# MalIoT：面向物联网网络的可扩展实时恶意软件流量检测

Ethan Weitkamp、Yusuke Satani、Adam Omundsen、Jingwen Wang、
Peilong Li 伊丽莎白镇学院计算机科学系 weitkampe、sataniy、
omundsena、wangjingwen、lip @etown.edu

arXiv:2304.00623v1 [cs.CR] 2 Apr 2023

**摘要**

机器学习方法对于物联网（IoT）恶意软件流量检测至关重要，因为它能够跟上恶意软件不断演变的步伐。机器学习算法能够快速、准确地分析物联网设备产生的大量数据，从而实时识别恶意网络流量。借助 Apache Kafka 和 Apache Spark 等分布式系统，该系统能够应对物联网设备的指数级增长；英特尔的 oneAPI 软件栈则可加快模型推理速度，使其成为实时恶意软件流量检测的实用工具。这些技术协同工作，构建出兼具可扩展性能和高准确率的系统，使其成为保护智慧社区和医疗机构免受网络威胁侵害的重要工具。

## 1 引言

自物联网（IoT）兴起以来，物联网设备遭受恶意软件攻击的问题始终存在。技术进步使这一问题日益严峻，攻击的规模和种类都随时间不断增加 [1]。研究显示，近期针对物联网设备的恶意攻击显著增多。Zscaler 的研究表明，在 2022年疫情期间，与企业网络相关的物联网设备遭受的恶意软件攻击增加了 700% [2]。互联网、社交网络、智能手机和物联网设备的发展，使不法分子能够制造出比以往更加先进的恶意软件。

为了防范网络攻击并维持网络稳定性，需要实时且可扩展的恶意软件检测解决方案，以识别并阻止物联网网络中的恶意软件流量。由于能够自动实时检测并应对新出现的恶意软件威胁，机器学习（ML）成为物联网恶意软件流量检测中的关键策略 [3], [4], [5]。机器学习模型可以快速扫描这些数据，以识别并防止恶意软件活动，因为物联网设备会生成海量数据，使得使用传统方法检测恶意软件变得困难。由于机器学习技术具有灵活性和适应性，因此可以持续升级模型，以便在新的恶意软件变体出现时进行检测。此外，机器学习模型能够从海量数据中学习并识别数据中可能表明恶意软件活动的模式，从而使其能够检测此前未被识别的恶意软件形式。

然而，目前基于机器学习的网络流量恶意软件检测方案往往难以应对不断增长的物联网设备数量，也难以低延迟地检测恶意流量。为解决这些问题，本文提出了一种可扩展的端到端网络流量分析系统，能够实时检测恶意软件。该系统采用 Apache Kafka 和 Apache Spark 等分布式系统，随着物联网设备数量增长，仍可高效扩展。此外，研究表明，在机器学习和深度学习模型中使用英特尔 oneAPI 软件栈，可将模型推理速度提升至原来的三倍。具体而言，本文的主要贡献有三点。(1) 通过扩充“IoT-23”数据集并加入更多样化的数据集“ToN IoT”，解决以往研究中的类别不平衡和模型过拟合问题。(2) 使用英特尔 oneAPI 软件加快恶意软件流量检测的推理速度。(3) 通过构建大数据平台提升系统的可扩展性和响应能力，该平台将 Apache Kafka 用作流处理引擎，并将 Apache Spark 用作数据处理器。

本文的组织结构如下。第 §2 节总结了以往关于物联网恶意软件检测的研究工作。第 §3 节介绍了数据管道的整体设计。第 §4 节评估了各方法的性能。最后，第 §5 节对全文进行了总结。

## 2 相关工作

检测恶意软件流量的一种传统方法是采用基于特征码的检测方法[6], [7], , 该方法使用已知恶意软件特征码数据库来识别并阻止与这些特征码匹配的流量。尽管这种方法能够有效检测已知恶意软件，但其最大的局限性之一是无法检测未知攻击，而且无法覆盖大规模攻击的检测。

为解决这一问题，提出了基于网络行为的检测方法，用于

恶意软件流量检测 [8]。行为建模方法侧重于根据恶意流量的行为，而非其特征码，来识别并拦截恶意流量 [9], [10], [11]。这种方法在检测新型或未知恶意软件方面更为有效，因为它不依赖已知恶意软件特征码数据库。不过，这种方法可能需要更多资源，也可能产生误报。该领域的大多数现有研究范围有限，侧重于特定类型的恶意软件（如僵尸网络）、特定类型的攻击（如拒绝服务攻击），以及特定协议或网络层中的异常 [12]。传统检测方案的另一项局限是依赖单一网络场所，因此无法检测源自其他网络场所的攻击。另一个问题是，设备感染后仍可能继续运行很长时间。

恶意软件流量检测中使用的另一种技术是沙箱技术，即创建一个虚拟环境来分析和测试疑似恶意软件 [13], [14]。这使分析人员能够安全地观察恶意软件的行为并判定其潜在威胁。

先前的工作将机器学习方法引入恶意软件流量检测 [15]。Kaluphahana 等人[16] 提出了 Adept，这是一种安全框架，可检测僵尸网络攻击，并按照时空维度将其划分为不同的攻击阶段。Adept 利用警报级和模式级信息，通过 k 近邻、随机森林和支持向量机等机器学习模型，对攻击类型和阶段进行分类。此外，研究人员提出了端到端监控系统 RTC，通过手动提取不同协议和网络层流量数据中的特征来识别新威胁 [17]。然而，机器学习方法严重依赖人工特征提取，这是一个耗时且需要大量资源的过程。

由于深度学习方法在特征提取和自动学习方面具有高效性，恶意软件流量攻击检测的研究正从机器学习转向深度学习 [18], [19], [20], [21]。Sahu 等人 [22] 提出了一种采用两阶段混合深度学习模型的物联网攻击检测安全框架。具体而言，卷积神经网络模型首先从物联网网络流量中学习特征，然后利用上一步生成的特征表示被用作 LSTM模型进行攻击检测的输入。然而，每当通过添加额外的 IoT 设备来扩展网络时，都需要配备一个额外的卷积神经网络模块及与其连接的网络交换机。

表 1 展示了机器学习和深度学习模型在物联网系统中对大规模攻击性访问进行分类时的性能。Goyal 等人 [23] 观察到，支持向量机和人工神经网络模型生成的准确率得分相近，而人工神经网络模型的精确率得分比支持向量机高出 4.3%。长短期记忆网络（LSTM）用于根据相对少量的数据确定能否检测到攻击或良性攻击。Liang 等人 [24] 得出结论：支持向量机模型的准确率得分为 88.18%，但召回率和 F1 分数均低于 50%。

表 1：综述中的模型准确率

| 参考文献 | 模型 | 准确率 | 精确率 | 召回率 | F1 |
|---|---|---|---|---|---|
| Goyal 等人 [23] | ANN | 99.74 | 95.99 | 100 | 97.95 |
| | SVM | 99.86 | 91.98 | 100 | 95.82 |
| Liang 等人 [24] | LSTM | - | 99.98 | 100 | 99.99 |
| | SVM | - | 88.18 | 45.43 | 59.97 |

另一方面，LSTM模型的准确率得分为 99.98%，召回率和 F1 分数均接近 100。

## 3 设计

本节将介绍 MalIoT 系统的设计，包括整体数据管道、数据收集与生成、离线模型训练以及在线模型推理。

### 3.1 数据管道概述

整体数据管道如图 1 所示。信息源自局域网中的物联网设备，这些设备通过智能网关传输其网络活动，而后续处理步骤均在该网关上执行。首先，智能网关上的软件嗅探器 Bro 会拦截网络活动，并根据接收到的数据生成 PCAP 文件。随后，这些 PCAP 文件会发送至 Kafka 生产者，由其将数据传输到 Kafka 主题。Kafka生产者确保数据处理流水线能够扩展，以接入网络中的更多物联网设备。然后，使用 Spark 流处理从 Kafka 主题摄取数据，从而使数据处理流水线能够适应任意数量的物联网设备。之后，可以利用这些数据重新训练机器学习或深度学习模型，也可以使用先前训练的机器学习或深度学习模型对网络活动进行在线推理。

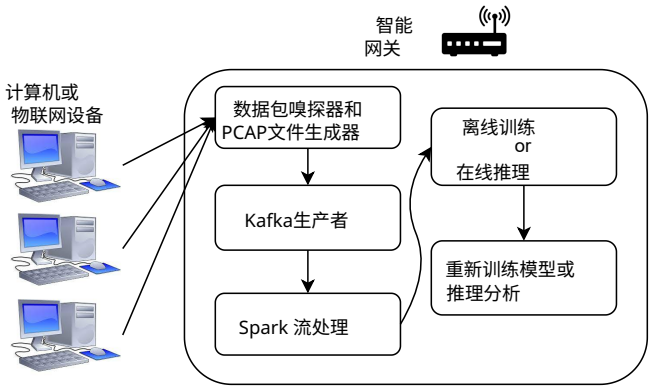



图1：整体设计流程

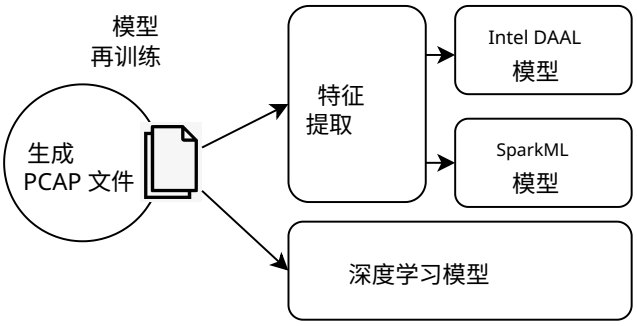



图2：模型训练流程

### 3.2 数据生成

要训练各种模型，我们需要一个初始数据集，以及一种利用新数据重新训练模型的方法，以适应新型攻击。重新训练过程如图 2 所示。数据包嗅探器生成 PCAP 文件后，可利用其中的实时网络流量重新训练模型。我们采用 IoT-23 数据集和 TON IoT 数据集作为训练初始模型的起始数据集。我们使用数据集提供的各种特征子集测试了机器学习和深度学习模型的准确率，并选择使用大部分特征，但不包括主机和接收方 IP 地址及端口。这些属性并未提升模型准确率，因为它们不应与恶意或良性攻击相关联，否则只会导致模型过拟合。使用现有数据集建立基础模型后，可根据需要利用新生成的 PCAP 文件对其进行重新训练。对于机器学习模型，重新训练前必须从 PCAP 文件中提取所需特征，并执行其他数据预处理步骤。

**数据集**：Aposemat IoT-23 数据集 [25] 采集自物联网（IoT）设备的网络流量。该数据集于2020年1月首次发布，涵盖了2018年至2019年的流量捕获数据。IoT-23 数据集包含从物联网设备捕获的20个恶意软件实例，以及3个良性物联网设备流量实例。

ToN_IoT 数据集 [26] 是新一代物联网（IoT）和工业物联网（IIoT）数据集，采集自一个包含异构数据源的真实、大规模网络。它由以下数据集组成：物联网和工业物联网传感器数据集、Windows 7 和 10 操作系统数据集、Ubuntu 14 和 18TLS 操作系统数据集，以及网络流量数据集。

各数据集的基本信息列于表 2。IoT-23 数据集包含来自不同物联网网络流量的 325,307,990 条捕获记录，其中包括在受感染的物联网设备上执行的恶意软件捕获记录 294,449,255 条，以及来自良性物联网设备流量的捕获记录 30,858,735 条。在 ToN IoT 数据集中，Windows 7 数据集包含 28,366 条记录和 132 个特征，Windows 10 数据集包含 35,975 条记录和 124 个特征，而网络数据集包含 21,978,632 条记录和 42 个特征。为生成最终合成训练数据集，我们将 TON IoT 与 IoT-23 合并，然后用来自 CIC IoT 数据集 [27] 的额外良性流量补充合成数据集，以平衡各分类类别。

表 2：数据集信息

| 数据集 | IoT-23 | TON_IoT |
|---|---|---|
| 良性 | 30,858,735 | 300,000 |
| 异常 | 294,449,255 | 161,043 |
| 总计 | 325,307,990 | 461.043 |
| 特征数量 | 23 | 45 |
| 物联网设备数量 | 3 | 9 |

**特征**：表 3 列出了两个数据集共有的特征。在实验中，我们为机器学习模型使用了两种特征选择方式：1）“完整特征集”，包含除时间戳和唯一标识符之外的所有特征；2）“去标识化特征集”，删除通信发起端和响应端的 TCP/UDP IP 地址及端口信息。我们删除这些信息，以避免机器学习模型根据 IP 地址和端口对恶意流量进行分类时产生潜在偏差。

表 3：IoT-23 和 TON IoT 特征选择

| IoT-23 特征 | TON_IoT 特征 | 描述 |
|---|---|---|
| ts | ts | 连接时间戳 |
| id.orig_h | 源 IP 地址 | 发起方 IP 地址 |
| id.orig_p | 源端口 | 发起方 TCP/UDP 端口 |
| id.resp_h | 目标 IP 地址 | 响应方 IP 地址 |
| id.resp_p | 目标端口 | 响应方的 TCP/UDP 端口 |
| 协议 | 协议 | 连接的传输层协议 |
| 服务 | 服务 | 服务 |
| 持续时间 | 持续时间 | 持续时间 |
| 发起方字节数 | 源字节数 | 发起方有效载荷字节数 |
| 响应方字节数 | 目标字节数 | 响应方负载字节数 |
| 连接状态 | 连接状态 | 连接状态 |
| 丢失字节数 | 丢失字节数 | 丢失字节数 |
| 原始数据包数 | 源数据包数 | 原始数据包数 |
| 原始 IP 字节数 | 源 IP 字节数 | 原始 IP 字节数 |
| 响应数据包数 | 目标数据包数 | 响应数据包数 |
| 响应 IP 字节数 | 目标 IP 字节数 | 响应 IP 字节数 |

### 3.3 离线模型训练

随着时间推移，新的恶意攻击不断出现并被设计出来。曾经被认为安全的事物，必须不断适应，才能在当下和未来维持安全。这在机器学习领域尤为如此。为了确保我们的预测模型始终安全，必须使用更新的数据对其进行训练。为此，我们集成了一种机制，可以利用自行生成的数据重新训练各种模型。经过数据处理流水线的数据会以一种便于日后重新训练模型的格式保存。因此，我们不必仅仅依赖现有数据集来保持模型的更新。

我们评估了五种机器学习模型和五种深度学习模型。机器学习模型分类器包括随机森林、决策树、逻辑回归、线性支持向量分类器和高斯朴素贝叶斯。深度学习模型包括

人工神经网络（ANN）、一维卷积神经网络（1DCNN）、二维卷积神经网络（2DCNN）、长短期记忆网络（LSTM），以及卷积神经网络与LSTM的组合。每种模型都有其独特的优势，这促使我们决定对它们进行评估。对于深度学习模型的超参数，除LSTM模型外，我们对所有人工神经网络和卷积神经网络模型都使用ReLU作为激活函数；LSTM模型则使用“双曲正切函数”作为激活函数，并使用Sigmoid函数作为循环激活函数。人工神经网络仅使用一层，而其他模型均使用两层。关于模型超参数的更多详情见表4。

表 4：深度学习超参数

| 参数 | 值 |
|---|---|
| 学习率 | 1e-3 |
| 衰减率 | 1e-5 |
| 丢弃率 | 0.5 |
| 全连接层单元数 | 128 |
| n_批次 | 100 |
| n_训练轮次 | 1 |
| 卷积核 | 卷积核 |
| 卷积核大小 | 4 |
| 步幅 | 1 |
| 卷积神经网络层 | 2 |
| 分类器 | 1e-5 |
| 回归器 | |

### 3.4 在线推理

网络流量被转换为 PCAP 文件并由 Spark 流处理后，我们之前训练的模型便可对这些数据进行推理。如前文所述，我们训练了五种机器学习模型和五种深度学习模型。

进入数据处理流水线的 Spark 流处理阶段后，所有新数据都会根据其时间戳由其中一个训练好的模型进行评估。机器学习模型通常需要更长的预处理时间，因为必须进行特征提取；但由于这些模型并不复杂，其推理/预测时间较短。另一方面，深度学习模型的预处理时间较短，但由于模型复杂，其推理时间较长。这些模型往往能在复杂数据上取得更高的准确率，例如我们数据处理流水线中的数据。目前，数据处理流水线会在网络流量完成一次推理后结束，但所得的推理结果可以按需使用。

## 4 评估

本节评估我们所提出系统的性能。具体而言，我们将介绍测试平台的规格，以及每个机器学习和深度学习模型的准确率与时间性能。

### 4.1 实验测试平台

我们在表5中总结了测试平台的系统规格和软件版本。我们使用 Python 的内置工具来收集数据，例如训练时间、推理时间、准确率和 CPU 使用率。

表5：实验平台规格

| Item | 规格 |
|---|---|
| CPU | 英特尔® 酷睿™ i9-9920X 处理器 @ 3.50GHz |
| GPU | GTX 2080 Ti，配备 11 GB DDR6 内存 |
| 内存 | 64 GB DDR4 @ 3600MHz |
| 存储 | 2TB 固态硬盘 |
| 主机操作系统 | Ubuntu 18.04 LTS |
| TensorFlow | 2.7 |
| Apache Spark | 3.0.1 |
| Apache Kafka | 2.6.0 |
| 加速器 | Intel DAAL v2020.1 |

### 4.2 模型准确率比较

我们计算了机器学习模型和深度学习模型的预测准确率。在机器学习推理部分，我们设计了五种机器学习模型（随机森林、决策树、逻辑回归、线性支持向量分类器和高斯朴素贝叶斯）以及两种特征集：完整特征集和去标识化特征集。

图 3 展示了机器学习模型的结果。此外，图 4 展示了五种深度学习模型的结果，分别是人工神经网络、卷积神经网络、二维卷积神经网络、长短期记忆网络，以及卷积神经网络与长短期记忆网络的组合。

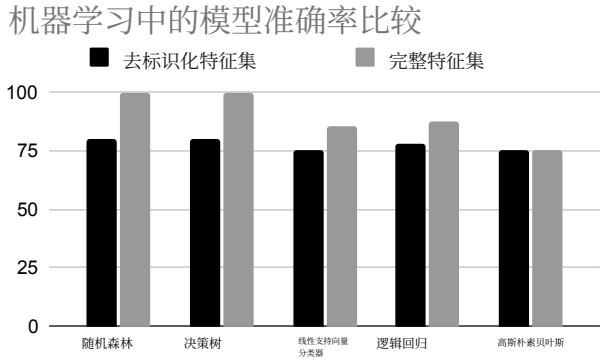



图 3：机器学习中的模型准确率比较

在机器学习实验中，完整特征集通常得分更高。前两种模型——随机森林和决策树——几乎达到了 100%。总体而言，去标识化特征集的得分为 75%。

在深度学习部分，前三个模型的准确率接近 100%。而 LSTM 和 LSTM+卷积神经网络的得分约为 75%。

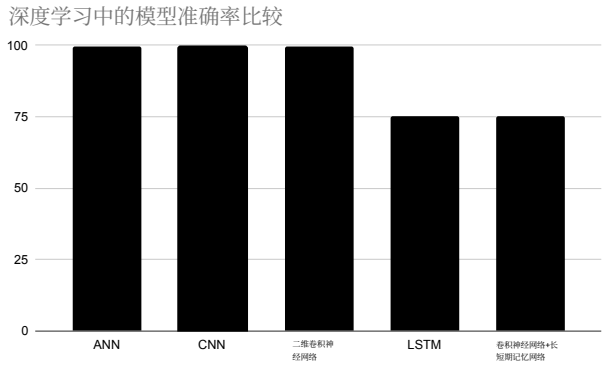



图 4：深度学习中的模型准确率比较

表 6：两种不同特征集上机器学习模型的推理时间

| | 去标识化特征集（毫秒） | 完整特征集（毫秒） |
|---|---|---|
| 随机森林 | 33.18 | 34.20 |
| 决策树 | 10.40 | 10.78 |
| 逻辑回归 | 10.30 | 10.71 |
| 线性支持向量分类器 | 10.30 | 10.68 |
| 高斯朴素贝叶斯 | 10.48 | 10.87 |

## 4.3 推理时间比较

我们测量了从 CSV 文件读取每一行数据进行推理所需的平均时间。与之前的实验一样，我们测试了机器学习和深度学习。图 5 展示了机器学习模型的结果，图 6 展示了深度学习模型的结果。

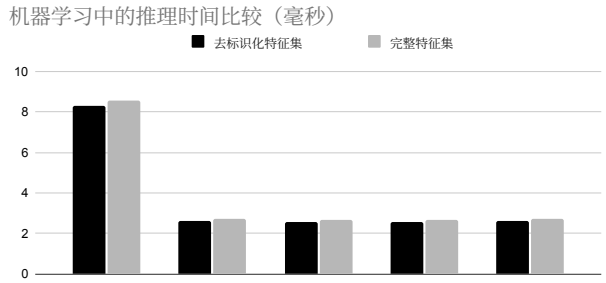



图5：机器学习中的推理时间比较（毫秒）

该实验表明，随机森林模型耗时比其他模型更长。原因似乎是随机森林并行运行决策树。其他模型逐行推理的耗时约为 2.6 毫秒。此外，完整特征集的预测耗时略长于去标识化特征集。

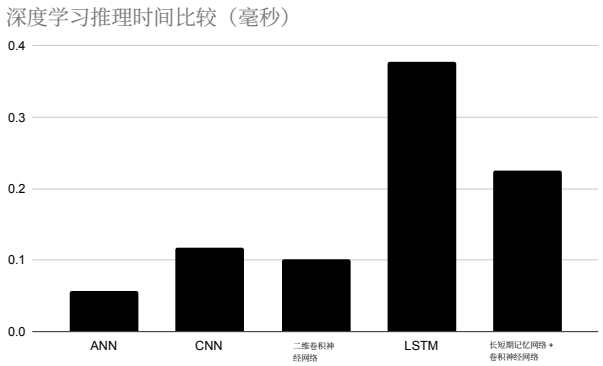



图6：深度学习中的推理时间比较（毫秒）

实验结果表明，所有深度学习模型的推理速度都远快于机器学习模型。LSTM模型比其他模型稍慢，但每行仅慢0.4毫秒

表7：深度学习模型的推理时间

| | 每行时间（毫秒） |
|---|---|
| ANN | 0.056 |
| CNN | 0.118 |
| 二维卷积神经网络 | 0.102 |
| LSTM | 0.378 |
| 卷积神经网络+长短期记忆网络 | 0.226 |

## 4.4 可扩展性

我们研究的另一个关键部分是可扩展性。我们测量了支持设备数量变化对推理时间的影响。在测试中，我们将设备数量从1台逐步增加到9台。图7展示了试验结果。

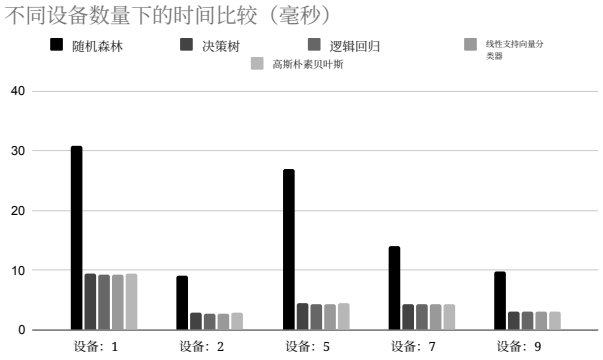



图7：不同设备数量下的时间比较（毫秒）

实验表明了我们模型的可扩展性。通常，随着联网设备数量逐渐增加，每行的推理时间会缩短。与其他模型相比，随机森林的预测耗时更长，这与前一次实验的结果相同。

## 5 结论

在我们的项目中，通过使用 Apache Kafka生产者和 Apache Spark，我们实现了实时且可扩展的恶意网络流量检测目标。我们的模型基于扩展后的数据集进行训练，该数据集包含 IoT-23 和 TON IoT，其中含有数百万条网络流，并包含恶意和良性网络流量的信息。在对数据进行预处理后，我们使用开发的各种模型对其执行了推理。此过程可以在智能网关上实现自动化，从而支持对局域网进行实时检测。我们已将该项目的源代码发布在 GitHub 上，以供未来研究使用。可通过以下链接访问：https://github.com/BlueJayADAL/NetSec。

## 致谢

本研究部分得到了英特尔公司的资助，以及伊丽莎白镇学院暑期奖学金、创意艺术与研究项目（SCARP）的资助。

## 参考文献

[1] D. Gibert、C. Mateu、J. Planes，《机器学习在恶意软件检测与分类中的兴起：研究进展、趋势与挑战》，网络与计算机应用期刊，*153*卷（2020），102526。

[2] Zscaler 的研究证实，物联网设备是安全受损的重要来源，这进一步凸显了零信任安全的必要性。[3] C. Jindal、C. Salls、H. Aghakhani、K. Long、C. Kruegel、G. Vigna，《Neurlux：无需特征工程的动态恶意软件分析》，载于第*35*届年度计算机安全应用大会论文集，第444–455页（2019年）。[4] R. Vinayakumar、M. Alazab、K. Soman、P. Poornachan- dran、S. Venkatraman，《使用深度学习进行稳健的智能恶意软件检测》，*IEEE Access*，*7*：（2019年），46717–46738。[5] Y. Fang、Y. Gao、F. Jing、L. Zhang，《基于DEX文件节特征的安卓恶意软件家族分类》，*IEEEAccess*，*8*：（2020年），10614–10627。[6] S. Hajiheidari、K. Wakil、M. Badri、N. J. Navimipour，《物联网中的入侵检测系统：全面调查》，《计算机网络》，*160*：（2019年），165–191，ISSN 1389-1286，doi:https://doi.org/10.1016/j. comnet.2019.05.014。[7] M. Masdari、H. Khezri，《基于模糊签名的入侵检测系统调查与分类》，《应用软计算》，*92*：（2020年），106301。[8] S. Nari、A. A. Ghorbani，《基于网络行为的自动恶意软件分类》，*2013*年国际计算、网络与通信会议（*ICNC*），第642–647页。

[9] ¨O. Aslan、M. Ozkan-Okay、D. Gupta，《云计算环境中的智能行为型恶意软件检测系统》，*IEEE Access*，*9*：（2021年），83252–83271。

[10] H. S. Galal、Y. B. Mahdy、M. A. Atiea，《用于恶意软件检测的基于行为的特征模型》，计算机病毒学与黑客技术期刊，*12*（2）（2016），59–67。

[11] W.Liu, P. Ren, K. Liu, H.-x.Duan，基于行为的恶意软件分析与检测，第39–42页。

[12]B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. A. Kemmerer, C. Kr¨ugel, G. Vigna，你的僵尸网络就是我的僵尸网络：僵尸网络接管分析，发表于计算机与通信安全会议（2009年）。[13]N. Miramirkhani, M. P. Appini, N. Nikiforakis, M. Polychronakis，无痕沙箱：利用磨损痕迹规避恶意软件分析系统，发表于*2017*年 *IEEE* 安全与隐私研讨会（*SP*），第1009–1024页（IEEE，2017年）。

[14] S. Liu, P. Feng, S. Wang, K. Sun, J. Cao，通过模拟用户行为增强恶意软件分析沙箱，《计算机与安全》，*115*：（2022年），102613。

[15] O. Barut, M. Grohotolski, C. DiLeo, Y. Luo, P. Li, T. Zhang，基于机器学习的加密流量恶意软件检测：全面的性能研究，载于第7届网络、系统与安全国际会议，第7届 NSysS 2020，页码 45–55（计算机协会，美国纽约州纽约，2020年），ISBN 9781450389051，doi:10.1145/3428363.3428365。

[16]K. L. K. Sudheera, D. M. Divakaran, R. P. Singh, M. Gurusamy, Adept：在物联网网络中检测和识别相关攻击阶段，*IEEE*《物联网期刊》，*8*(8)：(2021年)，6591–6607。

[17] D. Bekerman, B.Shapira, L. Rokach, A. Bar，使用网络流量分类检测未知恶意软件，*2015*年 *IEEE*通信与网络安全会议 *(CNS)*，页码 134–142。

[18] O. Barut、Y. Luo、T. Zhang、W. Li、P. Li，基于多任务层次化学习的网络流量分析，载于*ICC 2021 - IEEE*国际通信会议，第 1–6 页（2021年），doi:10.1109/ICC42927.2021.9500546。

[19] O. Barut、T. Zhang、P. Li，面向网络流量分类的弱监督学习，载于*NAS 2022 - IEEE*国际网络、架构与存储会议，NAS '22，第 1–4 页（美国宾夕法尼亚州费城，2022年）。

[20] O. Barut、Y. Luo、P. Li、T. Zhang，R1dit：基于注意力机制的神经网络的隐私保护型恶意软件流量分类，*IEEE* 网络与服务管理汇刊，第 1–1 页，doi:10.1109/TNSM.2022.3211254。

[21] T.-L. Huoh, Y. Luo, P. Li, T. Zhang，基于流的加密网络流量图神经网络分类，*IEEE* 网络与服务管理汇刊，第 1–1 页，doi:10.1109/TNSM.2022.3227500。

[22] A. K. Sahu、S. Sharma、M.Tanveer、R. Raja，使用混合深度学习模型检测物联网攻击，计算机通信，*176*:(2021年)，146–154，ISSN 1873703X，doi:10.1016/j.comcom.2021.05.024。

[23] M. Goyal、I. Sahoo、G. Geethakumari，使用网络流量分析检测物联网设备中的 HTTP 僵尸网络，载于*2019*年能源高效计算与通信最新进展国际会议（*ICRAECC*），第 1–6 页（IEEE，2019 年）。[24] X. Liang、T. Znati，一种支持长短期记忆的 DDoS 检测框架，载于*2019*年 *IEEE* 全球通信大会（*GLOBECOM*），第 1–6 页（IEEE，2019 年）。[25] S. Garcia、A. Parmisano、M. J. Erquiaga，IoT-23：包含恶意和良性物联网网络流量的带标签数据集（版本 1.0.0） [数据集]，doi:http://doi.org/10.5281/zenodo. 4743746。[26] N. Moustafa，一种用于在边缘评估基于人工智能的安全系统的新型分布式架构：Network Ton 物联网数据集，《可持续城市与社会》，*72*:(2021年)，102994。[27] S. Dadkhah、H. Mahdikhani、P. K. Danso、A. Zohourian、K. A. Truong、A. A. Ghorbani，迈向真实多维物联网画像数据集的开发，*2022*年隐私、安全与信任年度国际会议（*PST*），第 1–11 页。