Computer Science and Application 计算机科学与应用, 2025, 15(5), 548-557
Published Online May 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.155127

Hans汉斯

基于深度学习的端到端恶意软件检测方法研究

曾尚文[1]，陈丽芳[1,2]

[1]华北理工大学理学院，河北 唐山
[2]河北省数据科学与应用重点实验室，河北 唐山

收稿日期：2025年4月6日；录用日期：2025年5月8日；发布日期：2025年5月14日

摘　要

目前恶意软件对网络安全构成了严重威胁，现有基于机器学习的恶意软件检测方法，需要恶意软件分析师花费大量时间和精力来构建动态或静态特征，因此在实践中难以应用。为有效缓解上述问题，提出了一种基于深度学习的端到端恶意软件检测方法。与传统检测方法相比，所提方法具有端到端学习过程的优势。首先，提取恶意软件的前n字节，其中包含恶意软件关键信息作为模型输入；然后，基于卷积神经网络设计一种新的深度学习模型，引入残差网络和多头注意力机制，提高模型对不同输入的适应性以及对于复杂特征的提取能力。最后，经实验验证表明，该方法资源消耗低，并大大提升了检测精度。

关键词

恶意软件检测，深度学习，端到端，网络安全

Research on End-to-End Malware Detection Method Based on Deep Learning

Shangwen Zeng[1], Lifang Chen[1,2]

[1]College of Science, North China University of Science and Technology, Tangshan Hebei
[2]Hebei Key Laboratory of Data Science and Application, Tangshan Hebei

Received: Apr. 6th, 2025; accepted: May 8th, 2025; published: May 14th, 2025

Abstract

At present, malware poses a serious threat to network security. Existing malware detection methods based on machine learning require malware analysts to spend a lot of time and energy to construct dynamic or static features, so they are difficult to apply in practice. To effectively alleviate the above problems, an end-to-end malware detection method based on deep learning is proposed. Compared with traditional detection methods, the proposed method has the advantage of an end-

文章引用: 曾尚文, 陈丽芳. 基于深度学习的端到端恶意软件检测方法研究[J]. 计算机科学与应用, 2025, 15(5): 548-557. DOI: 10.12677/csa.2025.155127

to-end learning process. First, the first n bytes of malware are extracted, which contain key information of malware as the model input. Then, a new deep learning model is designed based on convolutional neural network, and residual network and multi-head attention mechanism are introduced to improve the adaptability of the model to different inputs and the ability to extract complex features. Finally, experimental verification shows that this method has low resource consumption and greatly improves the detection accuracy.

Keywords

Malware Detection, Deep Learning, End-to-End, Network Security

1. 引言

恶意软件是一种未经授权访问计算机，旨在损害、干扰网络设备的安全威胁[1]。随着计算机技术的发展，恶意软件的发展速度不断提高，使得对恶意软件进行准确分类变得越来越困难。除了进化速度快之外，恶意软件的识别严重依赖专家知识[2]是一个很大的挑战。

根据恶意软件代码的分析方式，恶意软件检测技术通常分为静态分析和动态分析[3]。静态分析需要专家非常熟悉恶意软件的架构和操作系统原理，但已被证明容易受到混淆。动态分析依赖于恶意软件的进程状态、网络数据、注册表操作日志等行为数据，但这些数据通常需要在沙箱中[4]运行，不光会消耗安全人员的时间和硬件资源，另一方面环境感知恶意软件[5]一旦在分析环境中被检测到，就会改变其行为。端到端[6]是一种为了减少模型训练中的人为干扰而设计的技术，它可以同时训练整个模型，而不是单独训练每个部分，端到端模型也会同时优化所有处理步骤。Dieleman 等人[7]使用端到端模型自主地从原始音频输入中提取特征，以便理解输入表示，同时他们注意到模型发现的特征中保留了一些不变性。此外，端到端模型还表现出更高的准确性和更好的泛化能力，经过端到端训练的模型通常更加紧凑，参数数量较少。近年来相继提出了一系列深度神经网络模型，直接对可执行文件的原始字节进行操作，从数据中有效的学习特征表示，无需其他语法语义信息，用以检测恶意软件并成功取得了较高的性能[8]。例如 Scott E. Coull 等人[9]通过对比多个不断提高数据量和正则化水平下训练的模型，用来探索这些训练变量之间的关系、模型从中学习的特征以及这些模型在正确分类恶意软件方面的有效性，证实了基于字节的深度学习分类器是传统机器学习的可行替代方案。Karpathy 等人[10]证明了卷积神经网络(Convolutional Neural Network, CNN)架构能够从弱标记数据中学习强大的特征，在性能上远远超过基于特征的方法。

为了最小化领域知识、减少对专家经验的依赖以及应对误报率高、安全性低等问题，本文提出了一种基于深度学习的端到端恶意软件检测模型，通过 CNN 直接对原始字节进行处理，提取恶意软件的高级特征，并通过设计的残差模块和多头注意力机制捕获分散在恶意软件不同块中的相关信息。

2. 理论基础

2.1. 卷积神经网络

卷积神经网络[11]是一种在深度学习中广泛应用的工具。其中最重要的结构组成部分是卷积层，卷积

操作具有局部连接和权值共享的特性，有助于减少参数数量和计算量，它通过采用一系列可训练的卷积核捕获输入数据的空间关系特征，进行特征提取。激活层使用激活函数在模型中引入非线性[12]，增加网络的表达能力。池化层主要用于降低特征图的空间尺寸，可以在一定程度上保持特征的平移不变性。全连接层连接所有神经单元，将这些特征映射到最终的输出类别或回归值。

2.2. 多头注意力机制

注意力机制解决了模型需要在编码过程中顺序进行[13]的困难，通过计算查询(query)与一组键值对(key-value pairs)之间的相关性，来动态地分配权重，从而聚焦于输入中的不同部分。然而，进一步的研究发现注意力机制在编码输入序列时会过分集中于自身位置的注意力，这种过度集中会导致模型在处理长序列或复杂关系时表现不佳。为了应对这一挑战，多头注意力机制应运而生。多头注意力机制[14]的核心思想建立在对注意力机制的扩展和改进之上，将注意力的计算分为多个“头”，每一组注意力机制都独立地处理输入序列，产生不同的注意力表示。这种设计的优势在于它能够更全面、更准确地捕捉序列中的关键信息，增强模型的表达能力。

2.3. 残差网络

残差连接是残差网络中的核心组成部分。它的基本思想是在神经网络的层与层之间添加一条直接的连接[15]，使得信息可以直接从前面的层传递到后面的层，而不仅仅是通过层层的非线性变换。具体来说，对于一个神经网络层 l，其输入为 x_l，输出为 y_l，传统的神经网络结构中 $y_l = H(x_l)$，其中 $H(\cdot)$ 是该层的非线性变换函数(例如卷积操作、全连接层等)。而在残差网络中，引入了残差块(residual block)，其输出 $y_l = H(x_l) + x_l$，这里的 x_l 到 y_l 的直接连接就是残差连接。

3. 研究方法

图 1 展示了本文提出的基于深度学习的端到端恶意软件分类模型的框架，称为 IMCP。模型架构的选择在很大程度上受到了可执行文件中存在的高度位置变化的影响。PE 二进制文件的内容几乎可以以任意顺序重新排列，唯一固定的常量是 MS-DOS 头部[16]，它以指向 PE 头部开头的指针结束。同时 PE 头部包含指向二进制文件所有其他内容(包括代码段、数据段和资源段等)的指针，它可以位于文件的任何地方，其部分内容可以分布在整个文件中。这使得在不影响程序正常执行的情况下，存在 PE 文件内容重组的可能性，所以如何在整个数据范围内捕获信息是一个很重要的方向。

IMCP 框架主要由五个部分组成：(1) 嵌入模块：同时引入词嵌入和位置嵌入，将原始字节序列映射为稠密向量，作为神经网络输入；(2) 卷积模块：该模块旨在从字节流输入中提取局部特征表示；(3) 残差模块：该模块有助于梯度的传播和模型更好地学习特征；(4) 多头注意力模块：该模块试图捕获不同恶意软件部分之间的相关性；(5) 全连接层：该模块用于输出样本为恶意软件的概率，即确定样本是否为恶意软件。

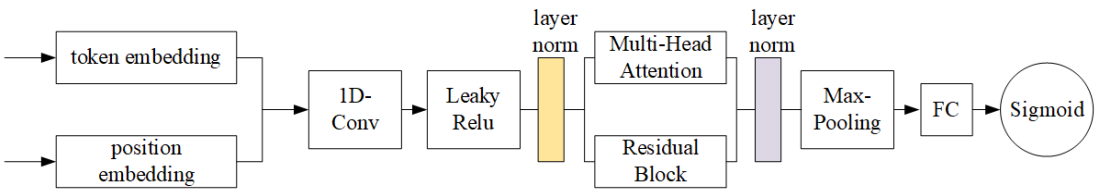

Figure 1. Flowchart of the IMCP algorithm
图 1. IMCP 算法流程图

3.1. 嵌入层

在模型初始化阶段，同时使用词嵌入和位置嵌入[17]。词嵌入层用于将输入的前 256 个字节转换为 32 维的向量表示，这样可以将离散的输入数据映射到连续的向量空间，使得模型能够更好地处理和理解输入的语义信息。位置嵌入则为每个元素提供了其在序列中的位置信息，同样将位置索引映射为 32 维向量，使得模型能够区分不同位置的特征，更好地捕捉序列中的上下文信息。这对于恶意软件检测非常重要，因为恶意软件的某些特征可能与其在代码中的位置相关，例如，某些恶意操作可能总是出现在特定的代码段位置。通过将两者进行相加操作，可以让模型同时学习到两者的特征，更好地理解输入样本的语义和结构，提高特征提取的准确性。在体系结构的测试过程中，发现常用的正则化方法 Dropout 是最有效的，这里设置为相对较大的 0.5，能够有效地减少过拟合风险，尤其是在训练数据量不足的情况下，可以避免模型对单个单词或位置信息的过度依赖。

$$\tilde{E} = Dropout\left(T + P\right) \tag{1}$$

其中词嵌入可以表示为 $T = \left(t_1, t_2, \cdots, t_n\right)$，$t_i$ 是输入序列中第 i 个元素的词向量表示；位置嵌入可以表示为 $P = \left(p_1, p_2, \cdots, p_n\right)$，$p_i$ 是输入序列第 i 个位置的位置向量表示。$\tilde{E}$ 表示词嵌入和位置嵌入融合后的嵌入表示。

3.2. 卷积层

实验选择基于 CNN 模型进行构建，以掌握这种高级位置不变性。将卷积激活与全局最大池化相结合，使模型能够根据检测到的特征的位置产生相应的激活。嵌入层与卷积层共同训练，使得即使网络层数较浅，也能对更广泛的输入模式产生激活。经过嵌入层融合后的向量作为卷积层的输入，卷积层通过卷积核在这些向量上滑动进行特征提取，能够捕捉到输入数据中的局部模式和特征。考虑到任务需求和输入的数据维度，卷积层的滤波器个数设置为 256，每个滤波器的大小为 128。之后通过 LeakyRelu 激活函数帮助网络更好的拟合数据中的非线性部分。

3.3. 多头自注意力机制模块

创新性地引入了多头自注意力机制技术。先将激活后的输出进行层归一化处理，接着在维度调整后将输入向量分成 4 个“头”，也就是每个注意力头在进行注意力计算时的向量维度。并进行独立的注意力计算，然后将结果合并起来，这种机制可以让模型在处理输入序列时，更加关注不同位置之间的关系，捕捉长距离依赖。

$$h_i = f\left(W_i^{(q)}q, W_i^{(k)}k, W_i^{(v)}v\right) \tag{2}$$

接下来使用定义好的线性层对分割后的值(values)、键(keys)和查询(query)向量进行线性变换，得到新的向量表示。再通过矩阵乘法和缩放操作计算注意力权重，将能量矩阵除以输入嵌入维度的平方根进行缩放，这种缩放操作可以帮助稳定训练过程，尤其是在输入维度较高时，防止注意力权重变得过于极端。最后通过注意力权重和“值”向量进行加权求和并映射回原始的嵌入维度，得到最终的输出。

$$MultiHead\left(Q, K, V\right) = Concat\left(H_1, H_2, \cdots, H_n\right)W^o \tag{3}$$

其中 $W_i^{(q)}$、$W_i^{(k)}$、$W_i^{(v)}$ 为 Q、K、V 投影到不同表示子空间的参数矩阵，H_i 为第 n 个头的输出，W^o 是指多头拼接后的参数矩阵。

3.4. 残差网络模块

残差模块的主要目的是解决随着网络层数增加而出现的梯度消失问题，同时能够让网络更容易学习

恒等映射，提高模型的性能和训练稳定性。这里通过两个一维卷积层进行进一步的特征提取，设置卷积核大小为 3，填充为 1，这样可以保持输入和输出的序列长度不变。第三个一维卷积层则会在当输入通道数和输出通道数不同时，对输入进行下采样实现统一。最后对将经过两次卷积和批归一化后的输出与恒等映射(可能经过下采样)相加，实现残差连接。对残差连接后的结果再次使用 ReLU 激活函数，得到最终的输出。

$$y = \mathrm{ReLU}\left(y_2 + x_{downsampled}\right) \tag{4}$$

其中 y_2 是 x 经过两次卷积操作和批归一化后得到的输出。$x_{downsampled}$ 为对原始输入 x 进行下采样操作后的输入结果。

3.5. 最终分类层

对经过注意力层处理后的输出和残差连接后的输出进行相加融合，并且使用 LayerNorm 对输出后的输出进行层归一化。在最后一个维度上进行最大池化，目的是将每个特征通道中最重要的信息提取出来，将输出特征压缩为一个固定长度的向量。这样可以减少参数数量，同时突出了最显著的特征，使得模型对输入数据的关键特征有更强的表征能力，有助于后续的分类决策。

再将全局最大池化后的特征输入到全连接层中，将其映射到一个标量输出，表示样本是否为恶意软件。全连接层对池化后的特征进行线性组合，通过学习到的权重和偏置，对输入进行最后的分类判断。这里将 256 维的特征映射为 1 维的输出，最后利用 sigmoid 函数[18]对得到的结果进行预测判别，即判断样本为恶意软件的概率得分，最后可以根据设定的阈值来确定样本是否为恶意软件。

算法 1 改进的 MalConvPlus：

输入：输入数据 x，嵌入维度 $embed\_dim$，最大长度 max_len，输出通道数 $out\_channels$，卷积核大小 $window\_size$，头个数 num_heads，丢弃率输出 $dropout$。

输出：样本属于某一类别的概率值。

初始化：词嵌入层 tok_embed；位置嵌入层 pos_embed；丢弃层 $dropout$；卷积层 $conv$；归一化层 $layer\_norm1$ 和 $layer\_norm2$；自注意力层 $self\_attention$；全连接层 fc。

前向传播

1) 获取 $batch\_size$ 和 seq_len 作为 x 的批量大小和序列长度

2) 计算词嵌入：$tok\ embedding \leftarrow tok\ embed(x)$

3) 创建位置索引 pos

4) 计算位置嵌入：$pos\ embedding \leftarrow pos\ embed(pos)$

5) 计算嵌入：$embedding \leftarrow dropout(tok\ embedding + pos\ embedding)$

6) 将 $embedding$ 赋值给 $conv\_in$

7) 卷积处理：$conv\ out \leftarrow conv(conv\ in)$

8) 使用残差连接：$residual\_out \leftarrow$ 残差模块$(conv\ out)$

9) 进行层归一化：$L_1\_out \leftarrow layer\_norm1(conv\ out)$

10) 计算自注意力：$att\_out \leftarrow self\_attention(L_1\_out, L_1\_out, L_1\_out)$

11) 注意力与残差相加：$out \leftarrow att\_out + residual\_out$

12) 进行层归一化：$L_2\_out \leftarrow layer\_norm2(out)$

13) 计算最终输出：返回 $fc(out)$

4. 实验与分析

4.1. 实验环境

实验的硬件环境为 11 g 的 2080ti，操作系统是 windows 11。编程语言为 python，深度学习框架为 PyTorch。模型训练过程中使用学习率为 0.001 的 Adam 优化器进行参数调整，批处理大小设置为 8，训练周期设置为 50。

4.2. 实验数据集

本研究通过从 Wikidll.com 和 Dasmalwerk.eu [19]两个网站下载字节数据进行采集，共计获得 1330 条样本数据。其中包括 820 个良性样本和 510 个恶意样本。良性样本来源于常见且广泛使用的 DLL 文件，而恶意样本则从专为研究用的恶意软件库获取，确保了样本的有效性。同时为每个样本分配标签，良性软件为 0，恶意软件为 1，并将数据集分为 80%训练集和 20%测试集。

4.3. 评价指标

模型性能评估是衡量模型在二分类任务[20]中有效性的关键环节。为了全面且准确地评价模型的表现，我们采用了一系列经典的评价指标：准确率(Accuracy)、精确率(Precision)、召回率(Recall)和 F1 值(F1-Score)，同时引入 AUC 更直观的判断模型在区分正负样本方面的能力，这些指标基于模型预测结果与真实标签构建的混淆矩阵而得出。

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$

$$AUC = \int_0^1 \frac{TP}{TP + FN} d \frac{FP}{TN + FP} \tag{9}$$

4.4. 实验结果及分析

为验证本文实验的有效性，本文实现了近年来同类型的恶意软件检测的方法或模型，这些方法均是使用本文数据集并在同一实验环境下比较的。其中包括 RCNN、AttentionRCNN、SimpleGRU、GRU-CNN 等模型，使用以上方法作为对比算法，实验结果见表 1。

Table 1. Comparative experimental results of algorithms
表 1. 算法对比实验结果

方法	Accuracy	Precision	Recall	F1-score	Auc
RCNN [21]	0.898	0.889	0.842	0.865	0.963
AttentionRCNN [22]	0.915	0.924	0.851	0.886	0.970
GRU-CNN [23]	0.875	0.874	0.789	0.829	0.934
SimpleGRU [24]	0.729	0.766	0.430	0.551	0.541
MalConvBase [25]	0.929	0.951	0.860	0.903	0.970
MalConvPlus	0.915	0.932	0.842	0.885	0.978
IMCP	0.983	0.991	0.965	0.978	0.998

由表 1 可见，本文采用的方法在准确率、精确率、召回率、F1 值和 AUC 值上均高于 96%，各项指标都优于其它对比方法。这表明 IMCP 模型在正确分类恶意软件和正常样本方面表现最为出色，能够更准确地判断样本的类别。同时 IMCP 模型能够访问整个文件，在数据中的任何位置可以检测到少量的信息特征，有效避免了各种过拟合问题。

根据其他对比模型的结果可以得出，SimpleGRU 的性能最差，各项指标均显著低于其他模型，这表明该模型在处理任务时能力有限，简单的模型可能无法捕捉到复杂数据中的重要信息，而需要结合更复杂的结构来提升性能。GRU-CNN 的性能表现中等，但相比 SimpleGRU 模型也有较大提升，这是由于 GRU 主要用于处理时间序列数据，能够有效建模时间依赖性和上下文信息，但对于处理恶意软件数据，可能难以捕捉到特征之间的空间关系；而 CNN 对局部特征的捕捉能力较强，能够学习到更强的特征，更加适合处理恶意软件字节文件。在 RCNN 模型中引入了残差连接的思想，将 RNN 最后一个时间步的输出和卷积层的最大值拼接在一起，这个拼接后的结果作为全连接层的输入，实现了残差连接，使得卷积层中的部分信息能够直接参与到后续的分类操作中。AttentionRCNN 的性能更为出色，这表明引入注意力机制能够有效提升模型对重要关键特征的关注，从而提高了整体性能。

MalConvBase 和 MalConvPlus 两个模型为实验初期设计的基础模型架构，同样都采用了词嵌入层、一维卷积层、门控线性单元(GLU)和全连接层的基本结构，用于恶意软件检测任务。不同之处在于 MalConvPlus 在 MalConvBase 的基础上增加了位置嵌入层，能够同时利用输入的语义信息和位置信息，更好地捕捉输入序列的特征。

本文设计的模型与其它方法相比的主要创新点在于：1. 同时引入词嵌入层和位置嵌入层，帮助模型更好理解输入样本的语义和结构；2. 模块间信息传递的创新，使得 CNN 提取的局部特征能够通过残差连接和层归一化更好地与多头注意力机制关注的全局特征相结合，增强了不同模块之间的协同性。

本文设计了消融实验，验证了引入多头注意力机制和残差网络连接以及层归一化技术的有效性，实验结果见表 2。

Table 2. Results of ablation study on model variants
表 2. 消融实验模型对比结果

模型	Accuracy	Precision	Recall	F1 Score	Auc-roc
IMCP	0.983	0.991	0.965	0.978	0.998
IMCP_NoAttention	0.973	0.949	0.982	0.966	0.997
IMCP_NoResidual	0.973	0.973	0.956	0.965	0.998
IMCP_NoLayerNorm	0.963	0.956	0.947	0.952	0.994

由表 2 可知，加入多头注意力机制操作后，F1 值提高了 1.2%，证明多头注意力机制不仅更准确的聚焦于恶意软件的特征，而且让模型关注到更多可能包含恶意软件特征的区域，有助于模型学习到更具代表性的特征从而提高检测性能。残差模块的操作使 F1 值提高了 1.3%，证明了残差连接在获取恶意软件邻域信息和全文信息方面的优越性，有助于提高模型的泛化能力和鲁棒性。加入层归一化的操作后，F1 值提高了 2.6%，说明层归一化操作可以帮助模型在训练过程中更稳定的学习特征，不容易受到数据分布变化的影响，从而提高了模型的精确率和召回率。实验中所有对比模型和消融模型的可视化结果如图 2 所示。

如图 2 柱状可视化图所示，在 IMCP 模型的基础上分别将残差模块、多头注意力机制模块和层归一化模块去掉，得到的测试结果均受到影响，在各项评价指标上都有所下降。这充分说明了每一个模块在模型中都不可或缺，他们分别从特征学习、信息传递、模型稳定性和泛化能力等方面提升了模型的性能。

同时所提方法相比其他算法模型具有明显的性能优势，能够较为准确地识别样本，在恶意软件领域中具有较强的检测能力。

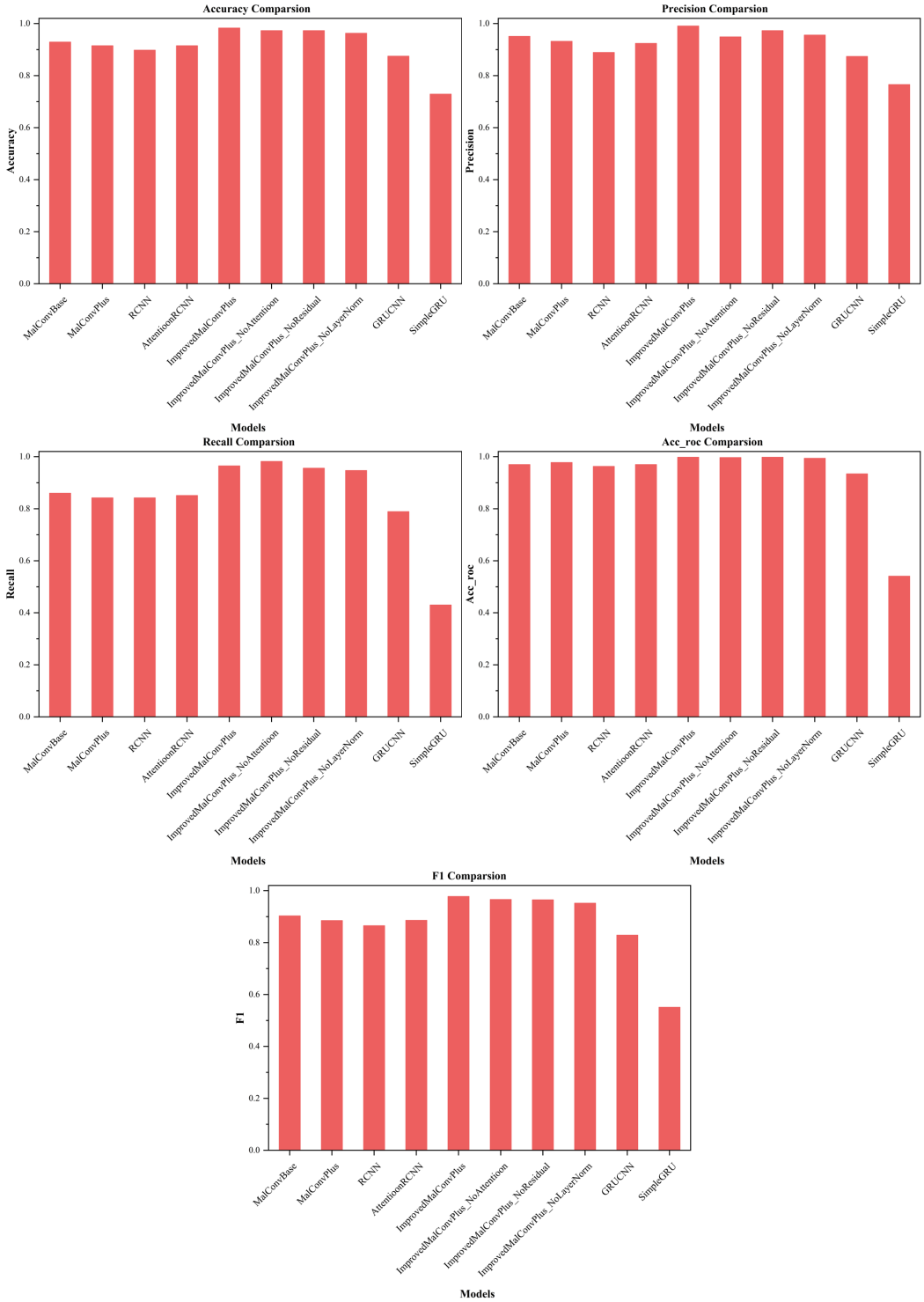

Figure 2. Comparative analysis of evaluation metrics across multiple models
图 2. 多种模型各项评价指标对比

5. 结论

最小化领域知识的深度学习方法提高了恶意软件检测的自动化程度，减少了对专家经验的依赖。相较于特征数据，原始文件数据更容易获取。因此本文针对恶意软件领域提出了一种基于深度学习的端到端检测模型。使用端到端深度学习可以充分利用大规模数据，同时端到端学习避免了人工特征提取的局限性，有望提高检测的准确性和泛化性能。未来的研究可以进一步探索深度学习中这些模块的改进和优化方法，以及它们之间的更好组合方式，以进一步提升模型的性能。同时，也可以考虑将其他相关技术或模块引入到模型中，更好地应对不同类型的恶意软件，以适应不断变化的恶意软件检测需求。

参考文献

[1] Maniriho, P., Mahmood, A.N. and Chowdhury, M.J.M. (2023) API-Maldetect: Automated Malware Detection Framework for Windows Based on API Calls and Deep Learning Techniques. *Journal of Network and Computer Applications*, **218**, Article 103704. https://doi.org/10.1016/j.jnca.2023.103704

[2] Zhu, H., Wei, H., Wang, L., Xu, Z. and Sheng, V.S. (2023) An Effective End-to-End Android Malware Detection Method. *Expert Systems with Applications*, **218**, Article 119593. https://doi.org/10.1016/j.eswa.2023.119593

[3] Hou, Z., Li, X., Li, L., Yuan, J. and Deng, K. (2022) An End-to-End Raw Bytes Based Malware Classifier via Self-Attention Residual Convolutional Network. 2022 *IEEE 8th International Conference on Computer and Communications* (*ICCC*), Chengdu, 9-12 December 2022, 1666-1670. https://doi.org/10.1109/iccc56324.2022.10065922

[4] 陈岑, 李暖暖, 蔡军飞, 等. 基于动态行为特征加权聚类的加壳恶意软件未知变种检测方法[J]. 重庆大学学报, 2023, 46(3): 129-136.

[5] 欧阳坜伶, 彭国军. 面向安卓系统的环境感知 API 自动化检测方案[J/OL]. 武汉大学学报(理学版), 1-13. https://doi.org/10.14188/j.1671-8836.2023.0176, 2024-11-02.

[6] 王金华. 端到端的基于深度学习的网络入侵检测方法[J]. 通信技术, 2022, 55(6): 762-770.

[7] Dieleman, S. and Schrauwen, B. (2014) End-to-End Learning for Music Audio. 2014 *IEEE International Conference on Acoustics*, *Speech and Signal Processing* (*ICASSP*), Florence, 4-9 May 2014, 6964-6968. https://doi.org/10.1109/icassp.2014.6854950

[8] Gopinath, M. and Sethuraman, S.C. (2023) A Comprehensive Survey on Deep Learning Based Malware Detection Techniques. *Computer Science Review*, **47**, Article 100529.

[9] Coull, S.E. and Gardner, C. (2019) Activation Analysis of a Byte-Based Deep Neural Network for Malware Classification. 2019 *IEEE Security and Privacy Workshops* (*SPW*), San Francisco, 19-23 May 2019, 21-27. https://doi.org/10.1109/spw.2019.00017

[10] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. (2014) Large-Scale Video Classification with Convolutional Neural Networks. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 1725-1732. https://doi.org/10.1109/cvpr.2014.223

[11] Snow, E., Alam, M., Glandon, A. and Iftekharuddin, K. (2020) End-to-End Multimodel Deep Learning for Malware Classification. 2020 *International Joint Conference on Neural Networks* (*IJCNN*), Glasgow, 19-24 July 2020, 1-7. https://doi.org/10.1109/ijcnn48605.2020.9207120

[12] 卢法权, 陈丹伟. 基于改进 CNN-LSTM 融合的僵尸网络识别方法[J]. 计算机应用与软件, 2024, 41(3): 328-335.

[13] Vaswani, A. (2017) Attention Is All You Need. *Advances in Neural Information Processing Systems*, **30**, 5998-6008.

[14] Voita, E., Talbot, D., Moiseev, F., Sennrich, R. and Titov, I. (2019) Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, July 2019, 5797-5808. https://doi.org/10.18653/v1/p19-1580

[15] Khan, R.U., Zhang, X. and Kumar, R. (2018) Analysis of Resnet and GoogleNet Models for Malware Detection. *Journal of Computer Virology and Hacking Techniques*, **15**, 29-37. https://doi.org/10.1007/s11416-018-0324-z

[16] Raff, E., Barker, J., Sylvester, J., *et al*. (2018) Malware Detection by Eating a Whole Exe. *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans. 2-7 February 2018, 292-300.

[17] Kakisim, A.G., Gulmez, S. and Sogukpinar, I. (2022) Sequential Opcode Embedding-Based Malware Detection Method. *Computers & Electrical Engineering*, **98**, Article 107703. https://doi.org/10.1016/j.compeleceng.2022.107703

[18] 李道全, 李玉秀, 任大用. 小样本下基于决策树-SNN 的恶意流量检测方法[J]. 计算机工程与应用, 2023, 59(21): 258-266.

[19] Luo, X., Fan, H., Yin, L., Jia, S., Zhao, K. and Yang, H. (2024) CAG-Malconv: A Byte-Level Malware Detection Method with CBAM and Attention-GRU. *IEEE Transactions on Network and Service Management*, **21**, 5859-5872. https://doi.org/10.1109/tnsm.2024.3424565

[20] 李红娇, 顾凡. 基于多注意力 Bi-LSTM 的恶意软件预测[J]. 计算机工程与设计, 2023, 44(12): 3529-3535.

[21] 高玮玮, 杨亦乐, 方宇, 等. 多特征尺度融合改进 Faster-RCNN 视网膜微动脉瘤自动检测算法[J]. 光子学报, 2023, 52(4): 228-239.

[22] Fang, Y., Zhang, C., Huang, C., Liu, L. and Yang, Y. (2019) Phishing Email Detection Using Improved RCNN Model with Multilevel Vectors and Attention Mechanism. *IEEE Access*, **7**, 56329-56340. https://doi.org/10.1109/access.2019.2913705

[23] 王相月, 赵利辉. 基于多阶段特征选择和 CNN-GRU 的网络入侵检测模型[J]. 中北大学学报(自然科学版), 2024, 45(1): 66-73.

[24] 盛振威, 徐国天. 基于融合 CNN 与 GRU 的 DGA 恶意域名检测方法[J]. 网络安全技术与应用, 2022(12): 29-32.

[25] Yang, L. and Liu, J. (2020) Tuningmalconv: Malware Detection with Not Just Raw Bytes. *IEEE Access*, **8**, 140915-140922. https://doi.org/10.1109/access.2020.3014245