使用稀疏卷积神经网络与快速梯度符号法进行恶意软件分类

1st Ashish Anand万豪国际布罗德兰兹，美国
gcp.ashish2020@gmail.com

2nd Bhupendra Singh万豪国际卡里，美国
bhupendra.research1@gmail.com

3rd Sunil Khemka*Persistent Systems*芝加哥，美国
sunilkhemka.tech@gmail.com

4th Bireswar Banerjee*Visa*奥斯汀，美国
bireswar.infosys@gmail.com

5th Vishi Singh Bhatia
塔塔咨询服务公司
美国路易维尔
vishisbhatia.research@gmail.com

6th Piyush Ranjan电气与电子工程师协会航空航天分会副主席美国爱迪生 piyush.ranjan@ieee.org

摘要——**安卓恶意软件已成为对组织、社会和个人日益严峻的威胁，对隐私、数据安全和基础设施构成重大风险。随着恶意软件在复杂性和精密程度上不断演变，缓解和检测此类恶意软件实例变得越来越耗时且更具挑战性，尤其是识别潜在恶意软件需要大量特征。为应对这些挑战，本研究提出了一种采用快速梯度符号法与稀疏卷积神经网络（FGSM-DICNN）的恶意软件分类方法。DICNN包含稀疏卷积，可扩大感受野，使模型能够以更少的特征捕捉长距离范围内分散的恶意软件模式，同时不增加参数。此外，FGSM策略通过在训练期间使用单步扰动来提高准确率，从而以较低的计算成本带来更强的防御能力。这种结合有助于在降低对庞大特征集的依赖的同时，保持较高的分类准确率。所提出的FGSM-DICNN模型达到99.44%的准确率，优于定制深度神经网络（DCNN）等其他现有方法。**

关键词——数据安全、稀疏卷积神经网络、快速梯度符号法、恶意软件分类、隐私。

I. 引言

据反恶意软件公司 McAfee 报告，平均每分钟都会发生多起恶意软件攻击，这对互联网构成了重大威胁 [1]。获取对各种数字资源的访问权限并对其进行渗透，会造成不良后果并增加系统的复杂性 [2]。电子商务、云技术以及运营技术（OT）与信息技术（IT）的融合等领域的数字化程度不断提高，推动了恶意软件暴露风险的持续上升 [3]。恶意软件有多种类型，例如勒索软件、病毒、蠕虫和间谍软件；每种类型都加剧了威胁的复杂性，使环境更具动态性 [4]。近年来，恶意软件攻击的复杂性和普遍性显著上升，凸显了这些网络威胁的严重性 [5]。由于数据集过时、概念漂移问题以及恶意软件变种不断演化以适应变化的计算环境，恶意软件的识别和分类仍是研究中的一项重大挑战 [6]。恶意软件种类日益多样且不断扩散，给网络安全领域带来了更多问题 [7]。

恶意软件的动态执行较不易受到代码混淆问题的影响，但这种方法也存在其他局限性。具体而言，要有效监控并记录恶意软件的行为，动态分析需要一个精确的环境 [8]。用于恶意软件分类的算法分为基于特征码的算法和人工智能驱动的算法。恶意软件特征码通常是通过静态分析、动态分析或混合分析提取的特征的哈希值 [9]。准确地对恶意软件进行分类和识别，对于增强计算机安全至关重要 [10]。关注恶意软件的传播有助于应对安全威胁并消除威胁，因此需要更先进、更完善的检测工具 [11]。尽管先进的深度学习（DL）方法有助于识别和分类复杂的恶意软件，但网络威胁不断演变的特性仍带来了重大挑战 [12]。总体而言，恶意软件分类能够提升网络安全水平，保护数字系统及资产的完整性、机密性和数据可用性，从而最大限度地降低网络风险 [13][14]。虽然各种传统方法在训练数据量很大时表现出较为可靠且显著的结果，但收集数据可能耗时，也受许多其他条件限制。 [15]。本研究的贡献如下：

- 稀疏卷积神经网络（DI-CNN）采用卷积滤波器，在保留表达能力的同时降低模型复杂度和参数数量，因此推理对准确率的影响很小，且训练速度更快。

- 快速梯度符号法（FGSM）有助于在训练期间生成对抗样本，使模型接触经过轻微修改的恶意软件输入，从而增强模型对规避攻击的抵御能力，并提升模型的泛化能力。

本研究其余部分安排如下：第 2 节介绍文献综述；第 3 节介绍所提方法的工作流程；第 4 节给出结果与讨论；第 5 节为结论。

II. 文献综述

本研究根据所提出的方法论，结合其优点与局限，对以下文献综述进行讨论。

Mudassar Waheed 等人 [16] 构建了随机森林（RF）模型，用于区分良性应用和恶意应用。研究采用探索性数据分析（EDA）进行预处理，以了解重要属性。通过纳入以下信息，从 APK文件中提取静态和动态特征：

权限、意图、API调用和行为轨迹。研究采用包装法、过滤法和嵌入法等特征选择方法，选出最具判别力的特征。该方法在维持性能的同时，有助于降低计算开销。然而，一些复杂的恶意软件类型无法绕过这些检测，从而影响了模型的整体准确率。

Nikolaos Polatidis 等人 [17] 开发了一种称为 FSSDroid 的安卓恶意软件检测特征子集选择技术，用于恶意软件检测。研究采用互信息（MI）根据特征与类别标签之间的统计依赖关系对特征进行排序。

基于相关性的特征选择（CFS）方法消除了高度相关的特征，以最大限度地减少冗余并增强泛化能力，同时降低过拟合问题。然而，由于同时使用了相同的特征，处理过程变得更具挑战性。

Sandeep Sharma 等人 [18] 提出了多包装器混合特征选择技术（MWHFST），该技术整合了基于包装器的特征选择，用于恶意软件检测和分类。所提出的 MWHFST 方法适用于各种静态特征，并不局限于 API调用。然而，从安卓应用程序中提取的混合特征通常维度过高。

Muhammad Umar Rashid 等人 [19] 实现了一种定制深度神经网络（DCNN），其中包含用于分层特征学习的多个隐藏层。丢弃和批归一化技术有助于防止过拟合，并利用 SoftMax 结果对不同恶意软件家族进行分类。此外，还采用递归特征消除（RFE）技术来识别信息量最大的特征。DCNN 对网络安全分析很有帮助，可用于了解威胁向量。然而，动态特征提取颇具挑战，因为其在实时系统中的检测速度较慢。

Arvind Prasad 等人 [20] 开发了 PermGuard 框架，用于 Android 恶意软件检测。该框架将权限映射到漏洞利用信息中，并支持增量学习。所提出的方法包含基于相似性的选择性训练方法，以减少训练所需的数据，并集中利用最相关的信息，从而降低计算开销。PermGuard 方法评估了其抵御对抗攻击的能力，从而增强了抵抗绕过其检测机制尝试的能力，提高了安全性。然而，该映射方法无法捕获零日威胁。

III. 所提方法

本研究采用所提出的 FGSM-DICNN 方法对恶意软件进行分类，以提高分类准确率。通过缺失值分析和数据清洗进行预处理，将原始数据转换为格式化数据。采用 RFE 特征选择方法筛选最具判别性的特征，以提升分类效率。图 1 展示了所提出的 FGSM-DCNN 方法的流程。

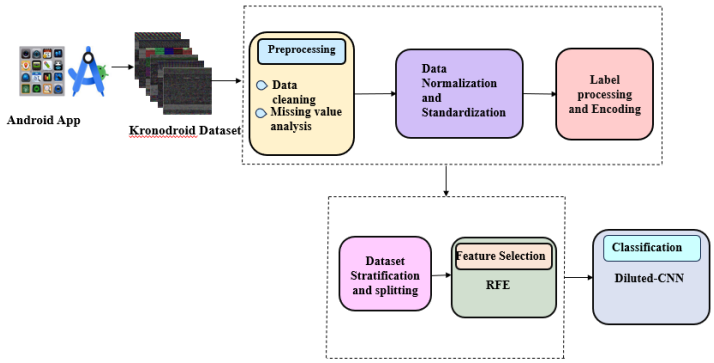

图 1. 恶意软件分类流程

A. 数据集

KronoDroid数据集 [21] 包含 78,137 个恶意软件样本，其中 41,382 个为恶意软件，36,755 个为良性样本；数据包含 489 个静态和动态特征。尽管数据库中包含多种恶意软件类别，但仅选取了短信、BankBot 和 Airpush 三类数据样本，并据此构建了三个均衡子集。

B. 使用数据清洗和缺失值进行预处理分析

在本节中，我们将数据集作为输入，对数据样本进行预处理。预处理是数据清洗的重要步骤，有助于提升数据集的可靠性和质量。在此阶段，主要处理潜在错误、不一致性和缺失值。通过评估数据质量，确定了数据集的训练和测试完整性。以下公式展示了缺失值分析的综合结果。

设本研究的数据集表示为 $D = \{d_1, d_2, \ldots, d_n\}$，其中每个特征向量为 d_i。每个特征的缺失值比率（MVR）为 f，其计算方法见公式（1）。

$$MVR(f) = \frac{\sum_{i=1}^{n} I(d_{if}=null)}{n} \quad (1)$$

其中，指示函数为 I，样本总数为 n，对所有特征进行的 $MVR = 0$ 分析证实了数据集的完整性。

1) 数据归一化与标准化：像归一化这样的基础技术用于改善数值特征，并使模型训练过程保持一致。将每个特征向量 x_j 转换为其在式（2）中给出的x_j' 标准化形式。

$$x_j' = \frac{x_j - \mu_j}{\sigma_j} \quad (2)$$

其中，特征的均值和标准差 σ_j 与 μ_j 相对于 j，分别由式（3）和式（4）计算得出。

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{i,j} \tag{3}$$

$$\sigma_j = \sqrt{\frac{1\sum_{i=1}^{n}(x_{i,j}-\mu_j)^2}{n}} \tag{4}$$

2) 标签处理与编码：类别过程标签经过高效处理并编码为数值表示，y 类别目标变量通过两阶段转换进行编码。Y 独热编码矩阵定义见公式（5）。

$$Y_{i,j} = \begin{cases} 1\ if\ y'_{i=j} \\ 0 \quad otherwise \end{cases} \tag{5}$$

标签编码：$\mathrm{y} \rightarrow \mathrm{y}$，其中 $y \in \{0,1,\ldots,k-1\}$，适用于 k 个类别。

One-Hot Encoding: $\mathrm{y} \rightarrow \mathrm{Y}$ where $Y \in \{0,1\}^{n\times k}$.

3) 数据集分层与划分：采用分层抽样，以确保训练集和验证集中各类别的均衡表示。在管理类别分布的同时完成数据集划分。对于每个c 类别，根据公式（6）、（7）和（8）处理以下约束：

$$\frac{|S_c^{train}|}{|S^{train}|} \approx \frac{|S_c^{val}|}{|S^{val}|} \approx P_c \tag{6}$$

$$|S^{val}| = \eta|s| \tag{7}$$

$$\left|S^{train}\right| = (1-\eta)|S| \tag{8}$$

其中，样本子集 S_c 属于g 类别c，数据中的原始类别传播为S_c。划分比例为η，分别设置在集合中为 0.2。

C. 使用 *RFE* 进行特征选择

在恶意软件分类任务中，API调用、提取属性和标头信息等特征被组合成特定模型的特征集。RFE方法根据性能得分，在每次迭代中系统地移除最不重要的特征，同时重新训练模型。该过程会重复进行，直到获得目标数量的最优预测特征。RFE通过提升可解释性并提高计算效率来减少过拟合，同时处理高维恶意软件数据。通过聚焦于区分良性与恶意软件的最具判别性的模式，消除无关属性。该特征子集不仅提升了分类性能，还能实现更快速、更可靠的恶意软件检测。

D. 分类

所提出的 DI-CNN 方法有助于对恶意样本进行分类，从而实现准确区分。基线卷积神经网络受卷积核尺寸的限制，存在一些主要问题：首先，CNN 只能捕捉短期依赖关系；其次，扩大卷积核会显著增加参数数量。DI-CNN 提高了整体网络效率，同时增强了对更多数据和更大视野尺寸的采集能力。这种分类方法有效扩展了分类任务中的感受野。DI-CNN 的单向扩展“1”将“E”信号与包含“r”特定视野尺寸的“R”核相结合。该方法可扩展至式（9）所示的二维卷积。

$$(E * p_x R)_s = \sum_\tau R_\tau E_{s-p_x\tau} \tag{9}$$

其中，空洞卷积中相应层的膨胀宽度呈指数级增加，为 $E * p_x$ “l”x。

1) 快速梯度符号法：这种输入扰动技术利用与梯度方向相关的梯度损失，同时添加微小扰动；输入数据如式（10）所示。

$$x_{adv=}X + \epsilon.sign(\nabla_x J(\theta, x, y)) \tag{10}$$

其中，扰动大小由 Epsilon（ϵ）确定，取值为 0.01、0.05、0.1；x 是输入中表示扰动强度的典型值，真实类别标签为 y，损失函数为 J。由此生成的对抗样本可用于恶意软件分类。

IV. 实验结果与讨论

所开发的 FGSM-DICNN 的性能根据用于训练和测试的数据集进行评估。该 FGSM-DICNN 在 Python 3.8 环境中实现，系统运行于配备 Intel Core i7-4200U CPU 处理器、Windows 10、64 位操作系统和 16GB 内存的设备上。该 FGSM-DICNN 技术采用多种分析指标进行评估，即召回率、精确率、F1 分数和精确率。各项指标的数学表达式分别由式（11）至式（14）给出。

数据集中被正确分类的像素所占的百分比称为准确率。

$$Accuracy = \frac{TP}{TP+FN+TN+FP} \tag{11}$$

所有预测为正的预测值中，真正值所占的百分比称为精确率。

$$Precision = \frac{TP}{TP+FP} \tag{12}$$

根据召回率的预测效用方程，计算所有被估计为正的正样本所占的比例。

$$Recall = \frac{TP}{TP+FN} \tag{13}$$

F1 分数衡量精确率与召回率之间的互补关系，如下所示。

$$F1 = \frac{2\times Precision \times Recall}{Precision \times Recall} \tag{14}$$

其中，TP – 真正例；TN – 真负例；FP – 假正例；FN – 假负例。

A. 性能分析

使用 KronoDroid 数据库对所开发的 FGSM-DICNN 进行全面分析。本节通过多种指标的性能验证所设计方法的有效性。图 2 展示了恶意软件分类分析。

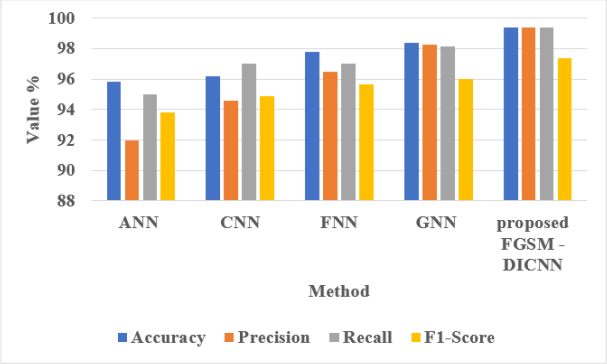

图 2. FGSM-DICNN 与现有深度学习模型的分析

图 2 分析了 FGSM-DICNN 与现有深度学习模型数据库的表现。我们将不同深度学习方法的效率进行比较：人工神经网络 [ANN], 、卷积神经网络 [CNN], 、前馈层 [FNN] 和图神经网络 [GNN] ，并与开发的 FGSM-DICNN 方法进行对比。与这些方法相比，开发的 FGSM-DICNN 展现出最佳的探索能力，并能识别导致准确分类的各种恶意软件类型。FGSM-DICNN 方法在 KronoDroid数据集上的准确率达到 99.41%。

B. 比较分析

所提出的 FGSM-DICNN 方法通过多项性能指标进行比较。表 1 展示了该方法与传统方法的比较结果，具体包括随机森林 [16], FSSDroid [17], MWHFST [18], DDCNN [19] 和 PermGuard [20] 。与现有模型相比，FGSM-DICNN 的准确率表现更优。我们采用评估指标对 FGSM-DICNN 方法的性能进行评估。

表 I. 现有方法与所提方法的比较分析

方法	准确率	精确率	召回率	F1 分数
随机森林 [16]	87.24	90.14	74.20	81.39
FSSDroid [17]	98.50	98.50	98.60	-
MWHFST [18]	88	88	88	88
DCNN [19]	94	94	94	94
PermGuard [20]	99.33	99.34	99.32	99.33
所提出的方法 FGSM-DICNN	99.41	99.38	99. 37	99.36

表 1 介绍了所提出的 FGSM-DICNN 方法及 KronoDroid 数据库的性能指标。与所分析的方法相比，所提方法表现出色，能够提高检测准确率。FGSM-DICNN 方法在 KronoDroid数据集上的准确率达到 99.41%。

C. 讨论

安卓操作系统不仅是使用最广泛的移动操作系统，也是网络犯罪分子最有利可图的目标，这归因于其庞大的用户群。空洞卷积层用于安卓恶意软件分类，使模型能够捕捉分布于不同片段中的模式，并捕捉应用中特征及特征表示的长程依赖关系。借助空洞卷积，模型无需牺牲分辨率即可扩大感受野，这对于检测分散的恶意活动尤为重要。FGSM 方法在训练期间提供扰动样本，这些样本难以由模型处理，可能误导检测并对输入进行细微修改。这种集成通过区分恶意样本和良性样本，提高了分类准确率。与其他方法相比，所提出的 FGSM-DICNN 模型在 KronoDroid数据集上的准确率达到 99.41%。

V. 结论

Android 是使用最广泛的移动操作系统，负责管理种类繁多的数据，从普通消息到敏感的银行信息。恶意软件针对该平台的爆炸式使用而出现，因此必须采用 DICNN，以高效地对恶意软件进行分类和检测。所提出的 FGSM-DICNN 方法通过识别问题，增强了模型区分分布于特征空间不同区域的模式的能力。FGSM 方法通过轻量且有效的技术提升对抗依赖性，使模型能够接触扰动样本。该集成方法可用于对 Android应用程序中的恶意软件进行有效分类。与其他方法相比，所提出的 FGSM-DICNN 模型在 KronoDroid数据集上的准确率达到 99.41%。未来将采用先进的深度学习模型来抵御非防御性攻击和非扩张攻击，增强防御能力和上下文建模，从而使其更适用于现实世界中的恶意软件系统。

参考文献

[1] A. Bensaoud 和 J. Kalita，“基于操作码和 API 调用的恶意软件分类 CNN-LSTM 与迁移学习模型”，Knowl.- Based Syst.，第 290 卷，文章编号 111543，2024 年 4 月。[2] V. Vouvoutsis、F. Casino 和 C. Patsakis，“超越沙箱：利用符号执行进行规避型恶意软件分类”，Comput. Secur.，第 149 卷，文章编号 104193，2025 年 2 月。[3] S. J. I. Ismail、B. Rahardjo、T. Juhana 和 Y. Musashi，“MalSSL——用于精准且标签高效的恶意软件分类的自监督学习”，IEEE Access，第 12 卷，第 58823–58835 页，2024 年 4 月。[4] G. Kale、G. E. Bostancı 和 F. V. Celebi，“面向机器学习恶意软件分类的进化特征选择”，Eng. Sci. Technol. Int. J.，第 56 卷，文章编号 101762，2024 年 8 月。[5] S. Duraibi，“采用蛇优化算法与深度卷积神经网络增强基于图像的恶意软件分类”，IEEE Access，第 12 卷，第 95047-95057 页，2024 年 7 月。[6] R. Alguliyev、R. Aliguliyev 和 L. Sukhostat，“利用深度学习在信息物理系统中基于 Radon变换的恶意软件分类”，Results Control Optim.，第 14 卷，文章编号 100382，2024 年 3 月。[7] M. E. Farfoura、I. Mashal、A. Alkhatib、R. M. Batyha 和 D. Rosiyadi，“一种基于矩阵块均值下采样的物联网恶意软件分类新型轻量级机器学习框架”，Ain Shams Eng. J.，第 16 卷，文章编号 103205，2025 年 1 月。[8] D. Gibert、G. Zizzo、Q. Le 和 J. Planes，“通过（去）随机平滑提升基于深度学习的恶意软件检测器的对抗鲁棒性”，IEEE Access，第 12 卷，第 61152–61162 页，2024 年 4 月。[9] A. Egitmen 、 A. G. Yavuz 和 S. Yavuz，“TRConv：通过目标调控卷积实现多平台恶意软件分类”，IEEE Access，第 12 卷，第 71492–71504 页，2024 年 5 月。[10] K. Alfarsi、S. Rasheed 和 I. Ahmad，“采用少样本学习方法进行恶意软件分类”，Inf.，第 15 卷，文章编号 722，2024 年 11 月。

[11] D. Z. Syeda 和 M. N. Asghar，“利用机器学习对 Windows 可移植可执行文件进行动态恶意软件分类和 API 分类”，Appl. Sci.，第 14 卷，文章编号 1015，2024 年 1 月。[12] M. Ashawa、N. Owoh、S. Hosseinzadeh 和 J. Osamor，“采用基于 Transformer 的卷积神经网络（卷积神经网络）增强基于图像的恶意软件分类”，Electron.，第 13 卷，文章编号 4081，2024 年 10 月。[13] O. Sharma、A. Sharma 和 A. Kalia，“MIGAN：用于促进恶意软件图像合成并提升新数据集上恶意软件分类效果的生成对抗网络”，Expert Syst. Appl.，第 241 卷，文章编号 122678，2024 年 5 月。[14] J. Jeon、B. Jeong、S. Baek 和 Y. S. Jeong，“TMaD：在安全融合 ICT 环境中利用多视图特征进行三层恶意软件检测”，Expert Syst.，第 42 卷，文章编号 e13684，2025 年 7 月。[15] P. Wu、M. Gao、F. Sun、X. Wang 和 L. Pan，“面向恶意软件分类的多视角 API 调用序列行为分析与融合”，Comput. Secur.，第 148 卷，文章编号 104177，2025 年 1 月。[16] M. Waheed 和 S. Qadir，“利用增强版 KronoDroid数据集实现有效且高效的 Android 恶意软件检测与类别分类”，Secur. Commun. Netw.，第 2024 卷，文章编号 7382302，2024 年 4 月。[17] N. Polatidis、S. Kapetanakis、M. Trovati、I. Korkontzelos 和 Y. Manolopoulos，“FSSDroid：安卓恶意软件检测特征子集选择”，World Wide Web，第 27 卷，文章编号 50，2024 年 7 月。[18] S. Sharma、R. Chhikara 和 K. Khanna，“一种新型特征选择技术：Android 恶意软件检测与分类”，Egypt. Inform. J.，第 29 卷，文章编号 100618，2025 年 3 月。[19] M. U. Rashid、S. Qureshi、A. Abid、S. S. Alqahtany、A. Alqazzaz、M. S. Al Reshan 和 A. Shaikh，“利用深度神经网络进行混合 Android 恶意软件检测与分类”，Int. J. Comput. Intell. Syst.，第 18 卷，文章编号 52，2025 年 3 月。[20] A. Prasad、S. Chandra、M. Uddin、T. Al-Shehari、N. A. Alsadhan 和 S. S. Ullah，“PermGuard：利用权限到利用映射进行 Android 恶意软件检测的可扩展框架”，IEEE Access，第 13 卷，第 507-528 页，2024 年 12 月。[21] KronoDroid数据集：https://www.kaggle.com/datasets/dhoogla/kronodroid-2021（访问日期：2025 年 9 月）。