计算机科学
COMPUTER SCIENCE

基于多模态融合的动态恶意软件检测方法

李鉴秋, 刘万平, 黄东, 张琼

引用本文

李鉴秋, 刘万平, 黄东, 张琼. 基于多模态融合的动态恶意软件检测方法[J]. 计算机科学, 2024, 51(11A): 240200098-7.

LI Jianqiu, LIU Wanping, HUANG Dong, ZHANG Qiong. Multimodal Fusion Based Dynamic Malware Detection [J]. Computer Science, 2024, 51(11A): 240200098-7.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.240200098

基于多模态融合的动态恶意软件检测方法

李鉴秋[1] 刘万平[1] 黄 东[2] 张 琼[3]
1 重庆理工大学计算机科学与工程学院 重庆 400054
2 贵州大学现代制造技术教育部重点实验室 贵阳 550025
3 重庆机电职业技术大学信息中心 重庆 402760
(lijq@stu.cqut.edu.cn)

摘 要 近年来,新型恶意软件数量越来越多,而传统的签名式恶意软件检测方法在面对这些新恶意软件时逐渐失效,亟需开发出新的检测方法。针对这一问题,提出了一种基于多模态的动态恶意软件检测方法,该方法使用 API 调用序列作为特征,并将 API 特征映射为多模态信息,使用 2 种不同的网络模型对多模态信息进行处理,并获得检测结果。通过在多个公开的数据集上对所提方法进行了测试,获得最高 99.98%的检测准确度。实验表明,所提方法具有高准确率以及良好的泛化能力。由于该方法无需任何反汇编操作,因此可以对使用了加壳技术的恶意软件进行检测,这一特点有效提高了检测方法的鲁棒性。

关键词:恶意软件检测;多模态融合;深度学习
中图分类号 TP309.5

Multimodal Fusion Based Dynamic Malware Detection

LI Jianqiu[1],LIU Wanping[1],HUANG Dong[2] and ZHANG Qiong[3]
1 College of Computer Science and Engineering,Chongqing University of Technology,Chongqing 400054,China
2 Key Laboratory of Advanced Manufacturing Technology of the Ministry of Education,Guizhou University,Guiyang 550025,China
3 Information Center,Chongqing Vocational and Technical University of Mechatronics,Chongqing 402760,China

Abstract In recent years,the number of new types of malware has been increasing rapidly,and traditional signature-based malware detection methods are ineffective in the face of these these emerging threats. Therefore,there is an urgent need to develop new detection methods. As a solution,a novel approach based on multimodal dynamic malware detection is proposed. The method utilizes API call sequences as features,mapping these API features into multimodal information,and employs two distinct neural network models to process the multimodal information,thereby obtaining detection outcomes. By testing the proposed method on multiple public datasets,a detection accuracy of up to 99.98% is achieved. Experiments demonstrate that the proposed method exhibits high accuracy and generalization capability. Because this method does not require any disassembly operations,it can detect malware that uses packing techniques,effectively enhancing the robustness of the detection method.

Keywords Malware detection,Multimodal fusion,Deep learning

1 引言

恶意软件是一种通过各种途径入侵计算机系统,并实施非法活动的软件。一旦系统被恶意软件侵入,攻击者就可以通过文件加密等方式执行勒索等网络犯罪活动。典型的恶意软件包括木马、蠕虫和勒索软件等。

传统的恶意软件检测方法通常依赖于反汇编等技术来提取识别特征,如字符串和动态链接库(DLL)列表。这些特征用于生成存储在数据库中的签名。然而,现代恶意软件常常采用加壳等混淆技术来逃避检测,这使得传统的签名式检测方法失效[1-2]。并且,现有的基于 API 的检测方法往往只将其作为序列数据处理并进行分类,这可能会导致信息来源单一,从而降低检测准确性。针对上述 2 个问题,本文提出了一种新颖的方法——基于多模态融合的动态恶意软件检测方法。该方法针对 Windows 平台设计,使用动态 API 调用序列作为原始特征,并将 API 调用序列映射为图像和 one-hot 序列数据,从而组成多模态信息来提升检测方法的准确性。其主要优势如下:

1)由于采用了动态检测方式,因此该方法无需反汇编,这解决了加壳技术对检测的影响,并通过多模态融合实现了高检测准确度。

2)该方法只需 API 调用序列作为原始特征,因此,相对于其他多模态方法,并没有增加特征获取难度。这是因为其他多模态方法往往需要获取包括操作码和 API 序列在内的多种特征,并且获取特征时可能需要手动操作。而该方法则仅需要一种原始特征,并将原始特征映射为多模态信息来处理。

所提方法在多个公开的数据集上进行了测试。实验结果表明,在所有数据集上,该方法都获得了良好的性能表现,

基金项目:重庆市自然科学基金(cstc2021jcyj-msxmX0594)
This work was supported by the Natural Science Foundation of Chongqing,China(cstc2021jcyj-msxmX0594).
通信作者:刘万平(wpliu@cqut.edu.cn)

明显优于使用相同数据集的现有方法。这表明了其出色的泛化能力和性能优势。本文第 2 章介绍了与恶意软件相关的现有研究;第 3 章讨论了本文提出的检测方法;第 4 章展示了实验结果;最后总结全文。

2 相关工作

恶意软件检测方法根据是否在分析过程中执行软件样本可分为动态检测和静态检测两类。动态检测涉及在受控环境中执行恶意软件样本,具有较高的鲁棒性,但计算成本较高。相反,静态检测在分析过程中不执行样本,计算开销较低。然而,攻击者可能使用加壳技术来阻碍静态分析。为了提高恶意软件检测精度,近年来也有一系列研究结合了深度学习技术来提高检测器的表现性能。

Ni 等[3]提出了一种基于深度学习的恶意软件分类方法。该方法首先对样本进行反汇编,提取操作码序列作为特征。然后,利用 Sim-hash 算法[4]将操作码序列编码成图像,并采用 CNN 进行分类。尽管取得了较高的分类准确率,但该方法对加壳技术敏感,需要手动对样本进行反汇编,这存在失效的风险。

Gibert 等[5]提出了一个多模态框架,用于恶意软件分类,该框架使用的特征包括操作码、静态 API 序列和原始二进制数据。尽管这种方法通过特征融合提高了分类准确率,但其依赖于复杂的特征提取操作,并且容易受到加壳技术的影响,特别是从反汇编文件中提取的操作码和静态 API 特征。而仅依赖原始二进制数据将损害此框架的高准确性优势。

Sun 等[6]提出了一种恶意软件分类方法。该方法利用操作码序列作为特征,并结合 CNN 和 BRNN(Bidirectional Recurrent Neural Network)进行恶意软件分类。实验结果表明,这种特征融合方法即使在训练样本数量较少的情况下也能实现高分类准确率。然而,由于该方法仍然需要进行反汇编操作,因此可能无法对加壳的恶意软件进行分类或检测。

上述的方法大多使用反汇编提取的静态特征,且通常只将特征作为单一类型的数据,如图像或者序列数据,这不仅使其无法检测加壳恶意软件,并且,仅将特征单独映射为图像或序列数据也可能造成特征丢失,从而降低检测准确率。针对上述问题,本文提出了一种使用动态 API 特征,并将该特征映射为多模态信息的检测方法,从而使其可以检测加壳恶意软件,并缓解特征丢失问题。

3 方法

本文提出的方法包括两个主要部分:特征工程和分类器部分,其框架如图 1 所示。

在特征工程部分,首先将软件样本在沙箱环境中运行,以生成行为日志。在本研究中,利用动态 API 调用序列作为特征来区分恶意软件和良性软件。这是因为恶意软件和良性软件都需要使用操作系统提供的 API 来实现其功能。由于它们的功能不同,生成的 API 调用序列也不同。分类器由两个不同的神经网络组成,用于处理两个不同模态的信息。这些模型的结构将在后文中介绍,并且说明作出这种选择的原因及其必要性。

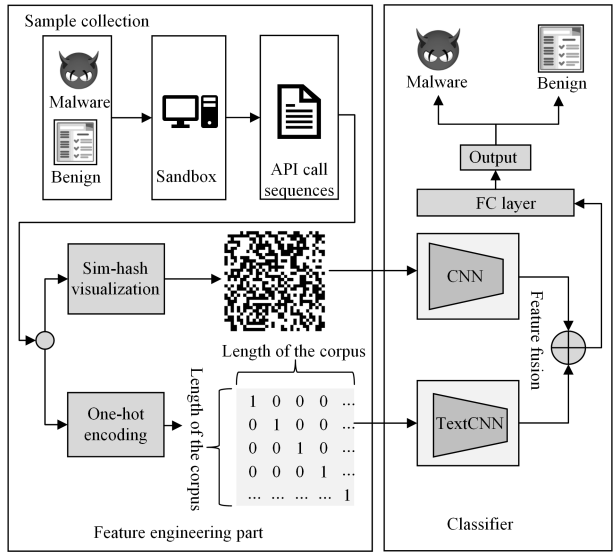

图 1 本文方法的框架

Fig. 1 Framework of the proposed method

3.1 特征工程

本文方法创新性地使用图像和序列信息来共同表示软件样本。这些不同模态的信息都源自 API 调用序列。通过这种多模态融合计算,可以提高检测的准确性。

在处理图像信息时,本文方法利用 Sim-hash 算法将 API 序列映射为图像,从而将恶意软件检测问题转化为图像分类问题。随后,使用 CNN 进行分类。Sim-hash 是一种局部敏感哈希算法,最初被应用于文本去重领域。由于 API 调用序列的结构与文本中的句子结构之间的相似性,因此选择 Sim-hash 来处理 API 序列。与传统的 hash 算法不同,Sim-hash 可以将相似的输入映射到相似的输出,提供了相似性表达的能力。Sim-hash 计算过程如图 2 所示。

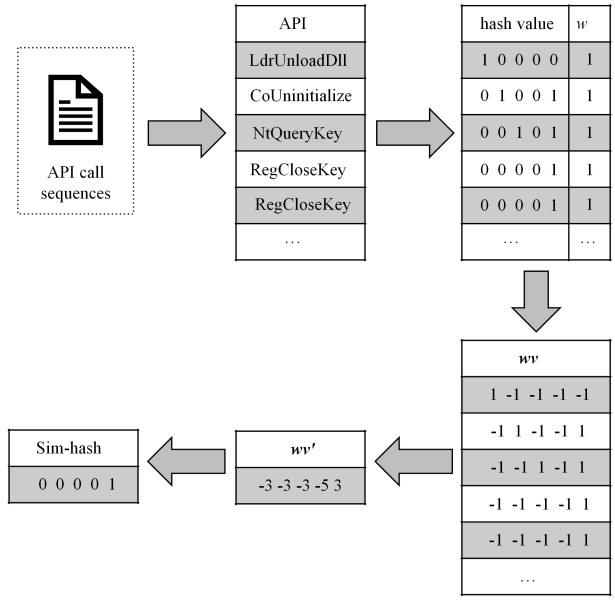

图 2 Sim-hash 计算过程

Fig. 2 Sim-hash computation process

其中,Sim-hash 算法会将每一个 API 函数映射为一个 hash 值,hash 编码方式为常见的 hash 算法,如 MD5 等。w 是算法为每个输入分配的权重。可将此 hash 值看作一个二进制串。当 hash 值的第 n 位为 1 时,权重向量 $\boldsymbol{wv}$ 的第 n 位

为 w，当 hash 值的第 n 位为 0 时，其则为 $-w$。为了方便演示，这里设所有的权重 w 均为 1，则权重向量 $\boldsymbol{wv}$ 仅由 1 和 −1 组成。

获得上述权重向量 $\boldsymbol{wv}$ 后，将所有的权重向量 $\boldsymbol{wv}$ 相加，即可获得的加权向量 $\boldsymbol{wv}'$，然后再对加权向量 $\boldsymbol{wv}'$ 进行变换，即对于该向量中的每一位数，大于 0 的位置取 1，小于 0 的位置取 0，就可以得到每一个 API 调用序列所对应的 Sim-hash 值。对于上述的例子，Sim-hash 值为 00001。

通过上述方法，就可以将一个不定长的 API 调用序列映射为等长的二进制序列，再将其重整为一个矩阵，其中 0 表示黑色，1 表示白色，这样便可得到一个代表软件样本的特征图。通过 Sim-hash 将 API 映射为图像的过程如图 3 所示。

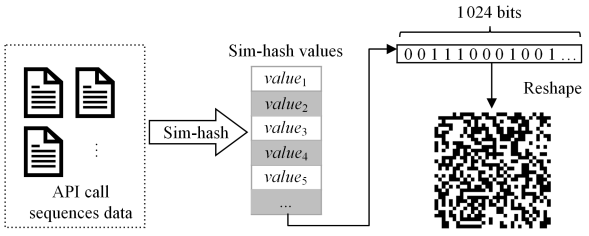

图 3　使用 Sim-hash 生成特征图

Fig. 3　Generate feature map using Sim-hash

鉴于不同的 hash 算法可能会产生不同长度的 Sim-hash 值，本文进行了一系列对照实验来评估 hash 算法选择对结果的影响。随后选择最有效的 hash 算法作为首选方法。

同时，本文使用 one-hot 向量来将 API 特征编码为数值序列。尽管 API 调用序列的结构与自然语言中的句子结构相似，但每个 API 函数本身所代表的语义与自然语言中的单词不同。此外，在恶意软件检测领域，当前并不存在专门用于表示恶意软件特征的预训练 embedding。

针对上述原因，本文尝试了使用不同维度、未预训练的 embedding 以及 one-hot 来表示 API 调用序列。实验结果表明，使用 one-hot 时能获得更高的准确性，可能的原因如下：

1）数据稀疏性：软件在沙箱中产生的动态 API 调用所包含的不同 API 种类较少，这就使得词库相对较小，从而缓解了 one-hot 带来的数据稀疏性问题。

2）本文使用的数据集比 NLP 领域的数据集更小，考虑到语料库与数据集的大小，使得 one-hot 提供的信息足以训练一个高质量模型，而不需要依赖更复杂的 embedding 结构。

并且，本文还考虑到计算开销的问题，由于 embedding 是可训练的，这意味着会在反向传播阶段对其进行更新，这会比 one-hot 耗费更多的计算时间。考虑到上述原因，特别是实验结果的差异，本文最终选择使用 one-hot 来表示序列数据。

3.2　分类器

分类模型由两个不同的神经网络组成：一个用于处理图像信息的 CNN 模型和一个用于处理序列数据的 TextCNN 模型。CNN 由于在图像处理方面有着优秀的性能表现而被广泛应用于计算机视觉领域，因此本文设计了一个浅层 CNN 模型来处理软件的特征图。该模型如图 4 所示。

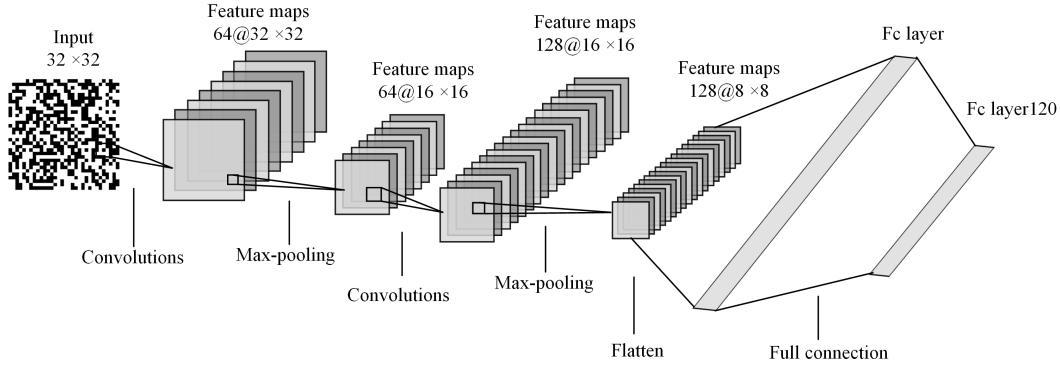

图 4　自定义 CNN 模型

Fig. 4　Custom CNN model

处理序列数据的部分使用了 TextCNN[7] 模型，该模型最初用于 NLP 领域中对文本进行分类。TextCNN 可提取输入数据中不同长度的语义信息，这是传统 CNN 所不具备的能力[7]。虽然诸如 LSTM 之类的循环神经网络也能处理序列数据，然而，TextCNN 相较于 LSTM 之类的循环神经网络具有多个优势。首先，TextCNN 的并行性使其训练过程具有更高的计算效率，能够在数据集上更快速地收敛。其次，由于没有 LSTM 那样的循环依赖，TextCNN 所需的内存资源较少，使其在处理大规模数据集或在资源有限的环境下具有优势。并且，相比 LSTM，TextCNN 的架构更加简洁，参数更少。这降低了过拟合的风险，有利于优化策略的选择和实施。同时，在本文的实验中，使用 TextCNN 时，检测准确率优于其他 RNN 模型。综上所述，基于计算效率、内存需求、适应性、简洁性等方面的考虑，在本文的应用场景下，TextCNN 是一种优于 RNN 的选择。TextCNN 的结构如图 5 所示。

其中，M 表示输入的 API 调用序列的最大长度，即序列中 API 函数的数量；L 表示语料库的长度，即使用一个维度等于 L 的 one-hot 向量去表示一个 API 函数。

TextCNN 使用 N 个大小为 $K\times L$ 的卷积核对输入的特征矩阵进行卷积运算。K 和 N 均是超参数，为用户自定义，不同的 K 和 N 值会影响特征图的大小以及数量。为了方便演示，这里设 $N=2$，$M=7$，$L=5$，那么会有 6 个 $K\times 5$ 大小（K 分别等于 2，3，4）的卷积核对输入数据进行卷积运算，经过激活函数之后就可以生产 6 个不同大小的特征图。对生成的特征图使用最大池化，分别提取每个特征图的最大值，就可以获得一个 6 维的特征向量。

上述 2 个模型的最后都添加了一个 120 维的全连接层，经过 2 个模型处理后的信息会在这里进行特征融合。消融实验表明这种特征融合方法可以提高检测的准确度。

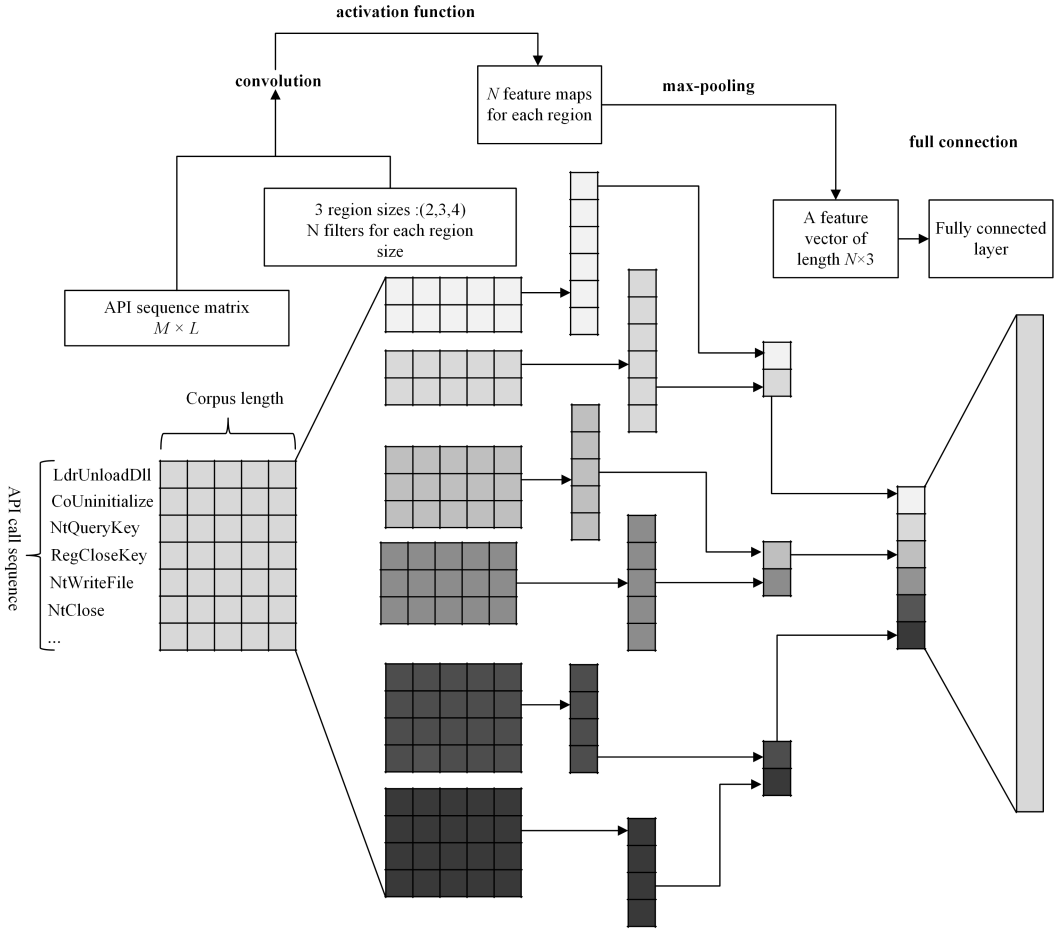

图 5 TextCNN 模型

Fig. 5 Structure of TextCNN

4 实验

在动态恶意软件检测领域，没有 ImageNet[8] 那样被广泛使用的基准数据集。虽然微软恶意软件数据集[9]较常使用，但其中只包含反汇编获得的静态特征，不适合动态检测。此外，许多研究中使用的数据集并不公开。为了测试本文方法并与将其已有方法进行比较，本文收集了所有可用于动态检测的开源数据集，并将恶意软件标记为 1，良性软件标记为 0，具体如表 1 所列。

表 1 实验使用的数据集

Table 1 Datasets used in experiments

Datasets	No. Malware	No. Benign	Total	Released date
MalBehavD-V1[10]	1 285	1 285	2 570	2022
Allan[11]	452	100	552	2019
Ki[12]	23 146	300	23 446	2015
Alibaba Cloud[13]	8 909	4 978	13 887	2018

每个数据集按 8∶2 的比例随机分为训练集和测试集。然后在测试集上评估所提出的方法。本文使用 Precision、Recall、F1-score 和准确率(Accuracy)来衡量所提方法的性能。其定义如下：

$$Precision=\frac{TP}{TP+FP} \tag{1}$$

$$Recall=\frac{TP}{TP+FN} \tag{2}$$

$$F1=\frac{2\times Precision\times Recall}{Precision+Recall} \tag{3}$$

$$Accuracy=\frac{TP+FN}{TP+FP+TN+FN} \tag{4}$$

其中，TP 是真阳性(True Positive)的数量，表示实际为阳性并且被正确预测为阳性的实例数量；FP 是假阳性(False Positive)的数量，表示实际为阴性但被错误预测为阳性的实例数量；FN 是假阴性(False Negative)，表示实际为阳性但被错误预测为阴性的实例数量；TN 表示真阴性(True Negative)，意为实际为阴性且被正确预测为阴性的实例数量。

4.1 特征选择

CNN 部分使用 Sim-hash 生成的特征图作为输入。较长的 hash 值可能包含更多信息，但也会生成更大的图像。为了研究 hash 值长度对模型性能的潜在影响，本文选择了几种常见的 hash 算法来获得不同长度的 Sim-hash 值，并在 Alibaba 数据集上进行了一系列对照实验，结果如表 2 所列。

表 2 hash 长度及其对应的准确率

Table 2 Hash lengths and their corresponding accuracies

Hash Length	Hash Algorithm	*Accuracy*/%
hash-128	MD5	92.48
hash-256	SHA-256	93.70
hash-512	SHA-512	94.35
hash-768	SHA-256 + SHA-512	94.50
hash-1024	SHA-512 + SHA3-512	95.79
hash-1536	SHA-512 + SHA3-512 + blake2b	94.31

实验结果表明，在大多数情况下，随着 hash 值长度的增加，检测准确率也在提高。然而，这种增长趋势并非持续的。结果显示，在 hash 长度为 1024 位时，模型的检测准确率最高。这样，Sim-hash 算法就能将 API 调用序列转换成一个长度为 1024 的二进制序列。通过将这个序列重整为 32×32 的矩阵，就可以获得代表软件的特征图。

常见的序列编码方法有 one-hot 和 embedding。为了选择合适的编码方式，实验使用 TextCNN 模型，在 Alibaba 数据集上测试了这两种编码方式。实验结果如表 3 所列。实验对比了不同维度的 embedding 以及 one-hot 向量。结果表明，使用 one-hot 方式来编码 API 调用序列时，获得了最佳的准确率，并且训练时间最短。因此最终选择 one-hot 作为 API 调用序列的编码方式。

表 3　编码方式的效率对比

Table 3　Efficiency comparison of encoding methods

Encoding	Training time/s	*Accuracy*/%
Embedding-300	318	96.36
Embedding-250	284	96.26
Embedding-200	260	95.61
Embedding-150	240	96.26
One-hot	202	96.62

4.2　模型选择

在 CNN 模型部分的选择过程中，本文在 Alibaba 数据集上进行了对照实验，比较了包括 VGG[14]，DenseNet[15]，ResNet[16] 这类常见的深层 CNN 模型以及自定义的浅层 CNN 模型。这些模型使用的特征图是用 1024 位的 Sim-hash 值生成的。实验结果如图 6 所示。

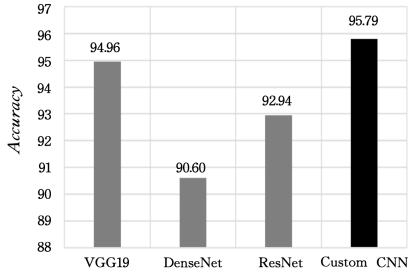

图 6　不同的 CNN 模型对准确率的影响

Fig. 6　Effect of different CNN models on accuracy

结果表明，自定义的浅层 CNN 模型达到了 95.79%的准确率，优于其他深层 CNN 模型。这可能归因于代表软件的特征图比 CV 领域中实体识别问题中的复杂图像更小，并涉及较少的分类类别。因此，没有必要使用复杂的深度结构来提取纹理特征。此外，实验中使用的数据量相对于 CV 领域中的数据量较少。使用过于复杂的模型来拟合这样的数据可能会导致过拟合。此外，浅层 CNN 具有更少的参数，消耗更少的内存，并且训练速度更快。因此，在本文的应用场景中，浅层 CNN 模型被证明是更好的选择。

另一个子模型的输入是经过 one-hot 编码的 API 调用序列。处理序列数据的常见模型包括 RNN 和 TextCNN。本文在 Alibaba 数据集上进行了一系列的对照实验来评估这些模型的性能，如表 4 所列。

实验测试了 1～3 层不同类型的 RNN 模型，以及 TextCNN 模型。结果显示，TextCNN 在测试集上取得了 96.62%的准确率，优于实验中的对比模型。此外，TextCNN 具有参数少和训练速度快的优点。

表 4　序列处理模型对比

Table 4　Comparison of sequence processing models

Model	Training time/s	Parameters	*Accuracy*/%
GRU-3	450	281202	94.78
GRU-2	416	220602	94.13
GRU-1	342	160002	94.20
LSTM-3	758	361602	95.14
LSTM-2	560	280802	94.71
LSTM-1	431	200002	94.20
TextCNN	202	80732	96.62

4.3　消融实验

为证明本文提出的多模态方法在恶意软件检测中的必要性，本文在所有 4 个数据集上进行了消融实验。实验比较了单独使用 CNN 模型、单独使用 TextCNN 模型以及使用多模态方法的性能。实验结果如图 7 所示。

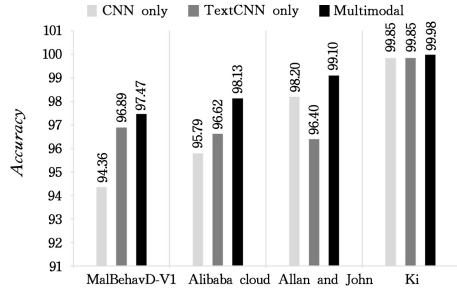

图 7　本文方法在不同数据集上的准确率表现

Fig. 7　Accuracy of the proposed method on different datasets

结果显示，本文提出的多模态检测方法在 4 个数据集上的性能均优于两种单模态方法。在这些数据集上，本文方法取得了很高的准确率，最高达到 99.98%。因此可得出结论：本文方法在保持高检测准确率的同时表现出良好的泛化性，而且在多个不同数据集上的消融实验证明了多模态方法优于单模态方法，这进一步验证了本文方法的必要性。本文还使用了多种指标来衡量所提方法的性能，如表 5 所列，可见本文方法在这些指标上都获得了良好的性能表现。

表 5　使用不同的指标来衡量本文方法

Table 5　Using different metrics to assess the proposed method

Datasets	*Accuracy*/%	*F1-score*	*Precision*	*Recall*
MalBehavD-V1	97.47	0.9747	1.0000	0.9508
Allan and John	99.10	0.9948	0.9897	1.0000
Ki	99.98	0.9998	0.9998	1.0000
Alibaba Cloud	98.13	0.9850	0.9885	0.9817

本文还将所提方法与现有的方法进行了比较。为了消除数据集不同造成的影响，本文仅选择了使用相同数据集的方法进行比较。对比结果如表 6 所列。

本文对比了使用相同数据集的现有方法，这些方法均使用 API 调用序列作为特征，但现有方法仅仅将 API 特征作为单一的序列数据处理，使用 embedding 或 one-hot 将这些特征数值化表示时，仍有可能无法准确表示其语义信息，从而导致检测准确率降低。而本文的方法考虑到单一类型数据可能导

致的信息丢失问题，同时使用数值序列和图像这2种模态的信息来表示API特征，并通过多模态融合来提高准确率。结果表明，在Accuracy和F1-score这2种主流的评价指标上，本文方法均优于现有方法，这充分说了对API调用序列进行多模态表示的必要性和优越性，也为本文方法的有效性提供了有力的支持。

表6 本文方法与现有方法的对比

Table 6 Comparison of the proposed method with other methods

Datasets	Approach	*Accuracy*/%	*F1-score*
Allan	MalDetConv[10]	95.73	—
	Proposed method	**99.10**	**0.9948**
MalBehavD-V1	MalDetConv[10]	96.10	—
	Proposed method	**97.47**	**0.9747**
Ki	Amer 等[17]	99.90	0.9990
	Ki 等[12]	99.80	0.9990
	Amer 等[18]	99.90	0.9990
	MalDetConv[10]	99.93	—
	Tran 等[19]	99.06	—
	Proposed method	**99.98**	**0.9998**
Alibaba Cloud	Gao 等[20]	—	0.8450
	Xu 等[21]	93.44	—
	Zhang 等[22]	97.30	0.9850
	Zhang 等[23]	94.49	0.9402
	Zhang 等[24]	96.10	0.9700
	Proposed method	**98.13**	**0.9850**

结束语 本文提出了一种基于多模态信息融合的动态恶意软件检测方法，其创新之处在于使用API调用序列作为原始特征，并将原始特征转换为多模态信息，并结合了CNN和TextCNN两个子模型进行多模态信息处理和特征融合，从而检测恶意软件。该方法的独特之处在于获得了多模态检测准确性高的优点，同时又没有增加特征获取的难度。

实验结果显示，本文方法能够高效区分恶意软件和良性软件，并且在多个公开数据集上均表现优越，超越了现有方法。值得强调的是，本文方法无需进行反汇编操作，这一特点解决了静态检测方法在应对恶意软件加壳技术时的困境，显著提高了检测方法的鲁棒性和实用性。

虽然本文提出的方法相比现有方法表现出了诸多优势，特别是其具有高检测准确率，然而，本文方法仅使用动态特征。在后续的研究中，会尝试将多模态检测的思想融入混合式检测或静态检测中，从而为计算机提供更全面的防护。

参 考 文 献

[1] GENG J, WANG J, FANG Z, et al. A survey of strategy-driven evasion methods for PE malware: Transformation, concealment, and attack[J]. Computers & Security, 2024, 137: 103595.

[2] LIU W, ZHONG S. Web malware spread modelling and optimal control strategies[J]. Scientific Reports, 2017, 7: 42308.

[3] NI S, QIAN Q, ZHANG R. Malware identification using visualization images and deep learning[J]. Computers & Security, 2018, 77: 871-885.

[4] MANKU G S, JAIN A, DAS SARMA A. Detecting near-duplicates for web crawling[C]// Proceedings of the 16th International Conference on World Wide Web. 2007: 141-150.

[5] GIBERT D, MATEU C, PLANES J. HYDRA: A multimodal deep learning framework for malware classification[J]. Computers & Security, 2020, 95: 101873.

[6] SUN G, QIAN Q. Deep learning and visualization for identifying malware families[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 18(1): 283-295.

[7] ZHANG Y, WALLACE B C. A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification[C]// Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2017: 253-263.

[8] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database[C]// 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009: 248-255.

[9] RONEN R, RADU M, FEUERSTEIN C, et al. Microsoft malware classification challenge[J]. arXiv: 1802.10135, 2018.

[10] MANIRIHO P, MAHMOOD A N, CHOWDHURY M J M. MalDetConv: Automated Behaviour-based Malware Detection Framework Based on Natural Language Processing and Deep Learning Techniques[J]. arXiv: 2209.03547, 2022.

[11] ALLAN N, NGUBIRI J. Windows PE API calls for malicious and benigin programs[J]. International Journal of Technology and Management, 2019, 3(2): 1-9.

[12] KI Y, KIM E, KIM H K. A novel approach to detect malware based on API call sequence analysis[J]. International Journal of Distributed Sensor Networks, 2015, 11(6): 659101.

[13] Alibaba Cloud Malware Detection Based on Behaviors [EB/OL]. [2018]. https://tianchi.aliyun.com/getStart/information.htm?raceId=231694.

[14] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[C]// 3rd International Conference on Learning Representations (ICLR 2015). Computational and Biological Learning Society, 2015.

[15] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 4700-4708.

[16] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[17] AMER E, ZELINKA I. A dynamic Windows malware detection and prediction method based on contextual understanding of API call sequence[J]. Computers & Security, 2020, 92: 101760.

[18] AMER E, EL-SAPPAGH S, HU J W. Contextual identification of windows malware through semantic interpretation of api call sequence[J]. Applied Sciences, 2020, 10(21): 7673.

[19] TRAN T K, SATO H. NLP-based approaches for malware classification from API sequences[C]// 2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES). IEEE, 2017: 101-105.

[20] GAO M, WU P, PAN L. Malware Detection with Limited Supervised Information via Contrastive Learning on API Call Se-

quences[C]// International Conference on Information and Communications Security. Cham: Springer International Publishing, 2022: 492-507.

[21] XU A, CHEN L, KUANG X, et al. A hybrid deep learning model for malicious behavior detection[C]// 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS). IEEE, 2020: 55-59.

[22] ZHANG Z, LI Y, DONG H, et al. Spectral-based directed graph network for malware detection[J]. IEEE Transactions on Network Science and Engineering, 2020, 8(2): 957-970.

[23] ZHANG S, WU J, ZHANG M, et al. Dynamic Malware Analysis Based on API Sequence Semantic Fusion[J]. Applied Sciences, 2023, 13(11): 6526.

[24] ZHANG Z, LI Y, WANG W, et al. Malware detection with dynamic evolving graph convolutional networks[J]. International Journal of Intelligent Systems, 2022, 37(10): 7261-7280.

LI Jianqiu, born in 1997, postgraduate, is a member of CCF (No. R6779G). His research interests is malware detection.

LIU Wanping, born in 1986, Ph.D, associate professor, master supervisor, is a member of CCF (No. 43152M). His main research interests include network and information security.