2022年第三届模式识别与机器学习国际会议（PRML）| 978-1-6654-9950-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/PRML56267.2022.9882232

基于文本转化器模型的APT恶意样本组织可追溯性

李静
国家电网信息通信分公司
北京，中国
gaokenan0616@163.com

高雅婷
国家电网信息通信分公司
北京，中国
gsn0647@163.com

史晨
北京邮电大学网络空间安全学院
北京，中国
finn_cs@bupt.edu.cn

刘建一
北京邮电大学网络空间安全学院
北京，中国
liujy@bupt.edu.cn

苏蓓蓓
国家电网信息通信分公司
北京，中国
758614897@qq.com

张静
国家电网信息通信分公司
北京，中国
1066016493@qq.com

摘要--
***传统的*恶意代码识别软件主要依赖于程序的静态签名，通过映射文件将恶意样本与特定签名进行匹配。目前，APT攻击组织的溯源主要依靠人工分析样本，存在着自动分析不足、难以通过单一特征进行准确溯源等困难。为解决上述问题，本文提出了基于文本转化器模型的APT恶意样本组织的溯源方法。以APT-28为首的6个组织样本的分类准确率达到93.24%，可以有效应对APT攻击的安全威胁。该方法的文本特征提取结构融合了卷积结构和Transformer模型，可以灵活处理各种恶意代码序列长度。与其他基于文本序列的恶意代码分类方法相比，本模型的准确性提高了1.5%。**

关键词--恶意代码，转化器模型，组织可追溯性

I. 简介

恶意样本的识别和家族分类是网络安全领域的一个重要研究领域。除了检测待测样本是否为恶意软件外，检测其家族分类也很重要。恶意软件的家族分类通常可以表明恶意行为类别和恶意软件的执行目的。这些恶意目的的分类也为恶意代码的危险程度等信息提供了重要参考。对恶意代码家族分类的检测，有利于快速跟踪恶意代码家族的发展，从而快速评估网络空间安全形势。

基于文本序列的模型通常依靠递归神经网络来分析恶意代码序列，在时间序列传输过程中存在一定的信息损失问题。对于恶意家族分类模型，有单一特征，因为极少数恶意样本可能偏离主族的恶意样本，导致误判。鉴于上述问题，本文主要开展了以下工作。

- 针对目前基于文本的分析方法不能充分利用恶意代码序列的问题，提出了一种与一维卷积结构相结合的文本转化器模型，提高了模型的扩展能力和浅层特征的学习能力。该模型在两个不同的数据集上进行了测试，以验证该模型提取恶意代码的文本特征的能力。
- 针对APT组织恶意样本难以识别的问题，提出了一种基于文本特征的Transformer模型[1]的恶意样本组织分类框架。利用APT组织[2]样本的文本特征进行分类，可以有效判断不同APT组织的样本。

II. 相关工作

传统的恶意样本分析技术主要依靠静态分析技术和动态分析技术，在人工专业知识的辅助下对样本进行研究和判断。常见的静态分析方法需要从样本中提取字符串信息、[3]操作码特征[4][5]、熵结构[6]等综合特征，并利用其中一个或多个特征生成样本的特征指纹来识别恶意样本。

动态分析通常在受控环境中运行恶意样本，并在恶意样本执行期间收集API调用信息[7][8]、网络通信信息[9]和内存活动[10]。对于一些使用混淆技术来逃避分析的样本，动态分析可以更准确地反映其行为。随着恶意代码攻击的不断发展和

防御技术，一旦检测到许多恶意样本在虚拟机环境中运行，它们可以选择不进行恶意行为来逃避分析，或者通过虚拟机逃逸反向控制主机，这使得运行分析的过程存在一定的安全风险。

近年来，以机器学习和深度学习为代表的分析技术已经满足了分析大量恶意代码样本的需要。随着计算机并行计算能力的发展，研究人员将目光转向了不需要先验知识的深度学习技术。2009年，Santos等人[11].提出了一种基于N-gram opcode签名的恶意代码检测方法来检测未知软件。许多基于恶意软件行为的方法使用RNN模型来分析执行期间调用的API序列[12-14]。然而，目前的恶意软件可以通过检测是否在虚拟机环境中避免执行恶意的API，这些基于动态行为的方法在获取特征方面有一些局限性。由于操作码包含特定的语义恶意代码执行的信息，对操作码进行语义嵌入，然后进行分类[15][16]。在文献[17]中，恶意代码汇编文件的API函数名称和对应的操作码组合被作为语义信息特征，并采用LSTM网络进行分类。基于循环神经网络的方法在特征处理上具有较高的复杂性，在对长序列建模的过程中也会出现信息丢失的问题。目前，在技术上仍有很大的改进空间。这些方法在训练语义嵌入时过于耗时，而且很难对恶意代码操作码的全局语义信息进行建模。

III. 从APT样本文本中提取特征

为了解决传统Transformer模型计算成本高的问题，本文提出在Transformer模型中加入文本一维卷积模型。恶意代码文本特征提取模型如图1所示。

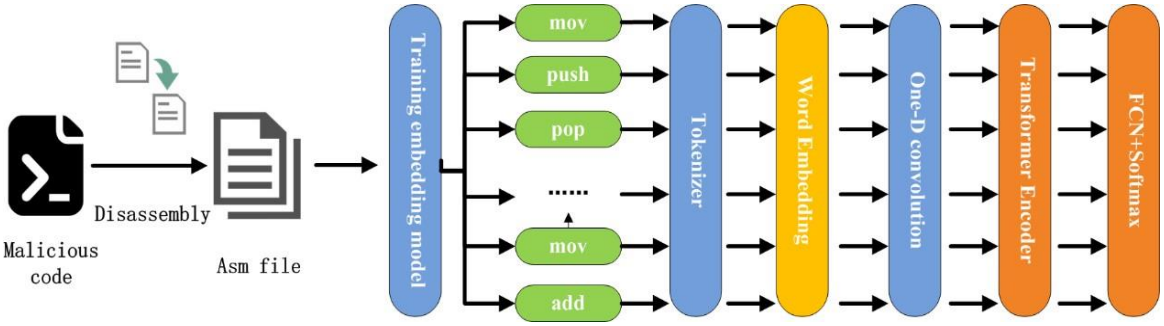


图1.基于文本转换器的恶意代码特征提取框架

A. 文本特征的预处理

由于恶意代码本身是一个PE可执行文件，因此可以使用IDA Pro工具进行反汇编，获得样本的.asm格式文件。该文件的每一行都可能包含章节名称、操作码指令、函数起始位置和操作数等。如图2所示。

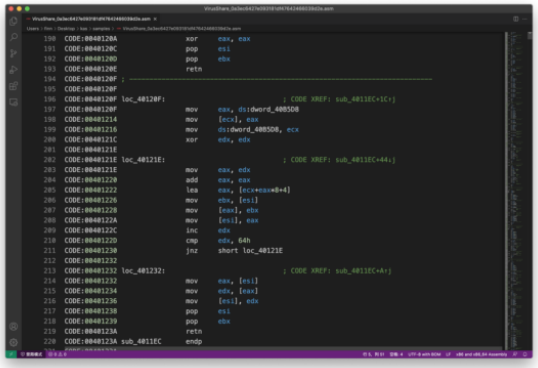


图2.恶意代码通过IDA专业工具反汇编的asm文件

在获得与恶意代码样本相对应的asm文件后，系统识别每一行的文本内容并提取操作码。在提取了操作码后，TF-IDF算法被用于去噪，TF值小于10的操作码被删除。Word2Vec[18]模型被用来学习操作码上下文之间的潜在语义关系以进行语义嵌入。为了防止语义嵌入的稀疏向量空间，本文将Word2Vec模型的嵌入维度调整为100。

B. 文本特征提取模型的构建

文本特征提取模型主要分为嵌入层和特征提取层。文本特征提取模型的处理流程图如图3所示。

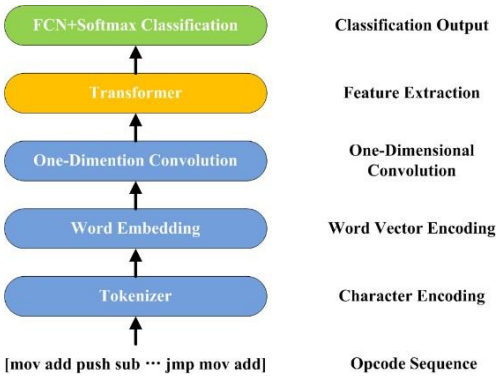


图3.文本特征提取模型构建框架

经过预处理，我们得到数据集样本的所有操作码序列和Word2Vec语义嵌入模型，并得到将文本字符映射为数字的Tokenizer。

在训练过程中，恶意代码样本的操作码序列中的前N个操作码被读取作为输入。经过嵌入层的语义嵌入，得到一个[N*D]维的向量矩阵，每一行代表样本中的一个操作码符号。为了使模型能够学习不同操作码之间的位置关系，本文将每个操作码对应的位置嵌入到语义向量中。

受Text-CNN模型的启发，本文对嵌入的文本序列进行了一维卷积。与原来的Transformer模型相比，本文提出的改进有以下两个优点：（1）该模型可以学习较长的文本序列，解决了文本Transformer模型对长序列的建模问题。(2）模型通过卷积结构提取序列中的浅层语义信息，然后应用注意机制学习不同符号之间的语义关联，提高了模型的学习效果。

嵌入完成后，序列矩阵为发送给Transformer模型进行特征信息提取得到一个维数为K的特征向量。矢量被发送到全连接层神经网络，得到一个长度为$Dfamily$的矢量。通过Softmax计算矢量后，可以得到样本属于每个恶意代码家族的概率。

IV. 实验结果和分析

A. 文本特征提取分类实验

为了获得最佳的文本特征提取结构，本节对恶意代码文本特征提取方法进行了消融实验。从文本序列的长度、编码器的深度和隐藏层单元的数量，探讨了不同参数对模型准确性的影响。

1) 文本特征提取实验数据集

为了确保模型的准确性和泛化能力，我们在两个不同的数据集上进行了实验。所有的实验都按4：1的比例分为训练集和验证集。

数据集1来自于VirusShare上抓取的数据。该数据集包括2019年至2020年间PE格式的恶意代码样本二进制文件，共包括9325个恶意样本。数据集1中的恶意代码家族分布如表I所示。

表一. 数据集1 恶意代码家族分布

恶意代码家族	恶意代码的数量	恶意代码定义
病毒	231	病毒
木马-间谍	158	间谍软件
木马-赎金	116	勒索软件
木马-PSW	2211	密码偷窃软件
木马-代理	45	Trojan代理
木马-游戏贼	579	游戏账号盗取软件
木马-滴答	325	恶意程序安装程序
木马-银行家	1491	银行业系统窃取软件
后门	303	后门
木马-假AV	62	假的反病毒程序
已打包	202	压缩的恶意软件
网虫	130	网络蠕虫
P2P-Worm	1937	P2P网络蠕虫
电子邮件蠕虫	921	邮件蠕虫
木马-下载器	646	恶意软件下载器

数据集2来自微软2015年在Kaggle机器学习竞赛平台上发布的数据。该数据集共包含9个恶意代码家族的10868个样本。与数据集1相比，数据集2对于恶意代码家族的具体类型更加详细，包括同一行为的恶意代码下的不同子种家族。数据集1中恶意代码家族的分布如表二所示。

表二.恶意代码家族的分布
数据集2

恶意代码家族	恶意代码的数量	恶意代码定义
拉姆尼特	1541	Virus程序
リリポープ	2478	恶意广告程序
Kelihos_ver3	2949	僵尸网络
文道	475	特洛伊木马程序
欣达	42	殭尸网络
Kelihos_ver1	1228	僵尸网络
门克	398	特洛伊木马程序
Obfuscator.ACY	1228	Obfuscated木马
袚芯禊芯写褋禈胁芯胁邪薪薪禈泄		751 窃听软件

2) 消融实验

a) 恶意代码文本序列的最佳长度实验

恶意代码的操作码是一个文本序列，不同的操作码长度对模型有重要影响。序列长度过短只能保留恶意代码的头部文本信息，而序列长度过长不仅会引入噪声，还可能造成巨大的计算开销。因此，恶意代码的序列长度对文本分类具有重要意义。实验结果如图4所示。

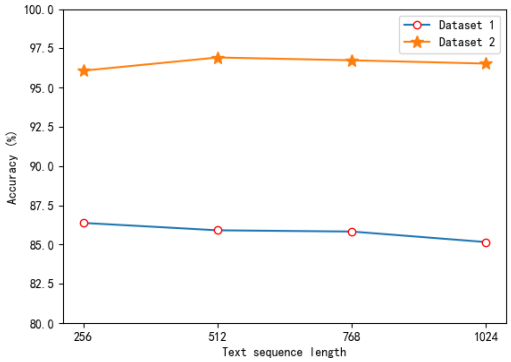


图4.恶意代码文本序列长度实验

经过十倍的交叉检查以消除错误，数据集1和2的最高准确率都出现在序列长度为384的实验中。数据集1的最高分类精度为86.38%，数据集2的精度为96.92%。模型的准确性和输入序列的长度之间没有线性增长模式。使用卷积结构对嵌入的文本序列进行预处理，该模型可以使用较少的文本学习更多的内容表示。

b) 恶意代码文本分类编码器层实验

除了模型输入的宽度外，模型的深度对分类结果的准确性也有至关重要的影响。我们分别在6层、4层和2层的编码器深度上训练了该模型。该模型在测试集上得到的结果如图5所示。

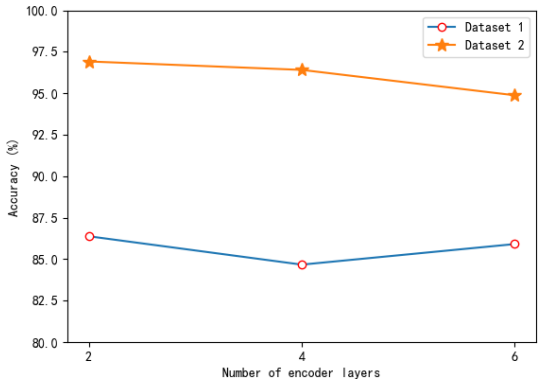


图5.编码器层对比实验

文本转化器模型在两个数据集上的准确性与网络层的数量呈负相关。因此，浅层编码器结构可以满足模型的分类需要。从实验结果可以看出，当只使用两层编码器结构时，模型在数据集1和数据集2上的分类精度分别达到86.38%和96.92%。

c) 隐层单元数量的对比性实验

除了词向量的嵌入维度外，本文还对三个不同的隐藏层维度进行了实验。该模型在测试集上得到的结果是如图6所示。

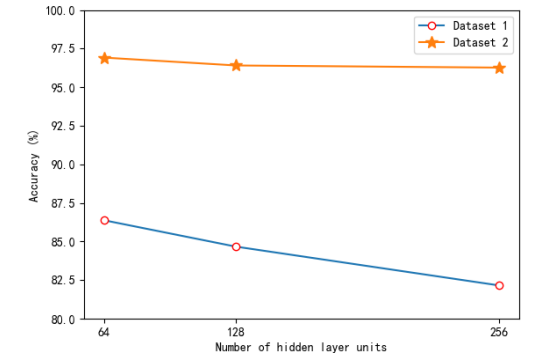


图6.关于隐藏层单元数量的实验

从实验结果可以看出，当隐藏层维度为64时，该模型在两个数据集上都取得了最好的分类精度，分别为86.38%（数据集1）和96.92%（数据集2）。综合来看，在两个数据集上，模型的分类准确率随着隐藏层维度的增加而缓慢下降。由于输入符号的嵌入维度为100，当模型的隐藏层维度为256时，模型在两个数据集上的最佳性能就会下降。可以认为，对于文本特征的Transformer模型来说，当模型的隐藏层维数与输入向量的维数相近或稍加压缩时，模型的分类效果较好。

3) 文本特征提取模型比较实验

为了证明第3.2节中提出的方法的有效性，我们将我们的方法与其他基于恶意代码文本序列的方法进行比较。所有的方法都是基于数据集2，即微软的恶意代码数据集。该比较结果见表三。

表 III. 方法与其他文本特征方法的比较

方法	序列长度	准确度	精度	召回率	F1
第3.2节中的方法	256	96.92%	96.70%	95.87%	0.9628
Cakir等人[19]。	所有序列	95.5%	95.7%	95.5%	0.956
Sung et等[20]。	所有序列	94.2%	95.0%	94.7%	0.946

B. APT 恶意样本组织可追溯性实验

1) APT数据集

本文根据公开的APT活动情报，收集了一些APT攻击样本，并选择了6个样本数量较多的APT组织进行了恶意代码分类实验。经过预处理，共获得1561个有效的APT样本。

表四显示了特定APT攻击家族的样本数量分布。

表IV. 1 APT数据集的样本分布

APT组织	样本数量	描述
APT-28	163个被怀疑的	俄罗斯黑客组织APT-29
	263个可疑的	俄罗斯黑客组织黑暗酒店267
		疑似北韩或南韩韩国黑客组织
活力熊	132疑似俄罗斯黑客组织	
方程式集团	396	被怀疑的美国黑客组织
高尔基集团	346被怀疑的巴基斯坦黑客组织	

2) 实验设置

在本节中，设计了一个基于文本特征的APT组织恶意代码分类模型。实验中提取了APT样本在256和384两个序列长度下的操作码序列，并使用两层的

编码器结构来提取特征。文本特征模型的实验参数设置见表五。

表V. T F M E P
实 验 参数的特点和模型设置

实验参数参数值	
学习率	0.001
辍学	10%
输入图像尺寸	(256, 256)
模型优化器	亚当优化器
批量大小	128

此外，在实验过程中，根据十倍交叉验证的结果对所有指标进行了平均。本文还通过混淆矩阵深入分析了该模型对不同恶意代码家族的分类结果。

C. 评价指标

本文采用准确性、精确性、召回率和F1值等指标进行实验评价。其计算公式如下。

- $Accuracy$。正确预测的样本占总样本数的比率。

$$Accuracy = \frac{TP+TN}{TP+TN+F+P+FN}$$

- $Precision$。真正的恶意样本数量与被判断为恶意的样本数量的比率。

$$Precision = \frac{T\ P}{TP+FP}$$

- $Recall$。恶意代码在样本中的比例被正确预测。

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

- $F1\text{-}Value$：用于综合评估精度和召回率。

$$F1 = \frac{2*Reacll*Prec}{Recall+Prec}$$

其中，TP代表样本被正确预测为相应正标的概率，TN代表样本被正确预测为负标的概率，FP代表样本被错误预测为正标的概率，F代表样本被错误预测为负标的概率。.

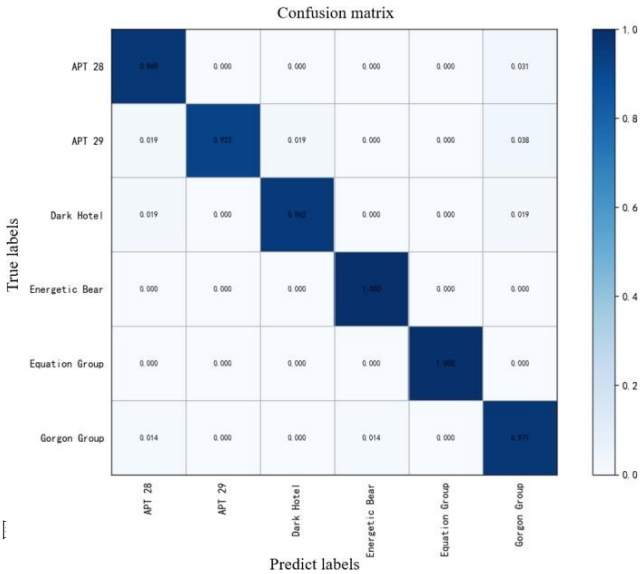
D. 实验结果和分析

实验结果见表六。

表六. 分类结果

特点	准确度	精度	召回率	F1
文本特点	90.67%	90.42%	90.60%	0.9050

图7显示了用于分类的混淆矩阵。每个APT样本的文本特征模型。



从实验结果来看，该模型对方程组的判别能力最高。APT组织被怀疑起源于美国。并且在地理信息上与其他APT组织存在一定的差异。这表明，该组织的APT攻击样本与在内部操作码行为方面，其他组织的恶意样本。从混淆矩阵中可以看出，该模型对APT 28组织的分类精度和黑暗酒店比平均水平低。在所有来自APT 28的恶意样本中，15.6%被错误地报告为APT 29组。而7.7%的 "能量熊 "组织的样本被错误地报告为来自APT 28组织。从公开的情报信息来看，APT28、APT29和能量熊组织被怀疑活跃在俄罗斯。在地理信息方面有一定的相似性，可能存在着相互的次生现象。发展。此外，11.3%的黑暗酒店集团的恶意代码样本被错误地报告为APT 29集团。虽然这两个团体在地理上没有直接联系，但这两个团体之间存在一定的误报，因为Dark Hotel可能在其开发项目中使用了俄罗斯的恶意攻击工具。

V. 总结

针对目前基于文本的分析方法对恶意代码序列利用不足的问题，本文提出了一种与一维卷积结构相结合的文本转化器模型，该模型改善了模型的

扩展能力和浅层特征学习能力。该模型在两个不同的数据集上进行了测试，以验证该模型提取恶意代码文本特征的能力。虽然本文提出的文本特征模型使用的是短的操作码序列，但它只使用不同恶意样本之间的顶级操作码信息。如何自动找到恶意样本操作码之间判别能力最强的文本序列，仍然是一个有待解决的问题。

针对识别APT组织恶意样本的困难，本文提出了一个基于Transformer模型的恶意样本组织分类框架，该模型是基于文本特征的。该模型可以有效地确定来自不同APT组织的样本。本文提出的APT恶意代码组织溯源模型是通过文本特征来分析APT样本。但如何利用这些特征信息来呈现不同APT样本的内部分布差异，是模型的可解释性和通用性的问题。

鸣谢

作者感谢匿名审稿人提出的宝贵意见和有益建议。该工作得到了国家电网信息通信分公司科技项目"面向多场景的安全保护框架研究与技术"（52993920002J）的支持。

参考文献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J].Advances in neural information processing systems, 2017, 30.

[2] Virvilis N, Gritzalis D. The big four-what we did wrong in advanced persistent threat detection? [C]//2013 international conference on availability, reliability and security.IEE, 2013: 248-254.

[3] Alhanahnah M, Lin Q, Yan Q, et al. Efficient signature generation for classifying cross-architecture IoT malware[C]//2018 IEEE Conference on Communications and Network Security（CNS）.IEEE, 2018:1-9.

[4] Abou-Assaleh T, Cercone N, Keselj V, et al. N-gram-based detection of new malware[C]//Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004.COMPSAC 2004。IEE, 2004, 2: 41-42.

[5] Darabian H, Dehghantanha A, Hashemi S, et al. An opcode-based technique for polymorphic Internet of Things malware detection[J].Concurrency and Computation:Practice and Experience, 2020, 32(6): e5173.

[6] Ren Z J, Chen G. 熵值可视化方法在恶意代码中的应用 代码分类中的应用 [J]. Computer engineering,2017,43(09):167-171.

[7] Wu D J, Mao C H, Wei T E, et al. Droidmat:通过清单和api调用追踪进行Android恶意软件检测[C]/2012第七届亚洲信息安全联合会议。IEEE, 2012:62-69.

[8] Zhang T, Wang J F. Family classification of malware based on text embedded feature representation［J］.四川大学学报（自然科学版）,2019,56（03）: 441-449.

[9] Bendiab G, Shiaeles S, Alruban A, et al. IoT malware network traffic classification using visual representation and deep learning[C]//2020 6th IEEE Conference on Network Softwarization（NetSoft）.IEE, 2020: 444-449.

[10] Nissim N, Lahav O, Cohen A, et al. Volatile memory analysis using the MinHash method for efficient and secured detection of malware in private cloud[J].Computers & Security, 2019, 87: 101590.

[11] Santos I, Penya Y K, Devesa J, et al. 基于N-grams的文件签名的恶意软件检测[J].iceis (2), 2009, 9: 317-320.

[12] X.Wang, X. Ma, W. E. L. Grimson, "Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models", TPAMI, Vol. 31, no.3, pp. 539-555, 2009.

[13] C.Shao和D. Liu，"基于分层语义认知的恶意软件检测"，2012年计算机科学与服务系统国际会议，南京，2012，第1615-1618页。

[14] 李晓勇和刘伟伟，"用综合行为分析检测恶意软件"，2009年机器学习和网络学国际会议，河北，2009，第2797-2801页。

[15] Zhang B, Xiao W, Xiao X, et al. Ransomware classification using patch-based CNN and self-attention network on embedded N-grams of opcodes[J].未来一代计算机系统, 2020, 110: 708-720.

[16] Darem A, Abawajy J, Makkar A, et al. Visualization and deep-learning-based malware variant detection using OpCode-level features[J].Future Generation Computer Systems, 2021, 125: 314-323.

[17] Kang J, Jang S, Li S, et al. 基于长短期记忆的信息安全恶意软件分类方法[J].Computers & Electrical Engineering, 2019, 77: 366-375.

[18] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv: 1301.3781, 2013.

[19] Cakir B, Dogdu E. Malware classification using deep learning methods[C]Proceedings of ACMSE 2018 Conference.2018:1-5.

[20] Sung Y, Jang S, Jeong Y S, et al. 使用先进的基于Word2vec的Bi-LSTM进行地面控制的恶意软件分类算法