基于XGBoost的混合模式门级硬件木马检测方法

张 颖* 李 森 陈 鑫 姚嘉祺 毛志明
(南京航空航天大学电子信息工程学院 南京 211106)

摘 要：针对恶意的第三方厂商在电路设计阶段中植入硬件木马的问题，该文提出一种基于XGBoost的混合模式门级硬件木马检测方法。该检测方法将电路的每个线网类型作为节点，采用混合模式3层级的检测方式。首先，基于提取的电路静态特征，利用XGBoost算法实现第1层级的检测。继而，通过分析扫描链的结构特征，对第1层级分离得到的正常电路继续进行第2层级的面向扫描链中存在木马电路的静态检测。最后，在第3层级采用动态检测方法进一步提升检测的准确性。Trust-Hub基准测试集的实测结果表明，该方法与现有的其他检测方法相比具有较优的木马检测率，可达到94.0%的平均真阳率(TPR)和99.3%的平均真阴率(TNR)。

关键词：硬件木马检测；XGBoost算法；门级网表；静态检测；动态检测
中图分类号：TP309.5; TN47 **文献标识码**：A **文章编号**：1009-5896(2021)10-3050-08
DOI: 10.11999/JEIT200874

Hybrid Multi-level Hardware Trojan Detection Method for Gate-level Netlists Based on XGBoost

ZHANG Ying LI Shen CHEN Xin YAO Jiaqi MAO Zhiming
(*College of Electronics and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing* 211106, *China*)

Abstract: A hybrid multi-level hardware Trojan detection method based on XGBoost algorithm is proposed for the problem of hardware Trojans implanted by malicious third-party manufacturers. The detection method treats each wire in gate-level netlist as a node and detects Trojans in three levels. Firstly, the effective static features of the circuit are extracted and the XGBoost algorithm is applied to detect the suspicious Trojan circuits. Common circuits distinguished at the first level continued to be detected at the second level by analyzing scan chain structural features. Finally, dynamic detection is used to increase further the accuracy of Trojans detection. Experimental results on Trust-hub benchmark show that this method has a higher accuracy compared with other existing detection methods. This detection method can finally achieve 94.0% average True Positive Rate (TPR) and 99.3% average True Negative Rate (TNR).

Key words: Hardware Trojan detection; XGBoost; Gate-level netlists; Static detection; Dynamic detection

1 引言

随着集成电路芯片全球化产业链模式的发展，硬件安全问题日益成为继软件安全问题后存在的新隐患。硬件木马电路可能会在芯片设计阶段和制造阶段被插入到芯片中[1,2]。目前硬件木马电路的检测技术主要分为动态检测和静态检测两类。动态检测方法指在对待测电路施加外部激励的情况下，观察模拟电路或实际电路的行为从而检测是否存在木马电路，旁路分析法是其主流方式之一[3,4]。文献[5]在系统层次对传统的旁路分析检测方法进行改进，提出一种基于支持向量机 (Support Vector Machine, SVM)算法检测的方案。考虑到木马电路的触发结构通常较为隐蔽且不易触发，动态检测需要建立特殊的测试激励来提高木马电路的触发概率[6]。文献[7]提出了一种基于变异分析的统计测试生成方法，以激活电路中存在的低活跃性硬件木马。文献[8]通过获取信号的可测性和可观性，使用聚类的机器学习算法进行硬件木马检测。文献[9]首次提出了一种利用路径延迟顺序的检测方法。

静态检测方法不需要对电路进行仿真测试，利

收稿日期：2020-10-12；改回日期：2021-07-20；网络出版：2021-07-30
*通信作者： 张颖 tracy403@nuaa.edu.cn
基金项目：国家自然科学基金(61701228, 61106029)，模拟集成电路重点实验室基金(61428020304)，航空科学基金(20180852005)
Foundation Items: The National Natural Science Foundation of China (61701228, 61106029), The Science and Technology on Analog Integrated Circuit Laboratory (61428020304), The Aeronautical Science Foundation of China (20180852005)

用木马电路与普通电路的差异分析对木马电路进行检测。文献[10]通过静态布尔函数分析得到具有弱影响力的输入，将其标记为可疑信号。但该方法不易检测时序电路木马，并且不适用于大规模集成电路。伴随着新型木马电路的不断涌现，近年来，机器学习算法成为解决硬件木马静态检测的有效方法之一[11,12]。文献[13]提出一种基于随机森林的木马检测方法。首先提出51个硬件木马特征，最终选择其中11个特征作为最终特征用于机器学习训练和分类。文献[14]是在文献[13]的基础上，针对文献[13]中不足提出的一种基于边界检测的优化方法。

动态检测在顺序触发木马和不可观测性木马方面相对静态检测会更有优势，而静态检测在组合触发、隐式触发、可观测性木马上有较大优势。因此，本文提出一种基于XGBoost算法的动态和静态检测相结合的混合模式门级硬件木马检测方法。

本文的贡献主要体现在以下方面：

(1)提出一种基于XGBoost算法的混合模式门级硬件木马检测方法。这是首次提出并使用静态检测和动态检测相结合的多层次检测方法对硬件木马进行检测，为硬件木马检测提供新方向。

(2)提出了两种新的静态木马电路特性：3级触发网络和环形振荡器结构，并作为特征用于硬件木马电路静态检测的机器学习算法中。实验结果证明，这两种特征对木马检测具有积极效应。

(3)针对扫描链中存在木马电路的风险，提出基于扫描链的木马电路特征和检测方法，进一步优化了静态硬件木马检测方法。

(4)对木马电路的基准测试集Trust-Hub[15]和DeTrust[16]木马电路实施检测，所得的实验结果表明，与现有的其他检测方法相比，该方法在大多数情况下具有更准确的硬件木马电路检测效果，对Trust-Hub测试集可以达到94.0%的平均真阳率(True Positive Rate, TPR)和99.3%的平均真阴率(True Negative Rate, TNR)的检测精度。对自实现的DeTrust木马电路可达到平均82.1%TPR和92.2%TNR的检测精度。

2　门级硬件木马特征

本节通过分析木马电路和普通电路存在的差异，提出两种全新的静态木马电路特征并与传统木马电路特征相结合，用于实现静态硬件木马检测。检测方法以电路的每个线网类型作为节点来提取一系列木马特征，传统的木马电路特征包括逻辑门的扇入数量、多路复用器、循环结构、触发器、原始输入输出距离等，两种新提出的木马特征为3级触发网络和环形振荡器结构。

2.1　3级触发网络(three_level_trigger)

在组合逻辑电路中，相比于普通电路，木马电路具有触发信号常为单一触发或少数触发且触发结构复杂等特点[17,18]。针对这一特点进行分析，提出一种3级触发网络特征。设net是待测电路的线网，3级触发信号(three_trigger_line)定义为连续通过3层逻辑级的输出信号，每一逻辑级为基本逻辑门中的1个或多个且该信号的第1级扇入数大于16。扇入数的阈值为16是根据对Trust-Hub木马库中同类特征电路进行统计分析而设定的。3级触发网络(three_level_trigger)定义为与3级触发信号(three_trigger_line)相关联的所有输入信号。3级触发网络特征如图1所示。

3级触发信号通过多级基本逻辑门等低触发的开关门电路，一定程度上减少了该信号在组合电路中触发的概率，契合木马电路触发概率低的特点。同时，较大的第1级扇入数契合木马触发电路具有复杂逻辑结构的特点。所以，本文将3级触发网络作为区分木马电路与普通电路的特征之一，并将在4.2节中给出其有效性的验证。

2.2　环形振荡器结构(ring_oscillator)

环形振荡器定义为由奇数个反相器首尾相连组成的环形电路。在相当一部分的硬件攻击中，环形振荡器电路会被用作木马电路的负载模块[17,18]。例如，图2所示电路通过木马触发信号控制一个29级环形振荡器木马负载，以便降低电路性能。

因此，本文首次提出一种环形振荡器的静态特征提取方案，以静态特征检测的方式对环形振荡器木马负载进行检测。设net是待测电路的一个线网类型，环形振荡器特征定义为线网类型net连续经过3级逻辑级以上的反相器结构，该线网类型net视为木马可疑信号。由于正常电路中也可能存在环形

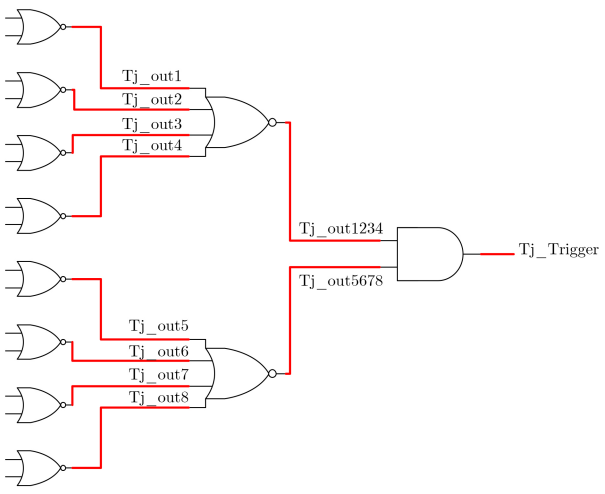


图 1 3级触发网络特征示意图

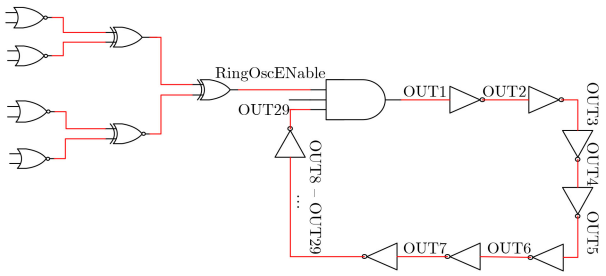


图 2 环形振荡器结构特征示意图

振荡器结构，所以环形振荡器特征还要附加一个判断条件：如果环形振荡器的原始触发信号为已经检测出的木马可疑信号且该信号通过一种开关门电路，则该环形振荡器结构中所有的信号为木马信号。环形振荡器特征的有效性将在3.2节中通过实验证明。

3 混合模式多层级硬件木马检测

基于XGboost的混合模式多层级硬件木马检测方法具体结构如图3所示。该方法将静态检测与动态检测相结合，采用多层级的结构对待测电路进行木马检测。首先通过分析门级网表的静态特征，应用XGBoost算法实现第1层级的硬件木马检测；继而对网表中的扫描链路进行分析，检测可能存在的针对扫描电路的木马攻击，完成第2层级的木马检测；最后利用动态检测和翻转率的数据分析，进行第3层级的木马检测，最终实现优化的混合多层级门级硬件木马检测。

3.1 静态特征木马检测

3.1.1 硬件木马特征提取

为了便于机器学习算法的执行，需要将待测电路的门级网表划分为信号模块和门电路结构模块。其中，输入信号、输出信号、线网信号均属于信号模块，它们也被分别放入相应的列表中。门电路结构模块将进行规范化处理，即将所有门级单元独立出来，包括门级单元的类型、名称、输入输出信号。

规范化处理后的门级单元将以文本匹配的方式对其进行分析。以线网类型net作为信号的起点，确定该信号所经过的门级单元类型以及该门级单元的输入输出信号。每个门级单元建立一个有向图节点，该节点的标签表示该门级单元的具体门结构。根据门级单元的输入输出信号的相互连接建立有向图的边，其中输入方向的最后一个节点为当前节点，输出方向的第1个节点为当前节点，这样就形成了具有输入输出特征的有向图。通过对得到的有向图的节点特性分析，可获得前文所述的各节点的木马电路特征，从而得到待检测电路的静态特征数据集。该数据集为第1层级检测中机器学习算法的处理对象。

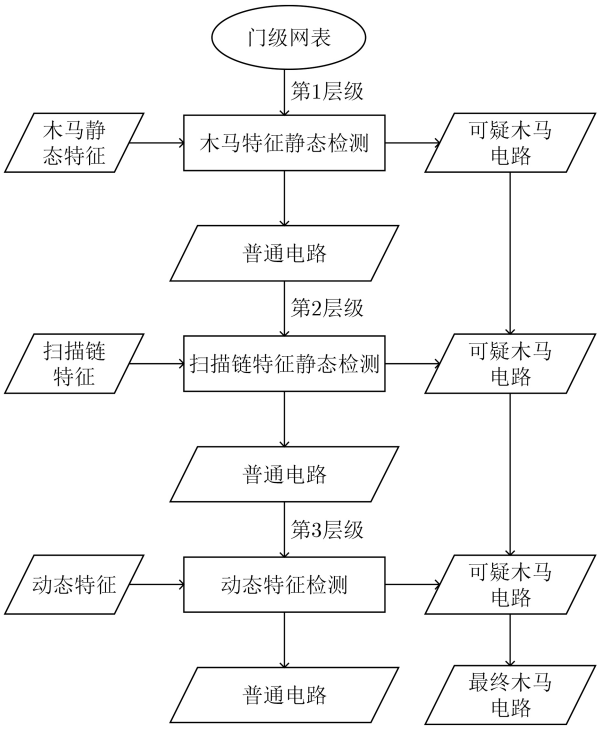


图 3 基于XGBoost的混合多层级硬件木马检测框图

3.1.2 XGBoost算法

XGBoost算法是一种梯度提升算法，由多棵回归树组成，其核心思想是将许多弱分类器集成在一起，形成一个强分类器，每一个新的分类器都是在原分类器的基础上沿着损失函数的负梯度方向生成[17]。相比于只通过1阶导数对函数进行优化的传统梯度提升树(Gradient Boosting Decision Tree, GBDT)算法，XGBoost算法则是对函数进行2阶泰勒展开，这使得XGBoost算法具有更快的模型收敛速度。此外，通过在损失函数中添加正则化项来抑制模型复杂度，XGBoost算法还可以有效地防止过拟合。而本文拟使用的实验样本数据相较于大规模的机器学习相对较小，较易出现过拟合的现象。此外，提取的门级木马特征在量纲和单位均有所不同，导致对应的特征值具有较大差异，可能造成机器学习模型难以收敛，而数值高的特征可能具备对模型的较大影响力。基于决策树的XGBoost算法是通过分析特征数据的分布以及数据特征之间的条件概率来进行叶子节点分裂的，并不是参考特征的具体数值，因而对上述现象具有较好的包容性，因而较为适合用于门级硬件木马的检测。

3.1.3 检测流程

静态特征的木马检测方法的流程是基于静态特征集和标准木马库Trust-Hub所得的训练数据集，

使用XGBoost算法的训练过程，进行特征集和算法模型的优化，得到最佳特征集和最优参数配置的训练模型，最后对测试数据集进行硬件木马检测，并进行准确率的分析。

其中，最佳特征集是通过交叉验证的方式对待测电路中提取的静态特征进行筛选而得到。基于每个特征对检测结果的重要性排名，设置阈值来筛选特征，最终通过检测结果的准确率来选择特征构建最佳特征集。

通过将最佳特征集得到的训练数据送到机器学习算法中进行模型训练。针对检测准确率方面对模型进行评估，调整训练参数，得到最优的训练模型。在参数调优过程中对模型性能影响较大的参数主要有学习率、迭代次数、最大树深度和最小样本权重。

测试集和训练集是采用留一法从Trust-Hub的14个门级木马电路得到的，即每次将14种待测电路中的一种电路作为测试电路，其余的13种电路都作为训练电路。留一法可以保证每个待测电路对于机器学习模型都是未知的电路结构，为实验的科学性提供了保证。

3.2 扫描链特征的静态检测

作为典型的可测性设计技术之一，扫描链通常会被添加到门级网表，以提升后期制造测试的效率，然而，由于扫描链提供了电路的访问通路，也极易被木马制造者所利用。例如，旁路检测方法需要向待测电路中输入测试激励，动态的分析电路实际工作情况，木马制造者则可能对门级扫描链的电路结构进行修改，从而避免木马电路在测试阶段暴露。通过分析硬件木马电路在扫描链电路存在的效应，提出两种扫描链结构中的硬件木马特征。旨在对基于机器学习的木马检测方法进行补充，从而得到更加完善的静态硬件木马检测方案。

3.2.1 扫描链使能木马信号

在添加扫描链的过程中，每个正常的时序电路单元都会被转换为扫描时序单元。例如，原始电路中的D触发器(D Flip-Flop, DFF)结构在添加过程中会被转化为扫描D触发器(Scan D Flip-Flop, SDFF)。木马制造者可以将木马电路的触发信号设置为SE使能信号经过反相器的输出信号，如图4中的使能木马信号所示。这将使得木马电路在整体电路进行功能测试时一直处于未激活状态，成功躲避功能检测。因此扫描链使能木马信号定义为：扫描链使能信号SE经过反相器的输出信号。

3.2.2 未转换可疑信号

在添加扫描链的过程中，还有一些木马电路的序列单元在木马设计者的操作下不会被转换为扫描测试单元，如图4的未转换模块所示。该电路为泄露信息型木马电路，在添加扫描链阶段，设计者故意未将木马电路中的D触发器转换为扫描D触发器，导致在进行功能测试时，木马电路并未接入扫描电路中，从而躲避检测。因此，未转换可疑信号定义为：在添加扫描链的过程中，未被转换成扫描测试单元的正常时序单元中的所有信号。

3.2.3 扫描链特征检测

通过对扫描链木马电路的特征分析，在第2层级对待测电路进行扫描链静态检测。由于木马电路在扫描链中所展现的特征是独特的，且是正常电路不可能具备的特征。所以，可以直接对扫描链中的木马电路特征进行提取，将具备特征的信号定义为木马可疑信号。扫描链特征提取使用正则匹配方式，将符合上述提出的特征信号标记为木马可疑信

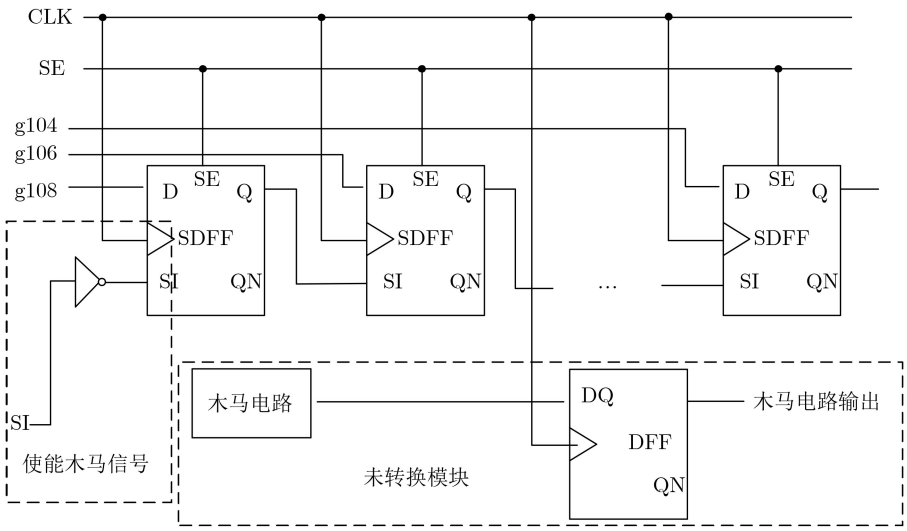


图 4 扫描链中木马电路结构特征

号。在第2层级检测中针对第1层级检测分类出的正常电路继续进行扫描链静态检测，并将电路分离为正常电路和木马电路。

3.3 动态特征检测

通过总结相关工作，静态检测对于木马电路检测存在一定的局限性。为了进一步提升检测效率，在第3层级，将经由第2层级扫描链静态检测得到的正常电路进行动态检测，利用动态特征检测和静态特征检测的相互补充，进一步提升检测方法的准确率。

动态检测的出发点在于，实际电路工作时，木马触发电路的活动率通常低于普通电路，统计分析信号活动率的差异，可以用于鉴别可疑木马信号。动态翻转率可以用于标示信号在实际工作时的活动率，假设在m个时钟周期内节点的翻转次数为n，则其动态翻转率为n/m。

选取Trust-Hub上的电路为测试基准，根据电路的功能，注入相应的测试激励，分析可得木马电路的平均触发概率为9.46^{-10}，因此，动态翻转率小于平均触发概率的信号将被标记为可疑信号。实验中，使用Synopsys EDA工具VCS，信号翻转次数可以从VCS生成的报告中提取。对经由第2层级扫描链静态检测得到的正常电路进行动态检测，可得到最终的木马电路鉴别结果。

4 实验结果及分析

4.1 检测环境与指标

目前被广泛认可的可作为测试基准的数字型木马电路为Trust-Hub木马库和DeTrust项目[19]，其中，Trust-Hub提供了门级、RTL级和板级的多种基准木马电路，DeTrust则是给出了优化隐蔽性的木马电路设计方法。我们选择Trust-Hub中的14个基准门级木马电路和5个自实现的DeTrust木马作为待测电路，使用基于XGBoost的混合模式检测方法对待测电路进行检测。特征提取框架由Python语言构建，使用XGBoost工具库[20]。静态特征提取实验在Win7服务器上进行，使用Intel E5-1607中央处理器，运行频率为3.1 GHz，内存为16 GB。实验结果可通过以下指标反映: 被标示为木马样本(负样本)的木马数目(True Negative, TN)、被标示为正确样本的木马数目(False Positive, FP)、被标示为木马的正确样本数目 (False Negative, FN)、被标示为正确的正确样本数目 (True Positive, TP), TPR, TNR。其中，TPR, TNR为最重要的检测准确率指标，分别表示正确样本被标示为正确的比率、木马样本被标示为木马的比率。而TPR, TNR指标的计算方法为

$$\mathrm{TPR}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}\times 100\% \tag{1}$$

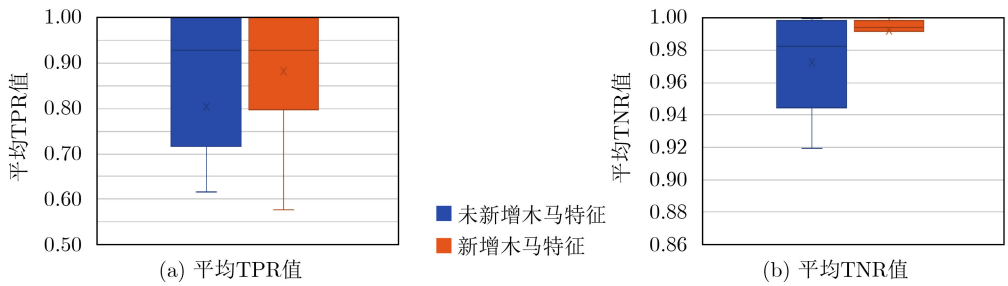
$$\mathrm{TNR}=\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}\times 100\% \tag{2}$$

4.2 特征有效性验证

为了验证第2节中木马特征的有效性，进行如下实验：采用添加新特征的特征数据集和不添加新特征的特征数据集分别对待测电路进行静态检测。考虑到测试电路的基准性，对Trust-Hub中的14个电路进行实验，根据实验结果绘制箱线图，如图5所示。从箱状图中可以看出在添加新特征后，无论TPR值方面还是TNR值方面都有所提高。从箱体结构来看，添加新特征后箱体展现出更优的数据分布，箱体长度明显缩短，对每个待测电路呈现出较小的波动，有更好的适应性。从平均值来看，TPR平均值提高了8%，TNR平均值提高了2%。这足以证明新提出的特征对特征数据集存在积极效应，有助于获得更精准的静态检测结果。

4.3 检测结果

经过对19种待测电路检测，多层级硬件木马检测方法对待测电路的检测时间平均为2.85～5 s。这表明检测方法在面对数万门以上的较大规模的集成电路时，仍然可以在较短的时间内完成待测电路的木马检测。待测电路在检测方法中各层级的检测结果如表1所示。检测结果表明，采用多层级检测平均可以提高3.9%TPR准确率，其中Trust-Hub电路中可提高5.6%TPR准确率，在s38417-T100中甚至提高了36.4%TPR准确率。随着待测电路通过每个



(a) 平均TPR值　　(b) 平均TNR值

图 5 特征有效性箱型图结果对比

表 1　各层级检测结果详细参数

层级	电路	第1层级						第2层级						第3层级					
测试电路		TN	FP	FN	TP	TPR	TNR	TN	FP	FN	TP	TPR	TNR	TN	FP	FN	TP	TPR	TNR
Trust-Hub	s38417-T100	5461	10	4	7	0.636	0.998	5461	10	4	7	0.636	0.998	5461	10	0	11	1	0.998
	s38417-T200	5462	9	0	11	1	0.998	5462	9	0	11	1	0.998	5462	9	0	11	1	0.998
	s38417-T300	5467	1	4	43	0.915	0.999	5467	1	3	44	0.936	0.999	5467	1	2	43	0.956	0.999
	s35932-T100	5867	0	0	17	1	1.000	5867	0	0	17	1	1	5867	0	0	17	1	1
	s35932-T200	5857	9	4	8	0.667	0.999	5857	9	2	10	0.833	0.999	5857	9	2	10	0.833	0.999
	s35932-T300	5862	4	2	34	0.944	0.999	5862	4	0	36	1	0.999	5862	4	0	36	1	0.999
	s15850-T100	2122	58	11	15	0.577	0.973	2122	58	9	17	0.654	0.973	2122	58	9	17	0.654	0.973
	RS232-T1000	238	2	0	37	1	0.992	238	2	0	37	1	0.992	238	2	0	37	1	0.992
	RS232-T1100	242	7	6	32	0.842	0.972	242	7	6	32	0.842	0.972	242	7	6	32	0.842	0.972
	RS232-T1200	252	1	3	30	0.909	0.996	252	1	3	30	0.909	0.996	252	1	3	30	0.909	0.996
	RS232-T1300	251	2	0	27	1	0.992	251	2	0	27	1	0.992	251	2	0	27	1	0.992
	RS232-T1400	237	2	0	44	1	0.992	237	2	0	44	1	0.992	237	2	0	44	1	0.992
	RS232-T1500	245	2	0	38	1	0.992	245	2	0	38	1	0.992	245	2	0	38	1	0.992
	RS232-T1600	250	2	3	22	0.880	0.992	250	2	3	22	0.880	0.992	250	2	1	24	0.960	0.992
	平均值	TPR:88.4% TNR:99.3%						TPR:90.6% TNR:99.3%						TPR:94.0% TNR:99.3%					
DeTrust	DT-1	575	53	4	17	0.810	0.916	575	53	4	17	0.810	0.916	575	40	4	19	0.826	0.920
	DT-2	570	49	5	18	0.783	0.921	570	49	5	18	0.783	0.921	570	47	4	18	0.818	0.924
	DT-3	552	51	4	15	0.789	0.915	552	51	4	15	0.789	0.915	552	46	4	17	0.810	0.923
	DT-4	582	55	5	19	0.792	0.914	582	55	5	19	0.792	0.914	582	52	5	21	0.808	0.918
	DT-5	563	49	3	15	0.833	0.920	563	49	3	15	0.833	0.920	563	46	3	16	0.842	0.924
	平均值	TPR:80.1% TNR:91.7%						TPR:80.1% TNR:91.7%						TPR:82.1% TNR:92.2%					
平均值		TPR:84.2% TNR:95.5%						TPR:85.3% TNR:95.5%						TPR:88.1% TNR:95.8%					

测试层级，检测方法的检测效果逐渐提高。在第1层静态检测，Trust-Hub电路的TPR平均值为88.4%，DeTrust电路的TPR平均值为80.1%。这证明第1层的静态检测可以有效地对木马电路进行筛选。在经过第2层级检测后，Trust-Hub电路TPR的平均值增加为90.6%，这表明第2层扫描链检测对检测结果进行了优化。由于88.4%的木马电路在第1层级中已经被识别，导致未被检测到的木马电路剩余数量基数稀少，且剩余木马电路检测难度更加困难。因此，TPR平均值增加2.2%对Trust-Hub电路的第2层级检测已是较优的结果。由于我们实现的DeTrust电路中并没包含扫描链电路，所以其TPR值并没变化。经过第3层动态检测后，Trust-Hub电路TPR的平均值增加为94.0%，DeTrust电路TPR的平均值增加为82.1%，DeTrust电路TNR的平均值增加为92.2%。这证明动态特征检测与静态特征检测相互补充的有效性。同时，最终检测结果相比于第2层级平均提高了2.8%，这也充分诠释对木马电路进行动态静态结合检测的必要性。

5　与现有方法结果比较

现有的门级硬件木马检测方法的相关文献的检测对象均为Trust-Hub库中的门级硬件木马电路，因此在本节，将通过与现有的其他机器学习检测方法对于Trust-Hub木马检测结果的比较，分析多层级检测方法的优势。由于文献[13]检测方法使用的测试基准与我们使用的测试基准相同，且文献[13]无论在检测效果上还是检测方法影响因数上都有出色的表现。所以，选择文献[13]作为对比较为合适。多层级检测方法与文献[13]检测方法的TPR和TNR的比较结果如图6所示。

从图6中可以看出在大多数电路中，多层级检测方法的TPR值均高于文献[13]检测方法。特别在一些电路中，如s35932-T200 (网表中具有较为明显的本文提出的特征)，多层级检测方法相对于文献[13]高出75%。在平均值方面，多层级检测方法可以达到94.0%的TPR值，明显高于文献[13]的72.8%。这表明多层级检测方法可以更准确地从待测电路中识别出木马电路。在TNR方面，从图6中可以看出两

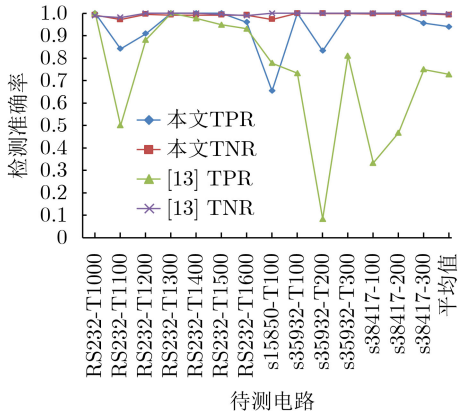


图 6 两种方法检测结果比较

种检测方法的TNR值几乎持平。由于多层级检测方法需要对木马进行精准的多层级检测，可能会导致TNR的值相较于文献[13]略有不足。硬件木马检测方法的主要目的是防止恶意的第三方厂商在电路中插入木马电路，识别出待测电路中所有的木马电路是检测方法的首要任务。所以，可将关注点主要集中在TPR的检测率，在确保得到较高的TPR检测率的基础上，同时保证获得较高的TNR检测率。与文献[13]相较于TNR减少0.4%，多层检测方法在TPR方面提高21.2%的精度。这对门级硬件木马电路的检测效果是最优的且最有效的提高。

6 结束语

基于XGBoost算法的混合模式多层级硬件木马检测方法能够对电路中的每个线网类型节点进行判别分类，并采用动态检测与静态检测相结合的方式识别待测电路中的木马电路，为硬件木马检测提供一个新方向。实验结果表明，该检测方法可以高效地识别出大部分木马电路，且识别效率显著优于现有的门级电路硬件木马检测方法。未来，为了进一步完善混合模式的硬件木马检测方法，可以与RTL级的木马电路检测相结合，搭建全面且高效的硬件木马检测平台。

参 考 文 献

[1] ELNAGGAR R, CHAKRABARTY K, and TAHOORI M B. Hardware Trojan detection using changepoint-based anomaly detection techniques[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019, 27(12): 2706–2719. doi: 10.1109/TVLSI.2019.2925807.

[2] CHEN Jinghui, DONG Chen, ZHANG Fan, *et al*. A Hardware-Trojans detection approach based on eXtreme Gradient Boosting[C]. 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), Beijing, China, 2019: 69–73.

[3] 张毅军, 张晓, 林少锋, 等. 基于功耗特征的硬件木马检测方法[J]. 电脑知识与技术, 2019, 15(31): 15–16, 26.
ZHANG Yijun, ZHANG Xiao, LIN Shaofeng, *et al*. Hardware Trojan detection method based on power consumption features[J]. *Computer Knowledge and Technology*, 2019, 15(31): 15–16, 26.

[4] SAAD W, SANJAB A, WANG Yunpeng, *et al*. Hardware Trojan detection game: A prospect-theoretic approach[J]. *IEEE Transactions on Vehicular Technology*, 2017, 66(9): 7697–7710. doi: 10.1109/TVT.2017.2686853.

[5] 佟鑫, 李莹, 陈岚. SVM算法在硬件木马旁路分析检测中的应用[J]. 电子与信息学报, 2020, 42(7): 1643–1651. doi: 10.11999/JEIT190532.
TONG Xin, LI Ying, and CHEN Lan. Application of SVM machine learning to hardware Trojan detection using side-channel analysis[J]. *Journal of Electronics & Information Technology*, 2020, 42(7): 1643–1651. doi: 10.11999/JEIT190532.

[6] 王晓晗, 王韬, 李雄伟, 等. 基于人工蜂群的硬件木马测试向量生成方法[J]. 上海交通大学学报, 2019, 53(10): 1218–1224.
WANG Xiaohan, WANG Tao, LI Xiongwei, *et al*. Test pattern generation method for hardware Trojan detection based on artificial bee colony[J]. *Journal of Shanghai Jiaotong University*, 2019, 53(10): 1218–1224.

[7] LIU Yanjiang, ZHAO Yiqiang, HE Jiaji, *et al*. A statistical test generation based on mutation analysis for improving the Hardware Trojan detection[J]. *Journal of Circuits, Systems and Computers*, 2020, 29(3): 2050049. doi: 10.1142/S0218126620500498.

[8] SALMANI H. COTD: Reference-free hardware Trojan detection and recovery based on controllability and observability in gate-level netlist[J]. *IEEE Transactions on Information Forensics and Security*, 2017, 12(2): 338–350. doi: 10.1109/TIFS.2016.2613842.

[9] CUI Xiaotong, KOOPAHI E, WU Kaijie, *et al*. Hardware Trojan detection using the order of path delay[J]. *ACM Journal on Emerging Technologies in Computing Systems*, 2018, 14(3): 33.

[10] WAKSMAN A, SUOZZO M, and SETHUMADHAVAN S. FANCI: Identification of stealthy malicious logic using Boolean functional analysis[C]. The 2013 ACM SIGSAC Conference on Computer & Communications Security (ACM-CCS), Berlin, Germany, 2013: 697–708.

[11] HASEGAWA K, OYA M, YANAGISAWA M, *et al*. Hardware Trojans classification for gate-level netlists based on machine learning[C]. 2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS), Sant Feliu de Guixols, Spain, 2016: 203–206.

[12] HASEGAWA K, YANAGISAWA M, and TOGAWA N. Hardware Trojans classification for gate-level netlists using

multi-layer neural networks[C]. 2017 IEEE 23rd International Symposium on On-line Testing and Robust System Design, Thessaloniki, Greece, 2017: 227–232.

[13] HASEGAWA K, YANAGISAWA M, and TOGAWA N. Trojan-feature extraction at gate-level netlists and its application to hardware-Trojan detection using random forest classifier[C]. 2017 IEEE International Symposium on Circuits and Systems (ISCAS 2017), Baltimore, USA, 2017: 1–4.

[14] HASEGAWA K, YANAGISAWA M, and TOGAWA N. A hardware-Trojan classification method utilizing boundary net structures[C]. 2018 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA, 2018: 1–4.

[15] Trust-Hub [EB/OL]. http://www.trust-hub.org, 2021.

[16] ZHANG Jie, YUAN Feng, and XU Qiang. DeTrust: Defeating hardware trust verification with stealthy implicitly-triggered hardware Trojans[C]. The 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, USA, 2014: 153–166.

[17] HUANG Zhao, WANG Quan, CHEN Yin, *et al*. A survey on machine learning against hardware Trojan attacks: Recent advances and challenges[J]. *IEEE Access*, 2020, 8: 10796–10826. doi: 10.1109/ACCESS.2020.2965016.

[18] BHUNIA S, HSIAO M S, BANGA M, *et al*. Hardware Trojan attacks: Threat analysis and countermeasures[J]. *Proceedings of the IEEE*, 2014, 102(8): 1229–1247. doi: 10.1109/JPROC.2014.2334493.

[19] HU Wei, CHANG C H, SENGUPTA A, *et al*. An overview of hardware security and trust: Threats, countermeasures, and design tools[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021, 40(6): 1010–1038. doi: 10.1109/TCAD.2020.3047976.

[20] CHEN Tianqi and GUESTRIN C. XGBoost: A scalable tree boosting system[C]. The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, USA, 2016: 785–794.

张　颖：女，1977年生，博士，讲师，研究方向为集成电路设计、验证与测试、硬件安全.

李　森：男，1995年生，硕士生，研究方向为集成电路验证与测试、硬件安全.

陈　鑫：男，1982年生，博士，副教授，研究方向为数字集成电路设计.

姚嘉祺：男，1996年生，硕士生，研究方向为集成电路验证与测试、硬件安全.

毛志明：男，1997年生，硕士生，研究方向为集成电路验证与测试.

责任编辑：余　蓉