基于深度学习的区块链蜜罐陷阱合约检测

张红霞，王琪，王登岳，王奔
（中国石油大学（华东）青岛软件学院、计算机科学与技术学院，山东 青岛 266580）

摘 要：针对当前检测方法准确率不高以及模型泛化性较差的问题，提出了基于 KOLSTM 深度学习模型的蜜罐陷阱合约检测方法。首先，通过分析蜜罐陷阱合约的特点，提出了关键操作码的概念，并设计了可用于选取智能合约中关键操作码的关键词提取方法；其次，在传统的 LSTM 模型中加入关键操作码权重机制，构建了可以同时捕获蜜罐陷阱合约中隐藏的序列特征以及关键操作码特征的 KOLSTM 模型。最后，通过实验表明，该模型具有较高的识别精确率，在二分类和多分类检测场景下的 F 值较 LightGBM 模型分别提升 2.39%与 19.54%。

关键词：区块链；以太坊；智能合约；蜜罐陷阱合约；深度学习
中图分类号：TP309.2
文献标志码：A
DOI: 10.11959/j.issn.1000−436x.2022011

Honeypot contract detection of blockchain based on deep learning

ZHANG Hongxia, WANG Qi, WANG Dengyue, WANG Ben
Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum, Qingdao 266580, China

Abstract: Aiming at the problems of low accuracy of current detection methods and poor generalization of model, a honeypot contract detection method based on KOLSTM deep learning model was proposed. Firstly, by analyzing the characteristics of honeypot contract, the concept of key opcode was proposed, and a keyword extraction method which could be used to select the key opcode in smart contract was designed. Secondly, by adding the key opcode weight mechanism to the traditional LSTM model, a KOLSTM model which could simultaneously capture the sequence features and key opcode features hidden in the honeypot contract was constructed. Finally, the experimental results show that the model had a high recognition accuracy. Compared with the existing methods, the F-score is improved by 2.39% and 19.54% respectively in the two classification and multi-classification detection scenes.

Keywords: blockchain, Ethereum, smart contract, honeypot contract, deep learning

0 引言

以太坊是一个开源的公共区块链分布式平台，可以通过图灵完备的虚拟机来处理点对点之间通信的智能合约。部署在以太坊公链上的智能合约具有不可篡改和自动执行的优越特性，使智能合约和区块链相结合的研究逐渐成为热点。但区块链技术的去中心化、匿名特性[1]以及缺少第三方机构直接管控等问题，造成在交易过程中很难获知参与者的身份信息。研究表明[2]，随着智能合约在各个领域的应用越加盛行，加密数字货币的价值不断攀高，基于区块链智能合约的漏洞和陷阱攻击正变得日益猖獗。蜜罐陷阱合约是一种隐藏在以太坊智能合约面具下的新型恶意陷阱合约。由于缺乏以太坊以及智能合约等领域的专业知识，普通用户和投资者通常对这类恶意陷阱难以防备。一份关于以太坊的数据分析[3]估计，截至 2019 年，以太坊上部署着数千个蜜罐陷阱合

收稿日期：2021−10−14；**修回日期**：2022−01−05
基金项目：中石油重大科技基金资助项目（No.ZD2019-183-004）；中央高校基本科研业务费专项资金资助项目（No.20CX05019A）
Foundation Items: The Major Scientific and Technological Projects of CNPC (No.ZD2019-183-004), The Fundamental Research Funds for the Central Universities (No.20CX05019A)

约。所以，针对以太坊上蜜罐陷阱合约的检测工作刻不容缓。

近年来，研究者采用符号执行的方法或是机器学习对以太坊上部署的蜜罐智能合约进行检测[3-5]，依照蜜罐陷阱合约的不同特性将蜜罐陷阱合约划分为 8 个类别，开发了利用符号执行可用于检测蜜罐陷阱合约的工具。通过引入智能合约中的交易行为作为特征，采用决策树等算法对蜜罐陷阱合约进行检测。但目前的研究仍存在以下问题：1) 未考虑到蜜罐陷阱合约数据的类别不平衡和重复性，导致训练出的模型存在过拟合问题；2) 基于频率的特征选取方法，导致机器学习模型训练特征过于庞大，泛化性较差。

本文通过分析蜜罐陷阱合约的特点，提出了关键操作码的概念，并在蜜罐陷阱合约检测领域引入深度学习方法，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，构建了能够同时捕获蜜罐陷阱合约中隐藏的序列特征以及关键操作码特征的关键操作码长短时记忆（KOLSTM, key-opcode long short-term memory）模型，提出了基于 KOLSTM 模型的蜜罐陷阱合约检测方法。具体来说，首先，采用基于简易数据扩充（EDA, easy data augmentation）的数据增强过采样技术对蜜罐陷阱合约数据集进行有效的数据增强；然后，利用关键操作码（Key-Opcode）提取方法在嵌入层加入权重机制，有效区分操作码特征的重要性，实现对蜜罐陷阱合约的隐藏序列特征及关键操作码特征的捕获；最后，采用真实数据集进行实验，结果表明，本文提出的方法在二分类和多分类检测场景下的 F 值较现有模型分别提升 2.39%与 19.54%。

本文的主要贡献如下。

1) 提出了 KOLSTM 深度学习模型用于检测蜜罐陷阱合约，该模型可以有效地捕获蜜罐陷阱合约中隐藏的序列特征和关键操作码特征。

2) 提出了一种可用于选取智能合约中关键操作码的关键词提取方法，通过该方法，可以获取对智能合约类别有重要影响的操作码以及输出对该操作码影响大小的数值量化。

3) 在真实数据集的实验结果表明，KOLSTM 模型在二分类和多分类检测场景下的 F 值分别为 95.59%与 91.01%，较现有模型分别提升 2.39%与 19.54%。

1　相关工作

由于区块链同时具有参与者匿名、无国界限制、金融支付[6]等天然特性，再加上缺乏有力的监管手段，相较于传统的互联网恶意陷阱，基于区块链的恶意陷阱具有更高的隐蔽性和伪装性[7]。对于普通的投资者和用户来说，这类陷阱更加难以防备。据研究统计[8]，仅 2013 年 9 月 2 日—2014 年 9 月 9 日，就有超过万名受害者深陷此类陷阱。Bartoletti 等[9]通过分析 192 个基于比特币的恶意陷阱案例，将比特币区块链网络中的恶意陷阱划分为 4 种类型。Chen 等[10-11]提出利用机器学习和数据挖掘的方法来对基于以太坊的庞氏骗局进行识别，在后续的模型应用中，估计以太坊上部署着超过 400 个庞氏骗局。张艳梅等[12]在基于以太坊的庞氏骗局检测领域引入深度学习方法，进一步提高了检测精度。

而基于以太坊的蜜罐陷阱合约是 2019 年新发现的区块链恶意陷阱类型，Torres 等[3]通过调查蜜罐陷阱合约的流行情况、交易行为以及对以太坊平台的影响程度，首次对蜜罐陷阱合约进行了系统分析。根据蜜罐陷阱合约不同的特性将它们划分为了 8 个类别，并提供了一个利用符号执行可用于检测蜜罐陷阱合约的工具。该工具对于特征明显和特定类型的蜜罐陷阱合约具有很好的识别效果，但由于是基于符号执行，而且目前智能合约的发展更加趋向于简洁化与智能化，该工具对复杂场景下的蜜罐陷阱合约的有效识别较弱。Camino 等[4]提出了一种基于合约交易行为的数据科学检测方法，通过分析在合约创建者、合约、交易发送者和其他参与者之间的所有可能的资金流动情况，然后根据分析结果引入相应的特征，训练了一个用于蜜罐陷阱合约检测的机器学习模型[13]。该方法有效地利用了除智能合约源码以外的其他特征，包括合约中的交易记录、编译信息等。但该方法提出的检测模型的训练必须建立在足够多的交易记录数据特征上，而大多数智能合约中并不包含足够的用以特征提取的交易记录，一旦蜜罐陷阱合约包含足够的交易记录，就意味着可能不止一个用户深陷陷阱。Chen 等[5]认为蜜罐陷阱合约可以看作一种漏洞攻击，但已有的智能合约漏洞检测工具，包括智能审查[14]、规则检查[15]以及合约模糊器[16]，却无法针对蜜罐陷阱合约进行有效检测。该研究从机器学习的角度出发，提出了一种基于 N-gram 模型[17]选取操作码特征，

并结合梯度提升决策树算法[18]来检测蜜罐陷阱合约的方法。该方法利用智能合约源码反编译得到的操作码作为特征进行模型训练，可以在字节码级别学习蜜罐陷阱合约的模式。但该方法未考虑到蜜罐陷阱合约数据存在的类别不平衡性和重复性[19]问题，以及基于 N-gram 模型选取特征所导致的训练特征太过庞大，对新出现的蜜罐陷阱合约识别效果不佳。本文针对当前研究存在的问题，分析蜜罐陷阱合约的特点，基于深度学习方法构建检测模型，以期提高蜜罐陷阱合约检测的准确度。

2 KOLSTM 模型

2.1 Key-Opcode 提取方法

Key-Opcode 提取方法是 KOLSTM 模型的重要组成部分，利用该方法可以获取对智能合约类别有重要影响的操作码，并对操作码影响大小进行数值量化处理。智能合约中的操作码是表示以太坊基于堆栈虚拟机的执行命令，不同类型的智能合约中的操作码以及操作码频率具有较大差异，如图 1 所示，蜜罐陷阱合约中表示判断以及跳转命令的操作码频率要比非蜜罐陷阱合约高得多。基于此，本文认为不同类型的智能合约中所包含的操作码及其频率是不同的，存在某些操作码对合约类别有着重要影响。这类操作码就是 Key-Opcode，其定义为智能合约中对合约类型判定起关键性作用的操作码。

(a) 蜜罐陷阱合约 (b) 非蜜罐陷阱合约

图 1 蜜罐陷阱合约与非蜜罐陷阱合约操作码词云

针对智能合约操作码的词频无法有效衡量操作码的重要程度，智能合约数据的长度分布毫无规律，且类别不平衡，无法对关键操作码进行有效度量等问题，本文基于逆向文本频率（IDF, inverse document frequency）[20]，提出选取智能合约关键操作码的 Key-Opcode 提取方法 sigmoid 逆向文本频率（SIDF, sigmoid inverse document frequency），通过 sigmoid 函数对初始权重值进行缩放处理，将操作码的权重值控制在[0,1]范围内，进而确定操作码对合约类别的影响程度，计算方法为

$$\mathrm{SIDF}_o = \mathrm{sigmoid}\left(\log\left(\frac{D}{D_o+1}\right)\lambda\right) \tag{1}$$

其中，D 为合约总数，D_o 为包含操作码 o 的合约数，λ为常数。

利用该方法对数据集进行关键操作码选取，创建权重字典并应用到之后的模型中。表 1 是权重字典的部分输出结果。

表 1 权重字典的部分输出结果

操作码	权重值
'SWITCH'	0.774 8
'JUMP'	0.575 1
'CALL'	0.601 9
'DUP'	0.499 9
'CALLDATASIZE'	0.507 6
'DIV'	0.501 0
'SHA'	0.627 4
'ISZERO'	0.497 7
'LT'	0.544 2
'MSTORE'	0.500 4
'STOP'	0.586 0

2.2 KOLSTM 模型

智能合约操作码的本质是基于堆栈操作指令的集合，操作码的前后顺序即序列特征，对蜜罐陷阱合约检测至关重要。LSTM 模型通过引入门结构克服长期依赖问题，在处理序列类型的输入数据时具有优越的性能。关键操作码和序列特征是蜜罐陷阱合约的重要特性，直接影响蜜罐陷阱合约检测的准确性。同时，关键操作码权重机制能够有效区分操作码的重要程度，基于此，本文构建了能够同时捕获关键操作码与序列特征的 KOLSTM 模型。

KOLSTM 模型的底层是合约操作码输入层，该层将合约中的操作码转换为相同映射表示的数字特征；合约操作码输入层之上的 Key-Opcode 权重层是关键操作码权重机制实现的重要部分，该层将 Key-Opcode 提取方法得到的结果应用到模型中，权重值的大小代表操作码的重要性，该层可以提高模型对关键操作码信息的关注度；词向量嵌入层使用基于海量智能合约操作码预训练的 Word2Vec[21]词向量模型，将操作码序列映射为高维特征向量表示；嵌入层之上的 LSTM 单元可以捕获合约中字节码级别潜在的序列信息特征，并且克服长期依赖问题；

Softmax 层用于分类模型结果输出前的归一化处理；最顶层的输出层输出模型的分类结果，通常以数字表示。KOLSTM 模型的网络结构如图 2 所示。

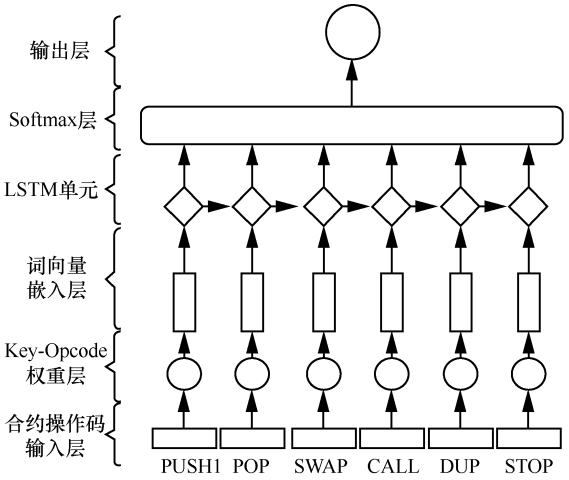

图 2　KOLSTM 模型的网络结构

2.2.1　Key-Opcode 权重层

Key-Opcode 权重层将关键操作码提取方法得到的权重字典应用到模型中，在进行词向量嵌入转换之前，通过该层提高检测模型对关键操作码特征的关注度。直观来看，Key-Opcode 权重层是对词向量嵌入层的改进。无论是独热编码还是无监督训练得到的词向量模型，对词汇的向量表示都是基于词语间的相关性，这样的模式对基于文本的词汇表示是十分合适的，但对于智能合约中的操作码却不太适用，其无法凸显关键操作码对智能合约的特殊性。因此，本文在词向量嵌入层之前加入了 Key-Opcode 权重层，可以表示为

$$y_o = W_o A_o, A_o = \left[A_1, A_2, \cdots, A_n\right]^{\mathrm{T}} \tag{2}$$

其中，W_o 为操作码 o 通过 Key-Opcode 提取方法得到的权重值，A_o 为操作码 o 的词向量表示矩阵。Key-Opcode 权重层的实质是关键操作码权重机制，通过该权重机制捕获对蜜罐陷阱合约检测有重要作用的关键操作码特征。

同时，在该层保留传统 LSTM 模型的权值共享机制，关键操作码权重也仅在输入层与词向量嵌入转换之间应用，这样的设置可以极大地减少模型参数量，降低训练时间。

2.2.2　词向量嵌入层

词向量嵌入层将输入的操作码信息转换为用以模型识别训练的高维特征向量。本文在该层引入了 Word2Vec 词向量模型进行特征向量转换，通过训练海量的文本信息在双层的神经网络中进行非监督学习，将每个词汇转换为高维向量。

Word2Vec 模型的训练过程采用的是 N-gram 模式，通过这种模式训练出的特征向量包含了一定的文本上下文信息。考虑到智能合约的特殊性，传统的英文预训练 Word2Vec 模型无法有效地表征智能合约中隐含的操作码特征信息，本文使用的 Word2Vec 模型是基于海量的智能合约数据预训练得到的。

2.2.3　LSTM 单元

针对蜜罐陷阱合约数据长度过长、操作码上下文关联性较强的特点，本文在模型中引入了 LSTM 单元进行合约数据处理。

LSTM 单元如图 3 所示，通过引入遗忘门、更新门、输出门结构以及 sigmoid、tanh 激活函数来控制信息的交互，维持和控制单元状态。本文设 t 时刻该单元输入的信息流为 $x^{<t>}$，细胞状态为 $c^{<t>}$，隐藏状态为 $a^{<t>}$，激活函数为 sigmoid 和 tanh，权重矩阵和偏置向量分别为 $\boldsymbol{W}$ 和 $\boldsymbol{b}$。

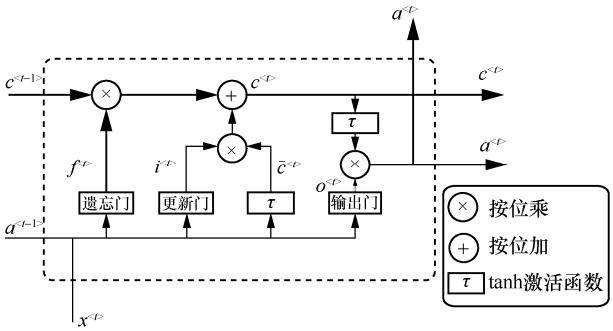

图 3　LSTM 单元

输入信息进入 LSTM 单元首先需要经过遗忘门，遗忘门可以对信息进行选择性遗忘，也就是对 t−1 时刻的隐藏状态 $a^{<t-1>}$ 和 t 时刻的输入信息 $x^{<t>}$ 进行筛选删除，保留重要信息。首先该门会读取 $a^{<t-1>}$ 和 $x^{<t>}$，然后经过 sigmoid 激活函数，遗忘门输出一个[0,1]范围的数值，输出 1 代表完全保留，输出 0 代表完全遗忘，可以表示为

$$f^{<t>} = \mathrm{sigmoid}(\boldsymbol{W}_f[a^{<t-1>}, x^{<t>}] + \boldsymbol{b}_f) \tag{3}$$

在进行遗忘之后，下一个阶段需要确定细胞状态中保存的信息内容，执行这一操作的是更新门以及其他激活函数。具体来说，首先更新门会通过 sigmoid 激活函数确定更新的内容，这时需要创建一个新的数字向量

$\overline{\boldsymbol{c}}$，用来保存通过 tanh 激活函数得到的新的信息。经过这两步可以得到用于细胞状态更新的准备信息。

$$i^{<t>}=\text{sigmoid}(\boldsymbol{W}_i[a^{<t-1>},x^{<t>}]+\boldsymbol{b}_i) \tag{4}$$

$$\overline{\boldsymbol{c}}^{<t>}=\tanh(\boldsymbol{W}_c[a^{<t-1>},x^{<t>}]+\boldsymbol{b}_c) \tag{5}$$

然后将准备信息以及遗忘门获取的信息进行拼接整合，得到新的细胞状态。

$$\boldsymbol{c}^{<t>}=f^{<t>}\boldsymbol{c}^{<t-1>}+i^{<t>}\overline{\boldsymbol{c}}^{<t>} \tag{6}$$

最后确定当前状态输出的信息。首先使用 sigmoid 激活函数确定细胞状态的输出部分，然后细胞状态通过 tanh 函数进行放缩之后再与 sigmoid 激活函数的输出结果相乘。这样便可以输出当前状态部分。

$$o^{<t>}=\text{sigmoid}(\boldsymbol{W}_o[a^{<t-1>},x^{<t>}]+\boldsymbol{b}_o) \tag{7}$$

$$a^{<t>}=o^{<t>}\tanh(\boldsymbol{c}^{<t>}) \tag{8}$$

3 实验

3.1 实验步骤

实验流程共分为 6 个步骤，包括数据获取、数据预处理、数据集增强、特征提取、模型训练及效果评估。在数据获取步骤中，基于以太坊平台获取实验所需的智能合约数据；通过数据预处理步骤将合约数据的源码文件反编译为模型训练所需的操作码格式；在数据集增强步骤中，针对训练数据存在的问题进行针对性的数据增强；特征提取是模型训练前的最后一个步骤，在这个步骤中将预处理后的操作码数据进行分词以及特征向量转换，获取可用于模型训练的有效特征；之后将在模型训练步骤中，训练本文提出的 KOLSTM 模型以及其他对比模型；最后，通过效果评估步骤对实验结果进行性能评价和分析。

3.2 数据获取及预处理

本文实验采用的蜜罐陷阱合约数据集[3]包含 857 个经过验证的蜜罐陷阱合约文件，并根据它们的特性划分为了 8 个不同的类别，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）、平衡障碍（BD, balance disorder）、隐藏转移（HT, hidden transfer）、跳空字符串（SESL, skip empty string literal）和类型溢出（TDO, type deduction overflow）。而非蜜罐陷阱合约数据是通过以太坊浏览器得到的 5 960 份具有唯一精准字节码匹配的合约数据。

原始合约数据无法直接用以模型训练，需要将数据集中的源码文件反编译为操作码格式。本文使用了以太坊虚拟机（EVM, Ethereum virtual machine）数据包来完成这一转换，图 4 展示了反编译后的操作码文件，可以看到其中共包含 2 种类型的数据信息，分别是操作码和指令地址，对于模型的训练仅需要保留操作码。

```
EQ
PUSH2 0x007a
JUMPI
DUP1
PUSH4 0xe5ed1d59
EQ
PUSH2 0x00a3
JUMPI
DUP1
PUSH4 0xe8147a25
```

图 4 反编译后的操作码文件

3.3 数据集增强及特征提取

对数据集进行检查时，本文发现蜜罐陷阱合约数据存在严重的合约复制[19]现象，而且由于时间问题部分合约已经自毁。在去除了重复合约以及自毁合约后，得到的蜜罐陷阱合约类别及数量如表 2 所示。

表 2 蜜罐陷阱合约类别及数量

类别	数量/个
HSU	160
BD	17
HT	12
ID	40
SESL	5
SMC	34
TDO	4
US	11

通过观察可得，蜜罐陷阱合约数据存在严重的类别不平衡以及数据量过少的问题，这对深度学习模型的训练是极其不利的。为了解决这一问题，本文使用了 EDA 数据增强过采样技术[22]来扩充原有数据。为了保持合约数据的序列特征和上下文相关性，本文在进行数据增强时并没有打乱操作码的前后顺序。最终，共得到了 4 650 份蜜罐陷阱合约以及 5 960 份非蜜罐陷阱合约操作码。数据集的合约长度统计情况如图 5 所示，其中合约长度指的是合约中所含操作码数量。

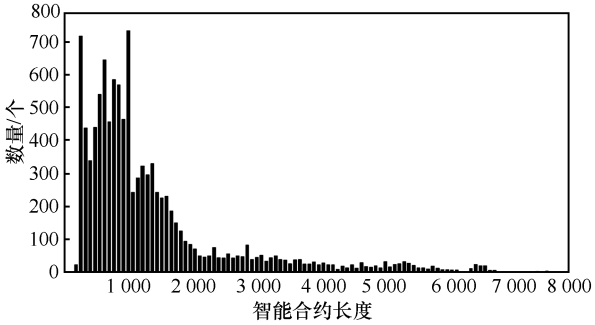

图 5　数据集的合约长度统计情况

3.4　实验参数设置

本文在实验中将数据集按照 6:2:2 的比例划分为训练集、测试集和验证集，其中训练集主要用于模型训练；验证集用于验证及优化模型参数，例如学习率或训练轮数等，并用来判断模型是否过拟合；测试集用来测试模型最终的性能及泛化能力。为了确定模型的最佳参数，本文在实验中使用了交叉验证[23]的方法。多次实验后得到的模型最佳参数设置如表 3 所示。

表 3　　模型最佳参数设置

参数	数值
学习率	0.001
Batch 大小	32
训练轮数/轮	20
词向量维度	250
隐藏层维度	150
Dropoutrate	0.5

3.5　实验结果分析

3.5.1　训练轮数分析

训练轮数的设置会影响模型的拟合情况，为了寻找 KOLSTM 模型的最佳训练轮数，本文分别比较了验证集和训练集在不同训练轮数下的模型损失值和准确率，实验结果分别如图 6 和图 7 所示。

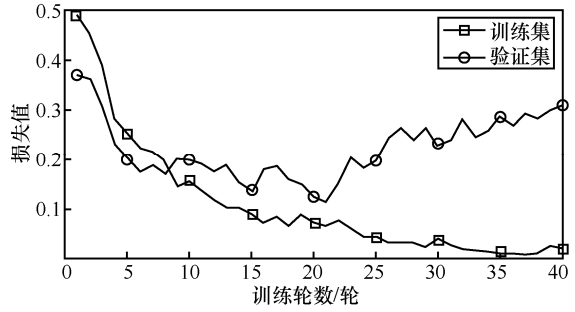

图 6　验证集和训练集在不同训练轮数下的模型损失值

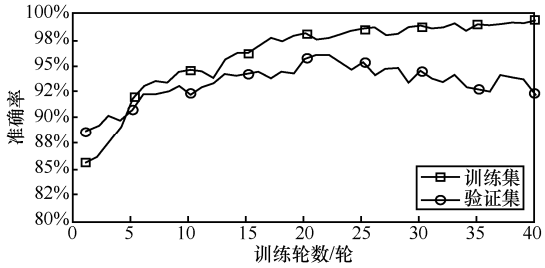

图 7　验证集和训练集在不同训练轮数下的模型准确率

由图 6 和图 7 可以看出，在训练轮数超过 20 轮以后，训练集损失值仍在下降，而验证集损失值虽然存在数据波动情况，但整体呈上升趋势；训练集准确率仍在上升，而验证集准确率呈下降趋势。这说明训练轮数超过 20 轮以后，模型会出现过拟合情况，因此本文实验将训练轮数设置为 20。

3.5.2　词向量分析

为了探究词向量对 KOLSTM 模型检测性能的影响，本文比较了不同类型预训练词向量下的检测性能。预训练词向量包括基于合约操作码预训练词向量和维基百科词向量，在相同维度（250 维）下的实验结果如表 4 所示。由表 4 可以看出，相较维基百科词向量，基于合约操作码预训练词向量在检测性能上有所提升。

表 4　不同类型词向量在相同维度（250 维）下的实验结果

词向量类型	精确率	召回率	F 值
基于合约操作码预训练词向量	96.91%	94.30%	95.59%
维基百科词向量	94.60%	92.37%	93.47%

本节实验进一步对比了不同维度下模型的收敛速度，词向量维度包括 100 维和 250 维。实验中都使用了基于合约操作码预训练词向量，结果如图 8 所示。由图 8 可以看出，相比于低维度的词向量模型，高维度词向量模型表征能力更强，具有更快的收敛速度。

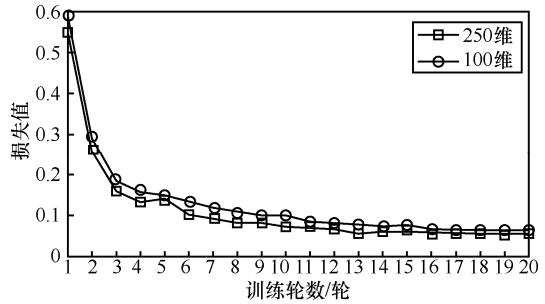

图 8　100 维和 250 维词向量对损失值影响

3.5.3　输入句长分析

为了验证 KOLSTM 模型的检测性能，将本文

提出的方法与卷积神经网络（CNN, convolutional neural network）、循环神经网络（RNN, recurrent neural network）以及 LSTM 模型进行了对比实验。为了避免实验中所用模型出现过拟合或欠拟合问题，所有模型都加入了 Dropout 机制[24]，且设置了相同的训练轮数。

本节实验针对输入句长对模型检测性能的影响进行分析，分别比较了不同输入句长下 4 类模型的检测性能，输入句长指的是模型设置的最大句长输入。实验结果如图 9 所示。

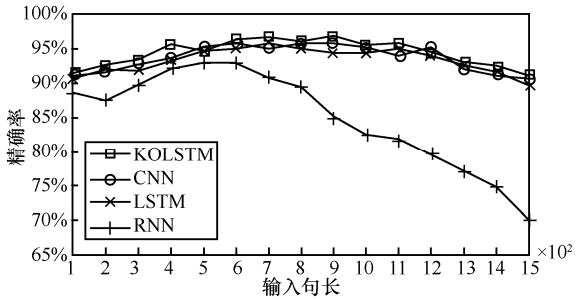

图 9　输入句长对不同模型的性能影响

由图 9 可以看出，在输入句长达到 600 以后，RNN 模型的精确率呈下降趋势，而其他 3 类模型的检测精确率相对稳定，都维持在 90%以上。本文分析认为 RNN 模型由于自身循环输入的结构限制，在处理较长序列输入时会出现梯度消失或爆炸问题，导致精确率的骤降，而 LSTM 模型以及 CNN 模型通过引入门结构或者卷积操作克服了长期依赖问题。KOLSTM 模型因为加入了权重机制，精确率相比 CNN 模型与 LSTM 模型都有所提升。当输入句长为 900 时，KOLSTM 模型的检测性能达到最佳。因此本实验将输入句长设置为 900。

3.5.4　收敛速度分析

为了分析模型的收敛速度，本文对比了 4 类模型训练时的损失值下降速率，结果如图 10 所示。

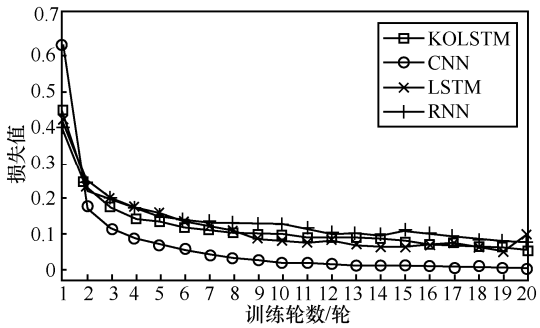

图 10　不同模型的损失值变化

由图 10 可以看出，KOLSTM 模型的收敛速度要比 LSTM 模型与 RNN 模型略快一些，但要慢于 CNN 模型。本文分析认为，KOLSTM 模型中的 Key-Opcode 权重机制其实可以看作预训练机制，通过这一机制可以提前告诉模型哪些是关键操作码，相较于 LSTM 模型与 RNN 模型，有效加速了模型收敛。而 CNN 模型特有的卷积操作可以直接捕获关键操作码特征，模型收敛速度要快于 KOLSTM 模型。

3.5.5　检测性能分析

进一步通过实验分析本文所提模型的性能，并与其他模型进行对比，实验结果如表 5 所示。

表 5　不同模型的实验结果

模型	精确率	召回率	F 值
RNN	79.65%	98.62%	88.25%
LSTM	94.60%	92.37%	93.47%
CNN	95.75%	90.69%	93.15%
LightGBM	97.09%	89.62%	93.20%
KOLSTM	96.91%	94.30%	95.59%

由表 5 可以看出，相较于其他深度学习模型，KOLSTM 模型具有最佳的检测性能。RNN 模型虽然具有极高的召回率，但精确率偏低，仅达到了 79.65%。相较于 RNN 模型，LSTM 模型的检测性能有了较大的提升，精确率达到了 94.60%。CNN 模型用于蜜罐陷阱合约检测也取得了较好的效果，精确率达到了 95.75%。LightGBM 模型为 Chen 等[5]在蜜罐陷阱合约检测分类实验中所用的模型，该模型具有较高的检测精确率，但召回率仅为 89.62%。相较于上述 4 种模型，KOLSTM 模型的检测性能有明显提升，精确率达到了 96.91%，F 值达到了 95.59%。

3.5.6　多分类实验结果分析

为了验证 KOLSTM 模型在多分类场景下的检测性能，将其与 LightGBM 模型[5]进行对比，实验结果如表 6 所示。多分类实验指的是对蜜罐陷阱合约进行类别判断的实验。由表 6 可以看出，LightGBM 模型对于某些类别的分类出现了明显的过拟合和欠拟合现象，平均精确率与平均 F 值仅达到 74.71%与 71.47%。而 KOLSTM 模型在蜜罐陷阱合约类型识别工作上同样具有不错的表现，类别平均分类精确率达到 89.34%，平均 F 值达到 91.01%。但同时也可以注意到，KOLSTM 模型对于某些类别的分类精确率相对偏低，比如 ID 类别的分类精确

表 6　　多分类实验结果

类别	KOLSTM			LightGBM		
	精确率	召回率	F 值	精确率	召回率	F 值
US	88.17%	91.98%	90.03%	100%	90.55%	95.04%
BD	92.42%	89.17%	91.04%	100%	100%	100%
HSU	88.71%	82.09%	85.27%	100%	92.57%	96.14%
HT	91.91%	100%	95.79%	100%	91.07%	95.33%
ID	73.68%	80.77%	77.06%	100%	82.39%	90.34%
SESL	94.00%	100%	96.91%	0	88.89%	0
SMC	87.00%	98.86%	92.55%	97.73%	92.21%	94.89%
TDO	98.86%	100%	99.43%	0	0	0
平均值	89.34%	92.86%	91.01%	74.71%	79.71%	71.47%

率仅达到 73.68%。本文分析认为可能是由于该类别原始合约数据过少，在进行数据扩充和模型训练时无法有效捕获类别特征，导致该类别的分类结果不佳。

4　结束语

本文针对基于区块链的蜜罐陷阱合约检测识别进行研究，通过分析蜜罐陷阱合约的特点，提出了 KOLSTM 深度学习模型。该模型通过 Key-Opcode 提取方法引入权重机制，可以有效捕获关键操作码特征以及序列特征。实验结果表明，本文所提模型的检测性能要优于传统检测模型。本文计划下一步将更多的智能合约特征引入模型训练，包括交易记录、历史信息等，希望以此来进一步提高模型的检测性能。

由于匿名和去中心化特性，区块链的应用将更加广泛，但基于区块链以及数字货币的恶意陷阱也正变得越加猖獗。在未来的工作中，将持续追踪和研究基于区块链的其他类型的漏洞和陷阱攻击，以维护区块链生态健康。

参考文献：

[1] ZHENG Z B, XIE S A, DAI H N, et al. An overview of blockchain technology: architecture, consensus, and future trends[C]//Proceedings of 2017 IEEE International Congress on Big Data (BigData Congress). Piscataway: IEEE Press, 2017: 557-564.

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

[3] TORRES C F, STEICHEN M. The art of the scam: demystifying honeypots in Ethereum smart contracts[C]//Proceedings of 28th USENIX Security Symposium (USENIX Security 19). Berkeley: USENIX Association, 2019: 1591-1607.

[4] 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. Piscataway: IEEE Press, 2020: 1-9.

[5] 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.

[6] 曾诗钦, 霍如, 黄韬, 等. 区块链技术研究综述: 原理、进展与应用[J]. 通信学报, 2020, 41(1): 134-151.
ZENG S Q, HUO R, HUANG T, et al. Survey of blockchain: principle, progress and application[J]. Journal on Communications, 2020, 41(1): 134-151.

[7] CONTI M, SANDEEP KUMAR E, LAL C, et al. A survey on security and privacy issues of bitcoin[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 3416-3452.

[8] VASEK M, MOORE T. There's no free lunch, even using bitcoin: tracking the popularity and profits of virtual currency scams[C]//International Conference on Financial Cryptography and Data Security. Berlin: Springer, 2015: 44-61.

[9] 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.

[10] 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.

[11] CHEN W L, ZHENG Z B, CUI J H, et al. Detecting ponzi schemes on Ethereum: towards healthier blockchain technology[C]//Proceedings of the 2018 World Wide Web Conference on World Wide Web. New York: ACM Press, 2018: 409-1418.

[12] 张艳梅, 楼胤成. 基于深度神经网络的庞氏骗局合约检测方法[J]. 计算机科学, 2021, 48(1): 273-279.
ZHANG Y M, LOU Y C. Deep neural network based ponzi scheme contract detection method[J]. Computer Science, 2021, 48(1): 273-279.

[13] 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.

[14] TIKHOMIROV S, VOSKRESENSKAYA E, IVANITSKIY I, et al. SmartCheck: static analysis of Ethereum smart contracts[C]// Pro-

ceedings of 2018 IEEE/ACM 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB). Piscataway: IEEE Press, 2018: 9-16.

[15] LIU C, LIU H, CAO Z, et al. ReGuard: finding reentrancy bugs in smart contracts[C]//Proceedings of 2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion). Piscataway: IEEE Press, 2018: 65-68.

[16] JIANG B, LIU Y, CHAN W K. ContractFuzzer: fuzzing smart contracts for vulnerability detection[C]//Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. New York: ACM Press, 2018: 259-269.

[17] ZHANG Y, JIN R, ZHOU Z H. Understanding bag-of-words model: a statistical framework[J]. International Journal of Machine Learning and Cybernetics, 2010, 1(1/2/3/4): 43-52.

[18] KE G, MENG Q, FINLEY T, et al. LightGBM: a highly efficient gradient boosting decision tree[J]. Advances in Neural Information Processing Systems, 2017, 30: 3146-3154.

[19] LIU H, YANG Z Q, LIU C, et al. EClone: detect semantic clones in Ethereum via symbolic transaction sketch[C]//Proceedings of ESEC/FSE 2018: Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. New York: ACM Press, 2018: 900-903.

[20] 施聪莺, 徐朝军, 杨晓江. TFIDF 算法研究综述[J]. 计算机应用, 2009, 29(S1): 167-170, 180.
SHI C Y, XU C J, YANG X J. Study of TFIDF algorithm[J]. Journal of Computer Applications, 2009, 29(S1): 167-170, 180.

[21] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. arXiv Preprint, arXiv: 1301.3781, 2013.

[22] 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). Boston: Association for Computational Linguistics, 2019: 6382-6388.

[23] ARLOT S, CELISSE A. A survey of cross-validation procedures for model selection[J]. Statistics Surveys, 2010, 4: 40-79.

[24] GAL Y, GHAHRAMANI Z. A theoretically grounded application of dropout in recurrent neural networks[J]. arXiv Preprint, arXiv: 1512.05287, 2015.

[作者简介]

张红霞（1981− ），女，山东东营人，博士，中国石油大学（华东）副教授、硕士生导师，主要研究方向为边缘计算、区块链技术、服务计算等。

王琪（1997− ），男，山东枣庄人，中国石油大学（华东）硕士生，主要研究方向为区块链技术、数据挖掘。

王登岳（1996− ），男，山东聊城人，中国石油大学（华东）硕士生，主要研究方向为网络与服务计算。

王奔（1997− ），男，山东临沂人，中国石油大学（华东）硕士生，主要研究方向为计算机视觉、多目标跟踪。