·自动化技术·

文章编号：1003－1251(2026)01－0001－07

基于特征融合的恶意代码同源性检测模型

赵运弢，白皓成

（沈阳理工大学 信息科学与工程学院，沈阳 110159）

摘 要：为解决当前恶意代码同源性分析普遍采用单一特征而导致样本信息表示不足、模型分类准确率较低的问题，提出一种基于特征融合的恶意代码同源性检测模型，该模型将卷积神经网络(CNN)、双向门控循环单元(BiGRU)与引入位置编码和卷积层的多头自注意力(positional encoding convolution multi-head self-attention，PC-MSA)机制相结合，有效增强了对序列数据长程依赖关系的建模能力。首先，利用正则表达式从恶意代码反汇编文件中提取应用程序接口(API)序列和操作码(Opcode)序列；然后，构建文档向量化模型，将提取的序列转换为特征向量并进行特征融合，融合后的序列可更好地体现语义行为与底层逻辑；最后，将融合后的特征向量输入本文所建模型(CNN-BiGRU-PC-MSA)，进行家族同源性检测。实验结果表明，本文模型的同源性检测准确率可达到98%，证明了特征融合方法及模型的有效性。

关 键 词：特征融合；CNN-BiGRU模型；多头自注意力机制；同源性检测

中图分类号：TP309 **文献标志码**：A **DOI**：10.3969/j.issn.1003－1251.2026.01.001

Homology Detection Model of Malicious Code Based on Feature Fusion

ZHAO Yuntao，BAI Haocheng

（Shenyang Ligong University，Shenyang 110159，China）

Abstract：To address the issue that the current malicious code homology analysis generally uses a single feature，which leads to insufficient sample information representation and low accuracy of model classification，a homology detection model of malicious code based on feature fusion is proposed. The model combines convolutional neural networks(CNN)，bidirectional gated recurrent units(BiGRU) and positional encoding convolution multi-head self-attention(PC-MSA)，effectively enhancing the ability to model long-range dependencies of sequence data. Firstly，the API sequence and Opcode sequence are extracted from the malware disassembly file using regular expressions. Then，a document vectorization model is built to convert the extracted sequences into feature vectors and perform feature fusion. The fused sequence can better reflect the semantic behavior and underlying logic. Finally，the fused feature vector is input into the model constructed in this paper(CNN-BiGRU-PC-MSA) for family homology detection. Experimental results show that the accuracy of the proposed homology detection model can reach 98%，proving the effectiveness of the feature fusion method and model.

Key words：feature fusion；CNN-BiGRU model；multi-head self-attention mechanism；homology detection

收稿日期：2025－05－16

基金项目：辽宁省应用基础研究计划项目(2023JH2/101300203)

作者简介：赵运弢(1978—)，男，教授，博士。

随着全球数字化转型的加速推进,网络恶意攻击的目标逐渐从个人用户扩展到政府、企业以及关键基础设施,对医疗、能源、金融等关键行业造成了严重影响,给全球网络安全带来了巨大的挑战[1]。同时,恶意代码的数量和复杂性呈指数级增长[2],尤其是特洛伊木马、网络蠕虫、间谍软件、勒索软件和挖矿软件等更为突出[3]。《2024年上半年度网络安全态势研判分析报告》[4]指出,半年内针对网络的攻击次数达到1 397.1亿次,深信服拦截恶意程序153.16亿次,其中木马远控占比30.43%、挖矿占比24.73%、僵尸网络占比14.01%、蠕虫占比11.75%,总体态势严峻。

恶意代码的同源性分析[5]旨在通过识别恶意代码之间相似代码片段、结构以及上下文语义关联[6],确定新出现的恶意代码是否属于特定的恶意代码家族[7],从而加速恶意代码的分类和检测。恶意代码逆向分析[8]通过将恶意代码从二进制文件还原为汇编代码,发现其真实逻辑和恶意行为。借助逆向汇编技术,研究人员能够深入解析二进制代码的底层指令和控制结构,更加准确地识别不同恶意代码样本之间的相似性,有效提高同源性分析的准确性。

从第一个病毒AppleII诞生之日起,恶意代码同源性分析一直是网络空间安全领域的研究热点。Kakisim等[9]提出了一种针对恶意代码变种的同源性分析方法,为每个恶意代码样本构建co-opcode图,从图中提取特定的操作码模式,通过向量的形式表示编码结果。Laurenza等[10]提出了一种静态分析方法用于恶意代码的特征提取,利用机器学习算法对恶意代码进行同源性分析。Zhang等[11]针对现有恶意代码检测方法中应用程序接口(API)序列缺乏语义信息建模以及长序列数据检测性能较低的问题,提出了一种融合API序列语义关系的恶意代码检测模型,该模型通过向量表示方法(API2Vec)对API函数进行降维表示,结合双向长短期记忆(BiLSTM)网络捕获序列特征,提升了对恶意代码检测的精度和鲁棒性。刘紫煊等[12]为解决传统检测模型对复杂特征提取不足的问题,提出了一种基于多特征融合与BiLSTM的分类方法,通过对融合特征进行深层次学习,取得了良好的检测效果,检测精度显著高于随机森林及其他传统分类模型。

本文针对逆向汇编后的恶意代码文件进行家族同源性检测,主要研究工作如下。

1)针对恶意代码样本单一特征信息表示不足而导致模型分类准确率较低的问题,提出一种向量化特征融合方法,构建Doc2Vec模型,将API序列和操作码(Opcode)序列转换为特征向量后,在向量空间中进行融合。向量化特征融合能够更好地捕捉特征之间的关联,并将高维的稀疏词汇表示映射到低维的稠密向量空间,提高模型分类准确率和效率。

2)针对特征融合后的长序列结构复杂及顺序依赖问题,设计一种基于多个自注意力层[13]、位置编码和卷积层的多头自注意力(positional encoding convolution multi-head self-attention, PC-MSA)机制,以更全面地理解输入信息的结构和内容,更有效地捕捉代码序列中的重要特征。

3)建立基于卷积神经网络(CNN)和双向门控循环单元(BiGRU),并融合PC-MSA的恶意代码同源性检测模型(CNN-BiGRU-PC-MSA),通过卷积层提取局部特征,利用多层BiGRU捕捉序列的双向依赖关系,通过PC-MSA自适应提取关键特征,生成高效且可解释的上下文特征向量。

1 特征向量化及特征融合

1.1 特征提取

1.1.1 API序列提取

API序列构成了应用程序与底层操作系统之间的重要接口,在描述软件的特定意图或行为方面具有独特的作用。提取API序列的静态特征时,首先分析恶意代码或可疑软件的二进制文件,对其进行反汇编,再将识别到的API序列按其调用顺序提取出来,这样可以捕捉程序的完整执行路径和不同调用频率,揭示复杂的行为模式和依赖关系。API序列的调用频率、前后顺序以及某些敏感函数的调用均可作为家族同源性分析的依据。

本文采用正则表达式进行API序列提取,正则表达式是一种用于匹配文本模式的强大工具,是提取API序列的一种快速、有效的方式,可自动从反汇编文件中提取出API序列。本文API序列提取算法如表1所示,API序列提取流程如图1所示。

由于一些恶意代码采用了加密、加壳、混淆等抗检测技术,导致部分样本的API序列难以被直接提取和分析。因此,为了更有效地进行恶意代码的检测与分类,需要将API序列与其他特征结合使用,以增强检测的全面性和准确性。

表 1　API 序列提取算法

Table 1　API sequence extraction algorithm

输入:.asm 文件
输出:API 序列 API_call

```
1. open .asm file
2. for file in os.listdir //遍历所有 .asm 文件
3. pattern = r"\s + call\s + ds:(? P < api_name > [a-zA-Z0-9] + )"//定义正则表达式
4. os.path.join(folder_path,filename)//读取 .asm 文件
5. re.findall(pattern,data)//查找 API 序列
6. os.path.splitext(filename)[0]//将结果写入输出文件
7. return file_id
```

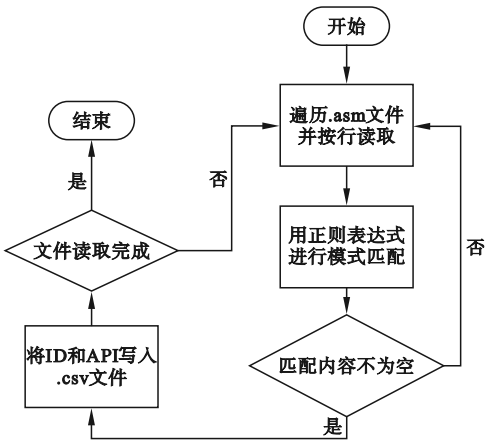

图 1　API 序列提取流程

Fig. 1　API sequence extraction process

1.1.2　Opcode 序列提取

Opcode 是指令的核心,是最底层的执行单元,代表了处理器执行的具体操作。因此,对恶意代码可执行文件[14]中的 Opcode 序列进行静态分析,即通过分析 Opcode 序列的分布和频率,可以快速识别出恶意代码常用的模式及行为特征,揭示其底层执行逻辑,发现其恶意行为。本文将 Opcode 序列作为第二个需要提取的序列,用于恶意代码的分类任务。

通过正则表达式可精准提取每行代码中的 Opcode 序列,避免提取其他不相关的部分,如地址、机器码或注释等。通常只对 .text 节进行处理,因其为程序代码的主要部分,某些 Opcode 序列(如 align)用于对齐内存,无法反映程序的功能行为,可以过滤掉。本文基于正则表达式的 Opcode 序列提取算法如表 2 所示。

表 2　Opcode 序列提取算法

Table 2　Opcode sequence extraction algorithm

输入:.asm 文件
输出:Opcode 序列 op_fe

```
1. open .asm file
2. for line in file //按行读取文件
3. pattern = re.compile(r'\s([a-fA-F0-9]{2}\s) + \s * ([a-z] + )')//定义正则表达式
4. if line.startwith(".text")//是否以".text"开头
5. if 找到匹配内容,操作码 opcode = s[0][1]
6. if opcode 不为"align"
7. opcode = [opc for opc in opcode if opc top_opcodes][: max_opcodes]
8. opcode_fe//添加一个操作码
```

1.2　向量化特征

本文通过构建 Doc2Vec 模型对 API 序列和 Opcode 序列进行向量化表示。Doc2Vec 是一种基于深度学习的文本表示方法,是 Word2Vec 的扩展,其能够为整个文档生成稠密的向量表示。Doc2Vec通过将文档的上下文信息与每个词的向量表示结合起来,能够捕捉到文档级别的语义特征,为机器学习模型提供高质量的输入特征。

Doc2Vec 的主要优势在于对文本上下文信息的良好捕捉能力,适用于恶意代码的检测场景。在恶意代码分析中,文本数据往往包含大量上下文信息,Doc2Vec 能够通过学习文本的语义关联,将相似的文本映射到相似的向量空间中,这不仅有助于恶意代码家族的分类,还能提高模型在应对复杂语义模式和变种代码时的泛化能力。与传统的词袋模型相比,Doc2Vec 生成的向量表示更加紧凑且包含更多语义信息,可显著提升模型的检测效果。

基于 Doc2Vec 的 API 语义表征示意图如图 2 所示。图中:FindWindowA(用于查找已存在窗口)、VirtualAlloc(用于内存分配)和 GetProcAddress(用于解析地址)分别为本文数据集中某条恶意代码样本的前三个 API 序列,预测目标词 HeapDestroy(用于销毁进程中创建的堆对象)为第四个 API 序列。

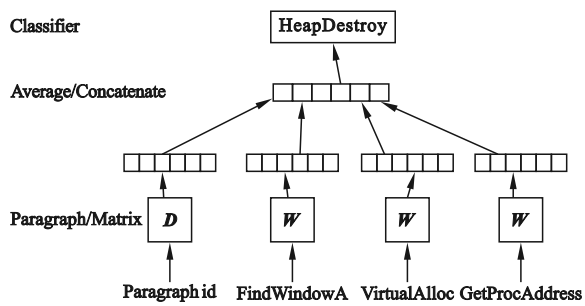

图 2　基于 Doc2Vec 的 API 语义表征示意图

Fig. 2　Schematic diagram of API semantic representation based on Doc2Vec

本文利用 Doc2Vec 模型通过构建段落向量(Paragraph/Matrix,图中用矩阵 ***D*** 表示)与词向量(图中用矩阵 ***W*** 表示)联合建模,实现对 API 序列的深层语义表示。段落向量是代表整个 API 序列

的唯一标识,是 Doc2Vec 特有的结构,用于捕捉整段调用序列的语义特征;各词向量分别对应于具体的 API 序列,表示局部功能语义。在训练过程中,模型将段落向量与上下文的词向量通过拼接(Concatenate)或平均(Average)的方式融合,形成一个表示上下文整体语义的向量,通过分类器(Classifier)预测下一条可能出现的 API(如 Heap-Destroy)概率。通过最小化预测输出与真实目标词之间的交叉熵损失,模型不断优化段落向量与词向量,从而提升对整体序列语义的表示能力。

本文利用 Doc2Vec 模型对每条恶意代码的 API 序列和 Opcode 序列分别建模,获得其语义向量表示,通过对两类特征向量进行融合,进一步提升恶意代码样本的特征表达能力,为后续同源性检测模型提供更加丰富和准确的输入表示。

1.3 特征融合

依靠单一维度的信息进行恶意代码样本检测和分类存在一定局限性,可能会因为忽略恶意代码样本间的关联而影响分类效果。API 序列代表恶意代码样本的行为特征,体现了恶意代码样本与操作系统的交互,而 Opcode 序列代表恶意代码样本的指令级特征,揭示了其底层执行逻辑,这些特征在一定程度上代表了恶意代码的核心行为。通过融合 API 序列与 Opcode 序列特征,可以充分整合多维信息,形成更全面的特征表达,显著提高恶意代码检测与家族分类的效果。

特征向量的融合过程为:首先分别采用 API 序列与 Opcode 序列训练独立的 Doc2Vec 模型,获取各自的语义嵌入空间;然后利用训练好的 Doc2Vec 模型将每个 API 序列与 Opcode 序列分别映射到对应的向量空间中,获得 API 向量与 Opcode 向量;最后通过拼接的方式对两个特征向量进行融合。

相较于传统直接序列拼接的融合方法,基于 Doc2Vec 的向量化特征融合方法通过捕捉文本序列上下文语义信息,显著提升了嵌入特征的语义表达能力,能够挖掘序列内部隐含的潜在语义关联,并且有效保留了序列的结构与上下文信息。通过向量化融合策略获得的特征向量长度可控且相对较短,可有效避免维度灾难问题,提升后续深度学习模型的训练效率及模型的分类准确率。

2 恶意代码同源性检测模型

2.1 多头自注意力机制

针对 API 序列和 Opcode 序列融合后得到的长序列结构复杂及顺序依赖问题,本文设计了一种引入位置编码和卷积层的多头自注意力机制 PC-MSA。PC-MSA 结构如图 3 所示,图中 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$ 分别表示查询(Query)向量、键(Key)向量和值(Value)向量。

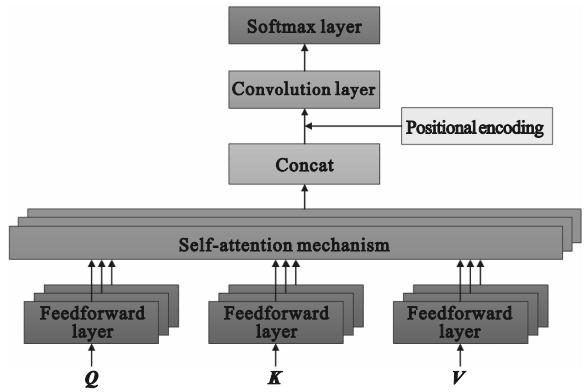

图 3 PC-MSA 结构图

Fig. 3 Structure diagram of PC-MSA

首先,通过前馈层(Feedforward layer)捕捉全局的非线性关系,学习初步的注意力权重;其次,通过多头自注意力机制(Self-attention mechanism)在多个不同的子空间中并行学习输入数据的不同依赖关系,捕捉到更多样的全局特征;在自注意力层之后进行拼接(Concat)处理,将各注意力头输出的独立向量合并为一个整体向量,从而增强模型的表达能力;引入位置编码(Positional encoding)可帮助模型理解序列数据中的时序信息,有效处理顺序依赖问题;通过卷积层(Convolution layer)的使用,模型能够在注意力机制后进一步提取局部特征,增强对局部特征的学习能力;最后,通过 Softmax layer 得到归一化的注意力权重,以反映序列中所有位置的信息,根据注意力权重大小进行过滤和融合,得到上下文特征向量。

2.2 双向门控循环单元

门控循环单元(GRU)是循环神经网络(RNN)的一种变体,通过引入门控机制来解决 RNN 中不能长期记忆和梯度消失的问题,与长短时记忆网络(LSTM)相比,GRU 的结构更加简单、参数更少、更容易计算,同时能够有效捕捉长程依赖关系。

GRU 的门控机制包括两种:更新门和重置门。更新门决定当前状态从前一状态继承信息的多少;重置门控制如何结合新输入与前一隐藏状态,用于决定是否丢弃以前的信息,候选状态是基于重置门计算得到的用于更新当前隐藏状态的新状态。这两个门决定哪些信息作为最终的输出,

同时不会随时间推移而清除不相关的信息,因此可以有效避免 RNN 的梯度消失和梯度爆炸问题。更新门的输出 $\boldsymbol{Z}_t$、重置门的输出 $\boldsymbol{R}_t$、新的隐藏状态 $\boldsymbol{h}_t$ 表达式如下。

$$\boldsymbol{Z}_t=\sigma(\boldsymbol{W}_{\mathrm{z}}\cdot[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_{\mathrm{z}}) \tag{1}$$

$$\boldsymbol{R}_t=\sigma(\boldsymbol{W}_{\mathrm{r}}\cdot[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_{\mathrm{r}}) \tag{2}$$

$$\boldsymbol{Y}_t=\boldsymbol{h}_t=(1-\boldsymbol{Z}_t)\cdot\boldsymbol{h}_{t-1}+\boldsymbol{Z}_t\cdot\tanh(\boldsymbol{W}_{\mathrm{h}}\cdot[\boldsymbol{R}_t\cdot\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{b}_{\mathrm{h}}) \tag{3}$$

式中:σ 表示激活函数 sigmoid;$\boldsymbol{x}_t$ 为输入;$\boldsymbol{h}_{t-1}$ 为上一时刻($t-1$)的隐藏状态;$\boldsymbol{W}_{\mathrm{z}}$、$\boldsymbol{W}_{\mathrm{r}}$、$\boldsymbol{W}_{\mathrm{h}}$ 分别为更新门、重置门和新的隐藏状态对应的权重矩阵;$\boldsymbol{b}_{\mathrm{z}}$、$\boldsymbol{b}_{\mathrm{r}}$、$\boldsymbol{b}_{\mathrm{h}}$ 分别为更新门、重置门和新的隐藏状态对应的偏置向量;$\boldsymbol{Y}_t$ 为输出;tanh 表示双曲正切函数。

双向门控循环单元由一个前向(Forward)GRU 和一个后向(Backward)GRU 组成,两个 GRU 单元相互独立,可以更好地捕捉序列数据的上下文关系和双向依赖关系,提高模型对数据的理解能力。本文使用三个 BiGRU 层,通过更加复杂的模型结构来处理更复杂的序列关系,适用于家族同源性分析检测任务。BiGRU 结构如图 4 所示。

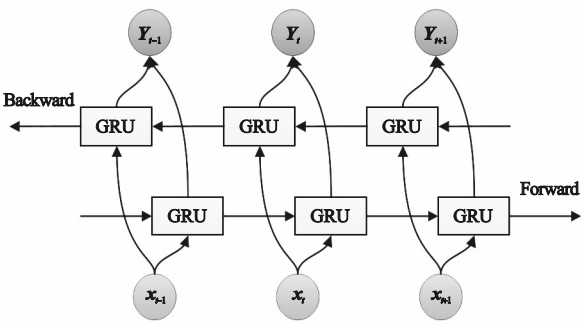

图 4　BiGRU 结构图

Fig. 4　BiGRU structure diagram

2.3　CNN-BiGRU-PC-MSA 模型

为有效利用前文中提取并融合的 API 序列与 Opcode 序列特征,充分挖掘其语义与结构信息,本文提出一种融合 CNN、BiGRU 及 PC-MSA 的混合模型(CNN-BiGRU-PC-MSA),其结构如图 5 所示。该网络架构旨在全面捕获序列数据中的局部特征和长程依赖关系,同时通过 PC-MSA 增强对重要信息的关注,从而提高模型的整体性能。

首先,采用两层一维卷积(Convolution layer)提取序列的局部特征,并通过池化与正则化(Dropout)机制降低复杂性。每层卷积的卷积核大小为3,卷积核数量为64,激活函数为ReLU,

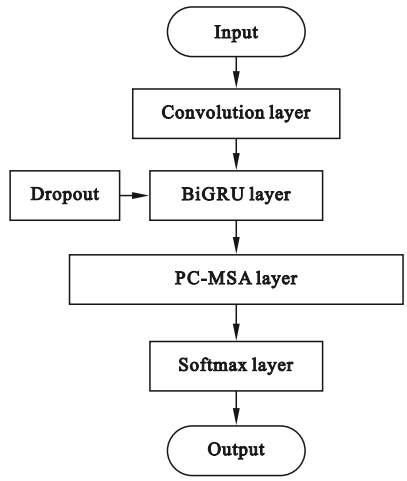

图 5　CNN-BiGRU-PC-MSA 模型结构图

Fig. 5　CNN-BiGRU-PC-MSA model structure diagram

在卷积层后引入比例为 0.2 的 Dropout 层防止过拟合。然后,通过三个 BiGRU 层(BiGRU layer)从序列的正向和反向捕捉序列中的长程依赖关系和上下文信息,每个方向的 GRU 单元数为 32,同样使用比例为 0.2 的 Dropout 防止过拟合。再后,通过 PC-MSA 层(PC-MSA layer)自适应地关注并提炼序列关键信息,获得高效且可解释的上下文向量,其中第一个卷积层的核大小为 3、通道数为 64、激活函数为 ReLU,第二个卷积层核大小为 1、通道数为 1、激活函数为 tanh。最后,将获得的上下文向量送入 Softmax 层(Softmax layer),完成恶意样本家族同源性检测任务。

3　实验及结果分析

3.1　实验环境

本文提出的恶意代码家族同源性检测模型基于 Windows 系统下的 Tensorflow + Keras 深度学习框架。模型实现的硬件环境为英特尔酷睿 GPU-RTX4060、i9-13900HX、16 G 内存,软件环境为 Windows11 系统、IDAPRO8.3、python3.7、Tensorflow2.11、Keras2.11。

为保证模型的有效训练,本实验网络参数设置如下:训练轮数(Epochs)为 500,批处理大小为 32,使用 Adam 优化器,采用其默认学习率。

3.2　模型评价指标

本文采用准确率和交叉熵损失函数作为模型的评价指标。准确率是模型预测正确的样本数与总样本数之比,计算式为

$$A=\frac{T_{\mathrm{P}}+T_{\mathrm{N}}}{T_{\mathrm{P}}+T_{\mathrm{N}}+F_{\mathrm{P}}+F_{\mathrm{N}}} \tag{4}$$

式中:A 表示准确率;T_P 表示真正例;T_N 表示真负例;F_P 表示假正例;F_N 表示假负例。

交叉熵损失函数 H 表达式为

$$H(p,q) = -\sum p(x)\log q(x) \qquad (5)$$

式中:x 表示当前样本的真实类别标签;$p(x)$ 表示目标分布;$q(x)$ 表示预测分布。

3.3 实验结果分析

本实验基于 PyCharm 2023.3 和 TensorFlow 深度学习框架。数据集包括两部分:第一部分是微软分类挑战赛数据集中数量最多的五类汇编文件样本;第二部分是来自 VirusShare 网站的数据样本。通过 VirusTotal 防病毒引擎进行家族判定,确保所选恶意代码与微软分类挑战赛数据集中相应家族匹配。通过 IDA PRO 工具对恶意代码样本进行反汇编,生成.asm 反汇编文件。共收集并处理了 8 578 个恶意代码样本,涵盖多个常见的恶意代码家族,恶意代码样本家族及数量如表 3 所示。数据集与测试集按照 8:2 的比例进行划分。

表 3 恶意代码数据集

Table 3 Malicious code dataset

恶意代码家族	训练样本数	恶意代码种类
Ramnit	1 533	Worm
Lollipop	2 389	Adware
Kelihos_ver3	2 936	Backdoor
Obfuscator. ACY	699	Obfuscated malware
Gatak	1 021	Backdoor

图 6 和图 7 所示为特征融合后采用 CNN-BiGRU-PC-MSA 模型进行同源性检测的准确率和损失函数值的运行结果图。

由图 6 可见,模型具有较好的收敛性,在 200 轮后准确率趋于稳定,其中训练集准确率接近 100%,测试集准确率可达到 98%。由图 7 可见,训练集损失值趋近于 0,测试集损失值稳定在 0.105 左右。

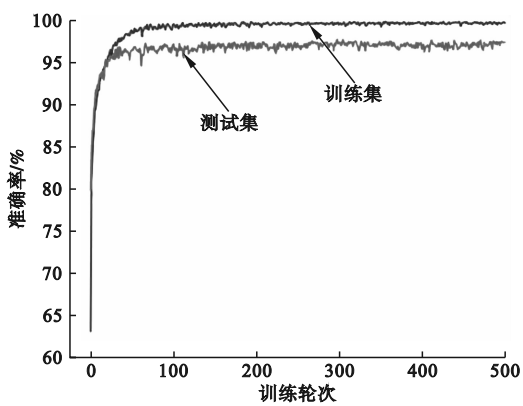

图 6 训练集和测试集的准确率

Fig. 6 Training set and test set accuracy

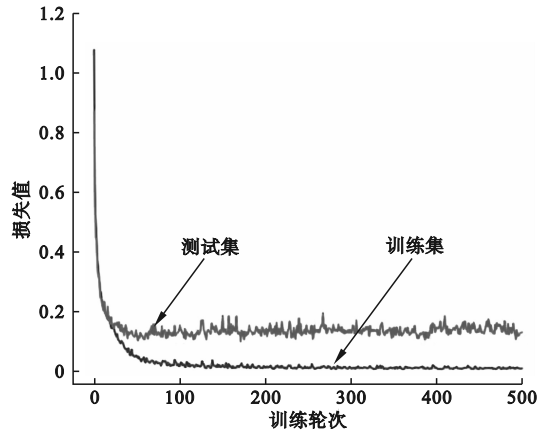

图 7 训练集和测试集损失值

Fig. 7 Training set and test set loss

为验证本文提出特征融合方法的有效性,应用同一模型分别采用混合特征与单一特征进行检测,结果如表 4 所示。

表 4 采用不同特征进行检测的结果对比

Table 4 Comparison of testing results with different characteristics

特征	准确率/%	损失值
API	94.53	0.250
Opcode	95.15	0.270
API + Opcode	98.05	0.105

由表 4 可见,采用混合特征时模型的准确率明显高于采用单一特征时的准确率,说明 API 序列与 Opcode 序列特征在识别模式和区分样本方面具有互补性,联合使用能够更全面地捕捉样本的潜在信息,进而提升模型的判别能力与鲁棒性,提高模型的分类准确率。

为验证本文构建的 CNN-BiGRU-PC-MSA 恶意代码家族同源性检测模型的有效性,同时采用 GRU-Attention[15]、CNN-BiLSTM-Attention[16] 和 CNN-BiGRU-Attention[17] 三个模型进行对比实验,对比结果如表 5 所示。

表 5 对比实验结果

Table 5 Results of comparison experiment

模型	准确率/%	损失值
GRU-Attention	89.00	0.610
CNN-BiLSTM-Attention	96.85	0.240
CNN-BiGRU-Attention	97.23	0.150
CNN-BiGRU-PC-MSA	98.05	0.105

由表 5 可见,本文提出的 CNN-BiGRU-PC-MSA 模型在准确率和损失值上均优于其他三种模型。CNN-BiGRU-PC-MSA 模型的准确率比 GRU-Attention 模型、CNN-BiLSTM-Attention 模型

和 CNN-BiGRU-Attention 模型分别提升了 10.17%、1.24% 及 0.84%。多头自注意力机制的引入能够使模型更全面地理解输入信息的结构和内容,加强对重要特征的提取,进而提高了模型的分类准确率。此外,本文 CNN-BiGRU-PC-MSA 模型的损失值低至 0.105,表明模型具备较强的泛化能力和稳定性。

为进一步验证本文提出模型中各模块对模型整体性能的影响,保持数据集和训练参数一致的前提下进行消融实验。对检测模型中的关键组成部分(PC-MSA、CNN、BiGRU)逐一消融,分析其对模型性能的影响,实验结果如表 6 所示。

表 6　消融实验结果

Table 6　Results of ablation experiment

模型	准确率/%	损失值
CNN-BiGRU-PC-MSA	98.05	0.105
CNN-GRU-PC-MSA	96.90	0.150
BiGRU-PC-MSA	96.75	0.180
CNN-BiGRU	97.10	0.450

由表 6 可以看出:将 BiGRU 替换为 GRU 后,模型的性能有所下降,说明双向结构更有利于捕捉时序依赖信息;移除 CNN 模块后,模型准确率下降最明显,说明 CNN 提取的局部关键信息在模型性能中发挥着关键作用;去除 PC-MSA 模块后,模型的损失值提升最为显著,说明 PC-MSA 模块在捕捉长程依赖关系、提升模型稳定性方面具有重要作用。

4　结论

本文提出的 CNN-BiGRU-PC-MSA 模型在恶意代码家族同源性检测的应用中表现准确、高效,具有较好的性能。通过 Doc2Vec 模型对序列信息进行向量化表示,并进行向量化特征融合,有效捕捉了序列的重要信息以及特征之间的关联,并有效避免了特征维度灾难问题;BiGRU 通过双向门控机制,有效捕捉了序列的上下文关联与双向依赖关系;PC-MSA 通过引入位置编码和卷积层,使模型能够有效处理时序信息,增强了其对局部特征的学习能力,有效提高了模型的分类准确率。

参考文献(References):

[1] MANIRIHO P, MAHMOOD A N, CHOWDHURY M J M. A study on malicious software behaviour analysis and detection techniques: taxonomy, current trends and challenges[J]. Future Generation Computer Systems, 2022, 130: 1 – 18.

[2] 王博. 新形势下网络安全的挑战与应对策略[J]. 网络安全技术与应用, 2022(10): 95 – 96.

[3] ALAWIDA M, OMOLARA A E, ABIODUN O I, et al. A deeper look into cybersecurity issues in the wake of Covid-19: a survey[J]. Journal of King Saud University: Computer and Information Sciences, 2022, 34(10): 8176 – 8206.

[4] 中国网络空间安全协会. 2024 年上半年度网络安全态势研判分析报告(第 9 期)[R/OL]. (2024 – 07 – 31)[2025 – 04 – 16]. https://www.thepaper.cn/newsDetail_forward_28267884.

[5] 刘昕仪, 彭国军, 刘思德, 等. 面向多样化编译环境的恶意代码同源性分析[J]. 信息安全学报, 2024, 9(6): 28 – 42.
LIU X Y, PENG G J, LIU S D, et al. Malware homology analysis under diverse compilation environments[J]. Journal of Cyber Security, 2024, 9(6): 28 – 42. (in Chinese)

[6] QIAN L P, CONG L. Channel features and API frequency-based transformer model for malware identification[J]. Sensors, 2024, 24(2): 580.

[7] 张宇迪, 冯永新, 赵运弢. 一种基于 FastText 的恶意代码家族分类方法[J]. 沈阳理工大学学报, 2024, 43(1): 61 – 68, 90.
ZHANG Y D, FENG Y X, ZHAO Y T. A classification method of malicious code family based on FastText[J]. Journal of Shenyang Ligong University, 2024, 43(1): 61 – 68, 90. (in Chinese)

[8] MEGIRA S, PANGESTI A R, WIBOWO F W. Malware analysis and detection using reverse engineering technique[J]. Journal of Physics: Conference Series, 2018, 1140: 012042.

[9] KAKISIM A G, NAR M, SOGUKPINAR I. Metamorphic malware identification using engine-specific patterns based on co-opcode graphs[J]. Computer Standards & Interfaces, 2020, 71: 103443.

[10] LAURENZA G, LAZZERETTI R, MAZZOTTI L. Malware triage for early identification of advanced persistent threat activities[J]. Digital Threats: Research and Practice, 2020, 1(3): 1 – 17.

[11] ZHANG S F, WU J H, ZHANG M Z, et al. Dynamic malware analysis based on API sequence semantic fusion[J]. Applied Sciences, 2023, 13(11): 6526.

[12] 刘紫煊, 王晨. 基于多特征融合的 BiLSTM 恶意代码分类[J]. 电子设计工程, 2022, 30(18): 67 – 72.
LIU Z X, WANG C. Malware code classification based on multi-feature fusion BiLSTM[J]. Electronic Design Engineering, 2022, 30(18): 67 – 72. (in Chinese)

[13] SELVAM P, KUMAR S N, KANNADHASAN S. An adaptive multi-head self-attention coupled with attention filtered LSTM for advanced scene text recognition[J]. International Journal on Document Analysis and Recognition (IJDAR), 2025: 1 – 20.

[14] LING X, WU L F, ZHANG J Y, et al. Adversarial attacks against windows PE malware detection: a survey of the state-of-the-art[J]. Computers & Security, 2023, 128: 103134.

[15] WANG H M, ZHAO Y T, WANG Z J. Doc2vec-GRU: a behavior classification method for malicious code[J]. International Journal of Network Security, 2024, 26(3): 467 – 476.

[16] DAI W, LI X H, JI W X, et al. Network intrusion detection method based on CNN-BiLSTM-Attention model[J]. IEEE Access, 2024, 12: 53099 – 53111.

[17] 陈思雨, 马海龙, 张建辉. 基于注意力机制的 CNN 和 BiGRU 的加密流量分类[J]. 计算机科学, 2024, 51(8): 396 – 402.
CHEN S Y, MA H L, ZHANG J H. Encrypted traffic classification of CNN and BiGRU based on self-attention[J]. Computer Science, 2024, 51(8): 396 – 402. (in Chinese)

(责任编辑:宋颖韬)