# 基于多维结构特征的硬件木马检测技术

严迎建 赵聪慧* 刘燕江

(战略支援部队信息工程大学 郑州 450000)

**摘 要**：硬件木马是第三方知识产权(IP)核的主要安全威胁，现有的安全性分析方法提取的特征过于单一，导致特征分布不够均衡，极易出现较高的误识别率。该文提出了基于有向图的门级网表抽象化建模算法，建立了门级网表的有向图模型，简化了电路分析流程；分析了硬件木马共性特征，基于有向图建立了涵盖扇入单元数、扇入触发器数、扇出触发器数、输入拓扑深度、输出拓扑深度、多路选择器和反相器数量等多维度硬件木马结构特征；提出了基于最近邻不平衡数据分类(SMOTEENN)算法的硬件木马特征扩展算法，有效解决了样本特征集较少的问题，利用支持向量机建立硬件木马检测模型并识别出硬件木马的特征。该文基于Trust_Hub硬件木马库开展方法验证实验，准确率高达97.02%，与现有文献相比真正类率(TPR)提高了13.80%，真负类率(TNR)和分类准确率(ACC)分别提高了0.92%和2.48%，在保证低假阳性率的基础上有效识别硬件木马。

**关键词**：硬件木马检测；IP核；有向图；结构特征；支持向量机

**中图分类号**：TN918; TP309+.1 **文献标识码**：A **文章编号**：1009-5896(2021)08-2128-12

**DOI**: 10.11999/JEIT210003

# Hardware Trojan Detection Based on Multiple Structural Features

YAN Yingjian ZHAO Conghui LIU Yanjiang

(*Strtegic Support Force Information Engineering University*, *Zhengzhou* 450000, *China*)

**Abstract**: Hardware Trojans are the main security threats of the third-party Intellectual Property (IP) cores. The existing pre-silicon hardware Trojan detection methods are difficult to be used in a large amount of hardware Trojans detection and the detection accuracy is hard to be enhanced. A gate-level netlist abstract modeling algorithm is proposed to reduce the cost of trustworthiness analysis method, which establishes a directed graph of the gate-level netlist and stores the graph data into the crosslinked list. Furthermore, the characteristics of hardware Trojans are analyzed in the view of the attacker view and a 7-dimensional feature vector based on the directed graph is proposed. Moreover, a hardware Trojan feature extraction algorithm is proposed to extract the 7-dimensional feature of the gate-level netlist, and a Trojan feature expansion algorithm based on the Synthetic Minority Oversampling Technique and Edited Nearest Neighbor (SMOTEENN) is introduced to expand the number of Trojan samples and the Support Vector Machine (SVM) algorithm is utilized to identify the existence of hardware Trojan. 15 benchmark circuits from the Trust-hub are used to validate the efficacy of the proposed approach and the accuracy rate we achieved is 97.02%. True Positive Rate (TPR) is increased by 13.80%, True Negative Rate (TNR) and ACCuracy (ACC) is increased by 0.92% and 2.48% respectively compared with the existing reference.

**Key words**: Hardware Trojan detection; Intellectual Property (IP) core; Directed graph; Structural feature; Support Vector Machine (SVM)

## 1 引言

近年来赛博空间安全事件频繁爆出，使得信息安全问题再次受到了广泛的关注。集成电路作为信息产业的基础，其“自主可控”与“安全可信”是信息安全的根基。由于集成电路的先进性和复杂性，第三方知识产权(Intellectual Property, IP)核，包括软核、固核和硬核等，大量应用在集成电路设计阶段来缩短产品的开发周期。然而，外购的IP核可能由境外、外资或者合资企业提供，一旦一个环节出现安全问题，将直接影响整个芯片的安全可信[1]。第三方IP核是恶意攻击者的理想藏身之所，黑盒设计中可能早已内置恶意电路，即硬件木马，如图1所示，它可于无声处泄露内部私密信息、篡

收稿日期：2021-01-04；改回日期：2021-03-10；网络出版：2021-06-24
*通信作者： 赵聪慧 1024600921@qq.com

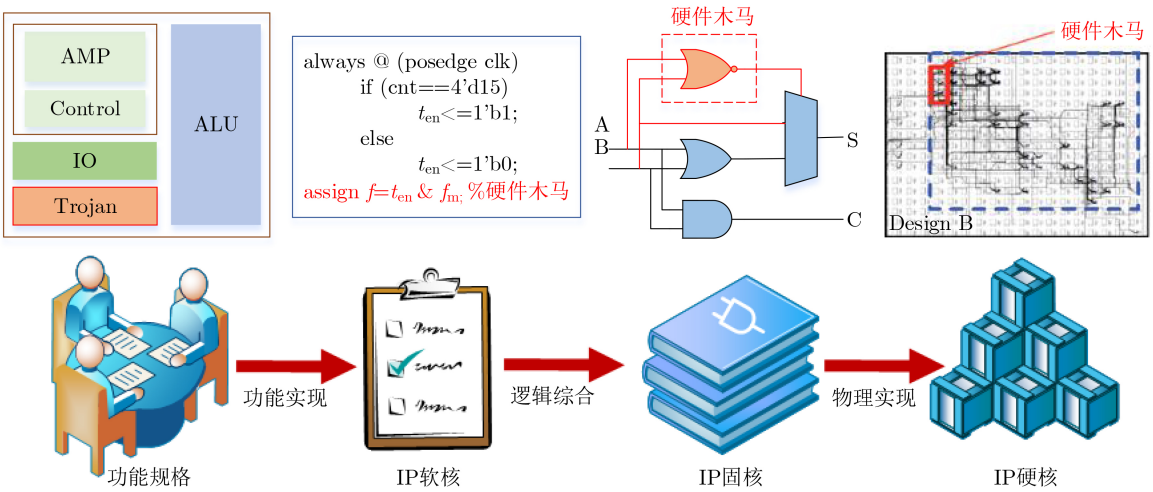



图 1 IP核安全隐患分析

改电路功能和升级系统权限等。此外，目前缺乏IP核的安全可信分析标准和行业规范，第三方IP核已成为硬件木马的“天堂”，进口的集成电路乃至自主设计的芯片的安全可信水平更加难以保障[2]。

硬件木马是IP核的主要安全威胁，如何检测硬件木马受到了国内外研究学者的广泛关注。目前安全性分析主要有形式化验证和木马特征识别两类方法。形式化验证方法评估IP核的属性违例情况来确定其可信任度，评估难度随着电路规模的增加呈指数级增长，验证边界是主要瓶颈，另外安全属性构建大多是“一事一议”，硬件木马类型繁多，构建的安全属性很难涵盖所有的硬件木马类型。硬件木马虽然种类繁多，但在结构上存在多个共性特征，因此，基于特征识别的硬件木马检测方法被广泛研究并成为主流方法。

具体来说，Oya等人[3]总结了9种木马特征并对每种特征赋予特定的分值，通过分值的高低来确定是否存在硬件木马。但该文并未阐述这些特征的性质及与硬件木马触发机制的联系。Yao等人[4]基于数据流图提出4种硬件木马特征，利用硬件木马特征匹配算法来检测硬件木马，并形成了检测工具FASTrust。然而基于数据流图的木马特征构建方法是从寄存器层面进行的，大量的组合逻辑被忽略，误识别率较高。Hasegawa等人[5]提出了LGFi, FFi, FFo, PI, PO等5种硬件木马特征，并利用支持向量机算法来训练并识别木马节点，然而在训练集中，硬件木马特征集较少，训练集分布并不平衡，即便是采用动态加权的支持向量机依然存在较大的误识别情况。Chen等人[6]计算待测电路中两级AONN门的分数，认为分数较高的门是硬件木马。该方法对单触发型硬件木马有效，然而对于多触发条件的硬件木马无能为力，且未考虑有效载荷电路及其功能。

因此，本文构建了扇入单元数、扇入触发器数、扇出触发器数、输入拓扑深度、输出拓扑深度、选择器数量和反相器数量的硬件木马特征。另外，本文建立了基于图结构的电路分析模型，将门级网表映射为有向图模型，最终形成了网表简化分析流程。最后，提出广度优先搜索算法计算网表顶点的硬件木马特征值得分，利用基于最近邻不平衡数据分类算法(Synthetic Minority Oversampling Technique and Edited Nearest Neighbor, SMOTEENN)的硬件木马特征扩展算法来解决木马特征数据集不平衡问题，借助支持向量机(Support Vector Machines, SVM)算法建立硬件木马检测模型并检测出IP核中的硬件木马。

## 2　基于有向图的门级网表抽象化建模算法

目前的IP核安全性分析方法大多基于门级网表开展研究，分析网表的状态是否存在违例情况或者提取网表中的隐藏性结构特征等，然而网表分析效率随着电路规模呈指数级增长，严重限制了验证范围。为了简化硬件木马分析效率，本文研究了门级网表的抽象化建模算法，将门级网表映射为有向图，形成利于分析的数据存储结构，大大提高了分析效率，降低了验证成本。另外，基于有向图可将硬件木马的行为级描述转换为可量化的数据指标，可扩展应用未知硬件木马检测，更具普适性。

### 2.1　门级网表的有向图模型

首先介绍有向图的基本概念。图是由顶点的有穷非空集合和顶点之间边的集合组成的。顶点$v_i$和$v_j$之间的边有方向称为有向边$< v_i, v_j >$。若图中任意两个顶点之间的边均是有向边，则称该图为有向图[7,8]。下面以图2所示的简单电路为例介绍网表的有向图模型，其中$I_1, I_2, I_3, I_4, I_5$和clk为电路的输入，$O_1$和$O_2$为电路的输出。

将电路中所有的输入($I_1, I_2, I_3, I_4, I_5$和clk)、输出端口($O_1, O_2$)和器件单元($N_1, N_2, \cdots, N_8$)映射为有向图的顶点，组成顶点集$\boldsymbol{V}$。将顶点之间的连线映射为有向图的边，每条边的弧尾为与该节点相连的上一级器件单元，弧头为与该节点相连的下一级器件单元，构成边集$\boldsymbol{E}=\{e_1, e_2, \cdots, e_{18}\}$。基于此映射规则，任何一个网表都可以映射为由顶点集$\boldsymbol{V}$和边集$\boldsymbol{E}$组成的有向图$\boldsymbol{G}=(\boldsymbol{V}, \boldsymbol{E})$。

### 2.2 基于十字链表的有向图数据存储

将门级网表映射为有向图后，需要存储有向图的顶点集$\boldsymbol{V}$和边集$\boldsymbol{E}$。邻接表是一种数组与链表相结合的存储方法，由于只存有关联的信息，不存在空间浪费的问题[9]。因此本文采用图的邻接表来存储有向图数据。具体来说，数组用来存储所有的顶点信息，链表用来存储顶点对应的边的信息。

在用邻接表来存储网表的有向图时，需统计各顶点链表中的结点数目，便可得到所有顶点的出度，但要获取各顶点的入度则需要遍历整个邻接表，或者为该有向图建立一个逆邻接表。为了同时计算有向图的出度和入度，本文采用将邻接表和逆邻接表相结合的十字链表，图3的十字链表结构存储如图4所示。其中，顶点表中的data存储可唯一表示该顶点的信息，firstin表示入边表头指针，指向以该顶点为终点的边，firstout表示出边表头指针，指向以该顶点为起点的边。边表中tailvex是指有向边的起点在顶点表的下标，headvex是指有向边的终点在顶点表的下标，headlink是指入边表指针域，指向终点相同的下一条边，taillink是指出边表指针域，指向起点相同的下一条边。

## 3 硬件木马结构特征模型

硬件木马的结构千差万别，类型丰富多样，然而硬件木马在触发和载荷方面具有隐蔽性，在功能方面具有恶意破坏性，因此可以提取出共性特征。本文分析了硬件木马库Trust_Hub[10]以及现有文献给出的多种硬件木马，提出了FAN_IN, FF_IN, FF_OUT, DPI, DPO, MUX和INV 7种硬件木马共性结构特征。

(1)扇入单元特征FAN_IN。硬件木马为了保证隐蔽性，通常会选择多个稀有逻辑值或者状态作为其触发条件，保证在测试验证阶段难以“误触发”，即硬件木马的触发逻辑输入个数较多。图5(a)为硬件木马RS232-T1400的结构，其触发部分是一个组合比较器，当多个条件同时满足时，硬件木马被激活，改变原有信号的值。图5(b)为图5(a)的有

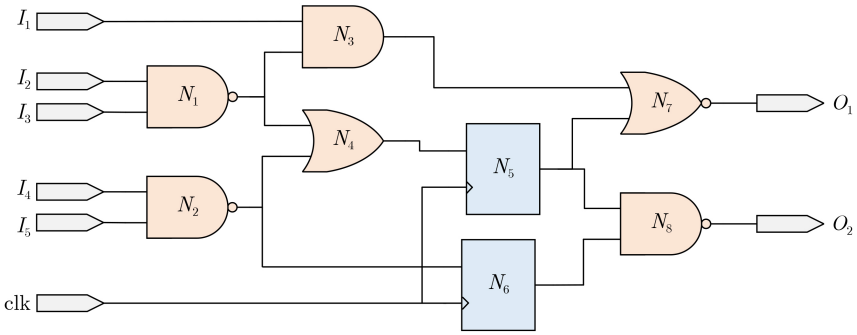



图 2 门级网表等效电路图

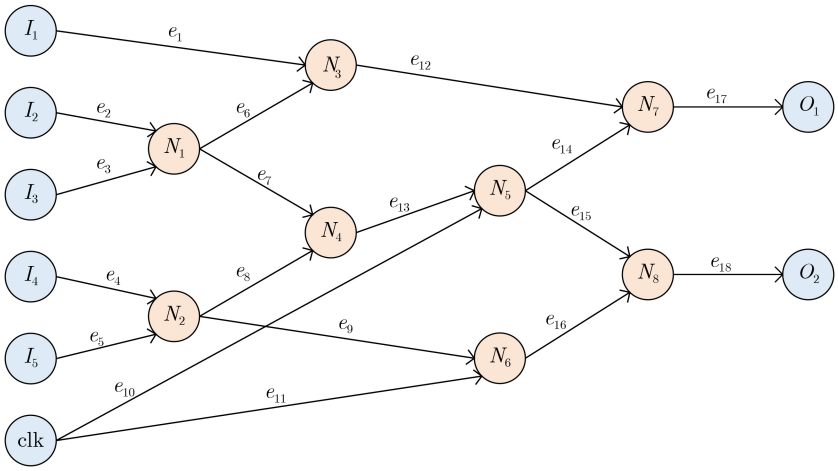



图 3 门级网表的有向图模型

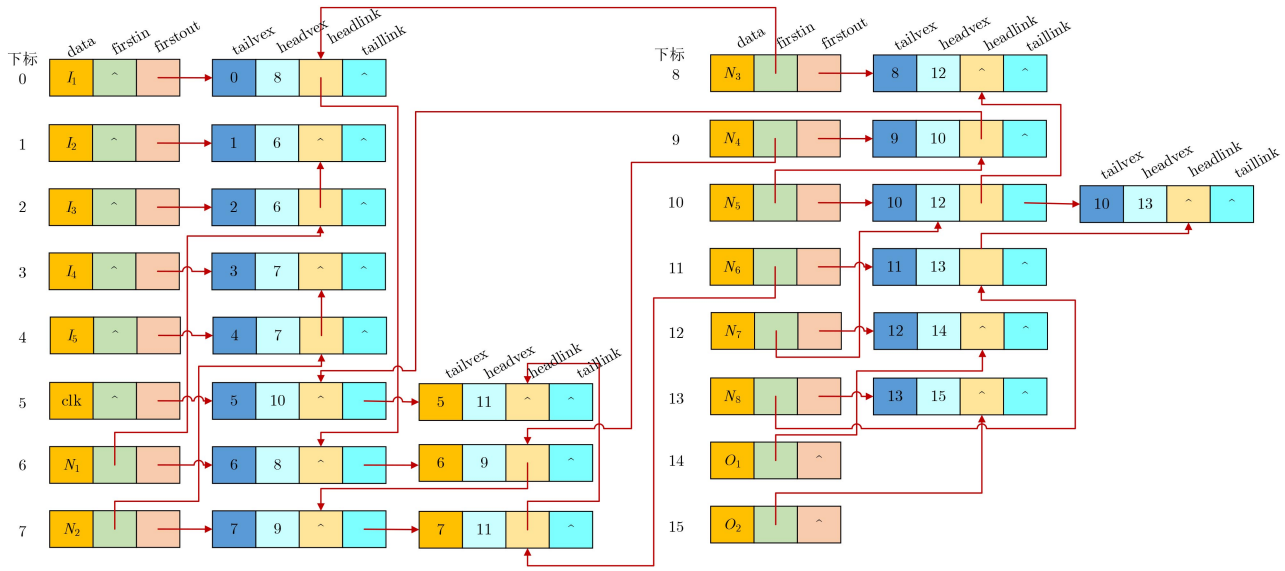
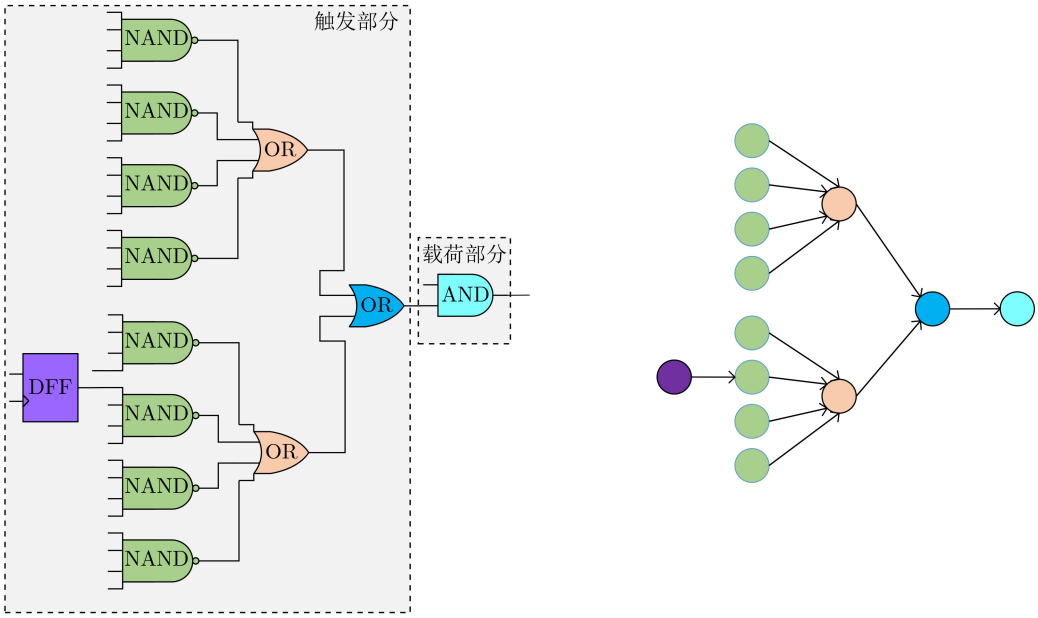



图 4 有向图的十字链表结构



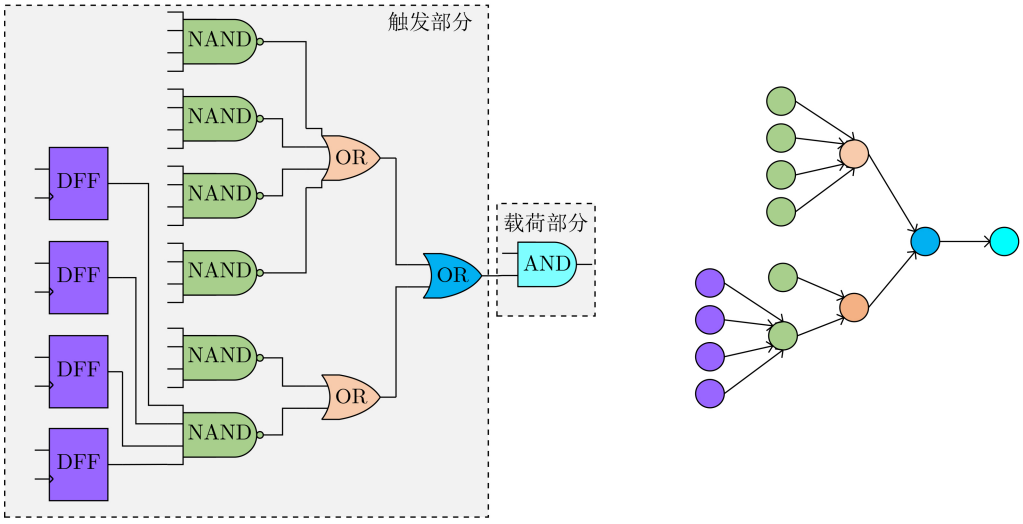

(a) 硬件木马电路　　(b) 有向图模型

图 5 RS232-T1400中的硬件木马电路及其有向图模型

向图，触发逻辑包含4层扇入顶点，且扇入单元数量大于11。本文将从输入方向距离单元$n$ 4层逻辑门的扇入单元总数FAN_IN作为判断硬件木马的结构特征。

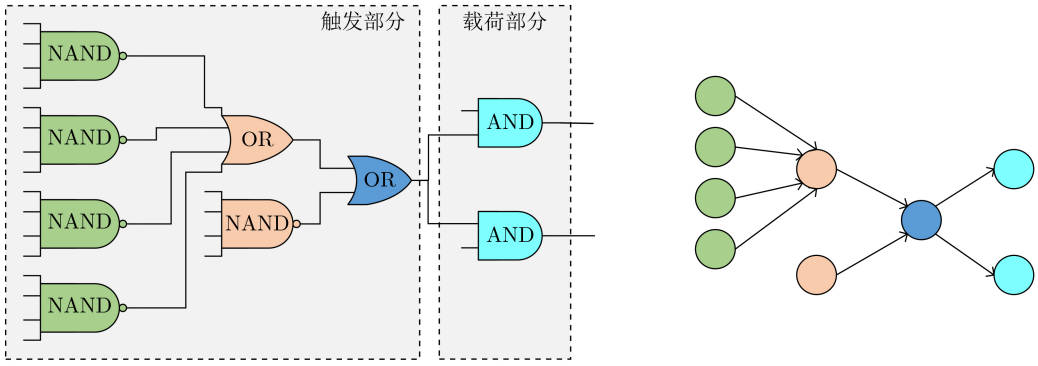
(2)扇入触发器数FF_IN和扇出触发器数FF_OUT。触发器是时序电路的基本单元，由触发器组成的状态机的特定状态转移序列和计数器的计数值均可作为硬件木马的触发条件。图6(a)为RS232-T1200电路中的硬件木马电路部分，该木马电路的触发逻辑是一个时序比较器，当特定状态满足时，硬件木马被激活。触发逻辑的触发器单元较多，本文利用扇入触发器数FF_IN和扇出触发器数FF_OUT来量化触发器单元数量。图6(b)为RS232-T1200的有向图，本文以输入和输出方向距离单元4级逻辑门的触发器单元数目FF_IN和FF_OUT作为判断硬件木马的结构特征。

(3)输入拓扑深度DPI和输出拓扑深度DPO。信息泄露型硬件木马常常复用电路的输出端来泄露母本电路内的关键信息，功能型硬件木马通常监测母本电路的输入端来激活特定序列。基本输入和基本输出或者距离基本输入输出非常近的电路节点可能是硬件木马节点。图7(a)为硬件木马RS232-T1300的结构，硬件木马的有效载荷输出作为母本电路的输出。控制母本电路的输出，当硬件木马被




(a) 硬件木马电路 (b) 有向图模型

图 6 RS232-T1200中的硬件木马电路及其有向图模型




(a) 硬件木马电路 (b) 有向图模型

图 7 RS232-T1300中的硬件木马电路及其有向图模型

触发后，硬件木马的有效载荷控制母本电路的输出，并用来泄露母本电路的私密信息。因此，本文选择基本输入和输出的逻辑单元深度DPI和DPO作为判断硬件木马的结构特征。

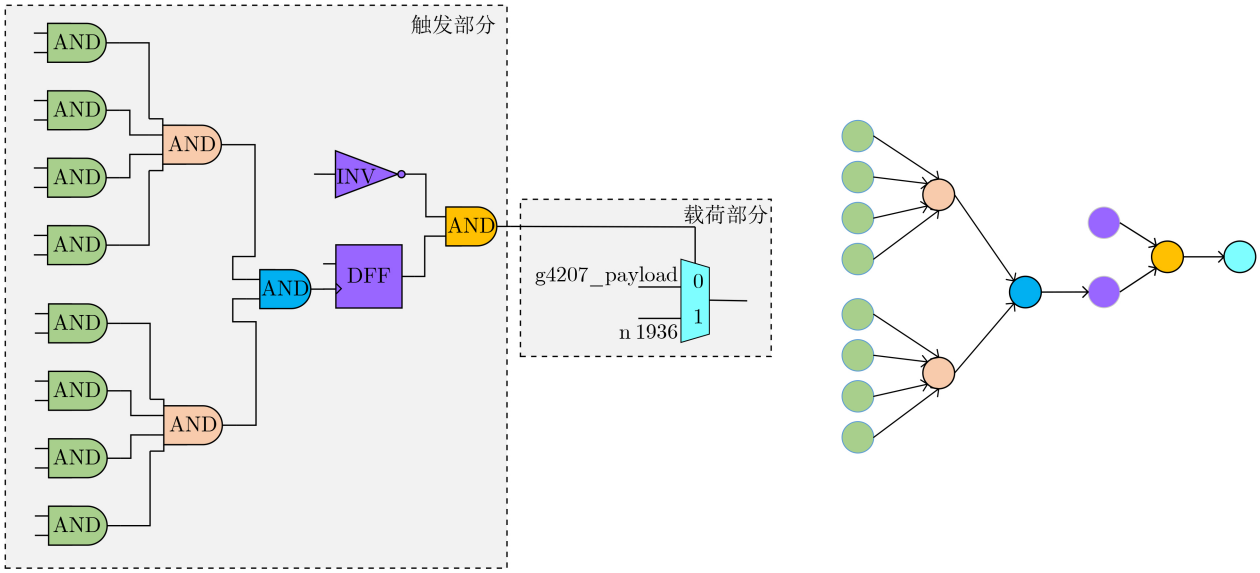
(4)多路选择器数量MUX。为了避免在测试与验证阶段被检测输出，攻击者通常会选择特定输入逻辑序列或者内部状态值作为硬件木马的触发条件。因此，多路选择器在硬件木马触发逻辑中广泛应用，主要用来判断当前状态是否满足其触发条件，触发条件越苛刻，多路选择器数量就越多。图8(a)为硬件木马s15850-T100的结构，当输入序列满足预设值时，硬件木马才激活，并选择内部信号n1936进行输出，从而达到泄露节点n1936状态的目的。本文选择单元$n$前后4级包含的多路选择器的数量作为判断硬件木马的结构特征。

(5)反相器数量INV。对于降低性能型的硬件木马来说，通常选择环形振荡器作为硬件木马的载荷部分。当输入满足硬件木马的触发条件时，植入在关键路径上的有效载荷被激活，导致电路出现时序违例情况。因此，路径上的反相器链可作为硬件木马的结构特征。图9(a)为s35932-T300电路中硬件木马结构的载荷部分，共由20级反向器、3级数选器以及1个与门组成，图9(b)为有向图模型。本文将单元$n$前后4级所包含的反相器数量作为判断硬件木马的结构特征。

基于上述讨论，本文共总结了7种和硬件木马密切相关的结构特征，具体描述如表1所示。

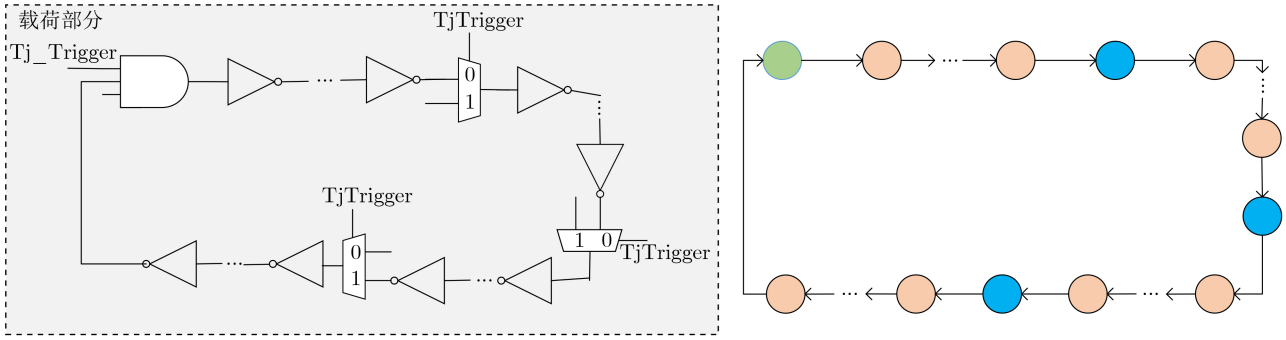
## 4 硬件木马特征提取与识别算法

本文利用门级网表抽象化建模算法将网表映射为有向图，基于表1的描述来计算各个顶点的硬件木马特征值得分，形成7维特征向量。硬件木马逻辑与母本电路逻辑的特征值存在差异，将硬件木马的检测问题转化为二分类问题，利用支持向量机来建立最优的分类平面并识别硬件木马特征，保证木马识别风险最小化和准确率最高。然而在分类器训练过程中，硬件木马的特征集数量远远小于母本电




(a) 硬件木马电路　　(b) 有向图模型

图 8 s15850-T100中的硬件木马电路及其有向图模型




(a) 硬件木马电路　　(b) 有向图模型

图 9 s35932-T300中的硬件木马电路及其有向图模型

**表 1　硬件木马结构特征描述**

| 结构特征 | 具体描述 |
|---|---|
| FAN_IN | 距离单元$n$ 4级逻辑门的扇入单元总数 |
| FF_IN | 从输入方向距离单元$n$ 4级逻辑门的触发器单元数目 |
| FF_OUT | 从输出方向距离单元$n$ 4级逻辑门的触发器单元数目 |
| DPI | 单元$n$距最近的基本输入的距离 |
| DPO | 单元$n$距最近的基本输出的距离 |
| MUX | 单元$n$前后4级包含的多路选择器数量 |
| INV | 单元$n$前后4级包含的反相器数量 |

路，这种不平衡的特征集分布很容易导致建立的最优超平面并不准确，导致分类结果出现较高的误识别率。因此，本文提出了基于SMOTEENN的硬件木马特征扩展算法来扩充木马特征集。

### 4.1 基于广度优先搜索的硬件木马特征得分量化算法

将待测电路网表转换为有向图模型后，依据硬件木马特征描述符对有向图顶点进行特征提取，计算出有向图所有顶点的硬件木马特征得分，具体过程如表2所示。$\boldsymbol{G}$为门级网表的有向图，$n$为$\boldsymbol{G}$中顶点个数，$m$为遍历层数。对于第$i$个顶点$v_i$，利用初始化函数intialize来初始化队列$\boldsymbol{Q}$，入队列函数enquene将该顶点放入到队列$\boldsymbol{Q}$中。当$\boldsymbol{Q}$为非空集合且满足遍历层数条件$(f < m)$时，取出$\boldsymbol{Q}$中的第1个元素$v$，利用函数dequene取出队列$\boldsymbol{Q}$中的第1个元素$v$，并依据有向图结构和相邻边计算函数adjacentEdges获取顶点$v$的相邻顶点集合$\boldsymbol{\Phi}$。相邻边计算首先找到第1条与顶点$v$相邻的边$e_1$，将与$e_1$相连的顶点$w_1$放入集合$\boldsymbol{\Phi}$中，再根据$e_1$的边表指针域找到与顶点$v$相连的下一条边$e_2$，同时将与边$e_2$相连的顶点$w_2$放入集合$\boldsymbol{\Phi}$中，直到顶点$v$的邻接点全部访问完毕，最后将结果放入$\boldsymbol{\Phi}$中。将$\boldsymbol{\Phi}$中顶点$w_1$依次放入到队列$\boldsymbol{Q}$中，当队列$\boldsymbol{Q}$不为空则进入下一轮遍历循环，直至队列$\boldsymbol{Q}$为空或者遍历层数条件不满足，执行特征提取函数$F_j(v_i)$来依次计算输出顶点$v_i$的第$j$个特征得分值，直到所有硬件木马特征计算完毕。

**表 2 基于广度优先搜索的硬件木马特征扩展算法**

输入：$\boldsymbol{G}, n, m$
输出：$\boldsymbol{\psi}$

```
for i ≤ n do
  for j ≤ 7 do
    Q←intialize();
    Q←enqueue(v_i);
    while Q = ∅ and f < m
      v←dequeue(Q);
      Φ←adjacentEdges(G, v);
        for w ∈ Φ do
          Q← enqueue(w);
          Endfor
      Endwhile
      ψ(i, j) ← F_j(v_i);
  Endfor
Endfor
```

### 4.2 基于SMOTEENN的硬件木马特征扩展算法

目前国内外文献仅公开了几十种类型的硬件木马，可建立的硬件木马特征样本非常有限。然而母本电路的规模庞大，特征样本数量较多，导致各类的训练集样本分布不够平衡。数据集的不平衡性可能会造成多数样本所属类的过度拟合，进而影响分类器的性能。在硬件木马检测中，任何可疑节点都不应该忽略，因此需要研究硬件木马特征扩展算法，对木马数据集进行过采样来扩充硬件木马特征集不足的短板，避免木马样本学习的不足。

SMOTEENN为过采样与欠采样相结合的采样技术，生成少数类样本后再利用数据清洗技术删除重叠样本，形成更利于正确分类的平衡数据集[11]。该算法是人工少数类过采样法(Synthetic Minority Oversampling TEchnique, SMOTE)和最近邻(Edited Nearest Neighbor, ENN)算法的结合，先利用SMOTE过采样技术生成新的少数类样本，获得新的数据集，对新数据集中的每一个样本使用K近邻法预测，若预测结果和实际类别标签不同则剔除该样本，最后形成平衡的数据集，将平衡后的数据集应用于分类器的训练，从而建立更加完善的分类模型。因此本文采用SMOTEENN算法对数据集进行预处理，以扩充硬件木马特征集。

### 4.3 基于SVM的硬件木马特征识别算法

支持向量机是一种基于统计学习理论的有监督机器学习算法，可以依据数据的特点建立自适应的分类超平面，相比其他算法来说，准确率更高[12,13]。因此，本文选择SVM算法建立分类模型，将硬件木马的检测问题转换为机器学习中的二分类问题，通过学习已知硬件木马和母本电路的特征向量，建立最优的硬件木马分类器，可以有效识别出硬件木马的特征。

SVM算法分为训练和测试两个过程。在训练过程，通过对已知硬件木马特征库的学习来建立分类模型。首先，根据硬件木马特征库构造特征向量集$\boldsymbol{V}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k,\boldsymbol{x}_{k+1},\cdots,\boldsymbol{x}_n\}$，其中木马特征向量集$\boldsymbol{A}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_k\}$，母本电路特征向量集$\boldsymbol{B}=\{\boldsymbol{x}_{k+1},\boldsymbol{x}_{k+2},\cdots,\boldsymbol{x}_n\}$。支持向量机把分类问题转化为寻找最大间隔超平面，这个最优分类超平面可表示为

$$\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x})+b=0 \tag{1}$$

其中，所有木马特征向量满足

$$\left.\begin{array}{c}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_{k+1})+b\geq 1\\ \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_{k+2})+b\geq 1\\ \vdots\\ \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_{n-1})+b\geq 1\\ \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_{n})+b\geq 1\end{array}\right\} \tag{2}$$

母本电路特征向量满足

$$\left.\begin{array}{c}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_{k+1})+b\leq 1\\ \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_{k+2})+b\leq 1\\ \vdots\\ \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_{n-1})+b\leq 1\\ \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_{n})+b\leq 1\end{array}\right\} \tag{3}$$

$\boldsymbol{\varphi}$称为非线性不可分核函数，当正负样本线性不可分时，该函数可将输入特征向量$\boldsymbol{x}$映射到高维空间，重新转变为线性可分的问题；$\boldsymbol{\omega}$是这个超平面的法向量，$b$是它的偏置截距。寻找最大间隔超平面的过程即是求解式(2)和式(3)的过程，调整$\boldsymbol{\omega}$和$b$的值，以最大化样本点到决策面距离。

同时，考虑到为了满足个别“离群点”的正确分类而对间隔距离造成的影响，本文引入了松弛变量$\xi$和惩罚因子$C$，最终转换为对以下最优化问题的求解

$$\left.\begin{array}{l}\min\left(\dfrac{1}{2}\|\boldsymbol{\omega}\|^2+C\sum\limits_{i=1}^{l}\xi_i\right)\\ y_i(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_i)+b\geq 1-\xi_i\\ \xi_i\geq 0(i=1,2,\cdots,l)\end{array}\right\} \tag{4}$$

其中，$y_i$为特征向量$\boldsymbol{x}_i$对应的样本标签。

在测试过程中，利用训练得到的最优分类超平面来验证待测样本$x_i$的类号$f(\boldsymbol{x}_i)$，决策函数如式(5)所示。若$f(\boldsymbol{x}_i)$为1，则认为待测样本$x_i$为硬件木马节点，否则为母本电路节点。

$$f(\boldsymbol{x}_i)=\mathrm{sgn}(\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{\varphi}(\boldsymbol{x}_i)+b) \tag{5}$$

## 5　实验结果与分析

为了验证本文方法的有效性，本文选择Trust-Hub库中的15种硬件木马开展实验。对该15个测试电路进行统计分析，得到其电路规模以及木马结构和功能信息如表3所示。本文所选测试电路的规模涵盖了几百到几千门，而木马电路仅包含几到几十门，数据不平衡性非常严重，难以完全识别出所有的硬件木马特征，因此需要研究硬件木马特征扩展算法来优化分类器模型。

**表 3　木马电路的具体描述**

| 测试电路 | 电路规模 | 木马单元数量 | 正常单元数量 | 触发电路类型 | 触发概率 | 木马功能 |
|---|---|---|---|---|---|---|
| RS232-T1000 | 242 | 13 | 229 | 组合型 | $3.55\times10^{-13}$ | 改变功能 |
| RS232-T1100 | 244 | 12 | 232 | 时序型 | $3.55\times10^{-13}$ | 改变功能 |
| RS232-T1200 | 243 | 14 | 229 | 时序型 | $5.00\times10^{-11}$ | 改变功能 |
| RS232-T1300 | 240 | 9 | 231 | 时序型 | $8.00\times10^{-10}$ | 改变功能 |
| RS232-T1400 | 242 | 13 | 230 | 时序型 | $5.20\times10^{-15}$ | 改变功能 |
| RS232-T1500 | 243 | 14 | 229 | 时序型 | $3.55\times10^{-13}$ | 改变功能 |
| RS232-T1600 | 241 | 12 | 229 | 时序型 | $5.77\times10^{-9}$ | 改变功能 |
| s15850-T100 | 2432 | 28 | 2404 | 混合型 | – | 拒绝服务，改变功能 |
| s35932-T100 | 5999 | 16 | 5983 | 混合型 | – | 改变功能，泄露信息 |
| s35932-T200 | 5999 | 12 | 5987 | 组合型 | – | 拒绝服务 |
| s35932-T300 | 6019 | 36 | 5983 | 组合型 | – | 拒绝服务，降低性能 |
| s38417-T100 | 5677 | 12 | 5665 | 组合型 | $1.42\times10^{-7}$ | 改变功能，拒绝服务 |
| s38417-T200 | 5680 | 15 | 5665 | 组合型 | $1.66\times10^{-44}$ | 改变功能，拒绝服务 |
| s38417-T300 | 5711 | 46 | 5665 | 混合型 | $1.66\times10^{-44}$ | 改变功能，拒绝服务 |
| s38584-T100 | 7026 | 9 | 7017 | 组合型 | – | 改变功能，拒绝服务 |

本文的主要实现流程如图10所示。首先对门级网表进行分析，将电路图抽象为有向图模型并以十字链表存储有向图。其次，依据有向图模型，提取电路的结构特征，构造表征木马信号的7维特征得分值矩阵。再次，利用SMOTEENN算法平衡数据集，用平衡后的数据集训练SVM分类器，建立最优的分类模型。最后，利用训练好的分类器来验证待测电路是否存在硬件木马信号列表。

为了评估本文所提出的硬件木马检测方法的有效性，本文选取真正类率(True Positive Rate, TPR)、真负类率(True Negative Rate, TNR)和分类准确率(ACCuracy, ACC)这3个常用指标，具体表示如式(6)、式(7)和式(8)所示[14]。其中，TP指被正确识别的木马单元数量，TN指被正确识别的正常单元数量，FP指正常单元被错误识别为木马单元的数量，FN指木马单元被错误识别为正常单元的数量。TPR表示正类样本的分类准确率，即硬件木马的检测率，TPR越高，木马检测效果越好；同理，TNR表示负类样本的分类准确率，即正常单元的检测率，如果TNR过低，说明误把大量正常单元归类为木马单元，导致误判率高；ACC表示所有样本的分类准确率，ACC越高，整体的分类效果越好[15]

$$\mathrm{TPR}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{6}$$

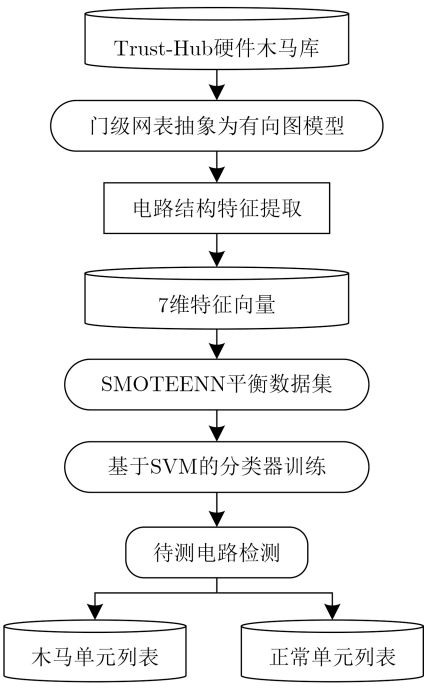



图 10 基于SVM的硬件木马识别流程

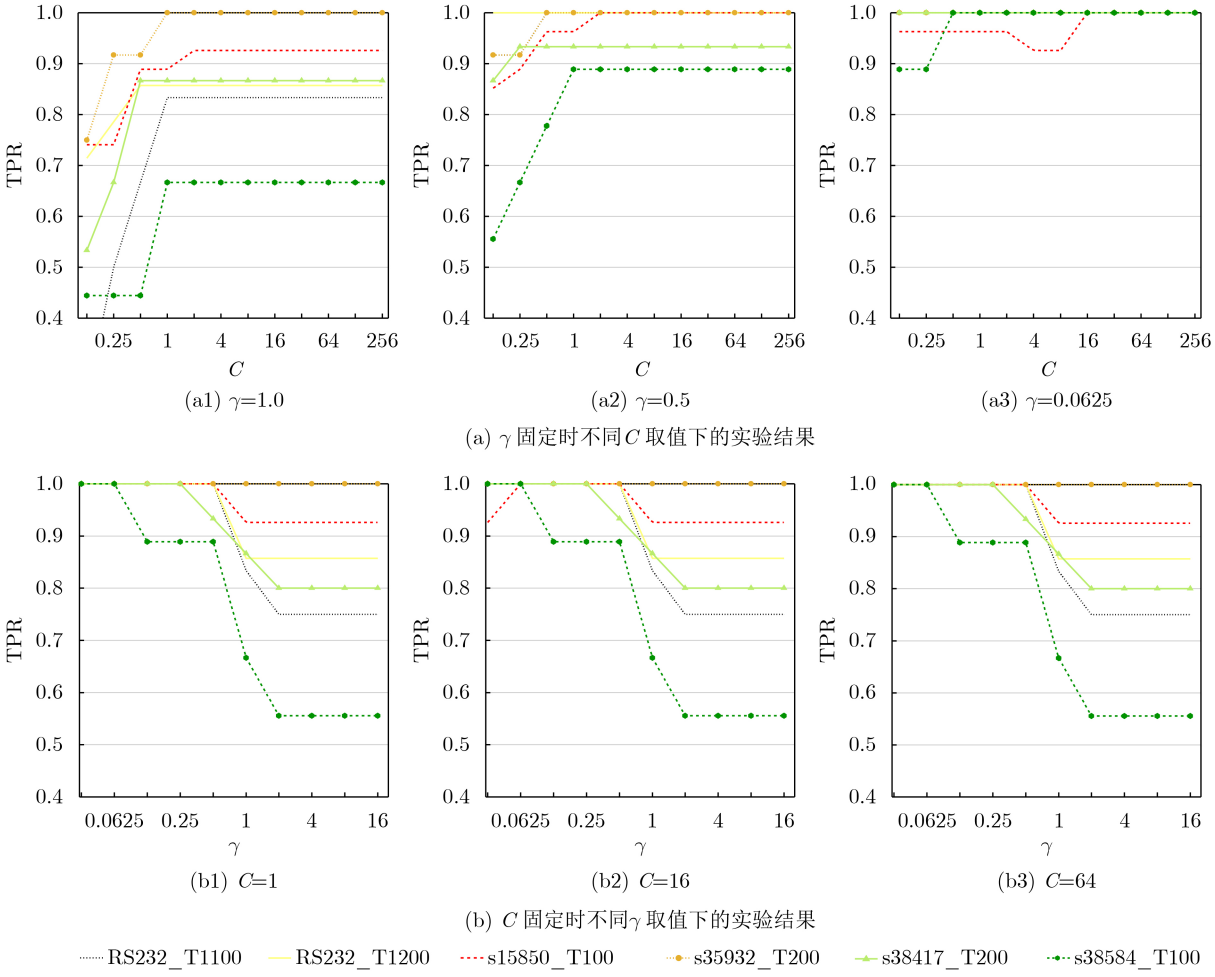
$$\mathrm{TNR}=\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}} \tag{7}$$

$$\mathrm{ACC}=\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \tag{8}$$

在训练SVM分类器的过程中，本文使用的核函数是高斯核函数，该函数自带一个参数$\gamma$, $\gamma$和惩罚因子$C$是需要重点优化的参数，这两个参数直接关系到分类器的性能。遴选SVM参数是一个具有较大工作量的环节，本文按照60%/40%的比例随机划分为训练集和测试集，用训练出的模型对所有电路进行测试，以TPR为目标函数进行参数的调节。本文分别选取12个$C$值和10个$\gamma$值共120个$C-\gamma$组合共进行120次训练，1800次测试，将部分结果展示如图11所示。

图11展示了多个电路在$C$和$\gamma$一方取值固定，一方变化时的实验结果变化规律，由图11(a)可以看出，当$\gamma$取值固定时，从整体上来看，$C$的值越大，分类结果越好；同时由图11(b)可以看出，当$C$取值固定时，$\gamma$的值越小，分类结果越好。为了验证该规律的正确性，将s15850电路的实验数据展示如图12，可以得到同样的结果：当$\gamma$取一固定值时，横向观察各图，可以看出，$C$的值越大，分类结果越好；同时当$C$取一固定值时，纵向观察各图，可以看出，$\gamma$的值越小，分类结果越好。其中$C$是惩罚因子，$C$越高，说明在训练时越不能容忍出现误差，容易导致过拟合；$\gamma$是选择高斯核函数作为核函数后，该函数自带的一个参数，隐含地决定了数据映射到新的特征空间后的分布，$\gamma$值越小，支持向量越多，容易造成平滑效应，影响测试集的准确率。为了使SVM分类模型在得到较好的分类结果的同时具有更强的普适性，本文最终选取$C$=16, $\gamma$=0.0625作为本实验中SVM的训练参数。

在完成硬件木马特征扩展和分类模型参数的选取后，将本文方法应用于测试电路进行实验验证，该实验在个人笔记本电脑(Intel(R) Core(TM) i5-8265U CPU@1.60 GHz, 8 GB RAM)上进行。由于硬件木马特征库的建立和分类器的训练均在前期准备工作中完成，且后续检测未知电路时不需重复执行该项工作，因此该段时间开销不计入总的时间开销，检测效率由待测电路的特征提取时间和分类




(a1) $\gamma$=1.0 (a2) $\gamma$=0.5 (a3) $\gamma$=0.0625

(a) $\gamma$ 固定时不同$C$ 取值下的实验结果

(b1) $C$=1 (b2) $C$=16 (b3) $C$=64

(b) $C$ 固定时不同$\gamma$ 取值下的实验结果

图 11 不同参数下SVM分类器的实验数据

器的分类时间共同决定。本实验中对15个测试电路进行检测，特征提取共用时42.6 min，分类器分类仅耗时4.5 s，即平均每2.845 min即可完成对一个木马电路的检测。最终的实验结果及与文献[16,17]的结果比较如表4所示。

可以看出，本文方法在半数以上电路中取得了90%以上的硬件木马检出率，达到了很好的检测效果。文献[16,17]所提出的硬件木马检测方法在现有的基于特征识别的硬件木马检测方法中处于领先水平，本文与之相比仍具有一定的优势。文献[16]在文献[5]的基础上进行了硬件木马特征的扩充，形成了目前为止较为完善的硬件木马特征集，但是却存在冗余特征过多的问题，平均木马检出率只达到68.32%。与文献[16]相比，本文在小幅牺牲TNR的前提下，将平均硬件木马检出率提升了13.80%。文献[17]同样利用SVM算法构造分类模型，本文TNR和ACC的表现均优于文献[17]所提方法，虽然文献[17]中方法的TPR高于本文，但其构造的分类模型在对不同电路分类时，采用的参数$C$和$\gamma$是不同的，这样做虽然可以提高实验结果，但对不同电路均须寻找最优参数，时间开销较大，并且会导致在检测未知电路时，没有一组固定的参数来进行分类模型的训练，从而难以应用到实际的硬件木马检测问题中。此外，文献[16,17]均未提及检测用时，本文方法执行一次分类任务只需2.845 min，是一种十分高效的硬件木马检测方法。综上所述，本文方法在现有方法基础上进一步提升了硬件木马检出率，检测效率高且具有良好的实际应用价值，是一

**表 4　本文方法实验结果及与现有方法的比较(%)**

| 测试电路 | 文献[16] | | | 文献[17] | | | 本文 | | |
|---|---|---|---|---|---|---|---|---|---|
| | TPR | TNR | ACC | TPR | TNR | ACC | TPR | TNR | ACC |
| RS232_T1000 | 100.00 | 98.90 | 99.06 | 100.00 | 96.77 | 97.08 | 100.00 | 99.50 | 99.59 |
| RS232_T1100 | 50.00 | 98.20 | 92.81 | 100.00 | 97.58 | 97.78 | 100.00 | 99.02 | 99.18 |
| RS232_T1200 | 88.20 | 100.00 | 98.76 | 100.00 | 96.67 | 96.92 | 64.29 | 100.00 | 97.94 |
| RS232_T1300 | 100.00 | 100.00 | 100.00 | 88.89 | 97.64 | 97.06 | 100.00 | 99.02 | 98.75 |
| RS232_T1400 | 97.80 | 99.60 | 99.69 | 91.67 | 97.52 | 96.99 | 92.31 | 98.51 | 98.35 |
| RS232_T1500 | 94.90 | 99.00 | 99.07 | 92.31 | 96.88 | 96.45 | 100.00 | 99.50 | 99.59 |
| RS232_T1600 | 93.10 | 100.00 | 98.44 | 90.00 | 96.24 | 95.80 | 83.33 | 100.00 | 99.17 |
| s15850_T100 | 77.80 | 100.00 | 99.71 | 95.83 | 95.05 | 95.06 | 74.07 | 94.18 | 93.96 |
| s35932_T100 | 73.30 | 100.00 | 99.94 | 91.67 | 100.00 | 95.06 | 100.00 | 98.88 | 98.98 |
| s35932_T200 | 8.30 | 100.00 | 99.83 | 100.00 | 99.46 | 99.50 | 8.33 | 99.41 | 99.28 |
| s35932_T300 | 81.10 | 100.00 | 99.88 | 37.50 | 100.00 | 83.74 | 97.22 | 99.23 | 99.29 |
| s38417_T100 | 33.30 | 100.00 | 99.86 | 91.67 | 97.30 | 97.22 | 33.33 | 99.74 | 99.59 |
| s38417_T200 | 46.70 | 100.00 | 99.86 | 86.67 | 95.64 | 95.50 | 46.67 | 89.99 | 89.88 |
| s38417_T300 | 75.00 | 100.00 | 99.81 | 30.00 | 91.89 | 89.22 | 100.00 | 99.16 | 99.21 |
| s38584_T100 | 5.30 | 100.00 | 99.73 | 87.50 | 86.74 | 86.74 | 66.67 | 82.60 | 82.58 |
| 平均值 | 68.32 | 99.71 | 99.10 | 85.58 | 96.36 | 94.67 | 77.75 | 97.25 | 97.02 |

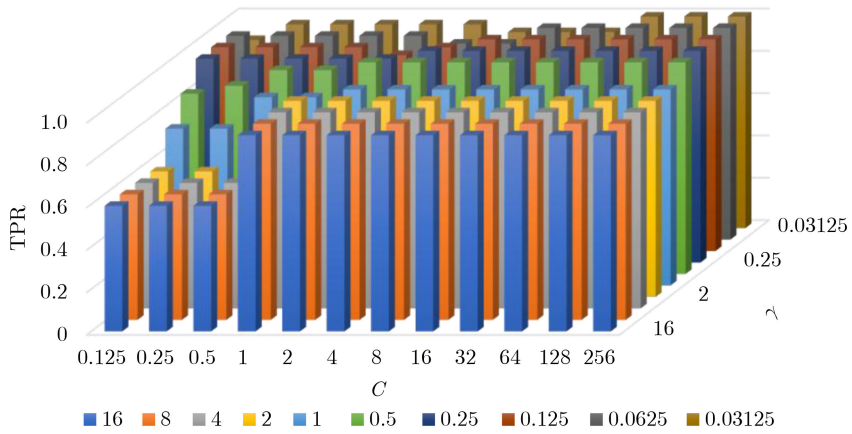



图 12 s15850电路在不同参数下的实验结果

种综合性能更好的硬件木马检测方法。

## 6 结束语

本文提出一种基于有向图结构的电路简化分析模型，并在此基础上提出了基于结构特征的硬件木马检测方法。通过将电路图抽象为有向图，简化电路结构分析过程，提取与硬件木马紧密相关的7维结构特征，利用SVM分类器建立分类模型实现硬件木马检测。由实验结果可知，本文方法在半数以上电路的TPR达到了90%以上，所有测试电路的平均TNR和ACC分别为97.25%和97.02%，实现了超高的准确率，且检测一个木马电路平均仅需2.845 min，具有极高的检测效率。与文献[16]的方法相比，平均硬件木马检出率提高了13.80%，与文献[17]的方法相比更具有实际应用价值，是一个综合性能更好的硬件木马检测方法。今后将继续分析电路结构，挖掘更多与硬件木马相关的结构特征，在现有基础上继续扩充硬件木马特征库，进一步提高硬件木马检测方法的通用性。

## 参 考 文 献


[1] LIU Yanjiang, HE Jiaji, MA Haocheng, *et al*. Golden chip free Trojan detection leveraging probabilistic neural network with genetic algorithm applied in the training phase[J]. *Science China Information Sciences*, 2020, 63(2): 129401. doi: 10.1007/s11432-019-9803-8.

[2] 张伟, 冯建华. IP保护方法研究进展[J]. 微纳电子与智能制造, 2020, 2(1): 95–101. doi: 10.19816/j.cnki.10-1594/tn.2020.01.095.

ZHANG Wei and FENG Jianhua. Research progress on IP protection techniques[J]. *Micro/Nano Electronics and Intelligent Manufacturing*, 2020, 2(1): 95–101. doi: 10.19816/j.cnki.10-1594/tn.2020.01.095.

[3] OYA M, SHI Youhua, YANAGISAWA M, *et al*. A score-based classification method for identifying hardware-Trojans at gate-level netlists[C]. 2015 Design, Automation & Test in Europe Conference & Exhibition, Grenoble, France, 2015: 465–470. doi: 10.7873/DATE.2015.0352.

[4] YAO Song, CHEN Xiaoming, ZHANG Jie, *et al*. FASTrust: Feature analysis for third-party IP trust verification[C]. 2015 IEEE International Test Conference, Anaheim, USA, 2015: 1–10. doi: 10.1109/TEST.2015.7342417.

[5] HASEGAWA K, OYA M, YANAGISAWA M, *et al*. Hardware Trojans classification for gate-level netlists based on machine learning[C]. The 22nd IEEE International Symposium on On-Line Testing and Robust System Design, Sant Feliu de Guixols, Spain, 2016: 203–206. doi: 10.1109/IOLTS.2016.7604700.

[6] CHEN Fuqiang and LIU Qiang. Single-triggered hardware Trojan identification based on gate-level circuit structural characteristics[C]. 2017 IEEE International Symposium on Circuits and Systems, Baltimore, USA, 2017: 1–4. doi: 10.1109/ISCAS.2017.8050673.

[7] LI Chensheng, QIN Xiaowei, XU Xiaodong, *et al*. Scalable graph convolutional networks with fast localized spectral filter for directed graphs[J]. *IEEE Access*, 2020, 8: 105634–105644. doi: 10.1109/ACCESS.2020.2999520.

[8] SAADATNIAKI F, XIN Ran, and KHAN U A. Decentralized optimization over time-varying directed graphs with row and column-stochastic matrices[J]. *IEEE Transactions on Automatic Control*, 2020, 65(11): 4769–4780. doi: 10.1109/TAC.2020.2969721.

[9] 薛春艳. 基于邻接表结构的拓扑排序的全序列算法研究[J]. 现代计算机, 2016(19): 74–76. doi: 10.3969/j.issn.1007-1423.2016.19.018.

XUE Chunyan. Research on the algorithm for all topology sorting based on adjacency list structure[J]. *Modern Computer*, 2016(19): 74–76. doi: 10.3969/j.issn.1007-1423.2016.19.018.

[10] Trust-HUB. Chip-level Trojan benchmarks[EB/OL]. https://www.trust-hub.org/benchmarks/chip-level-trojan.2020.09.

[11] MANJU B R and NAIR A R. Classification of cardiac arrhythmia of 12 lead ECG using combination of SMOTEENN, XGBoost and machine learning algorithms[C]. The 9th International Symposium on Embedded Computing and System Design, Kollam, India, 2019: 1–7. doi: 10.1109/ISED48680.2019.9096244.

[12] 刘东启. 基于支持向量机的不平衡数据分类算法研究[D]. [硕士论文], 浙江大学, 2017.

LIU Dongqi. Support vector machine based classification algorithms research for imbalanced data[D]. [Master dissertation], Zhejiang University, 2017.

[13] 张剑飞, 王真, 崔文升, 等. 一种基于SVM的不平衡数据分类方法研究[J]. 东北师大学报: 自然科学版, 2020, 52(3): 96–104. doi: 10.16163/j.cnki.22-1123/n.2020.03.014.

ZHANG Jianfei, WANG Zhen, CUI Wensheng, *et al*. Research on an unbalanced data classification method based on SVM[J]. *Journal of Northeast Normal University: Natural Science Edition*, 2020, 52(3): 96–104. doi: 10.16163/j.cnki.22-1123/n.2020.03.014.

[14] KOK C H, OOI C Y, MOGHBEL M, *et al*. Classification of Trojan nets based on SCOAP values using supervised learning[C]. 2019 IEEE International Symposium on Circuits and Systems, Sapporo, Japan, 2019: 1–5. doi: 10.1109/ISCAS.2019.8702462.

[15] 魏建安, 黄海松, 康佩栋. 针对不平衡数据的PSO-DEC-IFSVM分类算法[J]. 数据采集与处理, 2019, 34(4): 723–735.

doi: 10.16337/j.1004-9037.2019.04.018.

WEI Jian'an, HUANG Haisong, and KANG Peidong. PSO-DEC-IFSVM classification algorithm for unbalanced data[J]. *Journal of Data Acquisition & Processing*, 2019, 34(4): 723–735. doi: 10.16337/j.1004-9037.2019.04.018.

[16] 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, Baltimore, USA, 2017: 1–4. doi: 10.1109/ISCAS.2017.8050827.

[17] 高良俊, 于金星, 陈鑫, 等. 基于特征提取和SVM的硬件木马检测方法[J]. 微电子学, 2020, 50(6): 914–919. doi: 10.13911/j.cnki.1004-3365.200034.

GAO Liangjun, YU Jinxing, CHEN Xin, *et al*. Hardware Trojan detection method based on feature extraction and SVM[J]. *Microelectronics*, 2020, 50(6): 914–919. doi: 10.13911/j.cnki.1004-3365.200034.

严迎建：男，1973年生，教授，研究方向为安全专用芯片设计技术等.

赵聪慧：女，1995年生，硕士生，研究方向为安全专用芯片设计与防护.

刘燕江：男，1990年生，讲师，研究方向为硬件木马检测、安全专用芯片设计技术等.

责任编辑：余　蓉