SVM算法在硬件木马旁路分析检测中的应用

佟 鑫 李 莹* 陈 岚
(中国科学院微电子研究所EDA中心 北京 100029)

摘 要：集成电路(ICs)面临着硬件木马(HTs)造成的严峻威胁。传统的旁路检测手段中黄金模型不易获得，且隐秘的木马可以利用固硬件联合操作将恶意行为隐藏在常规的芯片运行中，更难以检测。针对这种情况，该文提出利用机器学习支持向量机(SVM)算法从系统操作层次对旁路分析检测方法进行改进。使用现场可编程门阵列(FPGA)验证的实验结果表明，存在黄金模型时，有监督SVM可得到86.8%的训练及测试综合的平均检测准确率，进一步采用分组和归一化去离群点方法可将检测率提升4%。若黄金模型无法获得，则可使用半监督SVM方法进行检测，平均检测率为52.9%～79.5%。与现有同类方法相比，验证了SVM算法在指令级木马检测中的有效性，明确了分类学习条件与检测性能的关系。

关键词：硬件木马；旁路检测；支持向量机；有监督学习；半监督学习
中图分类号：TN406 **文献标识码**：A **文章编号**：1009-5896(2020)07-1643-09
DOI: 10.11999/JEIT190532

Application of SVM Machine Learning to Hardware Trojan Detection Using Side-channel Analysis

TONG Xin LI Ying CHEN Lan
(*EDA Center of Institute of Microelectronics*, *Chinese Academy of Sciences*, *Beijing* 100029, *China*)

Abstract: Integrated Circuits (ICs) are suffering severer threats caused by Hardware Trojans (HTs), some of which hide in routine operations by coercing firmware or hardware. Along with conventional side-channel detection not always getting golden-chip, HTs become more difficult to detect. An improved Support Vector Machine (SVM) machine learning frameworks for this is proposed using system-level side-channel analysis. Cross validation experimental results on Field Programmable Gate Array (FPGA) show that in the condition of golden-chip, supervised SVM achieves 85.8% test accuracy in average. After grouping, outlier-removing and normalization, it rises by 4%. Even if golden-chip is out of hand, semi-supervised SVM has accuracy to judge HTs existence, averaging in 52.9%-79.5% under different test modes. Comparing with existing researches, this work verifies the efficiency of SVM for HT detection in instruction level, and points out the relationship between diversified learning conditions with detection performance.

Key words: Hardware Trojan (HT); Side-channel analysis; Support Vector Machine (SVM); Supervised learning; Semi-supervised learning

1 引言

随着集成电路设计与制造供应链全球化的发展，使得各个工业环节进一步细化和分离，第三方知识产权(Intellectual Property, IP)核提供者或工艺厂商可以在设计或制造阶段植入恶意的电路——硬件木马(Hardware Trojans, HTs)以窃取保密信息、监视或控制核心功能或使系统失效，而且由于硬件木马仅在特定条件下被触发，极难被检测，因此近年来硬件木马受到了广泛的关注。目前硬件木马的检测方法主要有4类[1]：基于逆向工程、基于侧信道分析、逻辑测试和可测性设计。

基于逆向工程的版图比对技术将芯片剖片拍照再进行比较，对芯片具有很强破坏性，并且检测所需资金投入大、时间消耗长。其他3类非破坏性检

收稿日期：2019-07-15；改回日期：2020-03-06；网络出版：2020-04-22
*通信作者： 李莹 liying1@ime.ac.cn
基金项目：国家物联网与智慧城市重点专项对接(Z181100003518002)，北京市自然科学基金(4184106)，北京市科技专项(Z171100001117147)
Foundation Items: The National Internet of Things and Smart City Key Project Docking(Z181100003518002), The Natural Science Foundation of Beijing (4184106), The Beijing Science and Technology Project (Z171100001117147)

测方法中，逻辑测试根据特定条件测试硬件木马的逻辑输出值，随着集成电路规模增大，遍历所有测试矢量并不可行，该领域研究集中在如何找到有效测试矢量。

侧信道分析(side-channel analysis)方法，也称旁路检测，是目前硬件木马检测领域最有效的手段之一。旁路分析基于电路的瞬时信号、泄露电流、电路延时、电磁辐射等旁路参数或它们的多参数结合[2–6]，以发现被更改的电路设计。但是，复杂的木马增加了逻辑测试的难度，在特定模式下触发的木马所产生的影响，可能小到被工艺偏差和正常功能掩盖，一些违反运行时间操作的隐秘木马甚至可以基于设计规则绕过验证[7]，从而严重降低旁路检测手段的性能。为此研究人员也结合可测性设计提出了全生命周期的硬件木马测试，在电路中加入内置传感器[8]或管理动态热分布[9]。这些技术意在监测特定条件下的特定性质，因此不可避免地需要精确校准以符合环境变化。而机器学习(machine learning)通过研究计算手段，利用经验(训练数据)产生模型来改善系统自身性能。将其应用于硬件木马检测中，可以通过提取1维或多维的电路特征数据进行数据拟合或聚类，自动区分出木马电路并实现自校准，还可将大量数据快速分类。

Jap等人[10]使用SVM算法和非监督模型通过期望极大算法检测AES加密的泄露，Bao等人[11]使用K均值聚类和SVM区分IC电路版图，文献[12]比较了有无触发电路条件下SVM检测硬件木马的分类结果。目前将SVM用于硬件木马检测的研究主要包括对IC门级网表、反向电路或通信表现行为等的检测[10–13]，但这些工作都是针对单独的基准电路或芯片中独立的IP，对其系统级检测的适用性优化和分析方法研究尚不完善。若涉及嵌入式处理器系统情况可能极为不同，特别是在固件行为、物理状态映射和检查规则等方面，对硬件木马检测领域的特征提取和分类方法选择等问题提出了新的挑战。在文献[14]中Lodhi等人提出使用微控制器指令级别的功耗情况在运行测试中区分芯片行为，但没有考虑功耗提取时不同特征条件的影响；虽然比较了4种机器学习方法的测试准确率，但未涉及SVM算法，对有无黄金模型下的机器学习提升过程中也没有明确的说明。与之不同，本文利用指令级旁路功耗信息提取反映固硬件综合行为的特征，使用有监督和半监督两种SVM算法对多类木马进行多种模式的检测、比较，验证算法的适用性。主要完成以下几方面内容：

(1) 从特征指令集生成量化可学习的指令级旁路功耗分析并提出使用SVM方法进行分类的整体检测框架；

(2) 建立SVM算法在有监督学习和半监督学习两种方式下的检测模型，使用分组、归一化、去离群点等方式提升检测性能；

(3) 使用MC8051微控制器和开源的测试基准电路[15]在Altera FPGA上建立完整的硬件验证环境。依据检测准确率和运算开销，在多种学习和测试模式下分别开展不同核函数、不同无木马电路比例条件下的性能评估；

(4) 针对有监督SVM学习中核函数的选择、检测性能提升等进行实验和分析，并对半监督SVM在指令级木马检测中的有效性进行了验证，明确分类学习条件与检测性能的关系。

2 基于SVM的指令级硬件木马旁路功耗分析与检测框架

2.1 指令级旁路功耗特征提取

与文献[16]相同，采取I_{DDT}-I_{DDQ}分析，通过动态电流(I_{DDT})与静态电流(I_{DDQ})的比值消除工艺扰动对功耗数据造成的影响，从测试和目标端共同提取如下2类特征：

(1) 不同指令：根据MC8051微处理器的汇编指令，本文选择了典型的7类共21条指令，和8组操作数应用到指令集进行实验，指令集选择如表1所示；

(2) 不同木马：由于不同结构的HTs和它们攻击芯片的方式在行为上表现出巨大的差异，机器学习模型需要知道不同木马的差异以提高分类性能，因此第2个特征是HTs测试向量的类型。采用Trust-Hub上提供的6个针对MC8051的木马，3个(HT1～HT3)给原始设计增加了额外的逻辑功能，3个(HT4～HT6)移除/关闭/修改了原设计的部分逻辑。详细描述如表2所示。

2.2 基于SVM算法的硬件木马检测框架

提出的检测框架主要包括5个步骤，如图1所示。

(1) 预处理：将不同指令作为激励对目标电路测试下得到的功耗数据按照特征的约束组进行归一化、求比值、约束精度等处理，以适应SVM算法使用；

(2) 采样：选取与特征约束对应的学习模式：木马敏感(MOD1)，指令敏感(MOD2)，混合敏感(MOD3)以及混合不敏感(MOD4)等，并进行步进式随机抽样、去离群点及标准化处理等。在每一个模式中，定义比例随机抽取独立不相关的n个特征向量子集，一部分作为训练数据集，另一部分为测试数据集。在仿真中主要采用2种方式：(a) 从HTs基准中随机抽取一个整组作为测试数据(MOD1和

表 1　指令集

指令序号及名称			指令类型	描述
	1-NOP		类型1 NOP	无操作
2-MOV_A_RR	3-MOV_A_D	4- MOV_A_DATA		
5-MOV_RR_A	6-MOV_RR_D	7-MOV_RR_DATA	类型2 MOV	移动存储器，复制操作数2到操作数1
8-MOV_D_A	9-MOV_D_RR	10-MOV_D_DATA		
11-ADD_A_RR	12-ADD_A_D	13-ADD_A_DATA	类型3 ADD	加法器加操作，将操作数的值加到加法器上并存储
14-SUBB_A_RR	15-SUBB_A_D	16-SUBB_A_DATA	类型4 SUBB	从加法器中借位减操作
17-INC_A	18-INC_D	19-INC_RR	类型5 INC	增加操作数
	20-JMP_A_DPTR		类型6 JMP	跳转至数据指针+ DPTR代表的加法器地址
	21-JNC		类型7 JNC	跳转至相关地址如果进位没有设置

表 2　木马基准电路

名称	描述
HT1	MC8051-T200，这个木马在空闲模式激活8051内部计时器
HT2	MC8051-T300，这个木马在8051通过UART发送特定数据串时被触发。目的是通过UART收到任意信息
HT3	MC8051-T500，这个木马的触发器检测特定的命令，当木马激活后其负载可以替换特定的数据
HT4	MC8051-T600，这个木马使得微控制器上运行算法的任何跳转失效
HT5	MC8051-T700，这个木马用敌人预设数据替换一些输入数据
HT6	MC8051-T800，这个木马当UART接收特殊字符时篡改堆栈指针

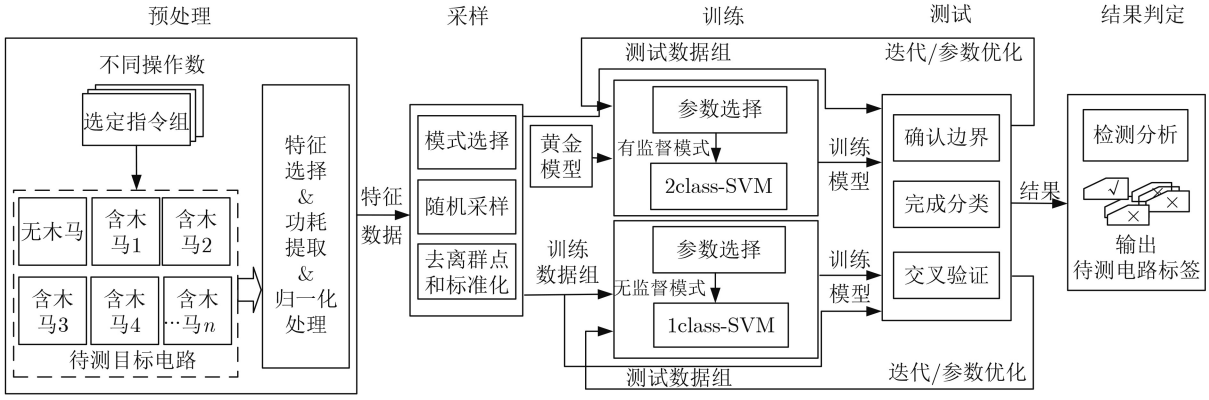


图 1 基于SVM算法的硬件木马检测框架

MOD3)；(b) 从所有特征向量中随机选取一部分数据作测试(MOD2和MOD4)。

(3) 训练：根据有无黄金模型使用有监督和半监督算法分别进行训练，确定初始参数，并使用黄金模型数据(如果存在)或抽样出的训练数据进行学习，收集反馈得到的参数优化建议，循环训练出满意模型。

(4) 测试：使用测试数据对训练模型界定判决边界、完成本轮分类和交叉验证，结果将反馈优化参数。

(5) 结果判定：当分类结果满足收敛和阈值要求后停止循环，输出判定的最终结果、检测准确率等的分析和待测电路的标签。

3　模型建立

支持向量机(Support Vector Machine, SVM)与其他经典的机器学习方法比较，优点在于可以做高维空间的小样本学习且结构风险最小化。根据“输入-方法-输出-评估”的机器学习一般流程，结合指令级功耗特征，建立基于SVM的硬件木马分类模型。

3.1　模型输入数据处理

为降低数据噪声影响，对于输入模型的动态功耗/静态功耗数据进行离群点的剔除。为了进一步减少误差和加速模型收敛，将训练集和其他测试集统一进行了以训练集中的均值作为标准的归一化处理。

使用进行处理后的数据建立基于SVM算法的硬件木马检测模型，主要包含3个步骤：

步骤 1　选择一个核函数应用训练数据集和SVM算法进行模型训练；

步骤 2　对训练模型进行交叉验证及改进；

步骤 3　应用改进的训练模型对待测电路进行分类。

在步骤1中，核函数$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)$实现了样本$\boldsymbol{x}_i$与$\boldsymbol{x}_j$从原始空间映射到一个更高维的特征空间，使得样本在这个特征空间内线性可分的功能。由于核函数是隐式定义特征空间，并且对学习模型性能影响重大，因此使用者需要通过比较结果性能选取最佳核函数。根据训练数据有无标签的区别，分别建立有监督学习模型和半监督学习模型。本文选取常用的线性核、多项式核和高斯核(见表3)并建立了有监督模式下的SVM学习模型，通过比较检测准确率和效率选取最优核函数。半监督SVM学习模型中沿用了该选择结果。

3.2 SVM有监督学习模型

以“黄金电路(golden chip)”作为无木马侵入的安全电路(Trojan free)，相应的功耗数据标记为TF(标签值为1)，通过第2节中HT1-HT6木马电路得到的数据标记为T1-T6(标签值为–1)，统称为TI(Trojan Inserted)数据。对这两类数据进行抽取采样，建立基于SVM算法的有监督学习模型。考虑到硬件木马既可能增加电路功耗，也有可能使之减小，为了更准确的学习模型和更快速的学习效率，与一般的机器学习过程不同，本文提出以TF组中位数为基准对训练和测试数据进行分组，分别建立上边界模型和下边界模型的策略，详细步骤如图2所示。

表 3　SVM常用核函数

名称	表达式	参数
线性核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{x}_j$	
多项式核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=(\boldsymbol{x}_i{}^{\mathrm{T}}\boldsymbol{x}_j)^d$	$d\geq 1$为多项式的次数
高斯核	$\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp\left(-\dfrac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}\right)$	$\sigma>0$为高斯核的带宽

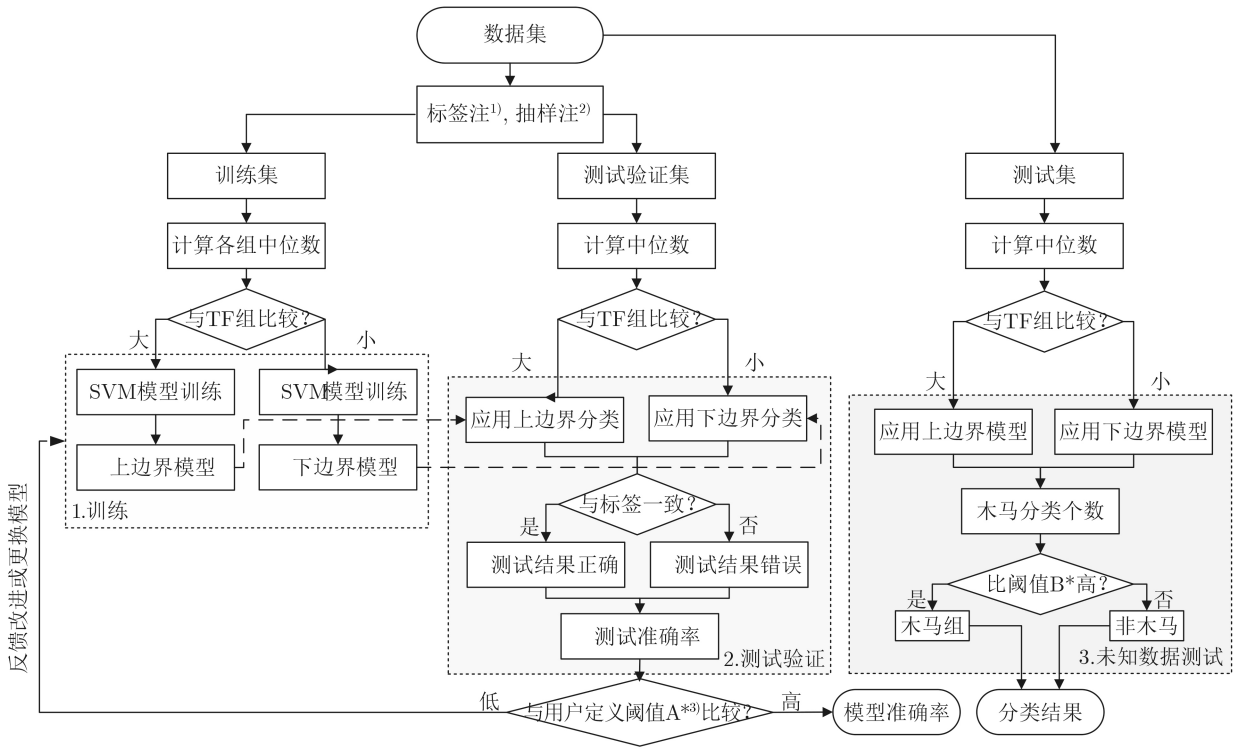


图 2 指令功耗数据分组的SVM有监督学习分类流程

1) 将正常组数据(Trojan Free, TF)标记为1，木马组(T1, T2, …)数据为–1。

2) 按照采样模式对数据进行分组抽样，每次抽出一部分木马数据做测试验证集，其余做训练集，测试集可选择未知标签的数据。

3) 阈值A为用户期望达到的模型测试准确率阈值B是用户定义的数据组中木马个数的占比，用于判断该组数据是否为木马组。

首先，对用于训练及验证的数据集进行标签和交叉验证抽样。分组抽样采用不同的模式，每次抽出一部分木马数据作为测试验证集，其余做训练集，并保证在训练数据中同时包含标签1和–1。

随后计算TF组数据的中位数并进行分组：大于或等于TF中位数的训练数据用于训练SVM上边界模型，小于的则训练下边界模型；同理，大于或等于TF中位数的测试数据应用训练得到上边界模型进行分类预测，小于的则应用下边界模型。

由于测试验证集数据本身带有标签，将预测分类结果与自身标签进行比较即可知道分类结果是否正确。在这里存在4种结果：

(1) True Positive (TP)：预测结果为1，数据本身标签为正常，为符合期望的正结果，记为真正类；

(2) True Negative (TN)：预测结果为–1，数据本身标签为木马，为符合期望的负结果，记为真负类；

(3) False Positive (FP)：预测结果为1，数据本身标签为木马，为错误正判断，记为假正类；

(4) False Negative(FN)：预测结果为–1，数据本身标签为正常，为错误负判断，记为假负类。

相应地上边界模型的预测结果分别为TP_H, TN_H, FP_H, FN_H，下边界预测结果为TP_L, TN_L, FP_L, FN_L。

定义准确率(Accuracy Rate, AR)为

$$AR=\frac{TP_H+TP_L+TN_H+TN_L}{TP_H+TP_L+FN_H+FN_L+FP_H+FP_L+TN_H+TN_L} \tag{1}$$

比对测试结果和预设期望测试准确率A(本文中设为75%)，如果AR$<A$，说明训练模型没有达到用户期望，需要反馈改进参数或更换新的模型，AR$\geq A$则说明训练模型可以用于接下来的未知数据测试。

对于预处理之后的未知类别电路的功耗数据测试集，同样通过比较TF中位数的方式选择应用上边界或下边界分类模型，累计得到属于–1类别(TI)的个数。阈值B是用户定义的可容忍的木马个数占比(文中设定为50%)，用于判断该组数据是否为TI——如果测试结果高于B，将被判定为有木马植入电路(TI)，低于则判定为安全(TF)。

3.3 SVM半监督学习模型

与有监督SVM相比，半监督SVM考虑未标记的样本，试图找到能将两类样本分开，且穿过数据低密度区域的划分超平面。由于在IC设计制造过程中被植入隐秘的硬件木马可能性极高，且黄金电路通常不易获得，半监督学习模型具有更普遍的实用性，即假设样本不携带标签，但在其中存在未知比例的TF和TI数据，期望通过模型进行分类。通过前期研究工作发现，在功耗数据集当中TF的占比与SVM半监督学习模型性能密切相关。因此专门针对TF占比变化对半监督SVM建立模型流程图如图3。

4　实验与结果分析

4.1 实验环境、测试方案、前提与假设

基于实验室SoC/IP快速原型验证平台搭建硬件验证环境，在Altera Stratix II FPGA中分别实现了MC8051微控制器和相关的Trust-Hub木马基准电路。通过Quartus II中PowerPlay工具收集功

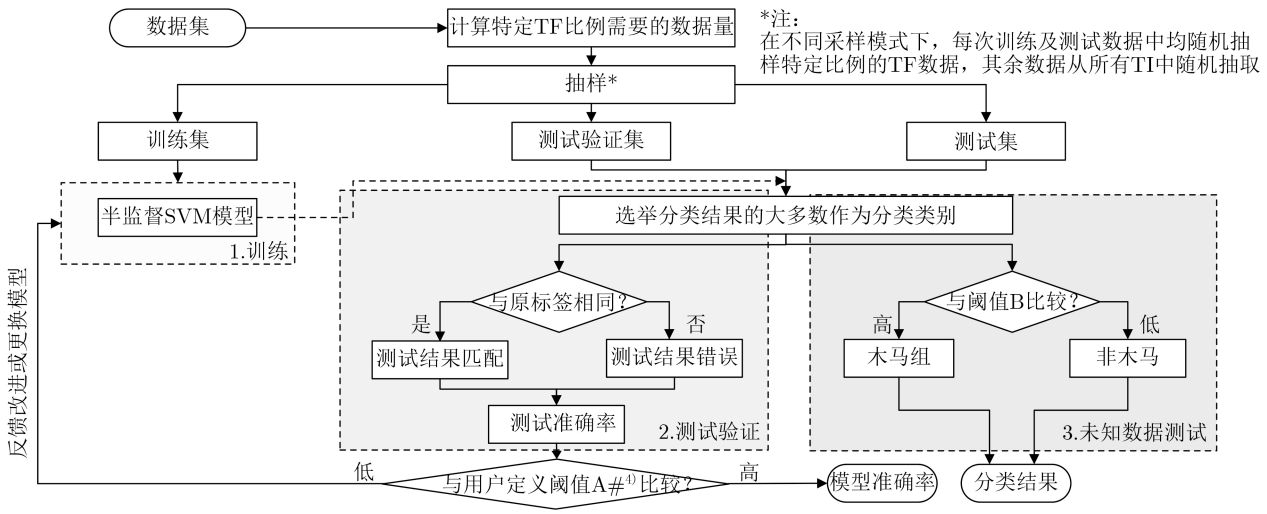


图 3 指令功耗数据分组的SVM半监督学习分类流程

4) 阈值A为用户期望达到的模型测试准确率阈值B是用户定义的数据组中占比多数的数据是否为木马，用于判断该组数据是否为木马组。

耗数据，随后将数据集导入计算终端机中，使用Python中Scikit-learn框架的SVM方法实现软件算法和模型，计算机使用主频为3.10 GHz的Intel i5-2400 CPU。在实验中通过蒙特卡罗方法消除环境噪声对实验结果的影响。

在有监督SVM学习中，对比分组与不分组两种方式，分别使用线性、多项式核和高斯核进行4种采样模式下的检测，观察检测准确率(以下简称检测率)——包括训练准确率(以下简称训练率)、测试准确率(以下简称测试率)——和运行时间。其中训练率表明了模型学习的可信度，测试率表明了模型检测的可用性，运行时间表明了模型的运行效率。

在半监督SVM学习中，使用有监督学习选择的SVM核函数，对比将TF数据在整体中的占比从0.1增加至0.9时，对训练率和测试率的影响。

4.2 实验结果分析

4.2.1 有监督SVM学习

如表4所示，在木马敏感的模式(MOD1)下，3种核函数的训练率和测试率均能达到较高的水平，分组方式对缩短运行时间的效果较为明显，线性核函数的用时最短。

在指令敏感模式(MOD2)下，受单指令数据抽样数量限制，无法保证同时拥有2个类别的数据，导致不能进行SVM模型学习。

如表5所示，在混合敏感模式(MOD3)下，整体趋势与木马敏感模式下具有一定相似性，分组实验降低了一定的测试率，但线性核在分组条件下具有最高训练率。

如表6所示，在混合不敏感模式(MOD4)下，不分组实验中训练率和测试率的结果近似，线性核用时最短。分组实验对检测率效果提升明显，其中线性核训练率最高，用时最短。

综上，分组的方式较为明显地提升了有监督SVM学习的效率和训练率，但不同的采样模式对检测结果影响较大；针对实验中的指令集功耗数据，线性核的表现较为均衡且效率更高。

表 4 MOD1检测率及运行时间对比表

	线性核函数准确率及运行时间			多项式核函数准确率及时间			高斯核函数准确率及时间		
	训练(%)	测试(%)	时间(s)	训练(%)	测试(%)	时间(s)	训练(%)	测试(%)	时间(s)
无预处理	83.30	100.00	0.02620	83.30	67.31	0.06912	83.30	57.24	0.10902
预处理+分组	98.00	83.30	0.01261	85.80	99.10	0.03929	98.00	83.30	0.02698

表 5 MOD3检测率及运行时间对比表

	线性核函数准确率及时间			多项式核函数准确率及时间			高斯核函数准确率及时间		
	训练(%)	测试(%)	时间(s)	训练(%)	测试(%)	时间(s)	训练(%)	测试(%)	时间(s)
无预处理	83.30	100.00	0.06944	83.30	100.00	0.06193	83.30	100.00	0.06800
预处理+分组	98.80	83.30	0.06526	66.70	100.00	0.07183	88.00	84.90	0.07139

表 6 MOD4检测率及运行时间对比表

	线性核函数准确率及时间			多项式核函数准确率及时间			高斯核函数准确率及时间		
	训练(%)	测试(%)	时间(s)	训练(%)	测试(%)	时间(s)	训练(%)	测试(%)	时间(s)
无预处理	85.94	85.08	0.02049	85.60	86.07	0.03807	85.83	85.38	0.05487
预处理+分组	97.80	97.80	0.01222	86.70	87.00	0.03904	97.60	98.40	0.03709

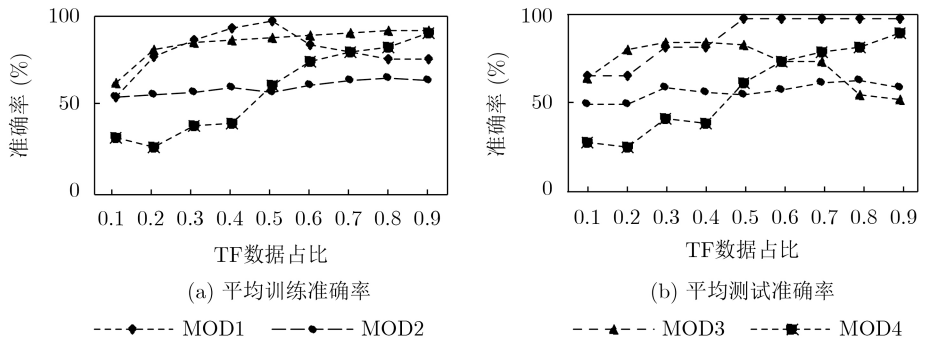


图 4 半监督学习下准确率均值随TF数据占比变化的情况

4.2.2 半监督SVM学习

在半监督SVM学习实验中，使用线性核函数，改变TF在数据中的占比，实验结果如下图4。

从图4中可以看出，在训练过程中，MOD1呈现了随TF比例增加先增大后减小的训练率趋势，在TF=0.5时达到峰值，且是所有TF占比在4种模式下的最大值。MOD2-4的训练率则是基本随TF的增大而增大，其中MOD2变化最为明显，在TF<0.5时准确率不超过50%，在TF=0.9时达到90%准确以上；MOD4变化最为平缓，训练率在(50%, 70%)范围内波动上升，说明TF的占比对准确率影响较小；MOD3在TF<0.3时训练率上升趋势较为快速，之后变为平稳上升。

在测试过程中，与训练率变化过程相似的是MOD2和MOD4的测试率曲线，数值也较为接近。MOD1随TF增大到0.5后测试率达到1并保持，MOD3先上升后下降，在TF=0.4时有测试准确率最大值86.67%。

从整体上看，检测率基本呈现了随TF占比增大而增大的趋势。MOD1在训练中表现出与其他3种模式的异常变化趋势有可能说明了不同类型的木马仍对半监督SVM的判断结果有较大影响。在测试验证表现中，MOD1, MOD3都表现极端，测试率非0即1，因此MOD3平均测试率的下降意味着测试指令对应的木马数据中无法被区分的更多了。这一现象的产生可能存在2种原因：(1)分指令数据规模小导致的误差；(2)训练数据造成过拟合。

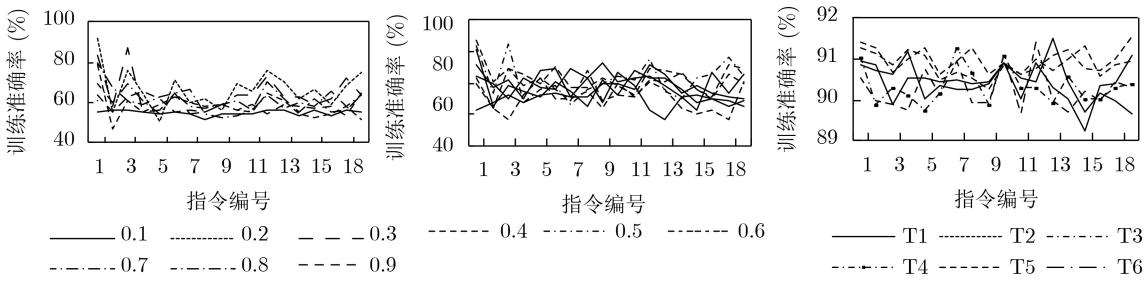
在指令敏感的MOD2下，将全部木马组数据按比例随机抽取测试组，抽取比例最大不超过0.5。取所有指令在4个抽取比例下的平均值，观察到随着TF占比的变化，指令敏感度如图5(a)，图5(b)所示。结果显示指令1的的训练率和测试率都是最高的，且在训练率随TF的增大而增大。若选择MOD4的测试数据抽取比例为0.2，TF比例为0.9时有最佳平均训练率；若抽取比例为0.4，则TF比例为0.8时有最佳平均训练率，且拥有全局最高平均训练率。

从混合敏感MOD3中可以观察到T1-T6对每条指令的敏感差异。取每条指令在TF占比0.1–0.9条件下的训练准确率均值，可得图5(c)。图中单一木马组对不同指令的准确率越高，说明对该条指令越敏感。T1敏感度最高的是指令19，最低的是指令15(记为<I19, I15>), T2则为<I19, I11>, T3-T6依次为<I15, I9>, <I7,I5>, <I13, I19>, <I12, I11>。这个差异可用于对安全性未知的测试电路潜在木马的判断，例如，若电路在测试中表现出与T1相同的指令敏感度组合，推测存在T1类型木马的可能性。

如图6所示，在混合不敏感的MOD4中不论抽取比例多少，均呈现了训练率和测试率随TF占比增大而上升的趋势。其中，相比其他3个比例值，抽取比例为0.4时平均训练率和测试率均是4个比例中最高的，波动也是最小的。

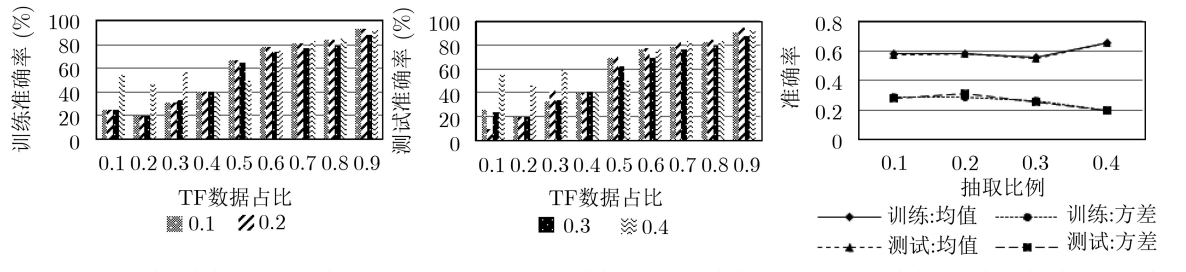
总结并分析上述实验结果，得出以下结论：

(1) 在有监督SVM学习中，综合训练率、测试



(a) MOD2对不同TF占比的指令训练敏感度　(b) MOD2对不同TF占比的指令测试敏感度　(c) MOD3对不同木马组的指令敏感度

图 5 各条指令表现的敏感度差异情况



(a) 训练准确率随TF占比变化　(b) 测试准确率随TF占比变化　(c) 准确率的均值方差随抽取比例变化

图 6 MOD4不同条件下的准确率变化情况

率和运行时间，在分组条件下训练率明显提升，使用线性核函数的测试结果最佳，木马与指令不敏感模式下可以达到97.8%的训练率和测试率，使用分组和归一化去离群点方法可进一步将平均检测率提升12.7%。分组后线性核函数的平均测试用时约0.03 s，比最慢的多项式核节约40%。

(2) 半监督的SVM机器学习用于无黄金参考模型的情况。MOD2方法能较为准确地区分电路中是否存在木马，而MOD3使用对木马组敏感的指令组合可以帮助判断是否存在某种类型的木马。半监督SVM的检测率范围为57.9%～88.7%，这说明了机器学习的聚类方式对于指令集功耗数据在检测硬件木马方面存在适用和有效性，但检测效果不如有监督的分类方式。

在现有文献[10–12]中，主要是基于各类电路特征使用SVM算法对不同功能或大小的硬件木马电路进行检测，由于检测对象和实验条件的不同，检测率波动范围较大。其中有监督学习的检测率基本在18%以上，无监督学习的检测率在57.6%以上，本文提出的系统级检测方法在两种模式下的检测率分别大于83.3%和57.9%。与同样进行指令功耗的有监督学习[14]结果相比，本文提出的SVM算法检测率仅次于K近邻算法(99.02%)，优于决策树(94.84%)、深度学习(87.09%)和朴素贝叶斯算法(86.46%)，根据同等实验条件下的比较结果，在数据量较大情况下，K近邻算法的效率低于SVM，但在文献[14]中未对此性能进行说明。因此，SVM算法在综合检测率和运算效率的条件下可作为优选的分类方法。

5 结束语

针对传统旁路检测手段中硬件木马隐蔽性带来的难度和黄金模型不易获得的现状，本文建立基于SVM的指令级旁路功耗硬件木马检测框架和有监督、半监督的SVM算法模型。在Altera FPGA下对功耗数据处理、SVM核函数、黄金电路占比等条件下的检测准确率进行分析。实验结果表明：在有监督模型中，线性核函数检测准确率较高，速度最快，平均训练率为84.18%、测试率为95.03%，使用分组和归一化去离群点方法可进一步将训练率提升4%；在半监督模型中，检测率与待测数据中预计的无木马电路比例成正比，在极端实验情况下检测平均值也可达52.9%。从集成电路系统层面，验证了SVM算法在指令级木马检测中的适用性，明确了分类学习条件与检测性能的关系。下一步将对多指令组合、多评估标准及半监督学习的检测性能提升等方面工作展开进一步研究。

参 考 文 献

[1] 钟晶鑫, 王建业, 阚保强. 基于温度特征分析的硬件木马检测方法[J]. 电子与信息学报, 2018, 40(3): 743–749. doi: 10.11999/JEIT170443.
ZHONG Jingxin, WANG Jianye, and KAN Baoqiang. Hardware Trojan detection through temperature characteristics analysis[J]. *Journal of Electronics & Information Technology*, 2018, 40(3): 743–749. doi: 10.11999/JEIT170443.

[2] RAD R M, WANG Xiaoxiao, TEHRANIPOOR M, *et al*. Power supply signal calibration techniques for improving detection resolution to hardware Trojans[C]. 2008 IEEE/ACM International Conference on Computer-Aided Design, San Jose, USA, 2008: 632–639. doi: 10.1109/ICCAD.2008.4681643.

[3] LAMECH C, AARESTAD J, PLUSQUELLIC J, *et al*. REBEL and TDC: Two embedded test structures for on-chip measurements of within-die path delay variations[C]. 2011 IEEE/ACM International Conference on Computer-Aided Design, San Jose, USA, 2011: 170–177. doi: 10.1109/ICCAD.2011.6105322.

[4] DU Dongdong, NARASIMHAN S, CHAKRABORTY R S, *et al*. Self-referencing: A scalable side-channel approach for hardware Trojan detection[C]. The 12th International Workshop on Cryptographic Hardware and Embedded Systems, Santa Barbara, USA, 2010: 173–187. doi: 10.1007/978-3-642-15031-9_12.

[5] HE Jiaji, ZHAO Yiqiang, GUO Xiaolong, *et al*. Hardware Trojan detection through chip-free electromagnetic side-channel statistical analysis[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017, 25(10): 2939–2948. doi: 10.1109/TVLSI.2017.2727985.

[6] NARASIMHAN S, DU Dongdong, CHAKRABORTY R S, *et al*. Multiple-parameter side-channel analysis: A non-invasive hardware Trojan detection approach[C]. 2010 IEEE International Symposium on Hardware-Oriented Security and Trust, Anaheim, USA, 2010: 13–18. doi: 10.1109/HST.2010.5513122.

[7] LIU Yu, JIN Yier, NOSRATINIA A, *et al*. Silicon demonstration of hardware Trojan design and detection in wireless cryptographic ICs[J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017, 25(4): 1506–1519. doi: 10.1109/TVLSI.2016.2633348.

[8] FORTE D, BAO Chongxi, and SRIVASTAVA A. Temperature tracking: An innovative run-time approach for hardware Trojan detection[C]. 2013 IEEE/ACM International Conference on Computer-Aided Design, San Jose, USA, 2013: 532–539. doi: 10.1109/ICCAD.2013.6691167.

[9] ZHAO Hong, KWIAT K, KAMHOUA C, *et al*. Applying chaos theory for runtime hardware Trojan detection[C]. 2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Verona, USA, 2015: 1–6. doi: 10.1109/CISDA.2015.7208642.

[10] JAP D, HE Wei, and BHASIN S. Supervised and unsupervised machine learning for side-channel based Trojan detection[C]. The 27th IEEE International Conference on Application-specific Systems, Architectures and Processors, London, UK, 2016: 17–24. doi: 10.1109/ASAP.2016.7760768.

[11] BAO Chongxi, FORTE D, and SRIVASTAVA A. On reverse engineering-based hardware Trojan detection[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2016, 35(1): 49–57. doi: 10.1109/TCAD.2015.2488495.

[12] INOUE T, HASEGAWA K, YANAGISAWA M, *et al*. Designing hardware Trojans and their detection based on a SVM-based approach[C]. The 12th IEEE International Conference on ASIC, Guiyang, China, 2017: 811–814. doi: 10.1109/ASICON.2017.8252600.

[13] KULKARNI A, PINO Y, and MOHSENIN T. SVM-based real-time hardware Trojan detection for many-core platform[C]. 2016 17th International Symposium on Quality Electronic Design, Santa Clara, USA, 2016: 362–367. doi: 10.1109/ISQED.2016.7479228.

[14] LODHI F K, HASAN S R, HASAN O, *et al*. Power profiling of microcontroller′s instruction set for runtime hardware Trojans detection without golden circuit models[C]. The Design, Automation & Test in Europe Conference & Exhibition, Lausanne, Switzerland, 2017: 294–297. doi: 10.23919/DATE.2017.7927002.

[15] TEHRANIPOOR M and SALAMANI H. trust-HUB[OL]. https://www.trust-hub.org/, 2018.

[16] 李莹, 周崟灏, 陈岚. 一种旁路检测方法及装置[P]. 中国专利, CN109684881A, 2019.
LI Ying, ZHOU Yinhao, and CHEN Lan. A bypass detection method and device[P]. China patent, CN109684881A, 2019.

佟　鑫：女，1987年生，助理研究员，主要研究方向为物联网硬件安全与集成电路设计.

李　莹：女，1982年生，副研究员，主要研究方向为物联网硬件安全与集成电路设计.

陈　岚：女，1968年生，研究员，博士生导师，主要研究方向为计算机系统架构与集成电路设计、集成电路硬件安全.

责任编辑：马秀强