arXiv:2611.29315v1 [cs.SD] 22 Nov 2026

SWIFT：面向实时音频深度伪造检测的手工设计、算术运算极简 FPGA 引擎

Chirantan Banerjee*[1]、Soumita Chatterjee*[2]、Bhargab B. Bhattacharya[2], 和 Hafizur Rahaman[1]

[1]信息技术系，印度工程科学与技术学院（希布普尔），豪拉，西孟加拉邦 711103，印度[2]超大规模集成电路技术学院，印度工程科学与技术学院（希布普尔），豪拉，西孟加拉邦 711103，印度

{chirantanbanerjee1208, csoumita45, bhargab.bhatta}@gmail.com、rahaman_h@yahoo.co.in

摘要

生成式语音克隆威胁着基于语音的身份认证，但当前的音频深度伪造检测器依赖高功耗、高精度模型，不适用于边缘硬件。我们提出了 SWIFT（流式白化整数取证可信引擎），这是一种手工设计的检测器，工作于合成伪迹集中的高频噪声域。一阶整数高通残差驱动九类定点描述符，其中包括整数提升小波倒谱分位数、马氏异常距离、庞加莱相位空间几何特征和相位抖动；随后通过校准的梯度提升树和稀疏 L_1 正则化元学习器进行融合。在采用泄漏受控嵌套交叉验证、并对真实音频和伪造音频使用同一预处理链的条件下，该参考模型在八项基准测试中实现了最高 99.86% 的准确率，等错误率（EER）最低达 0.03%。其整数前端和定点线性分类器运行于 PYNQ-Z2（Zynq-7020）开发板的可编程逻辑上，数据通路延迟为每帧 7.85 μ 秒（127.4k 帧/秒），功耗为 1.734 瓦，LUT 利用率为 11.2%。因此，无需预训练神经网络模型或浮点运算单元，即可在边缘实现纯整数音频深度伪造特征提取。

关键词： 音频深度伪造检测 ·DSP ·FPGA · 整数提升小波变换

1 引言

音频深度伪造尤其危险，因为声音与人类信任、个人身份和安全授权密切相关。如今，只需几秒钟的参考音频即可克隆声音，攻击者利用此类克隆语音绕过生物识别银行验证、实施企业欺诈并传播定向虚假信息。由于听众已无法可靠地区分真实语音与合成语音，自动检测已成为保护数字通信的必要手段。

早期的音频深度伪造基于生成对抗网络（GAN）声码器构建，由 WaveFake 进行基准测试 [1] 和 LibriTTS [2], 以及基于生成对抗网络判别器的检测器 [3], ASVspoof 2021 挑战赛 [4] 是那个时期的主要对策。随着合成技术转向扩散模型以及 VALL-E 等编解码器语言架构，这些针对生成对抗网络调优的检测器泛化能力较差，促使该领域转向 MLAAD 等较新的基准测试 [5], AUDETER [6], 和 ShiftySpeech [7]，这些基准旨在捕捉更广泛的分布偏移。

*共同第一作者。

此后，检测方法日趋多样化。Sundaram 等人 [8] 将手工设计的声学特征与序列模型结合起来，而 Doan 等人 [9] 指出，文本转语音（TTS）生成器很少对其进行建模的呼吸线索，是可靠的活体信号。深度学习检测器包括深度卷积 Transformer [10], 基于梅尔频谱图的轻量级卷积神经网络（CNN） [11], 双重自监督小波 Transformer [12], 以及 HuBERT-ECAPA 嵌入 [13]。近期研究还涉及编解码器感知标注 [14], WavLM 注意力融合 [15], 多阶段框架 [16], 编解码器语言对抗措施 [17, 18], 以及多模态音视频融合 [19, 20]。

尽管在基准测试中表现出色，这些方法大多依赖大型神经编码器，无法在呼叫中心、物联网设备或移动取证套件的嵌入式硬件上运行。迄今为止，深度伪造检测的硬件加速主要针对数据中心 FPGA 卡上的图像和视频深度伪造量化 CNN [21]；据我们所知，尚无手工设计的音频深度伪造检测器被映射到嵌入式 FPGA 上。

已部署的检测器本身也是机器学习系统，因此也面临针对这类系统的威胁，包括对抗性输入 [22], 、机器学习软硬件栈中的木马 [23], ，以及存储权重中的位翻转故障 [24]。相关综述 [25] 和基于加速器性能计数器的运行时防御（包括针对音频输入的防御） [26, 27], 均探讨了这些威胁，因此，紧凑且完全可检查的整数流水线对边缘部署颇具吸引力。

我们提出 SWIFT，这是一种无神经网络检测器，通过校准堆叠集成融合九类数字信号处理（DSP）衍生特征，并在 PYNQ-Z2 FPGA 开发板的可编程逻辑上运行其整数前端。我们的贡献有三方面。第一，提出一种面向整数运算的九类 DSP 描述符，用于表征声学噪声底，并通过 Platt 校准的 LightGBM 成员模型和稀疏 L_1 元学习器进行融合；该方法在八个基准测试中实现最高 99.86% 的准确率，等错误率低至 0.03%，且完全不使用神经网络。第二，在 PYNQ-Z2 上实现前端和线性分类器的位精确定点 FPGA 版本，并明确进行硬件–软件划分；其数据通路延迟为每帧 7.85 μ 秒（127.4k 帧/秒），功耗为 1.734 瓦，LUT 利用率为 11.22%。据我们所知，这是首个手工设计、无神经网络的音频深度伪造检测器 FPGA 实现。第三，采用片段级划分和嵌套交叉验证进行泄漏受控评估，涵盖文本转语音/编解码器、生成对抗网络声码器、ASVspoof 和混合基准测试，并包括变换比较以及在相同特征上与五种标准分类器的对比。

第2 节介绍 SWIFT 及其硬件–软件划分，第3节报告协议、分类器比较、跨域结果以及 PYNQ-Z2 [28] 实现，第4节作出总结。

2 方法

SWIFT 在声学噪声域中运行。声码器、扩散合成器、生成对抗网络和编解码器语言模型会在该域中留下语义特征无法揭示的微弱结构化痕迹。噪声残差可隔离该域，九个互补的 DSP 特征族可捕捉不同的伪影类别，而校准堆叠集成则将这些特征融合起来（图1）。完整集成模型是所有准确率结果所依据的软件参考模型；其整数前端和定点线性分类器构成第2.5节所述的 FPGA 路径。

2.1 预处理与信号标准化

真实音频和深度伪造音频采用相同的预处理方式。每段音频均转换为单声道，峰值归一化至 $[-1, 1]$，并通过第 $f_s = 16$ 节的共享处理链重采样至 kHz3.1，该处理链保留 7 kHz 以下的内容。采用长度为 40 ms、帧移为 20 ms（重叠率为 50%）的滑动窗口，将音频流离散为帧，每帧量化为 Q15定点精度。为抑制

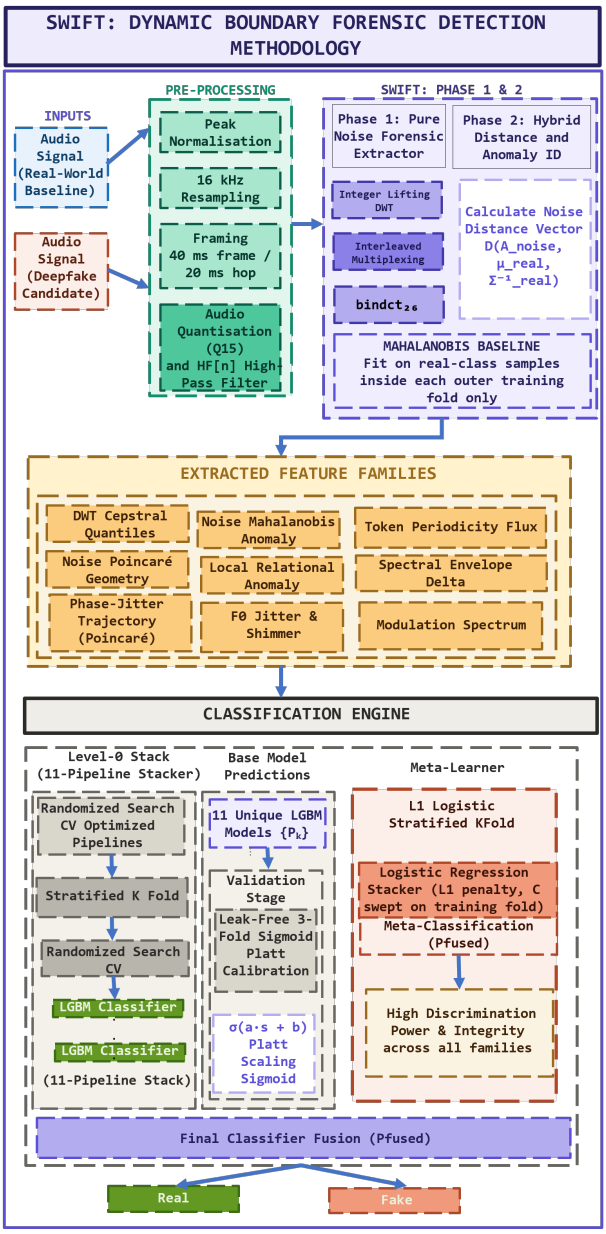

图1：所提出的 SWIFT 流水线框图。

语义语音内容，提取一阶高通残差，具体如下：

$$hf[n] = x_{Q15}[2n+1] - x_{Q15}[2n], \tag{1}$$

因为两个 Q15 采样值之差的范围为 $[-2, 2)$ 。这是一阶差分，其幅频响应为 $2|\sin(\omega/2)|$ ，随后进行 $2\times$ 抽取（即整数Haar提升方案中的预测步骤）。它会衰减承载大部分语音音系内容的低频声道共振峰，并凸显高频噪声底；上采样棋盘格模式、相位匹配失配和神经声码器渲染痕迹等生成伪影会在此处累积。选择一阶差分，是因为它对每个输出样本只需进行一次整数减法，且无需乘法器；全程采用 Q15 算术，可确保软件与硬件结果位精确。

图 2 验证了这一选择。马氏距离 [29] （式（2））衡量一帧与真实语音分布的距离，同时考虑特征相关性。对于噪声残差，真实语音和深度伪造语音形成两个明显分离的峰值，仅在

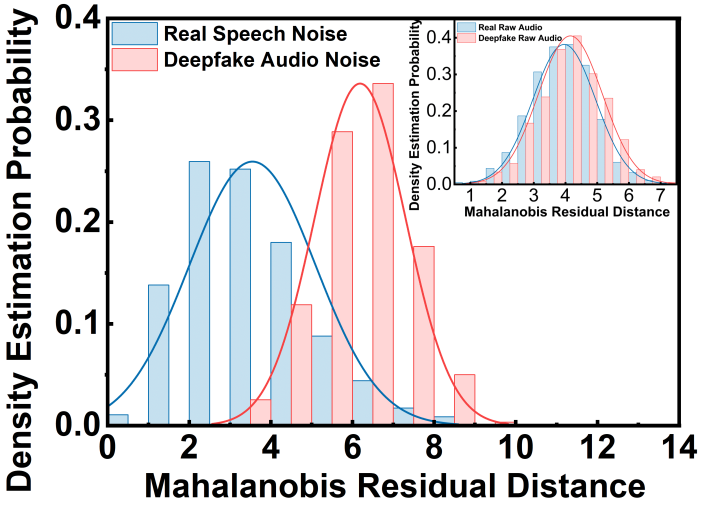

图 2：独立的噪声域马氏距离揭示了真实语音与深度伪造语音之间明显的分离，与原始音频（插图）不同。

距离约为 4 时出现重叠；而在原始音频（插图）中，两者则大幅重叠，因此这种区分源自对残差的提取，而非度量本身。

2.2 整数提升小波与全局异常编码

如图 1 所示，特征提取分为两个阶段。阶段 1 中，使用整数 LeGall 5/3 提升离散小波变换（DWT）[30, 31]以纯整数运算分解每一帧的噪声残差。舍弃第一级的近似分量，并将其细节子带 LH 再次分解为近似分量 HL 和细节 HH。通过非对称交错复用将这些细节子带合并为 $\mathrm{c} = 3\,\mathrm{s}_1 + (\mathrm{s}_2 \gg 1)$，其中 s_1 和 s_2是由 LH 和 HL (s_1)中每四个系数取一个、以及 HH 中偏移量为 0 和 2 的系数构成的 128 样本零填充序列 (s_2)，$\gg$ 表示算术右移。随后使用 BinDCT $_{26}$ [32]，仅通过移位和加法对结果进行压缩，并计算四项统计量（均值、中位数绝对偏差、第 5 和第 95 百分位数）；这些统计量共同构成 104 维（26×4）DWT 倒谱描述符，其理念与基于小波的语音特征 [33]一脉相承。与不规则、由物理过程驱动的声学湍流不同，基于分块的声码器合成会在小波细节子带中留下周期性、有规律的尖峰；整数 DWT 会保留并显露这些伪影。阶段 2 中，仅使用每个训练折内的真实样本拟合真实类基线（$\boldsymbol{\mu}_{\text{real}}, \boldsymbol{\Sigma}_{\text{real}}$），并通过马氏距离为每一帧打分，

$$D_M(\mathbf{x}_t) = \sqrt{(\mathbf{x}_t - \boldsymbol{\mu}_{\text{real}})^\top \boldsymbol{\Sigma}_{\text{real}}^{-1} (\mathbf{x}_t - \boldsymbol{\mu}_{\text{real}})}, \tag{2}$$

其中$\mathbf{x}_t$、$\boldsymbol{\mu}_{\text{real}}$和 $\boldsymbol{\Sigma}_{\text{real}}$ 分别表示帧特征向量、真实类别均值和协方差。$\{D_M(\mathbf{x}_t)\}$ 的十项汇总统计量反映了相对于真实麦克风噪声底的漂移，而当前生成器无法避免这种漂移。

2.3 其余特征模型族

其余七个模型族构成表1中的 192 维描述符。局部关系异常族会根据片段自身的质心为每一帧评分，从而标记那些整体过于平滑或在合成边界处突然爆发的录音。随后，将噪声残差嵌入为庞加莱图 [34]，并计算短轴和长轴上的分布范围

$$SD_1 = \frac{\sigma(y[n+\tau] - y[n])}{\sqrt{2}}, \qquad SD_2 = \frac{\sigma(y[n+\tau] + y[n])}{\sqrt{2}}, \tag{3}$$

表 1：特征族描述

# 特		维度 目	
1	离散小波变换倒谱分位数	104	小波边界不连续性
2	噪声马氏异常	10	分布偏移
3	噪声庞加莱几何	24	相空间扰动
4	噪声相位抖动	10	轨迹微变
5	局部关系异常	10	片段内非平稳性
6	词元周期性通量	8	帧率节律
7	F_0 抖动与微扰	8	韵律微变异
8	频谱包络差分	10	帧间变化
9	调制频谱	8	音节节律
总计		**192**	

其中， $y[n]$ 是样本 n处的噪声残差， τ 是嵌入时滞，而 $\sigma(\cdot)$ 是 n上的标准差。 SD_1 反映短期采样点间不规则性，而 SD_2 跟踪长期吸引子分布范围。自然发声会描绘出丰富且不规则的吸引子，而确定性合成器产生的吸引子则过于平滑或过于紊乱。配套的相位抖动描述符用于跟踪这种几何形态中的帧间偏移。最后四个模型族关注节律和浊音发声。词元周期性通量可揭示与编解码器帧率同步的周期性； F_0 抖动 J 和微扰 S [35]用于捕捉合成器会抹平或夸大的声带微颤；对数频谱差分 ΔS 和 $\Delta^2 S$ 则用于区分过度平滑的过渡与脉冲边界；调制频谱频带比

$$R_{\mathrm{mod}} = \frac{\sum_{f<4} M(f)}{\sum_{4\le f<20} M(f)}, \tag{4}$$

其中， $M(f)$ 是调制频率 f（单位：Hz）处的调制频谱幅度，对比了低于 4 Hz 的缓慢调制能量与 4–20 Hz 频段的调制能量，后者包含自然的 4–8 Hz 音节节律。这些模型族经过精心设计，彼此互补，因此某个模型族忽略的指纹可能会被另一个模型族检测到。

2.4 堆叠分类引擎

192维定点描述符被输入 SWIFT 堆叠分类引擎（算法1），该引擎是一种用于二元法证判定的双层集成模型。在第0层，训练了十一个并行成员流水线：每个特征族对应一个，一个基于经主成分分析（PCA）白化后的完整拼接特征，还有一个基于未经白化的完整拼接特征。每条流水线都遵循相同的四步级联流程：方差阈值筛选、标准化、可选的 PCA 去相关，以及采用极端随机划分的类别平衡 LightGBM 分类器；每个成员均通过 3 折交叉验证进行 Platt 缩放，以生成校准概率。在第1级，将十一个校准概率列组合成一个堆叠矩阵$\mathbf{P} \in \mathbb{R}^{N\times 11}$，其中 N 是样本数量，每列保存一个成员流水线的后验概率，并通过经 L_1正则化的逻辑回归进行融合，

$$\min_{\mathbf{w},b} \; \mathcal{L}_{\mathrm{CE}}(\mathbf{w}, b) + C^{-1}\|\mathbf{w}\|_1, \tag{5}$$

其中， $\mathbf{w} \in \mathbb{R}^{11}$ 为融合权重， b 为偏置项， $\mathcal{L}_{\mathrm{CE}}$ 为交叉熵损失， C^{-1} 为控制稀疏性的正则化强度。该 L_1 惩罚项会自动抑制冗余的成员流水线，仅保留互补性最强的概率流，用于最终的法证判定。协议第 3.1 节所述的外层训练折内拟合所有组件，Level-1 融合器则使用折外 Level-0 概率进行训练。

算法1 SWIFT两级堆叠分类引擎。

要求： 特征矩阵 $\mathbf{F} \in \mathbb{R}^{N\times 192}$，标签 $y \in \{0,\ 1\}^N$
确保： 堆叠矩阵 $\mathbf{P}$；成员模型 $\mathcal{M}$；融合器 g

第0层：训练11条校准成员流水线

$\mathcal{B} \leftarrow 9$ 模型族 $\cup\{\texttt{all},\ \texttt{all_nopca}\}$
对于所有 成员 $m \in \mathcal{B}$，**执行**
  根据方差 $< 10^{-5}$ 剪除特征
  标准化：$\tilde{x}_j = (x_j - \hat{\mu}_j)/(\hat{\sigma}_j + \varepsilon)$
  如果 $m \neq \texttt{all_nopca}$ **则**
    主成分分析白化，保留 99.5% 的方差
  结束条件判断
  拟合 LightGBM$_m$ (极端随机树、路径平滑、类别平衡)
  $\mathcal{M}[m] \leftarrow$ Platt 缩放(LightGBM$_m$，3 折)
  $\mathbf{P}[:\ m] \leftarrow \mathcal{M}[m]$ 在$\mathbf{F}$上的折外概率
结束循环

第1级：稀疏逻辑融合

在$\mathbf{P}$上拟合 L_1 融合器 g：$\min_{\mathbf{w},\ b}\mathcal{L}_{\mathrm{CE}} + C^{-1}\|\mathbf{w}\|_1$
返回 P，$\mathcal{M}$ g

表2：SWIFT 的硬件–软件划分

阶段	运行于	算术运算
噪声残差（式（1）），整数提升离散小波变换、交叉复用，BinDCT $_{26}$	PL	整数加法/移位
线性分类器（SWIFT-HW）	PL	定点乘加
权重加载、DMA、控制、分数读回	处理系统（Cortex-A9）	–
第2–9族	软件	Q15定点
LightGBM、Platt、L_1 融合；全部训练	软件	浮点

2.5 硬件与软件划分及 FPGA 部署

SWIFT 分为两个层级实现，如表 2 所示。第 2.2–2.4 节介绍的完整九类堆叠集成是软件参考模型：该模型在软件中离线训练并进行评估，表 3–6 中的准确率数据均指该模型。FPGA 实现路径 *SWIFT-HW* 则在 PYNQ-Z2 的 Zynq XC7Z020 可编程逻辑（PL）上实现整数前端和定点线性分类器。LightGBM 成员模型、Platt 校准以及 L_1 融合均未映射到 PL。乘累加（MAC）数据通路用于计算点积，而决策树集成需要基于比较器的树遍历，因此 SWIFT-HW 改为使用基于 26 个 BinDCT_{26} 系数离线训练的线性分类器，其权重为 int16，偏置项为 int32。将树集成模型映射到比较器逻辑留待未来工作。

在可编程逻辑（PL）上，Q15帧通过直接存储器访问（DMA）传入，并依次流经式（1）中的残差运算（每对样本进行一次减法）、两级整数提升离散小波变换和交叉复用（仅使用加法器和移位器）、步长为 BinDCT_{26}（移位加法蝶形结构），以及 DSP 切片上的分类器乘累加运算，整个过程不涉及浮点算术运算。ARM Cortex-A9 处理系统（PS）负责加载权重、发起 DMA 传输和控制握手，并读取分数；整个系统由运行实时音频的 Jupyter 笔记本驱动。我们将 IP 核的输出与数据通路的位精确软件模型进行了比较。

表3：噪声域变换比较

变换（针对噪声残差）	准确率 (%)	AUC	EER (%)
短时傅里叶变换	97.50	0.9986	1.67
梅尔频率倒谱系数	75.83	0.8600	23.33
恒定Q变换	56.67	0.6594	40.00
整数提升离散小波变换（本文方法）	**99.78**	**0.9995**	

3 结果与讨论

3.1 实验方案

每段音频片段生成一个 192 维描述符，因为所有帧级量（离散小波变换系数、马氏距离、庞加莱离散度、抖动等）都会汇总为片段级统计量。因此，所有划分均以片段为单位：同一片段中重叠的 40 ms 帧始终处于划分的同一侧，从而避免帧重叠导致训练数据的信息泄漏到测试数据中。我们使用嵌套交叉验证估计性能：外层采用分层五折划分进行评估；在每个外层训练折内，使用内层分层交叉验证进行随机超参数搜索。每个拟合组件（方差阈值、标准化、主成分分析、真实类别马氏距离基线、LightGBM 成员模型、三折 sigmoid 普拉特校准，以及 C 的正则化强度 L_1 融合器）都仅在外层训练折上拟合；Level-1 融合器则使用折外 Level-0 概率进行训练。在每个语料库内，外层各折均按标签分层，因此训练数据和测试数据中会出现相同的生成器；所以下文结果衡量的是已见生成器检测能力，而对未见生成器的泛化能力将在第 4节作为局限性讨论。由于 SWIFT 检查频谱高端部分，真实片段和伪造片段必须采用同一解码和重采样流程，否则检测器可能学到的是存储痕迹，而非合成伪迹。不过，各源语料库在这方面确有差异：在 MLAAD 中，伪造音频为 22.05 kHz WAV 文件，而真实音频（LibriSpeech dev-clean）为 16 kHz FLAC；在 CodecFake [36] 中，真实音频为 32 位浮点格式，而伪造音频为 16 位脉冲编码调制格式，采样率为 16–44.1 kHz。因此，每段音频都经过同一流程：浮点格式音频首先舍入至 16 位；所有音频都通过一个共用的 14 kHz 阶段重采样至 16 kHz，其滤波器具有相同的通带（< 6 kHz）和阻带（> 7kHz）边缘；随后裁去开头和结尾的静音。表 5 中的样本数指音频片段数；CodecFake 将全部 44,242 段真实音频与 50,000 段伪造音频合并。报告的数值根据汇总后的折外预测计算。

指标定义。全文均将伪造类视为正类。对于每个语料库，将融合器在所有外层测试折上得到的折外“伪造”概率汇总为一个分数向量，并据此计算所有指标：曲线下面积是该向量的受试者工作特征曲线下面积，按排序计算；等错误率是同一受试者工作特征曲线上错误接受率与错误拒绝率相等时的工作点；准确率和召回率（伪造类）基于分类器对相同样本的判定计算；F1分数则是两个类别的宏平均（在平衡数据上与准确率相近）。由于受试者工作特征曲线经过 (EER $1-$ EER)，因此每项结果都必须满足 AUC ≥ 1 –EER；评估代码会强制执行此项检查，且所有表格均由相同的分数向量重新生成。各折的指标记录见补充材料。

3.2 在噪声域中采用整数提升离散小波变换的依据

表3 在相同的 SWIFT 流水线下，对噪声残差采用相同处理方式，对比了四种变换。恒定 Q 变换（CQT）的表现最差（准确率为 56.67%，等错误率为 40.00%），因为其按对数间隔排列的频率箱会平滑掉

表4：ShiftySpeech 基准上的分类器比较（使用相同的特征和折；宏平均 F1）

分类器	准确率 (%)	召回率 (%)	F1 分数 (%)	AUC	EER (%)
支持向量机（径向基函数）	89.71	91.06	89.71	0.9685	10.42
多层感知机（3层）	92.96	93.48	92.96	0.9761	7.18
随机森林	82.45	82.78	82.45	0.9070	17.49
XGBoost	88.51	88.98	88.51	0.9559	11.42
LightGBM（单个）	88.62	89.00	88.62	0.9576	11.30
SWIFT（本文方法）	**99.78**	**99.66**	**99.95**	**0.9995**	**0.05**

表5：SWIFT 在文本转语音/编解码器、生成对抗网络声码器、ASVspoof 和混合基准上的性能

数据集	Type	样本	A准确率 (%)	召回率 (%)	F1 分数 (%)	AUC	EER (%)
AUDETER	文本转语音/编解码器	15,228	99.68	99.65	99.79	0.9896	0.29
MLAAD	文本转语音/编解码器	18,000	99.86	99.91	99.84	0.9898	0.12
CodecFake	文本转语音/编解码器	94,242	99.29	99.16	99.29	0.9889	0.66
ShiftySpeech	文本转语音/编解码器	281,945	99.78	99.66	99.95	0.9995	0.05
WaveFake	GAN	104,800	90.42	89.56	90.40	0.9235	1.54
LibriTTS	GAN	20,929	99.52	99.35	99.55	0.9992	0.03
ASVspoof	文本转语音/语音转换	121,461	97.84	97.12	97.48	0.9816	1.81
SONAR	混合	113,122	98.42	97.39	98.37	0.9741	1.27

声码器边界处的高频细微尖峰，而梅尔频率倒谱系数（MFCC，准确率为 75.83%）恰好压缩了伪影累积所在的高频内容。短时傅里叶变换（STFT）的表现具有竞争力（准确率为 97.50%，等错误率为 1.67%），但其单一固定分辨率无法同时捕捉细微的令牌边界尖峰，以及由上采样卷积引起的较宽子带能量变化。多分辨率整数提升离散小波变换兼顾了这两者，准确率达到 99.78%，等错误率为 0.05%。

3.3 分类引擎的合理性说明

使用相同的 192 维特征、外层折和预处理，训练了五种广为人知的分类器：支持向量机（SVM）、多层感知机（MLP）、随机森林、XGBoost 和单个 LightGBM，与 SWIFT 的设置一致（表4）；这些是本文经过匹配的基线模型。它们的准确率均未超过 93%，等错误率也均未低于 7%：随机森林表现最弱（准确率为 82.45%，等错误率为 17.49%），多层感知机表现最佳（准确率为 92.96%，等错误率为 7.18%）。将十一个经过校准的流水线进行堆叠后，SWIFT 的准确率提升至 99.78%，曲线下面积达到 0.9995，等错误率为 0.05%。从单个 LightGBM（准确率为 88.62%）到堆叠模型的跃升表明，没有任何单一模型能够捕捉完整情况。此处未单独分析各个特征族的贡献，将留待未来研究。

3.4 比较性能分析

表 5 涵盖生成对抗网络声码器语料库（WaveFake [1] 和 LibriTTS [2]）、现代文本转语音和编解码器基准（AUDETER [6], MLAAD [5], CodecFake [36], ShiftySpeech [7]）、ASVspoof [4], 以及 SONAR [37], , 后者涵盖九家商业和开源文本转语音提供商。在这四个语料库上，SWIFT 的准确率达到 99.29–99.86%，等错误率为 0.05–0.66%。LibriTTS 的等错误率为 0.03%，而 WaveFake 更难检测（准确率为 90.42%，等错误率为 1.54%），这是由于其多样化的多声码器构成；SONAR 和 ASVspoof 的准确率分别达到 98.42% 和 97.84%。

表 6：与代表性先前系统的比较。先前系统各行的数据摘自所引用论文中针对“基准测试”列所列基准测试的结果；所提出的方法一行展示的是 ShiftySpeech 上的软件参考模型

方法	Type	基准测试	准确率 (%)	召回率 (%)	F1 分数 (%)	AUC	EER (%)
WavLM+MFA [15]	SSL	ASVspoof 2019 LA	91.84	90.92	91.37	0.8470	0.52
XLS-R+CL [16]	SSL	ASVspoof 2019 LA	93.42	92.78	93.10	0.8517	
W2V2-AASIST+SAM [17]	SSL	CodecFake	92.84	91.97	92.53	0.9416	0.62
卷积神经网络+梅尔频率倒谱系数 [18]	CNN	真假分类	89.67	88.95	89.31	0.7597	
所提出的	DSP+堆栈	ShiftySpeech	99.78	99.66	99.95	0.9995	0.05

表 7：SWIFT-HW 在 50兆赫的 Zynq XC7Z020 上的实现

切片查找表		DSPs	可编程逻辑延迟每帧	Throughput (帧/秒)	功耗
5,968 (11.22%)	8,647 (8.13%)	42 (19.09%)	7.85 μ秒	127.4k	1.734 瓦

与先前的系统（表 6）相比，这些系统的指标引自原始论文，因此并非在相同的数据或折上 进行 测量；采用自监督学习（SSL）的方法——使用多融合注意力分类器的 WavLM（WavLM+ MFA） [15], 、采用对比学习的 XLS-R（XLS-R+CL） [16], 以及采用尖锐度感知最小化的 Wav2Vec 2.0 AASIST（W2V2-AASIST+SAM） [17] ——尽管拥有大型预训练编码器，准确率仍停留在 91–93%；而基于梅尔频率倒谱系数的卷积神经网络 [18] 达到 89.67%，这与梅尔滤波器组丢弃高频残差的情况一致。SWIFT 参考模型在没有预训练编码器的情况下达到 99.78% 的准确率（同类比较见表 4），其定点路径运行于 FPGA 硬件上，而该硬件无法承载上述任何方法。

3.5 FPGA 实现结果

该 IP 核占 XC7Z020 的查找表（LUT）资源的 11.22%，以及 DSP 切片的 19.09%（表 7）。在 50兆赫 下，其数据通路延迟为每个 40 毫秒帧 7。85 μ秒，即 127.4k 帧/秒，是 2，500× 20 毫秒帧移所需的 50 帧/秒的 2,548 倍。该指标仅涵盖可编程逻辑数据通路；不包括 DMA 和处理系统的开销。开发板功耗为 1.734 瓦。

已部署分类器的检测性能。 表 8报告了 SWIFT-HW 线性分类器在 ShiftySpeech 基准上的表现，采用第 3.1 节所述协议（相同的外层折、可编程逻辑前端的 26 个 BinDCT_{26} 特征）。浮点线性模型在每个训练折内进行训练，其权重量化为 int16（偏置项为 int32），并使用数据通路的位精确整数模型进行评估，结果与 IP 核输出一致。已部署的分类器准确率为 98.45%，等错误率为 0.42%；与软件集成模型相比，准确率下降了 1.33 个百分点，其中 0.35 个百分点仅由量化造成。这种权衡是将可编程逻辑上的 192 特征树集成模型改为 26 特征基于乘法器的分类器所付出的代价。

4 结论

本文介绍了 SWIFT，这是一种手工设计、无需神经网络的音频深度伪造检测器，将整数提升离散小波变换倒谱变换与八类互补描述符结合起来，并采用两级校准堆叠集成。软件参考实现的准确率最高可达 99.86%（MLAAD），等错误率最低可达 0.03%（LibriTTS）；这些数据与表 6 中列出的自监督学习和卷积神经网络结果相比颇具竞争力（报告结果来自不同的基准测试）

表8：部署在 ShiftySpeech 上的分类器检测性能（相同的折）

分类器	准确率 (%)	AUC	EER (%)
线性，26 个 BinDCT特征，浮点	98.80	0.9921	0.35
SWIFT-HW：线性，int16/int32（已部署）	**98.45**	**0.9904**	
软件参考集成模型（192 个特征）	99.78	0.9995	0.05

无需预训练编码器，其定点前端和线性分类器可在 PYNQ-Z2 上运行，耗时为每帧 7.85 μ 秒，功耗为 1.734 瓦。

仍有四项局限：训练折与测试折中使用了相同的生成器，因此结果衡量的是已见生成器检测；尚未测试对压缩、电话信道和噪声（这些因素会改变 SWIFT 所依赖的噪声底）、对抗攻击以及跨语料库迁移的鲁棒性；只有前端和线性分类器在可编程逻辑上运行，且部署的分类器仅在一个基准测试上进行了评估（表 8），而端到端（DMA 和处理系统）的延迟尚未报告；此外，各特征族的贡献尚未单独分析。未来工作将开展留一特征族消融，在所有基准测试上评估部署的定点分类器，将树集成模型映射到基于比较器的FPGA逻辑上，评估未见生成器、信道退化和跨语言场景，并探索针对新兴文本转语音架构的片上适配 [14]。

致谢

本研究获得**印度国家研究基金会（ANRF）、印度政府**的资助（批准令编号：**ANRF/PMP/2025/000071/ET**，日期为2026年5月20日），并部分得到**印度政府电子与信息技术部**通过**Chips to Startup（C2S）计划**资助（拨款编号：**EE-9/2/2021-R&D-E**。我们感谢**印度工程科学与技术学院（希布普尔）**（西孟加拉邦豪拉）为本研究提供必要的工具和系统支持。

参考文献

[1] Joel Frank 和 Lea Schönherr。WaveFake：一个用于促进音频深度伪造检测的数据集。*arXiv*预印本 *arXiv:2111.02813*，2021年。[2] Heiga Zen、Viet Dang、Rob Clark、Yu Zhang、Ron J Weiss、Ye Jia、Zhifeng Chen 和 Yonghui Wu。LibriTTS：一个由 LibriSpeech 派生的文本转语音语料库。*arXiv*预印本 *arXiv:1904.02882*，2019年。[3] Thien Phuc Doan、Kihun Hong 和 Souhwan Jung。基于 GAN 判别器的音频深度伪造检测。载于深度伪造与廉价伪造的安全影响研讨会论文集，第 29–32 页，2023年。[4] Xuechen Liu、Xin Wang、Md Sahidullah、Jose Patino、Héctor Delgado、Tomi Kinnunen、Massimiliano Todisco、Junichi Yamagishi、Nicholas Evans、Andreas Nautsch 和 Kong Aik Lee。ASVspoof 2021：迈向真实环境中的欺骗语音与深度伪造语音检测。*IEEE/ACM*音频、语音与语言处理汇刊，31：2507–2522，2023年。[5]Nicolas M. Müller、Piotr Kawa、Wei Herng Choong、Edresson Casanova、Eren Gölge、Thorsten Müller、Piotr Syga、Philip Sperl 和 Konstantin Böttinger。MLAAD：多语言音频反欺骗数据集。载于*IJCNN*论文集，第 1–7 页，2024年。

[6] Qizhou Wang、Hanxun Huang、Guansong Pang、Sarah Erfani 和 Christopher Leckie。AUDETER：面向开放世界音频深度伪造检测的大规模数据集。*arXiv*预印本 *arXiv:2509.04345*，2025年。[7] Ashi Garg、Zexin Cai、Lin Zhang、Henry Li Xinyuan、Leibny Paola García-Perera、Kevin Duh、Sanjeev Khudanpur、Matthew Wiesner 和 Nicholas Andrews。ShiftySpeech：一个具有分布偏移的大规模合成语音数据集。*arXiv*预印本 *arXiv:2502.05674*，2025年。[8] Vikram Sundaram、S Babitha 和 Susmitha Vekkot。利用声学特征和深度神经架构进行音频深度伪造检测。载于*ICACC*论文集，第 1–6 页，2024年。[9] Thien-Phuc Doan、Long Nguyen-Vu、Souhwan Jung 和 Kihun Hong。BTS-E：使用呼吸-说话-静音编码器进行音频深度伪造检测。载于*ICASSP*论文集，第 1–5 页，2023年。[10] Vinayak Pandey、Amiradha Satsangi、Bhavya Saini、Shivam Kapoor 和 Preetam Suman。VeriVoice：通过深度可分离频率卷积增强音频深度伪造检测。载于*DECoN*论文集，第 1–6 页，2025年。[11] Ammara Muhammad Shafi 和 Samar Mouti。使用卷积神经网络和梅尔频谱图分析进行深度伪造音频检测。载于*IRASET*论文集，第 1–5 页，2026年。[12] Rui Wang、Zirui Chen、Bo Wang、Zhongjie Ba 和 Kui Ren。AWaveFormer：用于通用音频深度伪造检测的音频小波变换器网络。*IEEE*音频、语音与语言处理汇刊，33：4020–4032，2025年。[13] Gul Tahaoglu。HuBERT衍生的自监督学习特征与 ECAPA-TDNN 匹配，用于稳健的音频深度伪造检测。载于*MLSP*论文集，第 1–6 页，2025年。[14] Yixuan Xiao、Florian Lux、Alejandro Pérez-González-de-Martos 和 Ngo. c Thang Vu。如何标注重合成音频：神经音频编解码器在音频深度伪造检测中的双重作用。载于*ICASSP*论文集，第 1–5 页，2026年。[15] Yinlin Guo、Haofan Huang、Xi Chen、He Zhao 和 Yuehai Wang。利用自监督WavLM和多融合注意力分类器进行音频深度伪造检测。载于*ICASSP*论文集，第 12702–12706 页，2024年。[16] Gautam Siddharth Kashyap、Zohaib Hasan Siddiqui、Mohammad Anas Azeez、Rafiq Ali、Shantanu Kumar、Navin Kamuni 和 Jiechao Gao。愚弄伪造者：用于音频深度伪造检测的多阶段框架。载于*ICASSP*论文集，第 1–5 页，2025年。[17] Yuankun Xie、Yi Lu、Ruibo Fu、Zhengqi Wen、Zhiyong Wang、Jianhua Tao、Xin Qi、Xi- aopeng Wang、Yukun Liu、Haonan Cheng、Long Ye 和 Yi Sun。Codecfake数据集及深度伪造音频通用检测的对策。*IEEE*音频、语音与语言处理汇刊，33：386–400，2025年。[18] Prathamesh Chiddarwar。人工智能生成的深度伪造音频实时检测：一种新方法。载于*ICTBIG*论文集，第 1–5 页，2024年。[19] Anoushka Panja、Debamalya bit Chakravorty、Diganta Saha 和 Arijit Das。利用卷积神经网络、视觉变换器和音视频分析进行深度伪造检测与内容审核的多模态框架。载于*CIACON*论文集，第 1–6 页，2025年。[20] Xiang Li、Zeyu Ma、Zhizhong Zhang 和 Mingang Chen。通过音视频特征融合与时间注意力进行多模态深度伪造检测。载于*ICBAIE*论文集，第 390–393 页，2025年。

[21] Omkar Bhilare、Rahul Singh、Vedant Paranjape、Sravan Chittupalli、Shraddha Suratkar 和 Faruk Kazi。DEEPFAKE CLI：利用 FPGA 加速深度伪造检测。arXiv 预印本 arXiv:2210.14743，2022年。[22] Abdullah Al Nomaan Nafi、Habibur Rahaman、Zafaryab Haider、Tanzim Mahfuz、Fnu Suya、Swarup Bhunia 和 Prabuddha Chakraborty。Dash：用于合成有效且隐蔽的对抗样本的元攻击框架，2026年。[23] Habibur Rahaman、Qipan Xu、Zafaryab Haider、Prabuddha Chakraborty、Swarup Bhunia 和 Fnu Suya。(a)ispy：针对机器学习基础设施的寄生木马，2026年。[24] Habibur Rahaman、Daniel Lobo、Md Rezoan Ferdous 和 Swarup Bhunia。Sanai：嵌入权重中的安全 ECC 分布式密钥奇偶校验，用于保护神经网络免受位翻转攻击，2026年。[25] Habibur Rahaman、Atri Chatterjee 和 Swarup Bhunia。强化人工智能系统：新兴威胁与安全对策。SN Computer Science，7(3):227，2026年2月。[26] Habibur Rahaman、Atri Chatterjee 和 Swarup Bhunia。利用性能计数器运行时检测人工智能加速器中的对抗攻击，2025年。[27] Habibur Rahaman、Atri Chatterjee 和 Swarup Bhunia。Samurai：利用人工智能性能计数器运行时检测人工智能加速器中的攻击。IEEE人工智能电路与系统汇刊，3(5):368–382，2026年。[28] TUL Corporation。PYNQ-Z2 FPGA开发板。https://www.tulembedded.com/fpga/ProductsPYNQ-Z2.html，2026年。访问时间：2026年7月。[29] Jorge Luis Rivero Perez、Bernardete Ribeiro 和 Carlos Morell Perez。用于基于实例的数据流分类的马氏距离度量学习算法。载于 IJCNN 会议论文集，第1857–1862页，2016年。[30] A. R. Calderbank、Ingrid Daubechies、Wim Sweldens 和 Boon-Lock Yeo。将整数映射为整数的小波变换。Applied and Computational Harmonic Analysis，5(3):332–369，1998年。[31] Y. Suhail 和 K.G. Oweiss。用于植入式神经接口设备实时处理的低复杂度整数提升小波模块。载于 IEEE EMBS 会议论文集，第2卷，第4552–4555页，2004年。[32] T.D. Tran。BinDCT：离散余弦变换的快速无乘法器近似。IEEE Signal Processing Letters，7(6):141–144，2000年。[33] Zekeriya Tüfekci 和 John N. Gowdy。利用离散小波变换进行语音识别的特征提取。载于 IEEE SoutheastCon 会议论文集，第116–123页，2000年。[34] Jiahao Huang、Xiang Chen、Hua Jin 和 Jin Li。基于改进的豪斯多夫距离支持向量机的庞加莱图自动分类算法。载于 CISP-BMEI 会议论文集，第1–5页，2019年。[35] Mireia Farrús、Javier Hernando 和 Pascual Ejarque。用于说话人识别的抖动与微扰测量。载于 Interspeech 会议论文集，第778–781页，2007年。[36] Haibin Wu、Yuan Tseng 和 Hung-yi Lee。CodecFake：增强反欺骗模型，以抵御来自基于编解码器的语音合成系统的深度伪造音频。载于 Interspeech 会议论文集，2024年。

[37] Xiang Li、Pin-Yu Chen 和 Wenqi Wei。SONAR：一种合成 AI 音频检测框架和基准测试。*arXiv* 预印本 *arXiv:2410.04324*，2024年。