN C, SU J Z, CHEN J C, et al. When ChatGPT meets smart contract vulnerability detection: how far are we?[J]. ACM Transactions on Software Engineering and Methodology, 2025, 34(4): 1-30.

[15] VASEK M, MOORE T. Analyzing the Bitcoin ponzi scheme ecosystem[C]// Financial Cryptography and Data Security. Berlin: Springer, 2019: 101-112.

[16] TORRES C F, STEICHEN M, STATE R. The art of the scam: demystifying honeypots in Ethereum smart contracts[C]//Proceedings of the 28th USENIX Conference on Security Symposium. Berkeley: USENIX Association, 2019: 1591-1607.

[17] CHEN W L, GUO X F, CHEN Z G, et al. Honeypot contract risk warning on ethereum smart contracts[C]//Proceedings of the 2020 IEEE International Conference on Joint Cloud Computing. Piscataway: IEEE Press, 2020: 1-8.

[18] ZHANG Y M, YU W Q, LI Z Y, et al. Detecting ethereum ponzi schemes based on improved LightGBM algorithm[J]. IEEE Transactions on Computational Social Systems, 2022, 9(2): 624-637.

[19] ZHOU Y, KUMAR D, BAKSHI S, et al. Erays: reverse engineering ethereum's opaque smart contracts[C]//Proceedings of the 27th USENIX Conference on Security Symposium. Berkeley: USENIX Association, 2018: 1371-1385.

[20] CHEN T, LI Z H, LUO X P, et al. Poster: SigRec - automatic recovery of function signatures in smart contracts[C]//Proceedings of the 2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS). Piscataway: IEEE Press, 2023: 1065-1066.

[21] ZHAO K S, LI Z H, LI J F, et al. DeepInfer: deep type inference from smart contract bytecode[C]//Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. New York: ACM Press, 2023: 745-757.

[22] ALLEN F E. Control flow analysis[J]. ACM SIGPLAN Notices, 1970, 5(7): 1-19.

[23] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[J]. arXiv Preprint, arXiv: 1609.02907, 2016.

[24] LOU Y C, ZHANG Y M, CHEN S P. Ponzi contracts detection based on improved convolutional neural network[C]//Proceedings of the 2020 IEEE International Conference on Services Computing (SCC). Piscataway: IEEE Press, 2020: 353-360.

[25] CUTLER A, CUTLER D R, STEVENS J R. Random forests[J]. Machine Learning, 2004, 45(1): 157-176

[26] CHEN T Q, GUESTRIN C. XGBoost: a scalable tree boosting system[C]// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2016: 785-794.

[27] PINKUS A. Approximation theory of the MLP model in neural networks[J]. Acta Numerica, 1999, 8: 143-195.

[28] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[J]. arXiv Preprint, arXiv: 1710.10903, 2017.

[作者简介]

**张艳梅**（1976– ），女，吉林省吉林市人，博士，中央财经大学教授、博士生导师，主要研究方向为区块链、金融科技、服务计算、商务智能等。

**郭思颖**（2001– ），女，四川南充人，中央财经大学硕士生，主要研究方向为区块链、金融科技。

**贾恒越**（1984– ），女，内蒙古海拉尔人，博士，中央财经大学副教授、硕士生导师，主要研究方向为区块链、量子信息处理。

**姜茸**（1978– ），男，云南凤庆人，博士，云南财经大学教授、博士生导师，主要研究方向为云计算、大数据、区块链、信息管理、软件工程和数字经济等。