摘　要

随着移动设备的普及，安卓系统正面临日益增长的恶意软件威胁。为了应对这一挑战，研究人员提出了各种各样的方法来检测和分类安卓恶意软件。但是随着恶意软件的快速演变和不断变化的攻击手段，传统方法在处理这些问题上具有局限性，特别是在处理大规模训练样本和样本不平衡问题上存在困难。近年来，图神经网络作为一种强大的机器学习工具，已经被广泛应用于安卓恶意软件检测与分类任务中。通过构建恶意软件样本之间的关系图，并利用图神经网络提取图结构的特征，能够更准确地识别和分类恶意软件。

目前，在安卓恶意软件检测任务中，大部分利用图神经网络的模型都是通过元路径确定邻居节点，然后从邻居节点聚合特征，忽略了元路径上其他节点的特征信息。同时，对于安卓恶意软件分类任务来说，收集到的恶意软件样本中，不同恶意软件家族之间存在样本分布不平衡的问题，导致分类效果不理想。为解决上述问题，本文的主要研究内容如下：

（1）针对使用图神经网络的恶意软件检测系统对异质信息图中信息利用不充分的问题，提出了一种基于元路径聚合图神经网络的安卓恶意软件检测方法，引入元路径聚合图神经网络，对整条元路径编码嵌入，聚合基于元路径的邻居节点、元路径上中间节点以及要嵌入节点本身的特征信息，利用该图神经网络以更少的样本获取更多更全面的有助于安卓恶意软件检测的信息。同时，还对元路径聚合图神经网络进行了改进，通过增加参数来限制节点在每种元路径下与邻居节点之间元路径实例的数量，以此来减少系统资源的消耗，提高效率。在实验中，使用 2,505 个恶意应用程序和 2,431 个良性应用程序，就实现了 99.68%的准确率。

（2）对于恶意软件家族分类任务中，不同家族之间的样本数量存在不平衡问题，提出了一种基于混合图神经网络的安卓恶意软件分类方法。通过梯度加权类激活映射提取带有家族敏感信息的 API，与权限和 Android 应用一同构建异质信息图，利用一层元路径聚合图神经网络和一个 GraphSage 层对节点进行嵌入，再引入 GraphSMOTE 方法对样本数量较少的家族进行过采样，最后利用 GNN 分类器对扩增后的图进行节点分类。使用 Drebin 数据集的前二十个家族对所提方法进行评估，实验结果显示，对于二十个家

族，达到了 91%的分类准确率。

关键词：图神经网络 安卓恶意软件检测与分类　元路径聚合 过采样

目　录

1　绪论

1.1　研究背景

如今，安卓系统广泛应用于众多用户和设备，并且拥有丰富的应用程序资源。由于其可拓展性和开放性，越来越多的移动智能设备选择安卓作为其首选操作系统。然而，随着其流行程度的增加，安卓恶意软件的安全威胁也在增加，包括隐私泄露[1]、数据安全[2]问题以及垃圾信息[3]的困扰。不仅对个人的信息安全构成风险，还对整个移动生态系统的稳定性和安全性构成了重大威胁。

据 360 公司《2023 年上半年度中国手机安全状况报告》[4]显示：2023 年上半年度，360 安全大脑共截获移动端新增恶意程序样本约 1699.4 万个，2022 年上半年度为 1079.7 万个，同比上升了 57.4%，平均每天截获新增手机恶意程序样本约 9.4 万个。图 1.1 显示了 2023 年上半年度移动端各月新增恶意程序样本量统计。

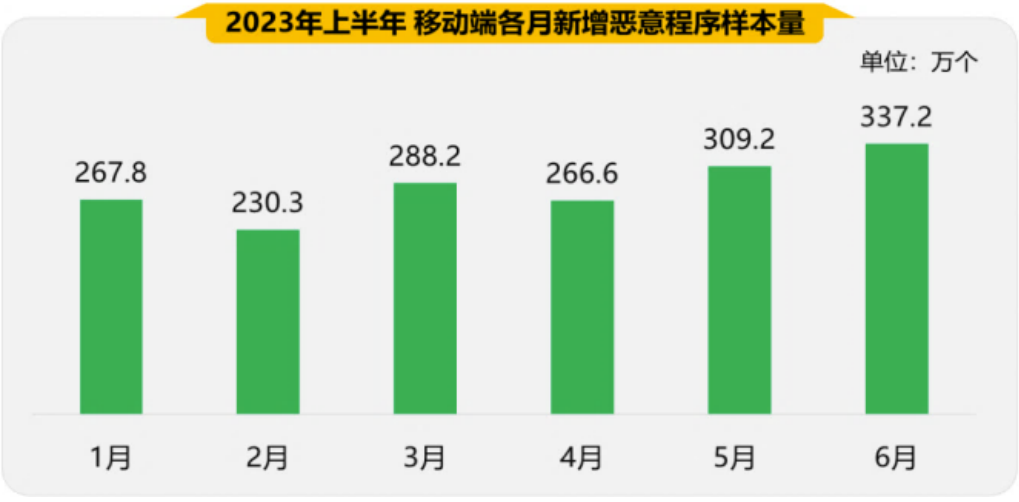

图 1.1　2023 年上半年移动端各月新增恶意程序样本量

每年都会涌现数以百万计的新型恶意软件，而且这些恶意软件也在不断演变，恶意软件作者采用越来越多的欺骗性技术来逃避传统的检测方法，通过加密或者混淆等手段来规避检测系统。为了应对日益增长和多样化的安卓恶意软件威胁，研究人员和安全专家不断努力开发新的检测方法和技术，以识别、分析和对抗这些恶意软件。同时，研究人员们也一直在对安卓恶意软件的家族分类开展研究，家族分类对于防范恶意软件威胁也是一种重要且有效的方法，通过将恶意软件样本分类到不同的家族中，从而更好地理

解和对应它们各自的独特的特征、行为和攻击模式。研究家族分类不仅能提高恶意软件检测的准确率和效率，还可以揭示恶意软件家族之间的相互关系和演化趋势，为安全研究以及应对措施提供更深入的见解。

然而，无论是安卓恶意软件的检测还是分类，仍面临许多挑战和困难。安卓恶意软件的数量庞大且持续增长，加上家族之间边界模糊和伪装的情况，使得检测与分类任务变得越来越复杂。随着人工智能技术的快速发展，研究人员开始探索利用机器学习和深度学习等技术来提高安卓恶意软件的检测和分类效果。目前，在安卓恶意软件检测与分类领域，图神经网络[5][6][7]现出巨大潜力，已成为研究热点。尤其是异质图神经网络，其能够分析丰富的节点和边，提取它们的语义信息，并通过学习节点本身的特征和图的拓扑结构来获取更深层次的隐含关系。而这些隐含关系不容易被一些恶意行为所隐藏掩盖，因此异质图神经网络特别适用于安卓恶意软件的检测与分类任务。

过去使用图神经网络进行安卓恶意软件检测与分类的方法中，例如 HAWK[8]中利用异质图注意力网络[9]进行恶意软件检测，仅仅是通过元路径来确定邻居节点，对元路径上其他节点的特征信息存在利用不充分的问题，这些信息对于恶意软件的检测和分类来说还是至关重要的。而且当前利用图神经网络的恶意软件分类方法中对于家族之间样本数量不平衡问题也没有给出较好的解决方案。

1.2 国内外研究现状

无论是安卓恶意软件的检测方法还是分类方法，都可以从特征来源的角度分为静态方法、动态方法以及动静结合的混合方法。大多数静态方法通过特征工程提取特征，然后利用机器学习或深度学习模型进行分类。动态方法则是通过在沙盒等虚拟环境或者实际中运行 apk 文件，并动态监控应用程序的执行过程及其与外部环境的交互来提取特征。混合方法则是将静态特征与动态特征结合，对 APP 进行更加全面的描述。一些利用图神经网络的方法，其本质也是提取静态特征构建图结构数据，分析不同应用程序之间的关联，下面论文将其单独作为一部分介绍其研究现状。

（1） 静态方法

大部分静态方法都是首先解压 apk 文件得到特定文件，例如“AndroidManifest.xml”“classes.dex”等，然后从这些文件中提取所需特征，代码信息、API 调用或权限等信息最常作为样本的典型特征。Zarni 等人[10]通过分析 Android 应用程序配置文件 AndroidManifest.xml，提取出 Android 应用程序调用的权限信息，并将其作为特征。Qiao

等人[11]提出了一种自动提取 API 调用和权限的恶意软件检测方法。作者将*.dex 字节码反编译为 Java 代码，并创建了一个 API 调用列表，发现这种方法比检查 AndroidManifest.xml 文件获取的权限更准确。Zhu 等人[12]使用反编译方法将“classes.dex”文件转换成 Smali 代码，从中提取 API 调用作为特征。Aafer 等人[13]提出了 DroidAPIMiner，它能提取在 API 层捕获的恶意软件行为特性的相关信息，并使用生成的特征集在不同分类器上评估分类效果。Arp 等人[14]提出了 Drebin，通过静态分析提取出所需硬件、应用权限、APP 组件、特殊 API 和网络地址等信息作为特征嵌入矩阵，然后通过 SVM（支持向量机）算法得到一个线性模型以及每个特征对应的权重，该方法进一步提高了检测精度。Zhu 等人[15]直接将“classes.dex”的二进制代码作为原始特征，提出了一种端到端的检测方法。Taheri 等人[16]提取出权限特征、API 特征以及意图特征后，将其组成一个二进制特征向量，利用随机森林回归器进行特征选择，再计算其相似度进行检测。Milosevic 等人[17]使用反编译方法获取 Java 源代码，然后使用自然语言处理中的 Bag-of-words 方法从源代码中提取语义特征，再使用 SVM 等模型进行二分类。Karbab 等人[18]从 apk 文件中提取权限特征、API 特征以及字节码的 n-gram 特征，并在降维后使用聚类方法对其进行分类。

以上方法尽管取得了不错的检测或分类精度，但是还存在一些缺点和不足。在基于 AndroidManifest.xml 的方法中，可能会受到应用程序开发者故意隐藏或修改权限信息的影响，且无法完全捕获应用程序的行为特征，因为 AndroidManifest.xml 可能无法反映应用程序的全部行为。在基于反编译提取 API 调用的方法中，反编译过程可能受到代码混淆和加固技术的影响，导致提取的 API 调用信息不完整或错误。同时，这些基于机器学习的方法，在提取特征和构建分类模型时可能会忽略一些关键信息，导致模型的泛化能力不足，且无法应对恶意软件的不断演化。

一些使用深度学习方法的安卓恶意软件检测分类模型正在获得更多关注，例如基于图像的检测技术，它先将 apk 文件或者解压后的部分文件转换为图像，然后对这些图像进行检测或分类。这种基于图像的方法不需要进行任何特征工程，可以有效地防止对称加密和恶意代码混淆。Zhang 等人[19]将 xml 文件的可视化特征与 dex 文件的数据部分结合起来创建灰度图像数据集，然后将这些图像输入到时间卷积网络[20]（TCN）以检测安卓恶意软件。Nisa 等人[21]将从预训练的深度神经网络提取的特征与基于分割分形纹理分析（SFTA）的恶意代码图像特征相结合，构建了恶意代码的多模态表示以检测灰度图像。

Zhu 等人[22]提出了一种新颖的卷积神经网络变体，名为 MADRF-CNN，它可以将 Dalvik 可执行文件的重点部分转换为 RGB 图像，并捕获 RGB 图像各部分之间的依赖关系。Zhu 等人[23]提出了 MSerNetDroid 方法，提取了权限、硬件信息和 API 特征，将这些特征转换为图像，并使用他们提出的 MSerNet 进行分类。Yuan 等人[24]首次从 apk 文件种读取字节序列，并基于字节间的邻近关系建立了一个尺寸为 256*256 大小的马尔科夫转移概率矩阵，然后使用 CNN 模型进行恶意软件家族分类。陈等人[25]将安卓软件 classes.dex 文件转换成 RGB 图像，并使用自然语言处理中表现突出的 Transformer 算法对图像进行多分类检测，实现了更高的准确率。

以上方法仍然存在一些挑战和限制。例如，将 apk 文件或部分文件转换为图像可能会造成信息损失，因为图像无法完全反映原始文件的所有细节和结构信息；而且图像转换需要使用特定的预处理技术，如灰度化、分割分形纹理分析等，这些预处理过程可能会受到数据质量和特定领域知识的限制，影响最终的检测分类效果。

（2） 基于图神经网络的方法

Allix 等人[26]提出了 CSBD（Classify Suspicious Behaviors from Dynamic Android API Calls）方法用来提取 apk 文件中的控制流图，从图中提取基本块，将每个 apk 文件表示为一组基本块的集合，然后根据每个样本中是否存在某个基本块构建二进制特征向量，最后使用随机森林等模型进行二分类。Xu 等人[27]从 apk 文件中提取了控制流图和数据流图，并将它们与权重结合起来，最后使用 CNN 模型对图的邻接矩阵进行分类。Pektas 等人[28]将 API 调用图作为恶意软件在运行时可以跟踪的所有可能执行路径的图形表示，并将 API 调用图嵌入到低维数值向量特征集中，然后通过深度神经网络进行有效的相似性检测。汪等人[29]提出了一种基于可达性关系提取的异构图压缩算法，通过对 n 阶关系矩阵的无穷幂级数的运算，提取并综合 apk 之间丰富的可达信息关系，将 apk-API 大型异构关系图压缩为 apk 带权同质图。Pei 等人[30]构建了一种新颖且高度可靠的深度学习框架—AMalNet，使用独立循环神经网络[31]（IndRNN）解码深层语义信息，充分利用文件结构中的远程相关信息独立提取单词、字符以及词汇等特征，然后基于 m 维特征构建一个图，每个维度特征都作为一个节点，最后使用图卷积网络对恶意软件进行分类。无论是提取控制流图还是提取 API 调用图，都可能受到代码混淆和数据加固的影响，导致提取的图不准确或不完整。同时，对于复杂的恶意软件行为，控制流图、数据流图以及 API 调用图的表示可能会变得非常庞大和复杂，导致模型难以有效地学习和分析。

上述方法都是挖掘单个 Android 应用程序内部各组件之间的关系，将其建模为图结构数据，还有一些方法是将不同 Android 应用程序及其所含组件构建异质信息网络（HIN），挖掘各个应用程序和组件之间更深层的语义和拓扑关系。Gao 等人[32]提出了 GDroid，这是一种基于图卷积网络的新方法。该方法的整体思路是将 Android 应用程序和提取出的 API 构建异质信息网络，Android 应用程序及包含的 API 作为节点，Android 应用程序与 API 之间的调用关系以及 API 之间的邻近关系作为边，然后将初始问题转换为节点分类任务。HinDroid[33]将 Android 应用程序、相关 API 以及它们之间丰富的关系构造为异质信息网络，利用元路径描述 APP 和 API 之间的语义关联，然后使用多核学习算法对 Android 应用程序进行分类。Hei 等人[8]基于异质图注意力网络提出了一种快速的 Android 恶意软件检测方法—HAWK，通过提取静态特征（权限、API、接口等）构建异质信息图，并从异质信息图中学习邻居节点的特征对 APP 节点进行节点嵌入，然后进行分类。尚等人[34]提出了一种基于 MLSTM(Mean and Long Short Term Memory)的安卓恶意软件检测模型，通过对基于 GraphSage 框架的改进，提出了一种 MLSTM 聚合器，以确保在对图中目标节点的邻居信息进行聚合时，能够具备良好的泛化能力和表达能力。

以上使用图神经网络的方法中，尤其是使用异质图神经网络的方法中，存在对于异质图中节点和边的信息利用不充分的问题，导致一些模型需要使用大量的安卓应用程序样本训练才能获得一个好的检测或者分类效果。同时，对于安卓恶意软件的分类任务中存在的家族之间样本数量不平衡问题，目前还没有利用图神经网络相关模型来解决的方法。本论文将针对上述问题开展进一步研究。

（3） 动态方法及混合方法

通过在实际或者虚拟环境运行 Android 应用程序，可以获得 API 调用序列、系统调用、内核调用以及动态数据流等动态特征。Mariconti 等人[35]提出了 Mamadroid，它根据马尔可夫链从抽象的 API 调用序列中构建行为模型，并利用这个行为模型来提取特征和对 Android 应用进行分类。Enck 等人[36]提出了 TaintDroid，该方法通过跟踪私有数据是否在传播过程中超出预设系统边界来确定私有数据是否泄露，从而确定是否为恶意软件。Wang 等人[37]提出了一种使用网络流量文本语义的自动恶意软件检测方法，将 Android 应用程序生成的每个 HTTP 流视为一个文本文档，通过自然语言处理的方法提取文本级特征。Hou 等人[38]提出了 Deep4MalDroid，一种新的动态分析方法——组件遍历，它可

以自动尽可能完整地执行每个给定的应用程序，并基于提取的 Linux 内核系统调用特征构建加权有向图，然后使用基于图特征的深度学习模型来检测未知的 Android 应用程序。Surendran 等人[39]提出了一种新颖的基于图信号的低维特征表示和提取机制，解决了基于系统调用的特征表示具有较高维度和缺乏系统调用依赖关系的问题。Wang 等人[40]提出了一种基于多维内核特征的恶意软件检测框架，通过对 Android 系统中执行任务时数据结构的内核属性进行分类和理解，实现了高精度的恶意软件检测。Xiao 等人[41]提出了一种基于深度学习的新型检测方法，该方法考虑到系统调用序列中存在一定的语义信息，将系统调用序列视为自然语言中的句子，并构建了基于长短期记忆（LSTM）语言模型的分类器。John 等人[42]提出了一种使用图卷积网络的新型 Android 恶意软件检测机制，将系统调用建模为图形来捕获系统调用之间的结构依赖关系，并使用图的中心性度量作为输入特征。Alzaylaee 等人[43]提出了 DL-Droid，利用应用程序的执行路径和环境状态动态生成输入数据，使用深度学习方法进行动态分析。实验证明，DL-Droid 在检测性能和代码覆盖率方面优于传统的机器学习方法。Han 等人[44]开发了一种基于特征转换的 Android 恶意软件检测器——FARM，首先获取静态特征、API 调用特征以及动态特征，利用三种不同的特征转换方法进行特征转换，然后使用随机森林算法对 Android 样本进行分类。

这些方法在安卓恶意软件检测分类方面取得了进展，但仍面临挑战。动态特征提取方法虽然可以捕获应用程序的动态行为，但可能受到样本覆盖不足、运行时开销大、特征提取精度不足等问题的影响。另一方面，静态特征和动态特征的融合需要合理的特征选择和组合策略，将静态和动态特征进行整合和加权，以确保提取的特征能够充分反映恶意软件的行为特征。

1.3 论文主要研究内容

本文的主要研究内容分为两部分：第一部分提出了一种基于元路径聚合图神经网络的新型安卓恶意软件方法——MAAMD（Metapath Aggregated Android Malware Detection），通过引入元路径聚合图神经网络[45]（MAGNN）来解决对异质图中的信息利用不充分的问题，MAAMD 能够对整条元路径进行嵌入，聚合整条元路径上的节点特征来获得更多有助于安卓恶意软件检测的信息，在样本数量有限的条件下获得更好的检测效果。第二部分提出了一种基于混合图神经网络的安卓恶意软件分类方法，来主要解决安卓恶意软件家族分类过程中不同家族之间存在的样本分布不平衡问题。利用