Anomaly Detection Method for Power System Containers Based on Graph Convolutional Neural Network*

WANG Ziying[1*], SUN Yunxiao[1], GUO Feng[2], ZHOU Wei[2], GU Zhimin[1], JIANG Haitao[1]
(1.State Grid Jiangsu Electric Power Research Institute Co., Ltd., Nanjing Jiangsu 211103, China;
2.Wuxi Power Supply Branch of State Grid Jiangsu Electric Power Co., Ltd., Wuxi Jiangsu 214062, China)

Abstract: With the development of cloud computing technology, container technology is widely used in the field of power Internet of Things. The normal operation of containers is crucial for maintaining the stability of the power system, therefore it is necessary to detect anomalies in power containers. However, the existing methods for detecting anomalies in power containers lack consideration of power characteristics, resulting in unsatisfactory detection accuracy and low detection efficiency. To address these issues, a graph convolutional neural network method is proposed for detecting anomalies in power system containers. Firstly, the whitelist generated by container images with high security in the power system is used to filter the system call sequences obtained through dynamic tracking, in order to improve the efficiency of anomaly detection. Then, a graph convolutional neural network model integrating power risk features is constructed to improve detection accuracy. The experimental results show that the method proposed can effectively improve the testing accuracy and enhance the detection efficiency.

Key words: power Internet of Things; system call; anomaly detection; container; graph convolutional neural network
EEACC: 7220; 8140 **doi**: 10.3969/j.issn.1005-9490.2026.04.022

基于图卷积神经网络的电力应用容器异常检测方法*

王梓莹[1*], 孙云晓[1], 过 烽[2], 周 玮[2], 顾智敏[1], 姜海涛[1]
(1.国网江苏省电力有限公司电力科学研究院, 江苏 南京 211103;
2.国网江苏省电力有限公司无锡供电分公司, 江苏 无锡 214062)

摘 要: 随着云计算技术的发展, 容器技术被广泛运用于电力物联网领域。容器的正常运行对于维持电力系统的稳定至关重要, 因此需要对电力容器的异常进行检测。然而, 现有电力容器异常检测检测方法缺乏对电力特征的考量, 导致检测准确度不理想, 并且存在检测效率低下问题。为了解决这些问题, 提出一种基于图卷积神经网络的电力应用容器检测方法。首先, 利用电力系统中安全性较高的容器镜像生成的白名单对动态跟踪获取的系统调用序列进行过滤, 以提高异常检测效率; 然后, 构造了融合电力风险特征的图卷积神经网络模型, 以提升检测准确率。实验结果表明, 所提方法能有效提升测试准确率, 并提高检测效率。

关键词: 电力物联网; 系统调用; 异常检测; 容器; 图卷积神经网络
中图分类号: TM769; TP183 **文献标识码**: A **文章编号**: 1005-9490(2026)04-0856-05

物联网技术和智能设备的迅猛发展使得电力物联网数据安全方面的问题日益凸显[1-2]。随着电力物联网的发展, 容器技术[3]也逐渐得到广泛应用。相比传统虚拟机技术, 容器只依靠宿主机内核的虚拟化[4]便能进行部署, 具有快速启动、低资源消耗以及更好兼容性等优势。此外, 容器启动或运行时会触发大量的系统调用[5], 这些系统调用的行为关系到容器甚至整个系统的安全, 因此, 对于电力场景下的容器[6]异常检测方面的研究有待展开。

现有容器异常检测方法主要分为基于规则学习的方法和基于机器学习的方法两种[7]。传统基于规则学习的方法, 如文献[8]设计一个轻量级 Linux 内核沙箱模型, 并利用 exec 系统调用将不可信程序装入到沙箱中运行的设定规则, 实现保护特定的文件不被访问和阻止程序使用网络资源两大功能; 文献[9]提出了一种基于无监督系统调用过滤规则生成的容器云异常检测系统, 通过无监督的容器系统调用规则来解决已有自动化 Linux 安全模块(Linux Security Module, LSM)配置生成方法中过宽松或过严格的问题, 然而该类方法较为依赖规则本身, 缺乏对数据特征的自主分析学习。基于机器学习的方法, 如文献[10]将变长模式和马尔可夫链相结合, 通过基于马尔科夫链模型的主机异常检测模型, 以状态序列出现的概率来检测异常, 然而这类方法只考虑了通用的序列特征, 却未关联电力风险特征, 对电力物联网场景不具备针对性, 所以导致检测准确度较低。此外, 单一

项目来源: 国网江苏省电力有限公司科技项目(J2023177)
收稿日期: 2024-07-19 **修改日期**: 2024-11-11

容器正常运行时,每秒可能产生上百万次系统调用[11],这样庞大的数据量会对检测模型造成极大负荷和压力,从而使检测效率大大降低。针对上述问题,本文提出一种基于图卷积神经网络的电力系统容器异常检测方法。所提方法的创新点如下:

①针对电力容器检测中处理系统调用序列数据量较大的问题,所提方法利用电力系统中安全性较高的容器镜像生成的系统调用白名单对动态跟踪获取到的系统调用序列进行过滤,以提高异常检测效率;

②针对电力场景中异常检测准确度不理想的问题,所提方法融合电力风险特征,构造图卷积神经网络模型,以提升检测准确率。

1　所提方法

本文重点解决由系统调用序列数据量过大导致的检测效率低下问题,以及因缺乏对电力物联网风险特征考量而造成的检测准确率不佳问题。为此,本文提出了图 1 所示的基于图卷积神经网络的电力系统容器异常检测方法。该方法分为基于生成白名单的系统调用过滤和基于图卷积神经网络的异常模型构建两个步骤。在基于生成白名单的系统调用过滤模块中,本文方法通过对目标容器应用动态跟踪获取相关系统调用序列数据,然后对电力系统中常用较为安全的容器镜像进行静态分析,并生成系统调用白名单[12],最后依照白名单对动态跟踪所获数据进行过滤;在基于图卷积神经网络的异常模型构建的模块中,本文方法对过滤后的系统调用序列进行邻接矩阵和特征矩阵构建,接着将矩阵输入设计好的图卷积神经网络模型中,最后输出分类结果,实现对电力容器的异常检测。

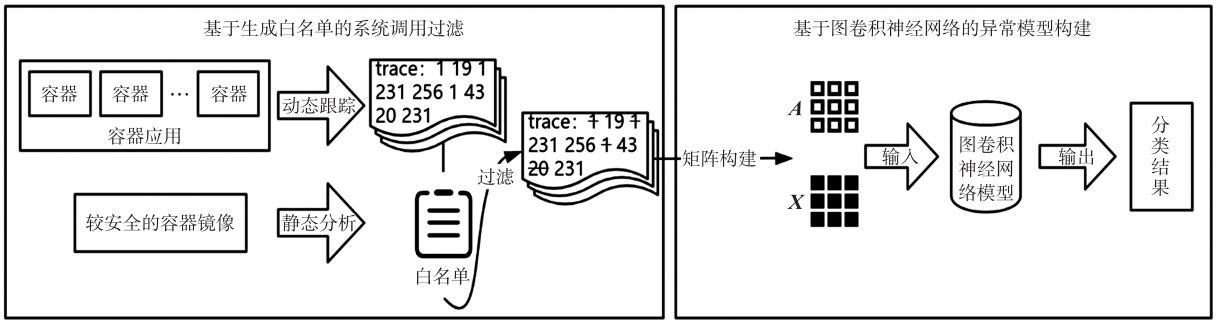

图 1　方法流程示意图

1.1　基于生成白名单的系统调用过滤

1.1.1　容器动态跟踪

为了实现动态跟踪,所提方法首先使用“cat/usr/include/asm/unistd_64.h”命令查看“unistd_64.h”文件中的系统调用表,然后,通过使用 strace 工具,监视容器应用内多个容器运行过程中产生的系统调用数据,从而得到一系列系统调用名称。最后,根据系统调用表中记录的系统调用号和系统调用名称的对应关系,将捕获到的系统调用名称处理成系统调用序列号,处理结果如表 1 所示。此处采集到的数据较为冗杂,在处理成系统调用序列号时,只需查看每行开始处的系统调用函数名称即可。

表 1　动态跟踪系统调用序列

系统调用数据	系统调用序列号
execve("/usr/sbin/nginx",["nginx"],0x7fff253c3440/* 10 vars */)=0	59
brk(NULL)=0x55924292a000	12
openat(AT_FDCWD,"/etc/ld.so.cache",O_RDONLY|O_CLOEXEC)=4	257
fstat(4,{st_mode=S_IFREG|0644,st_size=13830,…})=0	5
mmap(NULL,13830,PROT_READ,MAP_PRIVATE,4,0)=0x7f4f93a7b000	9
close(4)=0	3

1.1.2　系统调用白名单生成及过滤

所提方法通过选取一种电力场景中常用的安全性较高的容器镜像进行静态分析,最终获得系统调用白名单,并利用白名单对系统调用序列进行过滤。首先在启动的容器中提取目标二进制文件;接着,使用 IDA Pro 工具[13]对其进行反汇编并获取文件中包含的所有函数;然后,获取函数与系统调用间的映射关系;最后,通过映射关系得到对应的系统调用集合;最终生成白名单。

生成白名单后,进行系统调用过滤处理,具体规

则为:若单一容器内获取到的系统调用全在白名单上,则认为该序列风险较低,不予分析;若单一容器内有一个或多个系统调用不在白名单上,则保留该序列,以备之后的深度学习模型构建。

1.2 基于图卷积神经网络的异常检测模型构建

1.2.1 节点、邻接矩阵构建

本文将经由白名单过滤后的系统调用序列转化为系统调用轨迹图(图 2)。

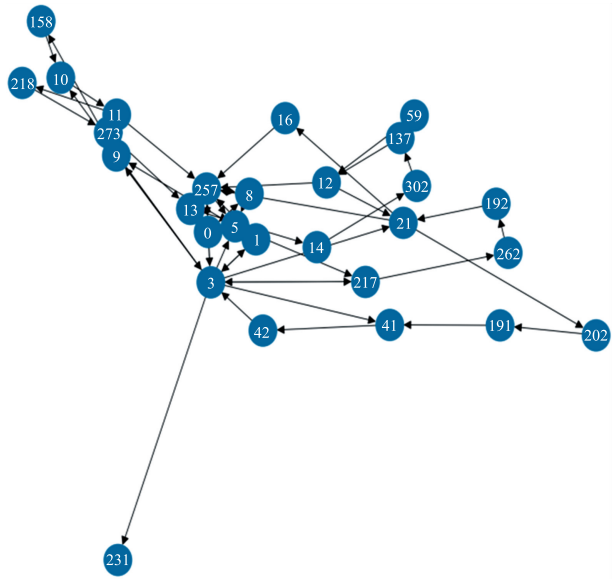

图 2 系统调用轨迹图

并将其抽象表示为:

$$G=(V,E) \tag{1}$$

式中:$V=\{v_1,v_2,\cdots,v_i,\cdots,v_j,\cdots,v_n\}$ 表示系统调用轨迹图的 n 个节点,一个系统调用函数为一个节点值,$E=\{e_{11},e_{12},\cdots,e_{1n},e_{21},\cdots,e_{2n},\cdots,e_{ij},\cdots,e_{nn}\}$ 则表示不同系统调用之间的连接。接着选取图卷积神经网络对得到的系统调用序列做进一步处理。图卷积神经网络以 (A,X) 为输入,其中 A 为邻接矩阵,X 为特征矩阵。邻接矩阵 A 是一个表示系统调用间关系的 n 阶方阵,一般写成 $\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & a_{ij} & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$ 形式,a_{ij} 为矩阵中的一个任意元素,若 i 系统调用与 j 系统调用之间存在 e_{ij} 连接,则 a_{ij} 为 1;若不存在,则 a_{ij} 为 0,即:

$$\begin{cases} a_{ij}=1 & e_{ij}\in E \\ a_{ij}=0 & e_{ij}\notin E \end{cases} \tag{2}$$

由于构建邻接矩阵 A 时忽略了系统调用自身的特征,本文在更新自身系统调用时为邻接矩阵添加了自回环,表示为:

$$\tilde{A}=A+I \tag{3}$$

1.2.2 特征矩阵构建

$X=[X_1,X_2,\cdots,X_i,\cdots,X_n]^{\mathrm{T}}$ 表示由 n 个 d 维行向量构成的特征矩阵,由每个系统调用的特征描述组成,n 代表上文中总的调用个数,d 则代表每个系统调用的特征数。本文中将第 i 个系统调用函数的特征向量 X_i 定义为:

$$X_i=[a_i,b_i,c_i,m_i] \tag{4}$$

式中:a_i,b_i,c_i 分别对应第 i 个系统调用的三个中心性特征,分别代表 Betweenness(中介中心性)、Degree(度中心性)以及 Closeness(紧密中心性)[14]。m_i 则对应第 i 个系统调用的语义特征,该特征通过改进词频-逆文本频率算法[15](Term Frequency-Inverse Document Frequency,TF-IDF)进行定义:

$$m_i=S_i\times T_i \tag{5}$$

式中:S_i 表示第 i 个系统调用的风险赋值分数,参照文献[6]中的风险等级划分,所提方法将风险等级“1”表示低风险,风险等级“2”表示中风险,风险等级“3”表示高风险,再结合电力系统对各个系统调用的使用情况,对风险特性进行调整,具体划分结果如表 2 所示;T_i 表示第 i 个系统调用的 TF-IDF 值。

表 2 电力系统调用风险等级分类

风险等级	系统调用序列号
1	90,92,91,268,93,94,260,82,264,316,132,235,280,261,2,257,304,1,18,20,296,76,77,285,277,162,306,74,75,26,87,263,59,134,62,200,234,173,172,101,105,106,109,116,112,113,114,119,117,122,123,129,297,43,288,49,42,50,41,53,28,249,248,250,175,313,176,165,166,164,227,159,305,156,219,169,246,320,103,45,47,229,9,11,0,3
2	32,33,292,73,16,319,133,259,83,258,84,85,88,266,86,265,179,251,80,81,161,72,8,7,271,213,291,233,232,281,23,270,17,19,295,89,267,40,95,188,189,190,197,198,199,206,207,209,221,323,284,290,282,289,298,303,253,294,254,255,300,301,57,58,56,12,60,231,317,155,142,144,24,141,160,302,153,31,30,67,64,65,220,66,202,158,205,154,314,203,218,310,311,308,273,256,279,126,135,157,61,247,35,230,299,44,46,307,54,48,321,25,10,238,237,149,325,151,150,152,324,38,222,226,223,283,286,171,170,167,168
3	21,269,262,5,138,79,78,217,6,187,4,137,139,136,191,192,193,194,195,196,208,210,212,252,163,125,108,107,104,115,121,111,39,186,110,140,120,118,124,102,97,98,146,147,143,145,148,312,204,211,309,315,274,52,51,55,239,27,36,224,225,287,318,96,201,100,99,228,63

1.2.3　图卷积神经网络构建

将系统调用轨迹图邻接矩阵 A、特征矩阵 X 输入图卷积神经网络,该网络由四层图卷积层(Graph Conv)、一层平均池化层(Pooling)和一层全连接层(Linear)组成。四层图卷积层由 ReLU 激活函数连接,随后将每张图的所有输出特征传入平均池化层求平均值,再将平均特征传入全连接层,最后将全连接层输出的整体特征传到 Softmax 函数后进行分类概率预测,构建过程如图 3 所示。

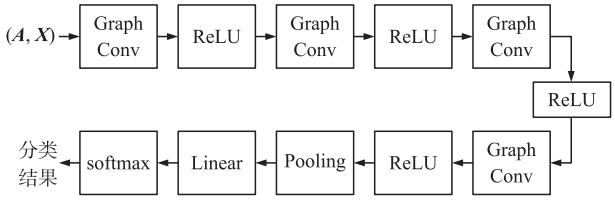

图 3　网络结构图

2　实验结果与分析

2.1　数据集

本文选用 ADFA-LD[16] 作为数据集,该数据集于 2013 年由澳大利亚国防学院发布,分成训练数据(Training)、验证数据(Validation)和攻击数据(Attack)3 部分,样本数分别为 746、4 372、833。参照文献[16],本文将 Training 与 Validation 标签定为 0,Attack 标签定为 1。考虑到样本均衡,实验只选取攻击数据和正常训练数据两部分,将总量为 1 579 的样本按 8∶2 划分训练集和测试集。

2.2　参数设置

本次实验在 AMD Ryzen 9 7945HX 硬件设备下,64 位 CentOS 7、Python 3.8、CUDA 12.1 以及 PyTorch 2.1.0 深度学习框架环境中进行。训练选用 Adam 优化器,将交叉熵损失函数作为目标函数[17],将 Batch size 设置为 8,Epoch 设置为 300,并将前 100 轮的学习率设置为 0.001,后 200 轮学习率设置为 0.000 2。

2.3　训练与测试

训练过程如图 4 所示,前 20 次 Epoch 内损失函数 Loss 快速下降,到 100 次 Epoch 时逐渐收敛。

为了验证所提方法的有效性,本文设置了四组对照实验:①对照组 1,仅采用原始数据样本,用 Betweenness、Degree、Closeness 作为特征矩阵进行训练;②对照组 2,仅采用过滤后的数据,以 Betweenness、Degree、Closeness 三个维度的节点特征进行训练;③对照组 3,将融合电力风险特征加入到节点特征中进行训练;④对照组 4,将通过白名单过滤的数据放入融合电力风险特征的模型中进行训练,效果如表 3 所示。

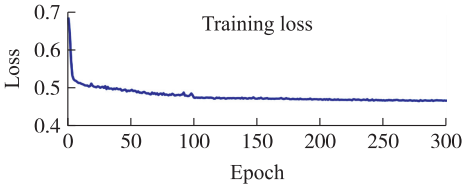

图 4　损失函数图

表 3　消融实验

	平均测试时间/s	测试准确率/%
对照组 1	5.316 7	80.70
对照组 2	4.890 8	82.03
对照组 3	5.267 0	87.34
对照组 4	4.859 0	86.27

将过滤并融合了电力风险特征的训练结果与未经处理的训练结果做对比,可以发现每轮的平均测试时间减少了 0.46 s,有效地提高了检测效率;测试准确率提升了 1.33%至 6.64%。此外,相比未处理的结果,仅过滤处理后的训练结果在测试准确率上有所提升。

3　结束语

本文提出一种基于图卷积神经网络的电力系统容器异常检测方法。所提方法利用电力系统常用安全性较高的容器镜像生成系统调用白名单,并对动态跟踪获取到的系统调用序列进行过滤,从而提高异常检测效率;融合电力风险特征,构造深度学习模型,从而提升检测准确率。最终通过消融实验获得高达 87.34%的测试准确率,并有效提高检测效率。

由于本方法使用的样本数目有限,在未来研究中,希望将所提方法用于更大数据量的样本检测。此外,在后续工作中,可以将 N-Gram 模型及 Word2vec 模型[18-21]等融入特征工程,进行更深入的语义分析,以进一步提升检测效果。

参考文献:

[1]　张娟,李俊午. 全时空量测环境下基于双向长短期记忆网络的电力物联网损耗计算[J]. 电子器件,2022,45(2):408-414.

[2]　周玮. 物联网技术在智能输配电系统中的应用[J]. 电子技术,2023,52(11):31-33.

[3]　吴建国,徐铭,廖清阳,等. 面向能源互联网的智能化运维管控平台研究[J]. 电子器件,2022,45(3):744-752.

[4]　陈轶阳,王小宁,卢莎莎,等. 面向高性能计算系统的容器技术综述[J]. 计算机科学,2023,50(2):353-363.

[5] 樊涛,武亚光,江帆. 基于关联规则的电力系统调用规则全链路生成方法[J]. 信息技术,2023(9):132-137,142.
[6] 刘晨. 基于内核函数监控的 Linux 系统防护方法的研究与实现[D]. 北京:北京交通大学,2018.
[7] 樊燚,郭义伟,胡涛,等. 基于系统调用的主机异常检测研究综述[J]. 信息工程大学学报,2024,25(1):100-109.
[8] 程香鹏,陈莉君. 基于 LSM 的沙箱模块设计与实现[J]. 计算机与数字工程,2014,42(8):1521-1525.
[9] 吴圣麟,刘汪根,严明,等. 基于无监督系统调用规则生成的容器云实时异常检测系统[J]. 信息网络安全,2023,23(12):91-102.
[10] 刘永庆,刘东生. 基于马尔科夫链的主机异常检测方法研究[J]. 计算机与数字工程,2010,38(7):20-23.
[11] 栗晓晗,张新有. 基于系统调用的 Docker 容器异常检测[J]. 微电子学与计算机,2022,39(12):77-85.
[12] 邢云龙,严飞,刘彦孝,等. 基于系统调用限制的容器安全防护方案[J]. 武汉大学学报(理学版),2022,68(1):35-43.
[13] 罗治祥,向栖,李乐言. 基于污点分析的二进制程序漏洞检测系统设计与实现[J]. 网络安全与数据治理,2023,42(11):1-7.
[14] 刘炜,宋友,卓佩妍,等. 融合多源图特征的 Kcore-GCN 反欺诈算法研究[J]. 计算机科学,2024,51(S1):905-911.
[15] 张阳,范俊杰,孙晓山,等. 基于系统调用序列学习的内核模糊测试[J]. 计算机系统应用,2023,32(9):19-31.
[16] 陈仲磊. 基于系统调用数据的容器入侵检测技术研究[D]. 郑州:战略支援部队信息工程大学,2023.
[17] 雷靖玮,伊鹏,陈祥. 基于图神经网络与深度学习的 PDF 文档检测模型[J]. 计算机工程与设计,2024,45(2):356-366.
[18] 李橙,罗森林. 基于系统调用行为相似性聚类的主机入侵检测方法研究[J]. 信息安全研究,2021,7(9):828-835.
[19] 陈仲磊,伊鹏,陈祥,等. 融合注意力与卷积的系统调用异常检测[J]. 信息工程大学学报,2023,24(4):475-483.
[20] 贺娇君,蔡满春,芦天亮. 基于 GCN 和 BiLSTM 的 Android 恶意软件检测方法[J]. 计算机科学,2024,51(4):388-395.
[21] 曹炳豪,汪智超,朱二周. 面向软件漏洞检测的改进代码属性图的图神经网络[J]. 微电子学与计算机,2024,41(1):74-82.

王梓莹(1991—),女,工程师,硕士,研究方向:电力网络安全;

孙云晓(1992—),女,工程师,硕士,研究方向:电力信息通信及网络安全;

过 烽(1981—),男,高级工程师,学士,研究方向:电力信息通信及网络安全;

周 玮(1987—),男,高级工程师,硕士,研究方向:电力通信网络运维及安全;

顾智敏(1993—),女,工程师,硕士,研究方向:电力网络安全;

姜海涛(1985—),男,高级工程师,博士,研究方向:电力系统网络安全。