软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software [doi: 10.13328/j.cnki.jos.007080]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

面向小样本的恶意软件检测综述*

刘 昊, 田志宏, 仇 晶, 刘 园, 方滨兴

(广州大学 网络空间安全学院, 广东 广州 510799)

通信作者: 田志宏, E-mail: tianzhihong@gzhu.edu.cn

摘 要: 恶意软件检测是网络空间安全研究中的热点问题, 例如 Windows 恶意软件检测和安卓恶意软件检测等. 随着机器学习和深度学习的发展, 一些在图像识别、自然语言处理领域的杰出算法被应用到恶意软件检测, 这些算法在大量数据下表现出了优异的学习性能. 但是, 恶意软件检测中有一些具有挑战性的问题仍然没有被有效解决, 例如, 基于少量新颖类型的恶意软件, 常规的学习方法无法实现有效检测. 因此, 小样本学习 (few-shot learning, FSL) 被用于解决面向小样本的恶意软件检测 (few-shot for malware detection, FSMD) 问题. 通过相关文献, 提取出 FSMD 的问题定义和一般流程. 根据方法原理, 将 FSMD 方法分为: 基于数据增强的方法、基于元学习的方法和多技术结合的混合方法, 并讨论每类 FSMD 方法的特点. 最后, 提出对 FSMD 的背景、技术和应用的展望.

关键词: 网络安全; 小样本学习; 恶意软件检测; 恶意行为

中图法分类号: TP311

中文引用格式: 刘昊, 田志宏, 仇晶, 刘园, 方滨兴. 面向小样本的恶意软件检测综述. 软件学报. http://www.jos.org.cn/1000-9825/7080.htm

英文引用格式: Liu H, Tian ZH, Qiu J, Liu Y, Fang BX. Survey on Few-shot for Malware Detection. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7080.htm

Survey on Few-shot for Malware Detection

LIU Hao, TIAN Zhi-Hong, QIU Jing, LIU Yuan, FANG Bin-Xing

(Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510799, China)

Abstract: Malware detection is a hotspot of cyberspace security research, such as Windows malware detection and Android malware detection. With the development of machine learning and deep learning, some outstanding algorithms in the fields of image recognition and natural language processing have been applied to malware detection. These algorithms have shown excellent learning performance with a large amount of data. However, there are some challenging problems in malware detection that have not been solved effectively. For instance, conventional learning methods cannot achieve effective detection based on a few novel malware. Therefore, few-shot learning (FSL) is adopted to solve the few-shot for malware detection (FSMD) problems. This study extracts the problem definition and the general process of FSMD by the related research. According to the principle of the method, FSMD methods are divided into methods based on data augmentation, methods based on meta-learning, and hybrid methods combining multiple technologies. Then, the study discusses the characteristics of each FSMD method. Finally, the background, technology, and application prospects of FSMD are proposed.

Key words: network security; few-shot learning (FSL); malware detection; malicious behavior

恶意软件[1,2]是攻击者实施网络攻击[3,4]的重要手段, 例如利用恶意软件进行远控或者发起勒索, 为有效保护计算机网络和系统免遭破坏, 恶意软件检测一直以来都是网络空间安全领域热点研究方向之一. 恶意软件检测的目标广泛, 既要覆盖众多软件平台, 又要针对多种类型, 因此将机器学习/深度学习算法应用于恶意软件检测一直是

* 基金项目: 国家自然科学基金 (U20B2046); 国家重点研发计划 (2021YFB2012402); 广东省高校创新团队项目 (2020KCXTD007); 广州市高校创新团队项目 (202032854)
收稿时间: 2023-04-11; 修改时间: 2023-07-17; 采用时间: 2023-10-19; jos 在线出版时间: 2024-01-24

该方向的活跃领域. 虽然结合大量训练数据, 可构建有效识别多种检测对象的模型, 但实际应用中却是新型恶意软件数量极少 (小样本), 常见的学习方法难以构建出有效的检测模型[5,6]. 由此, 探究如何在只提供少量样本的前提下有效识别未知类别的面向小样本的恶意软件检测 (few-shot for malware detection, FSMD) 方法应运而生, 并已成为当前网络空间安全领域的挑战性科学问题之一.

在图像识别领域已有大量的小样本学习 (few-shot learning, FSL) 方法[7], 并成功应用于稀有样本的学习, 例如灭绝动物图像[8]等. 显然, 现有 FSL 成果对网络空间安全领域的 FSMD 亦可形成有效借鉴, 图 1 展示了 Web of Science 数据库[9]中 2018–2022 年间的 FSMD 相关文献的关键词, 深度学习、特征工程等技术在文献中频繁出现. 目前虽已存在对小样本异常流量检测的梳理[10], 但是仍然缺乏对 FSMD 方法体系的总结和最新研究成果的整理. 因此, 有必要对 FSMD 进行回顾, 构建更加完整的 FSMD 方法体系. 本文的主要贡献包括 4 个方面.

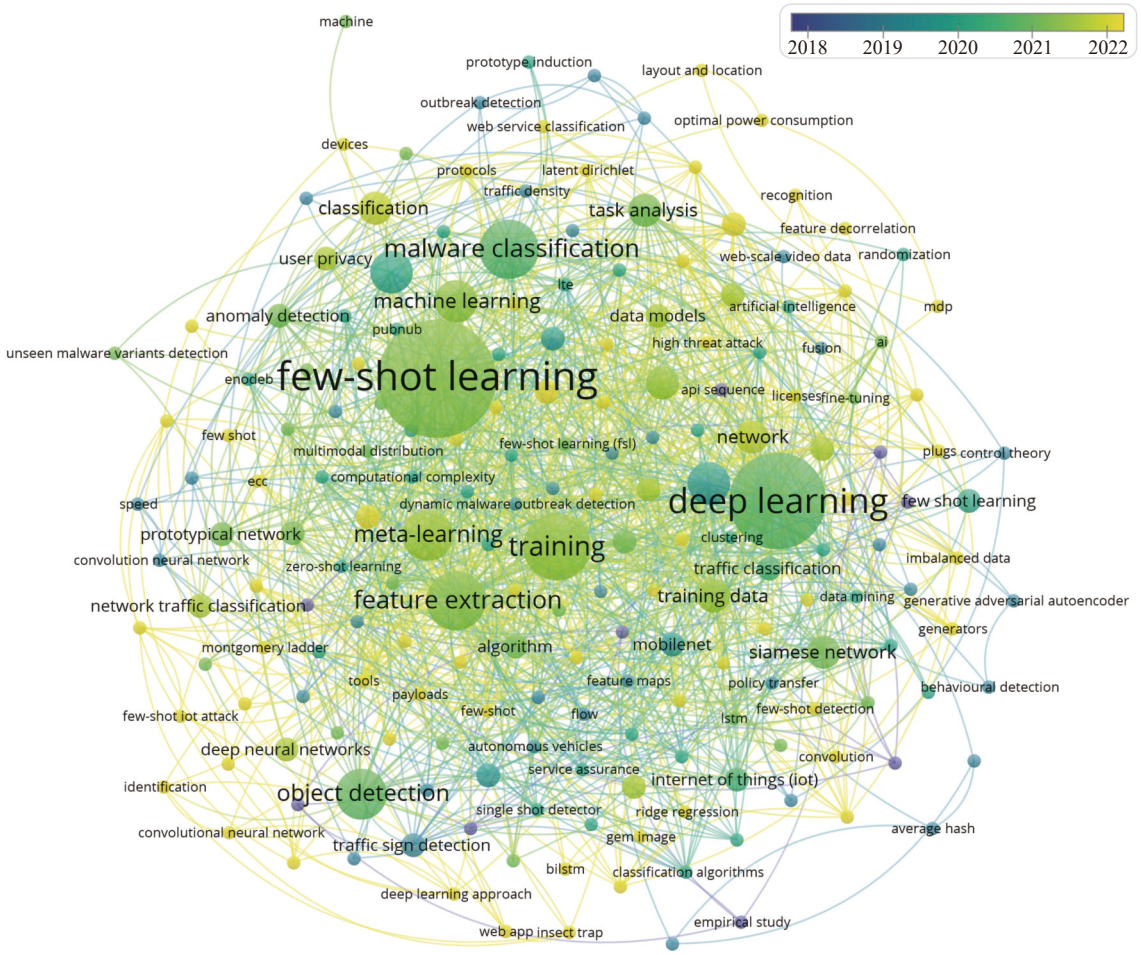

图 1　2018–2022 年 FSMD 相关文献关键词

(1) 全面收集 FSMD 相关文献与数据集, 并总结了 FSMD 问题类型、背景和目标, FSMD 问题是在可训练样本量极少的背景下, 构建一个可以有效识别小样本对象的监督学习问题.

(2) 从多种 FSMD 方法中总结出 FSMD 的一般流程, 包括: 样本集合处理、向量嵌入、向量集合处理、模型训练和样本检测. 并阐述了每个流程的作用. 结合总结的 FSMD 一般流程, 分析了 FSMD 产生的原因.

(3) 根据 FSMD 方法原理对相关文献进行了分类, 具体包括: 基于数据增强的方法、基于元学习的方法以及多技术结合的混合方法, 列举了每种方法的主要思路, 并分别介绍了每一个方法的具体实施流程.

(4) 总结了现有的 FSMD 工作, 并从问题背景、技术发展以及应用场景对 FSMD 的未来发展方向进行了展望.

本文第 1 节介绍了 FSMD 的背景、问题定义、一般流程, 列举了现有方法的检测对象、使用技术和数据集. 第 2 节介绍了基于数据增强相关技术和方法. 第 3 节阐述了基于元学习的研究. 第 4 节对使用多技术结合的混合方法进行了介绍. 第 5 节从问题假设、技术发展与应用场景这 3 个方面对 FSMD 发展方向进行了展望. 最后对本文工作进行了总结.

1 FSMD 概述

恶意软件检测主要基于人工分析和机器学习算法模型, 传统检测模型构建需要大量训练数据. 结合 MMCC2015[11] 等公开数据集和分析报告可知, 小样本在恶意软件检测领域并不是罕见现象, 例如新颖的挖矿类恶意软件[12], 功能复杂的 Simda 家族[13]以及隐蔽性强的伪装恶意下载器[14]等都是多个数据集中的小样本类. 以往的检测模型构建方法无法从少量训练数据中有效学习新型威胁的类别特性, 因此, 研究人员从多个方面提出了 FSMD 解决方法. 为更加系统化地进行阐述, 本章将从问题定义、一般流程和方法分类这 3 个方面对 FSMD 进行介绍.

1.1 相关概念与问题定义

定义 1. 恶意软件. 窃取计算机软件系统信息和影响系统功能的一系列软件的统称, 包括远控木马、勒索软件等.

定义 2. 恶意软件检测. 判断是否为恶意软件的方法, 利用识别模型判断恶意软件的类别, 识别模型的训练过程是一个监督学习[15]过程.

定义 3. 监督学习. 基于学习算法和标签明确的样本的模型建立过程. 样本的标签代表了样本的所属类别. 训练过程中, 算法构建从样本到标签的映射作为识别模型, 监督学习建立的模型可以预测测试样本的标签.

定义 4. FSL. 当监督学习的训练样本包含 N 个类别, 每个类别最多包含 K 个样本时, 称这类监督学习为 N-way K-shot FSL, 根据公式 (1), 当 K 满足不同条件时, 将学习问题分为一般监督学习、FSL 和零样本学习 (zero-shot learning, ZSL), 并且将对应学习类型覆盖的样本类型分别称为正常样本类型、小样本类型和零样本类型:

$$\begin{cases} K > 20, & \text{一般监督学习} \\ 0 < K \leqslant 20, & \text{FSL} \\ K = 0, & \text{ZSL} \end{cases} \tag{1}$$

定义 5. FSMD 问题. 若 FSL 问题中的识别对象本体为恶意软件时, 则该问题称为 FSMD 问题, 其目标是基于小样本恶意软件类构建准确识别的检测模型.

表 1 展示了关于 FSMD 的名词与缩写简介.

表 1 关于 FSMD 的名词介绍

名词	简介	符号
样本	一类数据, 通常指软件, 但不限于由软件生成的图像、序列等	s
训练样本集	用于训练模型的s集合, 总数量不超过$N\times K$	S_{train}
测试样本集	用于测试模型的s集合	S_{test}
标签	i类s的标注信息, 序号i, $i\in\{1, 2,\ldots, N\}$	l_i
标签向量	长度为N的向量, 每一维度范围[0, 1]	$\boldsymbol{l}$
训练标签集	S_{train}中s对应的l或$\boldsymbol{l}$构成的集合	L_{train}
测试标签集	S_{test}中s对应的l或$\boldsymbol{l}$构成的集合	L_{test}
样本向量	将s嵌入为同一维度向量空间中的向量, 亦称为特征	$\boldsymbol{v}$
向量维度	$\boldsymbol{v}$的长度, 序号d, $d\in\{1, 2,\ldots, D\}$	D
训练向量集	S_{train}中s对应的$\boldsymbol{v}$构成的集合	V_{train}
测试向量集	S_{test}中s对应的$\boldsymbol{v}$构成的集合	V_{test}
模型参数	模型算法与结构中待调整数据构成的向量	$\boldsymbol{p}$

表 1 关于 FSMD 的名词介绍 (续)

名词	简介	符号
初始模型参数	模型在经过训练前的$\boldsymbol{p}$	$\boldsymbol{p}_0$
识别模型参数	模型在经过训练后的$\boldsymbol{p}$	$\boldsymbol{p}_M$
训练轮次	模型训练的轮次, 范围[1, $\lvert S_{\text{train}} \rvert$]	t
模型参数变化	每轮学习过程中$\boldsymbol{p}$的变化, 所有的变化合成了$\boldsymbol{p}_0$到$\boldsymbol{p}_M$的变化	$\boldsymbol{\Delta}$
完美参数集	准确预测任何$\boldsymbol{v}$的$\boldsymbol{p}$, 是理论概念	$\boldsymbol{p}_P$
预测标签集	模型对V_{test}的预测l集合	L_{pred}

1.2 实现 FSMD 的一般流程

FSMD 的一般流程可分为如图 2 所示的训练和检测两个阶段. 训练包括 S_{train} 处理、$\boldsymbol{v}$ 嵌入、V_{train} 处理和模型训练. 检测则由 $\boldsymbol{v}$ 嵌入和 s 检测构成. 本节将对每个过程进行简述.

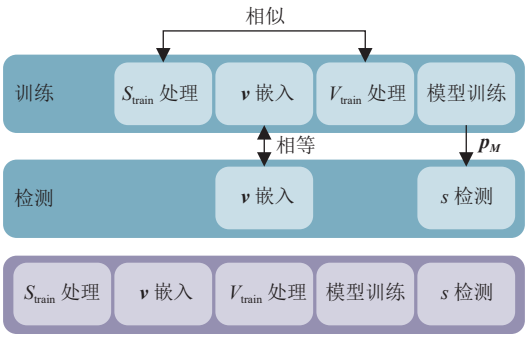

图 2 FSMD 一般流程

● S_{train} 处理. 全称为训练样本集处理, 在 FSMD 中 S_{train} 处理是一个基于小样本类 ($\lvert S \rvert$较小的类) 生成大量仿真数据的过程. 通过提升小样本类的$\lvert S \rvert$, 进而将模型训练过程从 FSL 转换为一般的监督学习过程, 提升模型训练的效果.

● $\boldsymbol{v}$ 嵌入. 全称为向量嵌入或特征工程, 用于将 s 转化为 $\boldsymbol{v}$, $\boldsymbol{v}$ 嵌入是 S_{train} 到 V_{train} 的一个映射. 由于 s 本身并不具备有效的比较结构, 无法直接设计识别规则和建模方法, 因此, $\boldsymbol{v}$ 嵌入采用人工或自动的方法, 间接赋予 s 之间的比较性质用于构建识别模型. 常见的人工特征包括: 文件结构特征、统计特征等.

● V_{train} 处理. 全称为训练向量集处理, 与 S_{train} 处理类似, V_{train} 处理是一个基于小样本类 $\boldsymbol{v}$ 生成大量仿真 $\boldsymbol{v}$ 的过程, 使用过采样方式[16]提升$\lvert V \rvert$.

● 模型训练. 该过程基于 V_{train}、$\boldsymbol{p}_0$ 和超参数, 不断更新模型参数 $\boldsymbol{p}$, 最终输出有效识别的模型参数 $\boldsymbol{p}_M$. 其中, V_{train} 和超参数在单独的一次训练过程中是固定的, 单独的一次训练通过 t 轮学习覆盖全部 V_{train}. 在模型参数的高维空间中, $\boldsymbol{p}_0$ 是 $\boldsymbol{p}$ 的起点, $\boldsymbol{\Delta}$ 是步长, $\boldsymbol{p}_M$ 是 $\boldsymbol{p}$ 的终点, $\boldsymbol{p}_P$ 是理论最优点, 实际情况中不可达.

● s 检测. 全称为样本检测, 以 V_{test} 和 $\boldsymbol{p}_M$ 为输入, 以 L_{pred} 为输出的模块. 在实际检测中, $\boldsymbol{p}_M$ 是源于相同或相似数据集的模型训练.

1.3 FSMD 相关研究概况

FSMD 的一般流程覆盖了 4 个不同的空间, 包括: 恶意软件的训练样本空间 (S_{train} 空间), 训练向量空间 (V_{train} 空间), 训练模型的参数空间 ($\boldsymbol{p}$ 空间) 和恶意软件的待测向量空间 (V_{test} 空间). 图 3(a) 展示了 FSMD 的一般流程与覆盖空间的关系, 结合 FSMD 背景, 可知, 数据不足和算法缺陷是干扰传统恶意软件检测模型构建过程的两种主要因素. 下面将对两种因素的产生、影响以及解决方案进行简要介绍, 并基于方法原理对现有的 FSMD 方法进行分类.

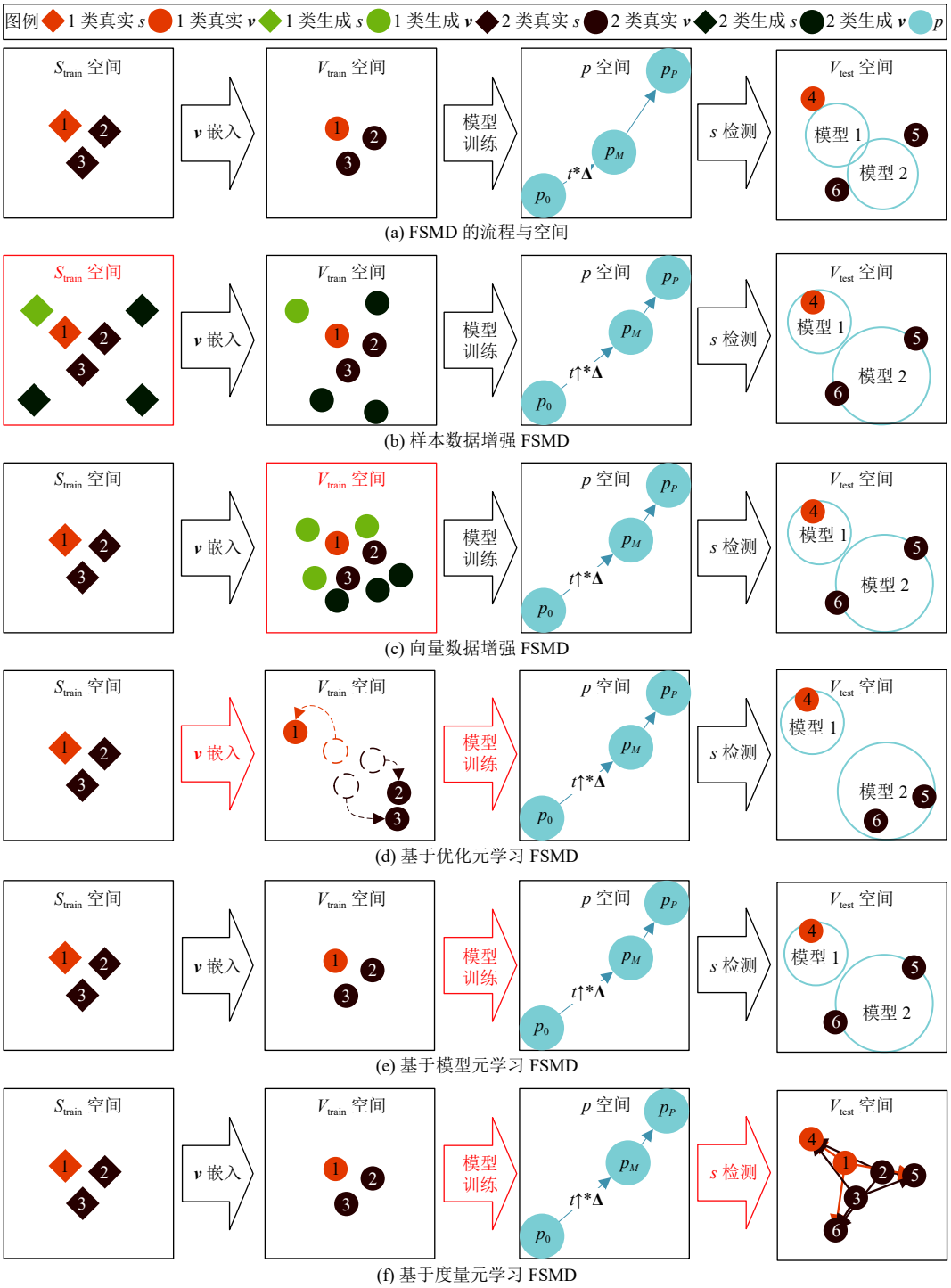

图 3 FSMD 方法比较

● 数据不足因素. 在 FSMD 中, $|S_{\text{train}}|$极小, $\boldsymbol{v}$ 嵌入的映射性质导致$|S_{\text{train}}| \geqslant |V_{\text{train}}|$, 因此, $|V_{\text{train}}|$极小; 学习轮次 t 与每轮训练数据量的乘积与$|V_{\text{train}}|$相等, 由此每轮训练数据量和 t 的范围都是 $[1, |V_{\text{train}}|]$, 每轮训练数据量极

小, 最终导致模型参数变化 $\boldsymbol{\Delta}$ 极小, 无法在较少训练轮次中生成用于有效识别小样本的模型参数 $\boldsymbol{p_M}$, 完整的关系见公式 (2). 通过数据增强方法增加| S_{train} |或| V_{train} |可以直接提升模型训练的数据总量, 大大提升学习轮次 t, 减小了 $\boldsymbol{p_M}$ 与 $\boldsymbol{p_P}$ 的差距. 如图 3(b) 和图 3(c) 所示, S_{train} 空间数据增强和 V_{train} 空间数据增强是从数据层面解决 FSMD 问题的方法.

$$\begin{cases} 1 \leqslant t \leqslant |V_{\text{train}}| \leqslant |S_{\text{train}}| \\ |\boldsymbol{p_M} - \boldsymbol{p_0}| = \boldsymbol{\Delta} \times t \end{cases} \tag{2}$$

● 算法缺陷因素. 即传统的监督学习方法无法从少量数据中构建有效的识别模型, 导致模型每轮训练的 $\boldsymbol{\Delta}$ 较小. 在 FSMD 训练过程中, 使用具有高效学习能力的元学习方法[17]可从多个方面增加 $\boldsymbol{\Delta}$, 例如优化特征和模型参数、设计针对性模型以及学习度量过程. 图 3(d)–图 3(f) 分别展示了 3 种元学习方法的作用范围.

经过上述分析, FSMD 方法可分为 3 类: 基于数据增强的 FSMD 方法、基于元学习的 FSMD 方法以及多技术结合的混合检测方法. 基于数据增强的 FSMD 包括面向样本空间的增强方法和面向特征空间的增强方法; 基于元学习的方法包括基于特征与模型参数优化的方法、基于模型设计的方法和基于相似度量的方法; 混合检测方法则结合多种数据增强或多种元学习思想. 图 4 展示了 FSMD 方法分类, 详细内容将在对应章节中进行介绍.

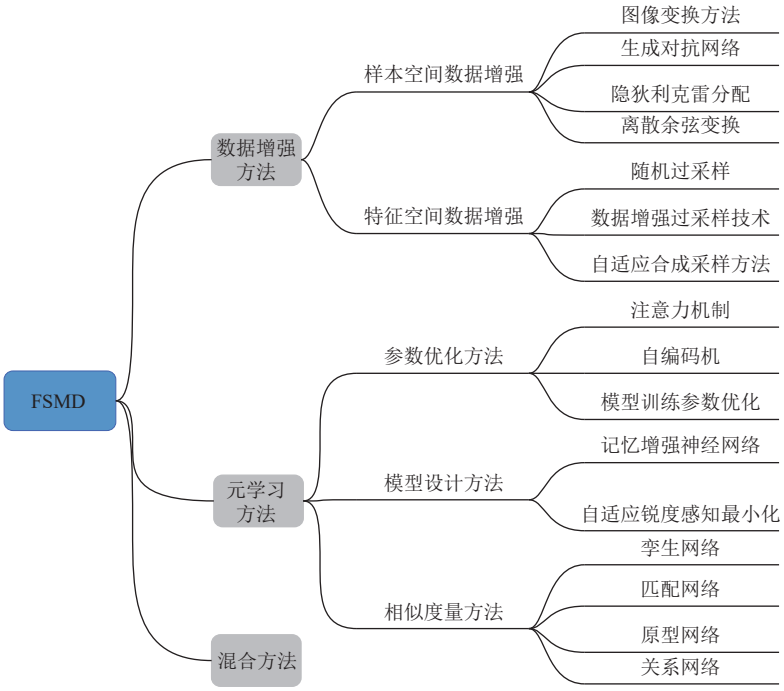

图 4 FSMD 研究分类概况

后文表 2 列举了 FSMD 相关研究的方法名称、方法分类、s 类型和实验数据集, 未命名的方法使用第一作者的姓作为方法名, 表中使用分类简称. 在网络空间安全领域, 一些面向异常流量等其他类型对象的检测方法也实现了有效的小样本学习过程, 因此表 2 也展示了这些可以借鉴到 FSMD 的方法.

2 基于数据增强的 FSMD 方法

数据不足因素导致在 FSMD 中存在公式 (2) 中数量关系: | S_{train} |≥| V_{train} |≥t, 使得模型训练过程收敛不充分, 无法产生有效识别模型的参数 $\boldsymbol{p_M}$. 数据增强方法的核心思想是通过分析学习小样本 s 和 $\boldsymbol{v}$ 生成仿真数据, 进而增加 t, 提高模型训练的收敛程度. 根据生成的仿真数据类型不同, 可将相关技术分为面向样本空间的数据增强方法和面向特征空间的数据增强方法, 下面将详细介绍各种方法的原理和相关工作.

表 2 FSMD 相关研究概况

名称	样本增强	参数优化	模型设计	相似度量	混合方法	*s*类型	数据集
FLAG[18]	√	—	—	—	—	异常流量	USTC-TFC2016[19], CIC-IDS2017[20]
FODA[21]	√	—	—	—	—	虹膜图像	LivDet-Iris 2017[22]
AEE-MSE[23]	—	√	—	—	—	恶意软件	非公开数据集
IMC[24]	—	√	—	—	—	恶意软件	APIMDS[25]
王方伟等人[26]	—	√	—	—	—	恶意软件	MalImg[27], MMCC2015[11]
MBL[28]	—	√	—	—	—	网页指纹	AWF dataset[29], DS-19_{100}[30]
FS-IDS[31]	—	√	—	—	—	异常流量	CICIDS2017[20]
META-WF[32]	—	√	—	—	—	Wi-Fi网络数据	AWID[33]
Li等人[34]	—	√	—	—	—	异常流量, IoT数据驱动信息	ISCX2012[35], CICIDIS2017[20], TON_IoT[36], 非公开数据集
MANNWARE[5,37]	—	—	√	—	—	恶意软件	Malicia-project[38], VirusTotal[39]
Tran's prototypical networks[40]	—	—	—	√	—	恶意软件	MalImg[27], MMCC2015[11]
SIMPLE[41]	—	—	—	√	—	恶意软件	VirusShare_00177[42], APIMDS[25]
Tran's matching networks[40]	—	—	—	√	—	恶意软件	MalImg[27], MMCC2015[11]
Hsiao等人[43]	—	—	—	√	—	恶意软件	VirusShare[44]
Zhu等人[45]	—	—	—	√	—	恶意软件	ImageNet[46], 非公开数据集
FSMC[47]	—	—	—	√	—	恶意安卓软件	CICInvesAndMal2019[48]
FC-Net[49]	—	—	—	√	—	异常流量	ISCX2012[35], CICIDIS2017[20]
CAD-FSL[16]	√	—	—	—	√	恶意软件	VirusTotal package[50]
DMMal[51]	—	—	√	√	√	恶意软件	BODMA[52], LargePEfiltered[53]
ConvProtoNet[54]	—	√	—	√	√	恶意软件	MMCC2015[11], LargePE[53], Drebin[55], VirusShare[44]
DPNSA[56]	—	√	—	√	√	恶意软件	LargePE[53]
Bai等人[6]	—	√	—	√	√	恶意安卓软件	Genome[57], Drebin[55], AMD[58]
UMVD-FSL[59]	—	√	—	√	√	异常流量	CICInvesAndMal2019[48], MCFP[60]
FCAD[61]	—	√	—	√	√	异常流量	CICAndMal2017[62], CIC-IDS2017[20]
Yuan等人[63]	—	√	—	√	√	内部威胁人员	CERT[64], Wikipedia[65]

2.1 面向样本空间的数据增强方法

面向样本空间的数据增强方法通过人工或自动的方法生成仿真 s, 提高 S_{train} 中小样本 s 的数量, 从而将 FSMD 问题转换为普通的监督学习问题.

2.1.1 图像变换方法

图像变换方法基于单一图像生成大量相似的图像. 深度学习[66]在图像的监督学习场景下具有良好的表现, 因此, 可将恶意软件转换为图像, 结合基于深度学习的图像识别方法实现高效的恶意软件检测. 对小样本恶意软件图像使用图像变换可以提升 s 数量, 例如, 平移[67]、翻转[68]、放缩[69]、旋转[70]等, 图 5 展示了恶意软件转换为图像的过程和图像 s 的变换过程.

2.1.2 生成对抗网络

生成对抗网络 (generative adversarial nets, GANs)[71]是一种基于博弈生成对抗型样本的网络. 在 GANs 中存在

两个互为对手的子网络构成: 用于生成数据的生成器网络 (generator network, G), 以及用于判断数据是真实数据还是生成数据的判别器 (discriminator network, D). G 以随机 $\boldsymbol{v}$ 为初始值, 经过非线性计算生成仿真的图像, D 是一个使用真实图像训练完成的监督学习分类器, 可以判别输入图像是否与训练图像是同类的数据. D 将识别的结果反馈给 G, G 根据 D 反馈的结果对进行参数调整, 并生成新的随机仿真图像, 重复上述过程, 直到 G 生成的仿真图像成功欺骗 D, G 网络训练完成, 使用 G 可以生成仿真的 s 图像, 图 6 展示了 GANs 的训练过程.

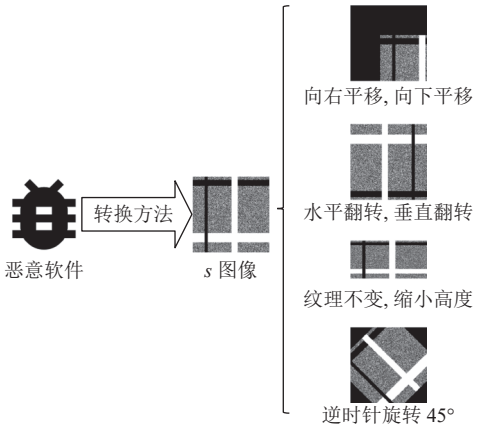

图 5 s 图像的变换处理

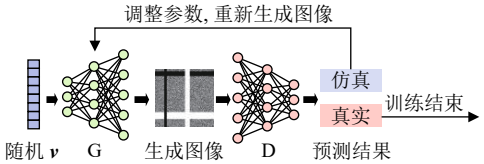

图 6 GANs 的训练过程

2.1.3 隐狄利克雷分配

隐狄利克雷分配 (latent Dirichlet allocation, LDA)[72]是一种用于处理离散数据集合的生成概率模型. LDA 分配在文本集合中提取主题, 本质上是一种聚类方法, 训练集中的 s 是语料库, 语料库的最小单位是词, 而一个语料库中具有一个或多个主题, 而不同的主题的词分布不同, 在不同的语料库中, 主题具有不同的分布. 小样本类型 s 转换为类语料库的对象后, 可以使用以往经验构建的主题集, 发现小样本类型中的主题子集, 并基于主题子集中的主题的词袋分布构建仿真 s. 图 7 展示了使用 LDA 通过小样本类型 s 生成仿真 s 的过程.

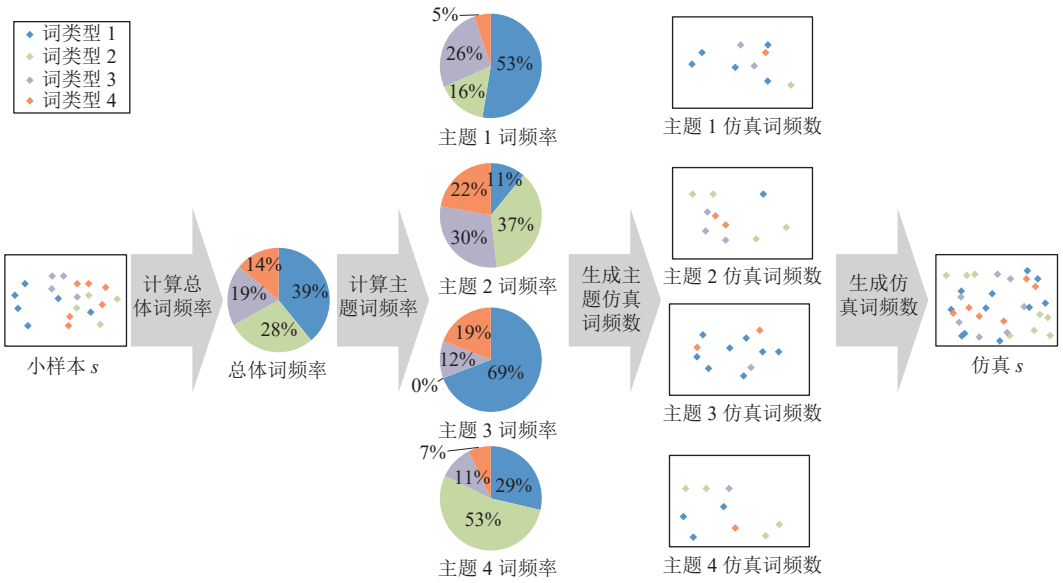

图 7 基于 LDA 生成仿真 s

LDA 的主题分布与词袋分布的独特思想对基于词袋结构的恶意软件检测方法具有启发意义, 例如基于字节词袋的小样本流量检测方法 FLAG[18], 以流量会话为主题构建 LDA 模型, 在仿真样本生成过程中, 仅从相似的主题中随机选择一个作为仿真小样本的分布, 基于主题分布概率随机生成字节分词, 生成完整的仿真 s, 重复上述过程可以生成任意数量的仿真 s. 在恶意软件检测领域, 如何确定词袋结构的主题, 将是有效利用 LDA 方法的关键.

2.1.4 离散余弦变换

离散余弦变换 (discrete cosine transform, DCT)[73]是一种将空域图像转换为频域频谱的算法, 逆离散余弦变换 (inverse discrete cosine transform, iDCT) 是 DCT 的逆运算, 可以将频域频谱转换为空域图像. 频谱中的高频区域代表空域图像中与周围像素差异过大的像素点, DCT 将高频信息集中在频谱的左上角. 高频对应图像的线性纹理, 低频对应图像的平滑纹理. 对于以图像为识别对象的方法, 将小样本类型 s 的频谱中的低频部分和其他 s 的频谱中的高频部分组合可以生成包含小样本平滑纹理的仿真频谱, 使用 iDCT 将仿真频谱转化为小样本仿真图像. 图 8 展示了使用 DCT 和 iDCT 生成仿真 s 的过程.

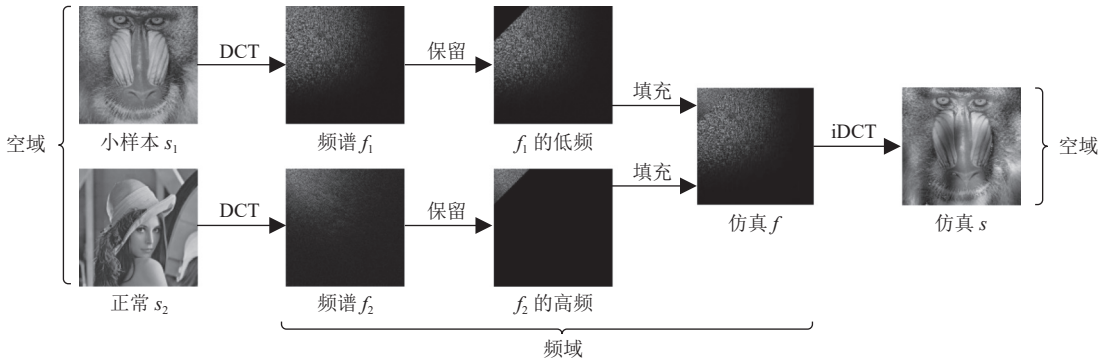

图 8 基于 DCT 和 iDCT 生成 s

基于 DCT 和 iDCT 仿真 s 生成过程可借鉴到基于图像的小样本恶意软件检测方法中, 参考虹膜呈现攻击检测 (presentation attack detection, PAD) 方法 FODA[21], 在频域组合普通样本的高频和小样本的低频, 并转换为保留了小样本平滑纹理的仿真图像, 用于训练小样本伪造虹膜图像识别模型. 该方法的有效性基于瞳孔图像的平滑纹理能保留更多伪造虹膜信息, 对于基于图像的恶意软件检测, 提升对恶意软件图像的解释性, 明确恶意软件图像中多种纹理所蕴含的信息是将 DCT 成功应用于 FSMD 所面临的重要科学问题.

2.2 面向特征空间的数据增强方法

在模型的训练和检测过程中, 并不能直接判别原始的 s 类型, 而是将 s 转化为符合模型要求的 $\boldsymbol{v}$, 再进行进一步操作. 在 FSMD 中, $|V_{\text{train}}|$的提升也可以提高 t, 有效提升模型训练收敛程度, 降低小样本识别模型训练难度. 面向特征空间的数据增强方法是生成仿真 $\boldsymbol{v}$ 的方法, 在特征空间等价于解决 V_{train} 不平衡问题[74]的 $\boldsymbol{v}$ 过采样方法. 如基于数据增强的恶意软件检测方法[75]等.

2.2.1 随机过采样

随机过采样[76]基于单个 $\boldsymbol{v}$ 生成大量仿真 $\boldsymbol{v}$. 随机过采样方法对 V_{train} 中的小样本 $\boldsymbol{v}$ 进行随机复制, 不改变 V_{train} 空间中 $\boldsymbol{v}$ 的分布位置, 可使用类型权重模拟过采样. 在二分类训练问题中, 随机过采样方法提升模型对小样本类型的学习能力, 但易产生过拟合现象[77]. 图 9(a) 展示了 V_{train} 中的 $\boldsymbol{v}$ 分布情况, 图 9(b) 展示了图 9(a) 经过随机过采样后的 $\boldsymbol{v}$ 分布情况.

2.2.2 数据增强过采样技术

数据增强过采样技术 (synthetic minority oversampling technique, SMOTE)[78]是一种基于多个 $\boldsymbol{v}$ 生成仿真 $\boldsymbol{v}$ 的方法. 首先在 V_{train} 中选择一个小样本 $\boldsymbol{v}_0$, 选择与 $\boldsymbol{v}_0$ 之间度量距离最近的 k 个小样本 $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k$, 分别在 $\boldsymbol{v}_0$ 和每个临近样本的直连线上随机取一个或多个作为仿真, 使用欧氏距离[79]或余弦相似[80]度量. 相比于随机过采样方法,

SMOTE 生成的仿真 $\boldsymbol{v}$ 更丰富. SMOTE 中 k 值的设置会影响生成样本的效果, k 过大, SMOTE 会忽略小样本中离群点, k 过小导致仿真 $\boldsymbol{v}$ 过于接近原始样本, 当 k 为 0, SMOTE 等价于随机过采样方法. 由 SMOTE 的过程可知, 仿真 $\boldsymbol{v}$ 都是沿着直连线生成, 对模型训练过程的要求较高. 图 9(c) 展示了图 9(a) 采用 SMOTE 生成仿真 $\boldsymbol{v}$ 的结果. SMOTE 具有多种衍生方法, 包括 Borderline-SMOTE[81]、K-means SMOTE[82]、SVM SMOTE[83]和 SMOTE-NC[84]等.

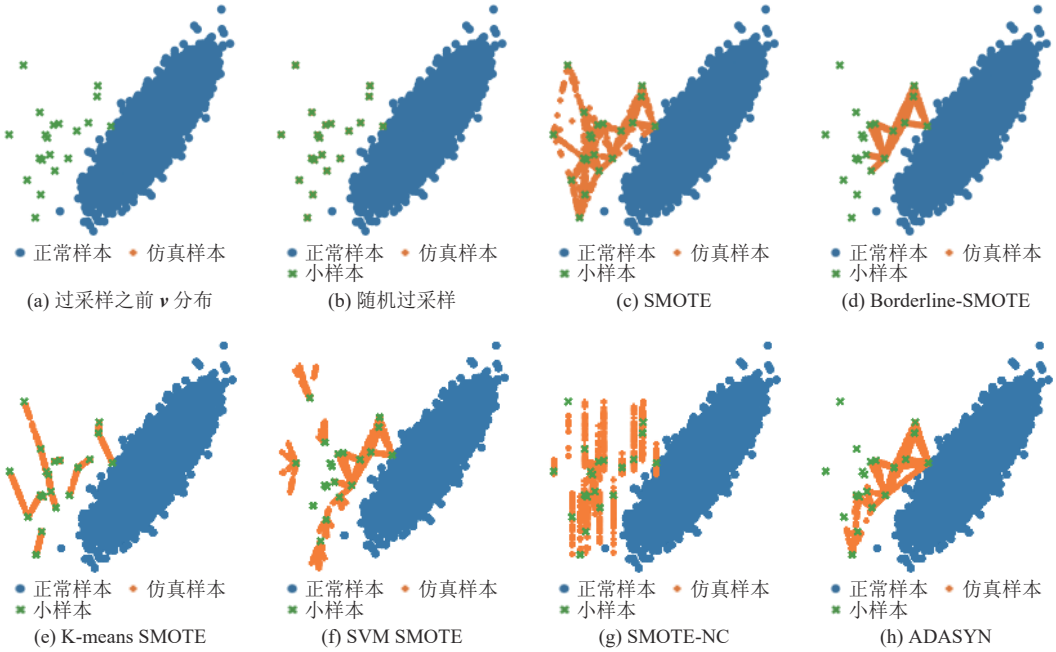

图 9 基于过采样生成 $\boldsymbol{v}$

Borderline-SMOTE 关注空间中小样本类型与其他类型的边界区域, 度量小样本 $\boldsymbol{v}$ 周围范围内的 $\boldsymbol{v}$ 类型数量占比, 筛选出正常类与小样本类占比接近 $\boldsymbol{v}$ 的作为边界 $\boldsymbol{v}$, 仅从处于样本边界的 $\boldsymbol{v}$ 中选择 $\boldsymbol{v}_0$, 生成仿真 $\boldsymbol{v}$. 图 9(d) 展示了图 9(a) 使用 Borderline-SMOTE 生成的仿真 $\boldsymbol{v}$ 的分布情况.

K-means SMOTE 关注集中的小样本 $\boldsymbol{v}$, 使用 K-means 算法[85]将所有 $\boldsymbol{v}$ 分为多个簇, 分别计算簇中的小样本占比, 从小样本占比高的簇中选择 $\boldsymbol{v}_0$ 进行过采样. 图 9(e) 是图 9(a) 使用 K-means SMOTE 产生的仿真 $\boldsymbol{v}$ 分布. SVM SMOTE 是 Borderline-SMOTE 的优化方法.

SVM SMOTE 通过支持向量机 (support vector machine, SVM)[86–88]算法计算支持 $\boldsymbol{v}$, 在支持 $\boldsymbol{v}$ 中选择小样本 $\boldsymbol{v}_0$, 最后使用 SMOTE 生成仿真 $\boldsymbol{v}$. 图 9(f) 展示了图 9(a) 使用 SVM SMOTE 生成的仿真 $\boldsymbol{v}$ 的分布.

SMOTE-NC 可产生包含枚举类型维度和连续类型维度的 $\boldsymbol{v}$, 而 SMOTE 自身并不能完成这一任务, 需提前对 $\boldsymbol{v}$ 的枚型维度使用独热 (one-hot) 编码[89]扩展成高维连续向量, 再使用 SMOTE 算法, 最后对生成的 $\boldsymbol{v}$ 进行降维, 实现对包含枚举型维度的 $\boldsymbol{v}$ 的过采样. 假设图 9(a) 的横坐标方向维度是枚举型, 图 9(g) 展示了使用 SMOTE-NC 生成的仿真 $\boldsymbol{v}$ 的分布.

2.2.3 自适应合成采样方法

自适应合成采样方法 (adaptive synthetic sampling approach, ADASYN)[90]是一种基于小样本 $\boldsymbol{v}$ 附近 $\boldsymbol{v}$ 类型分布计算生成仿真 $\boldsymbol{v}$ 数量的方法. ADASYN 首先判断输入类型是否为 V_{train} 中的小样本类型, 若是则计算生成仿真 $\boldsymbol{v}$ 的总数, 以及在每个小样本 v 附近需要生成仿真 $\boldsymbol{v}$ 的数量, 使用 SMOTE 生成仿真 $\boldsymbol{v}$. 反之则算法结束. 图 9(h) 展示了图 9(a) 使用 ADASYN 生成的仿真 $\boldsymbol{v}$ 的分布. ADASYN 的详细流程如图 10 所示, 其中小样本形状的大小表示 ADASYN 计算得到的其生成样本数量的多少.

V_{train} 处理方法在解决与 FSMD 类似的 V_{train} 不平衡问题中被广泛采用[91,92], 对 FSMD 具有借鉴意义.

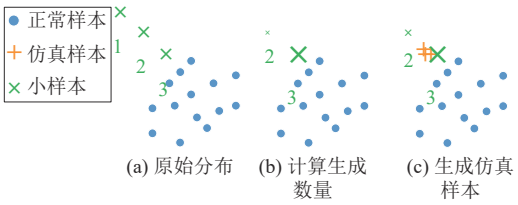

图 10 基于过采样生成 $\boldsymbol{v}$

3 基于元学习的 FSMD 方法

针对算法缺陷因素, 使用元学习方法提升算法学习效率是解决 FSMD 问题另一个思路. 相较于普通的监督学习, 元学习是一种学习如何学习的过程, 例如人类分辨猫或狗的类型, 仅需要几个特例图像.

元学习可分为训练和测试两个过程, 在训练过程中, 包括多组学习任务, 为了与元学习的训练过程和测试过程进行区分, 每个学习任务中的训练集称为支持集 (support set, S_{support}), 每个学习中的测试集称为查询集 (query set, S_{query}). 在测试过程中, 仅存在一个学习任务, 用于训练真正的小样本检测模型. 在元学习的训练过程中, 存在一个元学习器, 用于从多组学习任务中完成元学习任务, 图 11 展示了元学习的一般流程.

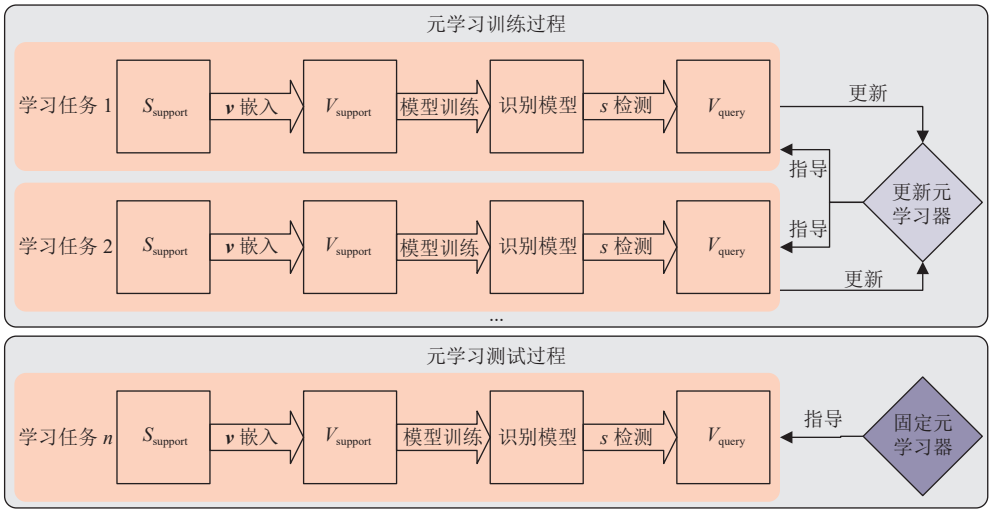

图 11 元学习的一般流程

基于元学习方法的 FSMD 包括: 基于异常特征与模型参数优化的方法、基于学习模型设计的方法以及基于威胁样本相似度量的方法. 优化方法针对整个学习过程的参数进行调整, 例如特征权重、特征选择、模型参数、训练参数等. 模型方法则设计了用于小样本识别的学习模型. 度量方法则是通过元学习方法构建一个有效的度量规则, 以适应当前的恶意软件对象.

3.1 基于特征与模型参数优化的方法

3.1.1 注意力机制

注意力机制[93]是对异常样本的特征进行赋权的方法. 若不采用注意力机制, 则输入到检测模型之前每一个维度的信息是等价的, 即所有维度的权重值为 1. $\boldsymbol{v}$ 权重调整从先验数据中学习 $\boldsymbol{v}$ 权重序列, 并基于权重序列对每一个维度的信息进行赋权. 而注意力机制不改变原始的 $\boldsymbol{v}$ 空间, 仅在相同的 $\boldsymbol{v}$ 空间中更改多个 $\boldsymbol{v}$ 的分布位置, 最终实现有效的 $\boldsymbol{v}$ 嵌入. 图 12(a) 展示了特征权重序列的生成过程, 图 12(b) 展示了注意力机制在 $\boldsymbol{v}$ 嵌入中的部署情况.

IMC[24]使用应用编程接口 (application programming interface, API) 序列[94,95]作为 s, 实现对恶意软件的检测.

在 $\boldsymbol{v}$ 嵌入阶段, IMC 使用词嵌入方法将 API 序列转化为定长的 $\boldsymbol{v}$. 在模型训练阶段, IMC 使用注意力机制保留每类的 $\boldsymbol{v}$ 权重序列. 在 s 检测阶段, IMC 对待测的 $\boldsymbol{v}$ 进行赋权, 再与对应类别的 $\boldsymbol{v}$ 进行比较, 通过余弦相似判断待测 $\boldsymbol{v}$ 与每个类别的相似性, 实现恶意软件识别.

MBL[28]是一种新型的网页指纹攻击[96-98]方法, 通过网络流量行为判断用户当前正在访问的网页. MBL 攻击并不是 FSMD 方法, 但在 FSL 背景下的 MBL 攻击过程可对 FSMD 带来启示. 首先, MBL 使用大量的先验数据构建了一个基于卷积神经网络 (CNN)[99–101]的网络流量 $\boldsymbol{v}$ 嵌入, 对于小样本, MBL 使用了深度网络特征初始参数的学习, 并在训练过程中通过限制 CNN 中的部分参数, 实现 $\boldsymbol{v}$ 嵌入的微调, 提高了训练效率, 加快了模型训练速度.

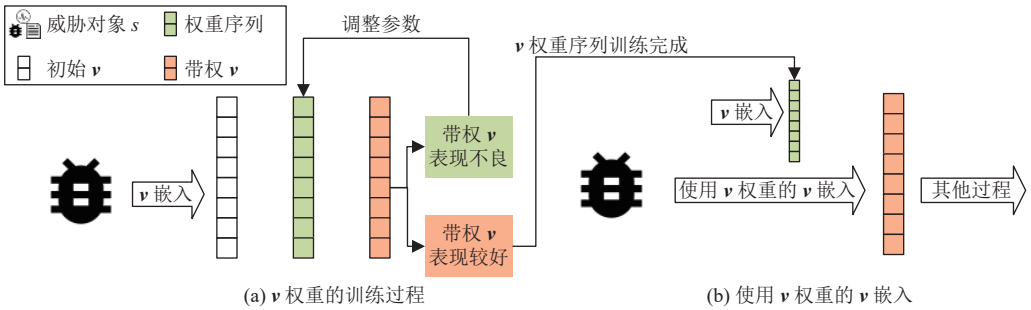

图 12 基于注意力权重的 v 嵌入

3.1.2 自编码机

自编码机 (auto encoder, AE)[102]是一种表示学习[103]算法. AE 包括用于提升表示效率的收缩型 AE (contractive AE, CAE)、用于标注标签的正则型 AE (regularized AE, RAE) 以及用于生成向量的变分型 AE (variational AE, VAE). 自编码机将输入的高维 $\boldsymbol{v}$ 转化为低维 $\boldsymbol{v}$, 并保证输出的 $\boldsymbol{v}$ 可以保留输入 $\boldsymbol{v}$ 的信息, 降低无效维度对模型训练的影响. 自编码机包括两个神经网络: 编码网络 (encoder network, EN) 和解码网络 (decoder network, DN). EN 将输入的高维 $\boldsymbol{v}$ 转换为隐藏层的低维 $\boldsymbol{v}$, DN 将隐藏层的低维 $\boldsymbol{v}$ 再次转换为输出的高维度 $\boldsymbol{v}$. 自编码机以输出和输入的相似性为调优指标, 不断训练 EN 和 DN, 当输出和输入的相似性达到预先设定的阈值, 则完成训练过程. 在 $\boldsymbol{v}$ 嵌入过程中, 将生成的高维 $\boldsymbol{v}$ 直接输入到训练完成的自编码机中, 并输出隐藏层的低维 $\boldsymbol{v}$, 将低维 $\boldsymbol{v}$ 作为模型训练和 s 检测的输入数据, 提高模型训练的收敛速度, 降低 $\boldsymbol{v}$ 嵌入的时间成本[104]. 图 13(a) 展示了自编码机模型的训练过程, 图 13(b) 展示了在 $\boldsymbol{v}$ 嵌入中的部署情况.

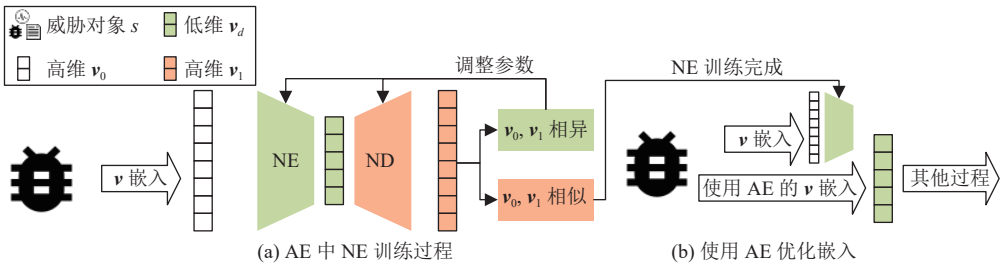

图 13 基于 AE 的 v 嵌入

AEE-MSE[23]是检测 Linux 平台下小样本恶意软件的方法, 以恶意软件执行的恶意 API 调用为输入, 构建恶意 API 图像, 使用自编码机对图像经过深度卷积生成的高维 $\boldsymbol{v}$ 进行简化, 以隐藏层 $\boldsymbol{v}$ 作为模型训练和 s 检测的输入. 类似的方法还包括用于小样本流量检测的 FS-IDS[31].

3.1.3 模型训练参数优化

模型训练参数优化是基于元学习范式进行训练过程优化方法. 相较于传统的监督学习方法, 模型参数初始值是随机的, 通过不断学习, 模型参数会经过多次变化, 直到模型收敛, 在深度学习模型中, 模型参数包括特征提取模型参数和识别模型参数, 模型训练参数优化可提升模型参数的变化效率, 极大提升训练速度. 模型不可知元学习

(model-agnostic meta-learning, MAML)[105]是该类方法的代表之一, MAML 通过大量先验数据进行学习, 并在元学习训练过程中优化二阶梯度学习模型初始参数, 在元学习测试过程中, 元学习器设置的模型参数初始值仅需要少量的数据训练便能收敛, 实现小样本识别.

王方伟等人[26]提出一种基于元学习过程和困难样本挖掘的小样本恶意软件识别方法. 该元学习训练过程是根据模型收敛程度终止. 若模型收敛, 则结束元学习过程, 输出分类模型; 反之, 则从分类效果最差的小样本类中随机抽取 $\boldsymbol{v}$, 加入下一轮学习任务 S_{suppert} 中, 继续元训练过程, 并在元学习器完成后, 通过 FSL 对模型进行微调. 用于检测 Wi-Fi 流量的 META-WF[32]和文献 [34] 中提及的基于递归特征金字塔 (recursive feature pyramid, RFP)[106]的流量检测方法也都应用了模型训练参数优化.

3.2 基于学习模型设计的方法

3.2.1 记忆增强神经网络

记忆增强神经网络 (memory-augmented neural network, MANN)[70]是一种基于外部记忆的深度网络. 在进行模型训练过程中, 外部记忆方法除了学习模型的超参数外, 还要将训练过程中的 $\boldsymbol{v}$ 存储到外部的存储单元 (记忆), 在分类过程中提供辅助. 结合记忆中的 $\boldsymbol{v}$, 提升小样本类的识别准确率. 如图 14(a) 所示, 准确的读写机制中, 准确索引是一个固定值 k, 仅对记忆索引为 k 的数据进行读写操作. 为使读写操作可进行学习, 外部记忆方法采用模糊读写机制, 神经图灵机 (neural Turing machine, NTM)[107], 如图 14(b) 所示, 在模糊读写机制中, 模糊索引是一个索引概率序列, 记忆索引 i 的概率非负, 并使用 K_i 表示, 所有索引概率和为 1, 根据索引概率对记忆中所有数据进行读写操作.

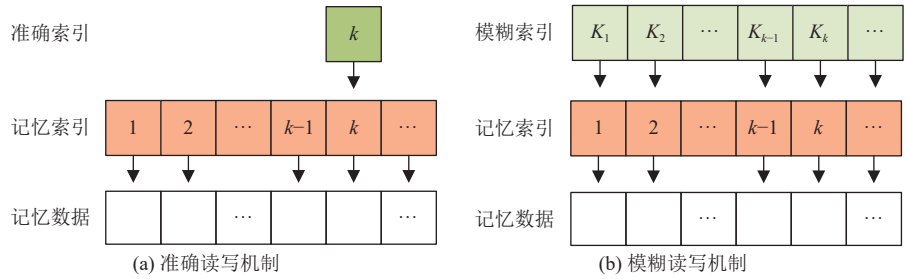

图 14 两种读写机制的对比

NTM 的结构如图 15 所示, 包括控制器 (controller, C), 读头 (read head, RH), 写头 (write head, WH) 和外部存储器 (memory, M). 在训练过程中, C 学习模糊索引生成规则和分类规则; RH 学习 M 的模糊读取规则, WH 学习 M 的模糊写入规则. 在一次读记忆过程中, C 生成 $\boldsymbol{v}$ 的模糊索引并发送给 RH, RH 根据模糊读取规则从 M 中读取外部记忆 $\boldsymbol{v}_{\text{read}}$ 并发送给 C, C 识别 $\boldsymbol{v}_{\text{read}}$ 的类型, 作为 $\boldsymbol{v}$ 的预测类型. 在一次写记忆的过程中, C 生成 $\boldsymbol{v}$ 的模糊索引并发送给 WH, WH 根据模糊写入规则将 M 更新为 M_{write}.

Tran 等人首次使用了 MANN 实现了针对恶意软件图像的 FSMD[5], 并进一步设计了 MANNWARE[37], 应用于勒索软件检测[37]. MANNWARE, 在元学习训练阶段对 MANN 中的参数进行学习, 并最终应用到元学习测试阶段.

3.2.2 自适应锐度感知最小化

FSMD 算法需具备更易达成的收敛条件, 方能在小样本训练集上快速训练出具有良好泛化能力的识别模型. 相关研究, 已证明损失平面的形状与模型的泛化能力具有密切关系, 损失平面锐度较小时, 训练的模型具有良好的泛化能力, 自适应锐度感知最小化 (adaptive sharpness-aware minimization, ASAM)[108]是当前最为先进的方法之一, 具有更加高效的损失平面最小锐度感知范围, 结合深度学习网络结构和元学习训练范式, 可有效提升 FSMD 识别模型的泛化能力.

3.3 基于相似度量的方法

3.3.1 孪生网络

孪生网络 (siamese networks, SNs)[109]由度量两个 s 之间相似性的 NN 对构成, 每个 NN 的参数完全相同. 为保

证相似性度量准确性, SNs 中 NN 的参数通过 s 数量充足且类型丰富的数据集训练得到. 为保证输入有效性, 训练 SNs 的 s 与小样本类型 s 一致. 在 s 识别过程中, SNs 的 NN 对同时输入一个 S_{train} 中的 s_1 和一个待测 s_2, 之后, SNs 会输出二者的嵌入 $\boldsymbol{v}_1$ 和 $\boldsymbol{v}_2$ 或类别标签 l_1 和 l_2. 使用高维空间的相似性度量方法比较 $\boldsymbol{v}_1$ 和 $\boldsymbol{v}_2$ (或 l_1 和 l_2) 的相似性, 当二者相似性高于设定的阈值时, 则认为 s_1 和 s_2 的类型相同. 图 16 展示了 SNs 模型训练和 s 识别过程.

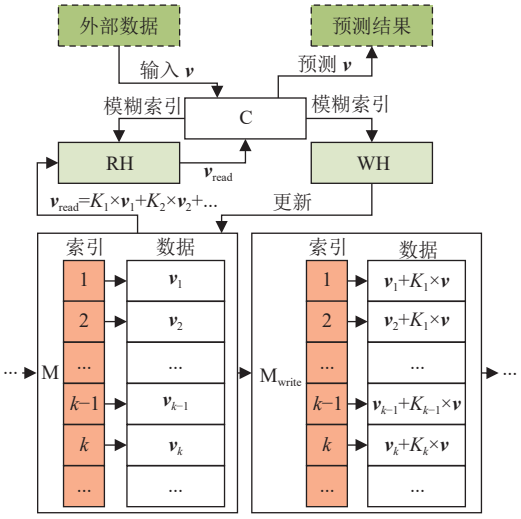

图 15 NTM 的构成

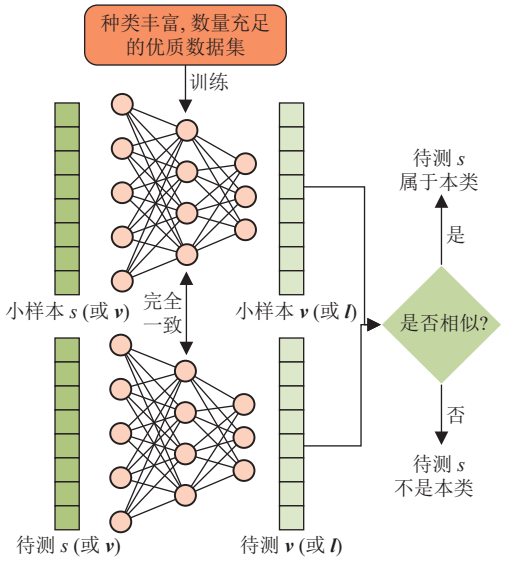

图 16 SNs 的一般流程

Hsiao 等人[43]使用 SNs, 并以 CNN 对构成类 SNs, 采用与小样本类恶意软件图像相似的 s 对的 CNN 进行训练. 利用曼哈顿距离[110]度量待测 s 与标签已知的 s 相似性相, 从而实现 1-shot 检测. 由于实验使用的是从 VirusShare 网站[44]收集的无标签数据集, 其恶意软件类型标签由图像的平均哈希 (average Hash, aHash)[111]计算得出.

Zhu 等人[45]使用 SNs 对勒索软件的熵值图进行识别. 其 SNs 由两个参数一致的 VGG16[112–114]构成, VGG16 的参数由公共数据集 ImageNet[46]训练生成, 对经过 VGG16 生成的两个 $\boldsymbol{v}$, 采用中心损失 (center loss)[115]分类方法, 判断 s 的类型.

3.3.2 匹配网络

匹配网络 (matching networks, MNs)[116]是一种将待测 s 和所有 S_{train} 中的 s 进行对比的方法. 每次对待测 s 进行比较之前, 生成待测 s 的权重注意力 $\boldsymbol{v}$, 并基于注意力 $\boldsymbol{v}$ 和 S_{train} 中的 s 生成临时集合 $Temp_{\text{train}}$, 对待测 s 与 $Temp_{\text{train}}$ 中所有 s 与进行相似性度量, MNs 根据度量结果综合判定待测 s 的类型. 图 17 展示了 MNs 的 s 识别过程.

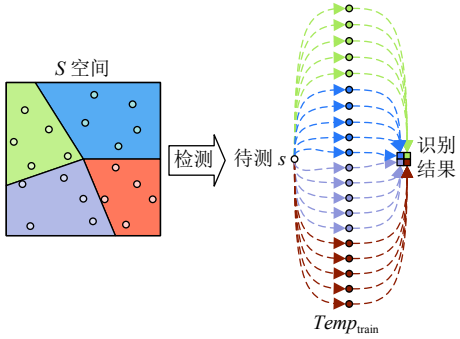

图 17 MNs 的一般流程

Tran 等人[40]使用 MNs 识别恶意软件图像, 并采用长短期记忆 (long shot-term memory, LSTM)[117–119]作为 MNs 的 $\boldsymbol{v}$ 嵌入方法, 通过待测 s 生成具有注意力机制的 LSTM, 用于小样本类 s 的识别. 最后, 在 MalImg[27]和 MMCC2015[11]两个数据集上进行了验证实验.

FSMC[47]使用 MNs 识别恶意安卓软件 API 调用序列. 通过 CNN 将 API 调用序列嵌入为 $\boldsymbol{v}$, 采用 K-近邻 (K-nearest neighbor, KNN)[120–122]衡量待测 $\boldsymbol{v}$ 周围 $\boldsymbol{v}$ 的类型, 综合判断待测 $\boldsymbol{v}$ 的类型.

3.3.3 原型网络

原型网络 (prototypical networks, PNs)[123]是一种更加高效的匹配网络. 不同于传统监督学习模型不保存 V_{train} 的训练过程, PNs 对 V_{train} 进行选择性保留, 保留的 $\boldsymbol{v}$ 称为原型. 通常 PNs 并不会直接保留 V_{train} 中的原始 $\boldsymbol{v}$, 而是保留统计性的 $\boldsymbol{v}$ 作为原型 $\boldsymbol{v}$, 如该类中所有 $\boldsymbol{v}$ 的平均值等. 在检测过程中, PNs 将待测 $\boldsymbol{v}$ 与每个原型逐一匹配, 综合判断待测 $\boldsymbol{v}$ 的类型, 图 18 展示了 PNs 的模型训练和检测过程, 其中小圆形表示 V_{train} 中的原始 $\boldsymbol{v}$, 大圆形表示该类的原型 $\boldsymbol{v}$. PNs 的模型训练过程具有实现简单, 训练快速等优点, 在 FSL 中被广泛使用.

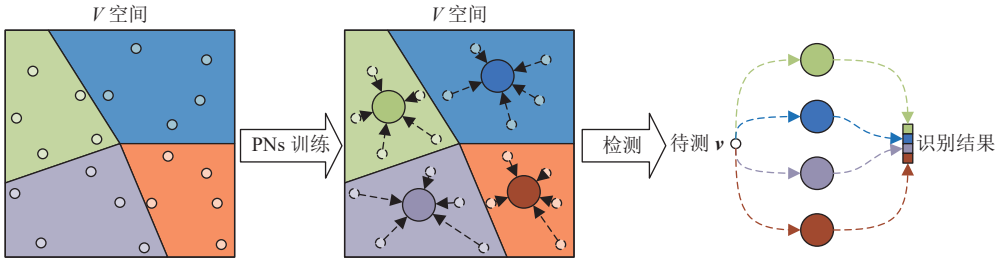

图 18 PNs 的一般流程

Tran 等人[40]使用 PNs 实现了一个针对恶意软件的 FSMD 方法. 该方法首先将恶意软件转换为图像, $\boldsymbol{v}$ 嵌入方法为 CNN, 通过对小样本类的学习, 构建出每一个小样本类的原型, 并使用欧氏距离[79]度量待测 $\boldsymbol{v}$ 与每个原型的相似性, 判断其类型.

SIMPLE[41]是另一种基于 PNs 的恶意软件 API 识别方法. 通过词嵌入将恶意软件的 API 转换为 $\boldsymbol{v}$, 经过元学习训练, 使元学习器不断学习 PNs 中的原型生成规律, 在元学习测试中指导 PNs 的构建过程, 提高 PNs 中生成原型的质量.

3.3.4 关系网络

关系网络[124]是一种可用于上述 3 种方法中的比较方法, 关系网络包含嵌入方法和比较方法, 与上述方法不同的是, 关系网络的表示方法是可以学习, 而非固定的, 针对不同数据采用不同的比较方法, 相较于无法学习的度量方法, 关系网络的应用场景更加广泛.

FC-Net[49]是一种基于关系网络的流量检测框架. 由两个 FC 网络构成, 其输入是两个 D 相同的 s, 输出是二者的相似性. FC 网络由特征提取网络 (F 网络) 和比较网络 (C 网络) 构成, F 网络将两个 s 嵌入为 $\boldsymbol{v}$, C 网络将两个 $\boldsymbol{v}$ 拼接, 并综合分析二者是否为同一类别. FC-Net 基于元学习训练过程生成 FC 网络的参数. FC-Net 方法的设计原则和关系网络的使用过程都可以迁移到基于度量的 FSMD 方法中.

4 多技术结合的混合检测方法

在 FSMD 研究中, 一些方法为了实现对小样本对象的有效识别采用了多种数据增强方法或多种元学习方法. 如图 19 所示, 相比于仅使用一种技术的 FSMD 方法, 多技术结合的混合检测方法至少采用两种技术, 构成了新型的检测方法.

CAD-FSL[16]结合了图像变换方法和 GANs 生成仿真小样本恶意软件图像. 通过对恶意软件图像与良性软件图像异或生成仿真的隐式图像, 并改变恶意软件图像部分区域像素点的顺序生成仿真的混淆图像. CAD-FSL 使用

恶意软件图像、隐式图像和混淆图像训练 GANs, 生成完备的仿真图像集, 使用监督学习方法实现对小样本恶意软件的检测.

ConvProtoNet[54]结合了特征注意力机制和基于度量的元学习方法. 首先, ConvProtoNet 将恶意软件转换为尺寸一致的灰度图像, 并使用先验知识集进行元学习过程, 以减小训练过程中小样本学习任务的损失为目标, 不断调整卷积层产生的特征, 最终构建一个有效的 $\boldsymbol{v}$ 嵌入方法, 并使用 PNs 作为度量样本相似性模型, 提高模型识别精度.

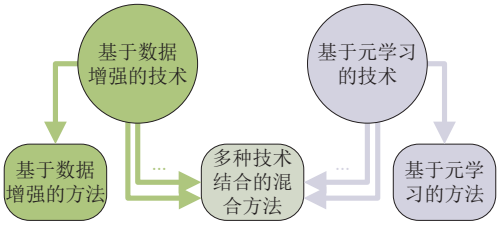

图 19 多技术结合的 FSMD 方法构成

DPNSA[56]将恶意软件转换为相同尺寸的灰度图, 使用元学习训练范式调整 $\boldsymbol{v}$ 权重, 并结合基于度量的元学习. DPNSA 的 $\boldsymbol{v}$ 嵌入过程与普通的学习任务一致, 但在 $\boldsymbol{v}$ 输入到 s 识别之前, DPNSA 使用双样本动态激活模块 (dual-sample dynamic activation module, DSDA) 对参与比较的两个 $\boldsymbol{v}$ 进行了权重调整, 以确保减小同类的 $\boldsymbol{v}$ 之间的差距, 并增大异类的 $\boldsymbol{v}$ 之间的差距. 使用 PNs 进行多个样本间的比较, 通过元学习训练过程, 不断完善 DSDA 的性能, 最终实现对提高小样本 s 识别准确率的效果.

DMMal[51]是一个结合元学习过程和 PNs 的恶意软件图像识别方法. 使用灰度图, 变分模态分解 (variational mode decomposition, VMD) 图[125]和熵值图[126]输入到 3 个通道, 构成三通道图像, 继而利用 PNs 构建小样本类的原型, 在元学习训练过程中, 元学习器对使用 ASAM 优化器的模型训练方法进行学习, 并将学习参数应用到元学习测试任务中, 构建出具有更强泛化能力的识别模型.

Bai 等人[6]使用安卓软件的权限, API 调用和组件间通信 (inter-component communication, ICC)[127]作为识别的 s, 分析待测软件是否是为恶意软件. 该多层感知器 (multi-layer perceptron, MLP)[128]将经过 one-hot 编码的安卓软件的权限, API 调用和 ICC 转换为 $\boldsymbol{v}$, 并通过先验数据训练识别模型, 结合权重注意力机制, 使同类的 $\boldsymbol{v}$ 更相似, 不同类的 $\boldsymbol{v}$ 更相异. 在 s 识别过程中使用了 SNs 度量待测 $\boldsymbol{v}$ 和已知类型 $\boldsymbol{v}$ 的相似性确定其具体的软件类别.

在对其他威胁的检测研究中, 依然存在有效的小样本检测方法, 具体包括: 基于注意力机制和 PNs 的小样本异常流量检测方法 UMVD-FSL[59], 基于 MAML 和 PNs 的小样本异常流量检测方法 FCAD[61]以及 Yuan 等人[63]提出的基于特征注意力和 PNs 的威胁人检测方法等, 上述方法对 FSMD 都具有借鉴意义.

5 FSMD 展望

通过 FSMD 研究现状的总体分析, 拟从问题背景, 关键技术和应用场景 3 个方面对 FSMD 提出展望.

5.1 问题背景

在以往研究中, few-shot 出现过多种含义. few-feature 表示| S_{train} |较大, 嵌入 $\boldsymbol{v}$ 的维度 D 较小; few-label 表示| S_{train} |较大, 明确标签的数量较少; few-sample 表示| S_{train} |较小. 显然, 在 few-sample 背景下, 模型训练的难度是最大的, 本文提及的 FSMD 属于 few-sample. 目前的 FSMD 方法使用相似领域的先验知识来减弱小样本对模型训练的影响, 大多数方法在 K=20 时体现出最好的总体性能. 基于此, 应进一步加强对相似领域先验知识的使用限制, 包括使用泛化性的先验数据, 以及使用更少的先验数据. 同时加强对 K 的限制, 将其限制在 10 以下. 还应加强对每类小样本恶意软件的识别性能优化的关注. 另外, 在实际的恶意软件发展过程中, 存在激增的新型恶意软件和缓慢增长的恶意软件, FSMD 应该明确针对的是缓慢增长的类型, 因为激增类型的恶意软件由于数量庞大, 可以直接

采用一般的监督学习等方法进行合理建模.

5.2 关键技术

每种 FSMD 方法都使用了独特的技术去解决小样本恶意软件识别问题, 但现有的技术中仍然存在亟待解决的科学问题.

仿真数据 (仿真 s 和仿真 $\boldsymbol{v}$) 的有效性验证问题. 数据增强方法通过生仿真数据提升模型训练的输入量, 但对于概率模型, 只有当仿真数据分布与真实数据分布相似时, 才能保证训练模型的识别能力, 如何在 FSMD 中验证仿真数的有效性将是提升数据处理方法效果的关键技术.

对抗性数据的影响问题. 由于 FSMD 的训练数据较少, 如果攻击者使用了对抗数据, 会对识别模型造成严重的影响, 从而导致 FSMD 失效, 发展对抗数据过滤技术或研究削减对抗数据干扰的技术, 成为了提高 FSMD 模型鲁棒性的重要手段.

增量数据的持续学习问题. 在 FSMD 模型训练的过程中, 如果提高训练数据量, 训练模型将提升对新数据的学习效果, 而削弱对旧数据的学习效果, 导致小样本情况下模型的良好表现并不会在大量训练数据中体现出来, 在数据增量的情况下, 保持 FSMD 模型具有持续学习能力的技术是亟待进行深入研究的技术.

识别模型的泛化性问题. FSMD 由于训练数据量少, 为了提高识别模型的效果, 大部分方法采用先验知识提升模型的收敛速度, 在基于先验知识的模型上进行微调, 少量方法采用具有泛化性的方法减弱模型对训练数据量的要求, 例如 ASAM. 相较于基于先验知识的方法, 泛化性方法具有更弱的先验知识依赖, 以及更高效的训练过程, 研发泛化性模型训练技术将是 FSMD 发展的方向之一.

5.3 应用场景

FSMD 作为网络空间安全领域的常见问题, 可以和多种技术结合, 以构成更加完备的网络防御体系, 具体包括:

结合 FSMD 和威胁情报实现面向 ZSL 的恶意软件检测, ZSL 的实现基于相同领域先验知识和描述信息, 威胁情报对网络安全攻击进行了丰富的描述, 如果描述可以转换为仿真 s 或仿真 $\boldsymbol{v}$, 那么就为 ZSL 转化为 FSMD 提供了应用基础, 进而促进了对 zero-sample 恶意软件类型的学习. 在本场景中, 应该着重攻克持续学习问题和模型泛化学习问题.

结合 FSMD 与蜜罐技术实现互补, FSMD 通常检出的对象是当前网络环境中的稀有恶意软件, 而蜜罐技术是网络空间中的新型攻击诱发节点, 出现稀有、功能复杂以及伪装性质的恶意软件的概率较大, FSMD 从蜜罐中发现新威胁, 蜜罐为 FSMD 提供新样本, 二者实现共同进化. 在本场景中, 应该着重关注对抗数据的干扰问题.

结合网络流量等可重现攻击数据提高 FSMD 方法的检测效率, 对于已知攻击手段的网络攻击, 可以通过多次微调攻击过程或数据实现大量相似流量, 这些可以重现的数据可以为基于数据增强的 FSMD 方法提供更加真实可靠的数据, 进而提成 FSMD 方法的检测效率. 在该场景中, 应该着重解决仿真数据的有效性验证问题.

6 结 论

本文对目前的 FSMD 研究进行了介绍和分析. 首先, 本文明确了 FSMD 的问题定义, 将 FSMD 定义为一个 FSL 问题; 其次, 本文将 FSMD 分为 5 个主要过程, 包括: 训练样本集处理、向量嵌入、训练向量集处理、模型训练和样本检测, 并根据每种 FSMD 方法的原理对现有研究分类阐述, 包括基于数据增强的方法、基于元学习的方法和多技术结合的混合方法; 最后, 本文对 FSMD 的背景、技术和场景提出了展望.

References:

[1] Ye YF, Li T, Adjeroh D, Iyengar SS. A survey on malware detection using data mining techniques. ACM Computing Surveys, 2017, 50(3): 41. [doi: 10.1145/3073559]

[2] Qiu JY, Zhang J, Luo W, Pan L, Nepal S, Xiang Y. A survey of Android malware detection with deep neural models. ACM Computing Surveys, 2021, 53(6): 126. [doi: 10.1145/3417978]

[3] He MS, Wang XJ, Zhou JH, Xi YY, Jin L, Wang XL. Deep-feature-based autoencoder network for few-shot malicious traffic detection. Security and Communication Networks, 2021, 2021: 6659022. [doi: 10.1155/2021/6659022]

[4] Al-Garadi MA, Mohamed A, Al-Ali AK, Du XJ, Ali I, Guizani M. A survey of machine and deep learning methods for Internet of Things (IoT) security. IEEE Communications Surveys & Tutorials, 2020, 22(3): 1646–1685. [doi: 10.1109/comst.2020.2988293]

[5] Tran TK, Sato H, Kubo M. One-shot learning approach for unknown malware classification. In: Proc. of the 5th Asian Conf. on Defense Technology (ACDT). Hanoi: IEEE, 2018. 8–13. [doi: 10.1109/ACDT.2018.8593203]

[6] Bai YD, Xing ZC, Li XH, Feng ZY, Ma DY. Unsuccessful story about few shot malware family classification and siamese network to the rescue. In: Proc. of the 42nd IEEE/ACM Int'l Conf. on Software Engineering (ICSE). Seoul: IEEE, 2020. 1560–1571.

[7] Wang YQ, Yao QM, Kwok JT, Ni LM. Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys, 2020, 53(3): 63. [doi: 10.1145/3386252]

[8] Zuffi S, Kanazawa A, Black MJ. Lions and tigers and bears: Capturing non-rigid, 3D, articulated shape from images. In: Proc. of the 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 3955–3963. [doi: 10.1109/CVPR.2018.00416]

[9] Clarivate. Web of Science. 2022. https://www.webofscience.com/wos/alldb/basic-search

[10] Duan RX, Li D, Tong Q, Yang T, Liu XT, Liu XL. A survey of few-shot learning: An effective method for intrusion detection. Security and Communication Networks, 2021, 2021: 4259629. [doi: 10.1155/2021/4259629]

[11] Gibert D, Mateu C, Planes J, Vicens R. Classification of malware by using structural entropy on convolutional neural networks. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence and the 30th Innovative Applications of Artificial Intelligence Conf. and the 8th AAAI Symp. on Educational Advances in Artificial Intelligence. New Orleans: AAAI Press, 2018. 952.

[12] Pastrana S, Suarez-Tangil G. A first look at the crypto-mining malware ecosystem: A decade of unrestricted wealth. In: Proc. of the 2019 Internet Measurement Conf. Amsterdam: Association for Computing Machinery, 2019. 73–86. [doi: 10.1145/3355369.3355576]

[13] Backdoor: Win32/Simda threat description. 2017. https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Backdoor:Win32/Simda

[14] Win32/FakeRean threat description. 2017. https://www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Win32/FakeRean

[15] Jiang T, Gradus JL, Rosellini AJ. Supervised machine learning: A brief primer. Behavior Therapy, 2020, 51(5): 675–687. [doi: 10.1016/j.beth.2020.05.002]

[16] Kasarapu S, Shukla S, Hassan R, Sasan A, Homayoun H, Sai Manoj PD. CAD-FSL: Code-aware data generation based few-shot learning for efficient malware detection. In: Proc. of the 2022 Great Lakes Symp. on VLSI. Irvine: Association for Computing Machinery, 2022. 507–512. [doi: 10.1145/3526241.3530825]

[17] Hospedales T, Antoniou A, Micaelli P, Storkey A. Meta-learning in neural networks: A survey. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2022, 44(9): 5149–5169. [doi: 10.1109/TPAMI.2021.3079209]

[18] Ye TP, Li GL, Ahmad I, Zhang CF, Lin X, Li JH. FLAG: Few-shot latent Dirichlet generative learning for semantic-aware traffic detection. IEEE Trans. on Network and Service Management, 2022, 19(1): 73–88. [doi: 10.1109/TNSM.2021.3131266]

[19] Wang W, Zhu M, Zeng XW, Ye XZ, Sheng YQ. Malware traffic classification using convolutional neural network for representation learning. In: Proc. of the 2017 Int'l Conf. on Information Networking (ICOIN). Da Nang: IEEE, 2017. 712–717. [doi: 10.1109/ICOIN.2017.7899588]

[20] Intrusion detection evaluation dataset (CIC-IDS2017). 2023. https://www.unb.ca/cic/datasets/ids-2017.html

[21] Li YC, Lian Y, Wang JJ, Chen YH, Wang CM, Pu SL. Few-shot one-class domain adaptation based on frequency for iris presentation attack detection. In: Proc. of the 2022 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP). Singapore: IEEE, 2022. 2480–2484. [doi: 10.1109/ICASSP43922.2022.9746635]

[22] Yambay D, Becker B, Kohli N, Yadav D, Czajka A, Bowyer KW, Schuckers S, Singh R, Vatsa M, Noore A, Gragnaniello D, Sansone C, Verdoliva L, He LX, Ru YW, Li HQ, Liu NF, Sun ZN, Tan TN. LivDet iris 2017—Iris liveness detection competition 2017. In: Proc. of the 2017 IEEE Int'l Joint Conf. on Biometrics (IJCB). Denver: IEEE, 2017. 733–741. [doi: 10.1109/BTAS.2017.8272763]

[23] Park S, Gondal I, Kamruzzaman J, Zhang L. One-shot malware outbreak detection using spatio-temporal isomorphic dynamic features. In: Proc. of the 18th IEEE Int'l Conf. on Trust, Security and Privacy in Computing and Communications and the 13th IEEE Int'l Conf. on Big Data Science and Engineering (TrustCom/BigDataSE). Rotorua: IEEE, 2019. 751–756. [doi: 10.1109/TrustCom/BigDataSE.2019.00108]

[24] Qiang Q, Cheng M, Hu Y, Zhou Y, Sun JW, Ding Y, Qi ZS, Jiao F. An incremental malware classification approach based on few-shot learning. In: Proc. of the 2022 IEEE Int'l Conf. on Communications. Seoul: IEEE, 2022. 2682–2687. [doi: 10.1109/ICC45855.

2022.9838295]

[25] Ki Y, Kim E, Kim HK. A novel approach to detect malware based on API call sequence analysis. Int'l Journal of Distributed Sensor Networks, 2015, 11(6): 659101. [doi: 10.1155/2015/659101]

[26] Wang FW, Chai GF, Li QR, Wang CG. Classification of few-sample malware based on parameter-optimized meta-learning and hard example mining. Journal of Wuhan University (Natural Science Edition), 2022, 68(1): 17–25. [doi: 10.14188/j.1671-8836.2021.2008]

[27] Nataraj L, Karthikeyan S, Jacob G, Manjunath BS. Malware images: Visualization and automatic classification. In: Proc. of the 8th Int'l Symp. on Visualization for Cyber Security. Pittsburgh: Association for Computing Machinery, 2011. 4. [doi: 10.1145/2016904.2016908]

[28] Chen MT, Wang YJ, Zhu XT. Few-shot website fingerprinting attack with meta-bias learning. Pattern Recognition, 2022, 130: 108739. [doi: 10.1016/j.patcog.2022.108739]

[29] Rimmer V, Preuveneers D, Juarez M, Van Goethem T, Joosen W. Automated website fingerprinting through deep learning. In: Proc. of the 25th Annual Network and Distributed System Security Symp. San Diego: The Internet Society, 2018.

[30] Gong JJ, Wang T. Zero-delay lightweight defenses against website fingerprinting. In: Proc. of the 29th USENIX Conf. on Security Symp. Berkeley: USENIX Association, 2020. 41.

[31] Yang JC, Li HW, Shao S, Zou FT, Wu Y. FS-IDS: A framework for intrusion detection based on few-shot learning. Computers & Security, 2022, 122: 102899. [doi: 10.1016/j.cose.2022.102899]

[32] Li TT, Hong Z, Liu LS, Wen ZY, Yu L. *META-WF*: Meta-learning-based few-shot wireless impersonation detection for Wi-Fi networks. IEEE Communications Letters, 2021, 25(11): 3585–3589. [doi: 10.1109/LCOMM.2021.3112518]

[33] Kolias C, Kambourakis G, Stavrou A, Gritzalis S. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. IEEE Communications Surveys & Tutorials, 2016, 18(1): 184–208. [doi: 10.1109/COMST.2015.2402161]

[34] Li KH, Ma WG, Duan HW, Xie H, Zhu JX. Few-shot IoT attack detection based on RFP-CNN and adversarial unsupervised domain-adaptive regularization. Computers & Security, 2022, 121: 102856. [doi: 10.1016/J.COSE.2022.102856]

[35] Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Computers & Security, 2012, 31(3): 357–374. [doi: 10.1016/j.cose.2011.12.012]

[36] Alsaedi A, Moustafa N, Tari Z, Mahmood A, Anwar A. TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access, 2020, 8: 165130–165150. [doi: 10.1109/ACCESS.2020.3022862]

[37] Tran K, Sato H, Kubo M. MANNWARE: A malware classification approach with a few samples using a memory augmented neural network. Information, 2020, 11(1): 51. [doi: 10.3390/info11010051]

[38] Nappa A, Rafique MZ, Caballero J. The MALICIA dataset: Identification and analysis of drive-by download operations. Int'l Journal of Information Security, 2015, 14(1): 15–33. [doi: 10.1007/s10207-014-0248-7]

[39] VirusTotal. 2023. https://www.virustotal.com/gui/home/upload

[40] Tran TK, Sato H, Kubo M. Image-based unknown malware classification with few-shot learning models. In: Proc. of the 11th Int'l Symp. on Computing and Networking Workshops (CANDARW). Nagasaki: IEEE, 2019. 401–407. [doi: 10.1109/CANDARW.2019.00075]

[41] Wang P, Tang ZJ, Wang JF. A novel few-shot malware classification approach for unknown family recognition with multi-prototype modeling. Computers & Security, 2021, 106: 102273. [doi: 10.1016/j.cose.2021.102273]

[42] VirusShare_00177. md5. 2023. https://virusshare.com/hashfiles/VirusShare_00177.md5

[43] Hsiao SC, Kao DY, Liu ZY, Tso R. Malware image classification using one-shot learning with siamese networks. Procedia Computer Science, 2019, 159: 1863–1871. [doi: 10.1016/j.procs.2019.09.358]

[44] VirusShare. com. 2023. https://virusshare.com

[45] Zhu JT, Jang-Jaccard J, Singh A, Welch I, Al-Sahaf H, Camtepe S. A few-shot meta-learning based siamese neural network using entropy features for ransomware classification. Computers & Security, 2022, 117: 102691. [doi: 10.1016/J.COSE.2022.102691]

[46] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma SA, Huang ZH, Karpathy A, Khosla A, Bernstein M, Berg AC, Li FF. ImageNet large scale visual recognition challenge. Int'l Journal of Computer Vision, 2015, 115(3): 211–252. [doi: 10.1007/s11263-015-0816-y]

[47] Ale L, Li LZ, Kar D, Zhang N, Palikhe A. Few-shot learning to classify Android malwares. In: Proc. of the 5th IEEE Int'l Conf. on Signal and Image Processing (ICSIP). Nanjing: IEEE, 2020. 1001–1007. [doi: 10.1109/ICSIP49896.2020.9339429]

[48] Investigation of the Android malware (CIC-InvesAndMal2019). 2023. https://www.unb.ca/cic/datasets/invesandmal2019.html

[49] Xu CY, Shen JZ, Du X. A method of few-shot network intrusion detection based on meta-learning framework. IEEE Trans. on Information Forensics and Security, 2020, 15: 3540–3552. [doi: 10.1109/TIFS.2020.2991876]

[50] Virustotal package—Rdocumentation. 2022. https://www.rdocumentation.org/packages/virustotal/versions/0.2.1

[51] Chai YH, Qiu J, Yin LH, Zhang LJ, Gupta BB, Tian ZH. From data and model levels: Improve the performance of few-shot malware classification. IEEE Trans. on Network and Service Management, 2022, 19(4): 4248–4261. [doi: 10.1109/TNSM.2022.3200866]

[52] Yang LM, Ciptadi A, Laziuk I, Ahmadzadeh A, Wang G. BODMAS: An open dataset for learning based temporal analysis of PE malware. In: Proc. of the 2021 IEEE Security and Privacy Workshops (SPW). San Francisco: IEEE, 2021. 78–84. [doi: 10.1109/SPW53761.2021.00020]

[53] Fang ZY, Wang JF, Li BY, Wu SQ, Zhou YJ, Huang HY. Evading anti-malware engines with deep reinforcement learning. IEEE Access, 2019, 7: 48867–48879. [doi: 10.1109/ACCESS.2019.2908033]

[54] Tang ZJ, Wang P, Wang JF. ConvProtoNet: Deep prototype induction towards better class representation for few-shot malware classification. Applied Sciences, 2020, 10(8): 2847. [doi: 10.3390/app10082847]

[55] Arp D, Spreitzenbarth M, Hubner M, Gascon H, Rieck K. DREBIN: Effective and explainable detection of Android malware in your pocket. In: Proc. of the 21th Annual Network and Distributed System Security Symp. San Diego: The Internet Society, 2014. 23–26.

[56] Chai YH, Du L, Qiu J, Yin LH, Tian ZH. Dynamic prototype network based on sample adaptation for few-shot malware detection. IEEE Trans. on Knowledge and Data Engineering, 2022, 35(5): 4754–4766. [doi: 10.1109/TKDE.2022.3142820]

[57] Zhou YJ, Jiang XX. Dissecting Android malware: Characterization and evolution. In: Proc. of the 2012 IEEE Symp. on Security and Privacy. San Francisco: IEEE, 2012. 95–109. [doi: 10.1109/SP.2012.16]

[58] Wei FG, Li YP, Roy S, Ou XM, Zhou W. Deep ground truth analysis of current Android malware. In: Proc. of the 14th Int'l Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment. Bonn: Springer, 2017. 252–276. [doi: 10.1007/978-3-319-60876-1_12]

[59] Rong CD, Gou GP, Hou CS, Li Z, Xiong G, Guo L. UMVD-FSL: Unseen malware variants detection using few-shot learning. In: Proc. of the 2021 Int'l Joint Conf. on Neural Networks (IJCNN). Shenzhen: IEEE, 2021. 1–8. [doi: 10.1109/IJCNN52387.2021.9533759]

[60] Datasets Overview—Stratosphere IPS. 2023. https://www.stratosphereips.org/datasets-overview

[61] Feng TT, Qi Q, Wang JY, Liao JX. Few-shot class-adaptive anomaly detection with model-agnostic meta-learning. In: Proc. of the 2021 IFIP Networking Conf. (IFIP Networking). Espoo: IEEE, 2021. 1–9. [doi: 10.23919/IFIPNetworking52078.2021.9472814]

[62] Android Malware Dataset (CIC-AndMal2017). 2022. https://www.unb.ca/cic/datasets/andmal2017.html

[63] Yuan SH, Zheng PP, Wu XT, Tong HH. Few-shot insider threat detection. In: Proc. of the 29th ACM Int'l Conf. on Information & Knowledge Management. New York: Association for Computing Machinery, 2020. 2289–2292. [doi: 10.1145/3340531.3412161]

[64] Glasser J, Lindauer B. Bridging the gap: A pragmatic approach to generating insider threat data. In: Proc. of the 2013 IEEE Security and Privacy Workshops. San Francisco: IEEE, 2013. 98–104. [doi: 10.1109/SPW.2013.37]

[65] Kumar S, Spezzano F, Subrahmanian V. VEWS: A wikipedia vandal early warning system. In: Proc. of the 21th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Sydney: Association for Computing Machinery, 2015. 607–616. [doi: 10.1145/2783258.2783367]

[66] Pouyanfar S, Sadiq S, Yan YL, Tian HM, Tao YD, Reyes MP, Shyu ML, Chen SC, Iyengar SS. A survey on deep learning: Algorithms, techniques, and applications. ACM Computing Surveys, 2018, 51(5): 92. [doi: 10.1145/3234150]

[67] Benaim S, Wolf L. One-shot unsupervised cross domain translation. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2018. 2108–2118.

[68] Shyam P, Gupta S, Dukkipati A. Attentive recurrent comparators. In: Proc. of the 34th Int'l Conf. on Machine Learning. San Diego: JMLR.org, 2017. 3173–3181.

[69] Lake BM, Salakhutdinov R, Tenenbaum JB. Human-level concept learning through probabilistic program induction. Science, 2015, 350(6266): 1332–1338. [doi: 10.1126/science.aab3050]

[70] Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T. Meta-learning with memory-augmented neural networks. In: Proc. of the 33rd Int'l Conf. on Int'l Conf. on Machine Learning. New York: JMLR.org, 2016. 1842–1850.

[71] Goodfellow I, Pouget-Abadie J, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y. Generative adversarial networks. Communications of the ACM, 2020, 63(11): 139–144. [doi: 10.1145/3422622]

[72] Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. The Journal of machine Learning Research, 2003, 3: 993–1022.

[73] Ahmed N, Natarajan T, Rao KR. Discrete cosine transform. IEEE Trans. on Computers, 1974, C-23(1): 90–93. [doi: 10.1109/T-C.1974.223784]

[74] Oksuz K, Cam BC, Kalkan S, Akbas E. Imbalance problems in object detection: A review. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2021, 43(10): 3388–3415. [doi: 10.1109/TPAMI.2020.2981890]

[75] Chen ZX, Yan QB, Han HB, Wang SS, Peng LZ, Wang L, Yang B. Machine learning based mobile malware detection using highly

imbalanced network traffic. Information Sciences, 2018(433–434): 346–364. [doi: 10.1016/j.ins.2017.04.044]

[76] Batista GEAPA, Prati RC, Monard MC. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter, 2004, 6(1): 20–29. [doi: 10.1145/1007730.1007735]

[77] Rice L, Wong E, Kolter JZ. Overfitting in adversarially robust deep learning. In: Proc. of the 37th Int'l Conf. on Machine Learning. San Diego: JMLR.org, 2020. 749.

[78] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 2002, 16: 321–357. [doi: 10.1613/jair.953]

[79] Fabbri R, Costa LDF, Torelli JC, Bruno OM. 2D Euclidean distance transform algorithms: A comparative survey. ACM Computing Survey, 2008, 40(1): 2. [doi: 10.1145/1322432.1322434]

[80] Ye J. Cosine similarity measures for intuitionistic fuzzy sets and their applications. Mathematical and Computer Modelling, 2011, 53(1–2): 91–97. [doi: 10.1016/j.mcm.2010.07.022]

[81] Hui H, Wang WY, Mao BH. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Proc. of the 2005 Int'l Conf. on Intelligent Computing. Hefei: Springer, 2005. 878–887. [doi: 10.1007/11538059_91]

[82] Douzas G, Bacao F, Last F. Improving imbalanced learning through a heuristic oversampling method based on K-means and SMOTE. Information Sciences, 2018, 465: 1–20. [doi: 10.1016/j.ins.2018.06.056]

[83] Wang Heyong. Combination approach of SMOTE and biased-SVM for imbalanced datasets. In: Proc. of the 2008 IEEE Int'l Joint Conf. on Neural Networks (IEEE World Congress on Computational Intelligence). Hong Kong: IEEE, 2008. 228–231. [doi: 10.1109/IJCNN.2008.4633794]

[84] Mukherjee M, Khushi M. SMOTE-ENC: A novel SMOTE-based method to generate synthetic data for nominal and continuous features. Applied System Innovation, 2021, 4(1): 18. [doi: 10.3390/asi4010018]

[85] Shi N, Liu XM, Guan Y. Research on K-means clustering algorithm: An improved K-means clustering algorithm. In: Proc. of the 3rd Int'l Symp. on Intelligent Information Technology and Security Informatics. Jinggangshan: IEEE, 2010. 63–67. [doi: 10.1109/IITSI.2010.74]

[86] Wang JT, Qian YH, Li FF, Liu GQ. Support vector machine with eliminating the random consistency. Journal of Computer Research and Development, 2020, 57(8): 1581–1593 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2020.20200127]

[87] Shen M, Zhang J, Zhu LH, Xu K, Zhang KX, Li HZ, Tang XY. SVM training mechanism for secure sharing of credit data. Chinese Journal of Computers, 2021, 44(4): 696–708 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2021.00696]

[88] Chen Q, Zhang L, Jiang J, Huang XY. Review analysis method based on support vector machine and latent Dirichlet allocation. Ruan Jian Xue Bao/Journal of Software, 2019, 30(5): 1547–1560 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5731.htm [doi: 10.13328/j.cnki.jos.005731]

[89] Rodríguez P, Bautista MA, Gonzàlez J, Escalera S. Beyond one-hot encoding: Lower dimensional target embedding. Image and Vision Computing, 2018, 75: 21–31. [doi: 10.1016/j.imavis.2018.04.004]

[90] He HB, Bai Y, Garcia EA, Li ST. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Proc. of the 2008 IEEE Int'l Joint Conf. on Neural Networks (IEEE world Congress on Computational Intelligence). Hong Kong: IEEE, 2008. 1322–1328. [doi: 10.1109/IJCNN.2008.4633969]

[91] Raff E, Nicholas C. Malware classification and class imbalance via stochastic hashed LZJD. In: Proc. of the 10th ACM Workshop on Artificial Intelligence and Security. Dallas: Association for Computing Machinery, 2017. 111–120. [doi: 10.1145/3128572.3140446]

[92] Tan XP, Su SJ, Huang ZP, Guo XJ, Zuo Z, Sun XY, Li LQ. Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm. Sensors, 2019, 19(1): 203. [doi: 10.3390/s19010203]

[93] Niu ZY, Zhong GQ, Yu H. A review on the attention mechanism of deep learning. Neurocomputing, 2021, 452: 48–62. [doi: 10.1016/j.neucom.2021.03.091]

[94] Sami A, Yadegari B, Rahimi H, Peiravian N, Hashemi S, Hamze A. Malware detection based on mining API calls. In: Proc. of the 2010 ACM Symp. on Applied Computing. Sierre: ACM, 2010. 1020–1025. [doi: 10.1145/1774088.1774303]

[95] Han WJ, Xue JF, Wang Y, Huang L, Kong ZX, Mao LM. MalDae: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics. Computers & Security, 2019, 83: 208–233. [doi: 10.1016/j.cose.2019.02.007]

[96] Sirinam P, Mathews N, Rahman MS, Wright M. Triplet fingerprinting: More practical and portable website fingerprinting with N-shot learning. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: Association for Computing Machinery, 2019. 1131–1148. [doi: 10.1145/3319535.3354217]

[97] Chen MT, Wang YJ, Qin ZQ, Zhu XT. Few-shot website fingerprinting attack with data augmentation. Security and Communication Networks, 2021, 2021: 2840289. [doi: 10.1155/2021/2840289]

[98] Chen MT, Wang YJ, Xu HZ, Zhu XT. Few-shot website fingerprinting attack. Computer Networks, 2021, 198: 108298. [doi:

10.1016/j.comnet.2021.108298]

[99] Xie KP, Lu Y, Jin ZM, Liu YQ, Gong C, Chen XW, Li T. FAQ-CNN: A flexible acceleration framework for quantized convolutional neural networks on embedded FPGAs. Journal of Computer Research and Development, 2022, 59(7): 1409–1427 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.20210142]

[100] Tian X, Wang L, Ding Q. Review of image semantic segmentation based on deep learning. Ruan Jian Xue Bao/Journal of Software, 2019, 30(2): 440–468 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5659.htm [doi: 10.13328/j.cnki.jos.005659]

[101] Zhou FY, Jin LP, Dong J. Review of convolutional neural network. Chinese Journal of Computers, 2017, 40(6): 1229–1251 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2017.01229]

[102] Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of Machine Learning Research, 2010, 11: 3371–3408.

[103] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798–1828. [doi: 10.1109/TPAMI.2013.50]

[104] Yin Z, Shen YY. On the dimensionality of word embedding. In: Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2018. 895–906.

[105] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proc. of the 34th Int'l Conf. on Machine Learning. Sydney: JMLR.org, 2017. 1126–1135.

[106] Diro AA, Chilamkurti N. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Generation Computer Systems, 2018, 82: 761–768. [doi: 10.1016/j.future.2017.08.043]

[107] Collier M, Beel J. Implementing neural turing machines. In: Kůrková V, Manolopoulos Y, Hammer B, Iliadis L, Maglogiannis I, eds. Artificial Neural Networks and Machine Learning—ICANN. Cham: Springer, 2018. 94–104. [doi: 10.1007/978-3-030-01424-7_10]

[108] Kwon J, Kim J, Park H, Choi KI. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In: Proc. of the 38th Int'l Conf. on Machine Learning. New York: PMLR, 2021. 5905–5914.

[109] Bromley J, Guyon I, LeCun Y, Sackinger E, Shah R. Signature verification using a "siamese" time delay neural network. In: Proc. of the 6th Int'l Conf. on Neural Information Processing Systems. Denver: Morgan Kaufmann Publishers Inc., 1993. 737–774.

[110] Shen YY, Zhang FZ, Liu D, Pu WH, Zhang QL. Manhattan-distance IOU loss for fast and accurate bounding box regression and object detection. Neurocomputing, 2022, 500: 99–114. [doi: 10.1016/j.neucom.2022.05.052]

[111] bjlittle/imagehash: A Python perceptual image hashing module. 2023. https://github.com/bjlittle/imagehash/

[112] Sun S, Li XJ, Liu M, Yang B, Guo XB. DNN inference acceleration via heterogeneous IoT devices collaboration. Journal of Computer Research and Development, 2020, 57(4): 709–722 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2020.20190863]

[113] Jiao LC, Sun QG, Yang YT, Feng YX, Li XF. Development, implementation and prospect of FPGA-based deep neural networks. Chinese Journal of Computers, 2022, 45(3): 441–471 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2022.00441]

[114] Li XR, Ji SL, Wu CM, Liu ZG, Deng SG, Cheng P, Yang M, Kong XW. Survey on deepfakes and detection techniques. Ruan Jian Xue Bao/Journal of Software, 2021, 32(2): 496–518 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6140.htm [doi: 10.13328/j.cnki.jos.006140]

[115] Wen YD, Zhang KP, Li ZF, Qiao Y. A discriminative feature learning approach for deep face recognition. In: Proc. of the 14th European Conf. on Computer Vision. Amsterdam: Springer, 2016. 499–515. [doi: 10.1007/978-3-319-46478-7_31]

[116] Vinyals O, Blundell C, Lillicrap T, Kavukcuoglu K, Wierstra D. Matching networks for one shot learning. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems. Barcelona: Curran Associates Inc., 2016. 3637–3645.

[117] Hu CW, Wu CX, Yang YL. Extended S-LSTM based textual entailment recognition. Journal of Computer Research and Development, 2020, 57(7): 1481–1489 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2020.20190522]

[118] Xie Z, Zhou Y, Wu KW, Zhang SR. Activity recognition based on spatial-temporal attention LSTM. Chinese Journal of Computers, 2021, 44(2): 261–274 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2021.00261]

[119] Duan X, Wu JZ, Luo TY, Yang MT, Wu YJ. Vulnerability mining method based on code property graph and attention BiLSTM. Ruan Jian Xue Bao/Journal of Software, 2020, 31(11): 3404–3420 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6061.htm [doi: 10.13328/j.cnki.jos.006061]

[120] Zhou CL, Ma CG, Yang ST. Location privacy-preserving method for LBS continuous KNN query in road networks. Journal of Computer Research and Development, 2015, 52(11): 2628–2644 (in Chinese with English abstract). [doi: 10.7544/issn1000-1239.2015.20140532]

[121] Li C, Shen DR, Zhu MD, Kou Y, Nie TZ, Yu G. kNN query processing approach for content with location and time tags. Ruan Jian Xue Bao/Journal of Software, 2016, 27(9): 2278–2289 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5046.htm [doi:

10.13328/j.cnki.jos.005046]
[122] Zhu L, Qiu YY, Yu S, Yuan S. A fast kNN-based MST outlier detection method. Chinese Journal of Computers, 2017, 40(12): 2856–2870 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2017.02856]
[123] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates, Inc., 2017. 4080–4090.
[124] Sung F, Yang YX, Zhang L, Xiang T, Torr HSP, Hospedales TM. Learning to compare: Relation network for few-shot learning. In: Proc. of the 2018 IEEE Conf. on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 1199–1208. [doi: 10.1109/CVPR.2018.00131]
[125] Dragomiretskiy K, Zosso D. Variational mode decomposition. IEEE Trans. on Signal Processing, 2014, 62(3): 531–544. [doi: 10.1109/TSP.2013.2288675]
[126] Xiao GQ, Li JN, Chen YD, Li KL. MalFCS: An effective malware classification framework with automated feature extraction based on deep convolutional neural networks. Journal of Parallel and Distributed Computing, 2020, 141: 49–58. [doi: 10.1016/j.jpdc.2020.03.012]
[127] Avdiienko V, Kuznetsov K, Gorla A, Zeller A, Arzt S, Rasthofer S, Bodden E. Mining APPs for abnormal usage of sensitive data. In: Proc. of the 37th IEEE/ACM IEEE Int'l Conf. on Software Engineering. Florence: IEEE, 2015. 426–436. [doi: 10.1109/ICSE.2015.61]
[128] Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr HSP. Fully-convolutional siamese networks for object tracking. In: Proc. of the 2016 European Conf. on Computer Vision. Amsterdam: Springer, 2016. 850–865. [doi: 10.1007/978-3-319-48881-3_56]

附中文参考文献:

[26] 王方伟, 柴国芳, 李青茹, 王长广. 基于参数优化元学习和困难样本挖掘的小样本恶意软件分类方法. 武汉大学学报 (理学版), 2022, 68(1): 17–25. [doi: 10.14188/j.1671-8836.2021.2008]
[86] 王婕婷, 钱宇华, 李飞江, 刘郭庆. 消除随机一致性的支持向量机分类方法. 计算机研究与发展, 2020, 57(8): 1581–1593. [doi: 10.7544/issn1000-1239.2020.20200127]
[87] 沈蒙, 张杰, 祝烈煌, 徐恪, 张开翔, 李辉忠, 唐湘云. 面向征信数据安全共享的 SVM 训练机制. 计算机学报, 2021, 44(4): 696–708. [doi: 10.11897/SP.J.1016.2021.00696]
[88] 陈琪, 张莉, 蒋竞, 黄新越. 一种基于支持向量机和主题模型的评论分析方法. 软件学报, 2019, 30(5): 1547–1560. http://www.jos.org.cn/1000-9825/5731.htm [doi: 10.13328/j.cnki.jos.005731]
[99] 谢坤鹏, 卢冶, 靳宗明, 刘义情, 龚成, 陈新伟, 李涛. FAQ-CNN: 面向量化卷积神经网络的嵌入式 FPGA 可扩展加速框架. 计算机研究与发展, 2022, 59(7): 1409–1427. [doi: 10.7544/issn1000-1239.20210142]
[100] 田萱, 王亮, 丁琪. 基于深度学习的图像语义分割方法综述. 软件学报, 2019, 30(2): 440–468. http://www.jos.org.cn/1000-9825/5659.htm [doi: 10.13328/j.cnki.jos.005659]
[101] 周飞燕, 金林鹏, 董军. 卷积神经网络研究综述. 计算机学报, 2017, 40(6): 1229–1251. [doi: 10.11897/SP.J.1016.2017.01229]
[112] 孙胜, 李叙晶, 刘敏, 杨博, 过晓冰. 面向异构 IoT 设备协作的 DNN 推断加速研究. 计算机研究与发展, 2020, 57(4): 709–722. [doi: 10.7544/issn1000-1239.2020.20190863]
[113] 焦李成, 孙其功, 杨育婷, 冯雨歆, 李秀芳. 深度神经网络 FPGA 设计进展、实现与展望. 计算机学报, 2022, 45(3): 441–471. [doi: 10.11897/SP.J.1016.2022.00441]
[114] 李旭嵘, 纪守领, 吴春明, 刘振广, 邓水光, 程鹏, 杨珉, 孔祥维. 深度伪造与检测技术综述. 软件学报, 2021, 32(2): 496–518. http://www.jos.org.cn/1000-9825/6140.htm [doi: 10.13328/j.cnki.jos.006140]
[117] 胡超文, 邬昌兴, 杨亚连. 基于扩展的 S-LSTM 的文本蕴含识别. 计算机研究与发展, 2020, 57(7): 1481–1489. [doi: 10.7544/issn1000-1239.2020.20190522]
[118] 谢昭, 周义, 吴克伟, 张顺然. 基于时空关注度 LSTM 的行为识别. 计算机学报, 2021, 44(2): 261–274. [doi: 10.11897/SP.J.1016.2021.00261]
[119] 段旭, 吴敬征, 罗天悦, 杨牧天, 武延军. 基于代码属性图及注意力双向 LSTM 的漏洞挖掘方法. 软件学报, 2020, 31(11): 3404–3420. http://www.jos.org.cn/1000-9825/6061.htm [doi: 10.13328/j.cnki.jos.006061]
[120] 周长利, 马春光, 杨松涛. 路网环境下保护 LBS 位置隐私的连续 KNN 查询方法. 计算机研究与发展, 2015, 52(11): 2628–2644. [doi: 10.7544/issn1000-1239.2015.20140532]
[121] 李晨, 申德荣, 朱命冬, 寇月, 聂铁铮, 于戈. 一种对时空信息的 kNN 查询处理方法. 软件学报, 2016, 27(9): 2278–2289. http://www.jos.org.cn/1000-9825/5046.htm [doi: 10.13328/j.cnki.jos.005046]
[122] 朱利, 邱媛媛, 于帅, 原盛. 一种基于快速 k-近邻的最小生成树离群检测方法. 计算机学报, 2017, 40(12): 2856–2870. [doi: 10.11897/SP.J.1016.2017.02856]

刘昊(1996－), 男, 博士生, 主要研究领域为网络空间安全, 恶意软件检测, 机器学习.

田志宏(1978－), 男, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为网络攻防对抗, 网络靶场, 主动实时防护.

仇晶(1983－), 女, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为网络空间安全威胁感知领域基础理论, 先进智能算法设计.

刘园(1986－), 女, 博士, 教授, 博士生导师, CCF 杰出会员, 主要研究领域为网络安全, 机制设计, 博弈理论.

方滨兴(1960－), 男, 博士, 教授, 博士生导师, 主要研究领域为计算机网络, 信息安全.