IMS：面向安全片上系统的智能硬件监控系统

Wadid Foudhaili[1*]，Aykut Rencber[2*]，Anouar Nechi[1],Rainer Buchty[1],Mladen Berekovic[1],Andres Gomez[2*]，以及 Saleh Mulhem[1*]/吕贝克大学计算机工程研究所，吕贝克，德国¨布伦瑞克工业大学计算机与网络工程研究所，布伦瑞克，德国

arXiv:2601.11447v1 [cs.CR] 16 Jan 2026

摘要——**在现代片上系统（SoC）中，高级可扩展接口（AXI）协议存在安全漏洞，可通过协议违规攻击导致部分或完全拒绝服务（DoS）。近期的防护措施缺乏专门的实时协议语义分析，且无法规避协议合规性检查。本文针对这一 AXI 漏洞问题，提出一种用于实时检测 AXI 协议违规的智能硬件监控系统（IMS）。IMS 是一种利用神经网络实现高检测准确率的硬件模块。为训练模型，我们通过报头字段操纵和系统性恶意操作发起拒绝服务攻击，同时记录 AXI 事务以构建训练数据集。随后，我们部署了一种经过量化优化的神经网络，在 <=3% 的延迟开销下，检测准确率达到 98.7%，吞吐量达到每秒 >2.5 百万次推理。之后，我们将该 IMS 作为存储器映射 IP 核集成到 RISC-V 片上系统中，以监控其 AXI 总线。为进行演示并初步评估其后续 ASIC 集成，我们在 AMD Zynq UltraScale+ MPSoC ZCU104 开发板上实现了该 IMS。结果表明，其硬件占用总体较小（查找表（LUTs）占 9.04%、DSP 切片占 0.23%、触发器占 0.70%），且对整体设计可实现频率的影响可以忽略不计。这证明了在资源受限的边缘环境中实现轻量级安全监控的可行性。**

索引词—**AXI 协议安全、硬件安全监控、基于机器学习的监控、片上系统安全、协议层攻击、拒绝服务、RISC-V。**

I. 引言

当代电子设备已无处不在，从智能手机到汽车系统皆是如此。这些设备主要由片上系统（SoC）架构驱动。在这些片上系统中，各个组件——即所谓的知识产权（IP）核——彼此互连，并通过片上总线进行通信。现代 SoC 架构依靠高级可扩展接口（AXI）协议，实现 IP 核之间的高性能通信 [1]。然而，AXI 的设计优先考虑性能和灵活性，而非安保，这无意中导致了安全漏洞和缺陷。

近期的安保分析揭示，AXI互连存在实现缺陷，可利用协议层漏洞发起新型攻击向量 [2], [3]。分析工具如 XRAY 已在某些 AXI 互连中识别出 41 种不同的实现漏洞 [2], [4], ，这表明即使是符合协议的流量，也可能被利用来绕过传统保护机制 [2]。这些漏洞源于实现缺陷，而非协议规范问题，却能助长复杂的攻击 [2], [3]。

A. 片上系统拒绝服务

拒绝服务（DoS）攻击是一种严重的攻击向量，恶意或受损组件会利用实现弱点和协议特性，破坏片上系统组件之间的通信 [5]。在此，部分阻塞会降低某些 IP 的可用性，并可能导致片上系统故障；而完全阻塞则会立即导致片上系统故障。图 1 展示了一种典型的片上系统架构，其中主机 CPU 通过 AXI 总线连接外设和 IP 核。恶意内核和存在缺陷的合法内核都可能触发协议违规，从而导致部分或完全拒绝服务。这类情况可能由格式错误的报头字段引发或通过其实现，例如非法突发长度、事务 ID 重用或信号泛洪。由于这类协议层面的恶意操作发生在片上系统的内部通信互连结构中，外部监控系统无法检测到。因此，片上系统可能继续运行，但性能会下降，进而可能违反服务级别协议或导致完全拒绝服务。

当前的安保对策和机制主要聚焦于访问控制，通过内存保护单元 [6], [7]和互连策略 [5], [8]防止未经授权的内存访问。然而，这些方法无法检测协议违规：恶意组件会利用合法的访问模式，通过操纵报头字段违反协议语义。这凸显了一类新的片上系统安全挑战与漏洞：

C1 利用片上总线协议语义的新型攻击向量。C2 现有保护机制无法检测或缓解此类底层实时威胁。

B. 论文贡献

本文提出了一种新颖的方法，利用机器学习（ML）检测并缓解针对片上系统 AXI总线的拒绝服务攻击。我们通过以下主要贡献推动了该领域的最新进展：

* 这些作者贡献相同。本研究部分由德国联邦研究、技术与航天部（BMFTR）通过 RILKOSAN 项目资助。本文已被欧洲设计、自动化与测试会议（DATE）2026 接收发表。

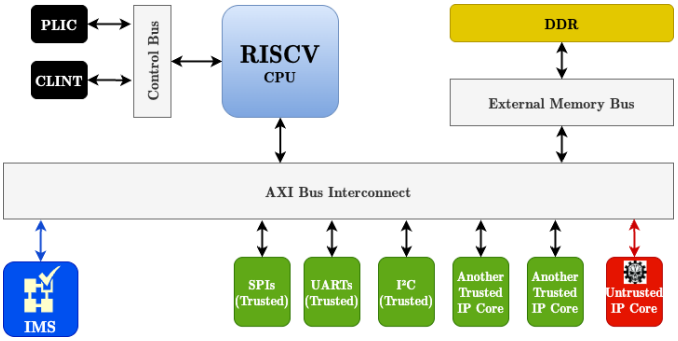


图 1：AXI 总线硬件监控系统概念

- 我们提出了一种名为智能硬件监控系统（IMS）的新型安保对策，用于防御可能导致片上系统部分或完全拒绝服务的严重 AXI 协议违规攻击。
- 我们展示了如何将 IMS 构建为机器学习模型。因此，我们首先生成学习数据集，并通过应用多种机器学习优化技术，提出一种经过充分优化的硬件友好型机器学习模型。
- 我们提出的 IMS 实现了 98.7% 的高检测率，适合在现实环境中部署。为此，我们在一款 RISC-V 片上系统中实现了 IMS，并展示集成 IMS 所需的硬件开销。

我们公开了用于检测 AXI 总线头部攻击的数据集，以便复现本文结果，并解决片上系统环境下协议级安全研究缺少公开数据集和基准测试的问题。

II. 背景与动机

本节介绍我们的威胁模型，并概述相关的最先进应对措施。随后，我们阐述提出该解决方案的动机。

A. 威胁模型

片上系统安全验证涉及以下安全原则：机密性、完整性、可用性和真实性 [5]。由此可归纳出三类主要威胁 [5]: **(i)可用性违规：**恶意或发生故障的组件/IP 核可能导致合法用户无法使用共享资源，**(ii) 机密性泄露：**主要涉及对敏感数据的未经授权访问，以及 **(iii)完整性受损：**不可信组件试图修改关键系统状态。

在这项工作中，我们重点关注可用性违规威胁。这类威胁是片上系统设计中一个紧迫而关键的问题，因为它们可能导致整个系统立即发生故障，并且可通过性能下降轻易观察到 [2][4]。虽然在全面的安全框架中，机密性和完整性违规同样重要，但可用性攻击往往是系统遭到入侵的首要迹象，并会直接影响运行功能。

因此，本文提出的威胁模型考虑了这样一种攻击者：其可能是恶意 IP 核，也可能利用发生故障的 IP 来破坏 AXI 总线基础设施上共享资源的可用性。该威胁模型扩展了 [8]，中已确立的模型，该模型侧重于 AXI 总线协议的事务级 [5], [8]。

B. 基于硬件的安全监控

当前面向片上系统的硬件安全监控方法采用两种主要策略：**内存保护单元（MPU）**和**访问控制策略（ACP）**。基于 MPU 的监控方法对特定存储器区域实施访问边界，确保只有经过授权的组件才能对其进行读写 [6],；基于 ACP 的监控方法则主要使用基于地址的过滤来定义组件/IP模块之间的通信规则 [5], [8]。然而，这些解决方案侧重于防止未经授权的访问，而非检测协议级语义违规。因此，它们存在根本性局限：依赖静态威胁模型，无法适应不断演变的攻击模式，也缺乏区分合法事务与符合协议的恶意行为的智能 [9], [10]。

C. 研究差距与动机

现有解决方案主要关注访问控制和内存保护，但缺乏专门的实时协议语义分析 [3]。虽然静态验证工具能够识别设计时漏洞，但无法应对规避协议合规性检查的运行时攻击。这表明存在一个关键的安全缺口，当前对策无法克服这一缺口。因此，我们引入智能硬件监控系统（IMS），部署机器学习模型来监控 AXI。机器学习已被广泛探索和研究，用于监控网络事务 [11], 、设备操作以及 CPU执行 [12]。我们的工作通过引入协议感知机器学习模型来弥补这一缺口，该模型以轻量级硬件引擎的形式部署，用于持续进行片上监控。据我们所知，现有方法中尚无一种能够在资源受限环境中利用机器学习进行实时 AXI 协议监控。

III. IMS设计方法与实现

本节介绍开发智能 IMS 的方法，用于实时 AXI 协议监控和安保分析。我们的方法包括数据集生成、预处理流水线设计、机器学习模型优化和评估指标。

A. IMS 设计步骤

我们提出的工作的核心思路是设计一种用于监控 AXI 总线流量的智能硬件监控系统（IMS）。IMS 作为被动式硬件监控器，策略性地部署在 AXI 总线互连中，观察系统组件之间的事务模式，同时不干扰正常运行。该系统架构以一个.....为核心



图 2：IMS 设计步骤概览

片上系统（SoC）通过分层总线基础设施与系统资源进行交互，多个设备（IP 核）通过 AXI 协议竞争总线访问，因此值得采用 AXI 互连。不可信组件或 IP 核（例如恶意主设备或发生故障的设备）可能扰乱总线运行，进而导致系统级停滞以及部分或完全拒绝服务（DoS）。因此，我们的目标是设计一种 IMS，能够实时监控 AXI 总线流量并分析 AXI 协议。图 2 展示了 IMS 设计步骤。首先，我们通过记录 AXI 总线流量中的事务来构建训练数据集。该数据集包含正常数据和恶意数据。具体而言，恶意数据指通过违反 AXI 协议头字段级规定或发动资源耗尽攻击所表现出的任何异常行为。随后，我们对数据集应用特征工程方法，以便在训练过程中使用最具代表性的特征。我们采用机器学习模型优化技术，并生成该机器学习模型的 RTL 代码，使其能够作为适用于任何 SoC 的 IMS。

B. 数据集构建与攻击合成

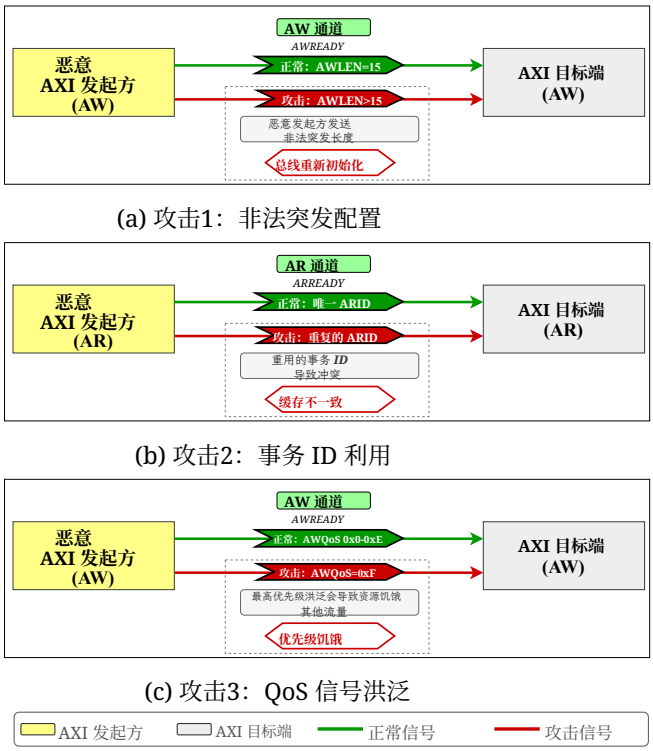
AXI 包含五个通道：写地址（AW）、写数据（W）、写响应（B）、读地址（AR）和读数据（R），异常可能出现在任意通道中。由于缺乏公开可用的 AXI 安全数据集，因此有必要创建全面的训练语料。我们针对具备标准 AXI4 互连能力的 RISC-V 片上系统，采用 Chipyard 平台 [13] 开发了双模式（**正常**和**恶意操作**）数据收集策略。该片上系统包含多个控制器（处理器、DMA 控制器、硬件加速器），它们作为 AXI 主设备（发起端），竞争访问由 AXI 互连的仲裁逻辑协调的共享外设（例如存储器、输入/输出）。

正常运行捕获：在标准 Linux 操作系统运行时，我们收集了 16,383 笔合法事务，确保全面覆盖五个 AXI 通道（AW、W、R、B、AR）上的典型系统行为。该基线捕获了生产环境片上系统中预期的自然事务模式。正常事务是在系统运行期间捕获的，此时 AXI 各通道中的事务不受任何干扰。

恶意操作合成：为记录恶意操作，我们在 AXI 协议中引入恶意行为，涵盖了多种攻击场景：恶意或发生故障的主设备（发起方）IP/组件未完成写事务，从而阻塞总线，导致系统级停滞和硬件级拒绝服务。我们系统地生成了 3,242 个攻击样本，代表三种关键攻击场景：

1) 非法突发配置 *[图3a]*：AWLEN 值超过 15，迫使总线重新初始化 2)事务 *ID* 利用 *[图3b]*：重复 ARID 值导致缓存不一致 3)*QoS* 信号洪泛 *[图3c]*：AWQOS 饱和（0xF），使低优先级流量挨饿

这些攻击会导致目标片上系统遭受拒绝服务。



(a) 攻击1：非法突发配置

(b) 攻击2：事务 ID 利用

(c) 攻击3：QoS 信号洪泛

图3：安全漏洞与攻击向量的 AXI 协议示例

例如，可以注入如下恶意事务：

发起方在 AW 通道收到 AWVALID 后，我们将 W 通道中的 WDATA 保持不变。这些攻击向量模拟了恶意发起端阻塞总线事务、导致系统级停滞和可用性拒绝服务攻击的硬件级场景。

我们的检测方案采用 Vivado 的集成逻辑分析仪（ILA）[14]，在 RISC-V 发起方和 AXI 目标端/设备接口处策略性地布置探针，以捕获所有相关 AXI 信号。通过执行所提出的三种攻击场景，我们捕获了 19,625 个事务（原始样本），其中包括 16,383 个正常样本和 3,242 个恶意样本，为稳健的机器学习训练提供了充足数据。我们以三种格式导出捕获的数据：原始格式、VCD 和 CSV 文件，其中原始格式用于硬件检测，VCD 用于数据集生成，CSV 格式用于机器学习模型训练。原始转储文件包含对应于 57 个特征的 290 个信号，其最简单的表示形式为比特。

C. 预处理流程与特征工程

我们的特征工程方法通过系统化的三阶段流程，将原始协议信号转换为适用于机器学习的表示；该流程旨在最大限度地提高信息含量，同时尽量降低计算开销。

(1)**阶段 1 - 信号编码：**我们依据标准化的 AXI 协议规范，将二进制和十六进制报头字段解码为十进制表示。移除 5 个不属于 AXI4 标准的 ILA 专用调试信号后，剩下 52 个与协议相关的特征，供后续处理。(2) **阶段 2 - 相关性分析：**我们采用统计相关性分析，识别并剔除样本中高度相互关联或取值恒定的冗余特征。此过程将特征空间从 52 维缩减至 22 维，减少量达 58%，同时保留了关键的判别信息。

(3)**阶段 3 - 维度优化：**主成分分析（PCA）将剩余特征压缩为低维表示，同时保留统计显著性。我们的迭代方法实现了大幅压缩：原始特征组合成 4 个组件时可保留 90% 的方差，6 个组件可保留 95%，而 8 个组件则能保留原始信息量的 97%。

反复应用相关性分析和 PCA，直至实现最佳特征减少量，同时维持最低可接受方差阈值（90–97%），确保在压缩后的特征空间中仍能检测到关键攻击特征。

表 I：数据集统计信息与预处理结果

指标	正常	恶意	混合 Data
原始特征	52	52	52
相关性后分析（特征）	22	22	22
PCA 后（90% 方差）（组件）	4	4	4
PCA 后（95% 方差）（组件）	6	6	6
PCA 后（97% 方差）（组件）	8	8	8

D. 机器学习模型架构与训练

我们选择监督式多层感知机（MLP）架构，因为它在处理表格型协议数据方面效果显著，且兼容硬件综合框架。我们的网络架构包含两个隐藏层，每层各有 32 个神经元，并采用 ReLU 激活函数；其后接一个针对二分类优化的 sigmoid 输出层。模型参数采用 Float32 数据格式。

训练配置：数据集划分遵循标准机器学习实践：80%用于训练（通过合成少数类过采样技术增强，以实现类别平衡），20%留作测试。我们采用 Adam 优化器，结合二元交叉熵损失和 L2 正则化（λ=0.0001），以防止过拟合，同时保持泛化能力。

模型优化：硬件部署要求采用积极的优化技术。我们实施了基于 QKeras 的 [15] 量化感知训练，将权重精度降至 <8、5>格式，并在训练期间进行 80% 的稀疏剪枝。这些优化显著减少了资源消耗，同时不牺牲检测准确率；通过准确率、精确率、召回率、F1 分数以及受试者工作特征曲线下面积（ROC 曲线下面积）分析等全面的性能指标验证了这一点。

IV. 硬件实现、优化与讨论

本节介绍实验结果，包括模型在不同量化级别下的性能、硬件实现结果以及攻击检测性能。我们还将结合 AXI 协议安全讨论这些发现的意义。

A. IMS 实现与优化

为了在目标 Chipyard 片上系统平台 [13]，上将开发的 MLP 实现为 IMS，我们使用高级综合（HLS）工具将训练好的 MLP 模型转换为硬件描述语言（HDL）。我们使用 HLS4ML[16]框架将训练好的模型转换为高级综合代码，最终通过 AXI- Stream 接口生成综合后的 FPGA 比特流。生成的 IP 核可通过标准化的 AXI- Stream 接口与现有片上系统架构无缝集成，从而可作为存储器映射外设或专用安全协处理器进行部署。

为实现高效的硬件映射，还需要进一步优化。我们采用基于 QKeras 的 [15] 权重量化，将精度降低至 <8,5>，以尽可能减少内存占用。训练期间，我们采用恒定稀疏度剪枝（目标为 80%），强制将一定比例的权重置零，从而进一步降低硬件资源占用。表 II 展示了量化级别对机器学习性能的影响。 <8,5>量化级别可实现最佳性能，甚至略微优于基线 Float32 模型，这是正则化效应所致。性能在达到极端量化（<2,0>）之前仍能保持可接受水平；此时准确率会显著下降。对于此应用，这

体现了权重压缩的实际极限。我们在实验中选择了 <8, 5>量化（QKeras [15]），其 100% 的召回率表明检测率很高。我们测试了其 ROC 曲线下面积，结果显示， <8, 5>量化（QKeras [15]）的 ROC 曲线下面积超过 99%。

表 II：量化对模型性能的影响

量化等级	准确率(%)	精确率(%)	R(%)	F1分数(%)
32位浮点数	98.9	94.9	99.99	97.4
<8.5>	99.1	95.8	99.99	97.9
<8.3>	98.7	94.2	99.99	97.0
<8.1>	97.3	91.8	99.99	95.7
<2.0>	85.2	78.9	92.1	85.0

注：准确率（准确率）、精确率（精确率）、召回率（召回率）、*F1*分数（*F1* 分数）

将量化且剪枝后的多层感知机映射到 FPGA 硬件（ZCU104）时，我们旨在满足资源受限的边缘设备需求。因此，我们重点关注以下指标：

- **延迟**以毫秒为单位，用于衡量每个样本的推理时间。
- **吞吐量** 以每秒推理次数（次推理/秒）表示。
- **资源利用率** 根据硬件设计的综合与展开结果得出，报告所使用的 DSP 切片、查找表、触发器和块 RAM 的数量。

表 III 展示了在 FPGA ZCU104 上的硬件实现。量化模型显著减少了资源使用量，尤其是 DSP 切片（减少 99.5%），并且在 ZCU104 上的整体硬件资源占用非常低，因此适用于资源受限的 SoC 部署。较低的利用率为增加安全功能或部署多个检测器实例留出了充足空间。

表 III：FPGA 资源利用率详情（ZCU104）

资源 Type	基线模型	量化模型	减少量 (%)	HW 资源	使用情况 (%)
DSP 切片	799	4	99.5	1,728	0.23
LUTs	45,238	20,841	53.9	230,400	9.04
触发器	8,450	3,224	61.8	460,800	0.70
块 RAM	12	8	33.3	312	2.56
时钟频率	250 MHz	250 兆赫	0	-	-

为了评估 IMS 的性能，我们逐步增加 AXI 总线负载（10%、25% 等），同时测量 IMS 的延迟和吞吐量。表 IV 总结了 IMS 的性能结果，显示 IMS 在不同 AXI 总线负载下的性能保持稳定。

表 IV：IMS 性能指标与系统负载

系统负载 (%)	推理延迟 (ms)	吞吐量 (次推理/秒)
10	1.523	2,567,891
25	1.544	2,542,103
50	1.566	2,509,578
75	1.589	2,478,920
100	1.612	2,445,682

*B. IMS*可检测性分析

为进行此类分析，我们首先调查并评估 IMS 的整体检测率，然后仔细查看每种攻击。

*1) IMS*评估：表V比较了基线模型和量化模型的性能。较高的召回率（R）分数表明，IMS能够正确检测恶意操作。较高的AUC-ROC分数意味着IMS能够可靠地区分恶意操作与正常操作。这些结果表明，IMS性能出色，且具有很高的可检测性。

表V：IMS性能评估

模型	Acc (%)	P (%)	R (%)	F1 (%)	ROC 曲线下面积
IMS（基线）	98.9	94.9	99.99	97.4	0.993
IMS（量化 <8,5>）	99.1	95.8	99.99	97.9	0.993

注：*Acc*（准确率）、*Pr*（精确率）、*R*（召回率）、*F1*（*F1* 分数）

2) 针对特定攻击的检测评估：表 VI 显示，所提出的 IMS 在所有攻击类型中均保持较高的检测率，其中 AWLEN 溢出攻击的检测率达到 100%。针对特定攻击进行的检测评估得出以下结论：

- **突发长度漏洞利用：** 对于导致总线停滞攻击的非法 AWLEN 值（>15），检测率达到 100%
- **QoS 洪泛攻击：** 通过异常 AWQOS 信号检测优先级反转攻击模式的精确率为 97.3%。
- **事务 ID 重用：** 对于缓存不一致场景（ARIDS 重复识别），召回率为 98.1%

代表现实世界复杂攻击的混合攻击模式，检测率均保持在 98% 以上。对各类攻击的检测率验证了该模型在复杂威胁场景下的鲁棒性。

表 VI：特定攻击检测性能评估

攻击向量	样本	T.P.	F.N.	D.R.(%)	精确率(%)
AWLEN 溢出（> 15）	642	642	0	100.0	98.7
ARID 重复	558	547	11	98.0	96.2
AWQOS 泛洪（0xF）	423	411	12	97.2	94.8
AWSIZE 无效	389	381	8	97.9	95.1
ARRPROT 违规	345	339	6	98.3	96.7
混合攻击模式	885	873	12	98.6	97.1
总体	3,242	3,193	49	98.5	96.4

注：*T.P.*（真阳性）、*F.N.*（假阴性）、*D.R.*（检测率）、*Pr*（精确率）

我们的研究结果表明，IMS 对非常严重的协议违规攻击（AWLEN 溢出）实现了完美检测（检测率为 100%），同时在所有攻击向量（包括复杂的混合攻击模式）中保持较高的检测率（精确率为 >97%），且假阳性率远低于 1%（召回率为 99%）。这证明了我们的模型在现实场景中的鲁棒性和可靠性。

V. 相关工作

近年来，AXI 监控已成为一项至关重要的任务，尤其是用于检测异常和恶意操作。

表 VII：AXI 安全与监控解决方案对比

参考文献	目标保护	HW/SW	时序指标	事务级	阶段级	协议检查	标头流程。⋆	性能指标	M.O.支持†	可扩展性
XRAY [2]	AXI	SW	–	✓	✓	✓	静态	–	✓	✓
eXpect [3]	AXI	SW	–	✓	✓	✓	静态	–	✓	✓
TMU Tiny [17]	AXI4	HW	✓	✓	–	–	基础	(错误日志)	✓	✓
TMU Full [17]	AXI4	HW	✓	✓	✓	–	基础	✓	✓	✓
DD-MPU [6]	通用	HW	–	✓	–	–	仅地址	–	–	✓
VAST IFT [18]	VP级别	SW	–	–	–	–	N/A	–	–	✓
Trust MU [19]	高级高性能总线/ Wishbone	硬件	✓	✓	–	–	基础	行为	–	–
本工作	AXI4	HW	✓	✓	✓	✓	**基于机器学习**	数字信号处理器、延迟、吞吐量	✓	✓(机器学习)

⋆**报头处理**。 表示协议报头字段分析的深度：静态 = 仅进行设计时验证；基础 =简单的基于规则的验证；仅地址 = 基于地址的过滤；基于机器学习 = 智能运行时报头分析，可检测被判定为有效事务的格式错误报头。†M.O.支持表示对多笔未完成事务（已发起但尚未完成）的支持。

此类监控可提高片上系统抵御 AXI 拒绝服务攻击的可用性，并在安保领域引起关注。

为展示我们所提出的 IMS 的优势，我们将其与最先进的解决方案进行了系统比较，如表 VII 所示。尽管 XRAY [2] 和 eXpect [3] 是稳健的静态分析框架，在设计阶段提供了广泛的协议覆盖范围，涵盖事务级和阶段级监控，但这些以软件为中心的策略仍存在显著的运行局限，尤其是无法应对运行时威胁或适应动态攻击模式。这种静态特性使其无法检测通过符合协议的恶意操作实施的复杂运行时漏洞利用。所提出的事务监控单元（TMU）变体 [17] 具备基于硬件的实时监控能力，其中 TMU Full [17] 可全面覆盖事务级和阶段级监控。然而，这些监控解决方案从根本上依赖简单的基于规则的验证方法，无法从语义层面理解协议违规。TMU Tiny [17]通过简化错误日志记录实现了最低开销，却无法识别复杂的攻击模式。相反，尽管 TMU Full [17] 提供了更广泛的覆盖范围，但其根本局限在于无法区分合法事务与符合协议标准、但在语义上具有恶意的行为。DD-内存保护单元（DD-MPU） [6]体现了以内存为中心的安全策略，这类策略仅关注基于地址的过滤机制。DD-MPU 在硬件实现方面具有效率和可扩展性，但其运作独立于协议级语义违规，而后者是关键的攻击向量。仅地址处理范式无法检测报头字段的变化（篡改），并且对非法突发配置或 QoS 泛洪攻击的防御能力有限。Vast IFT [18] 和 Trust MU[19] 是针对不同协议场景的专用框架。这些解决方案在各自的目标领域内表现出较高效率，但无法应对 AXI 协议监控面临的挑战。我们提出的 IMS 独特地采用基于机器学习的方法处理 AXI 协议报头特征。与仅在设计时运行的静态验证框架或基础的基于规则的硬件监控器不同，我们的 IMS 能够智能检测复杂的攻击模式这些攻击模式会呈现为合法事务，从而逃过传统验证系统的检测。

VI. 结论

针对系统级芯片（SoC）中所使用 AXI 总线的协议级违规行为，会导致片上系统（SoC）部分或完全拒绝服务。本文提出了首个全面的方法，通过将机器学习（ML）模型集成到基于 RISC-V 的片上系统中，并以存储器映射 IP 核的形式实现，用于检测协议级恶意操作。针对缺乏底层片上系统总线违规公开数据集的问题，我们构建了一个新颖的数据集，其中包含捕获的正常 AXI 流量以及系统注入的拒绝服务数据；这些数据均通过 Xilinx ZCU104 FPGA 平台上的 RISC-V 片上系统采集。优化后的机器学习模型实现了较高的准确率（ROC 曲线下面积最高达 99.11%），并在精确率、召回率和 F1 指标方面表现稳健。该模型作为 IP 核部署在 FPGA 上，实现了实时推理，同时资源占用极低（在 ZCU104 上的数字信号处理器占用率为 0.23%，触发器利用率为 0.70%）、延迟低（1.566 ms）且吞吐量高（>2.5 百万次推理/秒）。本研究表明，在资源受限的边缘环境中，可以有效实现协议感知的、基于机器学习的恶意操作检测，从而为保护片上系统互连免受复杂硬件攻击提供切实可行的途径。

作为未来工作，我们计划探索以下方向：

- 扩大攻击覆盖范围：扩展攻击场景，涵盖更多 AXI 通道，并纳入拒绝服务场景之外的协议违规
- 机器学习模型多样性：研究其他机器学习架构（例如 LSTM、GNN），用于零日攻击检测。
- 与片上系统安全框架集成：探索与现有 SoC 防火墙、虚拟机监控器或运行时监视器协同设计，以提供分层、纵深防御保护。
- 采用先进技术库进行 ASIC 集成。

参考文献

[1] R. Patil 和 P. Sangamkar，《片上系统总线协议综述》，《电气、电子与仪器工程高级研究国际期刊》，第 04 卷，第 271–281 页，2015 年 1 月。[2] M. Zonta、N. Hinderling 和 S. Shinde，《Xray：检测并利用 ARM AXI互连中的漏洞》，发表于 2025 年欧洲设计、自动化与测试会议（DATE），第 1–7 页，2025 年。[3] M. Zonta-Roudes、A. Meza、N. Hinderling、L. Deutschmann、F. Restuccia、R. Kastner 和 S. Shinde，《eXpect：AXI实现违规的安全影响》，第 183–191 页。美国纽约州纽约：计算机协会，2025 年。[4] Xilinx，《AXI协议违规》，技术报告 PG247，AMD Xilinx，2022 年。访问日期：2025 年 7 月 7 日。[5] E. N. D. Cos.kun、S. Ahmadi-Pour、M. Hassan 和 R. Drechsler，《系统级信息流安全覆盖指标》，发表于 2024 年第 29 届亚洲及南太平洋设计自动化会议（ASP-DAC），第 945–950 页，IEEE，2024 年。[6] C. Heinz 和 A. Koch，《DD-MPU：用于嵌入式片上系统的动态分布式存储器保护单元》，发表于嵌入式计算机系统国际会议，第 285–295 页，Springer，2023 年。[7] C. Ewert、A. Neskovic、C. Heinz、F. Muuss、A. Treff、M. Gourjon、R. Buchty、T. Eisenbarth、A. Koch、M. Berekovic 等，《系统级封装的轻量级认证集成与现场安全运行》，《ACM电子系统设计自动化汇刊》，第 30 卷，第 5 期，第 1–23 页，2025 年。[8] M. Goli 和 R. Drechsler，《利用电子系统级的基于SystemC的虚拟原型，尽早验证片上系统的信息流策略》，《ACM嵌入式计算系统汇刊》，第 23 卷，2024 年 8 月。[9] M. Ayache、E. Rama、S. Mulhem、M. Berekovic 和 M. Korb，《用于评估集成电路可信度的整体框架》，发表于 2024 年 IFIP/IEEE 第 32 届超大规模集成会议（VLSI-SoC），第 1–4 页，2024 年。[10] E. Rama、M. Ayache、R. Buchty、B. Bauer、M. Korb、M. Berekovic 和 S. Mulhem，《可信集成电路：从安全性到安保及更广领域》，IEEE Access，第 12 卷，第 69603–69632 页，2024 年。[11] W. Foudhaili、A. Nechi、C. Thermann、M. Al Johmani、R. Buchty、M. Berekovic 和 S. Mulhem，《面向智能入侵检测系统的可重构边缘硬件：系统化方法》，发表于 2024 年 IFIP/IEEE 第 32 届超大规模集成会议（VLSI-SoC），（柏林、海德堡），第 1–6 页，Springer-Verlag，2024 年。[12] D. Hirsch、F. Hoffmann、A. Neskovic、C. Thermann、R. Buchty、M. Berekovic 和 S. Mulhem，《面向敏感边缘设备与系统的高效人工智能攻击检测方法》，第 177–196 页。Rives Publishers，2024 年 2 月。[13] A. Amid、D. Biancolin、A. Gonzalez、D. Grubb、S. Karandikar、H. Liew、A. Magyar、H. Mao、A. Ou、N. Pemberton、P. Rigge、C. Schmidt、J. Wright、J. Zhao、Y. S. Shao、K. Asanovi´c 和 B. Nikoli´c，《Chipyard：面向定制片上系统的集成设计、仿真与实现框架》，《IEEE Micro》，第 40 卷，第 4 期，第 10–21 页，2020 年。[14] I. Xilinx / Advanced Micro Devices，《Vivado设计套件用户指南：编程与调试》（UG908），2025 年。版本 2025.1；参见“ILA – 集成逻辑分析仪”一节，尤其是第 24–30 页。[15] C. N. Coelho、A. Kuusela、S. Li、H. Zhuang、J. Ngadiuba、T. K. Aarrestad、V. Loncar、M. Pierini、A. A. Pol 和 S. Summers，《用于粒子探测器边缘低延迟推理的深度神经网络自动异构量化》，《自然·机器智能》，第 3 卷，第 675–686 页，2021 年 6 月。[16] FastML团队，《fastmachinelearning/hls4ml》，2025 年。[17] C. Liang、T. Benz、A. Ottaviano、A. Garofalo、L. Benini 和 D. Rossi，《迈向可靠系统：一种可扩展的 AXI4事务监控方法》，发表于 2025 年欧洲设计、自动化与测试会议（DATE），第 1–7 页，2025 年。[18] E. N. D. Cos.kun、M. Hassan、M. Goli 和 R. Drechsler，《VAST：利用静态信息流跟踪，依据可用性与安保属性验证基于虚拟原型的异构系统》，发表于 2023 年第 24 届国际优质电子设计研讨会（ISQED），第 1–8 页，IEEE，2023 年。[19] M. Flasskamp、C. Klarhorst 和 J. Hagemeyer，《通过运行时监控系统行为实现可信的片上系统》，发表于第一届异构开放片上系统平台安全与安保研讨会（SSH-SoC 2023）论文集，2023 年。