基于图注意力网络的门级网表功能识别

秦永旺,张洋*,胡星,刘胜,李少青
(国防科技大学计算机学院先进微处理器芯片与系统重点实验室,湖南 长沙 410071)

摘 要: 随着集成电路设计复杂度的急剧攀升,其呈现出全球化和分工化的发展趋势,需要越来越多的第三方知识产权(IP)核提供者的参与。第三方 IP 核的广泛使用会引入硬件木马,为了检测和评估第三方 IP 核是否存在硬件木马以及硬件木马的功能,迫切需要探索出一种可行的 IP 核硬件安全评估方法,数字电路模块的功能识别作为硬件木马分析的基础研究引起了人们的广泛关注。将电路功能检测任务转换为多分类任务,结合电路结构和图数据结构的特点,提出一种基于图注意力网络(GAT)的门级电路功能分类和检测方法。首先,针对门级网表缺乏功能识别数据集的问题,通过搜集具有代表性的寄存器传输级(RTL)代码并综合生成门级网表,构建一个规模适当、种类多样的门级电路数据集。然后,为了提取和处理电路特征信息,开发了一种基于文本识别的软件工具,将复杂的电路互连结构映射为结构简单的 JSON(JavaScript Object Notation)格式,便于神经网络处理。最后,采用图注意力神经网络,利用构建的门级网表数据集对多分类器进行训练,经过训练后的多分类器能够对未知门级电路进行分类和识别。实验结果表明,该多分类器通过对自建数据集中 6 类共计 3 000 多条网表数据进行学习后,最终对 6 类 645 个网表能够达到 90%的分类正确率。

关键词: 集成电路;电路网表;功能识别;深度学习;图神经网络
中图分类号: TP391 **文献标志码:** A **DOI:** 10.19678/j.issn.1000-3428.0068882

Gate-level Netlist Function Recognition Based on Graph Attention Networks

QIN Yongwang, ZHANG Yang*, HU Xing, LIU Sheng, LI Shaoqing
(Key Laboratory of Advanced Microprocessor Chips and Systems, School of Computer, National University of Defense Technology, Changsha 410071, Hunan, China)

【Abstract】 With the rapid increase in the complexity of integrated circuit design, a trend of globalization and division of labor has emerged, necessitating the involvement of an increasing number of third-party Intellectual Property (IP) core providers. The widespread use of third-party IP cores introduces risks of hardware trojans. To detect and evaluate the presence of hardware trojans and their potential functionalities in third-party IP cores, there is an urgent need to explore feasible hardware security evaluation methods for IP cores. The functional identification of digital circuit modules has garnered significant attention as a fundamental research area in hardware trojan analysis. In this study, the task of circuit function detection is transformed into a multiclassification problem. By leveraging the characteristics of the circuit and graph data structures, a gate-level circuit function classification and detection method based on Graph Attention Networks (GAT) is proposed. First, to address the lack of functional identification datasets for gate-level netlists, a representative set of Register Transfer Level (RTL) codes is collected and synthesized to generate gate-level netlists, thereby constructing a gate-level circuit dataset of appropriate scale and diversity. Subsequently, to extract and process the circuit feature information, a software tool based on text recognition is developed. This tool maps the complex interconnections of circuits into a structured and concise JSON (JavaScript Object Notation) format, thereby facilitating neural network processing. Finally, a graph attention neural network is employed to train a multiclassifier using the constructed gate-level netlist dataset. After training, the multiclassifier becomes capable of classifying and identifying unknown gate-level circuits. The experimental results demonstrate that the classifier, after learning from more than 3 000 netlists in the self-built dataset, achieves a classification accuracy of 90% for 645 netlists across six categories.

【Key words】 integrated circuit; circuit netlist; function recognition; deep learning; graph neural network

0 引言

随着集成电路逻辑复杂性的不断增加,集成电路设计难度持续上升,因此在集成电路设计过程中集成第三方知识产权(IP)核已成为常态。大多数 IP 核提供方为了维持技术壁垒,只提供了 IP 核的接口信号,隐藏了其内部细节。为了确保第三方 IP 核的安全,防止可能的信息泄露[1-3]和攻击[4-6],需要

收稿日期: 2023-11-20 **修回日期:** 2024-01-30
基金项目: 国家自然科学基金重点项目(61832018)。
通信作者 E-mail: * zhangyang@nudt.edu.cn

对 IP 核电路中各个模块的安全性进行检测和分析，而电路网表的功能检测是其中的一项重要工作。然而，一方面由于 IP 核的黑盒特性，传统的基于相似度匹配的方法无法适用；另一方面，随着电路规模越来越庞大，手工验证[7-8]难度不断上升，迫切需要寻求更高效的方法。

在 20 世纪末，随着数字电路设计行业逐渐发展，计算机辅助设计需要解决在大规模电路中寻找子电路的问题。为了在输入的网表中识别和定位子电路模块，最初文献中提出的方法大概分为两种思路：语法等价性验证[9-10]和语义等价性验证[11-12]。

近些年，业内人士越来越关注设计恢复、硬件逆向工程[13-14]和设计安全验证方面的问题。2016 年，COUCH 等[15]提出了一种在复杂器件设计中通过模糊匹配和分割算法来识别 IP 核硬件木马的技术。同年，MEADE 等[14]提出了一种方法来对任意未标注网表中的状态寄存器进行分析和分类，以解决识别过程中将控制逻辑与数据通路混淆而导致的设计功能恢复困难的问题。2017 年，SALMANI 等[16]提出了一种在门级网表中基于可控性可观察性的硬件木马检测(COTD)和恢复技术。

2012 年，SHI 等[17]提出了一种从门级网表中提取功能模块的方法，其使用简洁紧凑的模块库，通过与待检测电路匹配来达到识别功能的目的。2018 年，SILVA 等[18]提出了一种基于卷积神经网络的算术电路分类技术。2021 年，HONG 等[19]选用更适合表示电路网表结构特征的图神经网络[20]，提出了一种电路识别分类技术，此技术中图神经网络从 4 种不同结构的加法器电路网表中提取出各类模块特征，获得了将各类电路模块按照功能进行分类的能力。

SHEN 等[21]将电路网表文件作为文本文件，提出了一种使用自然语言处理(NLP)技术来识别恶意电路的方法，这种方法利用 n-gram 语言模型对电路门序列建模，在区分恶意逻辑与正常逻辑任务中表现出较好的效果。

硬件安全在芯片设计制造中具有至关重要的作用，电路网表识别是硬件安全分析流程中不可或缺的一环。由于当前没有公开的用于功能识别的数据集，本文首先构建了一个电路网表数据集，并将结构特征分析与机器学习算法相融合，实现了电路基本功能模块的检测和识别，然后根据寄存器传输级(RTL)硬件描述语言(HDL)代码生成门级网表文件，通过映射生成 JSON(JavaScript Object Notation)格式数据，最后交由图注意力网络(GAT)进行训练检测，达到功能模块分类的目的。

1 网表数据集构建

通过对学术界常用的包含硬件木马的 Trusthub 库进行调研，并查询公开数据来源的有关内容，发现开源途径中没有适合本研究的功能模块数据集。因此，本研究需要建立一个包含各种不同类型和规模的门级电路数据集，以便对本研究搭建的模型进行训练和对未知网表功能进行识别。

1.1 数据集构建流程和方法

1.1.1 RTL 代码搜集

本文从开源途径搜集基本电路模块的 RTL 代码，包括业界流行的 VerilogHDL 学习网站，比如 HDLbits 等，在开源代码托管平台上搜索硬件基础模块代码，如 GitHub、Gitee 等，从数字电路设计的配套教材中采纳部分模块。按照功能进行分类和筛选后将其作为待处理模块。

1.1.2 代码质量检查

借助芯片前端仿真软件，综合工具和 HDL 语法高亮插件，辅助人工判断，对 RTL HDL 代码进行语法检查和综合性判断。对 RTL 代码进行适当的修改和裁减，人工去除一般性的语法错误。此外，通过添加激励信号，进行仿真验证，确保模块功能的正确性。

1.1.3 综合 RTL 电路代码

使用电子设计自动化(EDA)工具将 RTL 代码综合为门级代码，通过修改约束等限制条件来更改综合条件，调用自动执行脚本再次使用 EDA 工具进行综合，以生成同一类型但具备不同特征的网表，充分扩展门级网表数量，增加数据集的丰富性。脚本执行流程如图 1 所示。

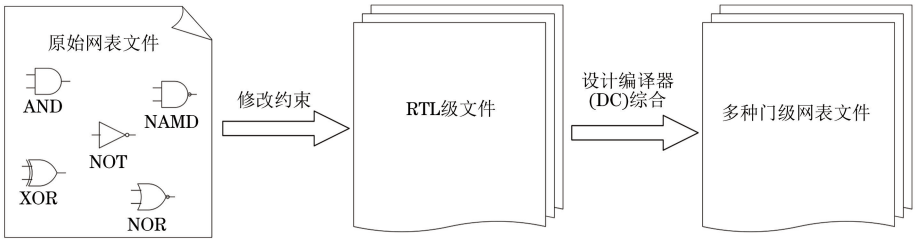

图 1 脚本执行流程

Fig. 1 Process of script execution

1.2 功能模块数据集

基于最常见的组合逻辑功能模块，构建了一个包含加法器、计数器、分频器、循环冗余校验(CRC)算法模块、乘法器和寄存器等多种不同功能模块的门级网表数据集，其中单个硬件功能模块是网表数据集的一个条目。

在每个功能类别中，依据不同实现方式和信号宽度特征进行了细分，使其包含具有不同特征的子类别。这样做一方面可以丰富数据集，另一方面有助于神经网络模型提取同一功能的本质特征，减少冗余结构的干扰。例如，加法器分为 BCD(Binary-Coded Decimal)码加法器、超前进位加法器、旁路进位加法器、选择进位加法器和行波进位加法器共 5 种实现思路。每种加法器依据输入信号的位宽(8 bit 或 16 bit)分为 2 类。乘法器和 CRC 模块子分类思路与之类似。对于更加基础的模块，如计数器和分频器，利用开源模块的多样性，具备不同应用功能，组成不同子类别。数据集中的数据构成如表 1 所示。

表 1 电路模块数据集

Table 1 Circuit module dataset

功能名称	代码种类/种	代码平均大小/Byte	网表种类/种	网表平均大小/Byte
加法器	10	1 615	745	4 984
计数器	5	3 473	515	13 277
CRC 模块	9	1 533	311	9 326
分频器	5	1 233	298	2 430
乘法器	6	2 302	928	19 814
寄存器	16	1 029	576	4 520

2 网表数据预处理

网表文件(.v)本质上依然是代码文本，基于图结构的 GAT 无法直接处理和识别网表数据，需要进一步映射将其转换成图算法能够识别的数据格式。

2.1 数据准备

GTECH 是一个常用的数字电路综合库，在专用集成电路(ASIC)设计中广泛使用。该库包含一系列预先综合好的标准单元库，例如基本逻辑门、多路器、寄存器、计数器、随机存取存储器(RAM)、只读存储器(ROM)等。

本文所用的网表数据集以 GTECH 库为目标库，因此在对网表文件进行解析、识别和转换之前，要先将 GTECH 库中相关的连接信息进行整合，便于工具调用和查询。GTECH 库中的标准单元模块代码具有很多冗余信息且不便调用查询。依据原始模块库生成可供查询的 JSON 文件，该文件中主要包括 GTECH 库中常用基础逻辑门的组成与连接方式，便于格式转换过程中使用。

2.2 JSON 格式

JSON 是一种轻量级的数据交换格式，其基于 JavaScript 语言的数据结构表示方式，被广泛用于 Web 应用程序和数据传输。JSON 格式具有简洁、易读、易编写和易解析等优点，将数据表示为键值对的方式，使用大括号包含对象，使用中括号包含数组。在 JSON 格式中，键名和键值之间使用冒号分隔，键值对之间使用逗号分隔，便于数据的序列化和反序列化。

批量的网表文件映射到一个 JSON 格式的文件中，JSON 文件单个网表表示格式如下：{"link":[[x_1,x_2,…],…],"net":[[n_1, id_1,id_1type],…],"type":"type_id"}，其中：link 表示连接关系，例如[33,64,125]表示数据从索引 64 号门(net[64])流向索引 33 号门(net[33])，从索引 125 号门(net[125])流向索引 33 号门；net 表示线网信号，例如["n74OAOR1",165,3],"n74OAOR1"表示信号名称，165 表示序号，3 表示类型序号；type 表示该网表的功能标签，例如加法器网表的标签设定为 0。

2.3 网表数据映射流程

将门级网表文件转化为 JSON 对象的主要流程如下：首先读入网表内容，然后对网表逻辑门信息进行处理，提取其中的 GTECH 门类型，并更新线网信号前驱后继信息，最后将连接关系列表和节点列表输出，并为功能类型打上对应的标签。转换流程逻辑如图 2 所示(彩色效果见《计算机工程》官网 HTML 版，下同)。

基于上述流程，采用 Qt 应用程序框架开发了一个用于将网表文件转换为 JSON 格式文件的工具，界面如图 3 所示。程序具有批量处理功能，选择输入输出路径后，设置图类型(网表标签)及其他输出设置，点击开始。在右侧输出框中显示运行过程中的流程，在左侧输出框中显示报错等信息。在运行完成后，在输出目录下出现目标文件。

2.4 图数据结构的生成

DGLGraph 是 DGL 库中一种图的基类，可以储存图的节点、边及其特征。门级网表映射到 JSON 格式后，需要从 JSON 格式的数据文件中读取数据并创建 DGLGraph 形式的图。具体来说，对于一个二选一的数据选择器，其网表的电路结构与一个 DGLGraph 相对应，如图 4 所示，虚线框中的部分是 GTECH 库中基本模块，整个电路被基本模块分割

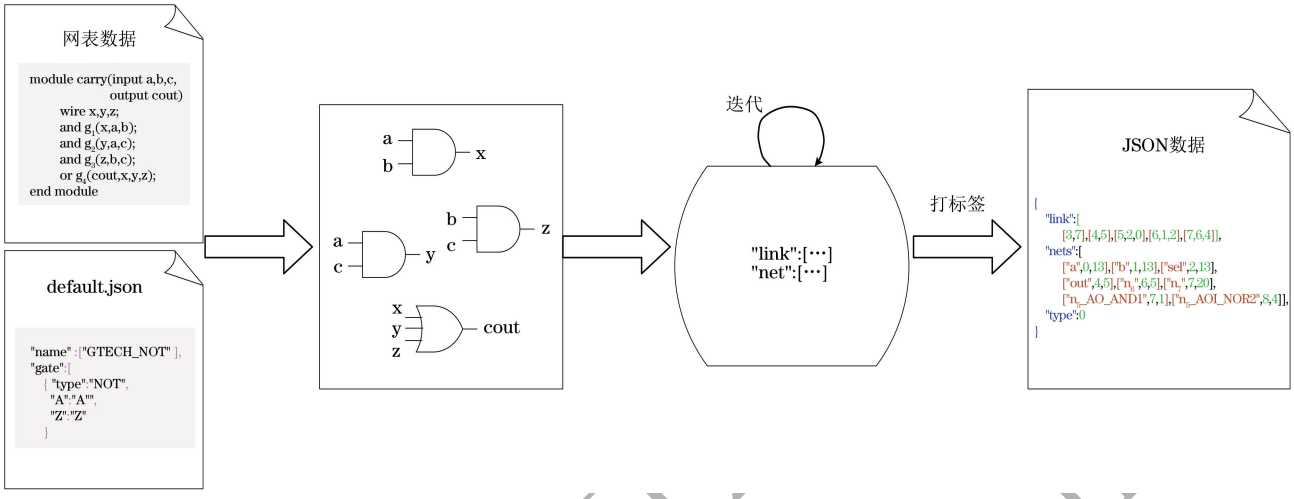

图 2 门级网表到 JSON 格式的映射流程

Fig. 2 Mapping process from gate-level netlist to JSON format

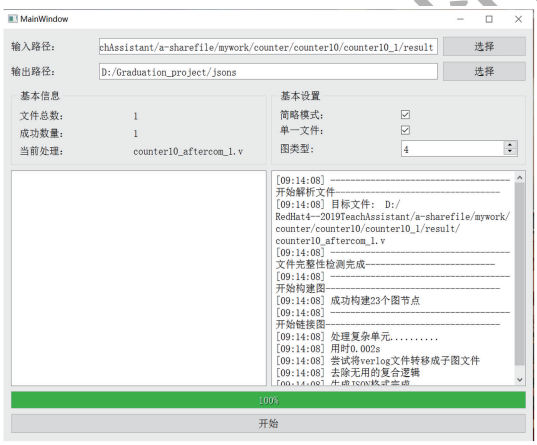

图 3 格式转换工具界面

Fig. 3 Format conversion tool interface

为不同的逻辑节点，例如 GETCH_OR_NOT 与 GTECH_NOT 模块之间的逻辑节点命名为 n_7，此节点对应 DGLGraph 中的同名节点。节点的名称没有特别意义。

节点的特征组成与第 2.2 节中介绍的 JSON 数据结构一致，包括节点的前驱后继的连接信息与基本门电路模块的类型。

在图数据的生成中，在图结构中引入超节点将图中的节点信息进行整合，以便进行图的特征提取和处理。同时，超节点中汇聚着所有节点的信息，使得整张图的特征可以实现更高效的提取和处理。超节点也使得整张图的信息能够实现更好的融合和表达，从而提高了模型的性能和效果。

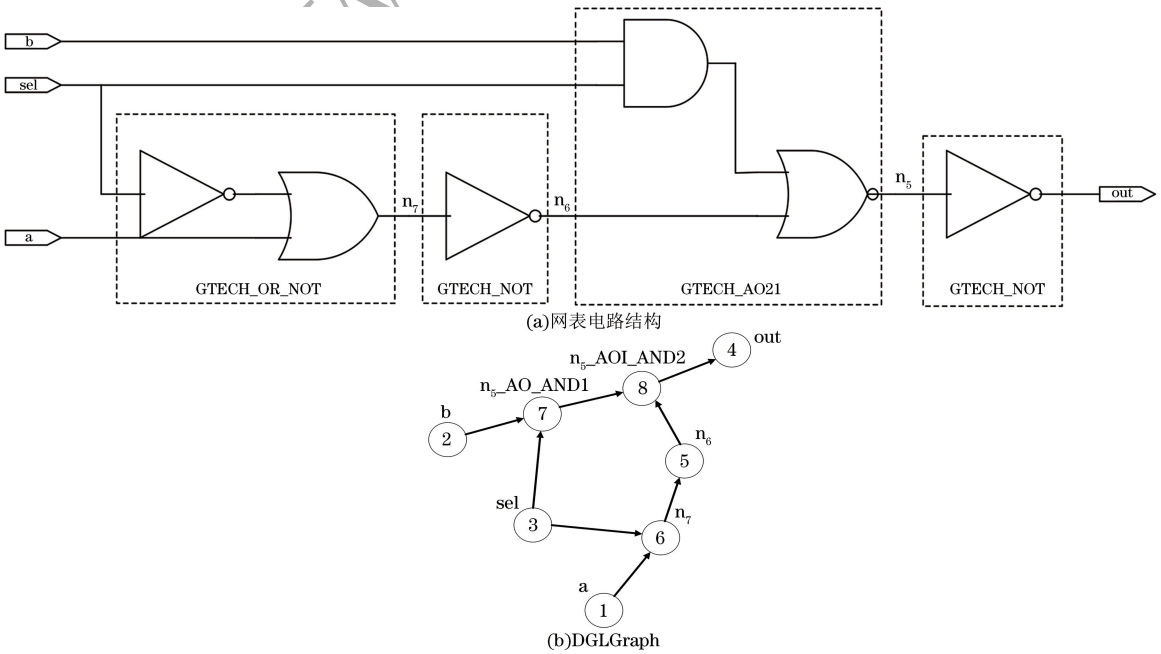

图 4 二选一选择器的网表结构与 DGLGraph

Fig. 4 Netlist structure and DGLGraph of 2-to-1 multiplexer

除此之外，超节点为图神经网络提供了一种有效的降维方法。因为超节点是由整张图的所有节点组成的，所以可以看作是整张图的一种低维表示方式，能够更方便地进行特征提取和处理，同时也使得整张图的信息可以用更少的维度来进行表达，提高计算效率。以上文提到的二选一数据选择器为例，序号为 0 的超节点与其余节点无向连接，聚合所有节点的特征，如图 5 所示。

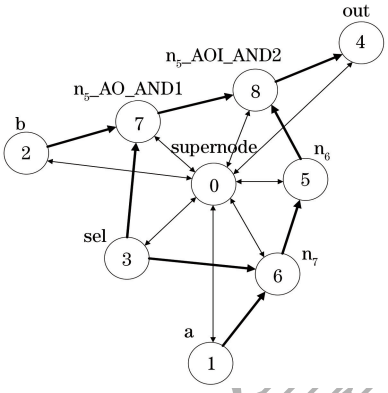

图 5 含超节点的 DGLGraph

Fig. 5 DGLGraph with a supernode

数据集中所有网表所生成的 DGLGraph 规模可以由边和节点的数量定性表示，具体统计数据如表 2 所示。

表 2 不同 DGLGraph 的节点与边数量

Table 2 Number of nodes and edges of different DGLGraphs

单位：个

模块	平均节点数	平均边数
加法器	140	228
计数器	233	574
CRC 模块	157	411
分频器	52	116
乘法器	369	876
寄存器	118	253

3 网表功能分类实现

3.1 GAT 模型设计

GAT[22-23]是一种处理图结构数据的神经网络，通过利用注意力机制对节点之间的关系进行建模。本节所描述的模型基于 PyTorch 和 Deep Graph Library 构建，具体实现包括 GATLayer 和 GateGraph 两个主要组件。

GATLayer 是一个基于图注意力机制的自定义层，接收两个参数，分别是输入节点特征维度和输出节点特征维度。在 GATLayer 中，首先利用一个全连接层将输入特征转换为所需的输出维度，然后将源节点和目标节点的特征连接起来，通过一个全连接层计算注意力权重，最后注意力权重通过激活函数进行非线性变换，得到的权重用于衡量两个节点之间的关系。

在前向传播过程中，首先将节点特征经过全连接层转换后存储在图的节点数据中，然后计算边的注意力权重，并将其存储在边的数据中进行消息传递，并对节点的邻居信息进行聚合，最后根据聚合后的信息更新节点特征。GATLayer 结构如图 6 所示。

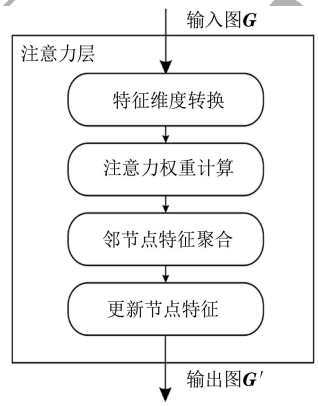

图 6 GATLayer 结构

Fig. 6 GATLayer structure

GateGraph 是一个多层 GAT，包含 3 个 GATLayer 层、1 个 Dropout 层以及 1 个全连接层。在前向传播过程中，首先将图中节点的特征与 Dropout 层进行连接，依次通过 3 个 GATLayer 层，然后对于每个 GATLayer 层，利用其前向传播函数更新节点特征，筛选出超级节点，并将其特征通过全连接层进行分类，最后将分类结果归一化为概率分布。GateGraph 结构如图 7 所示。

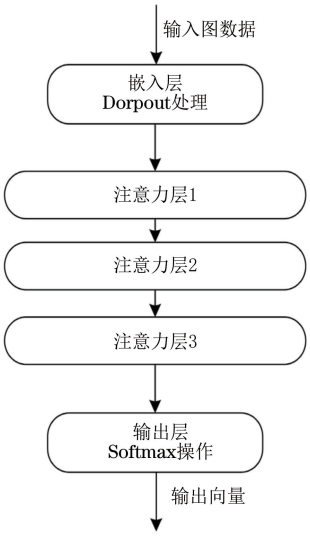

图 7 GateGraph 结构

Fig. 7 GateGraph structure

该模型是基于 PyTorch 框架的图注意力模型。GATLayer 层通过计算每条边的注意力系数来聚合节点信息，使用 Softmax 函数得到每个节点的输出向量。GAT 使用多层感知机作为输出层，对输入进行线性变换和 Softmax 激活函数操作，得到图的分类结果。

3.2 算法复杂度分析

GAT 模型的核心思想是首先将一个节点的邻居节点的特征向量进行自注意力加权，然后将加权结果进行加和得到该节点的表示。具体来说，对于顶点 i（具有特征 $\boldsymbol{h}_i$），首先逐个计算它和邻居节点之间的相似度系数：

$$e_{ij}=\alpha(\boldsymbol{W}\boldsymbol{h}_i,\boldsymbol{W}\boldsymbol{h}_j),\ j\in N_i \tag{1}$$

式中：公共参数 $\boldsymbol{W}$ 对顶点特征向量进行了线性变换，以增强特征辨识度。

对变换后的特征进行拼接，通过参数将拼接后的特征映射到实数上。

然后各个节点借助与其邻居节点的相似度系数，归一化注意力系数：

$$\alpha_{ij}=\frac{\exp(\mathrm{LeakyReLU}(e_{ij}))}{\sum_{k\in N_i}\exp(\mathrm{LeakyReLU}(e_{ik}))} \tag{2}$$

最后通过注意力系数将节点的特征向量进行更新：

$$\boldsymbol{h}'_i=\sigma\left(\sum_{j\in N_i}\alpha_{ij}\boldsymbol{W}\boldsymbol{h}_j\right) \tag{3}$$

在这个过程中，注意力机制可以学习每个邻居节点对当前节点的重要性，因此能够对节点的关联信息进行有针对性的聚合，从而实现更精确的节点表示。具体地，对于给定的一个图，GAT 首先对每个节点及其邻居节点的特征向量进行线性变换得到节点的表示，然后通过注意力机制来计算每个邻居节点对该节点的重要性权重，这个权重是一个实数，可以表示为一个可训练的参数。注意力权重可以通过计算节点特征之间的相似度来得到，例如采用点积、拼接或者加法等方式。在得到邻居节点的注意力权重后，将邻居节点的特征向量乘以它们的权重，并将所有邻居节点的加权结果进行加和，得到该节点的最终表示。$|V|$ 表示图的节点数，$|E|$ 表示图的边数，F 表示原始的特征维度，F' 表示输出的特征维度。由上述计算过程可知，GAT 模型主要包括了两个乘法环节。第 1 个乘法指顶点的特征映射部分，即 $\boldsymbol{W}\boldsymbol{h}_i$ 将 F 维的向量 $\boldsymbol{h}_i$ 映射到 F' 维的空间，$\boldsymbol{W}$ 是 $F\times F'$ 维的参数矩阵，遍历所有的图节点，则计算复杂度为 $O(|V|\times F\times F')$。第 2 个乘法环节为计算注意力系数过程中的 α 映射，α 映射是将 $2\times F'$ 维的向量映射到实数上，遍历图的所有边，则计算复杂度为 $O(|E|\times F')$。因此，GAT 模型的计算复杂度为 $O(|V|\times F\times F')+O(|E|\times F')$。

3.3 模型训练

本文采用 PyTorch 框架，实现 GateGraph 模型的训练流程。在整个训练程序中，导入了多种 Python 库，并利用了自定义模块、数据预处理工具和日志记录工具来跟踪训练的进展。

训练过程主要分为两个阶段：初始化与实际训练。

在初始化阶段，设定了模型的训练参数，包括模型结构、训练数据及相关的超参数。

在实际训练阶段，采用了 Adam 优化器与交叉熵损失函数。通过迭代方式，模型在每个批次中都会更新其参数。在每一轮次中，模型首先预测节点标签，然后计算预测值与实际标签之间的损失，最后利用梯度下降法优化损失，更新模型参数。整个过程不仅跟踪了损失和准确率，而且还记录了训练的时间。所有信息通过日志系统进行记录，方便后期的分析与评估。

实验中的主要训练参数如表 3 所示。

表 3 主要的训练参数

Table 3 Main training parameters

参数名	参数值
批次大小	200
训练轮次/个	1 200
门向量维度	128
学习率	0.000 05

在主要的训练参数中，批次大小、训练轮次和学习率的设定主要参考了同类型工作的典型值，依据数据库中模块的复杂度和电路规模确定门向量维度为 128。

在训练过程中，模型分类准确率从 0.171 8 上升至0.9213，学习效果较好。从图8中可以看出，

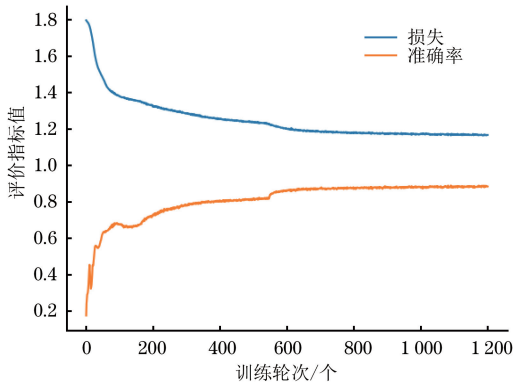

图 8 训练过程中损失和准确率的变化

Fig.8 Changes of loss and accuracy during the training process

前 100 个轮次中损失降低和准确率增长都比较显著。在之后的训练中，变化比较平稳，但损失稳中有降，准确率略有增加。

3.4 模型测试与结果分析

3.4.1 模型评价指标

混淆矩阵是一种常见的评估分类模型性能的工具，它展示了模型预测的类别与实际类别之间的关系。每一行表示实际的类别，而每一列则表示预测的类别。在理想情况下，所有的样本都应该分布在矩阵的对角线上，这意味着所有的预测都是准确的。

基于混淆矩阵，定义真正例(TP)、假正例(FP)、真反例(TN)和假反例(FN)。对于某一类型 X，定义见表 4。

表 4 性能度量参数定义

Table 4 Definition of performance measurement parameters

度量	具体含义
TP(X)	实际为 X 且预测为 X
FP(X)	实际不为 X 而预测为 X
TN(X)	实际为 X 而预测不为 X
FN(X)	实际不为 X 且预测不为 X

由上述 4 个参数可以定义常用模型性能评价指标，即准确率、精准率、召回率和 F1 值。

准确率(A)的定义是预测正确的结果占总样本的概率，即：

$$A=\frac{N_{TP}+N_{TN}}{N_{TP}+N_{TN}+N_{FP}+N_{FN}}\times 100\% \tag{4}$$

精准率(P)又叫查准率，是在所有被预测为正的样本中实际为正的样本的概率，定义如下：

$$P=\frac{N_{TP}}{N_{TP}+N_{FP}}\times 100\% \tag{5}$$

召回率(R)又叫查全率，是在实际为正的样本中被预测为正的样本的概率，定义如下：

$$R=\frac{N_{TP}}{N_{TP}+N_{FN}}\times 100\% \tag{6}$$

F1 值(F_1)同时考虑了查准率和查全率，是对两种度量方式的兼顾，定义如下：

$$F_1=\frac{2\times P\times R}{P+R}\times 100\% \tag{7}$$

3.4.2 实验测试结果

从整个数据集中随机选取了 645 个门级网表电路来测试已训练过的模型性能。测试的 6 类混淆矩阵如图 9 所示。

从给定的混淆矩阵中可以看到，加法器、分频器和乘法器这 3 类电路的预测效果相当出色，其对角线上的值分别为 1.00、1.00 和 0.93，显示了很高的分类准确性。

图 9 测试集上的混淆矩阵

Fig. 9 Confusion matrix on test set

寄存器和计数器这 2 类的电路存在一些预测上的误差。例如，寄存器有 9%的样本被误识别为分频器，而计数器有 10%的样本被误识别为乘法器。

总体而言，本文模型在大多数类别电路上的预测性能都表现得相当不错，但在一些特定类别上，如寄存器、计数器和 CRC 模块，还存在一定的优化空间。模型在测试集上的结果如表 5 所示，其中，每一行表示每一类别的性能指标，包括精确率、召回率、F1 值和该类别的样本数，最后一行的准确率表示本文模型在整个测试集上的分类准确率。

实验结果显示，本文模型在准确率方面取得了较好的结果，整体达到了 90%的准确率，在大多数类别上具有较高的精确率和召回率。这表明本文模型不仅能够正确地将样本分类到相应类别中，而且还能够识别出大部分属于某个类别的样本。此外，本文模型在 F1 值上也取得了较好的结果，这进一步证明了其在多分类任务中的优势。

表 5 分类性能

Table 5 Classification performance

模型	模块	准确率/%	召回率/%	F1 值/%	样本数/个
文献[19]模型	加法器	98	98	98	144
文献[24]模型	加法器	93	无	无	240
本文模型	加法器	99	100	98	144
	寄存器	87	82	92	100
	计数器	84	81	88	100
	分频器	80	100	67	57
	乘法器	93	93	92	180
	CRC 模块	84	75	96	64
	整体	90			645

文献[19]模型对 144 个加法器样本的分类准确率达到了 98%,与本文模型中对于加法器的分类性能持平。文献[24]对加法器电路依据位宽的不同进行了分类,使用图卷积神经网络(GCN)模型对 240 个网表进行分类,准确率为 93%。

本文实验所用的数据无论在数量和复杂度上都比上述工作有较大的提升。文献[19,24]中的实验仅对 4 类不同的加法器进行了分类,本文面向 6 大类共 51 小类的门级电路进行了分类,样本数量和种类大大增加,模型更加复杂。综上所述,本文实现的多分类实验结果更好,模型在准确率、精确率、召回率和 F1 值等评价指标上表现出色,验证了采用 GAT 对门级网表进行多分类的可行性,且 GAT 的计算复杂度与图的规模(节点数、边数)线性相关,面对较大规模的数据有继续研究的意义和优化的潜力。

4 结束语

本文首先搭建了一个规模适当、种类多样的网表数据集,然后依据门级网表的特点,开发出一个数据格式转换工具,将网表数据转换成图数据,最后基于图注意力网络模型,提出了一个多分类深度学习模型,并使用建立的数据集完成训练和测试。实验结果验证了图注意力网络对网表功能进行多分类的可行性,并为未来在硬件安全领域中进行复杂硬件木马电路的功能识别提供了有力的支撑。此外,本文通过引入超节点机制与注意力权重动态分配策略显著提升了模型对电路拓扑特征的捕捉能力,为大规模异构网表的功能识别提供了新思路。未来工作将探索多模态特征融合方法,结合时序分析与功耗侧信道信息,进一步提升硬件木马检测的鲁棒性与可解释性,推动该技术在芯片安全评估中的实际落地。

参考文献

[1] IOKIBE K, HIMURO M, TOYOTA Y. A study for improving signal-to-noise ratio measurement method in side-channel information leakage of cryptographic hardware[C]// Proceedings of the IEEE International Symposium on Electromagnetic Compatibility & Signal/Power Integrity. Washington D. C., USA: IEEE Press, 2022: 294-298.

[2] ALI N A, SHOKRY B, RUMMAN M H, et al. Low-overhead solutions for preventing information leakage due to hardware Trojan horses [C] // Proceedings of the 16th International Conference on Computer Engineering and Systems (ICCES). Washington D. C., USA: IEEE Press, 2021: 1-5.

[3] 于洋, 孙芳芳, 吕华, 等. 基于多尺度时空注意力网络的微表情检测方法[J]. 计算机工程, 2024, 50(6): 228-235.
YU Y, SUN F F, LÜ H, et al. Micro-expression detection method based on multi-scale spatiotemporal attention network[J]. Computer Engineering, 2024, 50(6): 228-235. (in Chinese)

[4] MUKHOPADHYAY D, CHAKRABORTY R. Hardware security: design, threats, and safeguards[M]. New York, USA: Chapman and Hall/CRC, 2014.

[5] 高路尧, 胡长虹, 肖树林. 基于超像素分割的图注意力网络的高光谱图像分类[J]. 吉林大学学报(理学版), 2024, 62(2): 357-0368.
GAO L Y, HU C H, XIAO S L. Hyperspectral image classification based on superpixel segmentation with graph attention networks[J]. Journal of Jilin University (Science Edition), 2024, 62(2): 357-0368. (in Chinese)

[6] TANJIDUR RAHMAN M, ASADIZANJANI N. Failure analysis for hardware assurance and security[J]. EDFA Technical Articles, 2019, 21(3): 16-24.

[7] HARJANI R, RUTENBAR R A, CARLEY L R. A prototype framework for knowledge-based analog circuit synthesis [C] // Proceedings of the 24th ACM/IEEE Conference Proceedings on Design Automation Conference. New York, USA: ACM Press, 1987: 42-49.

[8] WU P H, LIN M P, CHEN T C, et al. A novel analog physical synthesis methodology integrating existent design expertise[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015, 34(2): 199-212.

[9] OHLRICH M, EBELING C, GINTING E, et al. SubGemini: identifying subcircuits using a fast subgraph isomorphism algorithm[C]//Proceedings of the 30th ACM/IEEE Design Automation Conference. Washington D. C., USA: IEEE Press, 2006: 31-37.

[10] RUBANOV N. A high-performance subcircuit recognition method based on the nonlinear graph optimization[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2006, 25(11): 2353-2363.

[11] DOOM T, WHITE J, WOJCIK A, et al. Identifying high-level components in combinational circuits[C]//Proceedings of the 8th Great Lakes Symposium on VLSI. Washington D. C., USA: IEEE Press, 1998: 313-318.

[12] ECKMANN S T, CHISHOLM G. Assigning functional meaning to digital circuits: W-31109-ENG-38[R]. [S. l.]: Argonne National Lab, 1997: 1-10.

[13] CHIKOFSKY E J, CROSS J H. Reverse engineering and design recovery: a taxonomy[J]. IEEE Software, 1990, 7(1): 13-17.

[14] MEADE T, JIN Y E, TEHRANIPOOR M, et al. Gate-level netlist reverse engineering for hardware security: control logic register identification[C] // Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS). Washington D. C., USA: IEEE Press, 2016: 1334-1337.

[15] COUCH J, REILLY E, SCHUYLER M, et al. Functional block identification in circuit design recovery[C]//Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST). Washington D. C., USA: IEEE Press, 2016: 75-78.

[16] 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.

[17] SHI Y Q, GWEE B H, REN Y, et al. Extracting functional modules from flattened gate-level netlist[C]//Proceedings of the International Symposium on Communications and Information Technologies (ISCIT). Washington D. C., USA: IEEE Press, 2012: 538-543.

[18] SILVA L M, ANDRADE F V, FERNANDES A O, et al. Arithmetic circuit classification using convolutional neural networks[C]//Proceedings of the International Joint Conference

on Neural Networks (IJCNN). Washington D. C., USA: IEEE Press, 2018: 1-7.
[19] HONG X N, LIN T, SHI Y Q, et al. ASIC circuit netlist recognition using graph neural network[C]//Proceedings of the IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA). Washington D. C., USA: IEEE Press, 2021: 1-5.
[20] GORI M, MONFARDINI G, SCARSELLI F. A new model for learning in graph domains[C]//Proceedings of the 2005 IEEE International Joint Conference on Neural Networks. Washington D. C., USA: IEEE Press, 2005: 729-734.
[21] SHEN H H, TAN H Z, LI H W, et al. LMDet: a "naturalness" statistical method for hardware Trojan detection[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2018, 26(4): 720-732.
[22] ZHOU J. Graph neural networks: a review of methods and applications[EB/OL]. [2023-10-12]. https://ui.adsabs.harvard.edu/abs/2018arXiv181208434Z.
[23] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[EB/OL]. [2023-10-12]. https://arxiv.org/abs/1710.10903v3.
[24] 刘智毅. 数字电路门级网表通用语义建模与应用[D]. 南昌: 南昌大学, 2023.
LIU Z Y. General semantic modeling and application of digital circuit gate-level netlist[D]. Nanchang: Nanchang University, 2023. (in Chinese)

编辑 陆燕菲