基于源码结构和图注意力网络的以太坊蜜罐合约检测方法

王友卫[1]，侯玉栋[1]，凤丽洲[2]
（1. 中央财经大学信息学院，北京 102206；2. 天津财经大学统计学院，天津 300222）

摘 要：针对目前蜜罐合约检测方法准确率不高以及泛化性较差等问题，提出了基于源码结构和图注意力网络的以太坊蜜罐合约检测（CSGDetector）方法。首先，为了提取出智能合约 Solidity 源码的结构信息，对源码进行语法分析，将其转换为 XML 解析树；然后，筛选出可以表达合约结构特征和内容特征的特征词集，并构造出合约源码结构图；最后，为避免数据集不平衡性带来的影响，在集成学习理论基础上引入教师模型和学生模型的概念，分别从全局和局部的角度训练图注意力网络模型，并融合所有模型的输出作为合约最终检测结果。实验表明，与已有方法 KOLSTM 相比，CSGDetector 在二分类与多分类实验中的 F_1 值分别提升了 1.27%与 7.21%，验证了其具有较高的蜜罐检测能力；与已有方法 XGB 相比，CSGDetector 在掩蔽蜜罐检测实验中针对不同类型蜜罐合约的平均召回率提升了 7.57%，验证了所提方法在提升算法泛化性能方面的有效性。

关键词：以太坊；蜜罐合约；源码结构；图注意力网络；集成学习
中图分类号：TP309.2
文献标志码：A
DOI: 10.11959/j.issn.1000−436x. 2023178

Honeypot contract detection method for Ethereum based on source code structure and graph attention network

WANG Youwei[1], HOU Yudong[1], FENG Lizhou[2]
1. School of Information, Central University of Finance and Economics, Beijing 102206, China
2. School of Statistics, Tianjin University of Finance and Economics, Tianjin 300222, China

Abstract: To address the problems of low accuracy and poor generalization of current honeypot contract detection methods, a honeypot contract detection method for Ethereum based on source code structure and graph attention network was proposed. Firstly, in order to extract the structural information of the Solidity source code of the smart contract, the source code was parsed and converted into an XML parsing tree. Then, a set of feature words that could express the structural and content characteristics of the contract was selected, and the contract source code structure graph was constructed. Finally, in order to avoid the impact of dataset imbalance, the concepts of teacher model and student model were introduced based on the ensemble learning theory. Moreover, the graph attention network model was trained from the global and local perspectives, respectively, and the outputs of all models were fused to obtain the final contract detection result. The experiments demonstrate that CSGDetector has higher honeypot detection capability than the existing method KOLSTM, with increments of 1.27% and 7.21% on F_1 measurement in two-class classification and multi-class classification experiments, respectively. When comparing with the existing method XGB, the average recall rate of CSGDetector in the masked honeypot detection experiments for different types of honeypot contracts is improved by 7.57%, which verifies the effectiveness of the method in improving the generalization performance of the algorithm.

Keywords: Ethereum, honeypot contract, source code structure, graph attention network, ensemble learning

收稿日期：2023−05−29；**修回日期：**2023−09−04
基金项目：教育部人文社科基金资助项目（No.19YJCZH178）；国家自然科学基金资助项目（No.61906220）；国家社科基金资助项目（No.18CTJ008）；中央财经大学新兴交叉学科建设项目
Foundation Items: The Ministry of Education of Humanities and Social Science Project (No.19YJCZH178), The National Natural Science Foundation of China (No.61906220), The National Social Science Foundation of China (No.18CTJ008), The Emerging Interdisciplinary Project of CUFE

0 引言

以太坊是一个开源的分布式计算平台，它使用区块链技术来实现一个去中心化的、公开透明的、安全可靠的计算环境，并提供了一个完整的开发环境，可以让用户轻松地开发和部署智能合约[1-2]。智能合约作为一种特殊的计算机程序，能够自动完成指定的任务，但其执行过程是不可逆的，一旦被执行，就不能撤销或更改[2-4]。这种特性使智能合约在以太坊区块链上非常受欢迎，可以用来执行许多不同类型的交易和合同[5]。然而，也正是因为这种去中心化和不可逆的特点，智能合约中存在着许多安全隐患与漏洞。不同于利用合约漏洞主动攻击，蜜罐合约是一种新型的诈骗方式。它通常暴露出明显的漏洞，使受害者以为自己可以用较低的资金投入换取高额的回报，从而诱使受害者向其合约账户中转账。然而，当受害者按照要求转入一定数量的资金后，蜜罐合约中另一个不易察觉的陷阱便会阻止资金转出，蜜罐合约的部署者便从中获利。由于蜜罐合约具有种类繁多、隐蔽性强、生命周期长等特点，这种新型诈骗方式对以太坊生态系统的安全和稳定带来巨大的威胁。

目前，越来越多的学者投入蜜罐合约的识别与检测研究中[6-13]。按照识别原理的不同，这些识别方法可分为机器学习方法和启发式分析方法。机器学习方法可以让模型从大量数据中自主学习到分类的关键特征，不局限于事先规定好的匹配规则或模式，通常具有良好的泛化能力。然而，在真实的智能合约数据集中，蜜罐合约往往只占了很少的部分，而数据的高度不平衡会使模型很难从中学习到分类的关键特征，导致分类效果不佳；启发式分析方法借助预先定义好的匹配规则，可以在已知的蜜罐合约中达到较好的分类效果，但此类方法泛化性较差，难以发现规则以外的蜜罐合约。对比发现，上述方法所利用的特征信息有所不同，有些方法利用合约余额、交易信息、字节码等特征，而有些方法则直接利用合约源码特征。一般而言，攻击者要想成功利用蜜罐合约诱骗受害者，通常会主动公开源码，从而使受害者能够快速发现合约中的明显漏洞。换言之，没有公开源码的智能合约一般不会是蜜罐合约，即使它实现了与蜜罐合约相同的逻辑，受害者也无法仅通过字节码快速知悉合约中故意构造的漏洞，继而不会主动向该合约投入资金。

为此，本文提出了一种基于源码结构和图注意力网络的以太坊蜜罐合约检测（CSGDetector, code structure and graph attention network based detector）方法。通过构建 XML 解析树提取 Solidity 源码中的特征词和结构关系，并构造出合约源码结构图。针对训练数据不平衡的问题，引入教师模型和学生模型的概念，利用图注意力网络独立训练多个模型，并融合所有模型的输出获得合约检测结果。实验表明，CSGDetector 不仅实现了较高的蜜罐检测精度，而且具有良好的泛化性能。具体来说，本文的主要贡献如下。

1) 充分考虑 Solidity 源码的结构关系，将特征词和结构关系作为蜜罐合约检测的依据，基于 XML 解析树构建了智能合约对应的源码结构图。

2) 提出了基于图注意力网络的蜜罐合约检测方法，有效利用源码的结构特征和内容特征，通过图注意力机制提高了蜜罐合约检测效果。

3) 为降低合约数据集中不同类别样本集规模不平衡带来的影响，引入教师模型和学生模型的概念，利用 Bagging 集成学习思想独立训练不同模型，并通过融合不同模型的预测结果提高针对蜜罐合约检测的准确率。

1 相关工作

2019 年，Torres 等[6]首次对蜜罐合约进行了系统分析，将其分为 8 种类型，并在此基础上开发了蜜罐合约检测方法 HoneyBadger，通过启发式静态分析方法对蜜罐合约进行识别及分类。由于 HoneyBadger 依据字节码对合约进行检测，而部分合约在编译过程中丢失了某些与蜜罐合约相关的重要信息，因此该方法的检测效果并不理想。Camino 等[7]基于 XGBoost 算法提出了一种利用合约源码特征、交易特征和资金流特征的蜜罐合约检测方法（XGB），并在原有类型的基础上发现了 2 种新型蜜罐合约。然而，该方法需要从合约的交易信息和资金流信息中提取特征作为检测依据，这意味着其往往只有在受害者被骗之后才能检测出蜜罐合约，因此存在时效性差、漏报率高等问题。Chen 等[8]提出了一种新的蜜罐合约检测方法（LGBM），该方法基于 LightGBM 模型实现，通过合约字节码构造出操作码，并通过 n-gram 特征来检测蜜罐合约。张红霞等[9]在 LGBM 的基础上提出了“关键操作码”的概念，在传统的长短期记忆（LSTM, long

short-term memory）模型中加入关键操作码权重机制，提出蜜罐合约检测方法 KOLSTM（key-opcode long short-term memory）。上述 2 种方法均从字节码中解析出操作码，并通过机器学习方法对操作码中的特征进行提取，其对应的检测精度在二分类和多分类任务中较先前方法获得了一定程度的提升。但是，此类方法仅考虑了合约操作码中的序列关系，忽略了合约内部的逻辑关系和结构特征。冀甜甜等[10]提出了“蜜罐家谱”的概念以及一种不依赖于机器学习方法的蜜罐合约检测方法 CADetector。该方法提取了蜜罐合约的细粒度特征，但预先定义特征的方式导致其对未知类型的蜜罐合约检测能力较差。韩松明团队[11-12]提出了 DC-Hunter，这是一种基于字节码的危险合约检测方法，可以对部分类型的蜜罐合约进行检测。Hu 等[13]同样以字节码为特征，通过带有注意力机制的门控循环单元（GRU, gate recurrent unit）来判断是否存在包括蜜罐合约在内的诈骗合约。这 2 种方法以检测多种合约骗局或漏洞合约为主要目标，虽然实现了部分类型的蜜罐合约检测，但未能覆盖全部类型的蜜罐合约。

综上所述，现有的蜜罐合约检测方法仍存在以下不足。

1) 合约字节码在编译过程中将丢失继承关系[6]等重要信息，导致基于字节码的检测方法无法准确检测每种类型的蜜罐合约；利用资金流、交易内容等信息作为特征进行检测的方法存在明显的滞后性，通常仅当受害者被骗后才会出现明显特征，导致蜜罐合约检测的时效性不佳。

2) 合约源码的逻辑性和结构性较强，LGBM[8]和 KOLSTM[9]等方法忽略了源码的层次和结构信息，无法充分表达合约中蕴含的丰富特征。

3) 实际场景中蜜罐合约的数量远小于普通合约的数量，而现有方法中处理不平衡数据的方式较简单，模型训练通常在样本子集上进行，容易忽略原始样本空间中对蜜罐合约检测有用的重要信息。

与现有方法不同，CSGDetector 利用图注意力网络（GAT, graph attention network）融合源码中的词元语义信息和层次结构信息，通过引入教师模型和学生模型分别捕捉数据集中的全局特征信息和局部特征信息，充分挖掘不同数据集中蕴含的深层次特征，以此提高蜜罐合约检测效果。

2　相关理论

2.1　XML 解析树

Solidity 是常用来编写以太坊智能合约的编程语言，其语法类似于 JavaScript 和 C++，有很强的逻辑性与内部结构关系。为了提取 Solidity 源码的层次结构关系，Tikhomirov 等[14]于 2018 年提出了 SmartCheck 工具，其实现了通过 ANTLR 工具和 Solidity 语法将 Solidity 源码转换为 XML 解析树的过程。具体来说，在生成 XML 解析树的过程中，首先需要将 Solidity 源码拆解为不可再分的代码词元（如标识符、关键字等），然后根据语法规则将这些代码词元转换为抽象语法树，最后将抽象语法树中的每个节点转换为 XML 元素，并构造出 XML 解析树，其中，叶子节点即原 Solidity 源码中的代码词元，非叶子节点用来标识该词元在原语句中所处的位置。

2.2　GAT

图神经网络（GNN, graph neural network）是一种专门用于处理图数据的神经网络模型[15]，它的核心思想是在图上执行节点信息的传递和更新。GAT 在 GNN 基础上引入注意力机制，在信息传递的过程中为当前节点的不同邻居节点分配不同的注意力权重，以提高节点信息聚合效果[16]。基于 GAT 的节点信息聚合过程如图 1 所示。其中，不同线型的箭头表示基于不同注意力头的信息传递过程。

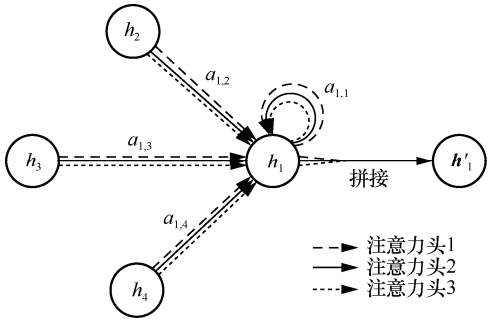

图 1　基于 GAT 的节点信息聚合过程

给定节点集 V 以及边集 E，首先计算节点 v_i 的邻居节点 v_j 在第 k 个注意力头上对于 v_i 的注意力分数 $e_{i,j}^k$

$$e_{i,j}^k = \text{LeakyReLU}\left(\boldsymbol{\theta}^{\mathrm{T}}\left[\boldsymbol{W}^k h_i \,\|\, \boldsymbol{W}^k h_j\right]\right) \quad (1)$$

其中，LeakyReLU 为激活函数，$\boldsymbol{W}^k$ 为第 k 个注意力头对应的权重矩阵，||为向量拼接操作，$\boldsymbol{\theta}$ 为参数

矩阵。然后，对 $e_{i,j}^k$ 进行归一化处理得到最终的注意力权重 $a_{i,j}^k$

$$a_{i,j}^k = \mathrm{softmax}_j\left(e_{i,j}^k\right) \tag{2}$$

在此基础上，按照式(3)计算节点 v_i 的第 k 个注意力头对应的输出向量 $\boldsymbol{h}_i^k$。最后，拼接所有注意力头对应的输出向量得到 v_i 对应的最终特征向量 $\boldsymbol{h}_i'$，如式(4)所示。

$$\boldsymbol{h}_i^k = \sigma\left(\sum_{j\in N(i)} a_{i,j}^k \boldsymbol{W}^k h_j\right) \tag{3}$$

$$\boldsymbol{h}_i' = \sum_{k=1}^{K} \boldsymbol{h}_i^k \tag{4}$$

其中，$N(i)$为节点 v_i 的所有邻居节点，K 为注意力头数，σ 为非线性激活函数。

3 本文方法

3.1 相关定义

本文以智能合约的 Solidity 源码为研究对象，相关定义如下。

定义 1 词元（token）。Solidity 源码中不可再分的词法单元，即组成 Solidity 源码的最小单元，用来表达合约的内容特征，对应 XML 解析树中的叶子节点。

定义 2 结构标签（tag）。标识词元位置与源码结构信息的标签，对应 XML 解析树中的非叶子节点。

定义 3 特征词集。从全部合约的词元和结构标签中筛选出的可以有效表达合约特征的一组特征词，表示为集合 T，即

$$T = \left\{t_i \mid 0 < i \leqslant N\right\} \tag{5}$$

其中，N 为特征词集中特征词的数量，第 i 个特征词 t_i 用其对应的 one-hot 向量表达，即

$$\boldsymbol{x}_i = [0,0,\cdots,1,\cdots,0] \tag{6}$$

其中，第 i 个位置为 1，其余位置均为 0。

定义 4 合约结构关系矩阵。智能合约 addr 的结构关系矩阵 $\boldsymbol{R}_{\text{addr}}$ 描述该合约中任意 2 个特征词之间的关系，表示为

$$\boldsymbol{R}_{\text{addr}} = \left\{r_{ij} \mid 0 < i, j \leqslant N_{\text{addr}}\right\} \tag{7}$$

在智能合约 addr 的 XML 解析树中，若特征词 t_i 为特征词 t_j 的子节点，则 $r_{ij}=1$，反之 $r_{ij}=0$。

3.2 方法描述

如图 2 所示，CSGDetector 执行过程可概括为以下 4 个步骤：XML 解析树生成、特征词筛选、源码结构图构建、模型训练及蜜罐合约检测，具体内容如下。

3.2.1 XML 解析树生成

为了提取合约结构信息，首先将合约的 Solidity 源码都转换成 XML 解析树的形式。以合约 0x38934f199b309a33e39adc0f1fbf66aa2a2f44f0 为例，该合约实现了一个将输入数字扩大 100 倍的功能，其对应的 Solidity 源码如图 3 所示，利用 SmartCheck 工具将其转换后得到的 XML 解析树如图 4 所示。其中，叶子节点表示 Solidity 源码中实际出现的词元，如“0.4.25”、“fus”等；非叶子节点表示词元在 Solidity 源码中出现位置的结构标签，如“<sourceUnit>”“<expression>”等。由图 4 可知，树中词元及结构标签呈现出明显的层次特征，例如，节点 sourceUnit 表示整个合约，由 pragmaDirective 和 contractDefination

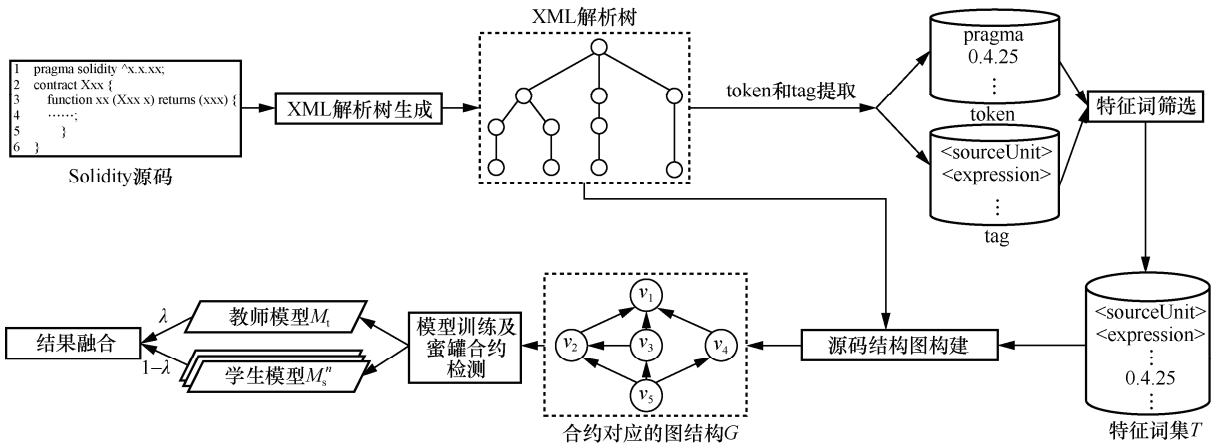

图 2 CSGDetector 方法执行流程

及 EOF 这 3 个节点构成，其中，pragmaDirective 为标识编译器版本的标签节点，contractDefination 为定义具体合约内容的标签节点，EOF 为结束符标签节点。

```
1    pragma solidity ^0.4.25;
2    contract MSD2 {
3        function fus (uint256 i) public pure returns (uint256) {
4        return i*100;
5         }
6    }
```

图 3　合约对应的 Solidity 源码

3.2.2　特征词筛选

为了提高蜜罐合约识别效果、降低模型训练复杂度，需要从全部合约的词元集和结构标签集中筛选高质量特征词以获得最终的特征词集 T。由于词元包含合约的内容信息，而结构标签包含词元在源码中的相对位置信息，可以有效反映合约的结构特征，因此本文将所有的词元和结构标签作为候选特征词集。然而，并非所有词元都可以有效表达合约的特征，出现频次较高的词元（如 address、pragma 等）过于普遍，难以区分不同的合约；出现频次过低的词元（如自定义变量名）存在较大的偶然性和随机性，在对识别结果造成干扰的同时还会增加模型的训练复杂度。为此，本文结合文档频率（DF, document frequency）和逆文档频率（IDF, inverse document frequency），从全部词元中过滤出现频次过高和过低的词元，并将过滤后的词元与结构标签集合并，以此构建最终的特征词集 T，即

$$T = \text{tags} \cup \{\text{token}_i \mid \min < \text{Freq}(\text{token}_i) < \max\} \quad (8)$$

其中，tags 为结构标签集，$\text{Freq}(\text{token}_i)$为第 i 个词元在全部合约中出现的频次，min 和 max 为预设阈值。由于蜜罐合约数据集的正负样本数量差异较大，本文引入词频缩放因子 p 和 q（$p < q$）来调整蜜罐合约和普通合约中不同特征词的重要性，以降低样本集规模不平衡性对特征筛选结果的影响。若词元 token_i 在 d_1 个普通合约和 d_2 个蜜罐合约中出现，则其对应的频次为

$$\text{Freq}(\text{token}_i) = pd_1 + qd_2 \quad (9)$$

3.2.3　源码结构图构建

为了避免智能合约中非重要词元的影响，本节在 XML 解析树构建结果的基础上利用筛选所得特征词集构建源码结构图。给定特征词集 T、智能合约 addr 及其对应的 XML 解析树 $\text{Tree}_{\text{addr}}$，首先，获取 $\text{Tree}_{\text{addr}}$ 的全部节点，并删除未在 T 中出现的词元或结构标签，通过去重操作得到合约 addr 的特征词集 T_{addr}；然后，计算 T_{addr} 中任意 2 个特征词之间的关系，在此基础上按照定义 4 获得其结构关系矩阵 $\boldsymbol{R}_{\text{addr}}$；最后，根据 T_{addr} 和 $\boldsymbol{R}_{\text{addr}}$ 构建 addr 对应的源码结构图 $G_{\text{addr}} = (V_{\text{addr}}, E_{\text{addr}})$，其中，$V_{\text{addr}}$ 为合约 addr 对应的节点集，节点集中的每个节点 v_i 对应 T_{addr} 中特征词 t_i 的 one-hot 向量表达；$E_{\text{addr}}=\{e_{ij}\}$为合约 addr 对应的边集，对于 V_{addr} 中任意节点对 v_i 与 v_j，若存在 $r_{ij}=1$，则 E_{addr} 中存在一条由 v_i 指向 v_j 的边 e_{ij}。

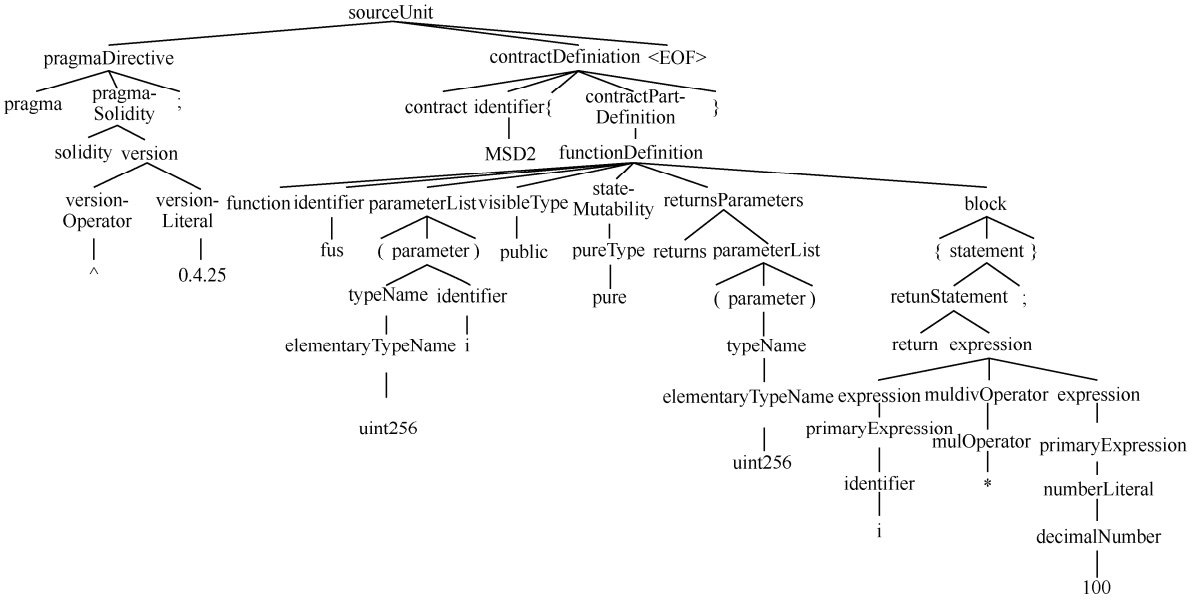

图 4　合约的 XML 解析树

以合约 0x38934f199b309a33e39adc0f1fbf66aa2a2f44f0 为例，按照上述步骤构造出的源码结构 G 如图 5 所示。由图 5 可知，G 中共包含 30 个节点，对应合约中 100、pure、<pureType>、<decimalNumber>等 30 个特征词。其中，100、pure 等为合约中的词元，<pureType>、<decimalNumber>等为合约的结构标签。由该合约对应的 XML 解析树（图 4）知，“100”为“<decimalNumber>”的子节点，因此 G 中存在由“100”节点指向“<decimalNumber>”节点的边。进一步发现，G 中还存在一条由“<expression>”节点指向其自身的边，这是因为在 XML 解析树中存在“<expression>”为“<expression>”子节点的情况，因此在经过去重处理后的节点集中出现了由该节点指向其自身的边。

3.2.4 模型训练及蜜罐合约检测

如图 6 所示，CSGDetector 由 3 个 GAT 层和一个全连接层构成。在每个 GAT 层中，图 G_{addr} 中的每个节点 v_i 都将按照注意力权重聚合每个邻居节点的特征信息，以此更新该节点的向量表达。

$$a_{i,j}^{kl}=\text{softmax}_j\left(\text{LeakyReLU}\left(\boldsymbol{\theta}^{Tl}\left[\boldsymbol{W}^{kl}\,\boldsymbol{h}_i^l \,\|\, \boldsymbol{W}^{kl}\,\boldsymbol{h}_j^l\right]\right)\right) \tag{10}$$

$$\boldsymbol{h}_i^{l+1}=\sum_{k=1}^{K}\sigma\left(\sum_{j\in N(i)}a_{i,j}^{kl}\boldsymbol{W}^{kl}\boldsymbol{h}_j^l\right) \tag{11}$$

其中，$\boldsymbol{h}_i^l$ 为节点 v_i 在第 l 层的向量，$\boldsymbol{W}^{kl}$ 为第 l 层中第 k 个注意力头对应的权重矩阵，$\boldsymbol{\theta}^{Tl}$ 为第 l 层的参数矩阵，$a_{i,j}^{kl}$ 为在第 l 层中第 k 个注意力头对应的节点 v_i 对其邻居节点 v_j 的归一化注意力得分。

经过 3 个 GAT 层，每个节点 v_i 获得其对应的特征向量 $\boldsymbol{h}_i$。在此基础上，使用全局累加池化函数 global_add_pool 聚合所有节点的特征向量以获得图 G_{addr} 对应的图级向量 $\boldsymbol{h}_{\text{addr}}$。

$$\boldsymbol{h}_{\text{addr}}=\text{global\_add\_pool}\left(\boldsymbol{h}_i|v_i\in V_{\text{addr}}\right) \tag{12}$$

最后，将合约的图级向量 $\boldsymbol{h}_{\text{addr}}$ 输入全连接层中，并结合合约 addr 的实际类别以及交叉熵函数来使损失最小化，以此实现模型训练过程。

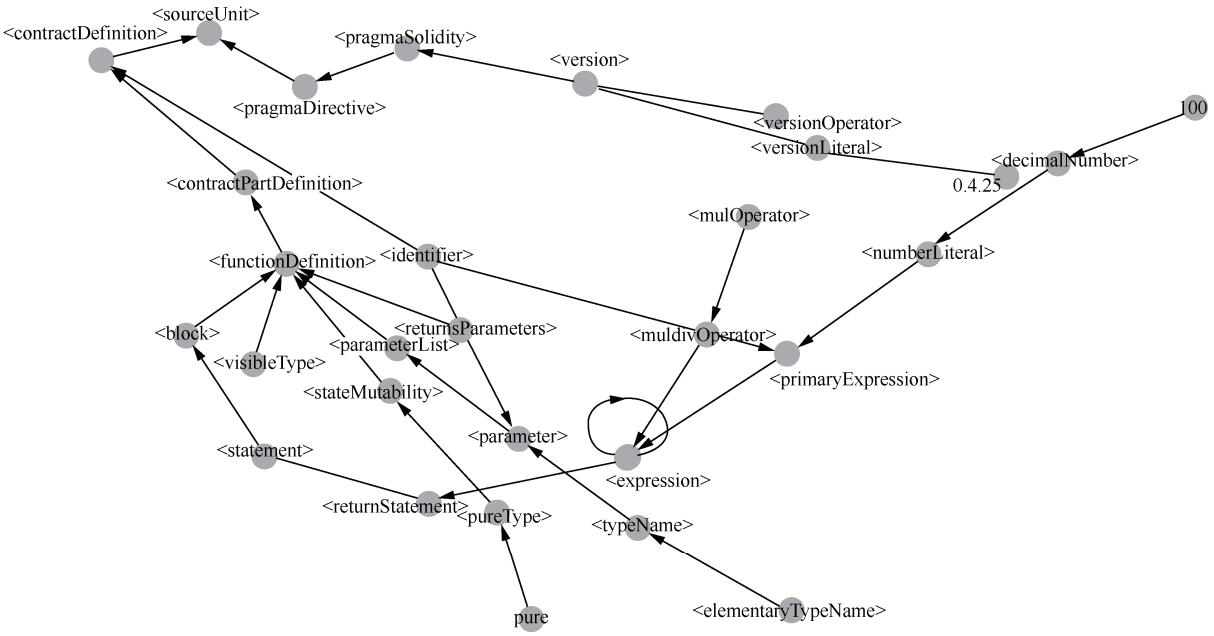

图 5 合约构成的源码结构 G

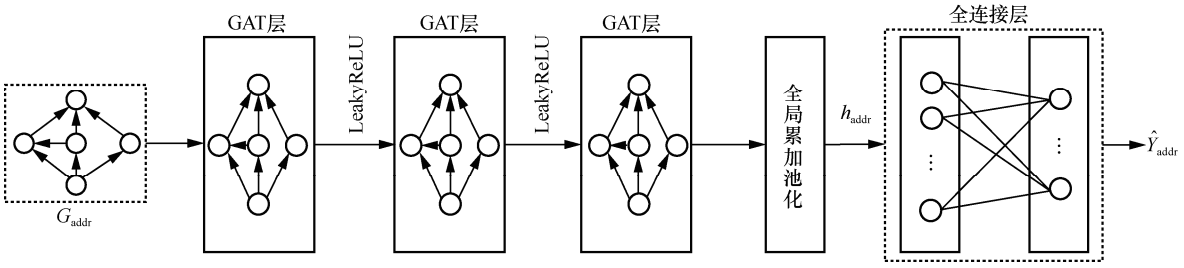

图 6 CSGDetector 网络模型示意

$$\hat{\boldsymbol{Y}}_{\text{addr}} = \text{softmax}\left(\boldsymbol{\alpha}\boldsymbol{h}_{\text{addr}} + \boldsymbol{\beta}\right) \tag{13}$$

$$\text{Loss} = -\sum_{\text{addr}\in C}\sum_{\text{tp}\in \text{CLS}} y_{\text{addr}}^{\text{tp}} \text{lb}\left(\hat{y}_{\text{addr}}^{\text{tp}}\right) \tag{14}$$

其中，$\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$ 为参数矩阵，C 为智能合约数据集，CLS 为智能合约的类别集合（二分类中分为普通合约和蜜罐合约，多分类中分为普通合约和各个不同类型的蜜罐合约），$\hat{\boldsymbol{Y}}_{\text{addr}}$ 为 addr 属于不同类别的归一化概率分布向量，Loss 为损失值，$\hat{y}_{\text{addr}}^{\text{tp}}$ 为 addr 的预测类别为 tp 的概率，$y_{\text{addr}}^{\text{tp}}$ 为合约 addr 的实际类别，定义为

$$y_{\text{addr}}^{\text{tp}} = \begin{cases} 1, & \text{合约addr属于tp类别} \\ 0, & \text{合约addr不属于tp类别} \end{cases} \tag{15}$$

由于数据集中普通合约与蜜罐合约的数量高度不平衡，直接使用原始训练集将使训练过程偏向于学习数量较多的普通合约样本集中的特征。此外，直接在传统合成少数类过采样技术（SMOTE, synthetic minority over-sampling technique）[17]、简单数据增强（EDA, easy data augmentation）[18]等采样方法得到的均衡化数据集上进行模型训练将使模型关注数据子集中所含的局部特征，而忽略原始数据集中所含的全局特征信息。为此，本文引入教师模型和学生模型的概念，通过教师模型捕捉原始数据集中蕴含的全局特征信息，通过学生模型捕捉均衡化处理后数据集中蕴含的局部特征信息。

具体来说，给定训练集 $S=\{S_h, S_m\}$，其中，S_h 为蜜罐合约样本集，S_m 为普通合约样本集，满足 $|S_m|>>|S_h|$（$|S_m|$、$|S_h|$分别为 S_m、S_h 中的样本数）。首先，直接在 S 上训练得到教师模型 M_{t}，随后借助 Bagging 集成学习的思想，将 S_m 随机划分为 n（n>1）份均匀的普通合约样本子集。

$$S_m = \left\{S_m^1, \cdots, S_m^i, \cdots, S_m^n\right\} \tag{16}$$

然后，如图 7 中灰色区域所示，按照式(17)构建学生模型对应的训练样本子集 $S^i(0<i\leqslant n)$，并在 S^i 上训练得到学生模型 M_{s}^i。

$$S^i = S_m^i \bigcup S_h \tag{17}$$

最后，针对待检测合约 x，通过式(18)融合教师模型与所有学生模型的预测结果，得到 x 对应的最终类别。

$$\text{Output} = \lambda\hat{\boldsymbol{Y}}_x + \sum_{i=1}^{n}\frac{1-\lambda}{n}\hat{\boldsymbol{Y}}_x^i \tag{18}$$

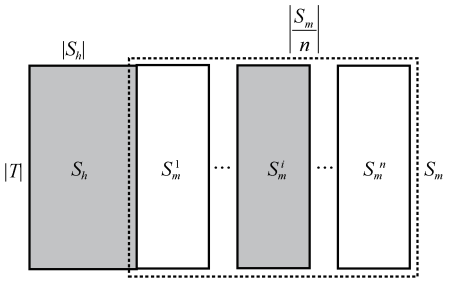

图 7　学生模型训练样本子集构建示意

其中，$\hat{\boldsymbol{Y}}_x$ 为使用教师模型对 x 的预测类别概率分布向量；$\hat{\boldsymbol{Y}}_x^i$ 为使用第 i 个学生模型对 x 的预测类别概率分布向量；λ 为教师模型的输出在最终预测结果中的占比；Output 为融合后的预测输出，取其中最大分量值对应类别作为最终的检测结果 pred(x)，即

$$\text{pred}(x)=\arg\max_{k_i\in K}\left(\text{Output}_i\right) \tag{19}$$

其中，Output_i 为 Output 在第 i 维上的分量值。

3.3　时间复杂度分析

本文方法的时间复杂度主要从以下 4 个模块进行分析：XML 解析树生成、特征词筛选、源码结构图构建、模型训练及蜜罐合约预测。其中，XML 解析树生成及特征词筛选模块需要遍历全部合约中的词元，对应的时间复杂度为 $O(N_t|C|)$（N_t为每个合约中的词元数量，$|C|$为训练集中的合约数量）。源码结构图构建模块需要遍历 XML 解析树中的全部节点，对应的时间复杂度为 $O(N\text{addr}|C|)$（Naddr 为每个合约的特征词数量）。模型训练及蜜罐合约检测模块，对于第 l 个 GAT 层，需针对每个合约将 N_{addr} 个节点的特征维度由输入空间维度$|\boldsymbol{h}^{l-1}|$映射到输出空间维度$|\boldsymbol{h}^{l}|$并计算注意力得分，对应的时间复杂度为 $O(N_{\text{addr}}|\boldsymbol{h}^{-1}||\boldsymbol{h}^{l}|)+O(|E||\boldsymbol{h}^{l}|)$；全连接层对应的时间复杂度为 $O(|\boldsymbol{h}^{3}||\text{CLS}|)$（$|\text{CLS}|$为类别个数）。因此，给定注意力头数 K、训练轮数 epoch，由于 GAT 层数为常数，本文方法训练阶段的整体时间复杂度为 $O((N_{\text{addr}}|\boldsymbol{h}^{l-1}||\boldsymbol{h}^{l}|K+|E||\boldsymbol{h}^{l}|K+|\boldsymbol{h}^{3}||\text{CLS}|)|C|\text{epoch})$。同理可知，本文方法针对单个合约检测的时间复杂度为 $O(N_{\text{addr}}|\boldsymbol{h}^{l-1}||\boldsymbol{h}^{l}|K+|E||\boldsymbol{h}^{l}|K+|\boldsymbol{h}^{3}||\text{CLS}|)$。

4　实验及讨论

4.1　数据获取及预处理

本文实验采用的数据集为冀甜甜等[10]开源的合约数据集，其中，蜜罐合约部分包含 798 个，包

括 Torres 团队利用 HoneyBadger[6]发现的蜜罐合约、利用 XGB[7]发现的蜜罐合约和冀甜甜等利用 CADetector 发现的蜜罐合约。这些蜜罐合约可以划分为 10 种类型，包括隐藏状态更新（HSU, hidden state update）、继承障碍（ID, inheritance disorder）、未初始化结构（US, uninitialized struct）、稻草人合约（SMC, straw man contract）、余额混乱（BD, balance disorder）、隐藏转移（HT, hidden transfer）、跳空字符串（SESL, skip empty string literal）、类型溢出（TDO, type deduction overflow）、未执行调用（UC, unexecuted call）和映射密钥编码骗局（MKET, map key encoding trick）。其中，样本数量最多的蜜罐合约类型为 HSU，共有 512 个样本；样本数量最少的蜜罐合约类型为 MKET，只有一个样本。数据集中的非蜜罐合约部分包括 330 万个区块上的 1 157 992 个普通合约地址。通过以太坊浏览器 Etherscan 对这些合约的 Solidity 源码进行获取，本文得到了 124 988 份普通合约源码及 790 份蜜罐合约源码（其余 8 个均已被销毁）。通过对合约源码进行去重，最终得到普通合约源码 43 152 份、蜜罐合约源码 763 份。

4.2 评价指标

本文选择精确率（Precision）、召回率（Recall）和 F_1 值作为模型的评估指标。不同指标的计算式如下

$$\text{Precision}=\frac{\text{TP}}{\text{TP}+\text{FP}}\times 100\% \tag{20}$$

$$\text{Recall}=\frac{\text{TP}}{\text{TP}+\text{FN}}\times 100\% \tag{21}$$

$$F_1=2\times\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}\times 100\% \tag{22}$$

其中，TP 表示正确预测为蜜罐合约的样本数量，TN 表示正确预测为普通合约的样本数量，FP 表示被错误预测为蜜罐合约的普通合约数量，FN 表示被错误预测为普通合约的蜜罐合约数量。

4.3 实验参数设置

将数据集按照 7:2:1 的比例划分为训练集、测试集和验证集。为了确定模型的最佳参数，本文进行了多次实验，得到的模型最佳参数如表 1 所示。本文后续对比算法中使用的实验参数使用对应文献的参数设置。

表 1 模型最佳参数设置

参数	数值
学习率	0.01
批尺寸	128
训练轮数	600
隐藏层维度	128
Dropout 率	0.6
普通合约采样频次 p	1
蜜罐合约采样频次 q	20
特征词频次下限 Min	500
特征词频次上限 Max	13 000
学生模型数量 n	3
教师模型融合系数 λ	0.800

4.3.1 特征词词元数量分析

为了筛选出可以有效表达智能合约特征的词元集，首先在全部合约中对出现频次不同的词元进行计数统计，按照词元出现频次从高到低排序，得到词元出现的频次直方图如图 8 所示。由图 8 可知，绝大多数词元的出现频次非常低，因此阈值 Min 的选择对于词元集构建影响较大。进一步发现，当阈值 Max 增大到一定程度（如 Max>10 000）时，Max 的取值变化对所选词元数量影响较小。为了获得过滤掉频次过高和过低的词元的最佳阈值，这里令 Max 分别为 10 000、13 000、16 000，令 Min 分别为 300、500 和 700，在此基础上统计本文方法在二分类任务中的 F_1 值，所得结果如图 9 所示。由图 9 可知，当 Max=13 000、Min=500 时，本文方法获得最大 F_1 值 94.15%，相对于表现次之的情况对应的 F_1 值高出约 0.44%。为此，本文设置 Max=13 000、Min=500 并将其应用于后续实验中。

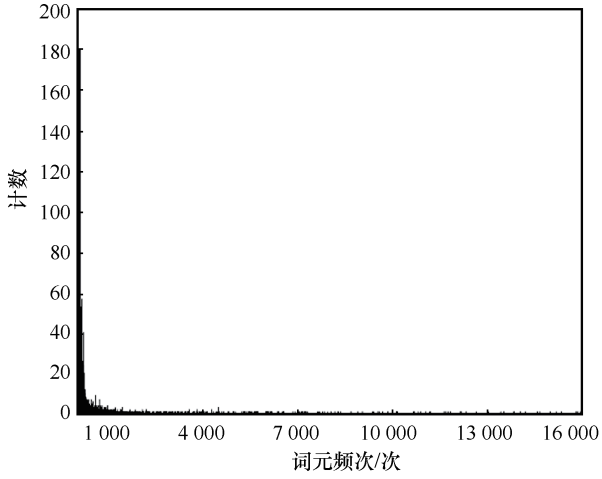

图 8 智能合约中词元的频次分布

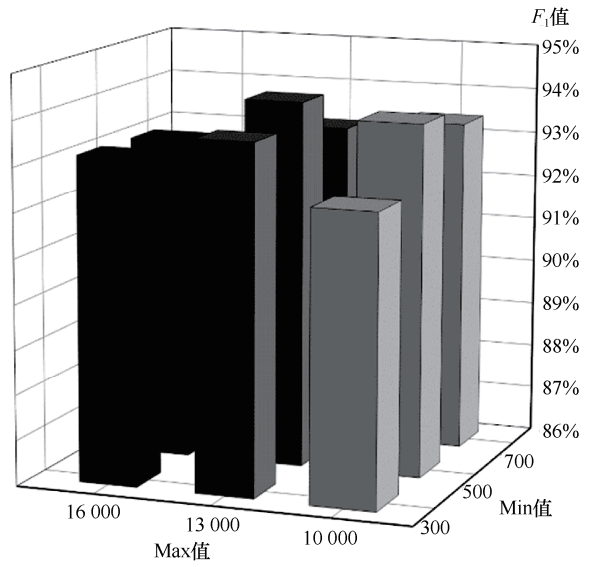

图 9　不同 Min 值和 Max 值对 F_1 值的影响

4.3.2　训练轮数分析

为了确定最佳的训练轮数，避免模型欠拟合或过拟合，本节验证了当训练轮数小于 800 时本文方法在验证集和训练集上的表现。为便于计算，这里将训练轮数以每组 50 轮为单位进行统计，并计算每组的平均 F_1 值，以反映模型的收敛情况，对应结果如图 10 所示。从图 10 中可以看出，对于训练集而言，随着训练轮数的增加，模型在训练集上的 F_1 值不断上升，说明此时模型处于不断学习的状态；对于验证集而言，当训练轮数小于 600 时，模型在验证集上的 F_1 值整体上升，但当训练轮数大于 600 时，模型在验证集上的 F_1 值开始呈现下降趋势，说明此时的模型已经达到过拟合状态。因此，为了获得最优的模型表现，本文设置训练轮数为 600。

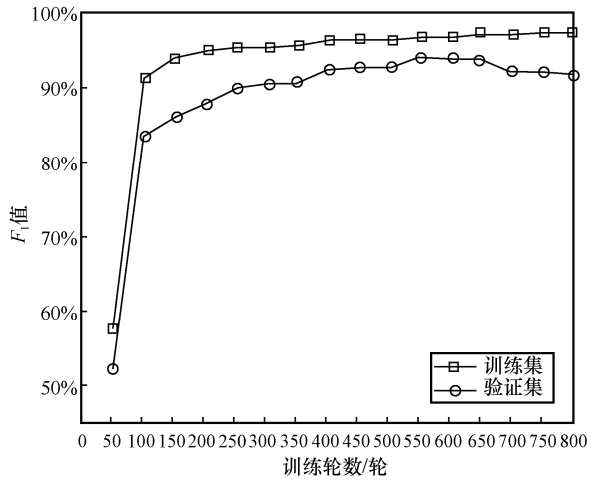

图 10　训练集和验证集在不同训练轮数下的 F_1 值

4.3.3　学生模型数量分析

为了确定学生模型的数量 n，这里令 n 分别取 3、5、7、9，并与同一教师模型融合，通过在二分类场景中的表现来评估不同学生模型数量对蜜罐合约检测能力的影响，实验结果如表 2 所示。从表 2 可知，学生模型数量为 3 时对应的精确率最高，达到 95.65%；学生模型数量为 9 时对应的召回率最高，达到 100%。综合来看，当学生模型数量为 3 时，本文方法表现最好，其对应的精确率、F_1 值均为最高值。这是因为随着学生模型数量 n 不断增加，训练样本子集中蜜罐合约样本数量减少，导致学生模型的分类性能下降且不同模型之间的差异性降低，继而影响了不同学生模型的融合效果。

表 2　不同学生模型数量对蜜罐合约检测能力的影响

学生模型数量	精确率	召回率	F_1 值
3	95.65%	98.09%	96.86%
5	91.76%	99.36%	95.41%
7	92.31%	99.36%	95.71%
9	90.23%	100%	94.86%

4.4　与现有典型方法的比较

4.4.1　二分类实验

为了验证 CSGDetector 方法针对普通合约与蜜罐合约的区分能力，本文将 CSGDetector 与典型的蜜罐检测方法 RNN[9]、LSTM[9]、CNN[9]、LGBM[8]、KOLSTM[9]和 CADetector[10]进行对比，实验结果如表 3 所示。由表 3 可知，RNN 方法获得了最高的召回率 98.62%和最低的精确率 79.65%；LSTM 和 CNN 方法分别采用门结构和卷积操作来捕捉数据的序列特征，虽然综合表现相较于 RNN 模型有所提升，但这 2 种方法并未考虑到合约内部的结构特征，因此对于蜜罐合约的识别仍有一定缺陷；LGBM 和 KOLSTM 方法在精确率方面表现良好，但在召回率方面表现略差，这可能是由于它们基于操作码（Opcode）或关键操作码（Key-Opcode）对蜜罐合约进行检测，仅考虑了数据的序列特征，出现了部分过拟合现象；与 CSGDetector 方法类似，CADetector 方法也利用源码进行蜜罐检测，但其通过预先定义特征的方式检测蜜罐，容易丢失某些深层的隐藏特征。总体来看，CSGDetector 表现最好，对应的召回率达到次最高值 98.09%，对应的 F_1 值达到最高值 96.86%，相较于获得次最高 F_1 值的

KOLSTM 方法提高了 1.27%，相较于 CADetector 方法提高了 3.64%。这是因为除了考虑词元语义信息外，CSGDetector 还充分考虑了合约内部的层次结构关系，并通过图注意力网络实现节点之间的信息传递，使模型能够更精准地捕捉到蜜罐合约的隐式特征，提高蜜罐检测的准确性。

表 3 不同方法的二分类实验结果比较

类型	方法	精确率	召回率	F_1 值
基于其他特征检测	RNN	79.65%	**98.62%**	88.25%
	LSTM	94.60%	92.37%	93.47%
	CNN	95.75%	90.69%	93.15%
	LGBM	**97.09%**	89.62%	93.20%
	KOLSTM	96.91%	94.30%	95.59%
基于源码检测	CADetector	94.60%	91.87%	93.22%
	CSGDetector	95.65%	98.09%	**96.86%**

4.4.2 多分类实验

使用蜜罐合约中 8 个类型：隐藏状态更新（HSU）、继承障碍（ID）、未初始化结构（US）、稻草人合约（SMC）、余额混乱（BD）、隐藏转移（HT）、跳空字符串（SESL）及类型溢出（TDO）中的样本并合并全部普通合约作为训练样本集，在此基础上将本文方法与其他典型方法进行对比，结果如表 4 所示。由表 4 可知，由于跳空字符串（SESL）和类型溢出（TDO）类型的蜜罐合约样本数量较少，LGBM 方法未能捕捉到这两类合约的有效特征，导致精确率和召回率等评价指标出现了部分为 0 的情况；KOLSTM 方法整体表现良好，但在某些类型的蜜罐合约上也出现了 F_1 值偏低的问题。这可能是由于 KOLSTM 基于从字节码中提取出的关键操作码，而智能合约从源码编译到字节码的过程中会丢失有关继承的所有信息。CADetector 在多分类场景下获得了较高的精确率，但其对应的召回率普遍较低。这可能是由于 CADetector 方法通过预先定义的方式获取特征，限制了对合约中隐式特征的深入挖掘。与其他方法相比，CSGDetector 综合表现最好，在不同评价指标上均获得最高值，在 F_1 值方面相对于 KOLSTM 方法提升了 7.21%。这是因为本文方法利用 Solidity 合约源码构建 XML 解析树，并通过图注意力网络捕捉蜜罐合约的结构特征和内容特征，从而有效挖掘区分不同类型蜜罐合约的深层特征信息，提升针对不同类型蜜罐合约的检测能力。

4.5 教师模型及学生模型的有效性验证

为了验证教师模型及学生模型的有效性，将在完整样本集上训练的教师模型 M_t 和 3 个分别在利用式(16)和式(17)获得的样本子集上训练的学生模型 $M_s^i(i=1,2,3)$ 与本文方法 CSGDetector 进行对比，结果如表 5 所示。由表 5 可知，M_s^1、M_s^2 和 M_s^3 对应结果相近，且均明显低于 M_t 对应结果，这是由于教师模型在训练的过程中使用了训练集中的

表 4 不同方法的多分类实验结果比较

类型	方法	指标	TDO	SESL	HT	US	BD	ID	SMC	HSU	均值
基于其他特征的蜜罐合约检测	LGBM	精确率	0	0	100%	100%	100%	100%	97.73%	100%	74.72%
		召回率	0	88.89%	91.07%	90.55%	100%	82.39%	92.21%	92.57%	79.71%
		F_1 值	0	0	95.33%	95.04%	100%	90.34%	94.89%	96.14%	71.47%
	KOLSTM	精确率	98.86%	94.00%	91.91%	88.17%	92.42%	73.68%	87.00%	88.71%	89.34%
		召回率	100%	100%	100%	91.98%	89.17%	80.77%	98.86%	82.09%	92.86%
		F_1 值	99.43%	96.91%	95.79%	90.03%	91.04%	77.06%	92.55%	85.27%	91.01%
基于源码的蜜罐合约检测	CADetector	精确率	100%	100%	100%	100%	100%	100%	86.96%	91.55%	97.31%
		召回率	100%	80.00%	100%	83.02%	100%	70.42%	35.09%	99.41%	83.49%
		F_1 值	100%	88.89%	100%	90.72%	100%	82.64%	50.00%	95.32%	88.45%
	CSGDetector	精确率	100%	100%	100%	96.30%	100%	98.57%	91.94%	99.61%	98.30%
		召回率	100%	100%	91.30%	98.11%	100%	97.18%	100%	99.22%	98.23%
		F_1 值	100%	100%	95.45%	97.20%	100%	97.87%	95.80%	99.41%	98.22%

全部普通合约。对于融合了教师模型和学生模型的 CSGDetector 方法而言，其在精确率和 F_1 值方面均获得显著提升。相对于 M_t 而言，CSGDetector 在精确率、召回率、F_1 值方面分别提升 3.02%、2.37% 及 2.71%；相对于 M_s^2 而言，CSGDetector 在精确率、召回率、F_1 值方面分别提升 8.44%、0.05%及 4.55%。可见，由于教师模型和学生模型能有效挖掘完整样本集中蕴含的全局特征信息与样本子集中蕴含的局部特征信息，因此 CSGDetector 能有效融合不同类型特征在区分不同类型合约方面的优势，继而提升针对蜜罐合约的检测能力。

表 5　教师模型、学生模型及本文方法的对比实验结果

模型	精确率	召回率	F_1 值
M_t	92.63%	95.72%	94.15%
M_s^1	86.36%	**99.35%**	92.40%
M_s^2	87.21%	98.04%	92.31%
M_s^3	88.37%	**99.35%**	93.54%
CSGDetector	**95.65%**	98.09%	**96.86%**

4.6　掩蔽蜜罐检测实验

为验证本文方法在学习已知蜜罐合约基础上对于识别未知类型蜜罐的泛化能力，这里针对数据集中 10 个蜜罐类型分别进行掩蔽，并将本文方法 CSGDetector 与 XGB[7]方法进行对比。由于测试集仅包含蜜罐合约，因此在该实验中仅通过召回率来评估不同方法的泛化能力，实验结果如图 11 所示。由图 11 可知，针对跳空字符串（SESL）、隐藏转移（HT）、未初始化结构（US）、余额混乱（BD）、继承障碍（ID）、稻草人合约（SMC）和隐藏状态更新（HSU）类型的蜜罐合约，CSGDetector 的表现均优于 XGB，例如，针对余额混乱（BD）类型，CSGDetector 相较于 XGB 提升了 15%；针对跳空字符串（SESL）类型，CSGDetector 相较于 XGB 提升了 10%。进一步计算得知，CSGDetector 在掩蔽蜜罐检测实验中针对不同类型蜜罐合约的平均召回率提升了约 7.57%，这说明不同类型蜜罐合约之间存在共性特征，且因为 CSGDetector 综合考虑了词元语义信息及层次结构关系，从而能基于已知类别的数据有效捕捉蜜罐合约的深层隐式特征，对于提高方法泛化能力、发现新型蜜罐具有重要意义。

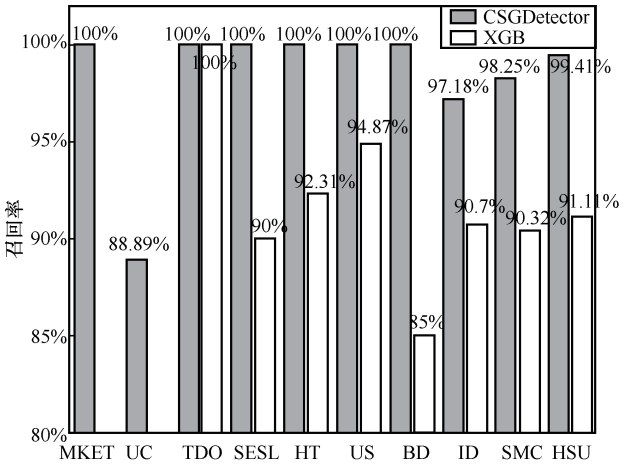

图 11　本文方法与 XGB 方法的掩蔽实验对比结果

5　结束语

本文提出了一种基于源码结构和图注意力网络的以太坊蜜罐合约检测方法 CSGDetector。为充分考虑 Solidity 源码的内容特征和结构特征，将源码转化为 XML 解析树，筛选出特征词集合，并构造合约源码结构图；为避免数据集不平衡性带来的影响，基于集成学习理论引入教师模型和学生模型的概念，分别从全局和局部的角度训练图注意力网络模型，最后融合所有模型的输出作为最终检测结果。实验表明，CSGDetector 获得了较高的精确率、召回率和 F_1 值，综合表现明显优于现有的典型蜜罐合约检测方法。此外，在掩蔽蜜罐检测实验中，CSGDetector 表现显著优于 XGB，验证了其具有较强的泛化性能。未来可将 CSGDetector 推广至以太坊中的诈骗检测、漏洞和陷阱攻击检测等相关领域。

参考文献：

[1] HEWA T , YLIANTTILA M, LIYANAGE M. Survey on blockchain based smart contracts: applications, opportunities and challenges[J]. Journal of Network and Computer Applications, 2021, 177: 102857.

[2] WOOD G. Ethereum: a secure decentralised generalised transaction ledger[J]. Ethereum Project Yellow Paper, 2014, 151(2014): 1-32.

[3] WANG S, YUAN Y, WANG X, et al. An overview of smart contract: architecture, applications, and future trends[C]//Proceedings of 2018 IEEE Intelligent Vehicles Symposium (IV). Piscataway: IEEE Press, 2018: 108-113.

[4] ZHENG Z B, XIE S A, DAI H N. An overview on smart contracts: challenges, advances and platforms[J]. Future Generation Computer Systems, 2020, 105: 475-491.

[5] CHRISTIDIS K, DEVETSIKIOTIS M. Blockchains and smart contracts for the Internet of things[J]. IEEE Access, 2016, 4: 2292-2303.

[6] TORRES C F, STEICHEN M. The art of the scam: demystifying

honeypots in Ethereum smart contracts[C]//Proceedings of the 28th USENIX Security Symposium. Berkeley: USENIX Association, 2019: 1591-1607.

[7] CAMINO R, TORRES C F, BADEN M, et al. A data science approach for detecting honeypots in Ethereum[C]//Proceedings of 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). Piscataway: IEEE Press, 2020: 1-9.

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

[9] 张红霞, 王琪, 王登岳, 等. 基于深度学习的区块链蜜罐陷阱合约检测[J]. 通信学报, 2022, 43(1): 194-202.
ZHANG H X, WANG Q, WANG D Y, et al. Honeypot contract detection of blockchain based on deep learning[J]. Journal on Communications, 2022, 43(1): 194-202.

[10] 冀甜甜, 方滨兴, 崔翔, 等. CADetector: 跨家族的各项异性合约蜜罐检测[J]. 计算机学报, 2022, 45(4): 877-895.
JI T T, FANG B X, CUI X, et al. CADetector: cross-family anisotropic contract honeypot detection method[J]. Chinese Journal of Computers, 2022, 45(4): 877-895.

[11] 韩松明, 梁彬, 黄建军, 等. DC-Hunter: 一种基于字节码匹配的危险智能合约检测方案[J]. 信息安全学报, 2020, 5(3): 100-112.
HAN S M, LIANG B, HUANG J J, et al. DC-hunter: detecting dangerous smart contracts via bytecode matching[J]. Journal of Cyber Security, 2020, 5(3): 100-112.

[12] HUANG J J, HAN S M, YOU W, et al. Hunting vulnerable smart contracts via graph embedding based bytecode matching[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 2144-2156.

[13] HU H W, BAI Q L, XU Y D. SCSGuard: deep scam detection for ethereum smart contracts[C]//Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Piscataway: IEEE Press, 2022: 1-6.

[14] TIKHOMIROV S, VOSKRESENSKAYA E, IVANITSKIY I, et al. SmartCheck: static analysis of Ethereum smart contracts[C]//Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain. New York: ACM Press, 2018: 9-16.

[15] WU Z H, PAN S R, CHEN F W, et al. A comprehensive survey on graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(1): 4-24.

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

[17] CHAWLA N V, BOWYER K W, HALL L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research, 2002, 16: 321-357.

[18] WEI J, ZOU K. EDA: easy data augmentation techniques for boosting performance on text classification tasks[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Stroudsburg: Association for Computational Linguistics, 2019: 6382-6388.

[作者简介]

王友卫（1987− ），男，山东临沂人，博士，中央财经大学副教授、硕士生导师，主要研究方向为内容安全、深度学习、数据挖掘等。

侯玉栋（2001− ），男，山西吕梁人，中央财经大学硕士生，主要研究方向为深度学习、区块链技术等。

凤丽洲（1986− ），女，吉林长春人，博士，天津财经大学副教授、硕士生导师，主要研究方向为文本分析、数据挖掘等。