计算机科学
COMPUTER SCIENCE

跨模态融合的少样本勒索软件分类器:基于预训练模型的多模态编码

尹创, 刘建毅, 张茹

引用本文

尹创, 刘建毅, 张茹. 跨模态融合的少样本勒索软件分类器:基于预训练模型的多模态编码[J]. 计算机科学, 2026, 53(4): 435-444.

YIN Chuang, LIU Jianyi, ZHANG Ru. Cross-modal Fusion Few-sample Ransomware Classifier:Multimodal Encoding Based on Pre-trained Models [J]. Computer Science, 2026, 53(4): 435-444.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.250500078

跨模态融合的少样本勒索软件分类器:基于预训练模型的多模态编码

尹 创 刘建毅 张 茹
北京邮电大学网络空间安全学院 北京 100876
(yinnchuang@163.com)

摘 要 勒索软件通过加密关键数据勒索受害者支付赎金。2023年勒索导致的赎金总额已超10亿美元。精确分类勒索软件对安全防护具有重要意义,但勒索软件样本的数量往往有限。鉴于此,提出了一种跨模态融合少样本勒索软件分类器CMFu,包含特征构建模块、编码模块和融合模块。特征构建模块用于生成跨模态特征。编码模块基于两个预训练模型构建编码器,对不同模态的特征进行编码。融合模块对编码数据进行整合,实现最终的分类。实验通过设置10%,30%和50%的训练样本比例来评估模型的性能。CMFu在所有指标上均优于对比模型。当样本比例为30%时,CMFu的精确率、召回率和F1分数分别为0.91,0.91和0.90,优于所有对比模型。当样本比例降至10%时,指标仍能保持在较高水平,为0.78,0.84和0.80,证实了CMFu的少样本勒索软件分类效果,消融实验验证了基于预训练的编码器的可行性以及使用骨干网络融合的必要性。

关键词:勒索软件;少样本;多模态;预训练模型;深度学习
中图分类号 TP393

Cross-modal Fusion Few-sample Ransomware Classifier:Multimodal Encoding Based on Pre-trained Models

YIN Chuang,LIU Jianyi and ZHANG Ru
School of Cyberspace Security,Beijing University of Posts and Telecommunications,Beijing 100876,China

Abstract Ransomware,defined by its mechanism of encrypting critical data to extort payment from victims,results in global ransom payments exceeding $1 billion in 2023. Precise classification of ransomware is crucial for effective security defense. However,ransomware samples are often small. To address this challenge,this paper proposes a cross-modal fusion few-shot ransomware classifier named CMFu,comprising a feature construction module,an encoding module,and a fusion module. The feature construction module generates cross-modal features. The encoding module employs two pre-trained models to construct encoders that encode features from different modalities. The fusion module integrates the encoded data to achieve the final classification. Experimental evaluation assesses model performance under training sample ratios of 10%,30%,and 50%. CMFu outperforms all baseline model across all metrics. At a 30% sample ratio,CMFu achieves precision,recall,and F1-score of 0.91,0.91,and 0.90,respectively,demonstrating superior performance. When the sample ratio decreases to 10%,these metrics remain high at 0.78,0.84,and 0.80,confirming its ability in few-shot ransomware classification. Furthermore,ablation studies validate both the viability of the pre-training-based encoders and the necessity of employing backbone networks for fusion.

Keywords Ransomware,Few samples,Multiple modalities,Pre-trained models,Deep learning

1 引言

随着物联网和云计算等信息与通信技术的发展,恶意软件对网络安全构成严重威胁[1]。勒索软件作为一种特殊的恶意软件,系统性地劫持并加密关键数据,以此勒索受害者支付赎金[2]。近年来,勒索软件攻击导致的赎金金额不断增长,2023年,勒索软件支付的总金额首次超过10亿美元[3]。传统的勒索软件检测和分类方法往往需要耗费大量的人力和物力。

深度学习技术提出后,被应用于恶意软件分类领域,以提高分类的效率[4]。这些方法也被进一步应用于勒索软件的分类任务中。但问题在于,勒索软件变种的快速演变[5],使得基于过往数据建模的模型在面对新出现的勒索软件时,分类效果表现不佳。此外,由于快速演变结构的特性,某些勒索软件可能只有少数样本,导致普通的方法难以学习到足够的知识以进行有效分类[6]。因此,如何通过有限的数据学习勒索软件的特征,并将其应用于分类任务,成为一个亟待解决的问题,即所谓的少样本学习任务。

到稿日期:2025-05-19 返修日期:2025-09-08
基金项目:国家自然科学基金(U21B2020)
This work was supported by the National Natural Science Foundation of China(U21B2020).
通信作者:刘建毅(liujy@bupt.edu.cn)

在少样本学习领域，虽然已经有一些先进的方法，例如模型无关的元学习(Model-Agnostic Meta-Learning，MAML)[7]，用于应对样本不足时的场景，但这些方法较少被用于勒索软件分类场景，且大多仅考虑单一模态的特征[8]。

为解决上述问题，需要设计一个用于少样本勒索软件分类的方法。近年来，拥有数百万参数的预训练模型不断涌现。通过复杂的预训练过程，这些模型获得了可迁移的知识，能够高效适应下游任务[9-10]，且预训练模型对少样本数据进行迁移是可行的[11]，并已开始应用于少样本学习领域，通过可迁移知识提供更好的分类效果[12]。与此同时，多模态特征识别方法也备受关注，它能为模型提供多种信息，且多模态信息已被证实对少样本学习具有促进作用[13]。

基于此，本文提出了一种跨模态融合的少样本勒索软件分类器(Cross-Modal Fusion for Few-Shot Ransomware Classifier，CMFu)。该方法以 OPCODEs 和 API 调用作为跨模态特征，引入预训练的 BERT 和 CLIP，对其改造后作为特征编码器提取关键信息。最终通过一个骨干融合模型整合跨模态编码，输出分类结果。

本文的主要贡献总结如下：

1)提出了一种基于预训练模型编码的跨模态勒索软件分类器，专门应用于少样本勒索软件分类场景。针对现存少样本勒索软件分类方法较少的问题，借助预训练模型的知识迁移能力，使少样本下的特征提取更加完善。

2)通过对两个不同模态预训练模型进行迁移并调整结构使其作为方法内部编码器的方式，解决少样本场景下仅考虑单一模态特征的问题。

3)对比了多个基准模型，证明了本文方法的优越性，且通过消融实验验证了各个模块的重要性。

本文第 2 章全面讨论了勒索软件分类领域的相关工作；第 3 章详细描述了所提出的方法，包括方法内部的各个模块；第 4 章进行了数据集描述和实验分析；最后总结全文并展望未来。

2 相关工作

1)勒索软件分类

勒索软件分类技术很大程度上沿袭并发展自恶意软件分类的研究路径。该领域技术经历了从早期的耗费大量资源且效率低下的基于签名扫描的方法[14]，到基于机器学习的分类算法(如 KNN，SVM[15-16])，再到复杂深度学习方法(如 CNN，LSTM[17]，并提出使用 GAN 进行数据增强[18])的转变过程。不同时期的方法在特征选择与模型设计上的结合，一直是研究的重点[19]。

虽然恶意软件分类方法为勒索软件分类提供了基础，但直接将前者迁移应用于后者并非总是有效[20-21]。此外，勒索软件相比一般恶意软件变体演化更为迅速且目的性强，当前研究普遍忽视了勒索软件研究的样本稀缺性。获取高质量、时效性的勒索软件样本非常困难且耗资巨大。现有勒索软件分类方法多依赖于相对充足的数据集进行训练和评估，未能考虑在现实环境中更常见的少样本场景下的分类可行性，这使得现有模型的鲁棒性和泛化能力在面对不断涌现的新勒索软件变体时面临严峻考验。

2)特征表示

在特征表示层面，恶意软件分析主要聚焦于图像和文本两大类。

将二进制文件转化为字节级图像[22]或反汇编指令级灰度图[23]，利用视觉模式进行识别。这种方法直观，但对软件行为的语义信息捕捉可能不够深入[24]。

提取 API 调用序列、网络交互信息、字符串常量或行为描述文本[25-26]等作为特征。此类方法能更直接地反映程序的行为逻辑，但可能丢失程序的整体结构信息。

认识到单一模态的局限性后，一些研究探索了图像与文本信息的融合，如 Lisa 等利用多模态特征进行深度学习[27]。多模态特征表示方法通过整合不同视角的信息，理论上能提供更全面、鲁棒的特征表示，可以提升少样本分类效果。

3)少样本分类

少样本学习方法可以有效应对样本稀缺下的分类任务，例如元学习框架 MAML[7]旨在训练模型快速适应新任务，基于预训练语言模型(如 BERT)的迁移学习则在少样本文本分类中展现了强大潜力[28]。

恶意软件少样本分类领域已有初步探索，包括使用加权原型融合操作码序列与恶意软件图像以及将恶意软件图像直接应用于 MAML 框架[8]。这些工作证明了少样本学习方法在恶意软件分类中是可行的，其效果很大程度上依赖于高质量、信息丰富的特征表示。

勒索软件相较于一般恶意软件具有更强的目的性和更快的变体迭代速度，其特征分布和行为模式可能更加特异化。然而，专门针对勒索软件场景、有效利用少样本学习技术的研究非常有限。

4)跨模态学习

少样本学习的核心挑战往往是信息不足，利用多模态信息互补成为提升性能的重要策略。Lin 等[13]从理论上探究了多模态信息有助于少样本学习，整合互补模态能够提供更丰富的特征表示，如图 1 所示。Srivastava 等[30]提出的专用编码器＋联合 Transformer 架构，证明了跨模态信息共享的有效性。

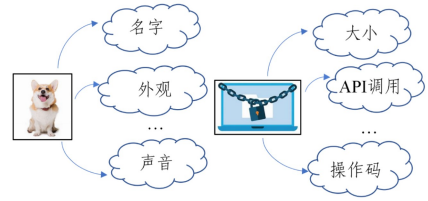

图 1 勒索软件的多模态特征

Fig. 1 Multi-modal features of ransomware

总结来看，勒索软件分类研究面临三重核心挑战：一是现实场景下新勒索软件变体的可用样本量极少，二是缺乏有效利用勒索软件多模态特征进行互补的策略，三是现有方法未能充分考虑利用预训练大模型的知识迁移能力来克服样本限制。现有研究要么专注于解决少样本问题但未专门针对勒索软件进行分类设计，要么未将跨模态融合用于勒索软件少样本分类下，要么利用少样本技术但未有效结合强大的预训练

模型,尤其缺乏将预训练模型的强大表征能力、跨模态信息的互补优势,以及针对勒索软件少样本特性的分类方法三者有效结合的方案。

因此,本文提出了融合大规模预训练模型的知识基础、勒索软件的多模态特征表示以及迁移学习技术,设计了 CMFu 框架,利用 BERT 提取文本模态特征、CLIP 提取图像模态特征,再通过联合骨干网络实现深层跨模态信息融合与交互,用于少样本勒索软件分类。

3　方法描述

CMFu 是一个跨模态融合的少样本勒索软件分类器,其框架如图 2 所示。为了获取多模态信息,CMFu 首先提取 API 调用作为文本模态特征。然后提取二进制 OPCODEs,将其构建为灰度图,作为视觉模态特征。在获得两种模态特征后,引入预训练的 BERT 和 CLIP 对这两种模态特征进行微调迁移,以便提取模型相应编码结构的输出,得到对勒索软件的视觉和文本模态特征编码器,用于独立计算并输出特征编码。此外,该方法中设计了一个骨干网络作为融合模块。该网络由一个编码器层(一个自注意力机制层、一个带有层归一化和残差连接的前馈网络)和两个连接层组成,用于整合来自预训练模型编码器的编码,输出最终结果。本章讨论了所提出的少样本勒索软件分类方法的总体方法论和各个模块。

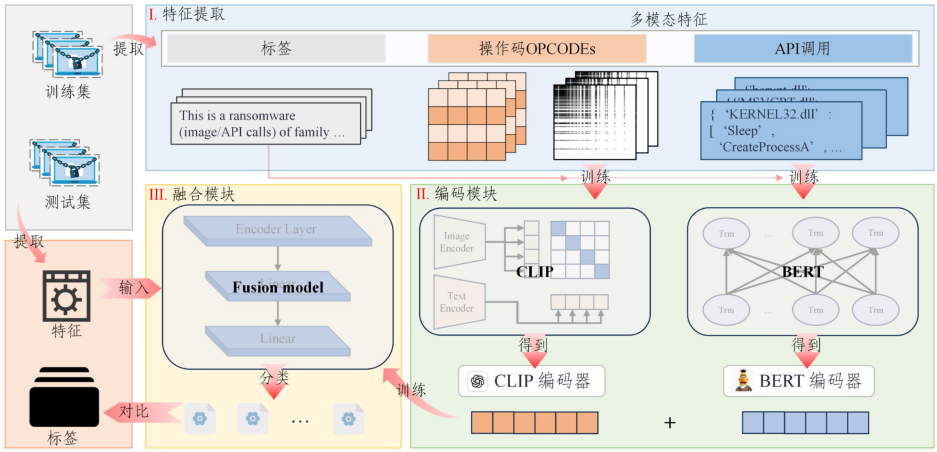

图 2　CMFu 方法框架

Fig. 2　Framework of CMFu

3.1　特征提取

勒索软件大多是 PE(Portable Executable)文件。PE 文件格式是 Windows 操作系统的标准可执行文件格式,分为 3 个主要部分,分别是 DOS 头、PE 头、节表(Section Tables)和节(Section),如图 3 所示。

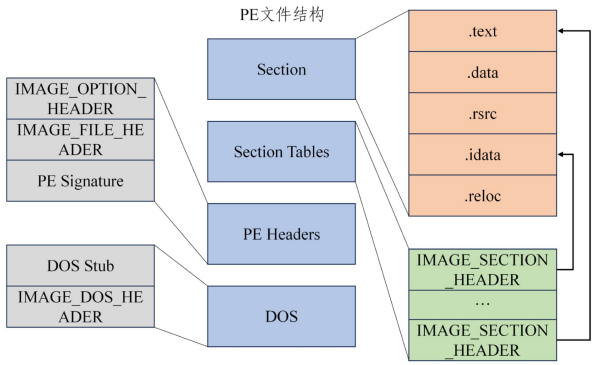

图 3　PE 文件结构

Fig. 3　Structure of PE file

本文方法的特征提取主要针对节(Section)部分进行,节本身存储 PE 文件的实际内容,例如. text 节(包含可执行代码)和. idata 节(包含从 DLL 导入的函数地址)。本文提出了一种跨模态融合的少样本勒索软件分类方法,在特征构建模块,该方法从原始的. data 节中提取 API 调用,并从. text 节中提取二进制序列数据并反汇编为操作码。整个提取过程使用了 Python 中的 pefile 包,它允许用户解析和分析 PE 文件,并访问 PE 文件的各种组件,例如节的部分。使用 pefile 包从. text 节提取 API 调用以及从. idata 节提取操作码(OPCODEs)的过程如图 4 所示。

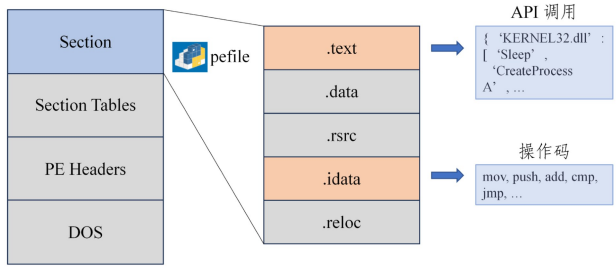

图 4　基于 PE 文件结构的特征提取

Fig. 4　Feature extractions based on PE file structure

在提取了 API 调用和操作码之后,对于这两种能够从行为上表征勒索软件的关键特征,采用不同的特征构建方法。API 调用在理解程序的行为和功能方面具有重要意义,可疑的 API 调用活动可能表明存在潜在的恶意行为。本文采用文本模态来处理 API 调用,便于将其输入后续的预训练语言模型 BERT 中学习从 API 调用到勒索软件类别的映射关系。操作码是 CPU 执行程序的直接指令,反映程序的底层逻辑和执行流程。由于其数量庞大,本文采用图像转化方法来处理操作码。本文采用的可视化方法的流程如下。

1)统计样本中出现频率最高的 128 个操作码,并将其按顺序排列为序列$\{OP_1, OP_2, \cdots, OP_{128}\}$。

2)对每个勒索软件样本构建 128×128 的矩阵 $\boldsymbol{M}$,通过反

汇编从.text部分提取OPCODEs，得到频率序列$\{F_1, F_2, \cdots, F_{128}\}$。

3)计算矩阵 $\boldsymbol{M}$ 中的元素，计算式如下：

$$\begin{cases} M_{1,i} = 255/(1+F_i) \\ M_{i,1} = 255/(1+F_i) \end{cases} \tag{1}$$

$$M_{i,j} = \mathrm{Max}(M_{1,j}, M_{i,1}) \tag{2}$$

其中，F_i 表示第 i 个操作码的出现次数，$\boldsymbol{M}$ 表示每个勒索软件的构造矩阵，$M_{i,j}$ 表示第 i 行和第 j 列的矩阵元素。由式(1)可知，操作码出现的频率与图像中相应行、列的暗度呈正相关。

4)将矩阵转换为灰度图像时，由于操作码频率范围为0～+∞，经过上一步计算，矩阵元素已被缩放到0～255。因此，可以直接将矩阵元素作为灰度图像的像素值进行转换。灰度图像的像素值介于0(黑色)到255(白色)之间，中间值表示不同的灰度级别。这种可视化方法得到的图像被称为最小逆频率共现矩阵灰度图像。整个过程如图5所示。

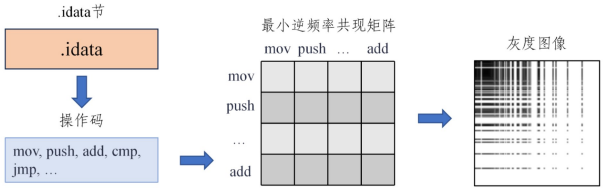

图5 构建图像特征

Fig. 5 Construction of visual feature

3.2 编码模块

1)BERT编码器

在从每个勒索软件样本中提取API调用后，本文方法使用BERT对其进行编码。BERT通过在大量未标记文本数据上进行预训练，在自然语言处理任务中取得了显著成果。它采用了掩码语言模型(Masked Language Model，MLM)和下一句预测(Next Sentence Prediction，NSP)两种策略，使其能够捕捉文本中的丰富上下文信息和依赖关系。

与传统的文本分类方法相比，BERT拥有数百万个参数，能够更精准地捕捉API调用序列中的细微差异和语义信息。通过在少量勒索软件样本上进行微调，BERT能够快速适应特定的分类任务，同时保留其强大的上下文建模能力。这种少样本微调策略不仅提高了模型的适应性和灵活性，还显著降低了对大规模标注数据的依赖，为勒索软件分类任务提供了一种高效且创新的文本编码解决方案。

BERT的架构和预训练目标使其能够很好地适应文本模态的特征提取任务，能够捕捉API调用序列中的细微差异和语义信息，为后续的分类任务提供高质量的文本特征表示。

2)CLIP编码器

本文从勒索软件中提取操作码，并构建最小逆频率共现矩阵，将其转换为灰度图像。在有限的训练集条件下，本文方法创新性地采用多模态学习模型CLIP作为图像编码器。

CLIP是一个多模态预训练模型，它同时包含图像编码器和文本编码器。CLIP通过对比学习在大量的图像-文本对上优化图像和文本向量之间的相似性，能够将图像特征映射到一个语义空间中。对于操作码生成的灰度图像，CLIP能够提取出具有语义意义的图像特征。

CLIP的预训练方式使其具有很强的泛化能力，即使在样本数量有限的情况下，也能通过微调有效地将图像特征编码为高质量的向量表示，利用预训练模型的知识库，快速适应新的分类任务，为少样本图像分类任务提供了一种全新的视角和高效解决方案。

CLIP本身就是基于图像和文本对齐的角度考虑而诞生的模型，本文考虑到其内部的结构支持CLIP编码出匹配文本语义性质的图像编码，通过多模态学习机制在语义层面上进行融合。这为本文的跨模态融合提供了基础，使得图像特征和文本特征能够在同一个语义空间中进行交互和整合。

文本和图像模态从不同角度反映勒索软件特征，预训练模型BERT和CLIP分别编码后可实现多模态信息互补。后续CMFu的融合模块进一步整合编码信息，提升分类效果。

编码器提取过程如图6所示。

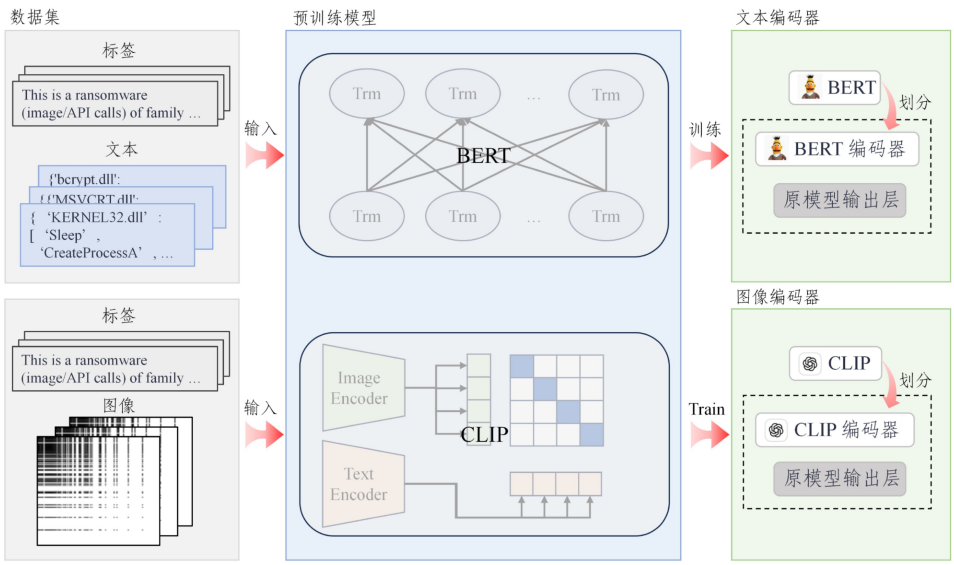

图6 构建CMFu编码模块

Fig. 6 Construction of encoding module in CMFu

BERT 接收 API 调用（文本格式）和描述勒索软件家族的标签，而 CLIP 接收操作码（图像格式）和标签。通过微调实现模型的知识迁移，训练完成的 BERT 和 CLIP 模型随后移除输出层，形成独立的编码模块。编码器输出的编码信息被用作后续模块的输入，为后续的分类任务提供了丰富的特征表示。这种结合文本和图像编码器的多模态方法不仅充分利用了不同模态数据的优势，还通过创新的模型改造和微调策略，显著提升了勒索软件分类的准确性和适应性。

3.3　融合模块

本文中，文本模态特征（API 调用）和图像模态特征（操作码生成的灰度图像）的提取过程基于相同的样本进行，因此从样本层面来说，同一勒索软件的两种模态特征保持了对齐。

此外，文本模态和图像模态的特征分别通过 BERT 和 CLIP 进行编码，能够将输入的文本和图像特征编码为固定维度的向量表示。通过对预训练模型内部结构的处理，得到具有模态意义的编码序列，便于后续的融合操作。

为了融合两种模态的编码信息，本文设计了一个融合模块来整合编码模块的输出。该模块包括一个自注意力机制层、一个带有层归一化和残差连接的前馈网络，连接两个线性层进行输出。在融合模块中，通过自注意力机制和前馈网络对拼接后的特征向量进行处理。自注意力机制动态调整不同模态特征的重要性，进一步优化特征的对齐和融合效果。

对于勒索软件样本，前述模块分别提取其文本模态（API 调用文本）和图像模态（操作码生成的灰度图像）的特征。这些特征分别输入 BERT 编码器和 CLIP 编码器进行编码处理。假设 BERT 编码器输出一个 d_{BERT} 维的编码序列，CLIP 编码器输出一个 d_{CLIP} 维的编码序列，将这两个编码序列进行拼接，形成一个（$d_{\text{BERT}}+d_{\text{CLIP}}$）维的融合序列。通过编码序列的拼接操作，可以将两种模态的特征向量合并为一个更高维度的向量，从而表示多模态的信息。

随后，该融合序列通过融合模块中的编码器层进行处理，进一步整合两种模态的编码信息。编码器层的设计采用了 Transformer 架构中的多头自注意力机制，从多个角度对融合序列进行特征提取和交互分析。通过这种方式，模型能够捕捉到两种模态信息之间的潜在关联，从而生成更加丰富和准确的融合特征。

与现有技术相比，本文融合模块通过引入自注意力机制，实现了对多模态信息的动态加权融合，而以往的研究中少有提及。其次，融合模块中层归一化和残差连接的结合提高了模型的训练效率和稳定性。最后，通过精心设计的编码器层和全连接层，实现了对融合特征的高效提取和分类，使得模型在勒索软件检测任务上取得了显著的性能提升。

随着数据输入模型进行训练，融合模块能够自动调整模型权重，动态地结合两种模态信息。这种自适应的融合策略使得模型能够更好地应对勒索软件样本的多样性和复杂性，为多模态勒索软件检测提供了一种高效且创新的解决方案。

3.4　跨模态方法 CMFu

CMFu 使用预训练模型，通过其知识库所赋予的对模态的理解能力，对多模态特征进行编码，从而提取每种勒索软件的特征。引入融合模块，用于整合两种模态的信息。本文方法包含两个主要创新策略：第一个是改造预训练模型，使其作为方法的编码器，用于进行特征处理以及提取；第二个则是考虑了少样本任务信息不足的缺陷，微调预训练模型进行知识迁移，并对勒索软件两种不同模态的表示特征进行结合，通过一个骨干网络融合编码信息。

CMFu 方法的实现方式如下：

$$O_{\text{t}}=M_{\text{BERT}}(F_{\text{t}}) \tag{3}$$

$$O_{\text{v}}=M_{\text{CLIP}}(F_{\text{v}}) \tag{4}$$

$$O_{\text{c}}=concat(O_t,O_{\text{v}}) \tag{5}$$

$$O_{\text{final}}=M_{\text{Fusion}}(O_{\text{c}}) \tag{6}$$

其中，F_{t} 和 F_{v} 分别表示从勒索软件中提取的文本特征和视觉特征；M_{BERT}，M_{CLIP} 和 M_{Fusion} 分别代表所提出方法中的 BERT 模型、CLIP 模型和融合模型；O_{t} 表示 BERT 的编码输出，O_{v} 表示 CLIP 的编码输出，O_{c} 为 O_{t} 和 O_{v} 拼接的结果；O_{final} 表示融合模块的最终输出。

4　实验验证

4.1　评价指标

由于 CMFu 用于分类问题，因此本文引入 3 种经典的评估指标来衡量该方法的性能，即精确率（Precision）、召回率（Recall）和 F1 值（F1-value）[31]。

在本文中，勒索软件分类是一个多分类问题。因此，对这些指标应用加权平均即可衡量模型效果，即 $Precision_w$，$Recall_w$ 和 $F1_w$。这些指标可以通过真正例（True Positive，TP）、真负例（True Negative，TN）、假正例（False Positive，FP）和假负例（False Negative，FN）来计算。

4.2　数据集描述

为了收集真实且完整的数据集，本文选择从恶意软件共享网站 VirusShare 获取样本。VirusShare 是一个恶意软件样本库，它为安全研究人员提供访问活跃恶意代码样本的权限[32]。本文从该网站通过 hash 码下载了 4 000 个恶意软件样本，全部是 PE 文件。

然而，并非所有样本都是勒索软件，还有其他类型的恶意软件，如蠕虫（Worm）、特洛伊木马（Trojan）和后门（Backdoor）。因此，本文通过一个在线恶意软件分析平台 VirusTotal 获取每个恶意软件样本的报告[33]。该网站提供了各种公开的恶意软件的分析报告，便于安全人员研究。根据报告，本文筛选出其中的勒索软件。

从 VirusTotal 收集到 4000 份恶意软件报告，描述了有关相应恶意软件的详细信息，包括其恶意软件类型和分类。因此，本文根据类别从 4000 个恶意软件中筛选出 488 个勒索软件样本，并根据报告中的分类对它们进行标记。样本的分类分布情况如表 1 所列。

表 1　样本类别的分布情况

Table 1　Classifications distribution of samples

类别	数量	类别	数量
pajetbin	298	palevo	11
mbrlock	27	vbclone	11
doina	22	tdss	10
allaple	16	lipler	10
trickbot	15	stopcrypt	10
bifrose	14	nymaim	10
renos	14	zbot	9
vobfus	11		

本文在实验部分首先展示了编码器对两种模态特征的提取效果，设计了两个实验。第一个实验采用一些对比模型与CMFu进行对比，其中包含处理文本和图像分类的经典模型，以及用于少样本情景下的模型从预训练模型中提取的编码器。第二个实验为消融实验，旨在展示CMFu的各个模块对少样本勒索软件分类的帮助。

4.3 编码效果

为了验证BERT和CLIP在微调后对勒索软件两种模态特征的提取能力，本文展示了它们提取后的编码结果，以证明相同类别下的编码序列具有相似性。具体而言，本文采用热力图呈现编码数据，通过颜色深浅直观反映不同样本在各特征维度上的差异。

1)BERT编码器

本文随机选取10个类别展示BERT编码器对API调用的编码效果，绘制其原始输出编码的热力图，如图7所示。由于原始输出编码维度较高，难以分析，因此本文同时使用主成分分析(Principal Component Analysis，PCA)对原始输出编码进行降维处理，进一步展示编码器对特征的编码效果，如图8所示。

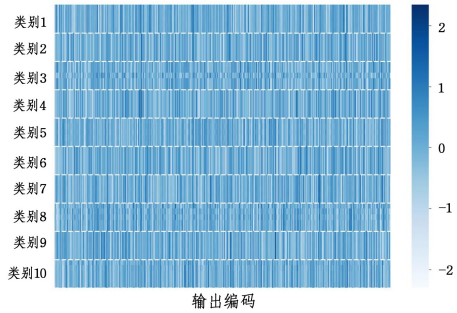

图7 BERT编码器原始输出编码热力图

Fig. 7 BERT encoder's original output encoding heatmap

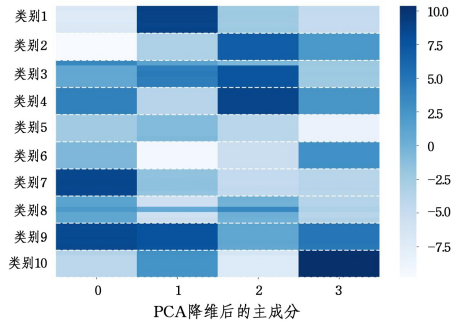

图8 BERT编码器编码主成分热力图

Fig. 8 BERT encoder's principal component heatmap

分析图8可知，该编码器对于API调用的编码效果较好，类别1、类别2、类别4等类别的主成分相似度都较高，有助于模型学习相关信息进行分类；并且所展示的编码器是基于原始数据集大小的50%的数据进行训练的，然后对测试集进行编码并展示，证明了其在少样本情况下仍然可以对勒索软件样本特征进行有效提取。此外，通过分析发现，BERT编码器仅在类别3和类别8上主成分相似度存在一定的偏差，分析原因可能是数据集较少导致表示效果降低，但总体的编码效果较好。

2)CLIP编码器

本文同样选取10个类别展示CLIP编码器对OPCODEs的编码效果，绘制其编码热力图，如图9所示。同样使用PCA对输出编码进行降维后绘制，展示编码器对特征的编码效果，如图10所示。

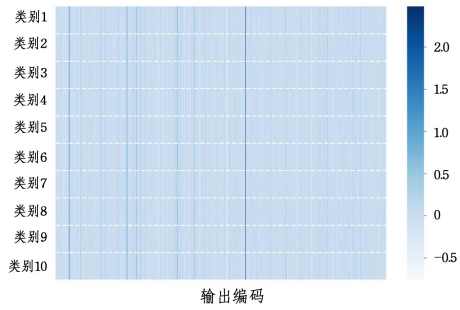

图9 CLIP编码器原始输出编码热力图

Fig. 9 CLIP encoder's original output encoding heatmap

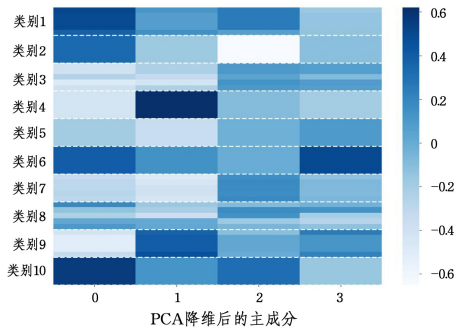

图10 CLIP编码器编码主成分热力图

Fig. 10 CLIP encoder's principal component heatmap

通过对图10进行分析可知，CLIP编码器对于OPCODEs的迁移效果同样较好，能通过主成分相似度证明其对勒索软件特征的编码能力。同样地，在类别8上各主成分相似度存在偏差，分析原因可能是模型对类别8的分类效果存在偏差，但本文在后续设计了融合模块，可以通过进一步的模型内部计算，整合两种不同的模态编码，改善分类表现。

4.4 模型对比

本文将收集到的勒索软件样本按比例划分为训练集和测试集。为了模拟少样本情况，分别选取原始数据集大小的10%，30%和50%作为训练集的比例。在这种小比例下，可以有效模拟少量样本训练集输入模型的情况，并体现模型在少样本学习下的性能。此外，为了避免随机选择训练集导致偏差，进而导致某些类别缺乏相应的训练数据，本文按等比例对每个类别进行随机抽样。每个类别的分布情况如表2所列。通过分析可以发现，当训练样本的比例为50%时，样本数量相对较少。然而，每个分类仍然可以为模型训练提供一定数量的样本。但是在这个比例下，zobt仅为模型提供了4个训练样本，十分考验模型的少样本学习能力。当比例下降到30%，甚至10%时，训练样本的数量变得异常少。具体来说，当划分比例为10%时，几乎所有的勒索软件类别都只包含1到2个样本，样本数量都非常少。在这种情况下，模型的

单样本学习能力面临巨大挑战。

在将数据集按不同比例划分后，本文首先将 CMFu 与常用基准模型进行对比评估。本文采用的基准对比模型有：支持向量机（Support Vector Machine，SVM），一种经典的机器学习模型；文本循环神经网络（Text Recurrent Neural Network，TextRNN），基于文本数据进行分类的强大模型；数据增强的卷积神经网络（Data Augmentation CNN，DACNN），结合数据增强技术以及图像处理中经典的 CNN 框架；残差网络（Residual Network，ResNet），同样是一种预训练模型、较深的图像网络。

此外，为了探讨单一模态是否影响了基准模型，本文还将不同模态的基准模型进行了组合，使其能使用多模态特征，并将其与 CMFu 进行对比，进一步说明 CMFu 的优越性。

CMFu 和基准模型在不同样本比例下的对比情况如表 3 所列。

表 2　不同比例下的样本分布

Table 2　Distribution of samples in different proportions

类别	比例/%		
	10	30	50
pajetbin	29	89	149
mbrlock	2	8	13
doina	2	6	11
allaple	1	4	8
trickbot	1	4	7
bifrose	1	4	7
renos	1	4	7
vobfus	1	3	5
palevo	1	3	5
vbclone	1	3	5
tdss	1	3	5
lipler	1	3	5
stopcrypt	1	3	5
nymaim	1	3	5
zbot	0	2	4

表 3　与基准模型的对比

Table 3　Comparison with baseline models

Model	Weighted Precision/%			Weighted Recall/%			Weighted F1-score/%		
	10	30	50	10	30	50	10	30	50
SVM	0.62	0.85	0.88	0.76	0.85	0.88	0.68	0.84	0.87
TextRNN	0.63	0.86	0.90	0.76	0.86	0.89	0.68	0.86	0.88
DACNN	0.40	0.62	0.69	0.61	0.68	0.76	0.48	0.64	0.72
ResNet	0.53	0.77	0.82	0.67	0.81	0.85	0.58	0.78	0.83
TextRNN＋DACNN	0.63	0.86	0.90	0.76	0.88	0.89	0.67	0.86	0.88
TextRNN＋ResNet	0.57	0.82	0.87	0.70	0.85	0.86	0.61	0.82	0.85
CMFu	**0.78**	**0.91**	**0.93**	**0.84**	**0.91**	**0.90**	**0.80**	**0.90**	**0.90**

通过分析表 3 可以发现，CMFu 的所有指标都高于其他模型，并且随着训练集比例的减少，CMFu 的优势愈发明显，普通的基准模型在训练集仅有 10%时，在精确率、召回率和 F1 值上最好的表现为 0.63，0.76 和 0.68，落后于 CMFu 的 0.78，0.84 和 0.80。这说明了 CMFu 在样本数不足时，仍然能保持较好的分类效果。此外，当样本比例降低时，CMFu 受到的影响是所有模型中最小的。当样本比例从 50%降为 30%时，CMFu 的精确率仅降低 0.02，召回率增加了 0.01，F1 值没有降低。当样本比例从 30%降为 10%时，其精确率仅降低 0.13，召回率降低 0.07，F1 值降低 0.10，为所有模型中降幅最小的。

TextRNN＋DACNN 和 TextRNN＋ResNet 两个基准模型的表现说明，简单组合两个模型并不能有效融合不同模态的信息，需要综合考虑模型内部对特征的编码情况，合理设计融合模块，才能有效利用多模态信息。CMFu 在 10%样本下比 TextRNN＋DACNN 和 TextRNN＋ResNet 的最佳指标分别高出 0.15，0.08 和 0.13。

本文在后续消融实验中进一步说明了 CMFu 的融合模块有助于分类效果。

除了与普通基准模型进行对比，本文还将 CMFu 与部分少样本学习模型进行对比，展示 CMFu 相比于其他少样本模型对于勒索软件分类的优越性。用于对比的第一个模型是预训练的 BERT[28]，借助预训练的知识库改善少样本学习；第二个是已被证明适用于少样本学习的原型网络（Prototypical Network，ProtoNet）[29]；第三个则是无关元学习方法（Model-Agnostic Meta-Learning，MAML），其作为一种元学习模型，可以用于快速适应新任务，适用于少样本学习[8]。

CMFu 和少样本学习模型在不同样本比例下的对比情况如表 4 所列。

表 4　与少样本学习模型的对比

Table 4　Comparison with few-shot learning models

Model	Weighted Precision/%			Weighted Recall/%			Weighted F1-score/%		
	10	30	50	10	30	50	10	30	50
BERT	0.75	0.75	0.89	0.82	0.80	0.89	0.77	0.77	0.88
ProtoNet	0.47	0.69	0.71	0.67	0.79	0.79	0.55	0.73	0.74
MAML	0.74	0.81	0.83	0.79	0.82	0.80	0.76	0.81	0.80
CMFu	**0.78**	**0.91**	**0.93**	**0.84**	**0.91**	**0.90**	**0.80**	**0.90**	**0.90**

通过观察表 4 可以发现，少样本学习模型在训练集比例为 50%和 30%时，大多逊于表 3 中的普通基准模型；当比例仅为 10%时，少样本学习模型的优势得到体现，大多优于基准模型。但 CMFu 在训练集比例仅为 10%时，表现依旧为所有模型中

最优的，比少样本学习模型的最优指标分别高出0.03，0.02 和 0.03，说明了 CMFu 在训练集极少的情况下依旧能够保持优秀的分类能力。具体对比 BERT 和 CMFu 可知，虽然 BERT 在 9 个指标上表现均较好，但 CMFu 结合了多模态的信息并进行了合理融合，表现均优于 BERT，尤其是训练集比例为 30% 时，3 个指标分别比 BERT 高出 0.16，0.11 和 0.13。分析原因可能是 BERT 利用的编码信息有限，而 CMFu 除了文本模态，还结合了图像模态，特征信息含量充分，表现更佳。此外，MAML 是一个优秀的少样本学习框架，虽然略逊于 CMFu，但在样本数极少时的优势得到了体现，然而其在 30% 和 50% 比例下的性能远落后于 CMFu，30% 下各指标比 CMFu 低 0.10，0.09 和 0.09，50% 下比 CMFu 低 0.10，0.09 和 0.10，说明其在样本数增加过程中得到的信息增益少于 CMFu。

本文进一步分析 CMFu 在不同样本比例下对各类勒索软件的分类效果，如表 5 所列，从而进一步分析导致本文方法产生漏报的具体勒索软件。

表 5 CMFu 对各类别的分类结果

Table 5 Classification result of CMFu on each category

Category	10%			30%			50%		
	Precision	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*
pajetbin	0.91	1.00	0.95	0.99	1.00	0.99	1.00	0.99	1.00
mbrlock	0.89	1.00	0.94	1.00	1.00	1.00	0.88	1.00	0.93
doina	0.73	0.95	0.83	1.00	0.94	0.97	0.85	1.00	0.92
allaple	0.88	0.93	0.90	0.92	1.00	0.96	1.00	1.00	1.00
trickbot	1.00	0.86	0.92	1.00	1.00	1.00	1.00	1.00	1.00
bifrose	0.00	0.00	0.00	0.31	0.40	0.35	1.00	0.14	0.25
renos	0.00	0.00	0.00	1.00	0.30	0.46	0.25	0.14	0.18
vobfus	0.50	0.10	0.17	0.62	0.62	0.62	0.33	0.67	0.44
palevo	0.14	0.10	0.12	0.44	0.88	0.58	0.67	0.67	0.67
vbclone	0.86	0.60	0.71	0.89	1.00	0.94	1.00	1.00	1.00
tdss	0.44	0.44	0.44	0.20	0.14	0.17	0.30	0.60	0.40
lipler	0.57	0.44	0.50	0.78	1.00	0.88	1.00	0.80	0.89
stopcrypt	0.41	1.00	0.58	1.00	1.00	1.00	1.00	1.00	1.00
nymaim	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
zbot	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.20	0.33

分析发现，当样本比例为 10% 时，CMFu 在 bifrose，renos，vobfus 和 palevo 类别（训练样本数均为 1）上表现较差，整体精确率、召回率和 F1 值分别为 0.78，0.84，0.80，原因主要在于样本稀缺导致模型泛化能力不足。将样本比例提升至 30% 后，上述类别的 F1-score 分别显著提升 35 个百分点、46 个百分点、45 个百分点和 55 个百分点。zbot 类别数量极少，模型的分类效果受限，当样本数增加为 4 时才有所提升。

随着样本数的增加，CMFu 在各类别上的分类性能呈现持续改善趋势，如 palevo，vbclone 和 lipler，即使它们的样本数依然少于 5。此外，随机选取训练样本可能导致训练样本代表性不足，限制了模型的泛化能力。

分析表明，制约 CMFu 性能提升的核心因素为：1）个别类别的样本数极少；2）样本代表性不足。本文将在后续工作中考虑如何通过已有的勒索软件类别对泛化能力不强的样本进行迁移。

4.5 消融实验

为了更深入地了解 CMFu，本文针对 CMFu 的结构，在少样本场景下进行了消融研究，分别将两种预训练模型与全连接层组合，以及在不使用融合模块的情况下，同 CMFu 进行比较。w-BERT 表示移除 BERT 编码器的 CMFu，w-CLIP 表示移除 CLIP 编码器的 CMFu，w-Fusion 表示没有融合模块的 CMFu。缺少不同模块的模型在不同样本比例下的表现如表 6 所列。

表 6 消融实验结果

Table 6 Ablation study results

Model	Weighted Precision/%			Weighted Recall/%			Weighted F1-score/%		
	10	30	50	10	30	50	10	30	50
w-BERT	**0.80**	0.86	0.89	0.83	0.88	0.89	0.79	0.86	0.88
w-CLIP	0.80	0.83	0.84	0.80	0.83	0.84	0.79	0.82	0.84
w-Fusion	0.78	0.88	0.89	0.83	0.89	0.89	0.79	0.88	0.89
CMFu	0.78	**0.91**	**0.93**	**0.84**	**0.91**	**0.90**	**0.80**	**0.90**	**0.90**

通过分析可以看出，各个模块有助于 CMFu 最终表现的提升，CMFu 在 9 个指标中有 8 个均为最佳，仅在 10% 训练样本比例下的精确率表现略差于 w-BERT 和 w-CLIP。其中，当缺少编码器时，模型的各个指标都显著降低。例如，w-CLIP 在 30% 样本训练集下的精确率为 0.83，相比于 CMFu 落后了 0.08。缺少融合模块对模型的表现影响较小，虽然可以通过预训练模型进行特征编码，但是无法通过融合模块的编码层进一步处理多模态信息，导致所有指标都落后于 CMFu。这些结果证明了融合模块对多模态信息集成的正向作用。三者的结合，能有效构建跨模态融合的少样本勒索软件分类器。

结束语 本文提出了一种跨模态融合少样本勒索软件分类器 CMFu，旨在更好地解决少样本场景下的勒索软件分类问题。该方法的框架由 3 个不同的模块组成：特征构建模块、

编码模块和融合模块。特征构建模块构建图像和文本模态特征，输入后续模块。编码模块引入预训练模型作为编码器，通过预训练模型的丰富知识库以及内部复杂的特征处理机制，对模块进行特征编码。融合模块将不同模态的编码进行整合，将自注意力和残差网络连接两个全连接层，输出最终的分类结果。

本文通过 PCA 降维展示编码效果，说明了基于预训练的编码器的特征表示能力。实验将 CMFu 与基准模型和少样本学习模型进行对比，与其他模型相比，CMFu 的 9 项指标均表现最优。当训练集占总数据量的 30%时，CMFu 比表现最好的少样本学习模型 MAML 在精确率上高出 10 个百分点，在召回率上高出 9 个百分点，在 F1 值上高出 9 个百分点。当训练集比例降至 10%时，CMFu 的优势得到了充分展现，指标分别为 0.78，0.84 和 0.80。此外，实验展示了 CMFu 对各勒索软件类别的分类结果，说明其在大部分类别仅有极少样本下，依旧能够保持分类效果，并且分析了制约 CMFu 性能提升的原因。将在后续工作中考虑如何通过已有的勒索软件类别对泛化能力不强的样本进行迁移。

实验证明了 CMFu 在少样本勒索软件分类中的优秀表现。通过消融实验，证明了基于预训练的编码器的可行性以及需要骨干网络进行融合的必要性。

参 考 文 献

[1] YAN S, REN J, WANG W, et al. A Survey of Adversarial Attack and Defense Methods for Malware Classification in Cyber Security [J]. IEEE Communications Surveys & Tutorials, 2023, 25(1): 467-496.

[2] SHAH N, FARIK M. Ransomware-Threats, Vulnerabilities and Recommendations [J]. International Journal of Scientific & Technology Research, 2017, 6: 307-309.

[3] The State of Ransomware 2023 [EB/OL]. (2023-05-10) [2025-06-24]. https://news. sophos. com/en-us/2023/05/10/the-state-of-ransomware-2023/.

[4] XUE D, LI J, LYU T, et al. Malware Classification Using Probability Scoring and Machine Learning [J]. IEEE Access, 2019, 7: 91641-91656.

[5] WASOYE S, STEVENS M, MORGAN C, et al. Ransomware Classification Using BTLS Algorithm and Machine Learning Approaches [EB/OL]. https://doi. org/10. 21203/rs. 3. rs-5131919/v1.

[6] ZHU J, JANG-JACCARD J, SINGH A, et al. A few-shot meta-learning based siamese neural network using entropy features for ransomware classification [J]. Computers & Security, 2022, 117: 102691.

[7] FINN C, ABBEEL P, LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks [C]// Proceedings of the 34th International Conference on Machine Learning. 2017: 1126-1135.

[8] JI Y, ZOU K, ZOU B. Mi-MAML: Classifying few-shot advanced malware using multi-improved model-agnostic meta-learning [J]. Cybersecurity, 2024, 7(1): 72.

[9] ZHAO W X, ZHOU K, LI J, et al. A Survey of Large Language Models [J]. arXiv: 2303. 18223, 2023.

[10] WU J, GAN W, CHEN Z, et al. Multimodal Large Language Models: A Survey [C]// 2023 IEEE International Conference on Big Data (BigData). IEEE, 2023: 15-18.

[11] ZHONG X, BAN H. Pre-trained network-based transfer learning: A small-sample machine learning approach to nuclear power plant classification problem [J]. Annals of Nuclear Energy, 2022, 175: 109201.

[12] CHEN X, LIU T, FOURNIER-VIGER P, et al. A fine-grained self-adapting prompt learning approach for few-shot learning with pre-trained language models [J]. Knowledge-Based Systems, 2024, 299: 111968.

[13] LIN Z, YU S, KUANG Z, et al. Multimodality helps unimodality: Cross-modal few-shot learning with multimodal models [C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 2023.

[14] CHRISTODORESCU M, JHA S. Static analysis of executables to detect malicious patterns [C]// Proceedings of the 12th Conference on USENIX Security Symposium. Washington, DC: USENIX Association, 2003.

[15] ABUSITTA A, LI M Q, FUNG B C M. Malware classification and composition analysis: A survey of recent developments [J]. Journal of Information Security and Applications, 2021, 59: 102828.

[16] FIRDAUSI I, LIM C, ERWIN A, et al. Analysis of Machine Learning Techniques Used in Behavior-Based Malware Detection [C]// 2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies. IEEE, 2010: 2-3.

[17] KOLOSNJAJI B, ZARRAS A, WEBSTER G, et al. Deep Learning for Classification of Malware System Call Sequences [C]// Australasian Joint Conference on Artificial Intelligence. Springer, 2016.

[18] SHARMA O, SHARMA A, KALIA A. MIGAN: GAN for facilitating malware image synthesis with improved malware classification on novel dataset [J]. Expert Systems with Applications, 2024, 241: 122678.

[19] DENG H, GUO C, SHEN G, et al. MCTVD: A malware classification method based on three-channel visualization and deep learning [J]. Computers & Security, 2023, 126: 103084.

[20] ABBASI M S, AL-SAHAF H, MANSOORI M, et al. Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection [J]. Applied Soft Computing, 2022, 121: 108744.

[21] AURANGZEB S, ANWAR H, NAEEM M A, et al. BigRC-EML: Big-data based ransomware classification using ensemble machine learning [J]. Cluster Computing, 2022, 25(5): 3405-3422.

[22] CHAGANTI R, RAVI V, PHAM T D. Image-based malware representation approach with EfficientNet convolutional neural networks for effective malware classification [J]. Journal of Information Security and Applications, 2022, 69: 103306.

[23] NI S, QIAN Q, ZHANG R. Malware identification using visualization images and deep learning [J]. Computers & Security,

2018,77:871-885.
[24] CONTI M,KHANDHAR S,VINOD P. A few-shot malware classification approach for unknown family recognition using malware feature visualization [J]. Computers & Security,2022,122:102887.
[25] PARISOT A,BENTO L M S,MACHADO R C S. Ransomware Detection: Leveraging Sandbox, Text Mining Techniques and Machine Learning [C]// 2024 IEEE International Workshop on Metrology for Industry 40 & IoT(MetroInd40 & IoT). IEEE,2024:29-31.
[26] ZHOU Y,LIU Z,XUE J,et al. LM-cAPI: A Lite Model Based on API Core Semantic Information for Malware Classification [C]// International Conference on Applied Cryptography and Network Security. Springer,2024.
[27] LISA F T,ISLAM S R,KUMAR N M. Multi-modal machine learning model for interpretable malware classification [C]// World Conference on Explainable Artificial Intelligence. Springer,2024.
[28] LIAOW,LIU Z,DAI H,et al. Mask-guided BERT for few-shot text classification [J]. Neurocomputing,2024,610:128576.
[29] LI H,CHEN S,WANG G,et al. Enhancing Few-Shot Malware Classification Through Joint Learning of Malware Images and Opcode Sequences [C]// 2024 IEEE International Symposium on Parallel and Distributed Processing with Applications(ISPA). IEEE,2024.
[30] SRIVASTAVA S,SHARMA G. Omnivec: Learning robust representations with cross-modal sharing [C]// Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. IEEE,2024.
[31] YACOUBY R,AXMAN D. Probabilistic extension of precision, recall,and f1 score for more thorough evaluation of classification models [C]// Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems. IEEE,2020.
[32] VirusShare. com [EB/OL]. (2025-06-25) [2025-06-25]. https://virusshare. com/research.
[33] VirusTotal [EB/OL]. (2025-06-25) [2025-06-25]. https://www. virustotal. com/gui/home/upload.

YIN Chuang, born in 2002, postgraduate. His main research interests include deep learning and cyber security.

LIU Jianyi, professor, Ph.D supervisor, is a member of CCF(No. 17814M). His main research interests include digital content security and data mining.

(责任编辑:何杨)