

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Állan Fabrício Garcia Ferreira

**Geração de Testes Indiretos de Circuitos
Analógicos baseados em Aprendizado de
Máquina utilizando Simulações SPICE**

Porto Alegre

2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Állan Fabrício Garcia Ferreira

**Geração de Testes Indiretos de Circuitos Analógicos
baseados em Aprendizado de Máquina utilizando
Simulações SPICE**

Projeto de Diplomação II, apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de Engenheiro Eletricista

UFRGS

Orientador: Prof. Dr. Tiago Roberto Balen

Porto Alegre

2023

Állan Fabrício Garcia Ferreira

**Geração de Testes Indiretos de Circuitos Analógicos
baseados em Aprendizado de Máquina utilizando
Simulações SPICE**

Projeto de Diplomação II, apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para a obtenção do grau de Engenheiro Eletricista

BANCA EXAMINADORA

Prof. Dr. Paulo Francisco Butzen
UFRGS

Prof. Dr. José Rodrigo Furlanetto
Azambuja
UFRGS

Prof. Dr. Tiago Roberto Balen
Orientador - UFRGS

Aprovado em 12 de maio de 2023.

Agradecimentos

Agradeço aos meus pais, Marcelo e Fabiane, e ao meu irmão, Marcelo Júnior, pelo apoio e suporte durante a graduação. Aos amigos do curso, muito obrigado pela companhia. Ao meu orientador, Tiago, pelos ensinamentos e conselhos que foram essenciais para a conclusão deste trabalho. À UFRGS pela educação de excelência. À Júlia Baretta, por ler cada linha desse texto, obrigado por tudo.

Resumo

Em CIs analógicos, uma parte significativa do preço final do produto é concentrado na etapa de teste, principalmente quando o teste funcional é utilizado. A alta qualidade desse tipo de teste se dá pela verificação de todas as especificações do circuito. Para tanto, são usados equipamentos de testes automatizados caros que possuem instrumentos de alta resolução e demandam longos tempos de teste. Uma alternativa na busca da diminuição do custo gerado pelo teste funcional é o teste indireto. Nessa técnica, os parâmetros funcionais de um determinado circuito são estimados a partir de um conjunto de medidas de baixo custo. Neste trabalho, é proposto um método para geração de testes indiretos de circuitos analógicos baseados em aprendizado de máquina. Ele visa apontar quais medidas indiretas (MIs) e algoritmos de aprendizado são os mais adequados para implementação do teste indireto de um determinado circuito. O objetivo é auxiliar o engenheiro de teste a avaliar a viabilidade do teste indireto em relação ao teste convencional. Para esse fim, são efetuadas simulações SPICE do circuito sob teste, criando um conjunto de dados sintéticos que permitem treinar e avaliar modelos de aprendizado. Os modelos são comparados entre si, e aqueles que apresentam melhor performance são selecionados para integrar o teste indireto. O uso da metodologia foi validado através de experimentos com dois estudos de caso: um filtro passa-baixas e um FDA operando em malha aberta. O primeiro caso resultou em testes indiretos que fazem uso de 19 MIs no total e dos algoritmos de aprendizado MARS e MLR, resultando em um R^2 igual a 0,76, 0,76 e 0,98, para as previsões do ganho DC, frequência de corte e SR, respectivamente. Já no segundo estudo de caso, foram apontadas 27 MIs no total, que devem ser usadas para treinar modelos MARS e MLR. Os modelos gerados apresentaram bom desempenho, com um R^2 aproximadamente igual a 1 para as estimativas do ganho DC, GBW e SR do circuito. Finalmente, é possível afirmar que o método apresentado facilita a implementação do teste indireto em circuitos analógicos. Uma grande combinação de parâmetros de projeto podem ser explorados, selecionando aqueles que melhor contribuem para a qualidade do teste indireto.

Palavras-chave: CIs analógicos, geração de testes, teste indireto, aprendizado de máquina.

Abstract

In analog ICs, a significant portion of the final product cost is concentrated in the test stage, especially when functional test is used. The high quality of this type of test is achieved by verifying all circuit specifications. Expensive automated test equipment is used, which has high-resolution instruments and requires long test times. An alternative in the search for a reduction in the cost generated by functional test is the indirect test. In this technique, the functional parameters of a particular circuit are estimated from a set of low-cost measurements. In this work, a method for generating indirect tests of analog circuits based on machine learning is proposed. It aims to identify which indirect measurements (IMs) and learning algorithms are most suitable for implementing the indirect test of a given circuit. The objective is to assist the test engineer in evaluating the feasibility of the indirect test in relation to the conventional test. For this purpose, SPICE simulations of the circuit under test are performed, creating a set of synthetic data that allows training and evaluating learning models. The models are compared to each other, and those with the best performance are selected to integrate the indirect test. The methodology was validated through experiments with two case studies: a low-pass filter and an open loop FDA. The first case resulted in indirect tests that use a total of 19 IMs and the MARS and MLR learning algorithms, resulting in a R^2 score equal to 0.76, 0.76 and 0.98, for predicting the DC gain, cutoff frequency and SR, respectively. In the second case study, a total of 27 IMs were identified, which should be used to train MARS and MLR models. The generated models presented good performances, with a R^2 score approximately equal to 1 for estimating the DC gain, GBW and SR of the circuit. Finally, it is possible to affirm that the presented method facilitates the implementation of the indirect test in analog circuits. A wide range of design parameters can be explored, selecting those that best contribute to indirect test quality.

Keywords: analog ICs, test generation, indirect test, machine learning.

Lista de Figuras

Figura 1 – Custo relativo por falha em cada etapa da produção de um chip	17
Figura 2 – Exemplos de testes indiretos orientados à classificação (a) e predição (b)	20
Figura 3 – Identificação dos CIs pelo teste indireto orientado à predição	21
Figura 4 – Arquitetura do teste indireto	22
Figura 5 – Exemplo de aplicação de uma regressão linear simples	24
Figura 6 – Exemplo de aplicação de um modelo MARS	25
Figura 7 – Exemplos de aplicações de modelos SVM com diferentes funções de Kernel	26
Figura 8 – Formato de uma DT	27
Figura 9 – Exemplo de aplicação de uma DT	27
Figura 10 – Tipos de métodos de seleção de características	28
Figura 11 – Funcionamento da estratégia de SFS	30
Figura 12 – Metodologia de geração de testes indiretos	34
Figura 13 – Método de análise da qualidade do teste indireto	41
Figura 14 – Diagrama esquemático do filtro passa-baixas de um estágio	43
Figura 15 – Blocos de ganho (a) e CMFB (b) do FDA	44
Figura 16 – Esquemático do FDA em malha aberta	46
Figura 17 – Distribuições das especificações do filtro passa-baixas	48
Figura 18 – NRMSE dos modelos na predição do ganho DC do filtro passa-baixas .	49
Figura 19 – NRMSE dos modelos na predição da freq. de -3dB do filtro passa-baixas	50
Figura 20 – NRMSE dos modelos na predição do SR do filtro passa-baixas	51
Figura 21 – Distribuições das especificações do FDA em malha aberta	53
Figura 22 – NRMSE dos modelos na predição do ganho DC do FDA em malha aberta	54
Figura 23 – NRMSE dos modelos na predição do GBW do FDA em malha aberta .	55
Figura 24 – NRMSE dos modelos na predição do SR do FDA em malha aberta . .	56

Lista de Tabelas

Tabela 1 – Versões dos principais programas usados neste trabalho	34
Tabela 2 – Valores dos parâmetros dos componentes do bloco de ganho do FDA .	44
Tabela 3 – Valores dos parâmetros dos componentes do bloco de CMFB do FDA .	44
Tabela 4 – Tempos de execução da geração de teste indiretos para os estudos de caso	47
Tabela 5 – Coeficientes de variação de cada conjunto do primeiro estudo de caso .	48
Tabela 6 – Testes indiretos gerados para o filtro passa-baixas	52
Tabela 7 – Coeficientes de variação de cada conjunto do segundo estudo de caso .	53
Tabela 8 – Testes indiretos gerados para o FDA em malha aberta	55

Lista de abreviaturas

AC	<i>Alternate Current</i>
ADC	<i>Analog to Digital Converter</i>
ATE	<i>Automatic Test Equipment</i>
ATPG	<i>Automatic Test Pattern Generation</i>
BIST	<i>Built-In Self-Test</i>
CI	Circuito Integrado
cLHS	<i>conditioned Latin Hypercube Sampling</i>
CMFB	<i>Common-Mode Feedback</i>
CMOS	<i>Complementary Metal-Oxide Semiconductor</i>
DC	<i>Direct Current</i>
dCor	<i>distance Correlation</i>
DfT	<i>Design for Testability</i>
DL	<i>Defect Level</i>
DPPM	<i>Defective Parts Per Million</i>
DT	<i>Decision Tree</i>
FF	<i>Fast-Fast</i>
freq.	frequência
FS	<i>Fast-Slow</i>
GBW	<i>Gain-Bandwidth Product</i>
MARS	<i>Multivariate Adaptive Regression Splines</i>
MC	Monte Carlo
MI	Medidas Indiretas
MLR	<i>Multiple Linear Regressions</i>
MR	<i>Misclassification Rate</i>
NRMSE	<i>Normalized Root Mean Squared Error</i>
PDK	<i>Process Design Kit</i>

RBF	<i>Radial Basis Function</i>
RMSE	<i>Root Mean Squared Error</i>
SFS	<i>Sequential Forward Selection</i>
SF	<i>Slow-Fast</i>
SoC	<i>System on a Chip</i>
SPICE	<i>Simulation Program with Integrated Circuit Emphasis</i>
SR	<i>Slew Rate</i>
SS	<i>Slow-Slow</i>
SVM	<i>Support Vector Machines</i>
trein.	treinamento
YL	<i>Yield Loss</i>

Sumário

1	INTRODUÇÃO	12
1.1	Motivação	13
1.2	Objetivos	14
1.2.1	Objetivo Geral	14
1.2.2	Objetivos Específicos	15
1.3	Estruturação do Trabalho	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Teste de Circuitos Analógicos	16
2.1.1	Teste em Produção	17
2.1.2	Teste Funcional <i>versus</i> Teste Estrutural	18
2.2	Teste Indireto	19
2.2.1	Testes Indiretos Orientados à Classificação e Predição	20
2.2.2	Arquitetura do Teste Indireto	22
2.3	Algoritmos de Regressão	23
2.3.1	Regressão Linear Múltipla	24
2.3.2	<i>Splines</i> de Regressão Adaptativa Multivariada	25
2.3.3	Máquina de Vetores de Suporte	26
2.3.4	Árvore de Decisão	27
2.4	Seleção das Medidas Indiretas	28
2.4.1	Métodos de filtro	29
2.4.2	Métodos <i>Wrapper</i>	30
2.5	Métricas de Avaliação	31
2.5.1	Raiz do Erro Médio Quadrático Normalizado	31
2.5.2	Coefficiente de Determinação	31
2.5.3	Taxa de Erro de Classificação	32
3	METODOLOGIA EXPERIMENTAL	33
3.1	Ferramentas de <i>Software</i> e <i>Hardware</i>	33
3.2	Geração de Testes Indiretos	34
3.2.1	Simulação dos Circuitos	35
3.2.2	Treinamento dos Modelos de Aprendizado	37
3.2.3	Seleção dos Testes Indiretos	40
3.3	Análise da Qualidade do Teste Indireto	41
3.4	Estudos de Caso	43
3.4.1	Filtro Passa-Baixas de Um Estágio	43

3.4.2	Amplificador Totalmente Diferencial de Um Estágio	46
4	RESULTADOS E DISCUSSÕES	47
4.1	Tempos de Execução dos Algoritmos	47
4.2	Filtro Passa-Baixas de Um Estágio	48
4.3	Amplificador Totalmente Diferencial de Um Estágio	53
5	CONCLUSÕES	58
5.1	Trabalhos Futuros	59
	REFERÊNCIAS BIBLIOGRÁFICAS	60

1 Introdução

Manter um crescimento exponencial no desenvolvimento do ramo da microeletrônica por mais de 50 anos foi pouco menos que um milagre, mas esse feito veio com um custo. O esforço para escalar o CMOS de silício dominou esmagadoramente os investimentos de capital intelectual e financeiro da indústria, governo e academia (WILLIAMS, 2017). A Lei de Moore (MOORE, 2006) ditou precedentes para uma série constante de avanços na tecnologia de semicondutores através da duplicação do número de transistores por Circuito Integrado (CI) a cada 2 anos. De acordo com (CHIEN; KARAMCHETI, 2013), buscou-se uma evolução na combinação de fatores como custo do dispositivo, número de CIs por *wafers* de silício, consumo de energia e confiabilidade.

Com a evolução dos CIs, mais funções estão sendo agregadas em um mesma pastilha de semicondutor. Atualmente, sistemas em um chip, ou SoC (*System on a Chip*), realizam processamento de sinais, comunicação sem fio, inteligência artificial, entre outras tarefas. Isso possibilita uma diminuição de custo, área e potência consumida. Apesar dessas vantagens, novos desafios são encontrados quando múltiplas funções são atribuídas a um mesmo CI, como problemas de alimentação, valor da tensão de limiar, efeito de canal curto, flutuações no número de dopantes, dentre outros (TAUR *et al.*, 1997).

Desse modo, a garantia do funcionamento correto dos CIs é conhecida como uma etapa árdua, mas indispensável. Os CIs devem ser testados durante o processo da fabricação, descartando os circuitos que não atinjam as exigências mínimas de projeto. No entanto, no estado da arte não existe um método de teste perfeito. A qualidade de um teste pode ser definida pela métrica *Defective Parts Per Million* (DPPM), que considera o número de *test escapes* (componentes defeituosos que passam no teste em produção) por milhão de CIs produzidos. A indústria de CIs busca reduzir o DPPM visando o aumento da produtividade da cadeia de produção.

Embora os circuitos digitais sejam peça-chave para o desenvolvimento da eletrônica, oferecendo vantagens como multifuncionalidade e simplicidade de projeto, os circuitos analógicos ainda são indispensáveis nos casos onde é necessária uma interação com variáveis físicas. A razão disso é que o mundo físico é por natureza analógico.

Segundo (ARABI; KAMINSKA, 1997), as especificações de circuitos analógicos geralmente são muito amplas, resultando em má cobertura de falhas, longos tempos de teste e a necessidade de equipamentos dedicados no processo produtivo. Em (SOUDERS; STENBAKKEN, 1991), foi reportado que, em circuitos de sinal misto, 95% do custo do teste é gasto no teste das partes analógicas, enquanto que a contraparte digital equivale a apenas 5% do preço total.

Por essa razão, uma parte significativa do preço final do produto está concentrado na etapa de teste, principalmente quando o teste funcional (baseado em especificações) é utilizado. Esse tipo de teste apresenta uma alta qualidade, porém ele gera um custo elevado, tendo em vista o uso de ATEs (*Automatic Test Equipments*), equipamentos de testes automatizados com instrumentos de alta resolução e longos tempos de teste. Uma maneira de reduzir o preço assimilado ao teste e manter uma boa qualidade é optar pelo uso do teste estrutural, onde o foco está voltado para detecção de defeitos. Contudo, o teste estrutural no contexto de circuitos analógicos é pouco empregado devido a falta de um modelo de falhas padronizado que represente os principais efeitos físicos que ocorrem em componentes analógicos.

1.1 Motivação

Visando a aplicação do teste estrutural e orientado a defeitos, uma ferramenta de geração automática de padrões de teste, conhecida como ferramenta de ATPG (*Automatic Test Pattern Generation*), para circuitos analógicos foi proposta inicialmente em (ZILCH; LUBASZEWSKI; BALEN, 2021). Por meio de injeções de falhas por simulação, a ferramenta é capaz de determinar os vetores de teste que maximizam a cobertura de falhas, considerando os nós de teste e modelo de falhas escolhidos pelo usuário.

Embora o teste estrutural da ferramenta seja eficiente, com alta cobertura de falhas e baixo número de medidas, ele não garante que falhas com maior probabilidade de ocorrência sejam identificadas. Logo, o teste funcional e orientado a especificações ainda é essencial em sistemas críticos onde a segurança do funcionamento do circuito precisa ser muito alta, como sistemas automobilísticos, aeroespaciais e nucleares.

Devido à importância do teste funcional para circuitos analógicos, estudos exploraram maneiras de diminuir seu custo e manter seu baixo DPPM. Nas últimas duas décadas, uma alternativa que vem sendo aprimorada é o teste indireto (*indirect test*).

O conceito de teste indireto se baseia na hipótese de que as variações do processo de fabricação têm influência em parâmetros que podem ser medidos por equipamentos de baixo custo, como amplitude de sinais AC e DC. Desse modo, é possível correlacionar essas Medidas Indiretas (MIs) com especificações do circuito (VARIYAM; CHATTERJEE, 1998). Entretanto, essa relação é muito complexa para construção de modelos analíticos. Nesse tipo de problema, o aprendizado de máquina se destaca. Os modelos de aprendizado são capazes de representar funções complexas com uma alta precisão.

No teste indireto, um algoritmo de aprendizado supervisionado treina um modelo que possui como entradas as medidas de baixo custo e como saída um parâmetro funcional. Ao final do treinamento, o modelo é validado com alguma métrica adequada para identificar se o aprendizado foi bem sucedido (STRATIGOPOULOS, 2018).

Em (BADAWI *et al.*, 2021), é mostrado um uso do teste indireto com algoritmos de aprendizado. Os dados foram coletados de circuitos de RF em um ambiente de teste em produção. Medidas funcionais e DC foram feitas para treinar modelos utilizando os algoritmos MARS e SVM. Nesse trabalho, foi utilizada uma estratégia em duas fases, a fim de que o nível de confiança da predição alcançasse patamares aceitos na indústria. Após o treinamento, os CIs tinham seus parâmetros funcionais estimados através das MIs. Caso a especificação predita de um circuito ficasse dentro de uma zona de tolerância, ele seguia para a segunda fase, onde um teste funcional convencional era aplicado. Assim, apenas um pequeno lote precisa ser testado utilizando os dois testes. O resultado é um número menor de CIs que passam pelo custoso teste funcional durante a produção.

Um estudo semelhante pode ser encontrado em (STRATIGOPOULOS; MIR; MAKRIS, 2009), onde uma estratégia composta de duas fases também foi aplicada. Porém, o teste indireto foi projetado como um problema de classificação. O modelo de aprendizado consiste em uma junção de duas redes neurais ontogênicas que são responsáveis por definir os limites de teste no espaço de MIs. Também é apresentado um método para gerar dados sintéticos que seguem a distribuição original dos dados coletados. Essa técnica visa balancear as quantidades de circuitos bons e defeituosos, visto que circuitos defeituosos são amostras raras no processo de fabricação. Desta forma, pode-se realizar um treinamento melhor dos modelos, permitindo que o erro da estimativa chegue na ordem de DPPM.

1.2 Objetivos

1.2.1 Objetivo Geral

Neste contexto, a proposta deste trabalho consiste no desenvolvimento de um método para geração de testes indiretos de circuitos analógicos baseados em aprendizado de máquina. O método proposto visa apontar quais MIs e algoritmos de aprendizado são os mais adequados para implementação do teste indireto de um determinado circuito. Para esse fim, são efetuadas simulações SPICE do circuito sob teste, criando um conjunto de dados sintéticos que permitem treinar e avaliar modelos de aprendizado. O objetivo é auxiliar o engenheiro de teste a avaliar a viabilidade do teste indireto em relação ao teste convencional, levando em consideração um conjunto de especificações e MIs do circuito.

O trabalho foi desenvolvido visando uma futura integração com a ferramenta de ATPG para testes estruturais apresentada em (ZILCH; LUBASZEWSKI; BALEN, 2021). Portanto, várias decisões de projeto foram tomadas tendo essa integração em mente.

1.2.2 Objetivos Específicos

- Gerar, por meio de simulações SPICE, conjuntos de dados que incluem parâmetros funcionais e estruturais de circuitos analógicos. A variabilidade das simulações é controlada pelos parâmetros de Monte Carlo da tecnologia do circuito, permitindo reproduzir um ambiente de processo de fabricação de CIs.
- Treinar um número específico de modelos de aprendizado, selecionando entradas por diferentes métodos de seleção de características.
- Selecionar o conjunto de MIs e algoritmos de aprendizado que resultam nos modelos capazes de predir com maior precisão as especificações do circuito.
- Definir os limites de teste do circuito por diferentes estratégias a fim de estimar a qualidade do teste indireto.
- Aplicar o método de geração de testes indiretos em dois circuitos usados como estudos de caso para validar a ferramenta de ATPG mencionada anteriormente.

1.3 Estruturação do Trabalho

Este trabalho é estruturado da maneira descrita a seguir. No próximo capítulo, é apresentada a revisão bibliográfica que serve de base para o seu desenvolvimento. Além de tratar brevemente do teste de CIs analógicos, o capítulo aborda o teste indireto de maneira mais profunda, assim como conceitos gerais de aprendizado de máquina. No capítulo 3, é descrita a metodologia aplicada na geração dos testes indiretos. Também são realizadas descrições resumidas dos algoritmos implementados. Já no capítulo 4, os resultados obtidos pelo método proposto são analisados. Por fim, no capítulo 5, juntamente com as conclusões retiradas desse projeto, são discutidos possíveis caminhos para a continuação do trabalho.

2 Revisão Bibliográfica

2.1 Teste de Circuitos Analógicos

Historicamente, os circuitos eletrônicos eram quase exclusivamente analógicos, sendo que seus componentes eram inteiramente discretos. Os circuitos eram montados em placas de circuito impresso e testados com testadores que possuíam acesso a todas as tensões de entrada e saída dos componentes. Como os dispositivos de um sistema eletrônico podiam ser testados individualmente, a rapidez na identificação da causa das falhas era mais importante (MILOR, 1998).

Com a evolução de técnicas de integração e encapsulamento, possibilitou-se a criação de circuitos muito maiores e numerosos. Consequentemente, a eficiência do teste passou a ser medida por dois parâmetros: tempo e cobertura de falhas (STRATIGOPOULOS; SUNTER, 2014). O teste deve ser rápido e eficaz, a fim de evitar atrasos na produção e detectar a maioria das prováveis falhas.

Os circuitos digitais se tornaram predominantes devido a sua facilidade de projeto e baixo consumo de energia. Isso não extinguiu a necessidade da utilização de circuitos analógicos, pois o sinal que emana de certos dispositivos, como sensores, é fundamentalmente analógico (MILOR, 1998).

Enquanto as metodologias de teste para circuitos digitais já estão bem estabelecidas, as técnicas de teste para CIs analógicos ainda são pouco padronizadas. Embora existam ferramentas computacionais para a geração automatizada de testes em CIs digitais há mais de três décadas, a determinação de sinais otimizados para testes em circuitos analógicos ainda não é totalmente automatizada. Normalmente, essa tarefa depende da experiência do projetista e de especificações sobre a funcionalidade do circuito (BUSHNELL; AGRAWAL, 2002).

Isso é explicado pelo fato que a representação de um sinal analógico pode assumir infinitos valores dentro de um intervalo de medição finito, ao contrário de sinais digitais, que retratam somente valores discretos (KABISATPATHY; BARUA; SINHA, 2005). Essa discrepância entre a natureza dos sinais ajuda a explicar o porquê de metodologias de testes para circuitos digitais estarem consolidadas no mercado, enquanto que os testes para circuitos analógicos não possuem um procedimento padrão. Um importante passo no teste analógico corresponde à determinação do limite que distinguirá um comportamento aceitável de um não aceitável para o circuito sob teste. Além disso, a existência de tolerâncias e não linearidades dos componentes dificulta ainda mais a criação de testes para esses circuitos (SRIMANI; GHOSH; RAHAMAN, 2016).

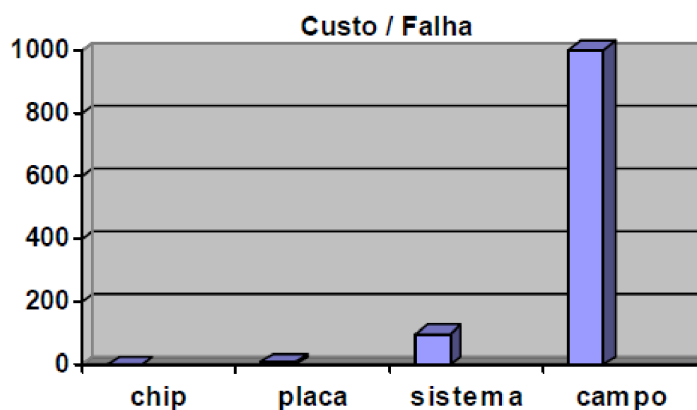
2.1.1 Teste em Produção

Testes em produção tem como objetivo principal averiguar o funcionamento e confiabilidade de um componente e são realizados o mais cedo o possível na cadeia de produção. Desse modo, a primeira etapa de teste é realizada no próprio *wafers* de silício (teste conhecido como *die sort*). O desempenho do circuito fabricado é comparado com as especificações do projeto. Caso o CI (*die*) falhe no teste, ele é descartado da linha de produção, impedindo que o circuito defeituoso seja encapsulado (MILOR; VISVANATHAN, 1987).

Além de garantir a qualidade do circuito produzido, o teste em produção tem como objetivo realimentar o processo de projeto, permitindo a identificação de erros ou problemas que podem comprometer o funcionamento do circuito. Com essa informação, é possível fazer ajustes no projeto ou no processo de produção, visando eliminar ou reduzir essas deficiências e melhorar a qualidade do produto final (BALEN, 2006).

De acordo com a teoria conhecida como *rule of ten* (BUSHNELL; AGRAWAL, 2002), o custo de reparo de uma falha pode crescer dez vezes a cada etapa de produção na qual não é identificada. A Figura 1 representa a estimativa de custo por falha não detectada. A fabricação de um circuito eletrônico possibilita aplicação de testes de produção em quatro momentos distintos: no *waffers*, nos CIs, nas placas de circuito impresso e no sistema.

Figura 1 – Custo relativo por falha em cada etapa da produção de um chip



Fonte: Balen, 2006

Circuitos analógicos possuem especificações que exigem uso de instrumentos analógicos de alta resolução, processamento de sinal digital e quantidade de memória para armazenar as respostas de teste (BUSHNELL; AGRAWAL, 2002). Por essa razão, a execução desses testes é feita através dos ATEs (*Automatic Test Equipments*), aparelhos de medição caros capazes de executar testes automatizados. Em consequência, o tempo de teste também é um fator determinante para seu custo final. Quanto menor é o tempo gasto para executar o teste, maior é o volume de CIs testados em um intervalo de tempo, reduzindo o custo do teste (assim como o preço final de fabricação do circuito) (SOUDERS; STENBAKKEN, 1990).

2.1.2 Teste Funcional *versus* Teste Estrutural

Os testes de circuitos podem ser classificados segundo seus objetivos. O teste funcional, também chamado de teste baseado em especificações (*specification-based test*), busca validar o funcionamento correto do circuito fabricado através de comparações com as especificações de projeto. Nesse contexto, se qualquer um dos parâmetros funcionais estiver fora dos limites de tolerância, o circuito é considerado falho e, conseqüentemente, é descartado. Dependendo da especificação sendo testada, essa técnica pode resultar em tempos longos de execução do teste, o que implica na elevação do custo do teste, que por sua vez é adicionado ao preço final do produto (SRIMANI; GHOSH; RAHAMAN, 2016).

Por outro lado, o teste estrutural, também conhecido como teste baseado em defeitos (*defect-based test*), procura identificar alterações na estrutura física do circuito. O método utilizado nesse teste se baseia em modelar um grande número de falhas que possam vir a ocorrer no circuito (RENOVELL, 2004). Em seguida, através de simulações, levanta-se o dicionário de falhas, que contém informações sobre quais falhas modeladas podem ser detectadas em quais testes. O maior obstáculo na execução desse tipo de teste é modelar as falhas de maneira precisa (ARSLAN; ORAILOGLU, 2013).

Geralmente, os testes de CIs analógicos seguem a metodologia de teste funcional, onde os limites de aceitação são estabelecidos pelas próprias especificações do *datasheet* (STRATIGOPOULOS; SUNTER, 2014). No entanto, a alta qualidade desse teste tem um custo de implementação consideravelmente maior do que o custo do teste de um circuito digital que ocupa a mesma área de um chip (POEHL *et al.*, 2010). Para tentar diminuir o custo dos testes funcionais, os próprios vetores utilizados para validação do circuito nas etapas de simulação do projeto podem ser utilizados no teste. Dessa forma, esses vetores já existem e não impõem nenhum custo adicional para sua geração. Em contrapartida, isso não é possível para o caso dos testes estruturais, cujos custos devem levar em consideração tanto a determinação dos vetores de teste, quanto a execução do teste propriamente dito (RENOVELL, 2004).

No entanto, para os CIs digitais é mais comum o emprego do teste estrutural, visto que a abstração dos valores assumidos nos nós do circuito ajudam na modelagem das falhas. O mesmo não pode ser dito no caso dos circuitos analógicos, devido às propriedades de sinais analógicos, como valores de amplitude e tempo infinitos, e a influência de não-idealidades, como deslocamento e ruído (BUSHNELL; AGRAWAL, 2002). Nesse sentido, é evidente que a busca por vetores para cobrir as falhas no circuito é uma tarefa mais custosa em testes analógicos. Isto motiva o estudo e pesquisa de novas metodologias, procurando encontrar modelos que possibilitem a padronização de testes, maximizando a eficiência na produção de CIs analógicos.

2.2 Teste Indireto

O teste funcional é geralmente usado em CIs analógicos para garantir uma alta qualidade dos dispositivos. Todavia, a necessidade de instrumentos de alta precisão e longos tempos de teste elevam consideravelmente o custo final de produção do circuito. Em razão disso, diferentes estratégias vêm sendo desenvolvidas para diminuir o custo dos testes de circuitos analógicos e manter um alto grau de confiabilidade no funcionamento do dispositivo. Metodologias de auto-teste integrado ou BIST (*Built-In Self-Test*), por exemplo, permitem que a geração de estímulos e análise de resposta sejam feitas internamente no circuito (CHATTERJEE; NAGI, 1997). Apesar disso, a área de silício consumida por circuito se torna maior, diminuindo a produtividade, e a inserção de elementos extras no circuito pode prejudicar o funcionamento do circuito.

Uma técnica que vem sendo estudada nas últimas duas décadas é o teste indireto. A principal motivação por trás de tal abordagem é diminuir a carga e flexibilizar as restrições sobre o equipamento de teste industrial para processar medidas de especificação de desempenho convencionais. Sem a necessidade de efetuar a medida de parâmetros funcionais, o tempo de execução do teste também é reduzido, o que diminui ainda mais o custo associado ao teste.

A abordagem de teste indireto foi introduzida na área de circuitos analógicos em (VARIYAM; CHATTERJEE, 1998) como teste alternativo (do inglês, *alternate test*). Essa estratégia se baseia na hipótese de que as variações do processo de fabricação têm influência em parâmetros funcionais, assim como em parâmetros que podem ser medidos por equipamentos de baixo custo, como amplitude de sinais AC e DC. Desse modo, é possível fazer o mapeamento dessas medidas indiretas (MIs) em especificações do circuito. O objetivo da busca dessa correlação é medir apenas esses parâmetros indiretos para verificar a performance do circuito sob teste (LEGER; BARRAGAN, 2018).

Entretanto, a relação entre as MIs e as especificações do circuito é muito complexa para construção de modelos analíticos, em grande parte devido à não linearidade presente em circuitos analógicos (BADAWI *et al.*, 2020). Para solucionar este problema, uma opção é a utilização de algoritmos de aprendizado de máquina.

As técnicas de aprendizado de máquina permitem que modelos estatísticos sejam criados automaticamente através da análise de grandes quantidades de dados, sendo capazes de identificar padrões complexos nos dados. Além disso, esses métodos lidam com dados em grande escala e alta dimensionalidade, o que pode ser um desafio para a criação de modelos analíticos. Porém, o uso de aprendizado de máquina também apresenta alguns desafios, como a necessidade de grandes quantidades de dados para treinar modelos precisos e a complexidade computacional dos algoritmos, que podem exigir recursos computacionais significativos para execução (BURKOV, 2019).

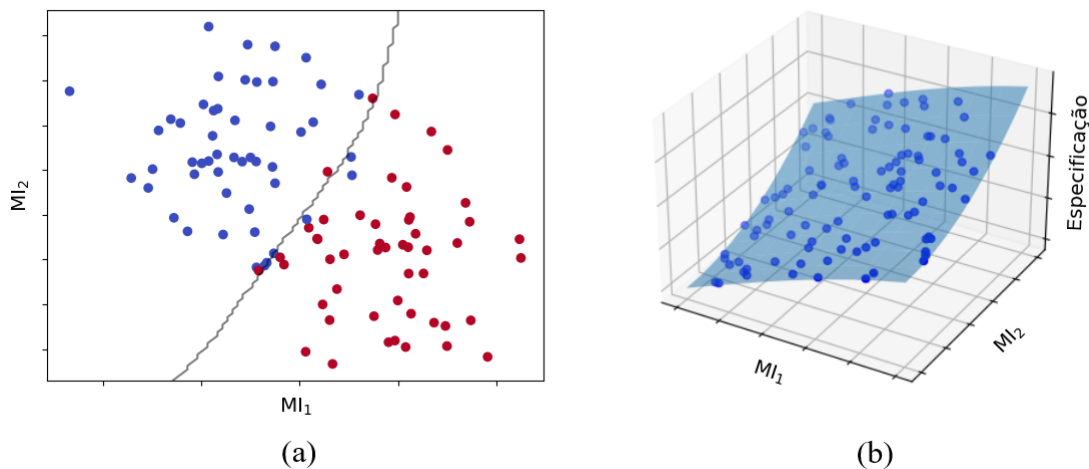
2.2.1 Testes Indiretos Orientados à Classificação e Predição

No contexto do aprendizado de máquina, o teste indireto é considerado um exemplo de aprendizado supervisionado. Neste tipo de aprendizado, o conjunto de dados (*dataset*) possui amostras com entradas X_i e saídas Y_j definidas. O propósito é treinar um modelo, representado por uma função $f_{X \rightarrow Y}$, capaz de prever as saídas de novas observações conhecendo apenas os valores de entrada, como mostra a Equação 1 (BURKOV, 2019).

$$f_{X \rightarrow Y} : [X_1, \dots, X_k] \rightarrow [Y_1, \dots, Y_l] \quad (1)$$

Dessa maneira, no teste indireto, as entradas (também chamadas de características) do modelo de aprendizado são as MIs. Já a saída (ou alvo) depende se o teste indireto é abordado como um problema de classificação ou regressão. Na Figura 2, é possível ver exemplos desses tratamentos ao teste indireto considerando duas MIs, MI_1 e MI_2 . Mesmo assim, ambos cenários podem ser vistos como uma tarefa de aproximação de funções (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Figura 2 – Exemplos de testes indiretos orientados à classificação (a) e predição (b)



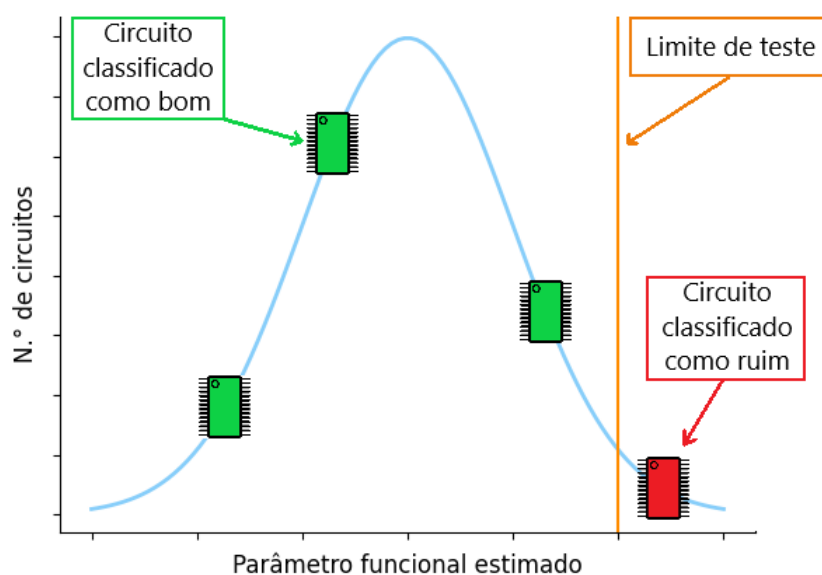
Fonte: o Autor, 2023

No caso do teste indireto voltado para classificação de circuitos, a saída do modelo de aprendizado é a condição do circuito. Os circuitos cujos parâmetros funcionais atendem as especificações de projeto são rotulados como bons, enquanto que os circuitos que não cumprem esses requisitos funcionais são classificados como defeituosos. Para que isso seja possível, é necessário conhecer previamente os limites das especificações, a fim de diferenciá-los. Nesta abordagem, o objetivo é treinar um modelo de aprendizado capaz de encontrar um limite de decisão no espaço de MIs que separa os circuitos e os classifica, como é mostrado na Figura 2(a). Em (STRATIGOPOULOS; MAKRIS, 2005), é visto uma aplicação do teste indireto, na qual foram treinadas redes neurais para criar limites de decisão não-lineares.

Por outro lado, o teste indireto pode também visar a estimação do parâmetro funcional de interesse. Para esse fim, uma função de regressão é ajustada para prever o valor da especificação, conforme ilustrado na Figura 2(b). Um exemplo do teste indireto como um problema de regressão é visto em (BARRAGAN; LEGER; HUERTAS, 2012), onde foi demonstrado que o emprego do teste indireto sobre múltiplas condições, como diferentes valores de tensão de alimentação e frequência de operação, aumenta a precisão da especificação estimada.

Todavia, a meta final do teste indireto ainda é realizar a diferenciação entre circuitos bons e defeituosos. Portanto, a especificação predita deve ser comparada com um limite de decisão em uma etapa posterior (BADAWI *et al.*, 2021). Na Figura 3, é apresentado como se dá a separação dos circuitos levando em conta um limite superior. Após a predição do parâmetro funcional pelo modelo de regressão, seu valor é comparado com o limite de teste. Desse modo, caso a especificação ultrapasse esse limiar, o circuito é considerado defeituoso e logo ele é descartado. Caso contrário, o circuito é tido como bom e continua na cadeia de fabricação.

Figura 3 – Identificação dos CIs pelo teste indireto orientado à predição



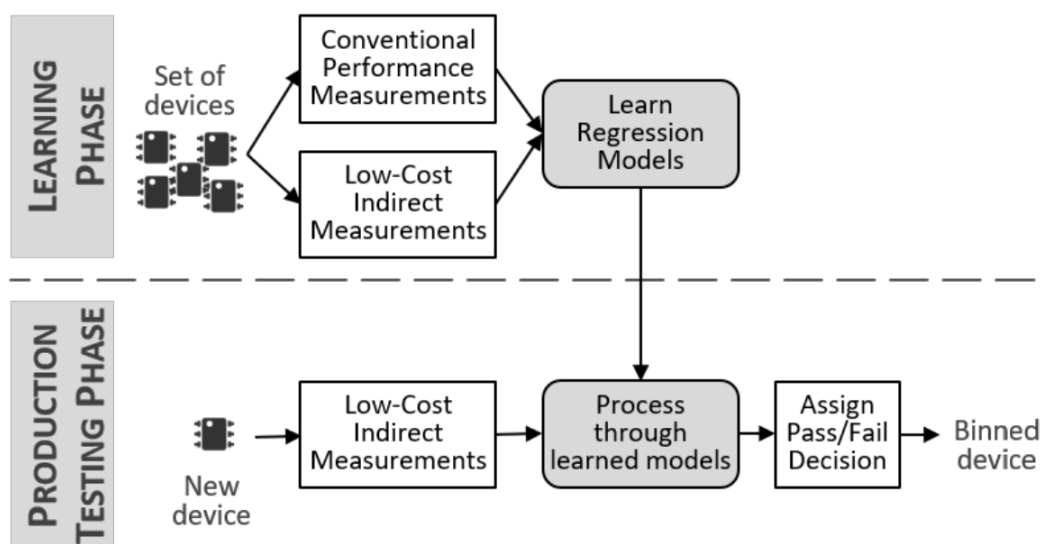
Fonte: o Autor, 2023

Embora essas duas perspectivas sobre o teste indireto sejam válidas, a abordagem com regressão possui vantagens importantes. Caso o limite de decisão entre um circuito bom e um defeituoso mude (devido ao ciclo de vida do CI, por exemplo) não será necessário retrainar o modelo de aprendizado, considerando que as variações do processo de fabricação que alteram o parâmetro funcional continuam as mesmas. Isso não é verdade em uma abordagem com classificação, onde o modelo deve ser retrainado para encontrar o novo limite de decisão no espaço de MIs. Outra vantagem de um teste indireto orientado à predição é o fácil diagnóstico e interpretação da especificação encontrada.

2.2.2 Arquitetura do Teste Indireto

A Figura 4 mostra como um teste indireto orientado à predição é implementado. A primeira fase é a fase de aprendizado. Nessa etapa, é encontrada as correlações entre as MIs e os parâmetros funcionais. Em um conjunto de dispositivos, são feitas tanto as medidas de baixo custo como os testes funcionais. Em seguida, modelos de aprendizado são treinados tendo como características as MIs e como saída as especificações. Já a segunda fase é o teste em produção, que acontece no processo de fabricação dos CIs, onde apenas medidas de baixo custo são executadas nos chips. Dessa forma, os parâmetros funcionais do circuito sob teste são preditos por meio dos modelos treinados na primeira fase. Por fim, o circuito é classificado como bom ou ruim tendo como referência os limites de especificação (BARRAGAN; LEGER, 2015).

Figura 4 – Arquitetura do teste indireto



Fonte: Badawi *et al.*, 2021

A fim de que o teste indireto seja executado de forma adequada, é essencial que o conjunto de amostras utilizadas na fase de aprendizado seja representativo das novas observações da fase de teste em produção. Isso garante que o modelo de regressão seja treinado com dados que reflitam a variabilidade do processo de fabricação e produza resultados confiáveis durante a fase de teste em produção.

Além disso, o modelo de regressão deve ser validado ainda na fase de aprendizado com métricas adequadas, a fim de garantir que ele seja capaz de prever as especificações de desempenho do circuito com alta precisão. Dessa maneira, é possível avaliar a precisão do modelo e, se necessário, ajustá-lo para melhorar o desempenho na fase de teste em produção. O modelo também deve ser examinado utilizando técnicas como validação cruzada e análise de resíduos, que permitem avaliar a capacidade do modelo de generalizar para novos dados e identificar possíveis problemas de sobreajuste.

2.3 Algoritmos de Regressão

Na escolha do melhor modelo de regressão para resolução de um determinado problema, é importante levar em consideração os atributos de cada modelo e avaliar qual é mais compatível com as propriedades dos dados de entrada. Tais modelos são amplamente empregados em variadas áreas do conhecimento, como: computação, engenharias, biologia, agronomia, saúde, etc. Para isso, a seleção do modelo adequado também deve contar com a experiência do projetista.

No contexto de algoritmos de regressão, aspectos relevantes conhecidos como subajuste (*underfitting*) e sobreajuste (*overfitting*) do conjunto de dados devem fazer parte da análise na escolha do modelo. O subajuste acontece quando o modelo não é capaz de capturar o padrão dos dados, devido a uma quantidade pequena de informações ou quando um modelo simples é selecionado para lidar com dados que possuem um comportamento complexo. Em contraste, o sobreajuste ocorre quando o modelo é treinado com dados que apresentam uma alta porcentagem de ruído, fazendo com que o resultado da identificação de um padrão do comportamento dos dados se torne enviesado (BURKOV, 2019).

Dessa maneira, modelos com alto viés e baixa variância tendem a ser mais adequados quando há um número baixo de amostras em relação ao número de entradas. Por outro lado, modelos com alta variância e baixo viés são mais apropriados quando há muitas amostras e poucas características (LEGER; BARRAGAN, 2018).

Sob a ótica de testes indiretos, onde o objetivo principal é reduzir o número de testes realizados, modelos que requerem muitas entradas podem não ser apropriados. Ainda, modelos mais complexos também tendem a apresentar um tempo de processamento maior, aumentando o tempo de teste. Por essas razões, é necessário levar em conta não apenas a precisão do modelo, mas também a eficiência e praticidade em sua utilização.

Em situações em que não é possível obter um grande conjunto de dados de treinamento para implementação do teste indireto, modelos que possam lidar com uma quantidade reduzida de amostras são mais indicados. Essa limitação pode ocorrer, por exemplo, em decorrência de uma baixa disponibilidade de CIs produzidos.

Sendo assim, nesta seção são apresentados os algoritmos de regressão abordados neste trabalho. Eles incluem modelos clássicos utilizados em problemas de regressão, bem como algoritmos que são frequentemente aplicados em outras referências da área de teste indireto. Além da explicação resumida de seus fundamentos teóricos, suas vantagens e desvantagens são destacadas.

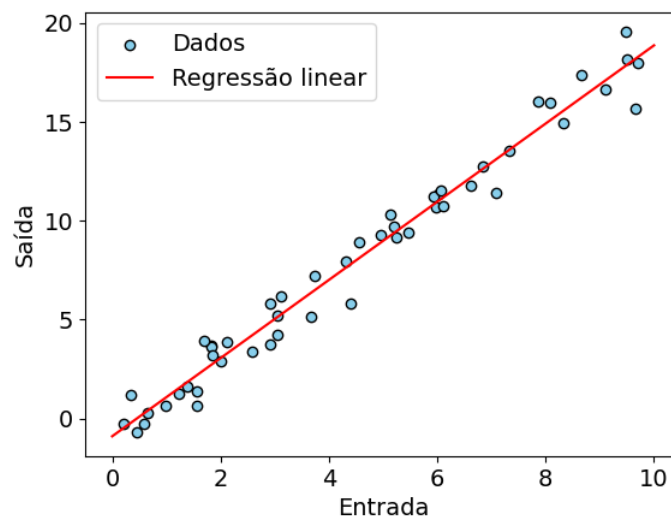
2.3.1 Regressão Linear Múltipla

O modelo de regressão linear múltipla (do inglês, *Multiple Linear Regressions - MLR*) apresenta uma relação linear entre a variável de saída e as variáveis de entrada. Este modelo faz previsões computando a soma ponderada das características, mais uma constante chamada de viés ou termo de interceptação, como mostra a Equação 2. As n variáveis de entrada x_i são multiplicadas pelos pesos θ_i e somadas junto do viés θ_0 para estimar a saída \hat{y} (GÉRON, 2019).

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (2)$$

Os coeficientes são ajustados de maneira a reduzir uma função custo, que mede a diferença entre os valores previstos pelo modelo e os valores observados (BURKOV, 2019). Em problemas de regressão, uma função custo normalmente utilizada é o erro médio quadrático, que será abordado mais profundamente na seção 2.5.1. Para minimizar a função custo, há diferentes estratégias que possuem suas vantagens e desvantagens, como por exemplo o algoritmo gradiente descendente, que busca otimizar a função de maneira iterativa (GÉRON, 2019). A Figura 5 exibe o resultado do treinamento de um simples modelo de regressão linear considerando uma entrada.

Figura 5 – Exemplo de aplicação de uma regressão linear simples



Fonte: o Autor, 2023

Devido a simplicidade analítica do modelo MLR, ele apresenta uma alta velocidade de processamento da saída. Além disso, mesmo que ele não seja capaz de modelar relações não lineares, em muitos casos sua performance supera a de modelos mais complexos, principalmente em situações de escassez de amostras ou alta quantidade de ruídos nos dados (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Em contrapartida, quando os dados apresentam uma alta variância, a regressão linear sofre de subajuste.

2.3.2 Splines de Regressão Adaptativa Multivariada

O modelo de *splines* de regressão adaptativa multivariada, conhecido como MARS (*Multivariate Adaptive Regression Splines*), é uma extensão dos modelos lineares capaz de modelar automaticamente não linearidades e lidar com interações entre as variáveis de entrada. Introduzida em (FRIEDMAN, 1991), esta regressão faz uso de expansões em funções de base linear por partes na forma da Equação 3. Essas funções, chamadas de funções de dobradiça, são definidas como *splines* lineares, que são curvas caracterizadas por nós.

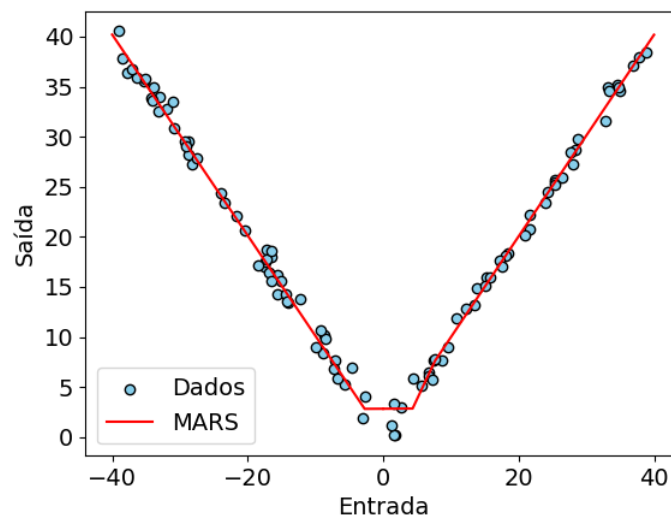
$$(x - t)_+ = \begin{cases} x - t, & \text{caso } x > t \\ 0, & \text{caso contrário} \end{cases} \quad (3)$$

A forma geral do MARS é apresentado na Equação 4. Os coeficientes β_0 e β_m são encontrados minimizando uma função custo, assim como uma regressão linear. O verdadeiro desafio do modelo está na construção das funções de dobradiças $h_m(x)$.

$$f(X) = \beta_0 + \sum_{m=1}^M \beta_m h_m(X) \quad (4)$$

O algoritmo possui duas etapas. Primeiro, é feita uma procura gananciosa por combinações de funções de base que minimizam localmente o erro no conjunto de treinamento. Em seguida, um subconjunto de termos é removido com base em uma pontuação de validação cruzada generalizada, penalizando a complexidade do modelo (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). A Figura 6 mostra a capacidade do modelo MARS.

Figura 6 – Exemplo de aplicação de um modelo MARS



Fonte: o Autor, 2023

O custo computacional do treinamento do MARS é elevado se comparado a uma regressão linear, porém ele possui a vantagem de considerar interações entre as características. Fora isso, o MARS apresenta um bom balanço entre viés e variância, sendo também capaz de realizar a predição rapidamente.

2.3.3 Máquina de Vetores de Suporte

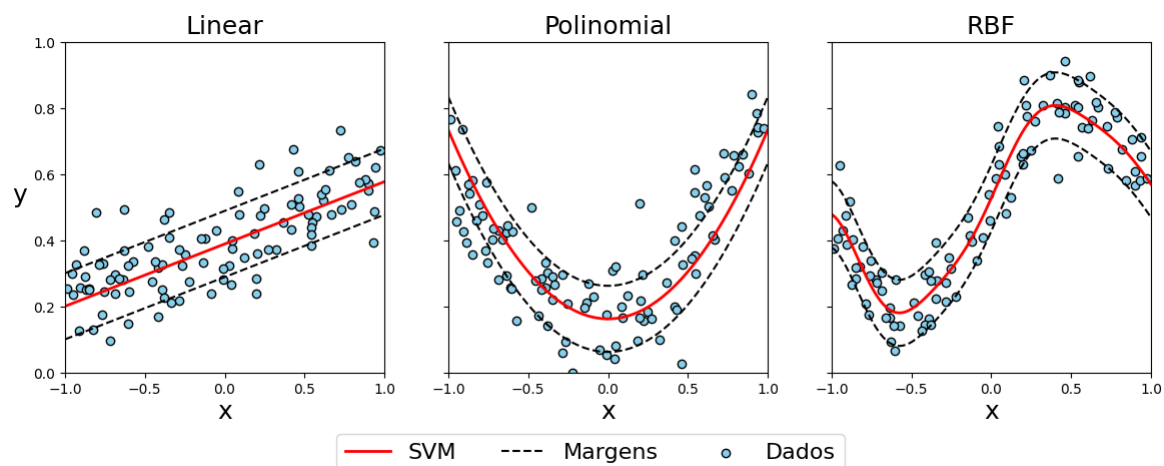
O modelo de máquina de vetores de suporte (*Support Vector Machines - SVM*) é um poderoso algoritmo de aprendizado bastante usado em problemas de classificação e regressão. Devido a sua versatilidade, ele permite modelar relações lineares ou não lineares (GÉRON, 2019). A Equação 5 exibe a função de regressão do SVM, onde \mathbf{w} é um vetor de pesos de mesma dimensionalidade do vetor de entradas \mathbf{x} e b é um termo de interceptação.

$$y = \mathbf{w}^T \mathbf{x} + b \quad (5)$$

Como é possível perceber, o modelo se assemelha ao de uma regressão linear. No entanto, ao invés de definir seus coeficiente de modo a minimizar o erro entre os valores reais e preditos, ele busca criar um hiperplano que contenha o máximo número de observações dentro de uma certa margem. As instâncias que pertencem as margens do hiperplano são chamados de vetores de suporte (BURKOV, 2019).

Caso não seja possível traçar um hiperplano linear para ajustar os dados, o SVM pode utilizar uma técnica matemática conhecida como truque de Kernel. Ele aplica uma transformação não linear do espaço, através de uma função de Kernel, mapeando o conjunto de dados para um espaço de maior dimensão onde é possível ajustar um hiperplano linear corretamente (BURKOV, 2019). Na Figura 7, são mostradas regressões com diferentes funções de Kernel lineares, polinomiais e de base radial (*Radial Basis Function - RBF*).

Figura 7 – Exemplos de aplicações de modelos SVM com diferentes funções de Kernel



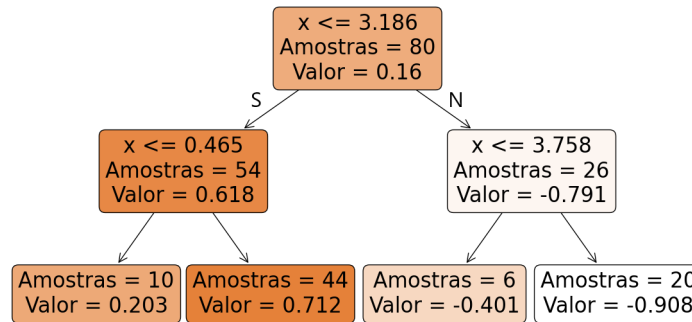
Fonte: o Autor, 2023

Uma das principais vantagens do SVM é sua performance eficiente em espaços de alta dimensionalidade. Ao mesmo tempo, ele é bastante suscetível a sobreajustes se não for corretamente parametrizado. Outro ponto negativo é que o número de coeficientes que define o hiperplano aumenta conforme aumenta o número de amostras no treinamento, o que eleva a complexidade computacional. Por essa razão, o SVM é mais adequado em casos onde o tamanho do conjunto de dados é pequeno ou médio (GÉRON, 2019).

2.3.4 Árvore de Decisão

Uma árvore de decisão, ou *Decision Tree* (DT), é um modelo usado para classificação e regressão, tendo o formato da Figura 8. A previsão para uma nova entrada é feita percorrendo a árvore da raiz até uma folha, seguindo as condições de divisão em cada nó. Assim, a saída do nó folha é o resultado da predição (BURKOV, 2019).

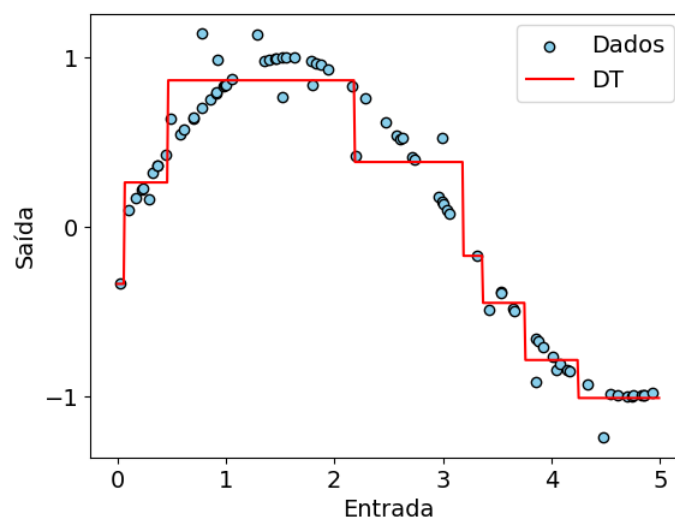
Figura 8 – Formato de uma DT



Fonte: o Autor, 2023

O treinamento da DT começa com um nó raiz que representa o conjunto de treinamento completo. Em seguida, divide-se os dados em dois subconjuntos menores. As condições são escolhidas de maneira a minimizar uma função custo nos subconjuntos criados. A cada divisão, um nó filho é criado para cada subconjunto menor, e o processo é repetido recursivamente em cada nó filho até que a árvore chegue em uma profundidade máxima ou não seja possível minimizar a função custo (GÉRON, 2019). A Figura 9 mostra o resultado de um modelo de DT.

Figura 9 – Exemplo de aplicação de uma DT



Fonte: o Autor, 2023

A DT modela relações não lineares e é fácil de interpretar, mas pode sobreajustar as amostras se os hiperparâmetros não forem definidos corretamente. Outra vantagem corresponde ao fato que o modelo não é afetado por diferenças nas escalas das entradas.

2.4 Seleção das Medidas Indiretas

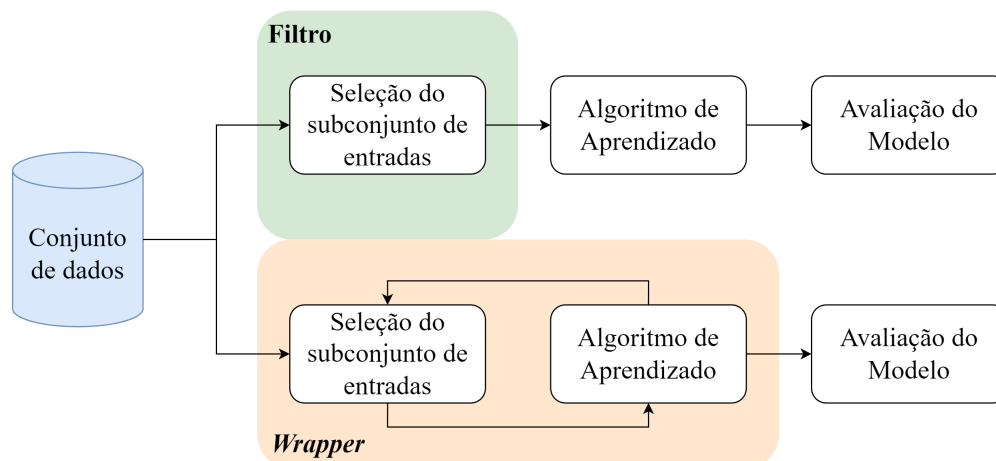
Independentemente do tipo de modelo de aprendizado, sua performance no teste indireto está atrelada à quais MIs estão disponíveis no conjunto de dados. A fim de realizar previsões significativas, é preciso que elas capturem a maior parte da variabilidade paramétrica do circuito. Portanto, a escolha de MIs pertinentes para o modelo é uma etapa crucial (BARRAGAN; LEGER, 2019).

Contudo, no teste indireto, o principal objetivo é reduzir o custo do teste. Logo, o modelo de aprendizado deve ser treinado com um número limitado de MIs. Além disso, se o número de entradas do modelo não for limitado, é possível que ocorra o fenômeno conhecido como *curse of dimensionality* (ou maldição da dimensionalidade), onde o erro de generalização do modelo aumenta conforme o número de entradas aumenta (GÉRON, 2019).

A escolha de um subconjunto de entradas dentro de um conjunto maior é uma fase importante no aprendizado de máquina. Esse processo recebe o nome de seleção de características (conhecido também como *feature selection*) e possui diversas estratégias para sua aplicação. Seu propósito é remover entradas redundantes e não informativas. Dessa maneira, o tempo de treinamento e a complexidade do modelo são reduzidos, além de em alguns casos resultar em um modelo mais preciso (KUHN; JOHNSON, 2013).

Neste trabalho, dois tipos de seleção de característica são abordadas: métodos de filtro e métodos *wrappers*. A Figura 10 apresenta como eles são usados no aprendizado de máquina. Nos métodos de filtro, a seleção é feita baseada apenas na relação entre entradas e saída, sendo efetuada antes do treinamento. Já nos métodos *wrappers*, a seleção é realizada treinando modelos com diferentes entradas e verificando quais aumentam a performance do modelo. Nas próximas seções, é explicado como cada método funciona e suas principais qualidades.

Figura 10 – Tipos de métodos de seleção de características



Fonte: o Autor, 2023

2.4.1 Métodos de filtro

Como visto anteriormente, os métodos de filtro são executados antes da etapa de treinamento, e portanto não possuem dependência com o modelo de aprendizado. Ao invés disso, a seleção é feita medindo a relevância das entradas em relação a saída através de testes estatísticos. As relações entre cada entrada e a saída são ordenadas segundo suas pontuações no teste estatístico. Então, um número arbitrário de entradas que melhor se relacionam com a saída são selecionadas (KUNH; JOHNSON, 2013).

Um teste estatístico simples, mas muito usado, é o coeficiente de correlação de Pearson, também conhecido como correlação populacional. Esta estatística mede a força da relação linear entre duas variáveis aleatórias. Como mostra a Equação 6, a correlação é definida pela covariância entre duas variáveis dividida pela multiplicação de seus desvios padrões. Assim, a correlação é simplesmente a normalização da covariância, que determina como duas variáveis se relacionam em termos de seus desvios em relação às suas médias. O valor do coeficiente de correlação varia entre 1 e -1, sendo 1 igual a uma relação linear positiva perfeita, e -1 igual a uma relação linear negativa perfeita. Caso o coeficiente seja igual a 0, isso indica que não há relação linear entre as variáveis (NAVIDI, 2020).

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \quad (6)$$

Uma outra opção de teste estatístico é a correlação de distância (SZÉKELY; RIZZO; BAKIROV, 2007). Diferentemente da correlação de Pearson, que só é capaz de medir relações lineares, essa estatística também é sensível a relações não lineares. Outro benefício é sua capacidade de considerar interações entre variáveis de entrada. A correlação de distância é baseada na ideia de comparar as distâncias euclidianas das observações em cada variável. A correlação entre duas variáveis aleatórias é obtida dividindo suas covariâncias de distância pelo produto de seus desvios padrões de distância, como mostra a Equação 7. O resultado desse teste varia entre 0 e 1, onde 0 indica independência entre as variáveis e 1 indica uma relação perfeitamente monotônica crescente ou decrescente.

$$dCor_{X,Y} = \frac{dCov(X,Y)}{\sqrt{dVar_X dVar_Y}} \quad (7)$$

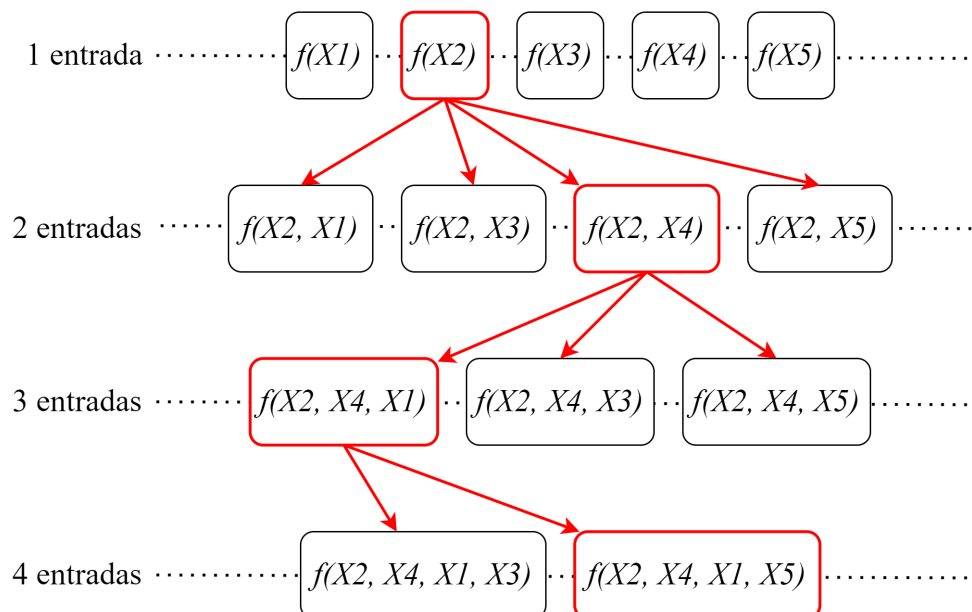
A principal vantagem da utilização dos métodos de filtro na seleção de características é seu baixo custo computacional, principalmente se comparados com os métodos *wrapper*. Porém, dependendo do teste estatístico aplicado, relações entre variáveis de entrada podem ser ignoradas. Ainda, por causa da falta de ligação entre o método e o modelo de aprendizado, a escolha da quantidade de entradas a serem selecionadas pelo método é subjetiva.

2.4.2 Métodos *Wrapper*

Os métodos *wrapper*, diferentemente dos métodos de filtro, utilizam o modelo preditivo para selecionar as melhores entradas. Eles consistem em treinar e avaliar modelos com diferentes conjuntos de características, escolhidos de maneira a minimizar uma função custo. Para realizar isso, esses métodos envolvem um processo ganancioso de seleção, em que diferentes combinações são testadas e o melhor subconjunto é escolhido (KUHN; JOHNSON, 2013).

Existem diferentes algoritmos para formar o melhor conjunto de características. Uma das estratégias mais simples e amplamente utilizadas é a seleção sequencial para frente (*Sequential Forward Selection* - SFS). Seu funcionamento é demonstrado na Figura 11. Esse procedimento começa com um conjunto vazio de características. Então, um modelo é treinado para cada característica no conjunto de dados. Por meio de alguma métrica, os modelos são comparados, e a entrada que gerou o modelo com melhor performance é adicionada no subconjunto de entradas. Na segunda iteração, modelos são treinados para cada par de características que incluem a entrada previamente selecionada. Novamente, os modelos são comparados e as entradas que criaram os melhores modelos são selecionadas. O algoritmo continua até que um critério de parada seja atingido, como por exemplo, um número determinado de características sejam selecionada (KUHN; JOHNSON, 2013).

Figura 11 – Funcionamento da estratégia de SFS



Fonte: o Autor, 2023

A principal vantagem dos métodos *wrapper* é que eles levam em consideração as interações entre as características, permitindo que sejam escolhidos conjuntos ótimos de variáveis para o modelo. Entretanto, essa abordagem pode ser computacionalmente custosa, especialmente quando alguma técnica de validação cruzada é usada para comparação dos modelos, tornando-a menos aplicável em casos com muitas entradas.

2.5 Métricas de Avaliação

Após a seleção das MIs mais relevantes e do modelo de regressão mais adequado para os dados e circuito sob teste, deve-se ser feita a avaliação dos modelos de predição criados. Essa análise é essencial para avaliar a eficácia da estratégia de teste indireto e para fornecer uma indicação da confiabilidade do modelo. No contexto do teste indireto orientado à predição, há várias métricas de avaliação relevantes que podem ser consideradas. Nas seções a seguir, são discutidas as métricas utilizadas neste trabalho para geração dos testes indiretos.

2.5.1 Raiz do Erro Médio Quadrático Normalizado

Uma das métricas mais aplicadas no contexto da regressão é a raiz do erro médio quadrático (*Root Mean Squared Error* - RMSE), apresentada na Equação 8 (RASCHKA, 2019). As variáveis y_i e \hat{y}_i representam o valor observado e o valor predito para inésima amostra, respectivamente, enquanto que n é o número total de dados observados. Nessa métrica, a exatidão do modelo é estimada pela raiz quadrada da média dos erros entre os valores observados e preditos pelo modelo elevados ao quadrado.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (8)$$

No entanto, no teste indireto orientado à predição é necessário criar modelos de aprendizado diferentes para cada parâmetro funcional de interesse. Como as especificações possuem diferentes escalas, não é adequado comparar diretamente os RMSEs de cada modelo. Para contornar esse problema, uma opção é realizar a normalização do RMSE, como demonstrado em (BADAWI *et al.*, 2021). Na Equação 9, o RMSE é normalizado pela média \bar{y} dos dados observados. Desse modo, define-se a raiz do erro médio quadrático normalizada, ou *Normalized Root Mean Squared Error* (NRMSE).

$$NRMSE = \frac{RMSE}{\bar{y}} \quad (9)$$

2.5.2 Coeficiente de Determinação

O coeficiente de determinação, também chamado de R^2 , é outra métrica que pode ser usada para avaliar modelos de regressão. O R^2 pode ser entendido como a fração da variância dos dados que o modelo consegue capturar. Ele é representado pela Equação 10. Quanto mais próximo de 1 for o R^2 , melhor é o ajuste do modelo em relação às amostras (RASCHKA, 2019).

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (10)$$

Embora o NRMSE também seja uma métrica normalizada que permite comparar variáveis de diferentes escalas, sua análise ainda é dependente da dispersão dos dados. Caso duas variáveis possuam dispersões muito diferentes, a comparação entre seus NRMSE perde o sentido. Por outro lado, o R^2 pode ser comparado diretamente entre variáveis diferentes. Por exemplo, um R^2 de 0,75 indica que 75% da variância na variável dependente é explicada pelas variáveis independentes incluídas no modelo.

2.5.3 Taxa de Erro de Classificação

Na área de teste de CIs, duas métricas bastante importantes são a perda na produtividade, ou *Yield Loss* (YL), e o nível de defeitos, ou *Defect Level* (DL). A perda na produtividade diz respeito ao percentual de circuitos que foram descartados no teste, mesmo estando funcionais. Já o nível de defeitos é o percentual de circuitos defeituosos que passam no teste (BUSHNELL; AGRAWAL, 2002).

Embora o teste indireto possa ser visto como um problema de regressão, mesmo assim o teste de CIs visa classificar circuitos como bons ou defeituosos. Definindo os limites de teste para as especificações do circuito, é possível fazer essa classificação, mesmo que a variável estimada seja o valor do parâmetro funcional.

Desse modo, a taxa de erro de classificação (*Misclassification Rate* - MR) de circuitos no teste indireto é dada pela Equação 11. Ela expressa a razão entre o número de circuitos classificados incorretamente e o número total de circuitos (BADAWI *et al.*, 2020).

$$MR = \frac{YL + DL}{N.^{\circ} \text{ de circuitos}} \quad (11)$$

3 Metodologia Experimental

Neste capítulo, são abordadas as etapas e escolhas de projeto realizadas durante o desenvolvimento deste trabalho. Primeiro, são descritas as ferramentas de *software* e *hardware* utilizadas. Em seguida, cada fase da metodologia de geração de testes indiretos é aprofundada, evidenciando os algoritmos criados. Também é apresentado o procedimento usado para medir a qualidade do teste indireto. Por fim, são introduzidos os estudos de casos usados para validar a geração de testes, descrevendo os circuitos utilizados e os parâmetros estruturais e funcionais considerados.

3.1 Ferramentas de *Software* e *Hardware*

Atualmente, a ferramenta de ATPG para testes estruturais que motivou este trabalho possui um protótipo desenvolvido em MATLAB® que se comunica com o simulador HSPICE®, ambos *softwares* pagos. Uma das linhas de pesquisa envolvendo a ferramenta consiste na sua portabilidade para um ambiente que inclua uma linguagem de programação e um simulador SPICE de código aberto. Por essa razão, visando uma futura integração com a ferramenta de testes estruturais, optou-se pelo desenvolvimento deste trabalho fazendo uso da linguagem Python e do simulador ngspice.

A linguagem Python é altamente utilizada em aprendizado de máquina devido seu grande número de bibliotecas que permitem treinar modelos de aprendizado. Além disso, por ser uma linguagem de alto nível, ela possui uma sintaxe de fácil entendimento.

O ngspice é um simulador de código aberto destinado à simulação de circuitos eletrônicos analógicos, digitais e mistos. Ele é baseado principalmente no simulador SPICE3, que é um programa livre referência na área de simulação de circuitos eletrônicos desde a década de 1970.

Vários pacotes de Python foram usados no desenvolver do trabalho. Eles são apresentados juntos de suas aplicações no decorrer das próximas seções. É importante salientar que a maioria das bibliotecas são construídas tendo como base os pacotes NumPy (HARRIS *et al.*, 2020) e Matplotlib (HUNTER, 2007).

Na Tabela 1, são informadas as versões de cada componente de *software*. Pacotes auxiliares dos quais as bibliotecas principais dependem foram ocultados.

Grande parte dos algoritmos foram implementados na plataforma Jupyter Notebook (KLUYVER *et al.*, 2016). O sistema operacional utilizado foi o Windows® 10. Todas simulações e programas foram rodados em um *notebook* de 64 bits com processador Intel® Core™ i5-6200U de 2.40 GHz e memória RAM de 8 GB.

Tabela 1 – Versões dos principais programas usados neste trabalho

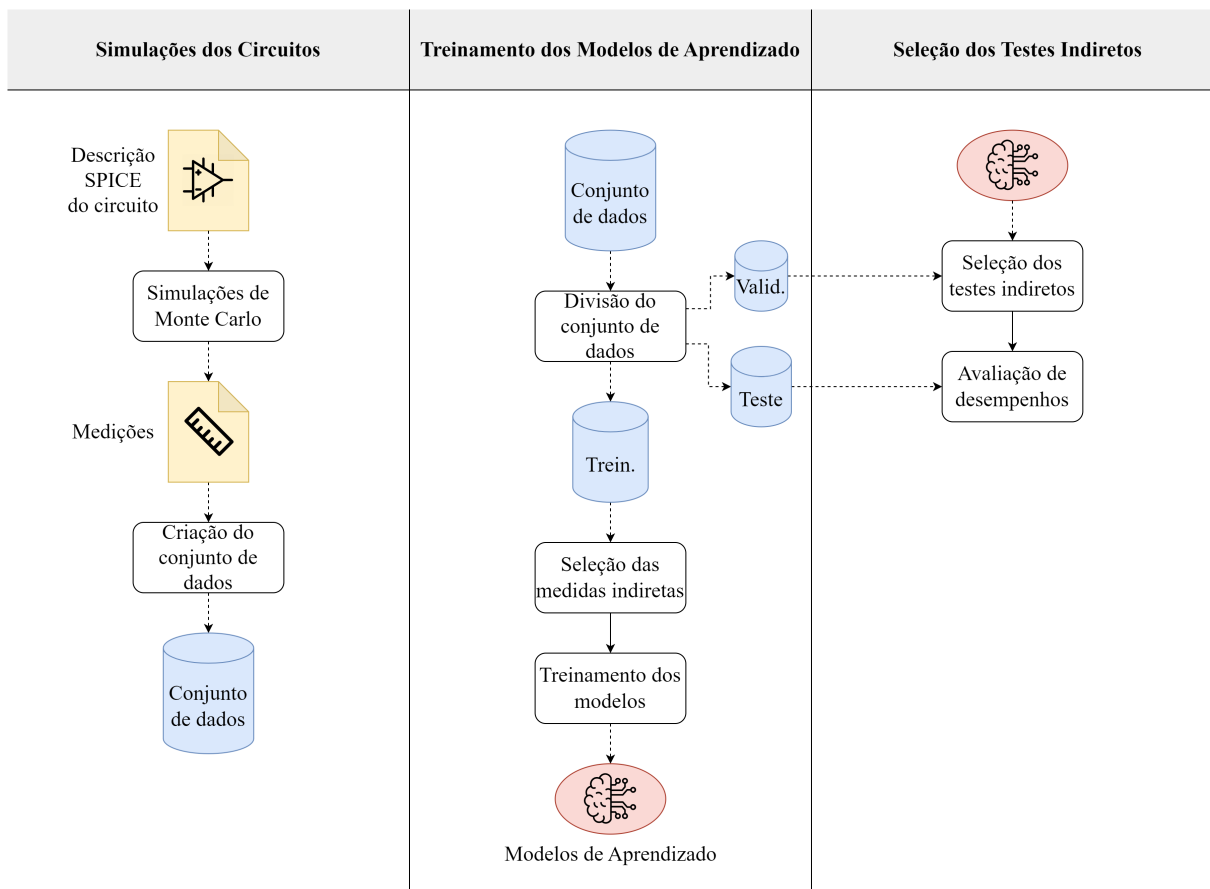
Programa	Versão
Python	3.8.10
ngspice	37
scikit-learn	1.2.1
pandas	1.5.3
seaborn	0.12.2
MLxtend	0.21.0
py-earth	0.1.0
dcor	0.6
clhs	1.0.2

Fonte: o Autor, 2023

3.2 Geração de Testes Indiretos

A Figura 12 mostra o método de geração de testes indiretos. A primeira fase é a criação de um conjunto de dados através de simulações de Monte Carlo (MC) do circuito. Após, vários modelos de aprendizado são treinados a partir desse *dataset*. Por fim, para cada especificação, escolhe-se um teste indireto por meio da performance dos modelos.

Figura 12 – Metodologia de geração de testes indiretos



Fonte: o Autor, 2023

3.2.1 Simulação dos Circuitos

A primeira fase da geração de testes indiretos consiste na criação de dados sintéticos a partir de simulações de MC do circuito sob teste. Para esse fim, é necessária a descrição do circuito na linguagem SPICE. Esse algoritmo, também chamado de *netlist*, reproduz em linhas de código todos os componentes do dispositivo e suas conexões.

Mais especificadamente, o arquivo com extensão ".sp" que contém a *netlist* deve estar escrito na linguagem SPICE referente ao simulador ngspice. Devido ao fato do SPICE3 ser um programa de código aberto, muitos outros simuladores da classe SPICE utilizam sintaxes semelhantes. Para evitar este problema, o ngspice oferece um modo de compatibilidade que permite ler códigos escritos em outras variações da linguagem SPICE. Essa funcionalidade é útil principalmente no uso de bibliotecas, que podem ser distribuídas pelo fabricante tendo em mente um simulador específico.

Para ser possível realizar as simulações, é essencial que a descrição do circuito inclua uma biblioteca que contenha os parâmetros de MC. Eles definem como as propriedades dos componentes de um circuito variam, e são inclusos no PDK (*Process Design Kit*) da tecnologia do CI, um conjunto de arquivos que modelam o processo de fabricação dos CIs. Assim, através desses parâmetros é possível reproduzir em simulação a variabilidade intrínseca do processo de fabricação.

As simulações são executadas por meio de um *script* escrito em SPICE, que é responsável por controlar as simulações e criar arquivos com os resultados das medições. Ele é resumido no pseudocódigo do Algoritmo 1.

Algoritmo 1: Simulações de Monte Carlo

Entrada: *netlist* do circuito sob teste

Saída: parâmetros estruturais e funcionais

```

1 rodadas_mc ← número de circuitos;
2 rodada ← 0;
3 repita
4   Carrega o circuito sob teste;
5   para cada análise faça
6     Executa a análise;
7     Calcula as especificações;
8     Escreve os parâmetros estruturais;
9     Escreve os parâmetros funcionais;
10  fim
11  rodada ← rodada + 1;
12 até rodada < mc_rodadas;
```

Primeiramente, define-se o número de rodadas de MC a serem executadas. Dentro de um laço de repetição, o circuito sob teste é carregado a partir de sua *netlist*. Nesse

momento, os parâmetros de MC são aleatorizados pelo simulador (respeitando o PDK da tecnologia), resultando, assim, em um circuito novo.

Para cada circuito gerado, são executadas três tipos de análise: AC, DC e transiente. Cada uma delas permite extrair especificações diferentes. Para calcular esses atributos, fez-se uso principalmente do comando *meas*, um método bastante útil em SPICE para efetuar medições complexas sobre as respostas obtidas de análises do circuito.

Após isso, os parâmetros estruturais, ou seja, os valores de sinais nodais do circuito, e os parâmetros funcionais são escritos em arquivos com extensão ".out". As especificações de todos os circuitos são salvas em um único arquivo. Por outro lado, os parâmetros estruturais são salvos em arquivos separados por circuito e análise. A diferença entre os armazenamentos das medições existe devido às limitações de escrita de arquivos da linguagem SPICE.

Esse procedimento de coleta de dados se repete até que a quantidade de circuitos determinada previamente sejam gerados. As faixas de varredura das análises AC, DC e transiente, assim como quais especificações são medidas dependem do circuito sendo testado. Portanto, elas são apresentadas nas próximas seções.

Após a conclusão da aquisição dos parâmetros estruturais e funcionais dos circuitos, esses dados são extraídos dos arquivos de saída das simulações e armazenados em um conjunto de dados para facilitar a visualização e manipulação. Essa tarefa é feita com o auxílio de um *script* escrito em Python. No Algoritmo 2, é visto seu código resumido.

Algoritmo 2: Criação do conjunto de dados

Entrada: parâmetros estruturais e funcionais

Saída: conjunto de dados

- 1 **para cada** *medições* **faça**
 - 2 Lê arquivos com *medições*;
 - 3 Cria um *DataFrame*;
 - 4 Formata o *DataFrame*;
 - 5 **fim**
 - 6 Funde os *DataFrames* de cada *medições*;
 - 7 Salva o conjunto de dados;
-

O algoritmo lê as medições presentes nos arquivos de saída da simulação e os armazena em um *DataFrame* do pacote pandas (MCKINNEY, 2010). Essa estrutura de dados é basicamente uma tabela de duas dimensões dividida em linhas e colunas. O *DataFrame* é então formatado de modo que cada linha represente um circuito, e cada coluna represente um parâmetro estrutural ou funcional diferente. É formado um *DataFrame* para cada tipo de medição do circuito (AC, DC, transiente ou funcional). Em seguida, os *DataFrames* de cada tipo de medição são fundidos em um único *DataFrame* a fim de criar o conjunto de dados, o qual é salvo em um arquivo com extensão ".csv".

3.2.2 Treinamento dos Modelos de Aprendizado

O *dataset* formado na etapa anterior inclui os parâmetros estruturais e funcionais de um determinado número de amostras de circuitos. Neste trabalho, optou-se pela geração de testes indiretos orientados à predição, tendo em vista suas vantagens descritas na seção 2.2.1. Dessa forma, objetiva-se treinar modelos de aprendizado que permita encontrar esses parâmetros funcionais a partir dos parâmetros estruturais (ou MIs).

No entanto, há inúmeras alternativas para criação do modelo, como por exemplo, quantas e quais variáveis de entrada utilizar. Assim sendo, esta fase da metodologia é destinada à criação de diferentes modelos de aprendizado. Ao seu final, é obtido um conjunto de modelos treinados variando o tipo de modelo, a técnica de seleção de características empregada e o número de entradas usadas no modelo.

Todos os procedimentos descritos a partir dessa fase foram desenvolvidos em código Python na plataforma Jupyter Notebook, com o objetivo de tornar a organização e visualização dos dados mais práticas. Como pode ser visto a seguir, a metodologia adotada faz amplo uso de métodos relacionados a aprendizado de máquina disponíveis no pacote scikit-learn (PEDREGOSA *et al.*, 2011). Além disso, a maioria dos gráficos foram gerados com o auxílio da biblioteca seaborn (WASKOM, 2021).

O primeiro passo é a divisão do conjunto de dados em conjuntos de treinamento, validação e teste. Como o nome indica, o conjunto de treinamento é usado na aprendizagem dos modelos. Já os conjuntos de validação e teste são usados para avaliar a capacidade dos modelos de estimar novas instâncias. Portanto, nesta etapa da metodologia é utilizado apenas o conjunto de treinamento.

Os dados são separados em diferentes conjuntos através da amostragem por hiper-cubo latino condicionada, ou *conditioned Latin Hypercube Sampling* (cLHS) (MINASNY; MCBRATNEY, 2006). Essa técnica permite reproduzir as mesmas propriedades estatísticas do conjunto original nos conjuntos resultantes, possibilitando uma avaliação mais justa do desempenho de um modelo. A vantagem da cLHS é a sua habilidade de fazer uma amostragem estratificada considerando múltiplas variáveis ao mesmo tempo. Essa propriedade é relevante no teste indireto, onde geralmente pretende-se estimar mais de uma especificação. Para implementar essa técnica em Python, fez-se uso do pacote clhs (WAGONER; ZHENG, 2019).

Antes de iniciar os treinamentos dos modelos de aprendizado, contudo, é preciso garantir que todas as entradas estejam em uma mesma escala. Essa fase de pré-processamento é conhecida como normalização de características. Ela é necessária pois muitos algoritmos são sensíveis às magnitudes das variáveis de entrada, tornando certas características mais importantes que outras e resultando em perda de informação.

Nesse trabalho, a normalização é feita padronizando as características por meio do objeto *StandardScaler* do pacote *scikit-learn*. Dessa forma, todas variáveis de entradas passam a ter uma média igual a zero e variância unitária. Esse procedimento é feito para as entradas dos três subconjuntos de dados, tendo como base os valores de amostras do subconjunto de treinamento.

Após a normalização dos valores de entrada, é realizada a seleção das MIs usadas para treinar os modelos. Nessa etapa, são aplicadas diferentes técnicas no conjunto de treinamento para selecionar subconjuntos de características que mais se relacionam com as especificações. O Algoritmo 3 resume como esse procedimento é efetuado.

Algoritmo 3: Seleção das medidas indiretas

Entrada: conjunto de treinamento
Saída: MIs selecionadas

```

1  $K \leftarrow$  máximo número de entradas;
2 para cada especificação faça
3   para cada medida estatística faça
4     Calcule a medida estatística entre cada MI e a especificação;
5     para  $k = 1, \dots, K$  faça
6       Selecione  $k$  melhores características;
7     fim
8   fim
9   para cada modelo faça
10     Realize SFS visando estimar a especificação através de um modelo;
11     para  $k = 1, \dots, K$  faça
12       Selecione as características da iteração  $k$ ;
13     fim
14   fim
15 fim

```

Primeiramente, é definido o número de MIs a serem selecionadas pelo algoritmo. Então, todas estratégias de seleção são aplicadas para cada especificação do conjunto de dados. As MIs escolhidas durante esse processo são armazenadas em estruturas de dicionários Python. Dois tipos de seleção de características são empregadas: métodos de filtro e métodos *wrapper*.

Nos métodos de filtro, as MIs são ordenadas de acordo com uma medida estatística da relação entre entrada e saída. Esse procedimento é feito a partir do objeto *SelectKBest* do pacote *scikit-learn* para cada medida estatística sendo considerada. Após isso, subconjuntos com diferentes números de entradas são criados selecionando as melhores características da ordenação.

Neste trabalho, é feito o uso da correlação de Pearson e a correlação de distância univariada como medidas estatísticas. Enquanto a primeira permite medir somente relações lineares, a correlação de distância também é capaz de medir relações não lineares. A

correlação de Pearson é calculada indiretamente através do método *f_regression* da scikit-learn. Já a correlação de distância é implementada a partir do pacote dcor (RAMOS-CARREÑO; TORRECILLA, 2023).

Por outro lado, os métodos *wrapper* são executados pelo método SFS, através do objeto *SequentialFeatureSelector* da MLxtend (RASCHKA, 2018). Como o propósito da SFS é otimizar a resposta de um modelo de aprendizado, a seleção de características se altera conforme o modelo usado. Neste trabalho, foram escolhidos três modelos com propriedades distintas: MLR, MARS e SVM com função RBF. Durante esse processo, é usada validação cruzada para estimar os erros dos modelos.

O último passo desta etapa da metodologia é o treinamento de diferentes tipos de algoritmos de aprendizado para todos subconjuntos de MIs selecionadas. O pseudocódigo do Algoritmo 4 mostra o treinamento dos modelos de aprendizados, onde cada modelo possui uma diferença na sua construção. No código em Python, todos objetos criados são organizados em uma estrutura de dicionário.

Algoritmo 4: Treinamento dos modelos

Entrada: conjunto de treinamento, MIs selecionadas

Saída: modelos de aprendizado

```

1  $K \leftarrow$  máximo número de entradas;
2 para cada especificação faça
3   para cada modelo faça
4     para cada seleção faça
5       para  $k = 1, \dots, K$  faça
6         Treina modelo com  $k$  MIs da seleção para predir a especificação;
7       fim
8     fim
9   fim
10 fim
  
```

Neste trabalho, os modelos usados para estimar as especificações são: MLR, MARS, SVM (com função RBF) e DT. Os algoritmos de aprendizado MLR, SVM e DT pertencem a biblioteca scikit-learn, sendo representados pelos estimadores *LinearRegression*, *SVR* e *DecisionTreeRegressor*, respectivamente. Porém, para aplicar o modelo MARS foi necessário fazer uso da implementação disponível no pacote py-earth (LIAO, 2022).

É importante ressaltar que no treinamento de modelos de aprendizado, é importante testar diferentes hiperparâmetros do modelo para encontrar a melhor solução para o problema. Essa exploração não é feita neste trabalho, visto que o objetivo final é apresentar um fluxo de geração de testes indiretos, e não buscar pelos melhores resultados possíveis. Por essa razão, os modelos foram utilizados com seus valores padrão de hiperparâmetros definidos nos pacotes.

3.2.3 Seleção dos Testes Indiretos

Na última etapa da metodologia, todos modelos treinados previamente são validados. A partir desses resultados, é selecionado o modelo com maior desempenho (juntos de suas entradas) para estimar cada especificação, definindo, assim, a melhor arquitetura de teste indireto para o problema.

O Algoritmo 5 demonstra como é feita a busca do melhor modelo para cada especificação. Os dados do conjunto de validação são usados para calcular uma métrica de avaliação do modelos, que quantificam a capacidade de predirar novas observações. A métrica usada nesse processo é o NRMSE. Desse modo, os modelos que apresentam melhor desempenho para cada especificação são buscados iterativamente, armazenando aqueles que apresentam menor valor de NRMSE. Ao final, são obtidos os algoritmos de aprendizado e as MIs que possibilitam estimar com maior confiança cada especificação do circuito sob teste.

Algoritmo 5: Seleção dos testes indiretos

Entrada: conjunto de validação, modelos de aprendizado

Saída: testes indiretos selecionados

```

1 para cada especificação faça
2   | melhor modelo  $\leftarrow \emptyset$ ;
3   | menor NRMSE  $\leftarrow \emptyset$ ;
4   | para cada modelo faça
5   |   | Calcula NRMSE para avaliar a predição da especificação pelo modelo;
6   |   | se NRMSE atual < menor NRMSE ou menor NRMSE =  $\emptyset$  então
7   |   |   | melhor modelo  $\leftarrow$  modelo;
8   |   |   | menor NRMSE  $\leftarrow$  NRMSE atual;
9   |   | fim
10  | fim
11 fim

```

Após a seleção dos testes indiretos, os melhores modelos são retreinados em um conjunto de dados que é a soma dos conjuntos de treinamento e validação. Como mais amostras são apresentadas aos algoritmos de aprendizado, esse processo objetiva aumentar a eficiência dos modelos. Posteriormente, o conjunto de teste é utilizado para estimar a performance dos modelos retreinados por meio das métricas NRMSE e R^2 . Através desses resultados, é representado o desempenho final da predição teste indireto.

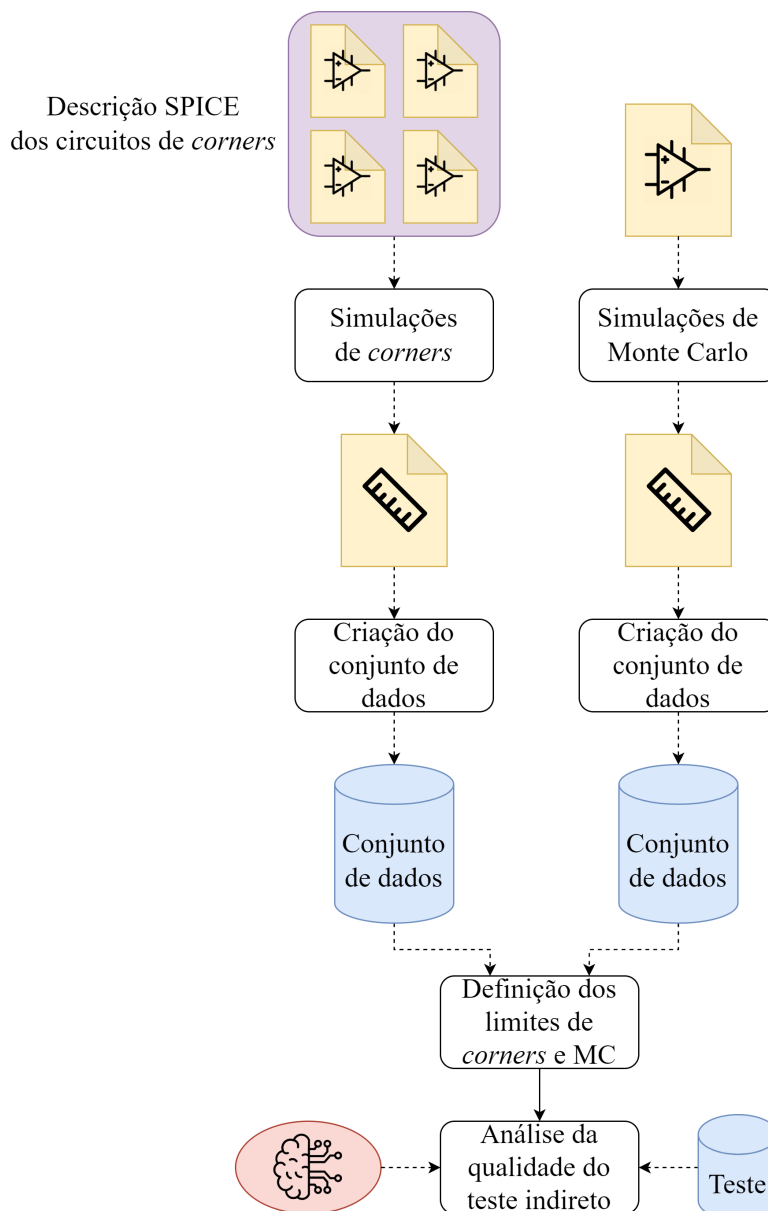
Sendo assim, considerando determinadas especificações do circuito sob teste, essa metodologia indica quais MIs e qual algoritmo de aprendizado dentre as opções propostas são mais adequados para a implementação do teste indireto. Além disso, a partir das métricas de avaliação dos modelos finais, o engenheiro de teste é capaz de verificar a viabilidade da substituição do teste tradicional pelo teste indireto.

3.3 Análise da Qualidade do Teste Indireto

Uma verificação importante que não foi abordada na metodologia da última seção é a capacidade do teste indireto de diferenciar circuitos bons e defeituosos. De fato, no teste indireto orientado à predição objetiva-se encontrar o valor do parâmetro funcional de interesse do circuito sob teste. Contudo, visando sua aplicação em uma cadeia de produção, essa distinção deve ser feita após a etapa de estimação.

A Figura 13 apresenta a metodologia aplicada para definir os limites de teste e estimar a qualidade do teste indireto. Primeiro, os limites de especificação são definidos. Então, a qualidade do teste, representada pela métrica MR, é obtida com o auxílio dos melhores modelos e o conjunto de teste definidos durante a geração de testes indiretos.

Figura 13 – Método de análise da qualidade do teste indireto



Fonte: o Autor, 2023

Para classificar os circuitos, é preciso primeiramente estipular limites para suas especificações. Duas abordagens diferentes são usadas para realizar essa tarefa: *corners* e MC. Ambos métodos requerem a simulação de circuitos e a criação de um conjunto de dados. O método por *corners* gera menos simulações se comparado à técnica de MC, contudo ela normalmente leva a condições excessivamente otimistas ou pessimistas quanto às variações dos circuitos.

Como visto anteriormente, a análise de MC define os parâmetros aleatoriamente seguindo uma certa distribuição. Já na análise de *corners*, os parâmetros dos componentes do circuito são alocados nos limites de variação definidos pela tecnologia. Dessa maneira, são explorados os casos extremos de variabilidade do processo de fabricação. Cada *corner* simulado representa um pior caso de fabricação. Neste trabalho, são considerados os *corners* representados pelas siglas SS (*Slow-Slow*), SF (*Slow-Fast*), FS (*Fast-Slow*), e FF (*Fast-Fast*), onde a primeira letra representa a condição de “rápido” ou “lento” aplicada aos transistores.

Visualizando a Figura 13, fica evidente que o processo adotado para simular os circuitos e criar o conjunto de dados nas duas análises é semelhante ao método discutido na geração de testes indiretos. Uma das alterações reside na simulação dos circuitos de *corners*. Nessa análise, os *corners* a serem simulados são definidos pelas bibliotecas da tecnologia importadas pela *netlist*. Portanto, é necessário haver um arquivo de descrição diferente para cada *corner*. Além dessa diferença, ao invés de serem gerados arquivos de saída com os parâmetros estruturais e funcionais do circuito, apenas as especificações são salvas, visto que deseja-se traçar os limites apenas nelas.

A partir dos dois conjuntos de dados, é feita a definição dos limites de *corners* e MC para cada especificação. Por análise de *corners*, os limites são determinados encontrando os valores máximo e mínimo que o parâmetro funcional assume considerando todos os *corners* da tecnologia. Por outro lado, no procedimento com MC, primeiramente a média e o desvio padrão do parâmetro funcional são calculados usando as amostras de circuitos coletadas. Em seguida, os limites são definidos por uma faixa de desvios padrões em torno da média.

Após a definição dos limites de teste, a qualidade do teste indireto gerado pela metodologia da seção anterior é analisada. Com esse propósito, é feito uso dos melhores modelos de aprendizado e o conjunto de teste formados durante a geração de testes indiretos. As especificações são comparadas com seus limites de modo a classificar as amostras de circuitos como bons ou defeituosos, permitindo calcular a taxa de erro de classificação dos modelos. Desse modo, é possível ter uma ideia da produtividade geral do processo de fabricação caso o teste indireto gerado fosse aplicado.

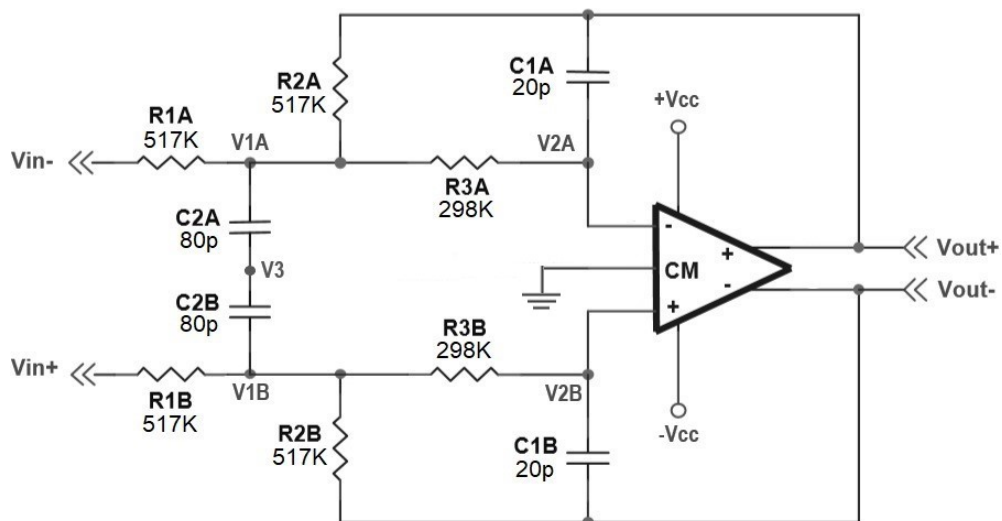
3.4 Estudos de Caso

Dentro desta seção, são apresentados os circuitos analógicos usados como estudos de casos. O método de geração de testes indiretos é aplicado nos circuitos para definir uma arquitetura de teste indireto que permita testar cada uma das especificações. Para medir a qualidade do teste indireto, foram determinados os limites de teste de acordo com a metodologia da seção anterior.

3.4.1 Filtro Passa-Baixas de Um Estágio

O primeiro estudo de caso é um filtro passa-baixas de primeira ordem na topologia de realimentação múltipla elaborado na tecnologia X-FAB 180nm. A Figura 14 mostra seu esquemático. Ele reproduz um filtro Butterworth com frequência de corte (também denominada frequência de -3 dB) em 10 kHz e ganho unitário na banda de passagem. Esse circuito analógico também foi usado como estudo de caso para validação da ferramenta de ATPG desenvolvida em (ZILCH; LUBASZEWSKI; BALEN, 2022).

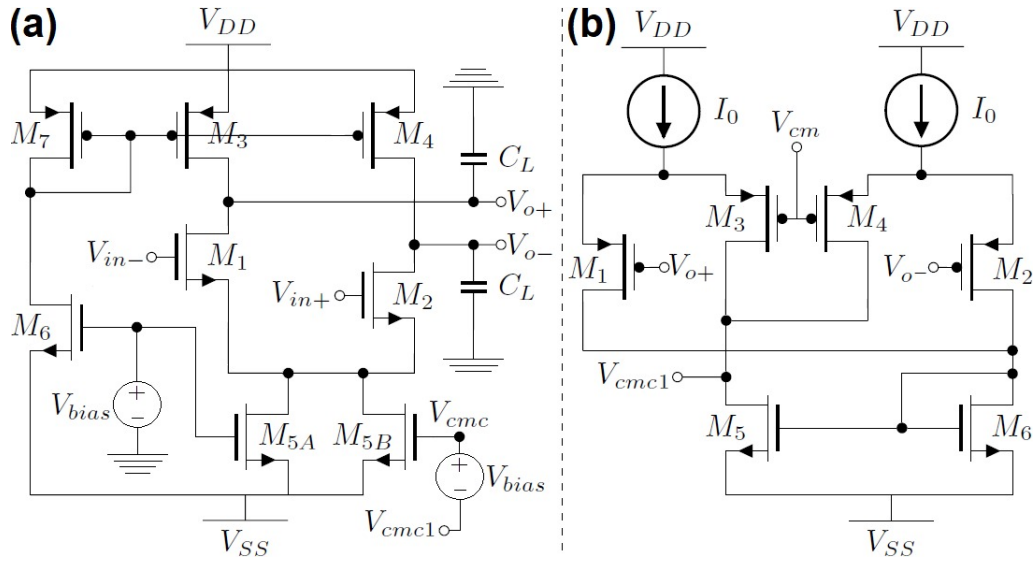
Figura 14 – Diagrama esquemático do filtro passa-baixas de um estágio



Fonte: adaptado de Zilch *et al.*, 2022

O filtro é implementado por meio do amplificador totalmente diferencial (*Fully Differential Amplifier* - FDA) projetado em (OLIVEIRA; SEVERO; GIRARDI, 2014). Esse modelo de amplificador possui uma saída diferencial, sendo utilizado principalmente no condicionamento de sinais para a entrada de ADCs (Analog to Digital Converters) (KARKI, 2002). Ele é constituído por dois blocos de circuito, como mostra a Figura 15. O bloco chamado de *Common-Mode Feedback* (CMFB), exibido na Figura 15(b), é responsável por estabilizar, em um valor desejado, a tensão de modo comum nas saídas. Já o bloco de ganho, representado na Figura 15(a), é o amplificador operacional propriamente dito.

Figura 15 – Blocos de ganho (a) e CMFB (b) do FDA



Fonte: Zilch *et al.*, 2022

Em (OLIVEIRA; SEVERO; GIRARDI, 2014), as dimensões dos transistores e os valores das fontes de tensão e corrente foram definidos por uma ferramental de otimização automática apresentada no trabalho. As Tabelas 2 e 3 apresentaram os parâmetros dos componentes de cada bloco do circuito.

Tabela 2 – Valores dos parâmetros dos componentes do bloco de ganho do FDA

Parâmetro	Valor
W_1/L_1 ($\mu\text{m}/\mu\text{m}$)	39,29/0,197
W_2/L_2 ($\mu\text{m}/\mu\text{m}$)	39,29/0,197
W_3/L_3 ($\mu\text{m}/\mu\text{m}$)	27,41/6,83
W_4/L_4 ($\mu\text{m}/\mu\text{m}$)	27,41/6,83
W_5/L_5 ($\mu\text{m}/\mu\text{m}$)	15,3/9,75
W_6/L_6 ($\mu\text{m}/\mu\text{m}$)	15,3/9,75
W_7/L_7 ($\mu\text{m}/\mu\text{m}$)	27,41/6,83
V_{bias} (mV)	-167,045

Fonte: adaptado de Oliveira *et al.*, 2014

Tabela 3 – Valores dos parâmetros dos componentes do bloco de CMFB do FDA

Parâmetro	Valor
W_1/L_1 ($\mu\text{m}/\mu\text{m}$)	35,91/0,8918
W_2/L_2 ($\mu\text{m}/\mu\text{m}$)	35,91/0,8918
W_3/L_3 ($\mu\text{m}/\mu\text{m}$)	35,91/0,8918
W_4/L_4 ($\mu\text{m}/\mu\text{m}$)	35,91/0,8918
W_5/L_5 ($\mu\text{m}/\mu\text{m}$)	7,19/0,4628
W_6/L_6 ($\mu\text{m}/\mu\text{m}$)	7,19/0,4628
I_0 (μA)	15,19

Fonte: adaptado de Oliveira *et al.*, 2014

As especificações escolhidas para representar o funcionamento do circuito no teste indireto foram o ganho DC, a frequência de corte e a velocidade de varredura, mais conhecida como *slew rate* (SR). Apesar do FDA possuir uma saída diferencial, nada impede que ele seja usado como um circuito de saída simples. Portanto, neste trabalho, as especificações são medidas apenas no nó de saída V_{out+} .

Como MIs, são consideradas apenas medidas de tensão DC do circuitos sob teste. Essa limitação foi feita pois, em geral, medições de sinais DC são mais atrativas se comparadas às medições AC. Por meio de técnicas de Design for Testability (DfT), ou projeto visando a testabilidade, por exemplo, é possível realizar o monitoramento interno de sinais DC do circuito de maneira simples.

Tendo em mente essa limitação, é evidente que o nó $V3$ não é um candidato a um nó de teste do circuito, visto que os capacitores $C2A$ e $C2B$ impedem a passagem de corrente nesse ponto durante a análise DC. Sendo assim, os nós de observação do circuito para realização das MIs são: $V1A$, $V1B$, $V2A$, $V2B$, V_{out+} e V_{out-} .

Ainda, para executar as MIs, o circuito é excitado com uma tensão de entrada, aplicada entre os nós V_{in-} e V_{in+} , que varia entre -0,9 V a 0,9 V, com passo de 0,1 V. Portanto, considerando os 6 nós de observação e os 19 possíveis valores de tensão de entrada, são definidas 114 MIs que são utilizadas no método de geração de testes indiretos.

Nas análises DC das simulações, não foi preciso extrair nenhuma especificação. Assim, a entrada foi variada de modo a produzir as MIs mencionadas anteriormente. Porém, para calcular as especificações com precisão, são necessários vários pontos de simulação. Por essa razão, nas análises AC, onde foram encontrados o ganho DC e a frequência de corte, a frequência de entrada foi variada entre 10 Hz a 100 MHz, com 100 pontos por década. Já na análises transientes, que permitiram calcular o SR dos circuitos, foram aplicados saltos com amplitude de 0 V a 0,9 V, com um tempo total de simulação de 100 μ s e passo igual a 0,01 μ s.

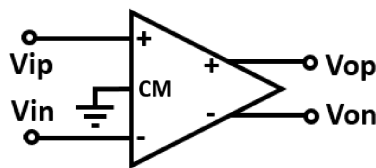
Foram criadas 1000 amostras de dados a partir de simulações de MC do circuito. O conjunto de dados resultante foi dividido em três subconjuntos: treinamento, validação e teste. A proporção da quantidade original de amostras para cada subconjunto foi de 64% para treinamento, 16% para validação e 20% para teste.

Na análise da qualidade do teste indireto, foram aplicados os limites de teste definidos por MC. O mesmo *dataset* da geração de testes indiretos foi usado para calcular as médias e desvios padrões das especificações. Assim, para cada especificação, os limites inferior e superior foram alocados em dois desvios padrões em torno da média.

3.4.2 Amplificador Totalmente Diferencial de Um Estágio

O segundo circuito consiste no mesmo FDA do filtro passa-baixas do primeiro estudo de caso, porém visto de maneira isolado. Neste cenário, o teste é feito com o circuito operando em malha aberta, como mostra a Figura 16. Ele pode ser visto como um subproblema do estudo de caso anterior. Considerando estruturas de DfT, o FDA presente no filtro pode ser desconectado do restante do circuito e testado isoladamente. Em (ZILCH; LUBASZEWSKI; BALEN, 2021), este estudo de caso foi usado para demonstração da ferramenta de ATPG para testes estruturais.

Figura 16 – Esquemático do FDA em malha aberta



Fonte: adaptado de Zilch *et al.*, 2021

Os parâmetros funcionais testados são o ganho DC, o produto ganho-faixa, mais conhecido como *Gain-Bandwidth Product* (GBW), e o SR. Novamente, essas especificações são medidas apenas no nó de saída positivo Vop do circuito.

Por outro lado, as MIs são realizadas nos nós de saída Vop e Von do circuito. Mais uma vez, somente medições DC são consideradas no estudo de caso. A tensão de entrada, aplicada entre Vip e Vin , varia entre -0,9 V a 0,9 V, com passo de 0,1 V. Isso resulta em um total de 38 MIs disponíveis para a geração de testes indiretos.

Nas análises DC, AC e transiente, fez-se uso dos mesmos valores de estímulo de entrada descritos no primeiro estudo de caso. O ganho DC e o GBW foram calculados usando a análise AC, enquanto que o SR foi encontrado a partir da análise transiente.

Para formação do conjunto de dados, foram executadas 500 simulações de MC. O número de amostras geradas foi reduzido nesse estudo de caso tendo em vista que o número de possíveis características para os modelos de aprendizado também é menor. O *dataset* resultante foi dividido em subconjuntos de treinamento, validação e teste, com proporções iguais a 64%, 16% e 20%, respectivamente.

Diferentemente do primeiro estudo de caso, optou-se por classificar os circuitos pelos limites de teste definidos por *corners*. Essa distinção entre as abordagens usadas nos estudos de caso permitiu comparar como os limites de teste influenciam na qualidade do teste.

4 Resultados e Discussões

Neste capítulo, são apresentados e discutidos os resultados obtidos aplicando a geração de testes indiretos abordada neste trabalho. A partir deles, é analisada a eficiência e a viabilidade do método proposto. Primeiramente, os tempos de execução dos códigos de cada etapa da geração de teste indiretos são examinados. Em seguida, os resultados da geração de testes indiretos para cada estudo de caso são apresentados.

4.1 Tempos de Execução dos Algoritmos

Na Tabela 4, é possível ver os tempos de execução de cada etapa do método de geração de teste indiretos. O estudo de caso 1 diz respeito ao circuito passa-baixas, enquanto o estudo de caso 2 representa o FDA em malha aberta.

Tabela 4 – Tempos de execução da geração de teste indiretos para os estudos de caso

Estudo de caso	Amostras	MIs	Etapas da geração de testes indiretos		
			Simulações dos circuitos	Trein. dos modelos de aprendizado	Seleção dos testes indiretos
1	1000	114	15min50s	1h2min33s	13s
2	500	38	6min13s	12min3s	9s

Fonte: o Autor, 2023

Como no primeiro estudo de caso o número de amostras de circuitos e MIs era maior que no segundo estudo de caso, era esperado que seu tempo de execução fosse consideravelmente maior nas duas primeiras etapas. O tempo para concluir a primeira fase é 2,5 vezes maior, enquanto que a segunda fase demora 5 vezes mais para ser processada.

Na etapa de treinamento dos modelos de aprendizado, para ambos circuitos, os processos que dominaram o tempo de execução é a efetuação dos métodos *wrapper*, pois o número de treinamentos realizados na técnica de SFS é proporcional ao número de MIs. Essa tarefa consome ainda mais tempo devido à validação cruzada usada para estimar os erros de generalização dos modelos.

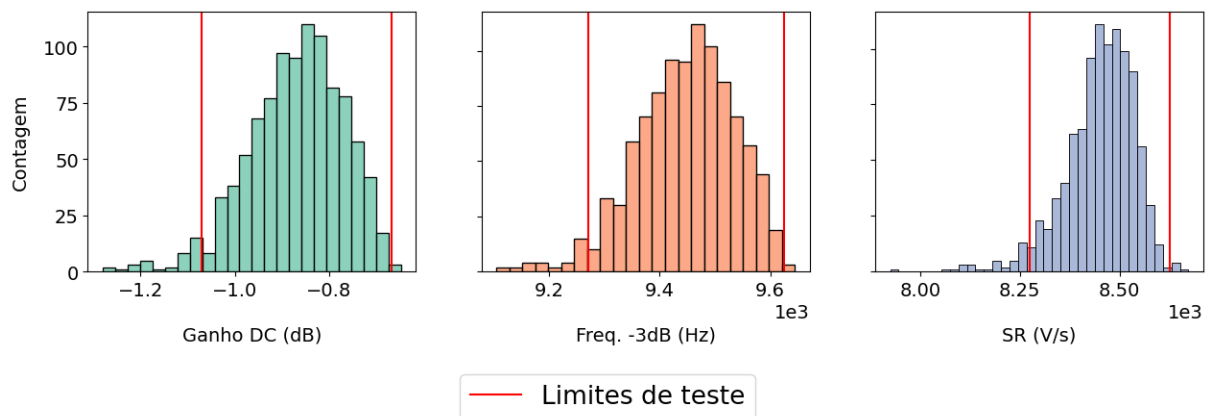
Visualizando os tempos da etapa de seleção dos testes indiretos, nota-se que essa fase possui um impacto insignificante no tempo total do método. Mesmo que os modelos sejam retreinados nessa fase, é feito apenas um treinamento para cada especificação.

Ressalta-se que durante as simulações dos estudos de caso foram calculadas especificações que necessitaram de operações relativamente simples. Dependendo do parâmetro funcional do teste, é possível que o tempo de simulação aumente. Além disso, a adição de uma etapa de otimização de hiperparâmetros aumentaria ainda mais o tempo de execução.

4.2 Filtro Passa-Baixas de Um Estágio

Nesta seção são apresentados os resultados obtidos ao longo da geração de testes indiretos para o primeiro estudo de caso. A Figura 17 ilustra as distribuições de cada especificação do conjunto total de dados sintéticos criado após a etapa de simulação. Também são mostrados os limites de testes definidos por MC, destacados em vermelho. Para auxiliar na análise das dispersões, a Tabela 5 reúne os coeficientes de variação das especificações para todos conjuntos de dados criados.

Figura 17 – Distribuições das especificações do filtro passa-baixas



Fonte: o Autor, 2023

Tabela 5 – Coeficientes de variação de cada conjunto do primeiro estudo de caso

Conjuntos	Especificações		
	Ganho DC	Freq. de corte	SR
Total	11,6%	0,94%	1,04%
Treinamento	11,5%	0,93%	1,07%
Validação	11,6%	0,93%	1,02%
Teste	11,9%	0,96%	0,97%

Fonte: o Autor, 2023

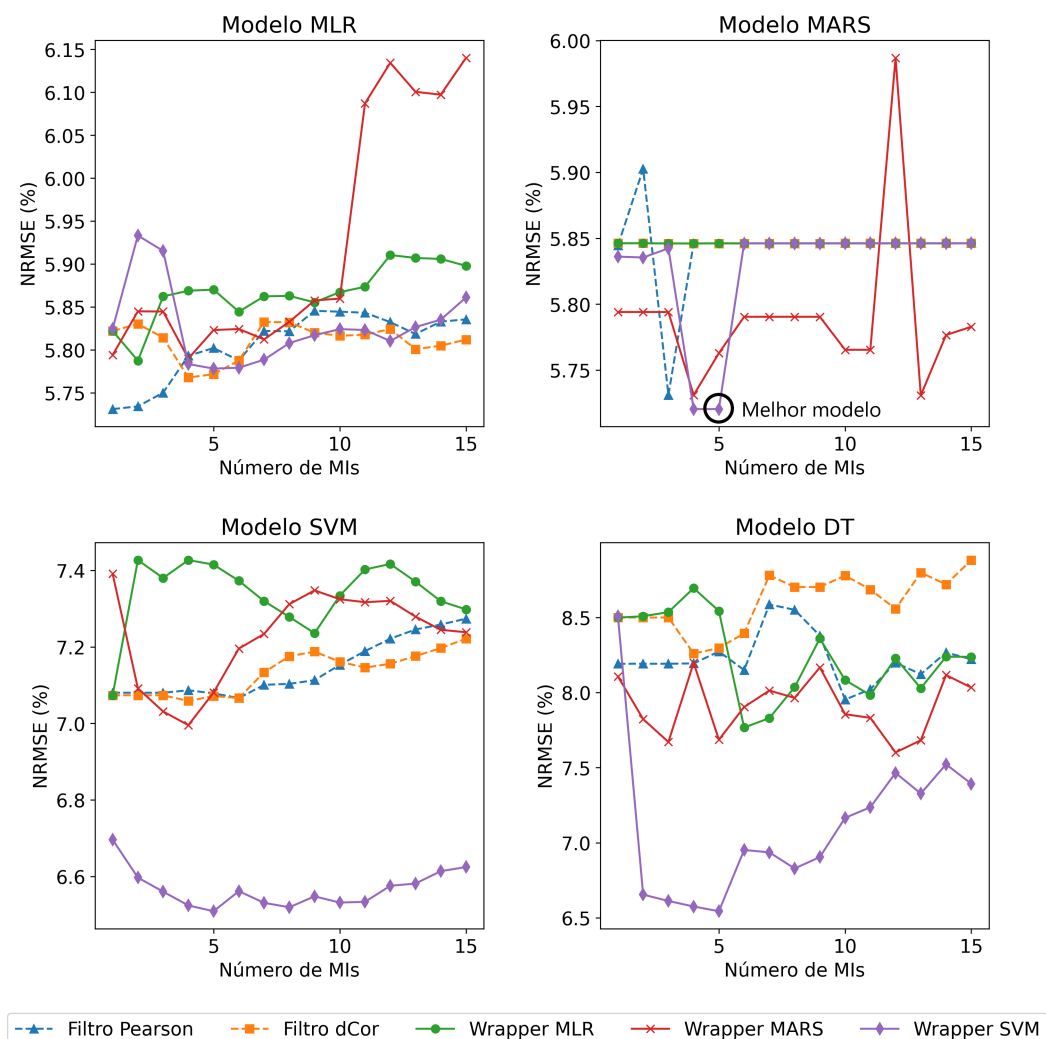
Percebe-se que os parâmetros funcionais não apresentam uma distribuição normal. Embora as distribuições do ganho DC e da frequência de corte pareçam semelhantes, o ganho DC possui uma variação mais de 10 vezes maior em torno de sua média. Por outro lado, a frequência de corte e o SR têm valores de dispersão aproximados. Além disso, nos três casos, os limites de testes estão alocados próximos aos fins das caudas das distribuições. Por essa razão, existe um número pequeno de circuitos defeituosos no conjunto de dados.

Ainda na Tabela 5, é possível notar valores similares dos coeficientes de variação para cada conjunto de dados, sugerindo que a amostragem cLHS foi eficaz na separação dos dados. Com esse resultado, entende-se que os modelos de aprendizado foram validados em um conjunto de dados com propriedades estatísticas semelhantes às do conjunto de treinamento.

Como mencionado na seção 3.2.3, após todo o processo de seleção de MIs e treinamento dos modelos de aprendizado, é feita a seleção dos testes indiretos, ou seja, quais MIs e algoritmo de aprendizado permite estimar com maior confiança as especificações. Isso é feito aplicando os modelos de aprendizado no conjunto de validação e avaliando as performances por meio do NRMSE. Para cada especificação, os modelos (juntos de suas entradas) que apresentarem menor NRMSE são selecionados.

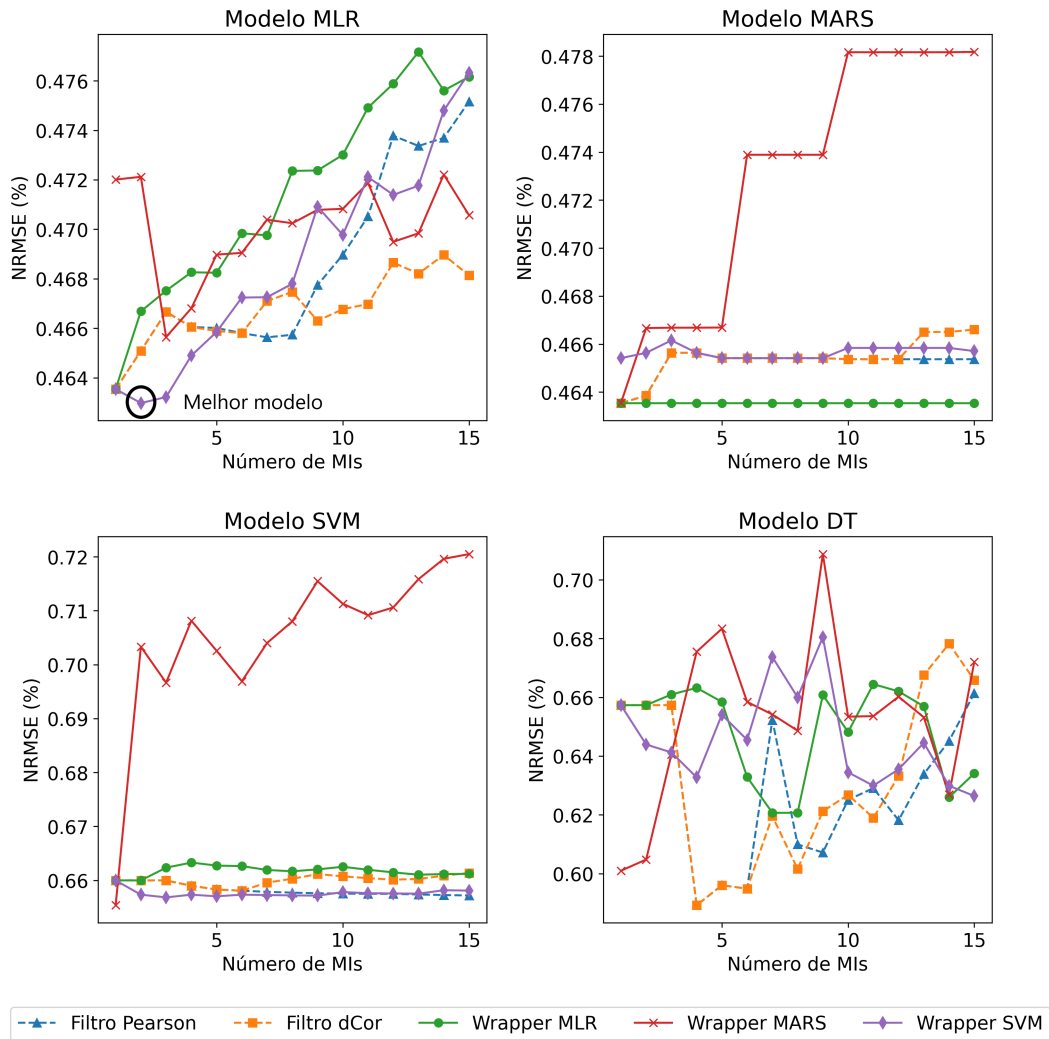
A seguir, são exibidas figuras que ilustram a etapa de avaliação dos modelos de aprendizado. Para cada especificação, são mostrados os desempenhos de todos modelos no conjunto de validação. As Figuras 18, 19 e 20 apresentam os resultados para o ganho DC, a frequência de corte e o SR, respectivamente. Cada gráfico contém um conjunto de modelos, onde eles são separados pelo seu algoritmo de aprendizado: MLR, MARS, SVM ou DT. Cada modelo dentro de um conjunto é representado por um curva diferente. Essas curvas dizem respeito ao método de seleção de características utilizado para escolher as MIs. O melhor modelo selecionado pelo método é apontado por um círculo preto.

Figura 18 – NRMSE dos modelos na predição do ganho DC do filtro passa-baixas



Fonte: o Autor, 2023

Figura 19 – NRMSE dos modelos na predição da freq. de -3dB do filtro passa-baixas

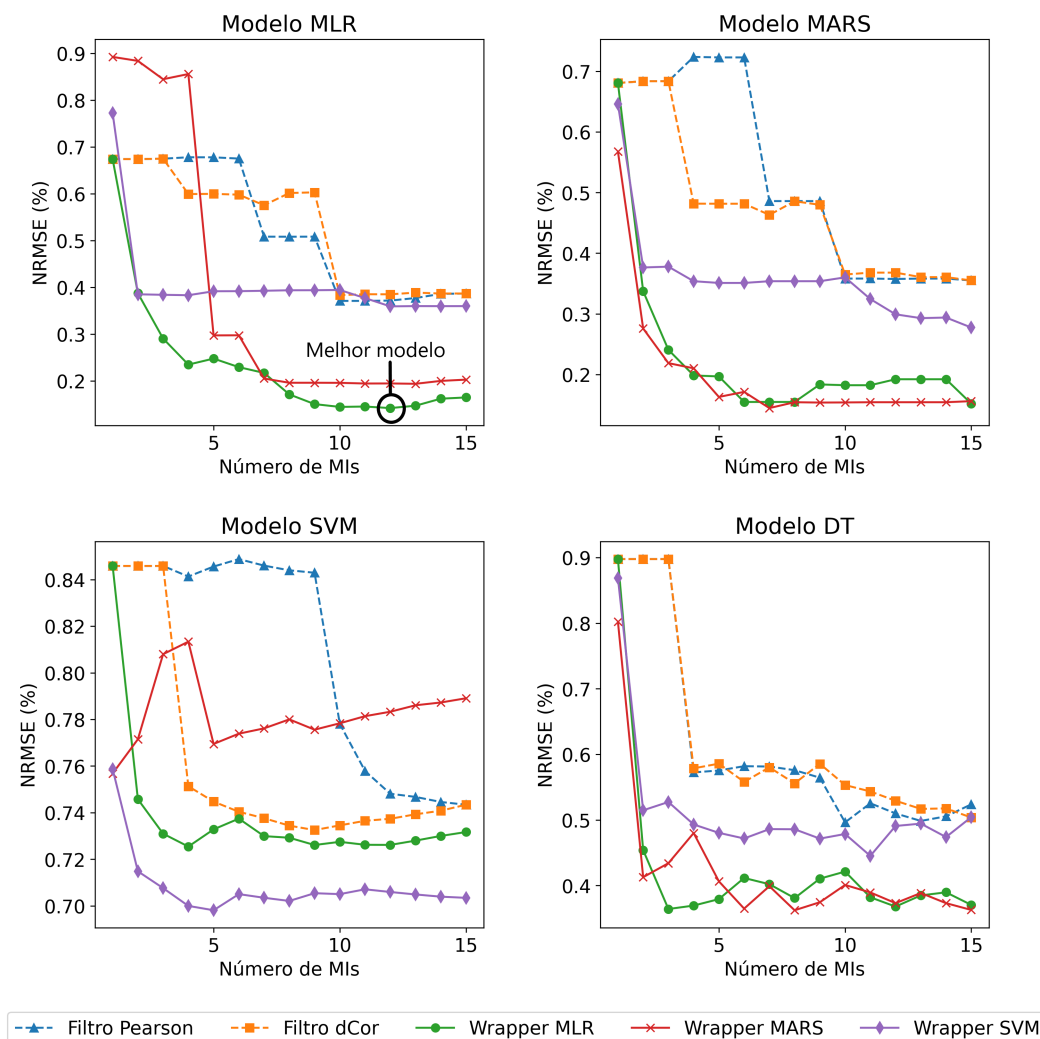


Fonte: o Autor, 2023

Observando os gráficos de desempenho, é possível realizar algumas análises quanto aos erros de predição. Primeiramente, nenhum modelo possuiu um NRMSE maior que 9%, sendo que os melhores modelos para alguns parâmetros funcionais ficaram abaixo dos 0,5%. Contudo, como explicado na seção 2.5.2, a avaliação da métrica NRMSE está relacionada com a dispersão dos dados. Por exemplo, na Tabela 5 foi visto que o ganho DC varia perto de 11,5% em torno de sua média, enquanto que o SR varia próximo de 1%. Com isso, é importante perceber que a alta precisão adquirida é consequência da baixa variabilidade das especificações.

Comparando as curvas, percebe-se que, para algumas combinações de algoritmo de aprendizado e técnica de seleção de características, o erro diminui conforme o número de MIs aumenta. Porém, em muitos casos o contrário acontece, ou seja, o erro cresce à medida que mais entradas são usadas.

Figura 20 – NRMSE dos modelos na predição do SR do filtro passa-baixas



Fonte: o Autor, 2023

Uma das razões para esse efeito é que MIs irrelevantes para a predição do modelo estão sendo selecionadas. De fato, deve-se lembrar que as correlações de Pearson e de distância univariada não consideram as interações entre variáveis de entrada, enquanto o SFS faz essa análise naturalmente. Em geral, nota-se nos gráficos uma superioridade dos métodos *wrapper*. Mesmo assim, em alguns casos, como o uso de DT para predir a frequência de corte, para alguns valores de número de entradas a seleção por filtro é igual ou superior que as por *wrapper*.

Outro fator importante é o sobreajuste dos modelos. No SFS, são adicionadas entradas que otimizam a estimação do modelo durante o treinamento. No entanto, examinando o desempenho do modelo MARS na predição da frequência de corte, com entradas selecionadas por SFS com MARS, há um claro aumento no NRMSE conforme o número de entradas aumenta. Isso é um indício do sobreajuste dos dados de treinamento, pois o modelo perde sua capacidade de generalização para novas amostras quanto mais complexo ele fica.

Após a seleção dos testes indiretos, os melhores modelos escolhidos pela metodologia foram retreinados e aplicados no conjunto de teste. Além das métricas de NRMSE e R^2 , também é calculada a MR através dos limites de teste calculados por MC, originando os dados da Tabela 6, que representam os resultados finais da geração de testes indiretos.

Tabela 6 – Testes indiretos gerados para o filtro passa-baixas

Especificação	MI	Modelo	NRMSE	R^2	MR
Ganho DC	5	MARS	5,8%	0,76	3,0%
Freq. -3dB	2	MLR	0,47%	0,76	3,5%
SR	12	MLR	0,14%	0,98	1,0%

Fonte: o Autor, 2023

O primeiro ponto de destaque são as performances estimadas. Embora o NRMSE apresente um valor pequeno para todas especificações, o R^2 , que quantifica o quanto da variância dos dados é ajustada pelo modelo, é igual a 0,76 para o ganho DC e a frequência de corte. Além disso, para o caso do SR, é obtido o menor NRMSE e o maior R^2 , mesmo sua dispersão sendo bastante próxima da vista na frequência de corte.

Os modelos selecionados foram MARS e MLR. Como esses modelos possuem uma baixa variância, entende-se que os dados sendo ajustados também possuem uma baixa variância, justificando o melhor desempenho desses modelos.

Analisando o número de MIs selecionadas, nota-se o SR foi a especificação que necessitou de mais MIs para ser melhor estimada. Contudo, verificando a Figura 20, nota-se que não há um decremento significativo no NRMSE se comparado ao modelo MLR da mesma curva, mas treinado com 10 entradas. Logo, caso fosse incluída alguma tolerância no Algoritmo 5 que impedisse o aumento no número de entradas nesses cenários, o número de MIs seria reduzido.

Na Tabela 6, não foram indicadas quais MIs foram selecionadas. Essa informação é importante, pois, como MIs representam testes de baixo custos executados no CI, é natural que as mesmas MIs possam ser usadas para predição de especificações diferentes. Assim, a união dos conjuntos de MIs para cada especificação resulta em um total de 19 MIs. Esse número é consideravelmente menor que as 114 MIs propostas no início do método.

Além disso, essas MIs selecionadas fazem uso dos 6 nós de testes proposto inicialmente. Por essa razão, todos nós do circuito, com exceção das entradas e do nó $V3$, precisariam ser medidos na implementação deste teste indireto.

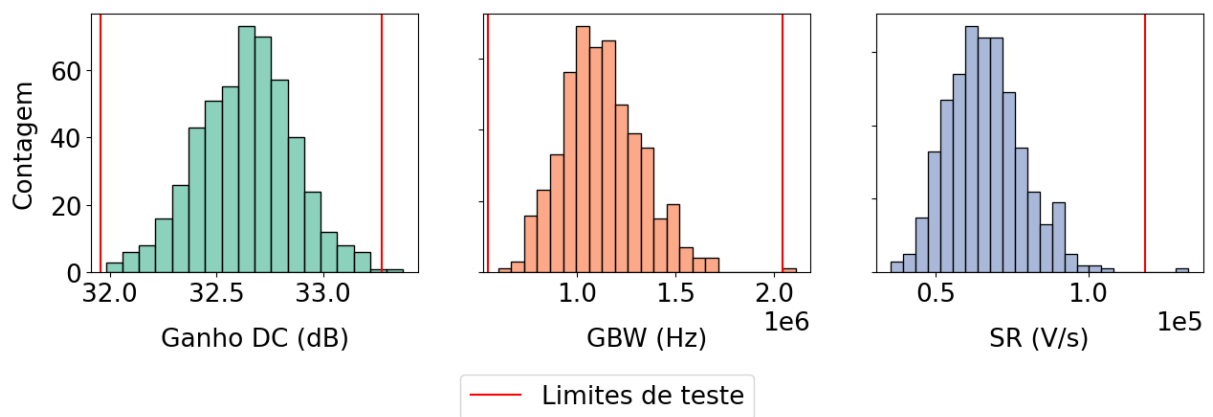
Por último, é possível fazer um estudo sobre a MR de cada modelo. Todas MRs apresentaram um valor abaixo de 3,5%, o que indica que o teste indireto teria um impacto reduzido, porém considerável na produtividade da cadeia de produção. A classificação dos circuitos levando em consideração apenas o SR mostrou uma taxa de erro igual a 1%, valor de MR mais próximo do ideal.

4.3 Amplificador Totalmente Diferencial de Um Estágio

Nesta seção, são discutidos os resultados adquiridos aplicando a geração de testes indiretos no FDA operando em malha aberta. Como a metodologia de teste aplicada é a mesma do primeiro estudo de caso, seus resultados são introduzidos de forma resumida, destacando os pontos de diferença entre os dois estudos.

A Figura 21 apresenta as distribuições das especificações do conjunto de dados sintéticos criado, enquanto a Tabela 7 mostra seus coeficientes de variação. Evidencia-se que, diferentes dos outros parâmetros funcionais, a distribuição do ganho DC se aproxima de uma curva gaussiana, com um baixo coeficiente de variação. Já o GBW e o SR variam bastante em torno de suas médias, com valores de dispersão semelhantes.

Figura 21 – Distribuições das especificações do FDA em malha aberta



Fonte: o Autor, 2023

Tabela 7 – Coeficientes de variação de cada conjunto do segundo estudo de caso

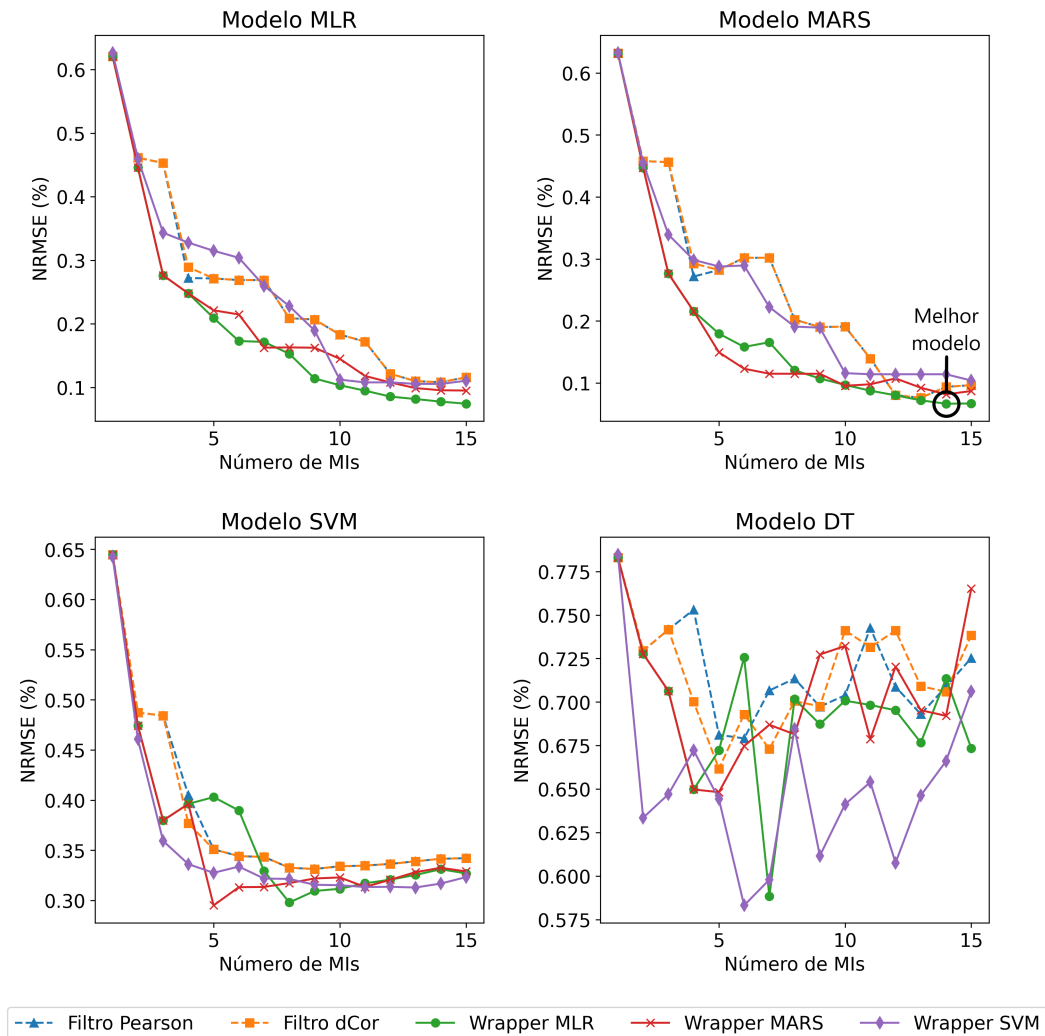
Conjuntos	Especificações		
	Ganho DC	GBW	SR
Total	0,69%	18,0%	19,0%
Treinamento	0,70%	18,4%	19,4%
Validação	0,72%	17,8%	18,6%
Teste	0,66%	17,0%	17,9%

Fonte: o Autor, 2023

Neste estudo de caso, os limites de teste foram traçados por *corners*. Como essa análise considera somente condições extremas do processo de fabricação do circuito, os limites de teste tendem a ficar mais afastados do valor médio. Isso é confirmado pelos limites representados por linhas vermelhas na Figura 21. No gráfico do SR, o limite inferior é muito menor que os valores da distribuição. Em razão disso, optou-se por não mostrá-lo. Dessa maneira, no conjunto de dados, para cada especificação, apenas um circuito foi classificado como defeituoso.

As Figuras 22, 23 e 24 exibem as curvas de desempenho de todos modelos de aprendizado treinados. Seus dados são apresentados de maneira análoga ao estudo de caso anterior.

Figura 22 – NRMSE dos modelos na predição do ganho DC do FDA em malha aberta

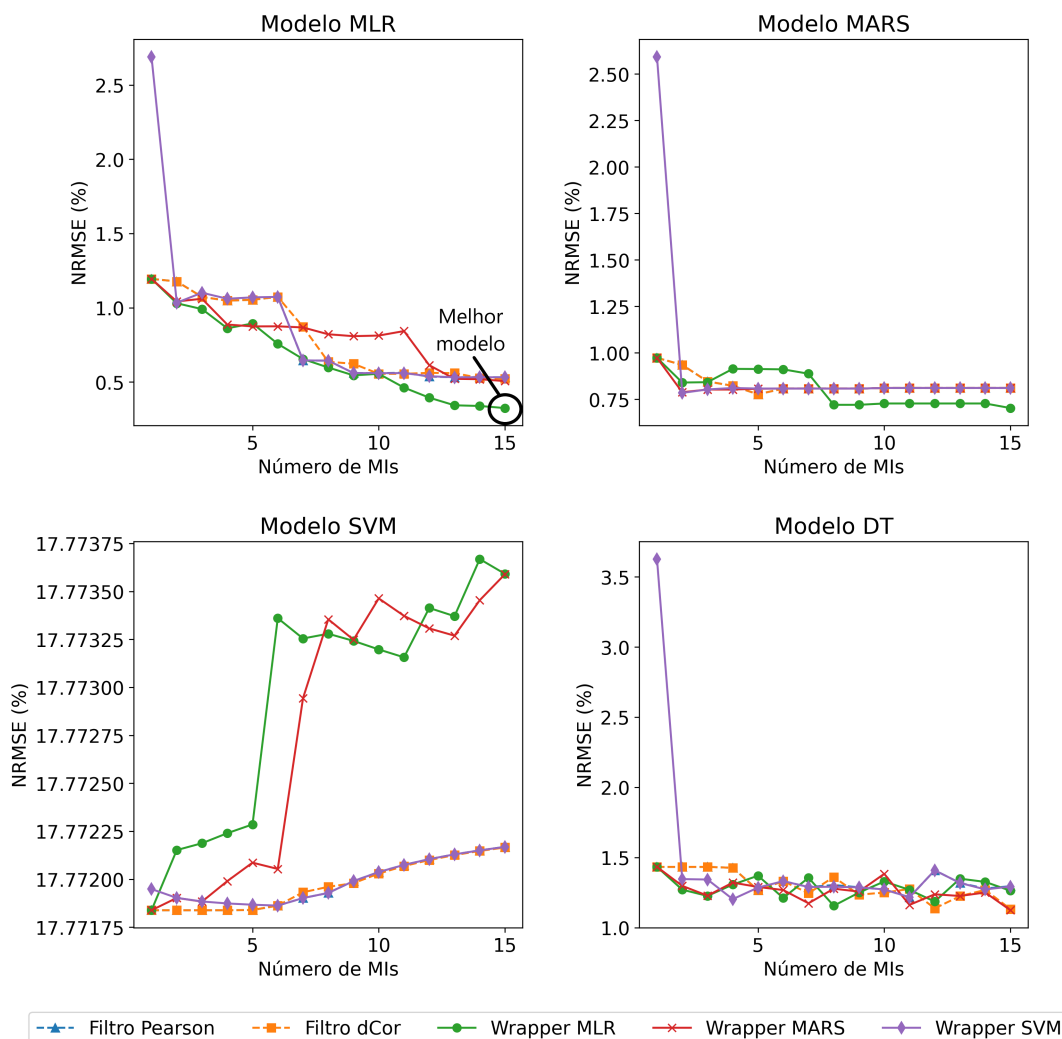


Fonte: o Autor, 2023

A grande maioria dos modelos apresentaram um decréscimo no NRMSE conforme o número de MIs foi aumentado, resultando em erros pequenos se comparados às dispersões das especificações. Nesses casos, percebe-se que as técnicas de seleção de características escolheram boas MIs para as predições. Ainda, é notado um desempenho levemente maior dos modelos treinados com MIs selecionados por *wrapper*.

Por outro lado, os modelos de DT usados para predir o ganho DC, não apresentaram bons resultados de desempenho. O NRMSE permaneceu em um valor semelhante ao coeficiente de variação para todas as entradas. Além disso, não foi possível perceber alguma tendência das performances, o que mostra que esse modelo é inadequado para predir o ganho DC.

Figura 23 – NRMSE dos modelos na predição do GBW do FDA em malha aberta



Fonte: o Autor, 2023

Outros casos que apresentaram performances fracas são os modelos de SVM para predição do GBW e SR. Os erros ficaram próximos de 18%, aumentando de maneira proporcional ao número de MIs. Como foi usada a RBF como função de Kernel do SVM, esses modelos possuem uma alta variância e tendem a sobreajustar as amostras. Caso fosse aplicada uma técnica de regularização ou utilizadas funções de Kernel mais simples, provavelmente os resultados seriam melhores.

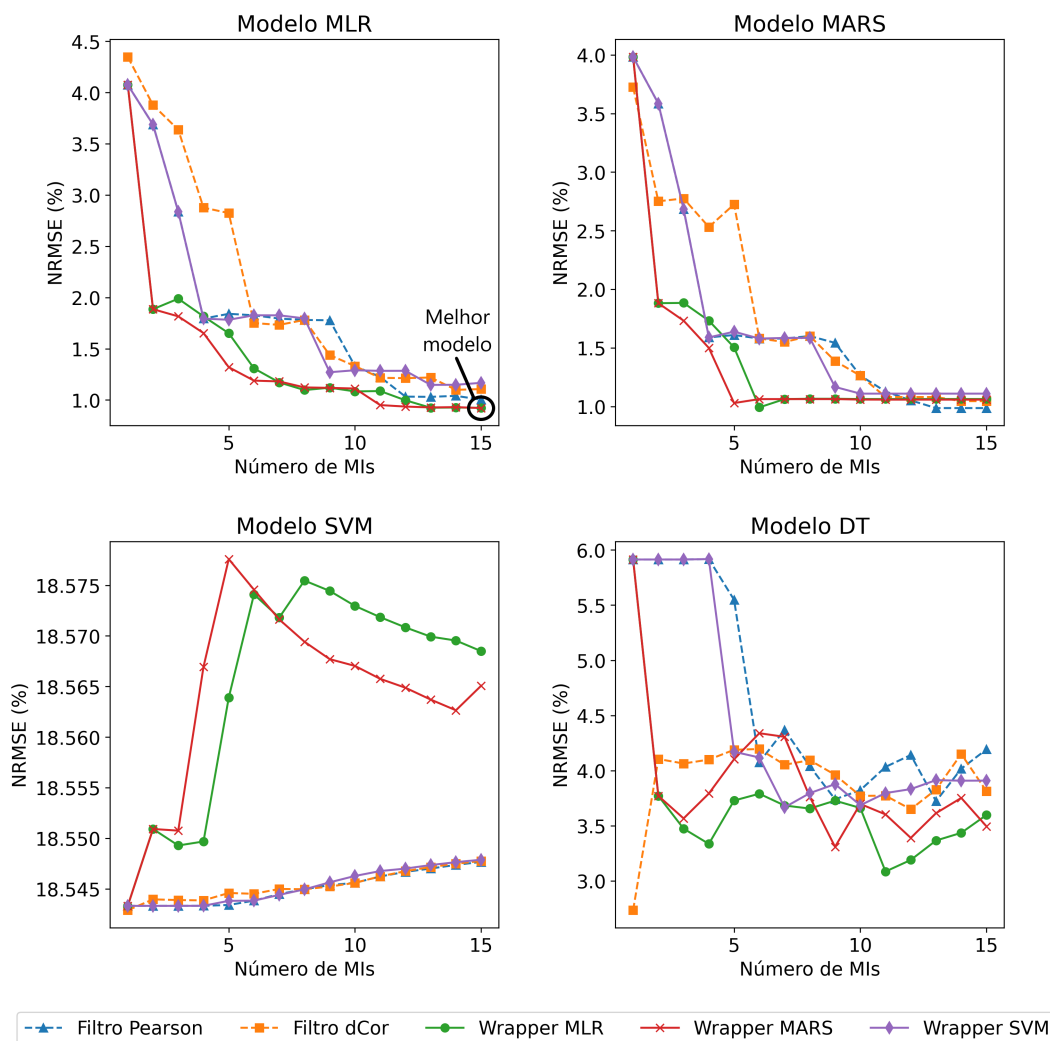
Os resultados obtidos após o retreinamento e avaliação dos melhores modelos são resumidos na Tabela 8. Também são exibidas as MRs para cada especificação.

Tabela 8 – Testes indiretos gerados para o FDA em malha aberta

Especificação	MIs	Modelo	NRMSE	R ²	MR
Ganho DC	14	MARS	0,057%	0,99	0,0%
GBW	15	MLR	0,26%	1,0	0,0%
SR	15	MLR	0,85%	1,0	0,0%

Fonte: o Autor, 2023

Figura 24 – NRMSE dos modelos na predição do SR do FDA em malha aberta



Fonte: o Autor, 2023

Novamente, os algoritmos de aprendizado selecionados pelo método de geração de testes indireto foram MARS e MLR. Através das figuras de desempenho, é observado que esses modelos apresentaram resultados superiores, principalmente para um número maior de entradas. Isso dá a entender que os dados deste estudo de caso também apresentam uma baixa variância.

Para o ganho DC, é possível notar um NRMSE cerca de 10 vezes menor que seu coeficiente de variação, junto de um R^2 igual a 0,99. Para o GBW e o SR, também são vistos baixos erros, sendo que nos dois casos o R^2 é praticamente 1. Portanto, embora o número de amostras desse estudo de caso seja igual a metade do usado no primeiro estudo, foram gerados testes indiretos para estimar todas as especificações com uma alta precisão. Mesmo o GBW e o SR possuindo uma dispersão maior, os modelos foram capazes de representar a variabilidade dos dados com sucesso.

Apesar do número grande de MIs selecionadas para cada modelo, assim como no estudo do filtro passa-baixas, é preciso verificar se a adição de entradas ocasiona em um aumento significativo no desempenho do modelo. Para o GBW e o SR, analisando as curvas referentes aos melhores modelos das Figuras 23 e 24, respectivamente, nota-se que, caso fossem usadas 13 MIs, não haveria um grande prejuízo na performance do teste indireto.

Além disso, ignorando as MIs selecionadas mais de uma vez, é obtido em um total de 27 MIs, um número de medições relativamente próximo do conjunto inicial de 38 MIs. Porém, comparando com o primeiro estudo de caso onde as MIs eram extraídas de 6 nós diferentes, para o teste indireto deste circuito as MIs são feitas somente nos nós de saída *Vop* e *Von*.

Por fim, para esse estudo de caso todas MRs apresentaram valores nulos. Esse resultado era esperado tendo em vista a localização dos limites de teste nas distribuições da Figura 21. Como os limites determinados por *cornes* ficaram bastante afastados do valor médio das especificações, é necessário tomar cuidado com a interpretação desses resultados. Mesmo assim, essa situação reproduz um cenário possível do processo de fabricação.

5 Conclusões

Neste trabalho foi proposto um método para geração de testes indiretos de circuitos analógicos. Por meio de simulações SPICE, são criados conjuntos de dados sintéticos contendo os parâmetros funcionais e estruturais do circuito sob teste. A variabilidade intrínseca do processo de fabricação é reproduzida a partir dos parâmetros de MC da tecnologia do circuito.

Através de técnicas de aprendizado de máquina, modelos de regressão são treinados com as amostras de circuitos para predir suas especificações. Diferentes estratégias de seleção de características são aplicadas para selecionar as MIs de entrada para os algoritmos de aprendizado. Os modelos que apresentam o menor NRMSE são escolhidos e seus desempenhos são avaliados.

Dessa forma, ao final da geração de testes indiretos, é obtido um conjunto de MIs e algoritmos de aprendizado. De posse desses resultados, o engenheiro de teste é capaz de avaliar a viabilidade do teste indireto em relação ao teste convencional.

O método foi aplicado em dois estudos de caso: um filtro passa-baixas e um FDA operando em malha aberta. O primeiro caso resultou em testes indiretos que fazem uso de 19 MIs no total, e dos algoritmos de aprendizado MARS e MLR, resultando em um R^2 igual a 0,76, 0,76 e 0,98, para predição do ganho DC, a frequência de corte e o SR do circuito, respectivamente. Já no segundo estudo de caso, foi apontado um total de 27 MIs, que devem ser usadas para treinar os modelos MARS e MLR. Os modelos apresentaram bom desempenho, com um R^2 aproximadamente igual a 1 para as estimativas do ganho DC, GBW e SR do circuito.

Embora os testes indiretos gerados neste trabalho tenham apresentado uma alta eficiência, é necessário interpretar os dados simulados com precaução. De fato, a eficácia do método proposto está diretamente atrelada a capacidade das simulações representarem o comportamento real dos circuitos. No processo de fabricação, falhas catastróficas no circuito, por exemplo, tendem a mudar drasticamente os resultados de circuitos. Isso dificultaria o treinamento de modelos de aprendizado. No entanto, esse obstáculo poderia ser ultrapassado pela adição de uma etapa anterior ao teste indireto. Essa fase seria responsável pela remoção de circuitos com grandes desvios em seus comportamentos. Essa tarefa poderia ser executada, por exemplo, por um teste orientado aos defeitos.

Finalmente, é possível afirmar que o método apresentado facilita a implementação do teste indireto em circuitos analógicos. Uma grande combinação de parâmetros de projeto podem ser explorados, selecionando aqueles que melhor contribuem para a qualidade do teste indireto.

5.1 Trabalhos Futuros

Como mencionado no capítulo 1, este trabalho foi desenvolvido com o intuito de integrar uma ferramenta de ATPG para testes estruturais. Desse modo, o projeto de uma ferramenta única de geração de testes que reúne dois paradigmas diferentes apresentaria muitas vantagens. Para isso, é possível o desenvolvimento de um pacote de Python que reúna essas funcionalidades e permita uma fácil acessibilidade.

Uma forma de validação do método proposto é sua aplicação com um conjunto de dados extraídos diretamente de uma cadeia de produção de CIs. Então, seria possível comparar os resultados obtido por simulação com os resultantes de dados reais. Além disso, é importante avaliar o método a partir de outros estudos de caso, principalmente naqueles compostos por circuitos mais complexos.

Referências Bibliográficas

- ARABI, K.; KAMINSKA, B. Testing analog and mixed-signal integrated circuits using oscillation-test method. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, IEEE, v. 16, n. 7, p. 745–753, 1997. DOI: 10.1109/43.644035.
- ARSLAN, B.; ORAILOGLU, A. Tracing the best test mix through multi-variate quality tracking. In: *2013 IEEE 31st VLSI Test Symposium (VTS)*. Berkeley: IEEE, 2013. p. 1–6. DOI: 10.1109/VTS.2013.6548886.
- BADAWI, H. E.; AZAÏS, F.; BERNARD, S.; COMTE, M.; KERZÉRHO, V.; LEFEVRE, F. Investigations on the use of ensemble methods for specification-oriented indirect test of rf circuits. *Journal of Electronic Testing*, Springer, v. 36, n. 2, p. 189–203, 2020. DOI: 10.1007/s10836-020-05868-3.
- BADAWI, H. E.; AZAÏS, F.; BERNARD, S.; COMTE, M.; KERZÉRHO, V.; LEFEVRE, F. Evaluation of a two-tier adaptive indirect test flow for a front-end rf circuit. *Journal of Electronic Testing*, Springer, v. 37, n. 2, p. 225–242, 2021. DOI: 10.1007/s10836-021-05934-4.
- BALEN, T. R. *Teste de Dispositivos Analógicos Programáveis (FPAAS)*. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006. Disponível em: <<http://hdl.handle.net/10183/7891>>. Acesso em: 17 de jun. de 2022.
- BARRAGAN, M. J.; LEGER, G. Feature selection for alternate test using wrappers: Application to an rf lna case study. In: *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Grenoble: IEEE, 2015. p. 1229–1232. DOI: 10.7873/DATE.2015.0179.
- BARRAGAN, M. J.; LEGER, G. Feature selection and feature design for machine learning indirect test: a tutorial review. In: *2019 16th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)*. Lausanne: IEEE, 2019. p. 69–72. DOI: 10.1109/SMACD.2019.8795292.
- BARRAGAN, M. J.; LEGER, G.; HUERTAS, J. L. Multi-condition alternate test of analog, mixed-signal, and rf systems. In: *2012 13th Latin American Test Workshop (LATW)*. Quito: IEEE, 2012. p. 1–6. DOI: 10.1109/LATW.2012.6261248.
- BURKOV, A. *The Hundred-Page Machine Learning Book*. 1. ed. Andriy Burkov, 2019. Disponível em: <<https://themlbook.com>>. Acesso em: 15 de set. de 2022.
- BUSHNELL, M.; AGRAWAL, V. *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. 1. ed. New York: Springer, 2002. (Frontiers in Electronic Testing). DOI: 10.1007/b117406.
- CHATTERJEE, A.; NAGI, N. Design for testability and built-in self-test of mixed-signal circuits: a tutorial. In: *Proceedings Tenth International Conference on VLSI Design*. Hyderabad: IEEE, 1997. p. 388–392. DOI: 10.1109/ICVD.1997.568158.

- CHIEN, A. A.; KARAMCHETI, V. Moore's law: The first ending and a new beginning. *Computer, IEEE*, v. 46, n. 12, p. 48–53, 2013. DOI: 10.1109/MC.2013.431.
- FRIEDMAN, J. H. Multivariate adaptive regression splines. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 19, n. 1, p. 1–67, 1991. DOI: 10.1214/aos/1176347963.
- GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2. ed. Sebastopol: O'Reilly Media, 2019.
- HARRIS, C. R. *et al.* Array programming with numpy. *Nature, Nature Portfolio*, v. 585, n. 7825, p. 357–362, 2020.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. New York: Springer, 2009. (Springer series in statistics). DOI: 0.1007/978-0-387-84858-7.
- HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering, IEEE*, v. 9, n. 3, p. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.
- KABISATPATHY, P.; BARUA, A.; SINHA, S. *Fault diagnosis of analog integrated circuits*. 1. ed. New York: Springer, 2005. (Frontiers in Electronic Testing). DOI: 10.1007/b135977.
- KARKI, J. *Fully-Differential Amplifiers*. [S.l.]: Texas Instruments Incorporated: Application Report, 2002. <<https://www.ti.com/lit/pdf/sloa054>>.
- KLUYVER, T. *et al.* Jupyter notebooks - a publishing format for reproducible computational workflows. In: *20th Conference on Electronic Publishing (Elpub)*. Göttingen: IOS Press, 2016. p. 87–90. DOI: 10.3233/978-1-61499-649-1-87.
- KUHN, M.; JOHNSON, K. *Applied Predictive Modeling*. 1. ed. New York: Springer, 2013. DOI: 10.1007/978-1-4614-6849-3.
- LEGER, G.; BARRAGAN, M. J. Mixed-signal test automation: Are we there yet? In: *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. Florence: IEEE, 2018. p. 1–5. DOI: 10.1109/ISCAS.2018.8351734.
- LIAO, C. PyEarth: A lightweight Python package for Earth science. Zenodo, 2022. DOI: 10.5281/zenodo.6368652.
- MCKINNEY, W. Data structures for statistical computing in python. In: *Proceedings of the 9th Python in Science Conference*. Austin: SciPy, 2010. p. 56–61. DOI: 10.25080/Majora-92bf1922-00a.
- MILOR, L. A tutorial introduction to research on analog and mixed-signal circuit testing. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, IEEE, v. 45, n. 10, p. 1389–1407, 1998. DOI: 10.1109/82.728852.
- MILOR, L.; VISVANATHAN, V. Efficient go/no-go testing of analog circuits. In: *International Symposium on Circuits and Systems (ISCAS)*. New York: IEEE, 1987. p. 414–417.

- MINASNY, B.; MCBRATNEY, A. B. A conditioned latin hypercube method for sampling in the presence of ancillary information. *Computers & Geosciences*, Elsevier, v. 32, n. 9, p. 1378–1388, 2006. DOI: 10.1016/j.cageo.2005.12.009.
- MOORE, G. E. Progress in digital integrated electronics. *IEEE Solid-State Circuits Society Newsletter*, IEEE, v. 11, n. 3, p. 36–37, 2006. DOI: 10.1109/N-SSC.2006.4804410.
- NAVIDI, W. *Statistics for Engineers and Scientists*. 5. ed. New York: McGraw-Hill, 2020.
- OLIVEIRA, A. C. de; SEVERO, L. C.; GIRARDI, A. Automatic design of fully differential amplifiers with common-mode feedback. In: *14th Microelectronics Students Forum*. Aracaju: SBMicro, 2014. Disponível em: <<https://sbmicro.org.br/sforum-eventos/sforum2014/25.pdf>>. Acesso em: 26 de abr. de 2022.
- PEDREGOSA, F. *et al.* Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, JMLR, v. 12, n. 85, p. 2825–2830, 2011. Disponível em: <<http://jmlr.org/papers/v12/pedregosa11a.html>>. Acesso em: 20 de out. de 2022.
- POEHL, F.; DEMMERLE, F.; ALT, J.; OBERMEIR, H. Production test challenges for highly integrated mobile phone socs — a case study. In: *2010 15th IEEE European Test Symposium*. Prague: IEEE, 2010. p. 17–22. DOI: 10.1109/ETSYM.2010.5512786.
- RAMOS-CARREÑO, C.; TORRECILLA, J. L. dcor: Distance correlation and energy statistics in Python. *SoftwareX*, Elsevier, v. 22, 2023. DOI: 10.1016/j.softx.2023.101326.
- RASCHKA, S. Mlxtend: Providing machine learning and data science utilities and extensions to python’s scientific computing stack. *The Journal of Open Source Software*, JOSS, v. 3, n. 24, p. 638, 2018. DOI: 10.21105/joss.00638.
- RASCHKA, S. *Python Machine Learning*. 3. ed. Birmingham: Packt, 2019. Disponível em: <<https://www.packtpub.com/product/python-machine-learning-third-edition>>. Acesso em: 11 de ago. de 2022.
- RENOVELL, M. Digital and analog system testing: fundamentals and new challenges. In: *Proceedings. The 16th International Conference on Microelectronics, ICM 2004*. Tunis: IEEE, 2004. p. 8–10. DOI: 10.1109/ICM.2004.1434191.
- SOUDERS, T.; STENBAKKEN, G. A comprehensive approach for modeling and testing analog and mixed-signal devices. In: *Proceedings. International Test Conference 1990*. Washington: IEEE, 1990. p. 169–176. DOI: 10.1109/TEST.1990.114015.
- SOUDERS, T.; STENBAKKEN, G. Cutting the high cost of testing. *IEEE Spectrum*, IEEE, v. 28, n. 3, p. 48–51, 1991. DOI: 10.1109/6.67285.
- SRIMANI, S.; GHOSH, K.; RAHAMAN, H. Parametric fault detection in analog circuits: A statistical approach. In: *2016 IEEE 25th Asian Test Symposium (ATS)*. Hiroshima: IEEE, 2016. p. 275–280. DOI: 10.1109/ATS.2016.55.
- STRATIGOPOULOS, H.-G. Machine learning applications in ic testing. In: *2018 IEEE 23rd European Test Symposium (ETS)*. Bremen: IEEE, 2018. p. 1–10. DOI: 10.1109/ETS.2018.8400701.

- STRATIGOPOULOS, H.-G.; MAKRIS, Y. Nonlinear decision boundaries for testing analog circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, IEEE, v. 24, n. 11, p. 1760–1773, 2005. DOI: 10.1109/TCAD.2005.855835.
- STRATIGOPOULOS, H.-G.; MIR, S.; MAKRIS, Y. Enrichment of limited training sets in machine-learning-based analog/rf test. In: *2009 Design, Automation & Test in Europe Conference & Exhibition*. Nice: IEEE, 2009. p. 1668–1673. DOI: 10.1109/DATE.2009.5090931.
- STRATIGOPOULOS, H.-G.; SUNTER, S. Fast monte carlo-based estimation of analog parametric test metrics. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, IEEE, v. 33, n. 12, p. 1977–1990, 2014. DOI: 10.1109/TCAD.2014.2360458.
- SZÉKELY, G. J.; RIZZO, M. L.; BAKIROV, N. K. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 35, n. 6, p. 2769–2794, 2007. DOI: 10.1214/009053607000000505.
- TAUR, Y. *et al.* Cmos scaling into the nanometer regime. *Proceedings of the IEEE*, IEEE, v. 85, n. 4, p. 486–504, 1997. DOI: 10.1109/5.573737.
- VARIYAM, P.; CHATTERJEE, A. Enhancing test effectiveness for analog circuits using synthesized measurements. In: *Proceedings. 16th IEEE VLSI Test Symposium (Cat. No.98TB100231)*. Monterey: IEEE, 1998. p. 132–137. DOI: 10.1109/VTEST.1998.670860.
- WAGONER, E.; ZHENG, Z. *cLHS: Conditioned Latin Hypercube Sampling*. [S.l.]: GitHub, 2019. <https://github.com/wagoner47/clhs_py>.
- WASKOM, M. L. seaborn: statistical data visualization. *Journal of Open Source Software*, JOSS, v. 6, n. 60, p. 3021, 2021. DOI: 10.21105/joss.03021.
- WILLIAMS, R. S. What’s next? [the end of moore’s law]. *Computing in Science & Engineering*, IEEE, v. 19, n. 2, p. 7–13, 2017. DOI: 10.1109/MCSE.2017.31.
- ZILCH, L. B.; LUBASZEWSKI, M. S.; BALEN, T. R. A versatile test set generation tool for structural analog circuit testing. In: *2021 34th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI)*. Campinas: IEEE, 2021. p. 1–6. DOI: 10.1109/SBCCI53441.2021.9529987.
- ZILCH, L. B.; LUBASZEWSKI, M. S.; BALEN, T. R. Automatic tool for test set generation and dft assessment in analog circuits. *Analog Integrated Circuits and Signal Processing*, Springer, v. 112, n. 2, p. 277–287, 2022. DOI: 10.1007/s10470-022-02039-6.