



(12) 发明专利

(10) 授权公告号 CN 116861431 B

(45) 授权公告日 2023. 11. 21

(21) 申请号 202311132105.8

(22) 申请日 2023.09.05

(65) 同一申请的已公布的文献号
申请公布号 CN 116861431 A

(43) 申请公布日 2023.10.10

(73) 专利权人 国网山东省电力公司信息通信公司

地址 250000 山东省济南市市中区经三路
17号

专利权人 山东省计算中心(国家超级计算
济南中心)

(72) 发明人 刘子函 黄华 李宁 任乐
王云霄 赵丽娜 张婕 张文斌
刘维特 赵大伟 徐丽娟

(74) 专利代理机构 济南圣达知识产权代理有限公司 37221

专利代理师 黄海丽

(51) Int. Cl.

G06F 21/56 (2013.01)

G06F 8/53 (2018.01)

G06V 30/41 (2022.01)

G06V 30/166 (2022.01)

G06V 30/18 (2022.01)

G06V 30/19 (2022.01)

G06V 10/54 (2022.01)

G06V 10/82 (2022.01)

(56) 对比文件

CN 115935360 A, 2023.04.07

CN 114936371 A, 2022.08.23

CN 113935033 A, 2022.01.14

CN 116389030 A, 2023.07.04

CN 115292702 A, 2022.11.04

CN 115630358 A, 2023.01.20

CN 113806746 A, 2021.12.17

CN 111552964 A, 2020.08.18

CN 116340944 A, 2023.06.27

US 2022138319 A1, 2022.05.05 (续)

审查员 刘旭恒

权利要求书3页 说明书7页 附图3页

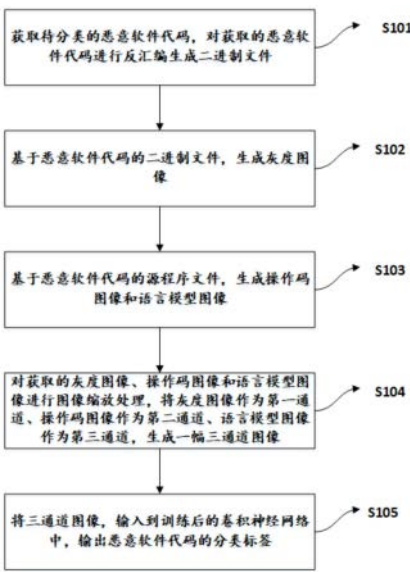
(54) 发明名称

基于多通道图像和神经网络的恶意软件分类方法及系统

(57) 摘要

本发明涉及恶意软件分类技术领域,公开了基于多通道图像和神经网络的恶意软件分类方法及系统,其中方法,包括:获取待分类的恶意软件代码,对获取的恶意软件代码进行反汇编生成二进制文件;基于恶意软件代码的二进制文件,生成灰度图像;基于恶意软件代码的源程序文件,生成操作码图像和语言模型图像;对获取的语言模型图像进行图像缩放处理,将灰度图像作为第一通道、操作码图像作为第二通道、语言模型图像作为第三通道,生成一幅三通道图像;将三通道图像,输入到训练后的卷积神经网络中,输出恶意软件代码的分类标签。在有效保留恶意软件样本信息的基础上,提高分类检测的效率和

准确性。



CN 116861431 B

[转续页]

[接上页]

(56) 对比文件

孙国辉.基于迁移学习的安卓恶意软件检测研究.《中国优秀硕士学位论文全文数据库(信息科技辑)》.2022,(第6期),全文.

Kumar, Sanjeev等.Identification of malware families using stacking of textural features and machine learning.《EXPERT SYSTEMS WITH APPLICATIONS》.2022,

第208卷全文.

侯留洋;罗森林;潘丽敏;张笈.融合多特征的Android恶意软件检测方法.信息安全.2020,(第01期),全文.

修扬;刘嘉勇.基于操作码序列频率向量和行为特征向量的恶意软件检测.信息安全与通信保密.2016,(第09期),全文.

1. 基于多通道图像和神经网络的恶意软件分类方法,其特征是,包括:
获取待分类的恶意软件代码,对获取的恶意软件代码进行反汇编生成二进制文件;
基于恶意软件代码的二进制文件,生成灰度图像;
基于恶意软件代码的源程序文件,生成操作码图像和语言模型图像;
对获取的语言模型图像进行图像缩放处理,将灰度图像作为第一通道、操作码图像作为第二通道、语言模型图像作为第三通道,生成一幅三通道图像;
将三通道图像,输入到训练后的卷积神经网络中,输出恶意软件代码的分类标签;
其中,生成操作码图像,具体包括:
依次遍历恶意软件所有操作码,对恶意样本文件进行操作码提取,得到操作码序列;
遍历操作码序列,记录相邻操作码对出现的次数;所述相邻操作码对,包括:每个操作码和其后继操作码;
统计每个操作码对在样本中出现的次数,根据相邻操作码对的次数,计算转移概率;转移概率,等于相邻操作码对出现的次数除以前一个操作码的总次数所得的商值;
创建一个大小为 $N \times N$ 的转移概率矩阵,其中 N 为256,转移概率矩阵的每个元素表示每个操作码到其它操作码的转移概率;根据转移概率矩阵,生成操作码图像;
所述基于恶意软件代码的源程序文件,生成操作码图像和语言模型图像,其中,语言模型图像的生成过程包括:
将每三个操作码作为一个子序列,将操作码序列拆分为若干个子序列;
将每个子序列转换成一个0~255之间的整数,作为对应字节的灰度像素值;
将所有操作码对应的灰度像素值,按照恶意样本中的二进制文件顺序排列,得到一个灰度像素序列;
将灰度像素序列分成若干段,每段包含256个连续的灰度像素;如果最后一段灰度像素不足256个,用0填充,直到长度为256,宽 W 则根据恶意样本文件大小与长度的比值进行确定;
将每一段256个灰度像素序列转换成一个 $256 \times W$ 的灰度图像;将填充后的矩阵作为灰度图像的像素值表示,使用灰度编码将矩阵中的整数值映射为图像像素值。
2. 如权利要求1所述的基于多通道图像和神经网络的恶意软件分类方法,其特征是,基于恶意软件代码的二进制文件,生成灰度图像,具体包括:
读取恶意软件代码二进制文件中的原始字节数据;
将长度为8的比特串转换成一个0~255之间的整数,作为对应字节的灰度像素值;
将所有字节对应的灰度像素值,按照恶意样本中的二进制文件中的顺序排列,得到一个灰度像素序列;
将灰度像素序列分成若干段,每段包含 W 个连续的灰度像素;如果最后一段灰度像素不足 W 个,则在末尾填充0,直到长度为 W ; W 取值为256;
将每一段灰度像素序列转换成一个灰度矩阵,其中矩阵的行数为恶意样本文件的大小与宽度的比值 H ,列数为 W ,即每个矩阵为一个 $H \times W$ 的矩阵;
将所有的灰度矩阵按照二进制文件中的顺序排列,得到一个二维灰度矩阵,继而生成灰度图像。
3. 如权利要求2所述的基于多通道图像和神经网络的恶意软件分类方法,其特征是,基

于恶意软件代码的二进制文件,生成灰度图像,还包括:

确保灰度图像的每个像素的取值范围在0到255之间;

通过将每个像素值除以2,将灰度级别从256减少到128;

根据离散化后的灰度图像,针对距离值为1和角度值为0、45度、90度和135度的情况,分别计算4个灰度共生矩阵;灰度共生矩阵是一个对称矩阵,表示在给定距离和角度的情况下,每对像素值在图像中出现的频率;

将计算得到的4个灰度共生矩阵重新调整为 $128 \times 128 \times 4$ 的形式,每个灰度共生矩阵对应一个角度值;

将4个灰度共生矩阵在水平和垂直方向上堆叠起来,形成一个大小为 256×256 的灰度矩阵图像;灰度矩阵图像包含所有4个角度值的纹理和空间信息;

将灰度矩阵图像的像素值归一化到0到1之间。

4. 如权利要求1所述的基于多通道图像和神经网络的恶意软件分类方法,其特征是,对获取的语言模型图像进行图像缩放处理,包括:

使用双三次插值算法,对语言模型图像进行归一化处理,将语言模型图像调整为相同长和宽。

5. 如权利要求1所述的基于多通道图像和神经网络的恶意软件分类方法,其特征是,将三通道图像,输入到训练后的卷积神经网络中,输出恶意软件代码的分类标签,其中,训练后的卷积神经网络,网络结构采用Resnet网络来实现。

6. 如权利要求5所述的基于多通道图像和神经网络的恶意软件分类方法,其特征是,所述训练后的卷积神经网络,训练过程包括:

构建训练集,所述训练集为已知恶意软件代码分类标签的恶意软件代码集;

将训练集输入到卷积神经网络中,对其进行训练,当损失函数值不再下降时,停止训练,得到训练后的卷积神经网络;

在训练阶段,使用交叉熵损失函数对卷积神经网络模型进行训练,使用优化器的学习模型参数。

7. 如权利要求6所述的基于多通道图像和神经网络的恶意软件分类方法,其特征是,交叉熵损失函数Cross Entropy Loss:

$$\text{Cross Entropy Loss} = -\sum_{i=1}^N p(x_i) \log q(x_i) + (1-p(x_i)) \log (1-q(x_i));$$

其中, N 为类别数, $p(x_i)$ 表示样本的真实分布, $q(x_i)$ 表示模型所预测的分布。

8. 基于多通道图像和神经网络的恶意软件分类系统,其特征是,包括:

获取模块,其被配置为:获取待分类的恶意软件代码,对获取的恶意软件代码进行反汇编生成二进制文件;

第一生成模块,其被配置为:基于恶意软件代码的二进制文件,生成灰度图像;

第二生成模块,其被配置为:基于恶意软件代码的源程序文件,生成操作码图像和语言模型图像;

第三生成模块,其被配置为:对获取的语言模型图像进行图像缩放处理,将灰度图像作为第一通道、操作码图像作为第二通道、语言模型图像作为第三通道,生成一幅三通道图像;

输出模块,其被配置为:将三通道图像,输入到训练后的卷积神经网络中,输出恶意软

件代码的分类标签;

其中,生成操作码图像,具体包括:

依次遍历恶意软件所有操作码,对恶意样本文件进行操作码提取,得到操作码序列;

遍历操作码序列,记录相邻操作码对出现的次数;所述相邻操作码对,包括:每个操作码和其后继操作码;

统计每个操作码对在样本中出现的次数,根据相邻操作码对的次数,计算转移概率;转移概率,等于相邻操作码对出现的次数除以前一个操作码的总次数所得的商值;

创建一个大小为 $N \times N$ 的转移概率矩阵,其中 N 为256,转移概率矩阵的每个元素表示每个操作码到其它操作码的转移概率;根据转移概率矩阵,生成操作码图像;

所述基于恶意软件代码的源程序文件,生成操作码图像和语言模型图像,其中,语言模型图像的生成过程包括:

将每三个操作码作为一个子序列,将操作码序列拆分为若干个子序列;

将每个子序列转换成一个0~255之间的整数,作为对应字节的灰度像素值;

将所有操作码对应的灰度像素值,按照恶意样本中的二进制文件顺序排列,得到一个灰度像素序列;

将灰度像素序列分成若干段,每段包含256个连续的灰度像素;如果最后一段灰度像素不足256个,用0填充,直到长度为256,宽 W 则根据恶意样本文件大小与长度的比值进行确定;

将每一段256个灰度像素序列转换成一个 $256 \times W$ 的灰度图像;将填充后的矩阵作为灰度图像的像素值表示,使用灰度编码将矩阵中的整数值映射为图像像素值。

基于多通道图像和神经网络的恶意软件分类方法及系统

技术领域

[0001] 本发明涉及恶意软件分类技术领域,特别是涉及基于多通道图像和神经网络的恶意软件分类方法及系统。

背景技术

[0002] 本部分的陈述仅仅是提到了与本发明相关的背景技术,并不必然构成现有技术。

[0003] 随着时间的发展,恶意软件的数量每年都在不断增加,还衍生出很多变体,同时恶意软件的危害性也在不断增加。恶意软件可以在未经用户同意的情况下,在计算机或者终端设备上安装运行,损害用户的合法权益。恶意软件的入侵和传播方式有很多,攻击者可以利用web 服务、浏览器和操作系统中的漏洞,或者使用社会工程技术让终端用户运行恶意代码,以达到传播恶意软件的目的。此外,攻击者还使用一些混淆技术,如代码集成、死代码插入、子程序重新排序、指令代替和代码转置等,以规避防火墙、防病毒和网关等传统防御手段的检测,并在一定程度上影响人工分析。因此,如何快速、准确地对恶意软件进行检测与分类,是目前网络安全领域的研究热点。

[0004] 恶意软件检测技术一般分为两类:静态分析方法和动态分析方法。静态分析方法不需要实际运行恶意软件样本,侧重于代码分析。例如,分析恶意二进制文件中的结构和语法,根据家族特征进行分类检测。动态分析方法可以揭示恶意软件的行为,不受混淆技术的影响,同时可以检测出未知的恶意软件样本。但是非常的耗时,操作步骤也非常的繁琐。虽然使用静态分析方法可以解决上述问题,但是却依赖于高效的反病毒引擎和全面的病毒数据库,从而导致样本检测的准确率降低。

[0005] 为了弥补静态分析方法的不足,相关人员开始研究基于可视化的恶意软件检测和分类方法,并且获得了良好的性能。尽管恶意软件的数量和种类不断增加,但是同类家族的恶意软件二进制代码或者汇编代码的核心具有相似性。可视化之后不会改变图像纹理及结构的本质特征,可以有效的对抗恶意软件混淆问题。基于机器学习的可视化技术操作简单,消耗时间短,对数据的依赖程度小。但是,容易受到特征工程的影响以及难以处理大量恶意软件等问题。虽然每天可以识别出大量恶意软件样本,但是单纯依靠人工来对恶意软件进行检测与分类仍然是一件非常不容易的事情。

[0006] 恶意软件数量急剧增加以及变体的出现,对网络安全构成了严重的威胁。传统的恶意软件检测方法需要大量的特征工程,严重影响检测的效率。现有的基于深度学习的恶意软件检测方法存在泛化能力不强、训练时间长等问题。

发明内容

[0007] 为了解决现有技术的不足,本发明提供了基于多通道图像和神经网络的恶意软件分类方法及系统,可以快速准确的对恶意软件进行分类,该方法将恶意软件转化为三通道图像,利用深度卷积神经网络模型与迁移学习相结合,综合灰度共生矩阵、双三次插值、多通道特征提取等技术,在有效保留恶意软件样本信息的基础上,提高分类检测的效率和准

确性。

[0008] 一方面,提供了基于多通道图像和神经网络的恶意软件分类方法;

[0009] 基于多通道图像和神经网络的恶意软件分类方法,包括:

[0010] 获取待分类的恶意软件代码,对获取的恶意软件代码进行反汇编生成二进制文件;

[0011] 基于恶意软件代码的二进制文件,生成灰度图像;

[0012] 基于恶意软件代码的源程序文件,生成操作码图像和语言模型N-gram图像;

[0013] 对获取的语言模型图像进行图像缩放处理,将灰度图像作为第一通道、操作码图像作为第二通道、语言模型N-gram图像作为第三通道,生成一幅三通道图像;

[0014] 将三通道图像,输入到训练后的卷积神经网络中,输出恶意软件代码的分类标签。

[0015] 另一方面,提供了基于多通道图像和神经网络的恶意软件分类系统;

[0016] 基于多通道图像和神经网络的恶意软件分类系统,包括:

[0017] 获取模块,其被配置为:获取待分类的恶意软件代码,对获取的恶意软件代码进行反汇编生成二进制文件;

[0018] 第一生成模块,其被配置为:基于恶意软件代码的二进制文件,生成灰度图像;

[0019] 第二生成模块,其被配置为:基于恶意软件代码的源程序文件,生成操作码图像和语言模型N-gram图像;

[0020] 第三生成模块,其被配置为:对获取的语言模型图像进行图像缩放处理,将灰度图像作为第一通道、操作码图像作为第二通道、语言模型N-gram图像作为第三通道,生成一幅三通道图像;

[0021] 输出模块,其被配置为:将三通道图像,输入到训练后的卷积神经网络中,输出恶意软件代码的分类标签。

[0022] 上述技术方案中的一个技术方案具有如下优点或有益效果:

[0023] 首先,将恶意软件二进制文件转化为三种不同类型的灰度图像,使用灰度共生矩阵法以及双三次插值算法对灰度图像进行处理;然后,将灰度图像合成三维的RGB图像作为特征图像;最后,通过卷积神经网络对恶意软件进行分类。本发明模型泛化能力强,同时可以有有效的保留恶意软件二进制文件中的信息,有效的提高分类的准确率;网络层数和模型参数减少,所以训练过程中消耗的时间和空间要少很多。

[0024] 提出了恶意软件可视化的新方法,将提取的恶意软件二进制文件信息转化为三种不同的灰度图,并融合成三通道RGB图像。可以从多维角度分析恶意软件,有效保留恶意软件二进制文件中的相关信息。

[0025] 相比于对样本图像进行截取或填充法,本发明使用灰度共生矩阵法、插值算法处理灰度图尺寸问题。在不损失图像信息的情况下,能够有效避免特征信息的丢失。

[0026] 本发明提出一种使用卷积神经网络进行恶意软件分类的新框架,该方法不需要逆向分析,在短时间内就可以达到一个很好的训练效果,有效提高恶意软件分类的准确率,增强模型的泛化能力。

附图说明

[0027] 构成本发明的一部分的说明书附图用来提供对本发明的进一步理解,本发明的示

意性实施例及其说明用于解释本发明,并不构成对本发明的不当限定。

[0028] 图1是本发明实施例一的基于图像和深度学习的恶意软件分类架构;

[0029] 图2是本发明实施例一的设计的卷积神经网络架构;

[0030] 图3是本发明实施例一的实线残差网络结构;

[0031] 图4是本发明实施例一的虚线残差网络结构。

具体实施方式

[0032] 应该指出,以下详细说明都是示例性的,旨在对本发明提供进一步的说明。除非另有指明,本文使用的所有技术和科学术语具有与本发明所属技术领域的普通技术人员通常理解相同含义。

[0033] 实施例一

[0034] 本实施例提供了基于多通道图像和神经网络的恶意软件分类方法;

[0035] 如图1所示,基于多通道图像和神经网络的恶意软件分类方法,包括:

[0036] S101:获取待分类的恶意软件代码,对获取的恶意软件代码进行反汇编生成二进制文件;

[0037] S102:基于恶意软件代码的二进制文件,生成灰度图像;

[0038] S103:基于恶意软件代码的源程序文件,生成操作码图像和语言模型N-gram图像;

[0039] S104:对获取的语言模型图像进行图像缩放处理,将灰度图像作为第一通道、操作码图像作为第二通道、语言模型N-gram图像作为第三通道,生成一幅三通道图像;

[0040] S105:将三通道图像,输入到训练后的卷积神经网络中,输出恶意软件代码的分类标签。

[0041] 进一步地,所述S101:获取待分类的恶意软件代码,是通过爬虫算法从网站中抓取得到,该步骤只对恶意软件代码进行抓取,不涉及用户隐私安全。

[0042] 进一步地,所述S102:基于恶意软件代码的二进制文件,生成灰度图像,其中,二进制文件,是指:二进制样本文件中按顺序排列的一系列字节,例如“0x12 0x34 0x56 0x78 ...”。

[0043] 进一步地,所述S102:基于恶意软件代码的二进制文件,生成灰度图像,具体包括:

[0044] S102-1:读取恶意软件代码二进制文件中的原始字节数据;

[0045] S102-2:将长度为8的比特串(也就是连续的字节)转换成一个0~255之间的整数,作为对应字节的灰度像素值;

[0046] S102-3:将所有字节对应的灰度像素值,按照恶意样本中的二进制文件中的顺序排列,得到一个灰度像素序列;

[0047] S102-4:将灰度像素序列分成若干段,每段包含W个连续的灰度像素;如果最后一段灰度像素不足W个,则在末尾填充0,直到长度为W;W取值为256;

[0048] S102-5:将每一段灰度像素序列转换成一个灰度矩阵,其中矩阵的行数为恶意样本文件的大小与宽度的比值H,列数为W,即每个矩阵为一个H×W的矩阵;

[0049] S102-6:将所有的灰度矩阵按照二进制文件中的顺序排列,得到一个二维灰度矩阵,继而生成灰度图像。

[0050] 进一步地,所述S102:基于恶意软件代码的二进制文件,生成灰度图像,还包括:

[0051] S102-7:确保灰度图像的每个像素的取值范围在0到255之间;

[0052] S102-8:灰度图像离散化:通过将每个像素值除以2,将灰度级别从256减少到128;

[0053] S102-9:计算灰度共生矩阵 (GLCM, Gray-level co-occurrence matrix):根据离散化后的灰度图像,针对距离值为1和角度值为0、45度、90度和135度的情况,分别计算4个灰度共生矩阵GLCM;灰度共生矩阵GLCM是一个对称矩阵,表示在给定距离和角度的情况下,每对像素值在图像中出现的频率;

[0054] S102-10:重新调整灰度共生矩阵GLCM:将计算得到的4个灰度共生矩阵GLCM重新调整为 $128 \times 128 \times 4$ 的形式,每个灰度共生矩阵对应一个角度值;

[0055] S102-11:堆叠子矩阵:将4个灰度共生矩阵GLCM在水平和垂直方向上堆叠起来,形成一个大小为 256×256 的灰度矩阵图像;灰度矩阵图像包含所有4个角度值的纹理和空间信息;

[0056] S102-12:归一化像素值:将灰度矩阵图像的像素值归一化到0到1之间。

[0057] 进一步地,所述S103:基于恶意软件代码的源程序文件,生成操作码图像和语言模型N-gram图像,其中,生成操作码图像,具体包括:

[0058] S103-11:依次遍历恶意软件所有操作码,对恶意样本文件进行操作码提取,得到操作码序列;

[0059] S103-12:遍历操作码序列,记录相邻操作码对出现的次数;所述相邻操作码对,包括:每个操作码和其后继操作码;

[0060] S103-13:统计每个操作码对在样本中出现的次数,根据相邻操作码对的次数,计算转移概率;转移概率,等于相邻操作码对出现的次数除以前一个操作码的总次数所得的商值;

[0061] S103-14:创建一个大小为 $N \times N$ 的转移概率矩阵,其中 N 为256,转移概率矩阵的每个元素表示每个操作码到其它操作码的转移概率;根据转移概率矩阵,生成操作码图像。

[0062] 示例性地,假设有一个包含 N 个不同操作码的操作码序列数据,用 $OP_1, OP_2, \dots, OP_N$ 来表示这些操作码,其中, N 表示操作码的总数,即操作码序列中不同操作码的数量。对于每个操作码 OP_i 在序列中的位置,记录相邻的操作码对 $(OP_i, OP_{(i+1)})$,其中, $OP_{(i+1)}$ 是操作码 OP_i 的后继操作码。将这些出现次数记录在 $Count(OP_i, OP_{(i+1)})$ 中。

[0063] 对于每个操作码 OP_i ,计算它在整个序列中出现的总次数,将这些出现次数记录在 $Count(OP_i)$ 中;其中 $Count(X, Y)$ 表示操作码 X 后跟操作码 Y 出现的次数(相邻操作码对 (X, Y) 出现的次数); $Count(X)$ 表示操作码 X 在整个序列中出现的总次数。

[0064] 例如,如果操作码序列为 $[OP_1, OP_2, OP_3, OP_4, OP_5, OP_6, \dots]$,则相邻操作码对为 $[(OP_1, OP_2), (OP_2, OP_3), (OP_3, OP_4), (OP_4, OP_5), (OP_5, OP_6), \dots]$ 。

[0065] 在计算转移概率时,关注的是相邻操作码对,其中前一个操作码作为起始状态,后一个操作码作为结束状态。所以,将相邻操作码对出现的次数除以前一个操作码的总次数,以得到从前一个操作码转移到后一个操作码的概率。

[0066] 举例说明:假设有一个操作码序列数据为 $[OP_1, OP_2, OP_3, OP_1, OP_2, OP_4, OP_1, OP_3]$,其中 OP_1, OP_2, OP_3, OP_4 分别为不同的操作码。在这个序列中,相邻的操作码对有 $(OP_1, OP_2), (OP_2, OP_3), (OP_3, OP_1), (OP_1, OP_2), (OP_2, OP_4), (OP_4, OP_1), (OP_1, OP_3)$ 。我们计算每个相邻操作码对出现的次数,并将它们除以前一个操作码的总次

数,从而得到操作码转移概率矩阵。例如,对于操作码OP1,它出现的总次数为 3,(OP1, OP2) 出现的次数为 2,因此转移概率等于 2/3。

[0067] 示例性地,所述操作码,包括:push、sub、mov、mov、jmp、jz。

[0068] push:将数据压入栈中。它将指定的值(通常是寄存器或内存中的值)放入栈的顶部,栈指针向下移动。

[0069] sub:执行减法运算。它从目标操作数中减去源操作数,并将结果存储在目标操作数中。

[0070] mov:将数据从一个位置(寄存器、内存等)复制到另一个位置。它用于在不同的寄存器、内存地址或寄存器与内存之间传输数据。

[0071] jmp:无条件跳转指令。它将程序的执行无条件地转移到指定的目标地址或标签处,使程序的控制流程发生跳转。

[0072] jz:零标志位跳转指令。它检查标志寄存器中的零标志位,并根据零标志位的状态来决定是否进行跳转。如果零标志位为真(值为0),则跳转到指定的目标地址或标签处。

[0073] 示例性地,将操作码序列视为具有时序特性的操作码流,遍历操作码样本文件中的所有操作码。例如,经过操作码提取后得到的操作码序列为:push、sub、mov、...、mov、jmp、jz、...。选取出现次数最多的255种操作码,其余的种类归为一类,时序特性指的是在时间上的序列性质或顺序关系。

[0074] 通过频率估计概率,计算各个状态之间的相互转移概率,即各个状态之间的转移概率通过各个状态出现的次数进行估计。

[0075] 将操作码序列转化为一维向量后,通过相邻操作码间的转移概率来构建二维转移概率矩阵,构建好的二维转移概率矩阵转化为操作码图像,操作码图像的节点表示各个状态,边表示各个状态之间的转移概率。

[0076] 进一步地,所述S103:基于恶意软件代码的源程序文件,生成操作码图像和语言模型N-gram图像,其中,语言模型N-gram图像的生成过程包括:

[0077] S103-21:将每三个操作码作为一个N-gram子序列,将操作码序列拆分为若干个N-gram子序列;

[0078] S103-22:将每个子序列转换成一个0~255之间的整数,作为对应字节的灰度像素值;

[0079] S103-23:将所有操作码对应的灰度像素值,按照恶意样本中的二进制文件顺序排列,得到一个灰度像素序列;

[0080] S103-24:将灰度像素序列分成若干段,每段包含256个连续的灰度像素;如果最后一段灰度像素不足256个,用0填充,直到长度为256,宽W则根据恶意样本文件大小与长度的比值进行确定;

[0081] S103-25:将每一段256个灰度像素序列转换成一个 $256 \times W$ 的灰度图像;将填充后的矩阵作为灰度图像的像素值表示,使用灰度编码将矩阵中的整数值映射为图像像素值。

[0082] 进一步地,所述S104:对获取的语言模型图像进行图像缩放处理,包括:

[0083] 使用双三次插值算法,对语言模型N-gram图像进行归一化处理,将语言模型N-gram图像调整为相同长和宽。

[0084] 双三次插值算法,利用待测采样点周围的十六个点的灰度值作三次插值,既考虑

到了相邻四个点的灰度影响,又考虑到了灰度值变化率的影响。

[0085] 像素值 $f(x,y)$ 表示为:

$$[0086] \quad f(x,y) = \sum_{i=0}^3 \sum_{j=0}^3 f(x_i, y_j) W(x - x_i) W(y - y_j),$$

[0087] 其中,

$$[0088] \quad W(x) = \begin{cases} (a+2)|x|^3 - (a+3)|x|^2 + 1, & |x| \leq 1 \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a, & 1 < |x| < 2 \\ 0, & \text{其他} \end{cases}$$

[0089] 其中, $W(x)$ 用于计算所求像素点的权重,权重的值与像素点之间的距离相关。

[0090] 应理解地,将操作码图像以及使用归一化处理后的灰度图像和语言模型图像进行融合,即将上述三种灰度图像融合成多通道彩色图像。将灰度图像作为第一通道,操作码图像作为第二通道,语言模型图像作为第三通道,形成三维的彩色图像。

[0091] 进一步地,所述S105:将三通道图像,输入到训练后的卷积神经网络中,输出恶意软件代码的分类标签,其中,训练后的卷积神经网络,网络结构采用Resnet网络来实现。

[0092] 进一步地,所述训练后的卷积神经网络,网络结构包括:

[0093] 首先,将预训练的模型参数全部载入到ResNet-34卷积神经网络中;

[0094] 然后,训练ResNet-34卷积神经网络中所有层的网络参数。

[0095] 进一步地,所述训练后的卷积神经网络,训练过程包括:

[0096] 构建训练集,所述训练集为已知恶意软件代码分类标签的恶意软件代码集;

[0097] 将训练集输入到卷积神经网络中,对其进行训练,当损失函数值不再下降时,停止训练,得到训练后的卷积神经网络。

[0098] 在训练阶段,使用交叉熵损失函数对卷积神经网络模型进行训练,使用ADAM优化器的学习模型参数。

[0099] 交叉熵损失函数Cross Entropy Loss:

$$[0100] \quad \text{Cross Entropy Loss} = -\sum_{i=1}^N p(x_i) \log q(x_i) + (1 - p(x_i)) \log (1 - q(x_i));$$

[0101] 其中, N 为类别数, $p(x_i)$ 表示样本的真实分布, $q(x_i)$ 表示模型所预测的分布。

[0102] 进一步地,如图2所示,所述训练后的卷积神经网络,工作过程包括:

[0103] 输入图像到一个具有 7×7 卷积核大小的卷积层。然后,通过四个卷积组,四个卷积组包括16个残差网络结构(3个虚线残差网络结构和13个实线残差网络结构)。第一个卷积组的下采样是由一个 3×3 卷积核大小的最大池化层实现的,其余卷积组的下采样,都是由邻接上一个卷积组的残差块实现。最后,通过平均池化下采样层和全连接层进行输出,使用softmax函数将的输出转化成概率分布。实线残差网络结构,如图3所示,虚线残差网络结构,如图4所示。

[0104] 与ResNet34卷积神经网络模型相比,将虚线残差网络结构的第一层和第二层卷积层的步距分别设置为1和2,同时使用GELU激活函数,同时与使用在ImageNet上训练好的模型参数进行迁移训练。因此,能够加速模型的训练速度,增强模型的泛化能力。同时,分类速度明显提升,准确率的效果会更好。如图2所示。

[0105] 在进行卷积神经网络模型的训练过程中,使用交叉熵损失函数作为评价标准来计

算预测值与实际值之间的差异,并采用ADAM优化器来调整模型中的参数,以使其在训练过程中逐渐趋近于最优状态。在本实验中,根据不同的恶意软件样本,我们设定了不同的批尺寸、训练周期和学习率,其中批尺寸设置为128,训练周期设置为50,学习率设置为 $1e^{-4}$ 。

[0106] 本发明网络设计基于ResNet网络,总共四个卷积组,包括16个残差网络结构(3个虚线残差网络结构和13个实线残差网络结构)。分类通过平均池化下采样层和全连接层进行输出,使用softmax函数将我们的输出转化成概率分布。

[0107] 实施例二

[0108] 本实施例提供了基于多通道图像和神经网络的恶意软件分类系统;

[0109] 基于多通道图像和神经网络的恶意软件分类系统,包括:

[0110] 获取模块,其被配置为:获取待分类的恶意软件代码,对获取的恶意软件代码进行反汇编生成二进制文件;

[0111] 第一生成模块,其被配置为:基于恶意软件代码的二进制文件,生成灰度图像;

[0112] 第二生成模块,其被配置为:基于恶意软件代码的源程序文件,生成操作码图像和语言模型N-gram图像;

[0113] 第三生成模块,其被配置为:对获取的语言模型图像进行图像缩放处理,将灰度图像作为第一通道、操作码图像作为第二通道、语言模型N-gram图像作为第三通道,生成一幅三通道图像;

[0114] 输出模块,其被配置为:将三通道图像,输入到训练后的卷积神经网络中,输出恶意软件代码的分类标签。

[0115] 以上所述仅为本发明的优选实施例而已,并不用于限制本发明,对于本领域的技术人员来说,本发明可以有各种更改和变化。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

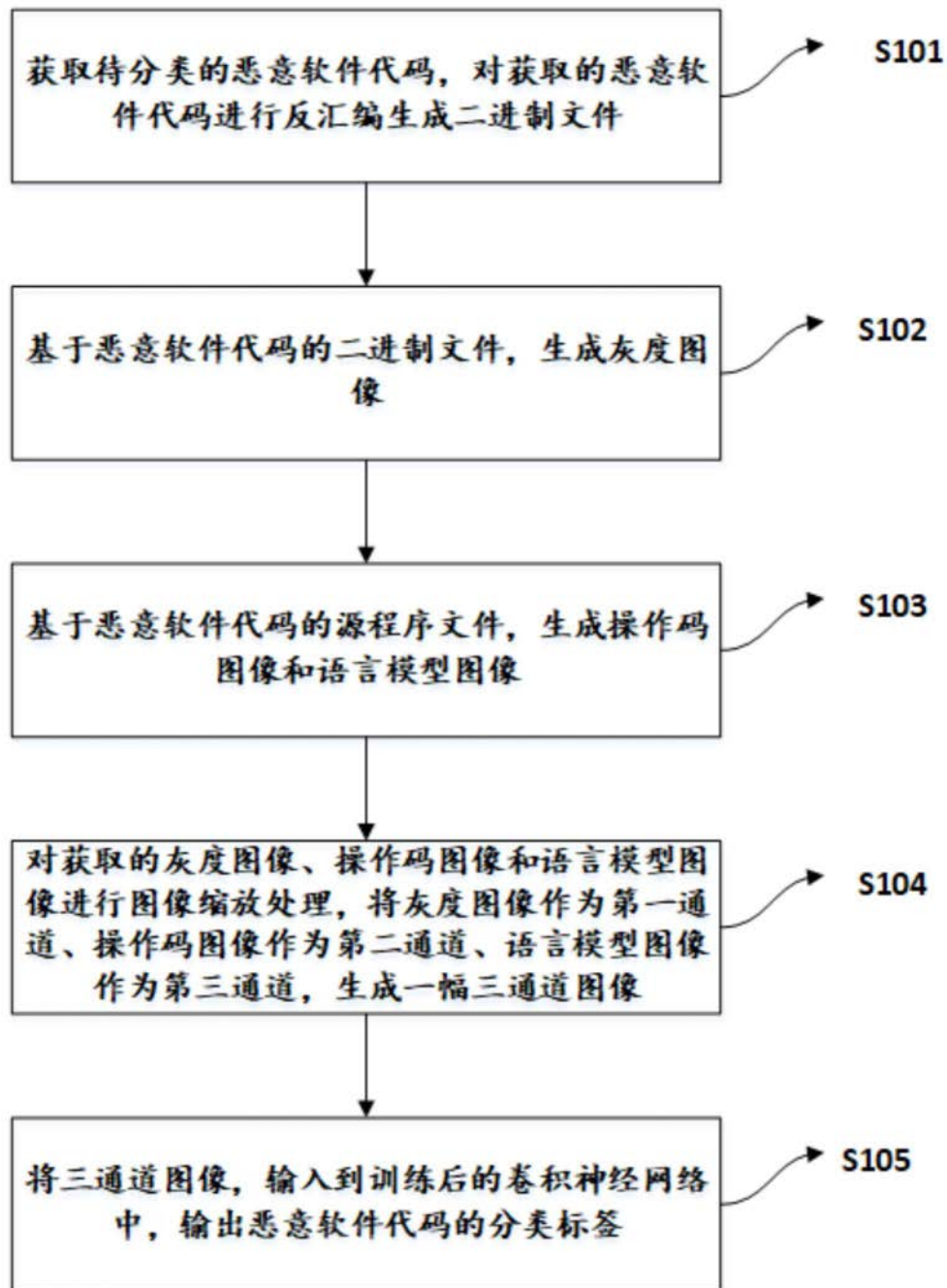


图1

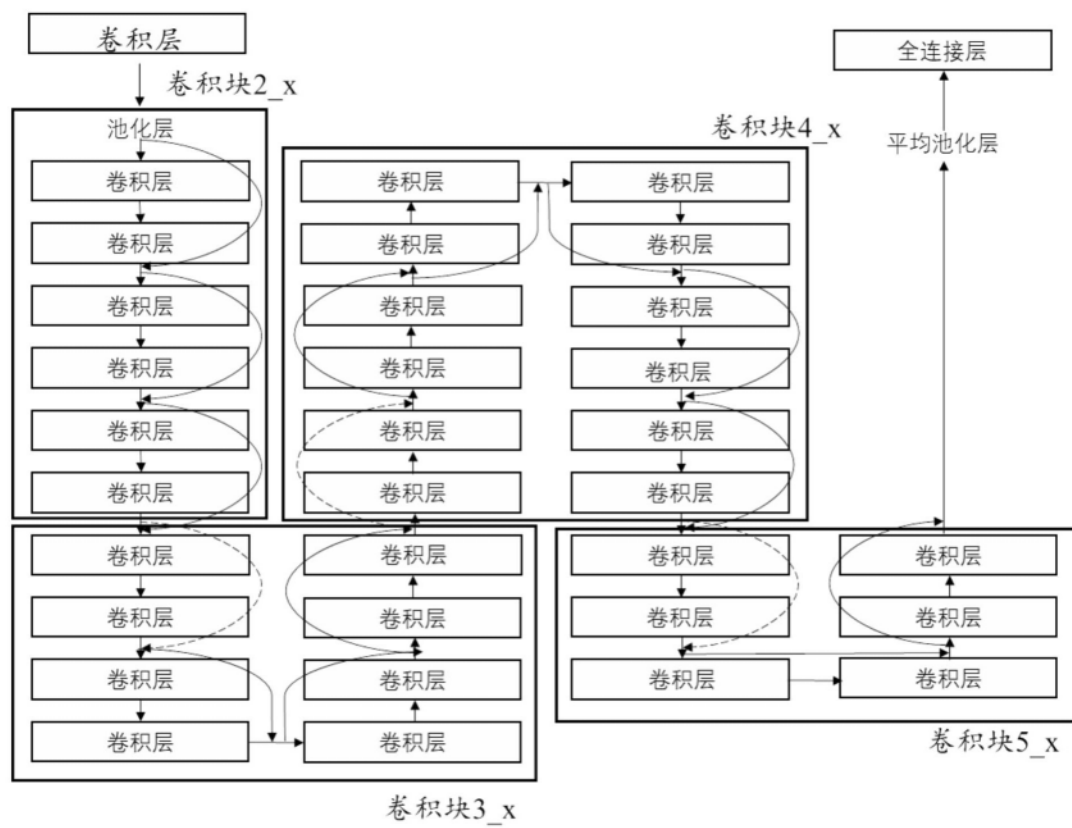


图2

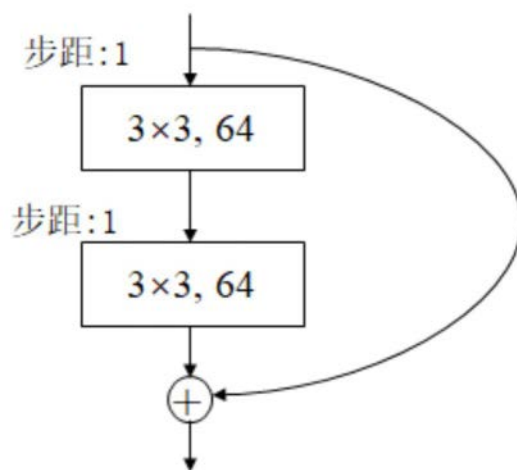


图3

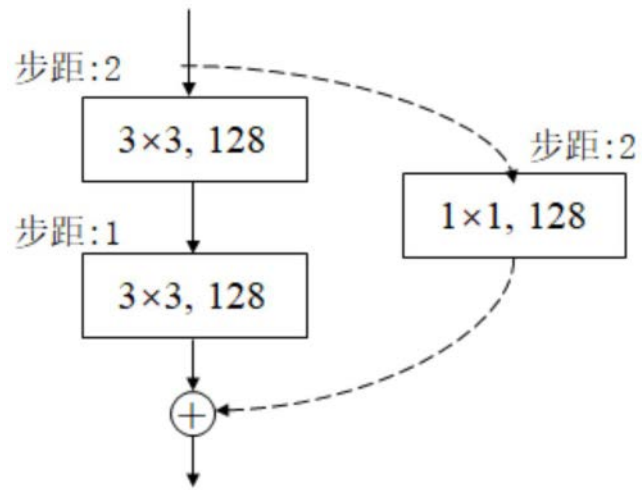


图4