Hardware Trojan Detection in Open-source Hardware Designs using Machine Learning

Victor Takashi Hayashi[1], **Wilson Vicente Ruggiero**[1]

[1]Escola Politécnica – Universidade de São Paulo (EPUSP)
Programa de Pós-Graduação em Engenharia Elétrica (PPGEE)
São Paulo – SP – Brasil

victor.hayashi@usp.br

Abstract. *The globalization of the hardware supply chain increases the risk of hardware trojans inserted by untrusted third parties, especially in reusable and open-source designs. This doctoral work proposed, developed, and evaluated methods for hardware trojan generation and detection in complex open hardware designs using Natural Language Processing (NLP), Machine Learning (ML), and Large Language Models (LLMs). The work produced datasets combining TrustHub, ISCAS85-89, RISC-V, MIPS, Web3, and Post-Quantum Cryptography (PQC) designs, totaling 3,808 instances. The best NLP/ML configuration, based on TF-IDF and Decision Tree, achieved 97.26% accuracy and 97.18% F1-score, outperforming related approaches in the evaluated setting. LLM-based prompt optimization achieved 99% recall, reducing false negatives in security-sensitive scenarios. The research also proposed an integrated framework for generation and detection of hardware trojans in open-source hardware workflows.*

Resumo. *A globalização da cadeia de suprimentos de hardware aumenta o risco de hardware trojans inseridos por terceiros não confiáveis, especialmente em designs reutilizáveis e de código aberto. Este trabalho de doutorado propôs, desenvolveu e avaliou métodos para geração e detecção de hardware trojans em designs complexos de hardware aberto utilizando Processamento de Linguagem Natural (PLN), Machine Learning (ML) e Large Language Models (LLMs). O trabalho produziu conjuntos de dados combinando designs TrustHub, ISCAS85-89, RISC-V, MIPS, Web3 e Criptografia Pós-Quântica (PQC), totalizando 3.808 instâncias. A melhor configuração de PLN/ML, baseada em TF-IDF e Decision Tree, alcançou 97,26% de acurácia e 97,18% de F1-score, superando abordagens relacionadas no cenário avaliado. A otimização de prompts baseada em LLMs atingiu 99% de recall, reduzindo falsos negativos em cenários sensíveis à segurança. A pesquisa também propôs um framework integrado para geração e detecção de hardware trojans em fluxos de hardware de código aberto.*

1. Identificação da Tese

Título em português: Detecção de hardware trojans em descrições de hardware de código aberto utilizando aprendizado de máquina
Título em inglês: *Hardware Trojan Detection in Open-source Hardware Designs using Machine Learning*

Autor: Victor Takashi Hayashi, Escola Politécnica da Universidade de São Paulo (EPUSP)
Orientador: Prof. Dr. Wilson Vicente Ruggiero, Escola Politécnica da Universidade de São Paulo (EPUSP)
Programa: Doutorado em Engenharia Elétrica, área de concentração em Engenharia da Computação. PPGEE - POLI/USP
Defesa: 11/03/2025

2. Motivação

A segurança de hardware tornou-se um desafio central para sistemas computacionais críticos [Hayashi and Ruggiero 2025a, Saha et al. 2024]. A terceirização do projeto, a reutilização de *designs* abertos e a fabricação distribuída reduzem custos e aceleram o desenvolvimento, mas também ampliam a superfície de ataque. Nesse cenário, um projetista malicioso ou um componente de terceiros comprometido pode inserir um *hardware trojan*: uma modificação deliberadamente oculta capaz de vazar dados, alterar funcionalidades, reduzir disponibilidade ou comprometer outras propriedades de segurança.

O problema é particularmente relevante para hardware aberto [Pearce 2022]. A disponibilidade do código-fonte em linguagens de descrição de hardware (HDL - *Hardware Description Language*), como Verilog, aumenta a transparência e permite verificação pela comunidade, mas não garante, por si só, que todos os módulos integrados sejam seguros. Além disso, muitos métodos de detecção são avaliados em *benchmarks* pequenos ou em circuitos simples, como AES, RS232 e VGA [Shakya et al. 2017, Elnaggar et al. 2022], o que limita a análise de escalabilidade para projetos mais complexos e atuais, como processadores RISC-V, carteiras de hardware, mineradores e módulos de criptografia pós-quântica (PQC - *Post-Quantum Cryptography*).

A Tabela 1 resume essa lacuna ao comparar o melhor resultado reportado por trabalhos relacionados e a diversidade de *benchmarks* usada nas avaliações. Embora abordagens baseadas em GNN (*Graph Neural Network*), GCN (*Graph Convolutional Network*), heurísticas de ML e LLMs proprietários (somente um trabalho nesta linha) apresentem resultados relevantes, elas foram validadas em menos famílias de circuitos. Esta tese amplia a cobertura experimental ao incluir 13 categorias, combinando *benchmarks* tradicionais com projetos complexos como RISC-V, Web3 e PQC.

Tabela 1. Comparação com trabalhos relacionados. Fonte: adaptado da tese.

Abordagem	Resultado	Métrica	Referência	Benchmarks
GNN	94%	F1-score	[Yasaei et al. 2022a]	5
GCN	93,1%	F1-score	[Yasaei et al. 2022b]	3
GCN	91,28%	F1-score	[Hassan et al. 2023]	7
ML baseado em heurísticas	95,65%	Acurácia	[Sutikno et al. 2023]	3
LLM proprietário	92%	Acurácia	[Saha et al. 2024]	1
Este trabalho	97%	Acurácia	[Hayashi and Ruggiero 2025a]	13

Com a adoção crescente de assistentes baseados em LLMs para programação e projeto digital [Thakur et al. 2023], surge ainda uma oportunidade e um risco: os mesmos modelos capazes de apoiar o desenvolvimento de hardware podem ser usados para injeção de *trojans* ou vulnerabilidades [Oh et al. 2024]. A tese parte dessa lacuna para investigar

como técnicas de PLN (Processamento de Linguagem Natural), ML (*Machine Learning*) e LLMs (*Large Language Models*) podem apoiar tanto a geração controlada de exemplos para avaliação quanto a detecção baseada em avaliação estática de *hardware trojans* em descrições HDL.

3. Objetivos

O objetivo geral da tese foi propor, desenvolver e avaliar métodos de detecção de *hardware trojans* adequados a descrições de hardwares complexos. Esse objetivo foi desdobrado nos seguintes objetivos específicos:

- Construir conjuntos de dados mais abrangentes para avaliação de detectores de *hardware trojans*, combinando *benchmarks* existentes e novos exemplos;
- Investigar o uso de LLMs para geração controlada de *hardware trojans* em projetos complexos, como RISC-V, Web3, MIPS e PQC;
- Avaliar técnicas de extração de características textuais, como Bag of Words, TF-IDF e *embeddings*, aplicadas à análise estática de HDL;
- Comparar classificadores de ML para detecção binária de projetos com e sem *trojans*;
- Avaliar LLMs com engenharia e otimização de prompts para detecção de *hardware trojans*;
- Propor um *framework* integrado de geração e detecção aplicável ao fluxo de desenvolvimento de hardware aberto.

4. Metodologia

A metodologia combinou uma abordagem ofensiva, voltada à geração de exemplos, com uma abordagem defensiva, voltada à detecção. Na etapa de geração, LLMs proprietários e abertos foram empregados para inserir *trojans* em implementações abertas de hardware, que foram consideradas como *golden models* (i.e., exemplos legítimos sem *trojans*). Os exemplos gerados foram inspirados na taxonomia existente TrustHub, contemplando principalmente *trojans* baseados em *trigger* externo e em tempo [Hayashi and Ruggiero 2024a].

A Figura 1 apresenta o *framework* proposto, que organiza o fluxo de geração de *hardware trojans* com LLMs e a detecção por análise estática usando PLN, ML e otimização de *prompts*. No contexto de hardware aberto, o avaliador atua antes da integração e da fabricação, reduzindo o risco de componentes RTL maliciosos seguirem no fluxo de desenvolvimento.

Na etapa de detecção, os arquivos Verilog foram tratados como artefatos textuais. Antes da classificação, termos que revelariam diretamente a classe, como "trojan", "Tj" e "payload", bem como comentários de código, foram removidos. Em seguida, foram avaliadas representações baseadas em Bag of Words, TF-IDF e *LLM embeddings*. Essas representações alimentaram classificadores, incluindo *Decision Tree*, *Random Forest*, *Gradient Boosting*, SVM, Regressão Logística, KNN e Naive Bayes [Hayashi and Ruggiero 2025a].

Uma segunda linha de detecção explorou LLMs diretamente como avaliadores de designs HDL. Foram comparados prompts com diferentes perfis de análise, incluindo papéis de engenheiro de verificação, analista de segurança e engenheiro de testes. Também

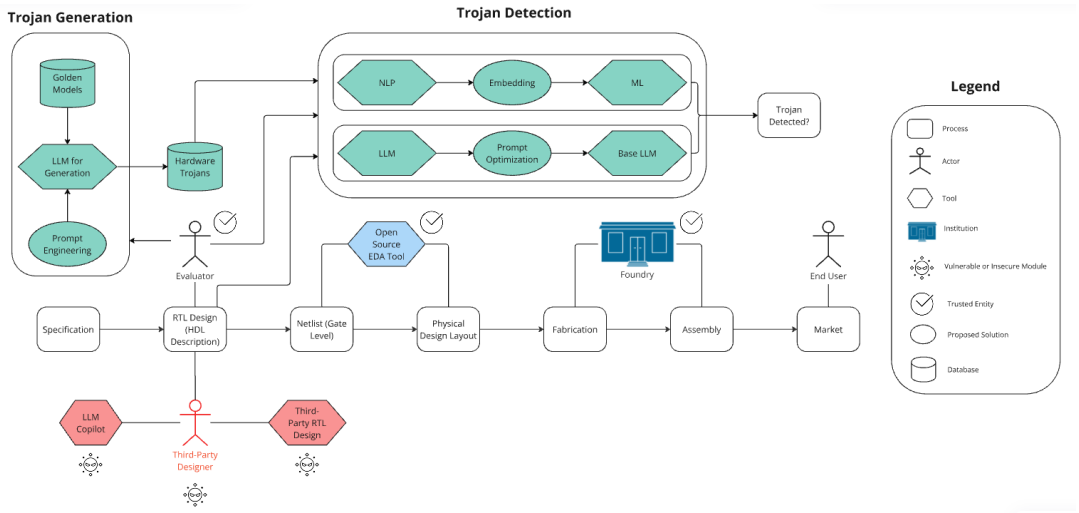

Figura 1. *Framework* para geração e detecção de *hardware trojans*. Fonte: adaptado da tese.

foram utilizadas estratégias de otimização de prompt baseadas em revisão por listas de verificação, com foco em maximizar *recall*, métrica prioritária quando falsos negativos podem permitir que um componente malicioso avance no fluxo de fabricação de dispositivos.

5. Resultados Obtidos

O primeiro resultado foi a construção de um conjunto de dados consolidado com 3.808 instâncias [Hayashi and Ruggiero 2025a], balanceado por *oversampling*, reunindo exemplos de TrustHub, ISCAS85-89 e dados sintéticos gerados com LLMs. A base resultante ampliou a cobertura em relação a trabalhos relacionados ao incluir, além de circuitos tradicionais, projetos como RISC-V, MIPS, hardware wallet, minerador PoW e módulos de criptografia pós-quântica [Hayashi and Ruggiero 2024a, Ravi et al. 2021, Pagliarini et al. 2024].

No subconjunto inicialmente gerado com ChatGPT-4, foram obtidos 290 exemplos Verilog [Hayashi and Ruggiero 2024a] a partir de 29 modelos dourados de RISC-V, hardware wallet e minerador PoW, com 10 *trojans* gerados para cada módulo. A análise mostrou que os LLMs foram capazes de produzir variações sintéticas úteis para avaliação, mas também evidenciou limitações: os exemplos tenderam a se concentrar em *triggers* baseados em condição externa ou tempo, sem cobrir *triggers* físicos ou analógicos.

Na detecção por PLN e ML, a configuração com TF-IDF e *Decision Tree* apresentou os melhores resultados [Hayashi and Ruggiero 2025a], alcançando 97,26% de acurácia e 97,18% de F1-score. Esses resultados superaram o estado da arte considerado na tese para o conjunto de dados avaliado, ao mesmo tempo em que preservaram custo computacional menor do que abordagens baseadas exclusivamente em LLMs. Já a abordagem com LLMs e otimização de prompts alcançou 99% de *recall*, indicando maior adequação para cenários em que a redução de falsos negativos é mais importante do que a maximização isolada da acurácia.

A Tabela 2 resume os principais resultados quantitativos. As representações BoW

e TF-IDF combinadas com classificadores clássicos obtiveram as maiores acurácias e F1-score, enquanto as estratégias baseadas em LLM apresentaram maior *recall*, característica relevante para detecção de trojans por priorizar a redução de falsos negativos.

Tabela 2. Principais resultados de detecção. Fonte: adaptado da tese.

Representação/abordagem	Modelo ou prompt	Acurácia	Recall	F1-score
BoW	Decision Tree	97,52%	95,12%	97,45%
TF-IDF	Extra Trees	97,26%	94,60%	97,18%
TF-IDF	Decision Tree	97,11%	94,29%	97,02%
LLM *embedding*	XGBoost	96,21%	96,63%	96,23%
LLM direto	Llama 3.3-70B com checklist	70,14%	99,58%	76,93%
LLM direto	Analista de segurança (Llama 3.3)	80,67%	98,84%	83,64%

Como síntese dos resultados, a tese propôs um *framework* para geração e detecção de *hardware trojans* (vide Figura 1). O *framework* posiciona o avaliador como uma camada de defesa no fluxo de hardware aberto: componentes RTL de terceiros podem ser analisados estaticamente antes da integração e antes do envio a etapas posteriores de síntese e fabricação. O modelo também contempla o uso de geração sintética como estratégia de *red team*, permitindo ampliar bases de treinamento e testar detectores em novas implementações de *designs* complexos.

6. Contribuições e Originalidade

A tese apresenta cinco contribuições principais para a área de segurança:

1. um método de geração de *hardware trojans* com LLMs para apoiar a avaliação de detectores em projetos complexos de hardware aberto [Hayashi and Ruggiero 2024a];
2. novos conjuntos de dados de *hardware trojans* em projetos complexos disponibilizados no repositório de dados da USP [1]
3. um método de análise estática usando PLN e ML para detecção de *hardware trojans* em descrições HDL [Hayashi and Ruggiero 2025a];
4. um método de detecção usando LLMs abertos com engenharia e otimização de prompts, priorizando alto *recall* [Hayashi and Ruggiero 2025a];
5. um *framework* integrado de geração e detecção de *hardware trojans*, com código disponível publicamente [2].

A originalidade do trabalho está na combinação entre hardware aberto, projetos complexos, geração sintética com LLMs e detecção estática com PLN/ML. Em vez de restringir a avaliação a circuitos pequenos ou *benchmarks* tradicionais, a tese incorporou cenários mais próximos de aplicações modernas de segurança, como RISC-V, Web3 e criptografia pós-quântica. Além disso, o trabalho tratou LLMs não apenas como detectores, mas também como ferramentas para ampliar bases de avaliação, explicitando suas capacidades e limitações.

[1] `http : / / repositorio.uspdigital.usp.br / handle / item / 723`, `http : / / repositorio.uspdigital.usp.br / handle / item / 724` e `http : / / repositorio.uspdigital.usp.br/handle/item/725`

[2] `https://github.com/vthayashi/hw-tj-framework`

7. Produção Científica Associada

As principais publicações diretamente associadas à tese são:

- (Periódico Fator de Impacto 3.6) Victor Takashi Hayashi e Wilson Vicente Ruggiero. "Hardware Trojan Detection in Open-source Hardware Designs using Machine Learning". *IEEE Access*, 2025 [Hayashi and Ruggiero 2025a].
- (Periódico Fator de Impacto 2.0) Victor Takashi Hayashi e Wilson Vicente Ruggiero. "Hardware Trojan Dataset of RISC-V and Web3 Generated with ChatGPT-4". *Data*, v. 9, n. 6, artigo 82, 2024 [Hayashi and Ruggiero 2024a].
- Victor T. Hayashi e Wilson V. Ruggiero. "Towards Hardware Trojan Education". *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, 2024 [Hayashi and Ruggiero 2024b]. Trabalho premiado com o primeiro lugar na *Hardware Demonstration Competition* [3].
- Victor T. Hayashi e Wilson V. Ruggiero. "Open Hardware Synthesis: A Precursor Case Study". *16th IEEE International Conference on Industry Applications (INDUSCON)*, 2025 [Hayashi and Ruggiero 2025b].
- Victor T. Hayashi, Wilson V. Ruggiero e Felipe V. de Almeida. "LabEAD AutoTest: Online Tests of Hardware Designs". *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, SBC, 2022, p. 95–102 [Hayashi et al. 2022].

Além das publicações diretamente associadas à detecção de *hardware trojans*, a tese foi apoiada por trabalhos complementares durante a trajetória do doutorado. O trabalho sobre laboratório remoto com experimentação prática em FPGA no periódico *IEEE Transactions on Education* foi a base para a abordagem de análise dinâmica apresentada na conferência IEEE HOST [Hayashi et al. 2023]. O artigo LabBitcoin, apresentado no SBSeg 2021, explorou um *testbed* FPGA/IoT para experimentos com Bitcoin, consumo de energia e implementações Verilog, contribuindo para o estudo de caso em aplicações Web3 considerado na construção do dataset de *hardware trojans* [Hayashi et al. 2021]. De forma complementar, os capítulos de minicursos do SBSeg 2024 e 2025 abordam aspectos de IA Adversarial e ameaças de computação quântica à criptografia, que foram importantes para o método de ataque utilizado na tese, e para o estudo de caso de PQC explorado[4], enquanto o artigo publicado no periódico *Applied Sciences* sobre sistemas conversacionais específicos de domínio com aprendizado contínuo contribuiu para a base de PLN e ML posteriormente aplicada à análise de descrições de hardware [Pinna et al. 2024].

8. Disponibilidade da Tese

A tese está disponível no banco de teses da USP [5].

[3] `https://www.poli.usp.br/noticias/doutorando-da-escola-politecnica-da-usp-ganha-premio-em-conferencia-internacional-por-demonstracao-de-hardware/`

[4] `https://doi.org/10.5753/sbc.15101.8.2` e `https://books-sol.sbc.org.br/index.php/sbc/catalog/download/178/788/1537?inline=1`

[5] `https://teses.usp.br/teses/disponiveis/3/3141/tde-06052025-090853/pt-br.html`

9. Uso de IA Generativa

ChatGPT-4 foi utilizado para revisão deste texto.

Referências

Elnaggar, R., Chaudhuri, J., Karri, R., and Chakrabarty, K. (2022). Learning malicious circuits in fpga bitstreams. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Hassan, R., Meng, X., Basu, K., and Dinakarrao, S. M. P. (2023). Circuit topology-aware vaccination-based hardware trojan detection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Hayashi, V., Dutra, J., Almeida, F., Arakaki, R., Midorikawa, E., Canovas, S., Cugnasca, P., and Ruggiero, W. (2023). Implementation of pjbl with remote lab enhances the professional skills of engineering students. *IEEE Transactions on Education*, 66(4):369–378.

Hayashi, V. T., de Almeida, F. V., and Komo, A. E. (2021). Labbitcoin: Fpga iot testbed for bitcoin experiment with energy consumption. In *Anais Estendidos do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 90–97. SBC.

Hayashi, V. T. and Ruggiero, W. V. (2024a). Hardware trojan dataset of risc-v and web3 generated with chatgpt-4. *Data*, 9(6).

Hayashi, V. T. and Ruggiero, W. V. (2024b). Towards hardware trojan education.

Hayashi, V. T. and Ruggiero, W. V. (2025a). Hardware trojan detection in open-source hardware designs using machine learning. *IEEE Access*.

Hayashi, V. T. and Ruggiero, W. V. (2025b). Open hardware synthesis: A precursor case study. In *2025 16th IEEE International Conference on Industry Applications (INDUSCON)*, pages 01–08.

Hayashi, V. T., Ruggiero, W. V., and de Almeida, F. V. (2022). Labead autotest: Online tests of hardware designs. In *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, pages 95–102. SBC.

Oh, S., Lee, K., Park, S., Kim, D., and Kim, H. (2024). Poisoned chatgpt finds work for idle hands: Exploring developers' coding practices with insecure suggestions from poisoned ai models. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 1141–1159. IEEE.

Pagliarini, S., Aikata, A., Imran, M., and Sinha Roy, S. (2024). Repqc: Reverse engineering and backdooring hardware accelerators for post-quantum cryptography. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pages 533–547.

Pearce, J. M. (2022). Strategic investment in open hardware for national security. *Technologies*, 10(2):53.

Pinna, F. C. d. A., Hayashi, V. T., Néto, J. C., Marquesone, R. d. F. P., Duarte, M. C., Okada, R. S., and Ruggiero, W. V. (2024). A modular framework for domain-

specific conversational systems powered by never-ending learning. *Applied Sciences*, 14(4):1585.

Ravi, P., Deb, S., Baksi, A., Chattopadhyay, A., Bhasin, S., and Mendelson, A. (2021). On threat of hardware trojan to post-quantum lattice-based schemes: a key recovery attack on saber and beyond. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 81–103. Springer.

Saha, D., Tarek, S., Yahyaei, K., Saha, S. K., Zhou, J., Tehranipoor, M., and Farahmandi, F. (2024). Llm for soc security: A paradigm shift. *IEEE Access*.

Shakya, B., He, T., Salmani, H., Forte, D., Bhunia, S., and Tehranipoor, M. (2017). Benchmarking of hardware trojans and maliciously affected circuits. *Journal of Hardware and Systems Security*, 1:85–102.

Sutikno, S., Septafiansyah, D. P., Wijitrisnanto, F., and Aminanto, M. E. (2023). Detecting unknown hardware trojans in register transfer level leveraging verilog conditional branching features. *IEEE Access*.

Thakur, S., Ahmad, B., Fan, Z., Pearce, H., Tan, B., Karri, R., Dolan-Gavitt, B., and Garg, S. (2023). Benchmarking large language models for automated verilog rtl code generation. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE.

Yasaei, R., Chen, L., Yu, S.-Y., and Al Faruque, M. A. (2022a). Hardware trojan detection using graph neural networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Yasaei, R., Faezi, S., and Al Faruque, M. A. (2022b). Golden reference-free hardware trojan localization using graph convolutional network. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 30(10):1401–1411.