基于深度学习的高鲁棒性恶意软件识别研究

汇报人:

2024-01-23

目录

CONTENCT

01

引言

研究背景与意义

恶意软件数量激增

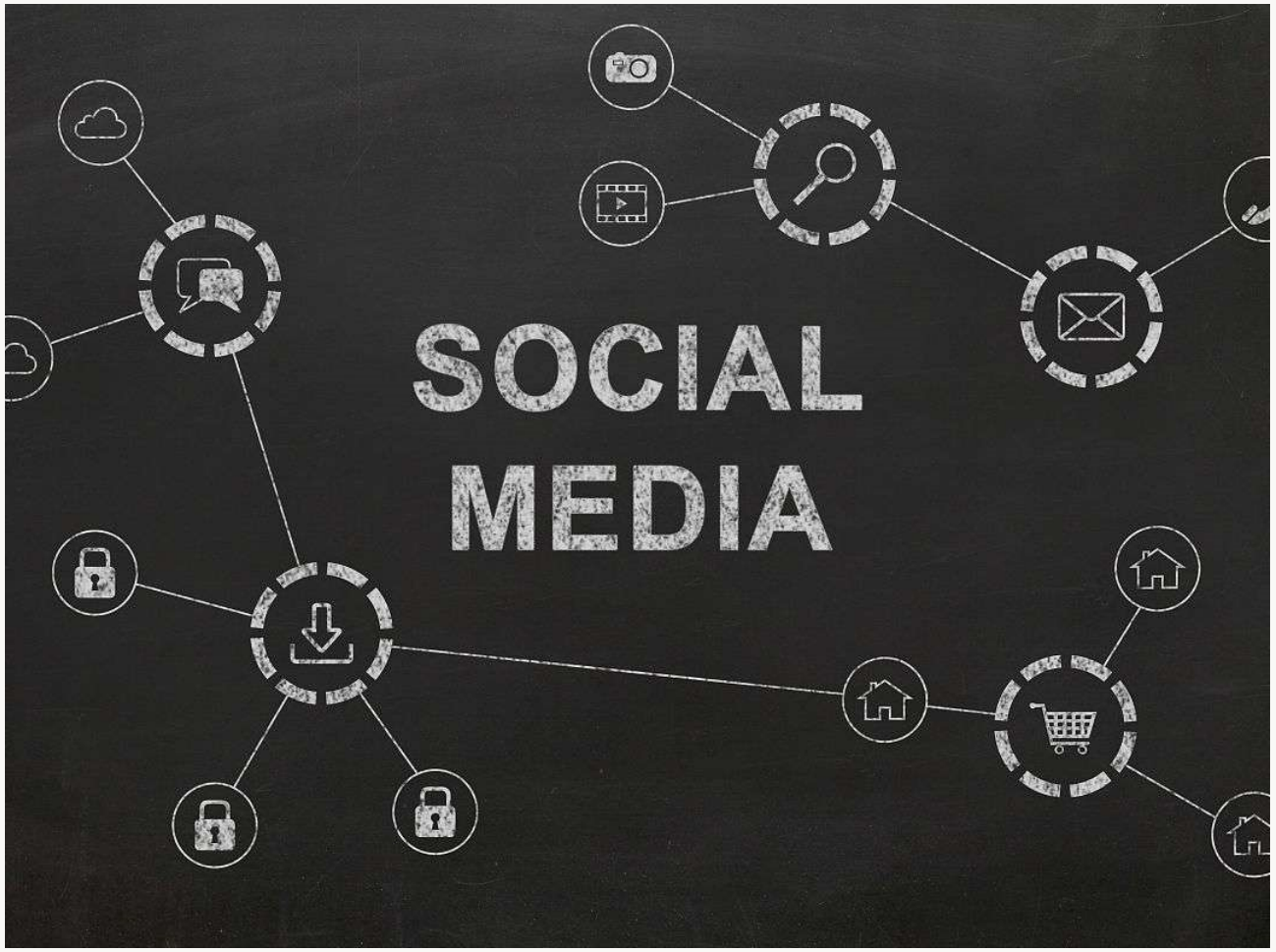
随着互联网和移动设备的普及，恶意软件数量呈指数级增长，对用户数据和系统安全构成严重威胁。

传统识别方法局限性

传统恶意软件识别方法主要基于静态特征或动态行为分析，但易受到混淆技术、加密技术等干扰，导致误报率和漏报率较高。

深度学习技术优势

深度学习技术能够从大量数据中自动提取有效特征，并处理复杂的非线性关系，为恶意软件识别提供了新的解决方案。



国内外研究现状及发展趋势

国内研究现状

国内在恶意软件识别领域已取得一定成果，如基于深度学习的恶意软件分类、恶意行为检测等。但仍存在数据集规模较小、模型泛化能力不足等问题。

国外研究现状

国外在恶意软件识别领域的研究相对成熟，提出了多种基于深度学习的恶意软件识别模型，并在大规模数据集上进行了验证。

发展趋势

未来恶意软件识别研究将更加注重模型的泛化能力和实时性能，同时结合其他安全技术，形成更加完善的恶意软件防御体系。

研究内容、目的和方法

研究内容

本研究旨在利用深度学习技术，构建高鲁棒性的恶意软件识别模型。具体内容包括数据预处理、特征提取、模型构建和评估等。

研究目的

通过本研究，期望提高恶意软件识别的准确率和效率，降低误报率和漏报率，为恶意软件防御提供更加有效的技术支持。

研究方法

本研究将采用深度学习中的卷积神经网络（CNN）、循环神经网络（RNN）等技术进行恶意软件识别。首先收集并预处理恶意软件和正常软件样本，然后提取有效特征并构建识别模型。最后通过大量实验验证模型的性能和鲁棒性。

02

恶意软件识别技术概述

恶意软件定义与分类

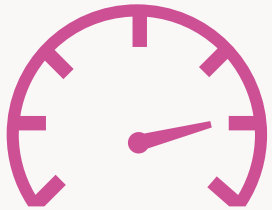
恶意软件定义

恶意软件（Malware）是指任何故意设计用于破坏、干扰、窃取或滥用计算机系统、网络或个人信息的软件。

恶意软件分类

根据其行为和目的，恶意软件可分为病毒、蠕虫、木马、勒索软件、间谍软件、广告软件等。

传统恶意软件识别技术



基于签名的识别

通过比对已知恶意软件的签名或特征来识别恶意软件。这种方法对于已知威胁有效，但无法应对未知威胁和变种。

基于行为的识别

通过分析软件在系统中的行为来识别恶意软件。这种方法可以识别未知威胁，但可能会产生误报和漏报。

基于启发式的识别

结合签名和行为分析，使用启发式算法来评估软件的恶意性。这种方法可以提高识别准确率，但仍然受限于已知威胁和变种。

基于深度学习的恶意软件识别技术

深度学习模型

利用深度学习模型（如卷积神经网络CNN、循环神经网络RNN等）对恶意软件进行自动特征提取和分类。这些模型可以从大量数据中学习恶意软件的特征和模式，从而实现对未知威胁的识别。

数据预处理

对恶意软件进行静态或动态分析，提取出可用于深度学习模型训练的特征，如API调用序列、网络流量、系统资源使用情况等。

模型训练与优化

使用大量标注的恶意软件和正常软件样本对深度学习模型进行训练，通过调整模型参数和结构来提高识别准确率。同时，可以采用迁移学习、集成学习等技术来进一步优化模型性能。

评估与测试

使用独立的测试数据集对训练好的深度学习模型进行评估和测试，以验证其在实际应用中的性能和准确性。

03

高鲁棒性恶意软件识别模型设计

模型整体架构设计

01

基于深度学习的恶意软件识别模型整体架构包括数据预处理、特征提取、深度学习模型训练和分类识别四个主要部分。

02

数据预处理阶段负责对原始恶意软件进行清洗、格式转换等操作，以便于后续的特征提取和模型训练。

03

特征提取阶段利用静态或动态分析技术从恶意软件中提取出具有代表性的特征，以供深度学习模型学习。

04

深度学习模型训练阶段使用大量标注好的恶意软件样本进行训练，通过反向传播算法调整模型参数，使模型能够学习到恶意软件的本质特征。

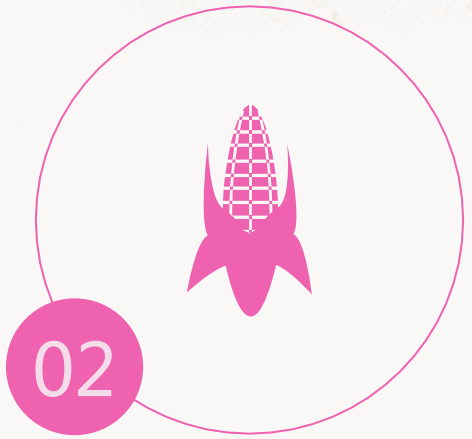
05

分类识别阶段将待检测的恶意软件输入到训练好的模型中，通过前向传播算法得到软件的分类结果。

输入数据预处理及特征提取方法

输入数据预处理

对原始恶意软件进行清洗，去除无关信息和噪声，将其转换为适合深度学习模型处理的格式，如二进制文件、图像或文本等。



静态特征提取

通过分析恶意软件的源代码或二进制代码，提取出与恶意行为相关的静态特征，如API调用、系统资源占用、文件操作等。

动态特征提取

通过运行恶意软件并监控其行为，提取出与恶意行为相关的动态特征，如网络流量、注册表操作、进程创建等。

特征选择

从提取出的特征中选择与恶意软件分类最相关的特征，以降低特征维度和提高模型训练效率。