文章编号： 1673-3193（2024）05-0601-07

基于特征调制图神经网络的智能合约源码漏洞检测

师自通，师智斌，刘冬明，石　琼，龚晓元

（中北大学 计算机科学与技术学院，山西 太原 030051）

摘　要： 随着区块链技术的广泛应用，智能合约的安全问题引起广泛关注。针对智能合约源码向字节码转化会丢失部分语义信息，而现有深度学习漏洞检测方法不能很好地检测重入漏洞和时间戳漏洞等问题，本文提出一种基于特征调制图神经网络的智能合约源码漏洞检测方法（GNN-film）。首先，分析重入漏洞和时间戳漏洞的特点，使用智能合约源码构建图结构并将其精简化；其次，搭建基于特征级线性调制的图神经网络模型，利用该网络模型强大的特征调制能力对合约漏洞特征进行精确表示；最后，将精简化后的图结构数据输入搭建的模型中获得检测结果。实验结果显示，本文方法对重入漏洞和时间戳漏洞检测的准确率达到91.00％和91.64％，相较基于图神经网络的方法分别提升了4.20百分点和9.70百分点，证明本文方法对相关漏洞检测的能力要优于其他检测工具。

关键词： 智能合约；漏洞检测；重入漏洞；时间戳漏洞；特征调制图神经网络

中图分类号： TP 309；TP 311.13　　**文献标识码：** A　　**doi:** 10.3969/j.issn.1673-3193.2024.05.006

引用格式： 师自通，师智斌，刘冬明，等. 基于特征调制图神经网络的智能合约源码漏洞检测[J]. 中北大学学报（自然科学版），2024，45（5）：601-607.

SHI Zitong，SHI Zhibin，LIU Dongming，et al. Feature-wise modulation graph neural network smart contract vulnerability detection based on source code[J]. Journal of North University of China（Natural Science Edition），2024，45（5）：601-607.

Feature-Wise Modulation Graph Neural Network Smart Contract Vulnerability Detection Based on Source Code

SHI Zitong，SHI Zhibin，LIU Dongming，SHI Qiong，GONG Xiaoyuan

（School of Computer Science and Technology，North University of China，Taiyuan 030051，China）

Abstract： With the wide use of blockchain technology，the security of smart contracts has attracted wide attention. The conversion of smart contract source code to bytecode will lose some semantic information，and the existing deep learning vulnerability detection methods cannot detect reentrancy vulnerabilities and timestamp vulnerabilities well. This paper proposed a smart contract source vulnerability detection method（GNN-film）based on feature-wise modulation graph neural network. Firstly，the characteristics of reentrancy vulnerabilities and timestamp vulnerabilities were analyzed，the graph structure was constructed and simplified by using smart contract source code. Secondly，constructing the model of feature-wise linear modulation graph neural network，and getting accurate representation of contract vulnerability features by

收稿日期： 2023-11-22

作者简介： 师自通（1996－），男，硕士生，主要从事智能合约、图神经网络方面的研究。

通信作者： 师智斌（1971－），女，副教授，博士，主要从事网络安全、信息处理方面的研究。E-mail：1637350520@qq.com。

using the powerful feature modulation ability of the model. Finally, put simplified graph structure data into the model to obtain the detection results. The experimental results show that the detection accuracy of reentrancy vulnerability and timestamp vulnerability is 91.00% and 91.64% respectively, which is 4.20 and 9.70 percentage points higher than that of graph neural network method. It is proved that the detection ability of this method for related vulnerabilities is better than other detection tools.

Key words: smart contract; vulnerability detection; reentrancy vulnerability; timestamp vulnerability; feature-wise linear modulation graph neural network

0 引 言

随着比特币[1]的成功应用，其底层的区块链技术开始进入研究者的视野。智能合约作为区块链技术最为成功的应用引起学者的广泛关注[2]，其中合约本身的安全问题成为了研究热点。

智能合约漏洞包括错误异常漏洞、拒绝服务漏洞、以太冻结漏洞、重入漏洞、时间戳漏洞等[3]，包含漏洞的合约一旦被攻击者利用，就会造成巨大的数字资产损失。由于合约被部署便无法更改，且会在以太坊中自动执行的特性，在部署之前对智能合约漏洞检测的工作十分必要。

现有的智能合约漏洞检测方法包括形式化验证、符号执行、模糊测试、中间表示法、深度学习方法等。VaaS是基于形式验证的一键式检测工具，采用形式化验证检测智能合约漏洞[4]；Osiris是采用符号执行方法的静态分析框架，由符号分析、污染分析和整数错误检测组成[5]；Contract-Fuzzer是基于模糊测试的智能合约漏洞检测方法，在合约执行时，通过记录、分析智能合约的行为来检测漏洞[6]；SmartCheck将智能合约源代码转化为基于xml的中间表示，根据Xpath模式进行漏洞检测[7]。上述检测方法存在严重依赖专家规则、检测精度不高、效率低下等问题。

RecChecker[8]是深度学习方法，该方法将智能合约切片之后，传入带注意力机制的双向长短期记忆神经网络中进行漏洞检测。沈晨凯[9]的研究使用由智能合约抽象语法树转化的结构化序列，输入Bi-GRU模型中进行漏洞检测。张铮等[10]搭建了基于CNN-LSTM网络模型，使用由智能合约字节码转化的操作码序列进行漏洞检测。赵波等[11]的研究，除了使用字节码生成节点特征，还使用长短期记忆神经网络提取字节码的语义特征，综合两种向量表示进行智能合约漏洞检测。白英民等[12]首次将时间序列引入智能合约漏洞检测领域，该方法将合约源码转化为操作码序列，提取操作码执行的Shapelet序列，同时嵌入词向量，使用机器学习方法对智能合约进行漏洞检测。对上述研究分析可知，基于字节码的智能合约漏洞检测方法在向字节码转化的过程中，会丢失部分语义、语法信息，导致漏洞检测的精度较低。另外，基于文本序列的深度学习方法，无法保留合约代码中的漏洞逻辑，也严重影响智能合约漏洞检测的精度。

针对上述智能合约漏洞检测方法存在的不足，本文基于源码，采用特征级线性调制的图神经网络[13](Feature-wise Linear Modulation Graph Neural Network，GNN-film)模型对智能合约进行漏洞检测。该网络是一种利用深度学习直接对图结构数据进行学习的框架。其最大的优势在于对节点信息的特征调制，经调制后的节点信息能够精确地表示图结构信息。对于程序片段、代码采用图结构化表示不但能够保留代码的逻辑与结构，而且能够保留程序的语法语义信息，还能使用图神经网络进行分析推理，相较于字节码和文本序列方法，能学习图结构数据中更深层的数据信息，而采用特征级线性调制图神经网络能够在原有图神经网络上，对节点信息进行特征调制，进而获得图结构信息的精确表示，从而提高智能合约漏洞检测的精度。

1 整体模型

本文方法分以下三个步骤实现：1）将智能合约源码进行图结构化表示并精简化，同时构建图数据集。2）搭建基于特征级线性调制图神经网络的智能合约漏洞检测模型，利用该模型的线性级特征调制模块获取更精确的图结构表示。3）向模型输入精简化的图结构数据，最终得到检测结果。本文方法的整体流程如图1所示。

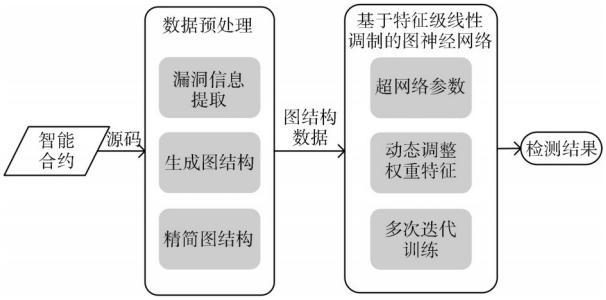

图 1　基于特征级线性调制的图神经网络智能合约漏洞检测框架

Fig. 1　Smart contract vulnerability detection framework of feature-wise linear modulation graph neural network

2　智能合约漏洞概述

智能合约是一种预先编写的、部署后可自动执行的代码。由于合约上链之后的不可更改性，合约本身存在漏洞会造成严重经济损失。智能合约重入漏洞和时间戳漏洞是两种常见且危害较高的漏洞，现有方法没有对其原理进行针对性研究，本章介绍两类漏洞的原理。

2.1　重入漏洞

重入漏洞是由于编码执行流程不当导致的。被攻击合约在执行完转账操作 .call.value()()后，合约中的 withdraw()函数会改变合约中余额变量状态使交易终止，并自动调用 fallback()函数，而攻击合约会在 fallback()函数中调用 withdraw()函数，使余额变量状态无法改变，交易持续进行从而盗取被攻击合约中的所有数字资产。

2.2　时间戳漏洞

时间戳漏洞也是智能合约中常见的一种漏洞，合约中包含不受信任控制范围约束的功能函数。合约中使用 block.timestamp 变量来表示当前块的时间戳，该戳是由矿工打包区块时设置的，若恶意矿工利用该变量间接作为下一区块的出块条件，就会获取非法收益，此时合约具有时间戳漏洞。例如：恶意矿工利用 block.timestamp 变量编写一个赌注合约，当该变量可以被 66 整除时可以获得出块奖励，恶意矿工就可以提前计算获得奖励的出块时间，依据该时间调整当前合约的发布时间而获得非法收益。

3　数据预处理

3.1　漏洞信息提取

本文使用智能合约源码，依据相关的漏洞模式，提取源码中代表漏洞信息的元素，依此来构建列表映射。漏洞模式是在已知特定漏洞发生条件的情况下，选取源码中函数、变量等相关元素，根据选取元素之间的执行、返回关系构建的判定模式。

重入漏洞模式为：合约代码中包含 withdraw()功能函数，且该函数包含转账函数 .call.value()()，执行转账操作后，才会更新账户余额变量 balances[msg.sender]。而时间戳漏洞模式为：合约代码中使用 block.timestamp 变量（该变量为当前块时间）直接或间接进行算数运算。以重入漏洞为例，如图 2 中Ⅰ所示，依据重入漏洞模式可以提取代码片段中的重入漏洞信息。

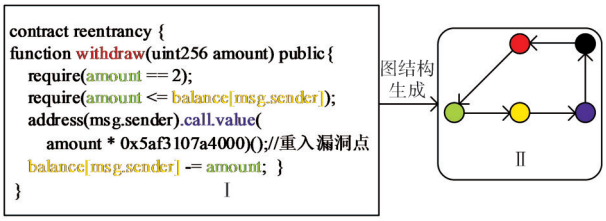

图 2　重入漏洞信息提取与图结构生成

Fig. 2　Reentrancy vulnerability information extraction and graph structure generation

3.2　生成图结构

Allamanis 等[14]研究表明，可以使用源代码生成图结构，保留源代码中的语义、语法信息。王守梁[15]的研究证明图结构中包含更丰富的数据信息，能体现代码深层次的特征。故本文将智能合约源码转化为图结构，相较文本序列结构数据，图结构数据能提取程序的漏洞逻辑，可以保留漏洞的语法、语义信息。

本文根据 3.1 节提取的漏洞信息，依据合约代码的执行顺序、元素之间的调用关系生成图结构。如图 2 中Ⅰ、Ⅱ所示，Ⅰ中使用彩色标识的元素与Ⅱ中彩色节点对应，而Ⅱ中黑色节点代表 fallback()函数，该函数在执行合约后会自动调用，本文设置自动生成该节点。

3.3　精简图结构

孙逊[16]的研究提出如下观点：在同一函数中，不同程序元素可分为不同级别，可以消去非核心级别的节点。图结构中每一节点在图神经网络模型训练过程中被视为一个可计算函数，过多的节点不但会增加神经网络的训练时间，而且会影响最终检测结果的精度。

在 3.2 节中生成的图结构中，导致重入漏洞

产生的元素为.call.value()()、withdraw()以及合约功能执行后自动调用的fallback()，而其余元素间接影响重入漏洞的产生。常见的图结构精简方法有节点消去法、邻接矩阵规范化、图规范化等。本文采用节点消去操作进行图精简化。具体操作为：保留图结构中的核心节点(例如重入漏洞中.call.value()()、withdraw()等节点)，消去其余节点并将此类节点信息聚合到邻域中的核心节点。如图3中Ⅲ所示，红色与紫色节点代表核心节点，绿色和橙色节点代表其余节点，椭圆图形将其余节点表示的信息聚合到核心节点处，同时消去其余节点，最终图像如Ⅳ所示。

最后本文使用word-to-vector方法对精简后的图结构数据进行向量化表示，经过向量化表示之后的图数据可用于图神经网络训练。

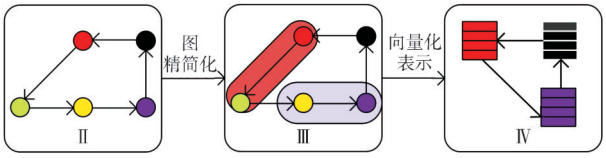

图3 重入漏洞图结构精简与向量化表示

Fig. 3 Reentrancy vulnerability graph structure simplification and vectorization representation

4 漏洞检测模型搭建

基于特征级线性调制的图神经网络是一种图神经网络的变体，其核心是使用超网络[17]完成聚合关于目标节点的表示同时更新最相关的特性，在训练过程中，允许目标节点对来自边的信息表示动态地调整权重特征。图4为本文漏洞检测模型。

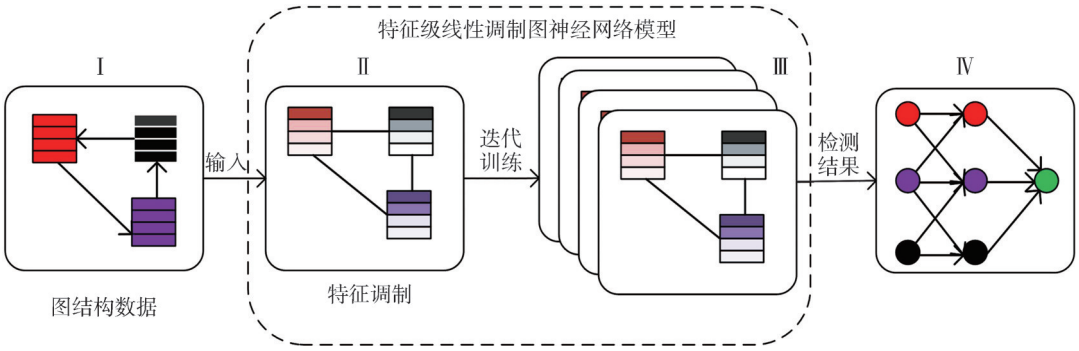

图4 漏洞检测模型

Fig. 4 Vulnerability detection model

将向量化表示的图结构数据输入漏洞检测模型中进行训练，图中Ⅱ表示数据经过一轮训练之后，图节点的向量表示发生变化，经过多次迭代训练后，使用读出函数计算图结构的信息表示，最后通过sigmoid函数得出检测结果。

超网络本质上是一个高阶函数，可以对图结构中目标节点获取的邻域信息动态更新权重，通过信息聚合，从而获得更精确的节点信息表示。在实际智能合约漏洞检测过程中，超网络是一个聚焦于目标节点的计算函数，在消息传播阶段，使用计算消息传播函数来计算权重，更新目标节点的最相关特征信息，聚合图结构中所有节点的信息，最后根据该信息检测合约本身是否存在某类漏洞。该阶段涉及的公式如式(1)所示。

$$\gamma_{\xi,v}^{(t)},\beta_{\xi,v}^{(t)}=g(h_v^{(t)}), \tag{1}$$

式中：$g()$代表可学习函数；$\gamma_{\xi,v}^{(t)}$、$\beta_{\xi,v}^{(t)}$为256维向量，代表元素级仿射转换，由所在边消息计算单元求得，用来调整权重特征，其中上标代表第t次迭代，下标代表当前网络指向目标节点v的各种边类型ξ；$h_v^{(t)}$代表目标节点第t次迭代的向量表示。

在图级读出阶段，即漏洞判定阶段，所涉及的公式如式(2)所示。

$$h_v^{(t+1)}=l\left(\sum_{u\xrightarrow{\xi}v\in\varepsilon}\sigma(\gamma_{\xi,v}^{(t)}\odot W_\xi h_u^{(t)}+\beta_{\xi,v}^{(t)})\right), \tag{2}$$

式中：W_ξ为ξ的权值；$h_u^{(t)}$为目标节点v的邻域节点第t次迭代的向量表示；σ为应用非线性函数；l为应用有界非线性函数；$h_v^{(t+1)}$为第$t+1$次迭代训练后v的节点表示。

在漏洞图结构训练的过程中，同一节点的信息表示随着训练次数的增加而动态地发生变化，利用特征级线性调制模块不断对节点信息进行调制，从而达到最优节点信息表示，最终获得模型的最佳检测结果。

图5为节点信息计算图示。图5中，左侧图像为原始智能合约漏洞图结构信息表示，右侧图像为下一轮次的图结构信息表示。图中绿色实线为节点

表示的自更新，虚线代表数据流，指向绿色节点的三条有向边中的数字代表不同类型的边关系，绿色节点为观察节点，黄色、红色、蓝色节点为邻域节点。

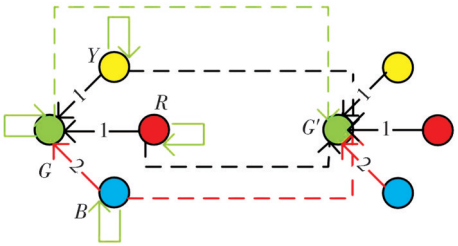

图 5　节点信息计算图示

Fig. 5　Node information calculation diagram

在一轮信息传递调整之后，绿色节点的表示信息会与原有信息不同。令自更新边，即图 5 中的绿色实线，G、B、Y、R分别代表对应节点的表示信息。采用传统图神经网络的节点信息表示为式(3)，而使用基于特征级线性调制的图神经网络的节点信息表示为式(4)。图中边上数字代表边的不同类型。

$$G'=\sigma(G+W_2*B+W_1*Y+W_1*R), \tag{3}$$

$$G'=\sigma(\beta_{0,G}+\gamma_{0,G}\odot W_0*G+\beta_{1,G}+\gamma_{1,G}\odot W_1*Y+\beta_{1,G}+\gamma_{1,G}\odot W_1*R+\beta_{2,G}+\gamma_{2,G}\odot W_2*B)。 \tag{4}$$

从式(3)可以看出，计算传统图神经网络中的节点表示仅将观察节点表示和邻域节点表示与权重矩阵的乘积相加，忽略观察节点自身的表示更新以及邻域节点表示的更新，导致图神经网络的漏洞检测结果低于基于特征级线性调制的图神经网络的结果，在 5.4 节实验分析部分也证实了这一结论，同时证明了本文模型检测重入漏洞和时间戳漏洞的优越性。

将图结构中所有节点表示聚合得到图结构的表示向量h_G，最终获得智能合约的检测结果y，如式(5)所示。

$$y=sigmoid(h_G)。 \tag{5}$$

5　实验分析

5.1　数据集

本文数据集将 SmartBugs Wild Dataset[18] 和 Tianchi Smart Contract Dataset 混合，选取 12 000 条智能合约构建重入漏洞数据集，14 000 条智能合约构建时间戳漏洞数据集。由于 Tianchi Smart Contract Dataset 不是智能合约源码文件，所以需要利用合约地址属性在 https://cn.etherscan.com/网站上下载源码文件。

5.2　实验指标

本文实验采用准确率（Accuracy）、精确率（Precision）、召回率（Recall）、$F1$ 分数（$F1$ score）四种评价指标。

$$Accurancy=\frac{N_{TP}+N_{TN}}{N}, \tag{6}$$

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

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

$$F1=\frac{N_{TP}}{2N_{TP}+N_{FP}+N_{FN}}, \tag{9}$$

式中：N_{TP}，N_{TN}，N_{FP}，N_{FN}分别为真阳样本、真阴样本、假阳样本、假阴样本的数量；N为样本总数。

5.3　实验环境

本文实验机器为一台处理器为 Intel（R）Core（TM）i7-11800H，16 G 内存，使用 RTX-3060 显卡的电脑，实验软件为 Pycharm2022，实验环境使用 python3.7、TensorFlow1.14.0、cuda11.7、keras2.2.4 等。实验参数设置如表 1 所示。

表 1　实验参数设置

Tab. 1　Experimental parameter settings

参数	GNN-film	GNN	GCN	GAT
hidden_size	256	256	256	256
graph_activation_function	ReLu	tanh	Relu	tanh
message_aggregation_function	sum	sum	sum	sum
graph_cell	/	RNN	/	/
optimizer	Adam	Adam	Adam	Adam
learning_rate	0.000 2	0.000 2	0.000 2	0.000 2
Input_dropout	0.9	0.9	0.9	0.9
epoch	60	60	60	60

5.4　实验结果分析

本节对不同漏洞分别进行检测实验，使用本文方法（GNN-film）与 Smartcheck、oyente、Mythril 三种传统检测方法，与 LSTM、GRU 两种基于深度学习文本序列方法以及 GNN、GCN、GAT 三种基于图神经网络方法进行对比实验。表 2 展示了 9 种方法对重入漏洞检测的实验结果，图 6 为 GNN-film 方法的重入漏洞损失曲线图。

从表 2 中的实验数据可以得出，基于深度学

习的LSTM和GRU方法的检测精度要高于Smartcheck、oyente、Mythril的传统智能合约漏洞检测方法，而GNN-film方法的各项指标要优于深度学习方法。其中，本文方法的Accuracy、Precision、Recall和*F*1 Score分别达到了91.00%，91.73%，83.94%和87.66%，相较于GNN方法分别提高了4.20，6.12，5.91和6.02百分点。

表2 检测重入漏洞的实验结果

Tab. 2 Performance comparison for detecting reentrancy vulnerabilities

方法模型	Accuracy/%	Precision/%	Recall/%	*F*1/%
Smartcheck	50.32	39.27	56.18	46.23
oyente	55.34	40.62	54.17	46.43
Mythril	60.48	44.18	52.78	48.10
LSTM	66.68	60.63	55.36	57.88
GRU	70.76	68.51	57.29	62.40
GNN	86.80	85.61	78.03	81.64
GCN	88.56	80.70	82.88	81.78
GAT	89.95	85.92	81.33	83.56
GNN-film	**91.00**	**91.73**	**83.94**	**87.66**

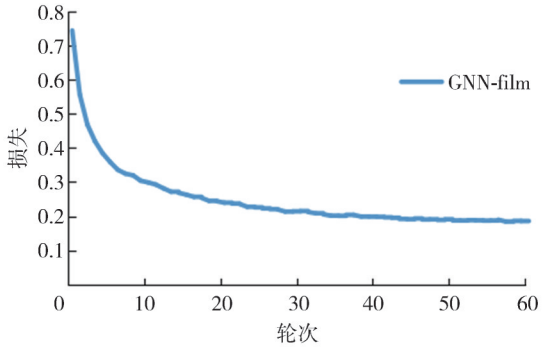

图6 重入漏洞检测的损失曲线图像

Fig. 6 Loss curve image of reentrancy vulnerability detection

表3展示了9种方法对时间戳漏洞检测的实验结果，图7为GNN-film方法的时间戳漏洞损失曲线图。

表3 检测时间戳漏洞的实验结果

Tab. 3 Performance comparison for detecting timestamp vulnerabilities

方法模型	Accuracy/%	Precision/%	Recall/%	*F*1/%
Smartcheck	50.51	38.16	58.27	46.12
oyente	58.39	40.41	56.28	47.04
Mythril	64.21	48.32	56.64	52.15
LSTM	65.72	51.23	59.20	54.93
GRU	69.12	63.14	61.04	62.39
GNN	81.94	80.04	83.25	81.61
GCN	83.66	83.63	80.39	81.98
GAT	85.47	88.84	78.78	83.51
GNN-film	**91.64**	**91.12**	**91.63**	**91.37**

从表3中的实验数据可以得出，GNN-film方法的各项指标要优于其他漏洞检测方法。其中，本文方法的Accuracy、Precision、Recall和*F*1 Score分别达到了91.64%，91.12%，91.63%和91.37%，相较于GNN方法分别提高了9.70，11.08，8.38和9.76百分点。

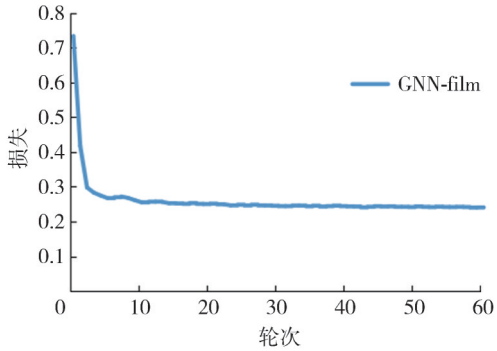

图7 时间戳漏洞的损失曲线图像

Fig. 7 Loss curve image of timestamp vulnerability detection

6 结 论

本文提出一种基于特征调制图神经网络的智能合约源码漏洞检测方法。文中对该方法的原理和主要优势进行了详细分析。该方法对智能合约进行更加精确的图结构表示，从而获得更加准确的漏洞检测结果。理论分析及仿真结果表明：

1）图结构数据能够表达智能合约源码中更深层次的信息；

2）在同一实验环境下，图神经网络方法要优于传统漏洞检测方法以及基于深度学习文本序列漏洞检测方法；

3）增加特征调制模块后的有效性证明了基于特征级线性调制的图神经网络执行漏洞检测任务具有更高准确性。

参考文献：

[1] WRIGHT C S. Bitcoin：A peer-to-peer electronic cash system [J]. SSRN Electronic Journal，2008：1-9.

[2] DI PIERRO M. What is the blockchain?[J]. Computing in Science & Engineering，2017，19(5)：92-95

[3] BUTERIN V. A next-generation smart contract and decentralized application platform[EB/OL]. [2023-11-22]. https://blog. lavoiedubitcoin. info/public/Bibliotheque/EthereumWhitePaper. pdf.

[4] GARFATTA I，KLAI K，GAALOUL W，et al. A survey on formal verification for solidity smart contracts [C]//2021 Australasian Computer Science Week Multiconference，2021：1-10.

[5] TORRES C F，SCHUTTE J，STATE R. Osiris：Hunting for integer bugs in ethereum smart contracts

[C]//Proceedings of the 34th Annual Computer Security Applications Conference, 2018, 664-676.

[6] JIANG B, LIU Y, CHAN W K. Contractfuzzer: Fuzzing smart contracts for vulnerability detection [C]//33rd ACM/IEEE International Conference on Automated Software Engineering (ASE). IEEE: 2018, 259-269.

[7] TIKHOMIROV S, VOSKRESENSKAYA E, IVANITSKIY I, et al. SmartCheck: static analysis of ethereum smart contracts [C]//Proceedings of the 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). Gpthenburg: IEEE,2018: 9-16.

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

[9] 沈晨凯. 基于深度学习的智能合约漏洞检测方法研究[D]. 武汉: 武汉大学, 2021.

[10] 张铮, 张星娜, 吕卓, 等. 基于深度学习的智能合约漏洞检测方法[J]. 重庆邮电大学学报(自然科学版), 2022, 34(5): 914-920.
ZHANG Zheng, ZHANG Xingna, LYU Zhuo, et al. Detecting vulnerabilities in smart contracts based on deep learning models[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2022,34(5): 914-920. (in Chinese)

[11] 赵波, 上官晨晗, 彭小燕, 等. 基于语义感知图神经网络的智能合约字节码漏洞检测方法[J]. 工程科学与技术, 2022, 54(2): 49-55.
ZHAO Bo, SHANGGUAN Chenhan, PENG Xiaoyan, et al. Semantic-aware graph neural network for smart contract bytecode vulnerability detection[J]. Advanced Engineering Sciences, 2022, 54(2): 49-55. (in Chinese)

[12] 白英民, 师智斌, 信文阁, 等. 基于词嵌入与 Shapelet 时序特征的智能合约漏洞检测方法研究[J]. 中北大学学报(自然科学版), 2023, 44(4): 381-387.
BAI Yingmin, SHI Zhibin, XIN Wenge, et al. Research on smart contract vulnerability detection method based on word embedding and shapelet time series features [J]. Journal of North University of China (Natural Science Edition), 2023, 44(4): 381-387. (in Chinese)

[13] BROCKSCHMIDT M. GNN-FiLM: Graph neural networks with feature-wise linear modulation [DB/OL]. (2019-06-28)[2023-11-22]. http://arxiv.org/abs/1906.12192v2.

[14] ALLAMANIS M, BROCKSCHMIDT M, KHADEMI M. Learning to represent programs with graphs. [DB/OL]. (2017-11-01) [2023-11-22]. http://arxiv.org/abs/1711.00740v2.

[15] 王守梁. 基于代码序列与图结构的源代码漏洞检测方案[J]. 中北大学学报(自然科学版), 2023, 44(6): 641-653.
WANG Shouliang. A source code vulnerability detection scheme based on code sequence and graph structure[J]. Journal of North University of China (Natural Science Edition), 2023, 44(6): 641-653. (in Chinese)

[16] 孙逊. 基于图注意力网络的智能合约漏洞检测[D]. 成都: 电子科技大学, 2022.

[17] HAMILTON W L, YING R, LESKOVEC J. Inductive representation learning on large graphs[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017: 1025-1035.

[18] DURIEUX T, FERREIRA J F, ABREU R, et al. Empirical review of automated analysis tools on 47, 587 ethereum smart contracts[C]//Proceedings of the 2020 ACM/IEEE 42nd International Conference on Software Engineering (ICSE). Seoul: IEEE, 2020: 530-541.