Computer Science and Application 计算机科学与应用, 2023, 13(9), 1785-1795
Published Online September 2023 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2023.139177

Hans汉斯

基于PCA和SE-ResNet-VIT的恶意软件检测方法

凡　聪[1]，张　杰[2]

[1]广东工业大学计算机学院，广东　广州
[2]广东工业大学自动化学院，广东　广州

收稿日期：2023年3月14日；录用日期：2023年9月21日；发布日期：2023年9月28日

摘　要

近年来，恶意软件的数量不断增加，为用户带来了严重的安全隐患。为了避免主机系统受到恶意软件的侵害，提高检测的准确率，提出一种基于主成分分析(Principal component analysis, PCA)降维和SE-ResNet-VIT集成模型的恶意软件检测方法。由于软件数据信息具有高维度，多噪点的特征，通过PCA对待检测软件数据进行主成分提取，去除样本数据中的冗余特征项。SE-ResNet-VIT模型是将改进为双线性融合机制的SE-ResNet和VIT (Vision Transformer)中的编码器相结合的集成模型。改进的SE-ResNet模型能够从局部特征中提取更多信息，并通过组合这些特征来提高模型的表示能力。VIT模型能够通过注意力机制来学习数据之间的依赖关系，并能够处理长序列数据。该方法通过结合SE-ResNet和VIT，以两种不同的方式提取特征，能够更准确地捕捉软件的语义信息，从而提高恶意软件检测的准确性。在Ember数据集上进行了对比实验，实验结果表明，该方法的准确率分别为97.05%和98.45%，并与现有的多种检测方法进行对比，在准确率方面分别提高1.94%~5.95%，该方法有更好的检测准确率和泛化能力。

关键词

恶意软件检测，主成分分析，SE-ResNet，Vision Transformer，集成模型

PCA and SE-ResNet-VIT Based Malware Detection Method

Cong Fang[1], Jie Zhang[2]

[1]School of Computer, Guangdong University of Technology, Guangzhou Guangdong
[2]School of Automation, Guangdong University of Technology, Guangzhou Guangdong

Received: Mar. 14th, 2023; accepted: Sep. 21st, 2023; published: Sep. 28th, 2023

文章引用: 凡聪, 张杰. 基于PCA和SE-ResNet-VIT的恶意软件检测方法[J]. 计算机科学与应用, 2023, 13(9): 1785-1795.
DOI: 10.12677/csa.2023.139177

Abstract

As the digital age continues to advance, so does the threat of malicious software, commonly known as malware. In recent years, the number of malware attacks has skyrocketed, putting users' information and systems at risk. To mitigate these security concerns, researchers have developed a novel malware detection method that leverages the power of Principal Component Analysis (PCA) downscaling and an integrated model combining SE-ResNet and VIT (Vision Transformer). The SE-ResNet model, enhanced with a bilinear fusion mechanism, excels at extracting local features and improving the representation capability of the model. Meanwhile, the VIT model, with its attention mechanism, is able to learn inter-data dependencies and process long sequences of data. By combining these two models, the proposed approach is able to accurately capture the semantic information of software, leading to an improvement in malware detection accuracy. To demonstrate its effectiveness, the proposed method was tested against the Ember datasets, yielding an impressive accuracy of 97.05% and 98.45% respectively. The results of these experiments clearly indicate that this novel approach outperforms existing methods, with an improvement in accuracy ranging from 1.94% to 5.95%. In conclusion, the proposed malware detection method based on PCA downscaling and the integrated SE-ResNet-VIT model offers a cutting-edge solution to the growing problem of malware attacks. With its ability to accurately capture semantic information and improve detection accuracy, this method is poised to be a critical tool in safeguarding against malicious software.

Keywords

Malware Detection, Principal Component Analysis, SE-ResNet, Vision Transformer, Ensemble Mode

1. 引言

在过去的十年里，互联网实现了飞速的发展，但与此同时，恶意软件及其应用程序也井喷式的增长。恶意软件作者通常会诱导用户下载恶意文件，其中恶意文件包括病毒、特洛伊木马、蠕虫、rootkit、广告软件、勒索软件和一系列恶意的可执行程序。恶意软件会有目的性获取用户信息、窃取用户财产、破坏用户体验和锁定用户主机。根据 McAfee2021 年度报告，相比 2020 年，2021 年的恶意软件开发量大幅增加，平均每分钟会新增 588 个恶意软件[1]。国家互联网应急中心 2021 年上半年，捕获恶意程序样本数量约 2307 万个，日均传播次数达 582 万余次，涉及恶意程序家族约 20.8 万个[2]。对此，为了有效保障人民的财产和数据信息安全，有效防止恶意软件入侵，所以长期以来，对于恶意软件检测的研究一直以来都是网络安全领域中的热点之一。

恶意软件检测有两种基本的分析检测方法：静态分析检测和动态分析检测。静态分析检测是在不执行恶意软件程序的情况下对其进行反编译，并对哈希值、操作码、N-gram、PE (Portable Executable)头信息和字符串等信息进行特征提取，建模分类。而动态分析检测是通过创建虚拟环境执行恶意软件程序并捕获 API 调用，注册表项更改、新日志条目和网络活动等信息进行特征提取，建模分类。本文是基于静态分析检测方法对反编译后的数据进行降维处理，训练分类。

目前，机器学习中的许多网络模型已经被应用到恶意软件检测中。引入机器学习模型的检测方法与传统需要人工标注的检测方法在效率上大大提升，且减少了人力成本。文献[3]提出了通过主成分分析对恶意软件数据进行降维，并使用支持向量机(Support Vector Machine, SVM)算法对降维后的数据进行分类，提高了恶意软件检测效率，但 SVM 相较于深度学习中的网络框架，存在着特征提取不充分的问题，从而导致检测准确率较低。文献[4]提出了一种由卷积神经网络(Convolution Neutral Network, CNN)进行特征提取，决策树进行分类的端到端的检测框架，并提出了一种特殊的损失函数。但 CNN 和决策树结合的网络模型不能精确地关注到恶意软件数据中的重要特征，从而导致检测的准确率不高。文献[5]提出了一种异步学习框架，通过结合 API、函数调用和动态链接库的使用频次来作为训练数据，并通过贪婪分成来提高模型的泛化效果。文献[6]将恶意软件转换为 RGB 图像，并提出了一种动态路由的胶囊网络框架用于恶意软件图像的分类。文献[7]提出了一种改进的 ResNeXt 模型，通过嵌入一种新的正则化技术来限制模型拟合能力，减缓收敛速度，从而改进分类任务。文献[8]改进了现有深度神经网络(Deep Neural Network, DNN)模型，通过引入定向 Dropout 正则化方法，提高了模型的迁移性。文献[9]结合了机器学习和深度学习的优势，引入了一种具有多流输入的新型 DeepMalware 架构，用于解决现有关键安全设备上恶意软件检测的性能开销问题。文献[10]通过反编译物联网设备平台恶意软件获取样本静态指令序列，使用 Word2Vec 将词频转换为向量，再通过构建一个多层的 Transformer 模型学习恶意软件中结构化序列和函数的语义表示，利用 Transformer 模型中的并行的多头注意力机制对输入特征进行自适应加权，从而能够关注到重要的特征，该方法充分考虑到恶意软件数据特征各项的内部相关性。但 Windows 平台的恶意软件特征维度较大，且有大量冗余数据，实验结果表明，在 Ember 数据集上，先对数据进行降维，再使用 Transformer 模型进行分类的检测准确率会比不使用 PCA 降维的检测准确率高[11]。而 VIT 是 Transformer 的改进版本，跟适用于图像分类领域，虽然 VIT 模型有着众多的优点，但在局部感知方面不及 SE-ResNet，且单一在检测精度方面有所欠缺。

综合上述恶意软件检测方法中存在的问题，本文提出一种基于 PCA 和 SE-ResNet-VIT 的恶意软件检测方法。将原始的软件数据转换为浮点型数据，再对高维的软件数据进行 PAC 降维处理，充分提取数据中的有效主成分，去除数据中的噪声和冗余，再转换成灰度图像输入到 SE-ResNet-VIT 集成模型中进行检测分类。并在 Ember 数据集与现有的多种检测方法进行了对比实验，通过实验结果验证该方法的有效性。

2. 基于 PCA 和 SE-ResNet-VIT 的恶意软件检测方法设计

2.1. 整体结构

本文提出基于 PCA 和 SE-ResNet-VIT 的恶意软件检测模型如图 1 本文网络结构模型图所示，结构包括将原始 PE 文件转换为浮点型数据、PCA 主成分提取降维、最大最小值归一化、改进的 SE-ResNet、VIT、对两个模型的结果加权重计算和输出检测结果。

集成模型是一种提高模型准确率的方法，模型之间的差异越大，集成后的性能改善越大[12]。两种不同的特征提取方法提取软件图像的特征，可以更全面地区分图像之间的差异，并获得更好的分类结果。SE-ResNet-VIT 集成模型结合了两种完全不同的特征提取方法，即改进的 SE-ResNet 模型和 VIT 模型以提取软件图像信息的特征并进行分类。将改进的 SE-ResNet 模型的输出结果乘以系数 0.7，VIT 模型的输出值乘以系数 0.3，然后将这两个结果相加作为最终预测结果。

卷积神经网络在恶意软件检测领域已经取得不错的效果[13] [14]，但由于反编译 PE 文件得到的函数调用块是具有先后时间顺序的，单一的卷积神经网络并不能有效的提取时间特征，为了解决这一问题，本文将机器翻译领域的 Transformer 与卷积神经网络相结合，利用 Transformer 模型中的多头注意力机制能够解决中长特征的依赖问题这一优势。因为在集成信息的时候，当前的特征项与整条数据中的任意特

征项发生联系时，都会通过正弦位置编码保留输入特征之间的相对位置信息，从而使得 Transformer 模型能够更高效提取恶意软件中的时间序列特征。

2.2. 数据预处理

本文采用 Ember 数据集，通过反编译工具将其解析，经过提取后得到七组主要特征类，其分别是：通用文件信息类、PE 头部信息类、区块信息类、字节统计信息类、字符串信息类、导入和到处函数类。将所有类的信息进行组合打包，添项，将数据转换成维度为 2401 的浮点型数据。

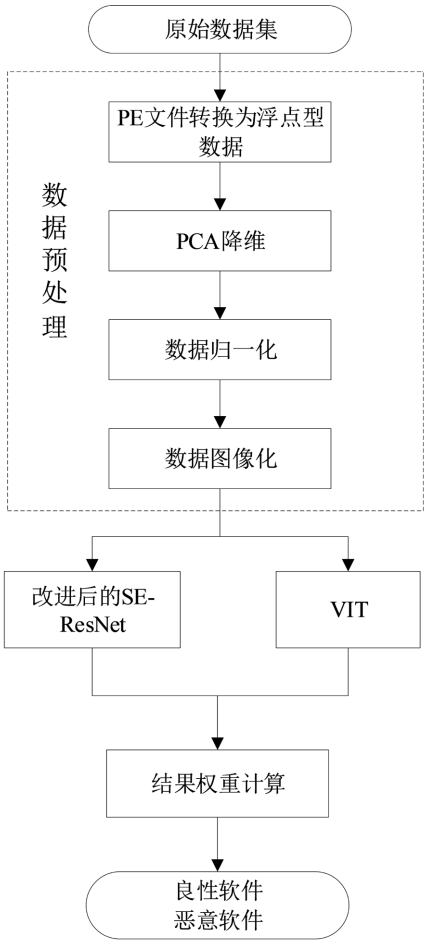

Figure 1. Network model structure diagram of this paper
图 1. 本文网络模型结构图

归一化处理，对转换成浮点型的数据进行归一化，将数据标准到[0, 1]的区间内。

其中 Y_{ij} 表示标准化后的值，i 表示第几个样本，j 表示第几个维度，为该特征项的最大值，为该特征项的最小值，则标准化如公式(1)所示：

$$Y_{ij} = \frac{X_{ij} - X_{min}}{X_{max} - X_{min}} \tag{1}$$

主成分分析(Principal component analysis, PCA)是机器学习领域中使用最广泛的数据降维算法之一[15]。同时也被广泛应用到恶意软件检测领域，文献[16]将 PCA 和 LDA (Linear discriminant analysis,

LDA)算法与多种检测模型组合，并在多个数据集上进行实验对比，结果表明，PCA 算法比 LDA 算法在降维方面有着更好的效果。本文通过 PCA 算法将原本 2401 维度的原始数据，通过主成分提取，降维到 900 维度。假设一个数据集中有 m 个对象，每个对象包含 n 个变量。要获得主成分，其步骤如下：

1) 将原始数据组成 m 行 m 列矩阵 X。对矩阵 X 进行标准化后得到新矩阵 Z：

$$z_{ij}=\frac{x_{ij}-\overline{x}_j}{s_j},i=1,2,\cdots,m;j=1,2,\cdots,n \tag{2}$$

其中，$\overline{x}_j=\frac{\sum_{f=1}^{m}x_j}{m}$，$s_j^2=\frac{\sum_{i=1}^{m}x_{ij}}{m}$。

2) 计算出协方差矩阵：

$$C=\frac{Z^TZ}{m-1} \tag{3}$$

3) 用特征值分解方法求出协方差矩阵 C 的特征值及每个特征值对应的特征向量。

4) 将特征向量按对应特征值大小从大到小排序按行排列成矩阵,取 n' 前行组成矩阵 P。

5) $Y=PZ$ 即为矩阵 X 即为降维到 n' 维后的数据。

2.3. 基于 Transformer-Encoder 模型

Transformer 是由 Vaswani A.等人在 2017 提出的一种机器翻译领域的新模型[15]。Transformer 模型主要由编码器(Encoder)和解码器(Decoder)两个大的模块组成，本文使用了 Vision Transformer 所改进的 Encoder 模块[17]。如图 2 Transformer Encoder 模块结构图所示，Transformer Encoder 模块由多个 Encoder 模块堆叠形成。单个 Encoder 模块有两个子层，一个是多头注意力机制(Multi-Head Attention, MHA)，利用 self-attention 学习软件灰度图像中的特征结构。另一个是多层感知机(Multilayer Perceptron, MLP)，通过线性映射将 MHA 放大后的维度缩小，再次输入到 Encoder 模块当中。每个子层前都使用了 LN (Layer Normalization)数据标准化方法，每个子层后都是用了残差链接。与传统的递归神经网络(Recurrent Neural Network, RNN)和 CNN 相比，Transformer 采用 MHA 并行计算，从而提升模型训练速度，并且，多头注意力机制允许每个特征关注所有其他特征，这使得每个特征都可以在其完整的特征中被考虑。多头注意力机制是自注意力机制的拓展，而自注意力机制则来自一个更一般的 Scaled Dot Attention 函数，其定义为：

$$(Q,K,V)=\text{softmax}(\frac{QK^T}{\sqrt{d}k})V \tag{4}$$

其中 Q、K 和 V 分别是由 Attention 中的 query 向量、key 向量、value 向量组成的矩阵，为隐藏层的维度，softmax 函数的作用是通过指数特性将所有预测结果映射在 0 到正无穷的区间内，再通过归一化将所有预测结果转化为 0 到 1 之间的概率分布。多头注意力机制是由多个并行的 Attention 构成，定义公式如下：

$$\begin{aligned}&MultiHead(Q,K,V)=concat(head_1,\cdots,head_h)W^O\\&head_i=Attention(QW_i^Q,KW_i^K,VW_i^V)\end{aligned} \tag{5}$$

其中 $W_i^Q\in R^{d_{modle}*d_k},W_i^K\in R^{d_{modle}*d_k},W_i^V\in R^{d_{modle}*d_k},W_i^O\in R^{hd_v*d_{modle}}$ 而 d_{modle}，d_v 分别表示模型的维度和隐藏层的维度。

在 Vision Transformer 中，线性嵌入层是的一个重要结构。线性嵌入层将图像分割成多个块，然后将每一块扁平化为一维张量，再将位置编码和类别编码共同嵌入到张量中，并输入到 Transformer 编码器当

中。再由编码器提取特征后输出到全连接层当中，由全连接层执行一个分类任务。本文是一个恶意软件检测的二分类任务，因此 Vision Transformer 最终输出为良性软件或恶意软件。Vision Transformer 检测模型如图 3 所示。

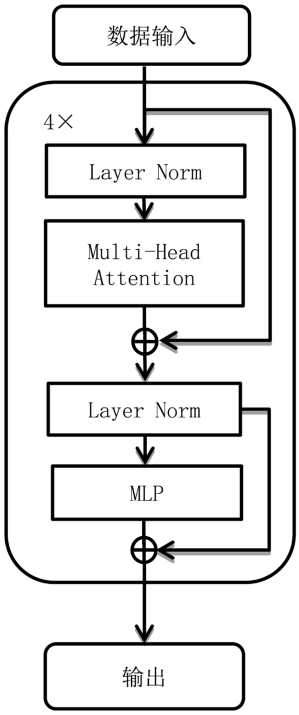

Figure 2. Transformer Encoder module structure diagram
图 2. Transformer Encoder 模块结构图

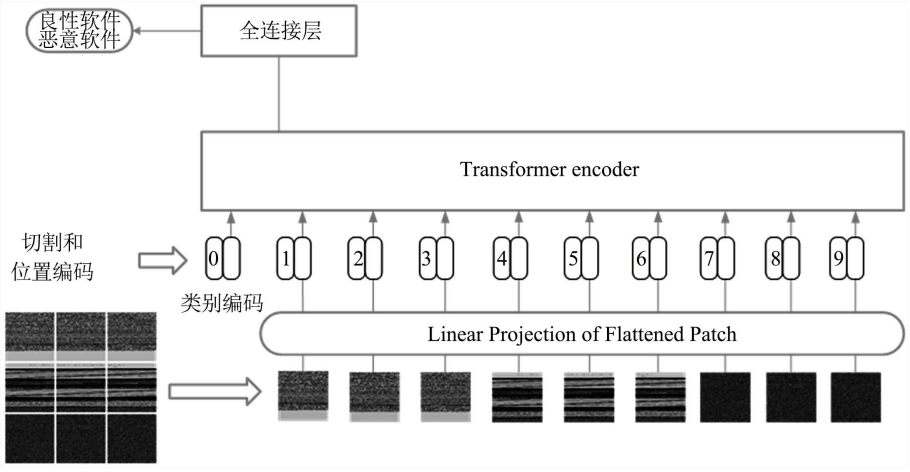

Figure 3. Vision Transformer detection model
图 3. Vision Transformer 检测模型

2.4. 改进的 SE-ResNet 卷积神经网络检测模型

卷积神经网络一直是计算机视觉、机器翻译、图像识别、目标检测等多个领域的主要模型架构。SENet (Squeeze-and-Excitation Networks)是 Jie Hu 等人在 2018 年提出的一种用于图像分类的卷积神经网络模型

[18]，其结构如图 4 所示。图中 Ftr 与常规的卷积操作相同，将输入通道数为 C'，高度为 H'，宽度为 W' 的矩阵向量 X，提取为通道数为 C，高度为 H，宽度为 W 的矩阵向量 U，SENet 的核心是 U 后面的 SE 模块，首先 SENet 对 U 进行一个 Squeeze 操作，通过 global average pooling 将每一个通道上整个空间特征编码为一个全局特征，从而提升卷积的感受野。接下来的 Excitation 操作学习每个通道上全局特征的关系，得到不同通道的权重，最终再将 Excitation 输出的权重和 U 相乘得到最终特征。SE-ResNet 是将 SENet 中的 SE 模块插入到 ResNet [19]中的每个残差块之间，利用 SE 模块捕捉通道之间的相互关系，提高模型的泛化能力。SE-ResNet 的 Squeeze 操作是通过全局池化实现的，全局池化将整个输入的特征图进行池化，能够将每个特征图的信息都汇总在一起，但它不考虑特征图中的位置信息，从而导致损失空间信息。同时在反向传播中，梯度可能会在多次池化之后减小到很小的值，导致梯度消失，从而影响整个模型的性能。

本文通过一种双线性池化融合机制，具体结构如图 5 所示，通过融合全局池化和最大池化的 Squeeze 操作，既保留了全局池化将特征信息汇总在一起的优势，又增加了最大池化能够保留了每个池化窗口中的最大值，从而不会丢失特征图中的位置信息。

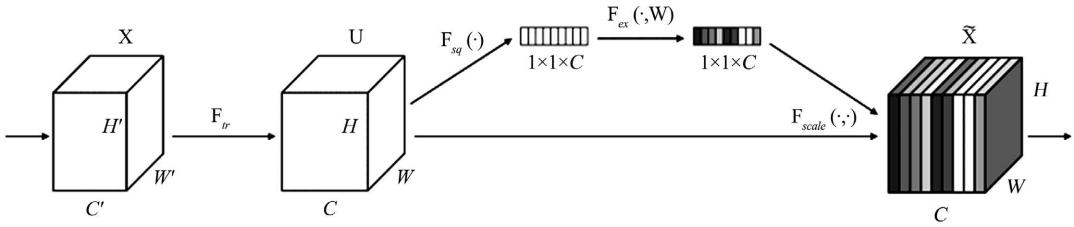

Figure 4. SENeT network model structure diagram
图 4. SENeT 网络模型结构图

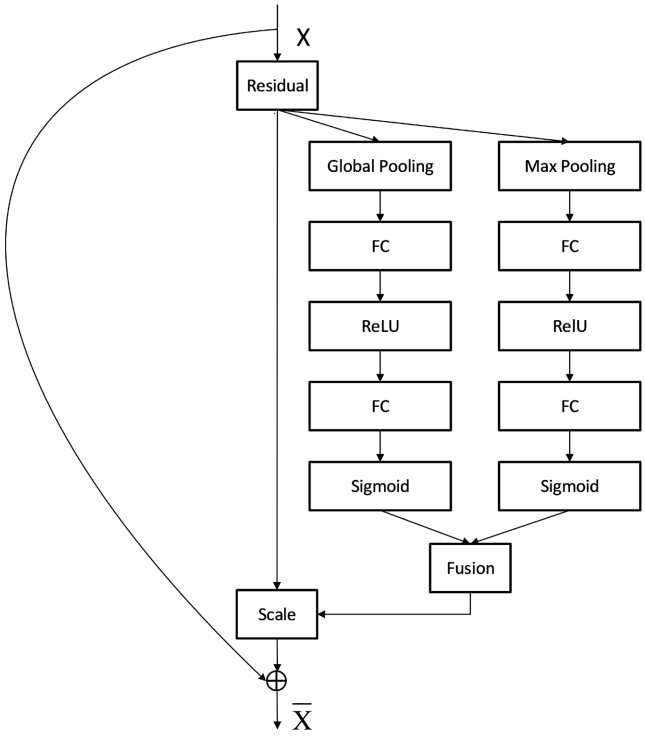

Figure 5. Improved SE-ResNet architecture diagram
图 5. 改进的 SE-ResNet 结构图

3. 实验结果及分析

3.1. 实验环境

文实验的环境搭建在 CentOS 7 操作系统中，其中编程语言采用的是 Python 3.7，深度学习框架采用的是 Pytorch 1.10.1，中央处理器为 Intel(R)Xeon(R)CPU E5-2620 v4 @ 2.10 GHz，显卡为 NVIDIA Tesla P100。

本文采用的 Endgame 在 2018 年发布的一个关于恶意软件的开源数据集 Ember，这是可用于研究 Windows 平台恶意软件检测的最大数据集之一。Ember 数据集扫描了 110 万个恶意软件样本，其中 90 万个训练样本，20 万个测试样本。而训练样本中包括 30 万个未标注样本，这些样本在模型训练中没有用处，会在预处理数据集时进行删除，表 1 是删除未标注样本后的数据分布。

Table 1. Type and quantity of experimental data
表 1. 实验数据的类别与数量

类别	训练集	测试集
良性软件	300,000	100,000
恶意软件	300,000	100,000

3.2. 评价指标及参数设置

3.2.1. 评价指标

在本文采用准确率(*Accuracy*)、精确率(*Precision*)、召回率(*Recall*)和 *F1* 值(F1 Score)对恶意软件检测方法性能进行评估。其指标详细定义为公式(6)，(7)，(8)和(9)：

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

其中 *TP* (True Positive)表示真实数据为恶意软件样本且模型预测结果也为恶意软件的数据总数，*TN* (True Negative)表示真实数据为良性软件样本且模型预测结果也为良性软件样本的数据总数，*FP* (False Positive)表示真实数据为良性软件样本且模型预测为恶意软件样本的数据总数，*FN* (False Negative)表示真实数据为恶意软件样本且模型预测为良性软件样本的数据总数。

3.2.2. 参数设置

本文中使用的 PCA 提取的主成分特征数目为 625，SE-ResNet-VIT 网络模型中有两层卷积层，滤波器数目分别为 10 和 20，滤波器大小分别为 4*4，6*6，降采样层的大小为 2。VIT 中设置为 6 个多头注意力机制，encoder 迭代 4 次。优化器选择 SGD，初始学习率为 0.003，batch size 设置为 256，epoch 为 200。使用 warm-up 策略来提高优化器 SGD 在训练过程中的稳定性。

3.3. 实验结果分析

图 6 是本文提出模型的训练集和测试集准确率变化曲线图。由图可见，模型的训练轮次达到 100 个

Epoch 之后，测试集的 Accuracy 已经停止增长，稳定在 0.97，说明模型已经拟合了。

3.3.1. 单一模型和组合模型对比

为了验证 SE-ResNet-VIT 组合模型的检测效果，将组合模型与改进的 SE-ResNet 和 VIT 进行比较。由表 2 的实验结果表明，组合模型在 Accuracy、Precision、Recall 和 F1 四个指标方面均有明显提升。

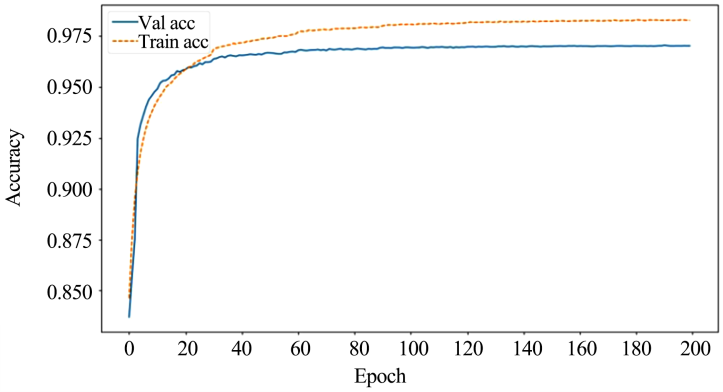

Figure 6. Model accuracy change curve of this paper
图 6. 本文模型准确率变化曲线图

Table 2. Single model and combined model performance comparison (%)
表 2. 单一模型和组合模型性能对比(%)

模型	Accuracy	Precision	Recall	F1
SE-ResNet50	90.26	88.15	93.02	90.52
改进 SE-ResNet	94.77	95.45	94.02	94.73
VIT	91.63	92.38	90.40	91.38
SE-ResNet-VIT	96.44	97.06	95.77	96.41

3.3.2. PCA 对组合模型的影响

为了验证 PCA 降维消除冗余后的所有软件数据对模型中有着更优异的表现，本文将经过多次降维比对后确定降维后的维度为 900。降维后的数据与原始数据使用相同参数的 CNN-Transformer 模型进行训练并测试模型性能，通过表 4 的实验数据能够说明，在加入 PCA 降维之后，能够有效的去除 Ember 数据集中的冗余特征项，提高 CNN-Transformer 模型的特征提取效率，从而使得模型在 Accuracy 提高了 0.61，同时单轮训练时间由原来的 53.41 s 降低到 42.75 s，减少 10.66 s，模型训练速度大幅降低。实验结果如表 3 所示：

Table 3. Comparison of model performance after dimensionality reduction using PCA
表 3. 使用 PCA 降维后模型性能对比

模型	Accuracy (%)	单轮训练时间/s
SE-ResNet-VIT	96.44	53.41
PCA- SE-ResNet-VIT	97.05	42.75

3.3.3. 与现有检测模型对比

为了验证本文提出模型性能的先进性，将本文提出的 PCA-SE-ResNet-VIT 模型与文献[3]提出的

PCA-SVM 模型、文献[8]改进的 DNN 模型、文献[10]的 Hierarchical Transformer 进行实验比较和现有的多种深度学习模型在 Ember 数据集上进行对比，实验结果表明，通过 PCA 降维去除冗余后，再结合卷积层和多头注意力机制的优势，该 SE-ResNet-VIT 检测模型在 Accuracy、Precision、Recall 和 F1 四个指标方面均优先现有的恶意软件检测模型。实验结果如表 4 所示：

Table 4. Detection effect of different models on Ember dataset (%)
表 4. 不同模型在 Ember 数据集上的检测效果(%)

模型	Accuracy	Precision	Recall	F1
文献[3]	91.10	91.00	91.20	91.10
文献[8]	95.11	96.73	95.45	96.09
文献[10]	95.68	96.37	95.11	95.74
CNN-LSTM	87.52	86.35	89.07	87.69
Resnet50	89.95	88.97	92.35	90.63
本文模型	97.05	97.45	96.64	97.04

4. 结束语

本文对恶意软件的原始数据进行了降维，去冗余，并将降维前后的数据和 SE-ResNet-VIT 集成模型展开研究，合并卷积层和多头注意力机制二者的优势，将其应用到恶意软件检测中。通过在 Ember 数据集上的实验结果表明，本文提出的方法在 Accuracy、Precision、Recall 和 F1 四个评价指标方面均优于文献[3]，文献[8]，文献[10]所提出的模型，且高于现有常见的深度学习网络模型，从而证明了本文提出模型的有效性。下一步将本文的模型应用在其他的恶意软件数据集上，尽管 Ember 数据集的数据已经很广泛，涵盖了大多数种类的恶意软件，但它并不包括所有可能存在的种类，但现有的数据集更新速度难以跟上恶意软件的迭代速度，未来将通过自行收集恶意软件样本，在自行收集的数据集下测试本文提出模型的性能。

基金项目

广州市重点领域研发计划项目(202007010004)。

参考文献

[1] Komatwar, R. and Kokare, M. (2021) Retracted Article: A Survey on Malware Detection and Classification. *Journal of Applied Security Research*, **16**, 390-420. https://doi.org/10.1080/19361610.2020.1796162

[2] 国家互联网应急中心. 2021 年上半年我国互联网网络安全监测数据分析报告[R]. https://www.cert.org.cn/publish/main/upload/File/first-half%20%20year%20cyberseurity%20report%202021.pdf

[3] Qi, P., Zhang, Z., Wang, W., *et al*. (2021) Malware Detection by Exploiting Deep Learning over Binary Programs. 2020 25*th International Conference on Pattern Recognition* (*ICPR*) *IEEE*, Milan, 10-15 January 2021, 9068-9075. https://doi.org/10.1109/ICPR48806.2021.9412227

[4] Gibert, D., Mateu, C., Planes, J., *et al*. (2018) Classification of Malware by Using Structural Entropy on Convolutional Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, **32**, 7759-7764. https://doi.org/10.1609/aaai.v32i1.11409

[5] Ye, Y., Chen, L., Hou, S., *et al*. (2018) DeepAM: A Heterogeneous Deep Learning Framework for Intelligent Malware Detection. *Knowledge and Information Systems*, **54**, 265-285.

[6] Wang, S., Zhou, G., Lu, J., *et al*. (2019) A Novel Malware Detection and Classification Method Based on Capsule Network. In: Sun, X.M., Pan, Z.Q. and Bertino, E., Eds., *International Conference on Artificial Intelligence and Security*, Springer, Cham, 573-584. https://doi.org/10.1007/978-3-030-24274-9_52

[7] Albahar, M.A., ElSayed, M.S. and Jurcut, A. (2022) A Modified ResNeXt for Android Malware Identification and Classification. *Computational Intelligence and Neuroscience*, **2022**, Article ID: 8634784. https://doi.org/10.1155/2022/8634784

[8] 张柏翰, 凌捷. 改进的基于 DNN 的恶意软件检测方法[J]. 计算机工程与应用, 2021, 57(10): 81-87.

[9] Sun, R., Yuan, X., He, P., *et al*. (2017) Learning Fast and Slow: Propedeutica for Real-Time Malware Detection. *IEEE Transactions on Neural Networks and Learning Systems*, **33**, 2518-2529.

[10] Hu, X., Sun, R., Xu, K., *et al*. (2020) Exploit Internal Structural Information for IoT Malware Detection Based on Hierarchical Transformer Model. 2020 *IEEE* 19*th International Conference on Trust*, *Security and Privacy in Computing and Communications* (*TrustCom*), Guangzhou, 29 December-1 January 2021, 927-934. https://doi.org/10.1109/TrustCom50675.2020.00124

[11] Anderson, H.S. and Roth, P. (2018) Ember: An Open Dataset for Training Static PE Malware Machine Learning Models.

[12] Abbasi, E., Moghaddam, M.R.A. and Kowsari, E. (2022) A Systematic and Critical Review on Development of Machine Learning Based-Ensemble Models for Prediction of Adsorption Process Efficiency. *Journal of Cleaner Production*, **379**, Article ID: 134588. https://doi.org/10.1016/j.jclepro.2022.134588

[13] 金逸灵, 陈兴蜀, 王玉龙. 基于 LSTM-CNN 的容器内恶意软件静态检测[J]. 计算机应用研究, 2020, 37(12): 3704-3707+3711. https://doi.org/10.19734/j.issn.1001-3695.2019.08.0565.5

[14] 傅依娴, 芦天亮, 马泽良. 基于 One-Hot 的 CNN 恶意代码检测技术[J]. 计算机应用与软件, 2020, 37(1): 304-308+333.

[15] Abdi, H. and Williams, L.J. (2010) Principal Component Analysis. *Wiley Interdisciplinary Reviews*: *Computational Statistics*, **2**, 433-459. https://doi.org/10.1002/wics.101

[16] Vaswani, A., Shazeer, N., Parmar, N., *et al*. (2017) Attention Is All You Need. 31*st Conference on Neural Information Processing Systems* (*NIPS* 2017), Long Beach, 4-9 December 2017, 30.

[17] Dosovitskiy, A., Beyer, L., Kolesnikov, A., *et al*. (2020) An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale.

[18] Hu, J., Shen, L. and Sun, G. (2018) Squeeze-and-Excitation Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745

[19] He, K., Zhang, X., Ren, S., *et al*. (2016) Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90