



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciência

Faculdade de Engenharia

Alvaro Cesar Otoni Lombardi


**Detecção de falhas em circuitos eletrônicos lineares
baseada em classificadores de classe única**

Rio de Janeiro

2011

Alvaro Cesar Otoni Lombardi

**Detecção de falhas em circuitos eletrônicos lineares
baseada em classificadores de classe única**



Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao programa de Pós-Graduação em Engenharia Eletrônica da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientador: Prof. Dr. Jorge Luís Machado do Amaral

Rio de Janeiro

2011

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC / B

L842 Lombardi, Alvaro Cesar Otoni.
Detecção de falhas em circuitos eletrônicos lineares baseada em classificadores de classe única / Alvaro Cesar Otoni Lombardi. - 2011.
87 f.

Orientador: Jorge Luís Machado do Amaral.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Detecção de falhas – Teses. 2. Circuitos analógicos – Teses. 3. Engenharia Eletrônica. I. Amaral, Jorge Luís Machado do. II. Universidade do Estado do Rio de Janeiro. III. Título.

CDU 621.38

Autorizo, apenas para fins acadêmicos e científicos, a reprodução total ou parcial desta tese, desde que citada a fonte.

Assinatura

Data

Alvaro Cesar Otoni Lombardi

**Detecção de falhas em circuitos eletrônicos lineares
baseada em classificadores de classe única**

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao programa de Pós-Graduação em Engenharia Eletrônica da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em: 05 de Agosto de 2011.

Banca Examinadora:

Prof. Dr. Jorge Luís Machado do Amaral (Orientador)
Faculdade de Engenharia – UERJ

Prof. Dr. José Franco Machado do Amaral
Faculdade de Engenharia – UERJ

Prof. Dr. Ricardo Tanscheit
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Rio de Janeiro

2011

DEDICATÓRIA

Tenho comigo que o homem por natureza progride e de alguma forma contribui para desenvolvimento humano, portanto o aprender e o ensinar fazem parte das atividades do progredir e para que isso ocorra, é necessário que algumas condições sejam satisfeitas como: família, trabalho, escola entre outras instituições importantes. Por isso dedico essa importante etapa da minha vida à minha família especialmente à minha dedicada esposa Heloisa, à minha filha Júlia, aos profissionais, aos colegas e aos alunos com quem convivo e me apóiam, aos meus professores do curso de pós-graduação e aos alunos e profissionais que de alguma forma se beneficiarão desse trabalho.

AGRADECIMENTOS

Lembrar de todas as pessoas que de alguma forma me ajudaram a concluir esse projeto de vida é uma tarefa difícil, porém vou começar agradecendo de coração a Deus, por continuar fazendo como sempre fez, me protegendo, me dando desafios para que eu consiga progredir, permitindo que eu tenha conquistado o que tenho até hoje e me ajudando desde sempre. Aos meus pais, aos quais estarão sempre na minha lembrança e que sempre me incentivaram, me cobraram sobre meus estudos, a minha esposa Heloisa que apóia incondicionalmente meus projetos e se dedica em me ajudar da melhor forma possível. Agradeço também a minha filha por observar e reconhecer meu esforço. É muito importante comentar o apoio e incentivo de toda minha família por essa conquista. Aos professores dessa instituição pelo conhecimento, vontade de ensinar e dedicação aos alunos e em especial ao meu orientador Prof. Jorge Luís Machado do Amaral por ter dado a mim os incentivos e a oportunidade de absorver novos conhecimentos para o desenvolvimento desse trabalho. Acho muito pertinente agradecer ao professor Orlando Lima de Saboya Barros pela valorosa contribuição para esse trabalho. Agradeço aos colegas de curso que aqui conheci pela ajuda e incentivo principalmente ao Márcio Sebastião Costa. Agradeço ao professor, colega e amigo Luis Antônio Corrêa Coelho pelo companheirismo, incentivo e apoio ao meu projeto, aos colegas da Associação Educacional Dom Bosco, aos diretores e coordenadores dessa entidade pelo incentivo e apoio e aos profissionais e colegas da Escola Técnica Pandiá Calógeras que também me apoiaram. E finalmente a todas as outras pessoas que me ajudaram a chegar até aqui.

RESUMO

LOMBARDI, Alvaro C. O. *Detecção de falhas em circuitos eletrônicos lineares baseada em classificadores de classe única*. 2011. 87f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2011.

Esse trabalho está baseado na investigação dos detectores de falhas aplicando classificadores de classe única. As falhas a serem detectadas são relativas ao estado de funcionamento de cada componente do circuito, especificamente de suas tolerâncias (falha paramétrica). Usando a função de transferência de cada um dos circuitos, são gerados e analisados os sinais de saída com os componentes dentro e fora da tolerância. Uma função degrau é aplicada à entrada do circuito, o sinal de saída desse circuito passa por uma função diferenciadora e um filtro. O sinal de saída do filtro passa por um processo de extração de atributos e finalmente, o sinal segue simultaneamente para os classificadores multiclasse e classe única. Na análise, são empregadas ferramentas de reconhecimento de padrões e de classificação de classe única. Os classificadores multiclasse são capazes de classificar o sinal de saída do circuito em uma das classes de falha para o qual foram treinados. Eles apresentam um bom desempenho quando as classes de falha não possuem superposição e quando eles não são apresentados a classes de falhas para os quais não foram treinados. O comitê de classificadores de classe única pode classificar o sinal de saída em uma ou mais classes de falha e também podem classificá-lo em nenhuma classe. Eles apresentam desempenho comparável ao classificador multiclasse, mas também são capazes detectar casos de sobreposição de classes de falhas e indicar situações de falhas para os quais não foram treinados (falhas desconhecidas). Os resultados obtidos nesse trabalho mostraram que o comitê de classificadores de classe única, além de ter um desempenho comparável como classificador multiclasse quando não há sobreposição, também detectou sobreposições existentes sugerindo as possíveis falhas.

Palavras-Chave: Detecção de falhas; Classificadores de classe única; Falha paramétrica; Resposta ao impulso; Circuitos analógicos.

ABSTRACT

This work deals with the application of one class classifiers in fault detection. The faults to be detected are related parametric faults. The transfer function of each circuit was generated and the outputs signals were analyzed. Pattern recognition and one class classifications tools are employed to perform the analysis. The multiclass classifiers are able to classify the circuit output signal in one of the trained classes. They present a good performance when the fault classes do not overlap or when they are not presented to fault classes that were not presented in the training. The one class classifier committee may classify the output signal in one or more fault classes and may also classify them in none of the trained class faults. It presents comparable performance in relation to multiclass classifier, but also is able to detect overlapping fault classes and show fault situations that were no present in the training (unknown faults).

Keywords: Faults detect; One class classifiers; Parametric fault; Impulse response; Analog circuits.

LISTA DE ILUSTRAÇÃO

Figura 1 – O ciclo do reconhecimento de padrões (Adaptado de KUNCHEVA, 2004).	24
Figura 2 – Tipos de características (Adaptado de KUNCHEVA, 2004).	26
Figura 3 – Exemplos de dígitos manuscritos (Adaptado de KUNCHEVA, 2004).	27
Figura 4 – Modelo canônico de um classificador (Adaptado de KUNCHEVA, 2004).	28
Figura 5 – Uma taxonomia de métodos para projeto de classificadores (Adaptado de KUNCHEVA, 2004).	29
Figura 6 – Exemplo de matriz de confusão (a) primeira parte de uma matriz de confusão; (b) segunda parte de uma matriz de confusão.	38
Figura 7 – Curva característica com o ponto de operação indicado pelo ponto.	43
Figura 8 – Diagrama em blocos do detector de falhas.	45
Figura 9 – Representação da resposta ao impulso de um sistema LTI.	45
Figura 10 – Representação PAA de um sinal sobreposta ao sinal impulso original.	46
Figura 11 – Bloco do classificador multiclasse.	47
Figura 12 – Arquitetura do comitê classificador combinado classe única.	48
Figura 13 – Exemplo de envoltória na função de transferência do circuito <i>sallen-key</i>	50
Figura 14 – Envoltórias mostrando (a) circuito dentro e (b) circuitos fora.	50
Figura 15 – Circuito <i>Sallen-Key</i>	51
Figura 16 – Resposta em frequência do circuito <i>sallen-key</i>	52
Figura 17 – Gráfico dos valores de <i>F1</i> em função da fração de rejeição (%) <i>sallen-key</i>	56
Figura 18 – Circuito <i>benchmark</i> <i>ctsv</i> (<i>continuous-time state-variable</i>).	61
Figura 19 – Resposta em frequência do circuito <i>ctsv</i>	62
Figura 20 – Gráfico dos valores de <i>F1</i> em função da fração de rejeição (%) <i>ctsv</i>	67
Figura 21 – Gráficos de resposta ao impulso de: (a) <i>Sallen-key</i> múltiplas falhas, (b) zoom de (a), (c) <i>ctsv</i> falhas simples, (d) zoom de (c), (e) <i>ctsv</i> múltiplas falhas, (f) zoom de (e).	75
Figura 22 – Matriz confusão (MC) o classificador multiclasse <i>knn SK</i> falhas simples	82
Figura 23 – MC classificador classe única <i>knn_dd SK</i> falhas simples	82
Figura 24 – MC classificador classe única <i>gauss_dd SK</i> falhas simples	82
Figura 25 – MC classificador classe única <i>kmeans_dd SK</i> falhas simples.	82
Figura 26 – MC classificador multiclasse <i>knn SK</i> múltiplas falhas.	83
Figura 27 – MC classificador classe única <i>knn_dd SK</i> múltiplas falhas.	83
Figura 28 – MC classificador classe única <i>gauss_dd SK</i> múltiplas falhas.	83

Figura 29 – MC para o classificador classe única <i>kmeans_dd SK</i> múltiplas falhas.....	83
Figura 30 – (a) e (b) MC classificador multiclasse <i>knn csv</i> falhas simples.....	84
Figura 31 – (a) e (b) MC classificador classe única <i>knn_dd csv</i> falhas simples.....	85
Figura 32 – MC classificador classe única <i>gauss_dd csv</i> falhas simples	85
Figura 33 – (a) e (b) MC classificador classe única <i>kmeans_dd csv</i> falhas simples.	86
Figura 34 – (a) e (b) MC classificador multiclasse <i>knn circuito csv</i> falhas múltiplas.....	87
Figura 35 – (a) MC classificador classe única <i>knn_dd csv</i> falhas múltiplas.	87
Figura 36 – (a) MC classificador classe única <i>gauss_dd csv</i> falhas múltiplas.	88
Figura 37 – (a) e (b) MC classificador classe única <i>kmeans_dd csv</i> falhas múltiplas.....	89

LISTA DE TABELAS

Tabela 1 – Classes relacionados às falhas de tolerância para o circuito <i>sallen-key</i>	53
Tabela 2 – Erro de validação cruzada do circuito <i>sallen-key</i> , (<i>knn</i>) por n° <i>knn</i>	53
Tabela 3 – Resultado da <i>k_nn</i> para o <i>Sallen-Key</i> com falhas simples.	54
Tabela 4 – Valores de <i>F1</i> em função da fração de rejeição (%) para <i>knn_dd</i> , <i>gauss_dd</i> e <i>kmeans_dd</i>	56
Tabela 5 – Resultado do comitê de classificadores de classe única para <i>SK</i> falhas simples.	57
Tabela 6 – Resultado da classificação <i>k_nn</i> para o <i>Sallen-Key</i> com falhas múltiplas.	59
Tabela 7 – Resultado do comitê de classificadores de classe única para <i>SK</i> , falhas múltiplas.....	60
Tabela 8 – Classe do circuito, início, fim e componente com falha	63
Tabela 9 – Erro de validação cruzada <i>ctsv</i> para o classificador <i>knn</i> pelo n° <i>knn</i>	64
Tabela 10– Resultado da classificação <i>k_nn</i> para o <i>ctsv</i> com falhas simples.	65
Tabela 11– Valores de <i>F1</i> em função da fração de rejeição (%).	66
Tabela 12– Resultado dos comitês classificadores classe única <i>ctsv</i> falhas simples.	68
Tabela 13– Resultado da classificação <i>k_nn</i> para o <i>ctsv</i> com falhas múltiplas.	70
Tabela 14– Resultado dos comitês classificadores classe única para <i>ctsv</i> falhas múltiplas...	71

LISTA DE ABREVIACOES

APCA	<i>Adaptive Piecewise Constant Approximation</i>
ATE	<i>automatic test equipment</i>
BIST	<i>built-in self-test</i>
ctsv	<i>continuous-time state-variable</i>
FDM	<i>fail detector module</i>
IR	<i>impulse response</i>
ISO/CD 10303	<i>International Organization of Standardization - Industrial automation systems, Product data representation and exchange, Part 26: Implementation methods: Binary representation of EXPRESS-driven data</i>
k_{nn}	<i>$k_{nearest}$ neighbor</i>
k_{nndd}	<i>$k_{nearest}$ neighbor data description</i>
LDC	<i>linear discriminant classifier;</i>
LTI	<i>linear time invariant</i>
LVQ	<i>learning vector quantization;</i>
MC	Matriz de Confuso
MLP	<i>multilayer perceptron</i>
PAA	<i>Piecewise Aggregate Approximation</i>
<i>pdf</i>	<i>probability density function</i> (funo densidade de probabilidade)
QDC	<i>quadratic discriminant classifier;</i>
RBF	<i>radial basis function</i>

ROC	<i>receiver operating characteristic curve</i>
SK	<i>Sallen-Key</i>
SOC	<i>system on a chip</i>
SVD	<i>Singular Value Decomposition</i>
SVDD	<i>support vector data description</i>

SUMÁRIO

	INTRODUÇÃO	14
1	DETECÇÃO DE FALHAS EM CIRCUITOS	16
1.1	Conceitos básicos	16
1.2	Revisão da literatura	18
1.3	Desafios nos diagnósticos de falhas em circuitos analógicos	19
2	FUNDAMENTOS DE RECONHECIMENTO DE PADRÕES	22
2.1	Conceitos básicos de reconhecimento de padrões	23
2.2	Classe e rótulos de classe e características	25
2.3	Conjunto de dados	26
2.4	Classificação, classificador e funções discriminantes	27
2.5	Taxonomia dos métodos de projeto do classificador	28
2.6	Classificadores lineares	30
2.7	Classificadores não paramétricos	32
2.8	Classificadores multinomial	33
2.9	Classificador parzen	34
2.10	A regra dos kvizinhos mais próximos	35
2.11	k vizinhos mais próximos	36
2.12	Matriz de confusão	37
3	CLASSIFICADORES DE CLASSE ÚNICA	40
3.1	Métricas para avaliação dos classificadores de classe única	42
3.1.1	<u>Cálculos dos erros</u>	42
3.1.2	<u>Área sobre a curva roc (AUC)</u>	43
4	SISTEMA PROPOSTO	44
4.1	Obtenção da resposta ao impulso	44
4.2	Extração de características usando o PAA	46
4.3	Arquitetura para detecção e diagnóstico de falhas usando classificadores	47
5	ESTUDOS DE CASO	49
5.1	Sallen-Key	50
5.1.1	<u>Identificação dos componentes com falhas simples</u>	51
5.1.1.1	Classificador multiclasse.....	52
5.1.1.2	Resultados do classificador multi-classe k-nn para o circuito SK com falhas simples.....	54
5.1.1.3	Comitê de classificadores de Classe única.....	55
5.1.1.4	Resultados do comitê de classificadores de classe única com falhas simples.....	56

5.1.2	<u>Múltiplas falhas</u>	58
5.1.2.1	Resultados classificador multiclasse k-nn para o Sallen-Key com múltiplas falhas.	58
5.1.2.2	Comitê de classificadores de classe única para o circuito SK para múltiplas falhas.	58
5.1.2.3	Resultados do comitê classe única knn_dd do circuito SK para múltiplas falhas.....	60
5.2	Filtro universal	61
5.2.1	<u>Descrição do circuito</u>	61
5.2.2	<u>Identificação dos componentes normais com falhas simples e falha múltipla</u>	62
5.2.2.1	Classificador multiclasse.....	63
5.2.2.2	Resultados do classificador multiclasse k-nn para o circuito ctsv.	64
5.2.2.3	Comitê de classificadores de Classe única	65
5.2.2.4	Resultados	66
5.2.3	<u>Múltiplas falhas</u>	69
5.2.3.1	Resultados do classificador multi-classe k-nn para o circuito ctsv com múltiplas falhas.	69
5.2.3.2	Comitê de classificadores de classe única para o circuito ctsv para múltiplas falhas.	70
5.2.3.3	Resultados do comitê classe única knn_dd do circuito ctsv para circuitos com múltiplas falhas.....	71
5.2.4	<u>Resultados obtidos pela análise de sinais</u>	73
6	CONCLUSÃO	76
	REFERÊNCIA	78
	ANEXO A	80

INTRODUÇÃO

O desenvolvimento de estratégias de teste para detectar e diagnosticar falhas em circuitos analógicos e de sinais mistos é uma tarefa desafiadora que tem encorajado uma boa quantidade de pesquisa, devido ao aumento do número de aplicações destes circuitos e ao alto custo dos testes. Muitas áreas, tais como, telecomunicações, multimídia e aplicações biomédicas, precisam de bom desempenho em aplicações de alta frequência, baixo ruído e baixa potência, que somente pode ser alcançado usando circuitos integrados analógicos e de sinais mistos. Assim, uma estratégia para detectar e diagnosticar falhas nesses circuitos é muito importante (Albustani, 2004). No passado, um circuito integrado era apenas um componente em um sistema, mas hoje o circuito integrado em si é o sistema inteiro (*System on a chip*). Com esse nível de integração, esse tipo de circuito gerou problemas difíceis de teste e projeto. Existem vários fatores que aumentam as dificuldades, tais como: a falta de bons modelos de falhas, falta de um padrão de projeto com vistas à testabilidade e o aumento da importância das falhas relacionadas ao tempo (Claasen, 2003). Portanto, a estratégia de testes para detecção e diagnóstico de falhas ainda é severamente dependente da perícia e da experiência que os engenheiros têm sobre as características do circuito. Sendo assim, a detecção e a identificação de falhas é ainda um processo iterativo e que consome bastante tempo. Um estudo na área de detecção e diagnóstico (Fenton, 2001) mostrou que, nas últimas décadas, uma boa quantidade de pesquisa em diagnósticos de falhas foi concentrada em desenvolver ferramentas que facilitassem as tarefas de diagnóstico. Embora tenha havido progressos importantes, essas novas tecnologias não tem sido largamente aceitas. Isso deve motivar os pesquisadores para investigar outros paradigmas e desenvolver novas estratégias para diagnósticos de falhas.

O uso de técnicas de inteligência computacional para diagnóstico é normalmente baseado na construção de modelos ou no uso de classificadores. O sucesso das abordagens baseadas na construção de modelos depende da qualidade do modelo obtido, que, no caso de um sistema complexo pode ser difícil de obter. Os classificadores procuram por comportamentos específicos de falhas e tornam-se vulneráveis quando existe superposição dos padrões de falha ou quando tem que tratar com padrões de falha que não foram apresentados a ele durante a fase de treinamento.

Classificadores de classe única podem ser treinados para resolver problemas de classificação binária onde apenas uma das classes é bem conhecida (Tax, 2001). Eles podem

ser organizados na forma de comitê de classificadores e com isso reduzir alguns dos problemas encontrados com classificadores multiclasse citados anteriormente.

Este trabalho apresenta o desenvolvimento de um sistema de detecção de falhas para circuitos lineares e invariantes no tempo baseado em um comitê de classificadores de classe única. O comportamento do circuito é representado pela Aproximação Agregada por Partes (PAA - *Piecewise Aggregate Approximation*) da resposta ao impulso do circuito. Os experimentos são realizados nos circuitos como citados: filtro passa-banda *Sallen Key* e o ctsv (*continuous-time state-variable*) que são três circuitos compostos por filtros passa baixa, passa alta e passa banda. Para cada um dos circuitos, foram realizados estudos de casos para avaliar a capacidade dos classificadores de classe única em detectar se o circuito possui ou não falha assim como a capacidade de sugerir o componente responsável pela falha.

Este trabalho está dividido da seguinte forma: o capítulo 1 aborda a detecção de falhas, os conceitos básicos, uma revisão na literatura e alguns desafios na detecção e no diagnóstico de falhas em circuitos analógicos; o capítulo 2 apresenta os fundamentos de reconhecimento de padrões, descreve os conceitos básicos de forma sucinta e alguns classificadores; o capítulo 3 descreve de forma sucinta os classificadores de classe única; o capítulo 4 apresenta a arquitetura utilizada pelo sistema proposto; o capítulo 5 descreve o estudo de caso e o capítulo 6 conclui este trabalho.

1 DETECÇÃO DE FALHAS EM CIRCUITOS

Esse capítulo pretende descrever os conceitos básicos de falhas com uma revisão de literatura e introduz os desafios na detecção e no diagnóstico de falhas.

1.1 Conceitos básicos

O termo *falha* é definido como uma condição anormal ou defeito (ISO/CD 10303), em um componente, equipamento ou sistema que pode conduzir ao mau funcionamento, isto é, uma diminuição parcial ou total na capacidade de desempenhar a função desejada por certo período de tempo.

Em circuitos analógicos, as falhas podem ser classificadas usando diferentes critérios. Em relação ao tipo de desvio, tem-se a falha *paramétrica* que é o desvio do parâmetro de um sistema no tempo, forçando-a assumir um valor que está fora de sua faixa nominal. Quando existe um desvio repentino muito grande do valor do parâmetro desejado, este é chamado de falha catastrófica. Este tipo de falha está associado à mudança da estrutura do sistema. Por exemplo, em circuitos elétricos, circuito aberto e curto-circuito são falhas catastróficas (DUHAMEL E RAULT, 1979).

No que diz respeito ao número de falhas que ocorrem num circuito, uma falha *simples* acontece quando apenas um parâmetro ou componente é alterado, enquanto uma falha *múltipla* acontece quando vários parâmetros são alterados simultaneamente (Duhamel e Rault, 1979). Duas falhas são *independentes* se não existem relações efeito e causa entre as ocorrências, caso contrário eles são chamadas de *dependentes*.

Observando uma falha ao longo do tempo, ela é chamada de *intermitente* quando ocorre durante certo período de forma aleatória e imprevisível alternando com comportamento normal. Quando a falha é *permanente*, somente o reparo do componente defeituoso pode fazer o circuito funcionar corretamente. O termo *transiente* é também usado por alguns autores para classificar as falhas causadas por uma mudança temporária no ambiente, enquanto outros autores usam o termo *incipiente* para identificar falhas que evoluem gradualmente tornando-se mais severas. (MANDERS et al., 2000).

Em relação à capacidade de detecção de falhas podem ser classificados como *mascarável*, *dominante*, *equivalente*, *não observável* ou *indetectável* e isolado ou *detectável*. As falhas *mascaráveis* são aquelas que, quando ocorrem ao mesmo tempo, podem compensar seus efeitos, tornando o sistema aparentemente livre de erros sobre certas condições. As falhas *dominantes* apresentam um efeito no circuito que sobressai sobre os defeitos de todas as outras falhas que acontecem ao mesmo tempo. Falhas *indistintas* ou equivalentes apresentam os mesmos efeitos sob certas condições, significando que elas não podem ser exclusivamente atribuídas a uma causa particular. As falhas *inconfundíveis* reveladas sob certas condições são chamadas *detectáveis*; e ao contrário são chamados de não observáveis ou falhas *indetectáveis* (SLAMANI E KAMINSKA, 1996).

A fim de descobrir se um circuito apresenta uma falha é necessário realizar o teste do circuito (Duhamel e Rault, 1979). Esse procedimento possui três passos: *detecção*, *localização* e *identificação*. A fase de *detecção* tenta determinar se realmente existe alguma falha no sistema. O segundo passo envolve a *localização* ou isolamento da falha, ou seja, descobrir os parâmetros ou componentes que estão causando a falha. Finalmente, na *identificação*, determinar o valor do parâmetro do defeito. O teste do circuito pode ter propósitos diferentes. Se o teste é realizado na linha de produção, provavelmente só interessa a fase de detecção, porque a finalidade do teste é ter certeza que todos os circuitos produzidos funcionam, mas não quer atrasar a linha de produção com procedimentos que consumam tempo como localização e identificação de falhas. A localização da falha será feita caso haja interesse em reparar a falha do circuito, enquanto a identificação será realizada quando o passo para localização da falha aponta repetidamente para um determinado componente ou parâmetro, que indica se o projeto ou processo do circuito de produção deve ser alterado.

Os testes realizados nos circuitos ajudam a encontrar e classificar em relação ao tipo de falha. O teste *funcional* (Duhamel e Rault, 1979) é feito quando se deseja verificar se o circuito responde corretamente a uma determinada função que esse circuito é capaz de realizar. O teste *paramétrico* é usado para checar se os parâmetros do circuito tal como tensão, corrente e impedância estão sob a tolerância especificada. A verificação do estado estático do circuito é feita pelo teste *estático* enquanto a observação das características dinâmicas é feita pelo teste *dinâmico* (DUHAMEL E RAULT, 1979).

Cada teste é realizado pela aplicação de um estímulo (Duhamel e Rault, 1979), ou seja, um sinal é aplicado na entrada do circuito. Um estímulo é classificado como *normal* (Duhamel e Rault, 1979) se este sinal é aplicado no circuito quando está em operação. Caso contrário, o sinal é chamado *artificial*, quando usado para a finalidade de teste.

O estímulo pode ser aplicado no circuito dentro do módulo e nesse caso é chamado *Built-in Self-Test* (BIST). Se um estímulo é aplicado por um equipamento externo, é chamado de *Automatic Test Equipment* (ATE).

Quando um estímulo é aplicado a um circuito sobre teste, os sinais de saída são medidos para verificar se o circuito tem o comportamento esperado. Isto pode ser feito usando um modelo e avaliando o erro entre o modelo e as saídas do circuito. Outra abordagem é construir um *dicionário de falhas* que é um conjunto organizado de comportamento do circuito sob diferentes situações de falha o qual pode ser usado para detectar e diagnosticar falhas no circuito.

1.2 Revisão da literatura

Nesse capítulo serão apresentados de forma sucinta alguns dos trabalhos e artigos relacionados à detecção e ao diagnóstico de falhas.

Apresentação do diagnóstico para falhas simples em elementos passivos em circuitos eletrônicos analógicos baseada na relação de tensão no nó. Este método consiste em duas partes: criação de um dicionário de falhas que descreve o estado nominal do circuito testado e contém parâmetros indiretos representando a respectiva falha e uma nova detecção de falha e um algoritmo de localização (CZAJA, 2008).

Diagnóstico de falha de circuitos analógicos é essencial para testes e manutenção em sistemas com sinais analógicos e mistos. A proposta é para o diagnóstico de múltiplas falhas em circuitos analógicos lineares no domínio da frequência. É aplicada a fórmula de *woodbury* para modificar a equação nodal para construir a equação de diagnóstico de falhas que relaciona o limite da resposta do circuito medido com as múltiplas falhas do circuito no modo linear. Para identificar diretamente os parâmetros defeituosos, é usada uma técnica de localização de grupo modificada para reduzir o custo de computação comparado às buscas combinacionais usadas nos métodos tradicionais de verificação de falhas. Apenas a medição de um nó é suficiente, mas são necessárias múltiplas excitações e assim fazer as medições correspondentes no nó acessível para identificação da falha (LIU E STARZYK, 2002).

(Czaja, 2009) Apresenta um método de detecção e localização de falhas em componentes analógicos com tolerância de elementos não defeituosos em sistemas embarcados com sinais mistos controlado por microcontrolador. O método consiste em três

estágios. No estágio de pré-teste é criado o dicionário de falhas. O estágio de medição baseia-se no tempo de duração da medida do sinal de saída dos comparadores analógicos implementados pelos recursos do microcontrolador. O tempo de resposta ao estímulo de onda quadrada aplicada à entrada da parte analógica com diferentes limiares de tensão. No último estágio, são realizadas a detecção e a localização das falhas pelo microcontrolador. Segundo (Czaja, 2009), a principal vantagem e novidade do método estão no fato que o BIST consiste apenas de comparadores analógicos e recursos internos do microcontrolador que já estão montados no sistema. Assim, essa abordagem simplifica a estrutura do projeto de BISTs, que permite diminuir os custos de testes.

O trabalho de (Costa et al., 2002) apresenta um testador analógico adaptativo. Na primeira etapa, o testador é capaz de detectar falhas em qualquer circuito linear aprendendo um comportamento de referência, na segunda etapa compara esse comportamento com a saída do circuito sob teste. Considerando a mesma estrutura básica, o método de diagnóstico consiste em injetar possíveis falhas no modelo matemático do circuito e depois comparar as saídas com a saída do circuito defeituoso real. O estudo de caso foi feito com um *biquad filter*. Foram considerados desvios paramétricos, catastróficos e falhas no amplificador operacional.

Liu et al., 2008, descreve que a extração de atributos é a chave em qualquer processo de reconhecimento de padrões. Não existe exceção no diagnóstico de falhas em circuitos analógicos, porque o diagnóstico de falhas é equivalente à questão do reconhecimento de padrões na natureza. (Liu et al., 2008) comenta algumas das várias abordagens para extrações de atributos no campo de diagnóstico de falhas em circuitos analógicos. Recentemente apareceram abordagens para extração de atributos baseado na entropia, na função *kernel*, na teoria fractal, em conjuntos brutos (*rough set*), além da abordagem *wavelet* que é largamente utilizada. Nesse trabalho também são discutidos as vantagens e desvantagens dessas abordagens. São indicadas as abordagens para extração de atributos, possíveis soluções e tendências de desenvolvimento.

1.3 Desafios nos diagnósticos de falhas em circuitos analógicos

O diagnóstico e a detecção de falhas em circuitos analógicos é uma tarefa bastante complexa. Existem muitos fatores que aumentam a dificuldade no teste destes circuitos tais

como: a dificuldade de medir correntes sem alterar as conexões (Bandler e Salama, 1985), a falta de bons modelos de falha, a falta de um padrão para projeto de circuitos analógicos com vistas à testabilidade e a crescente importância das falhas temporais (CLAASEN, 2003).

Outros fatores podem ser encontrados em (Mustapha e Bozena, 1992). Primeiro, os autores apontam que sistemas analógicos às vezes apresentam efeitos não lineares, ruídos e que os valores dos parâmetros podem sofrer grandes variações, o que faz com que métodos determinísticos normalmente sejam ineficientes; segundo, a distribuição estatística das falhas normalmente não é conhecida com precisão suficiente, o que dificulta o uso de métodos probabilísticos e terceiro, os equipamentos de teste automático (ATE – *automatic test equipment*) convencionais não possuem nem capacidade computacional nem de armazenamento para lidar com a crescente complexidade dos circuitos analógicos atuais. Conforme (Fanni et al., 2003) os métodos clássicos (identificação de parâmetros e dicionário de falhas) necessitam de grande poder computacional. Eles também apontam que um dos principais problemas no teste de circuitos é a presença de falhas não detectáveis. Isto pode acontecer devido à topologia do circuito e do número limitado de pontos de teste. Também é importante definir se a aplicação necessita de uma detecção de falhas *on line* ou apenas de um diagnóstico *off line*. No caso de procedimentos *on line*, estes não podem exigir um grande esforço computacional e nem utilizar outros sinais de entrada que aumentem a observabilidade da falha, a não ser que se utilize um módulo BIST (*built-in self-test*) integrado.

Uma pesquisa na área de detecção e diagnóstico em sistemas eletrônicos (Fenton et al., 2001) revelou que, nas últimas décadas, a pesquisa na área de diagnóstico de falhas se concentrou no desenvolvimento de ferramentas para simplificar o processo de diagnóstico. Embora tenham ocorrido progressos significativos, estas novas tecnologias ainda não foram amplamente aceitas. O uso de técnicas de inteligência computacional para diagnóstico, normalmente é baseado na construção de modelos ou no uso de classificadores. O sucesso das abordagens baseadas na construção de modelos depende da qualidade do modelo obtido, que, no caso de um sistema complexo pode ser difícil de obter. As abordagens que utilizam classificadores multiclasse, já foram aplicadas com sucesso na detecção de falhas de circuitos como citados em (Spina e Upadhyaya, 1997; Fanni et al., 1993; Z. Yi et al., 1993; Aminian, 2000 e 2001; Aminian et al., 2002 e Catelani, 2002). Associando as informações dos sinais de tensão e corrente do circuito com uma determinada falha, é possível treinar um classificador que receba como entrada características ou atributos extraídos dos sinais de tensão e corrente e forneça como saída a indicação da falha ocorrida. Entretanto, esta abordagem exige que as

classes de falha sejam cuidadosamente escolhidas. Uma escolha incorreta fará com que o classificador multiclasse tenha um desempenho ruim, pois não será capaz de indicar as falhas de forma correta. Além disso, o classificador multiclasse só pode indicar classes de falha para as quais ele tiver sido treinado, isto é, para as quais tenham sido apresentados ao classificador, exemplos destas classes de falha. Uma vez que a obtenção de todas as classes de falha possíveis é muito custosa e muitas vezes impraticável, normalmente este classificador multiclasse é treinado apenas com algumas classes de falha simples. Isto pode fazer com o classificador tenha um baixo desempenho caso ocorra uma falha não prevista.

Este problema pode ser contornado utilizando um comitê de classificadores, em particular de classificadores de classe única. Esta solução é promissora, mas ainda não foi devidamente explorada (Z. YI et. al., 2008).

2 FUNDAMENTOS DE RECONHECIMENTO DE PADRÕES

Segundo (Duda et al., 2002) as tarefas simples e corriqueiras como o reconhecimento do rosto de uma pessoa, de sua voz e a identificação de um objeto através do tato, escondem procedimentos complexos que são executados pelo cérebro humano. Durante o processo evolutivo, o sistema cognitivo desenvolveu uma sofisticada capacidade de reconhecimento de padrões que foi crucial para a sobrevivência dos seres humanos. Na tentativa de projetar máquinas que sejam capazes de realizar o reconhecimento de padrões de modo similar ao ser humano, os projetistas e desenvolvedores de sistemas se deparam com muitos problemas para construir tais sistemas e para compreender como o processo de reconhecimento se desenvolve. Existem diversas aplicações onde o reconhecimento de padrões exerce um papel central, com exemplo, pode-se citar: identificação de impressões digitais; reconhecimento automático de voz; reconhecimento de caracteres ópticos; identificação de sequência de DNA entre outros (DUDA et al., 2002).

Para (Júnior, 2004) quando o objetivo do reconhecimento de padrões é classificar objetos de interesse em uma categoria ou classe dentre um número finito de categorias ou classes, esses objetos de interesse são chamados genericamente de padrão. Para realizar qualquer classificação é necessário encontrar medidas e características inerentes a cada classe que possam ser usadas para diferenciar um objeto do outro. Essas características recebem o nome de atributos e para cada aplicação, diversas características podem ser utilizadas dependendo de quanta informação está disponível. Para aplicação de reconhecimento facial, como exemplo, utiliza-se a distância entre os olhos, a distância entre o ponto médio da distância dos olhos e o ponto médio da distância do queixo, entre outras medidas. O importante é utilizar os atributos que possam diferenciar o melhor possível os diversos objetos em estudo.

O modelo de classificação é a próxima etapa após a escolha dos atributos. Qualquer método que incorpora informação das amostras de treinamento no projeto de um classificador emprega aprendizado. Normalmente, um problema de reconhecimento de padrões possui um grau de dificuldade elevado, devendo-se, portanto, empregar a maior parte do tempo e dos esforços na etapa de aprendizagem. Para criar classificadores é necessário pressupor uma forma geral do modelo e do classificador e usar os padrões de treinamento para aprender ou estimar os parâmetros desconhecidos do modelo. O aprendizado refere-se ao algoritmo que de alguma forma reduza o erro em um conjunto de dados de treinamento. Existe uma variedade

de algoritmos que alteram os parâmetros de um classificador de forma a reduzir a medida de erro dentro do campo de reconhecimento de padrões estatísticos (DUDA et al, 2002).

2.1 Conceitos básicos de reconhecimento de padrões

Em Kuncheva (2004), a Figura 1 mostra as tarefas básicas e estágios para reconhecimento de padrões. Um usuário apresenta o problema de reconhecimento de padrões e um conjunto de dados disponível. Para resolver este problema, deve-se expressá-lo usando a terminologia de reconhecimento de padrões e usar as técnicas disponíveis e posteriormente comunicar ao usuário a solução encontrada.

Se o conjunto de dados não é fornecido, deve-se fazer um ensaio para consegui-lo. As características relevantes têm que ser nomeadas e medidas. O conjunto de características deve ser tão grande quanto possível contendo até características que podem não ser tão relevantes nesse estágio. Elas podem ser relevantes quando combinadas com outras características. A limitação para coleta de dados usualmente está relacionado ao custo de sua obtenção. Outra possível razão para tal limitação é a dificuldade de medir tais características ou atributos, como no caso de necessidade de realizar exames invasivos. Nem todas as características são igualmente relevantes. Algumas delas são importantes somente quando se relacionam a outras e algumas em um contexto particular podem ser apenas *ruídos*. Para melhorar a qualidade da descrição são usados os processos de extração e, posteriormente, de seleção de características.

Em reconhecimento de padrões existem duas grandes categorias: *não supervisionada* e *supervisionada*. Na categoria não supervisionada (chamado também de aprendizado não supervisionado), o problema é descobrir a estrutura do conjunto de dados, isto é, se existir alguma. Isso geralmente significa que o usuário quer saber se os dados formam grupos e quais características tornam os objetos semelhantes dentro do grupo e quais os tornam diferentes entre os demais grupos. A única indicação que aponta para um bom resultado é a estimativa subjetiva do usuário.

Na categoria supervisionada (chamado também de aprendizado supervisionado) cada objeto no conjunto de dados vem com um rótulo (*label*) de classe pré-atribuída. A tarefa é treinar um classificador para fazer a rotulagem de forma “sensata”. Para que isso seja possível, um conjunto de dados rotulados deve ser fornecido a uma máquina que tenha a capacidade de aprendizado, então o processo de aprendizado ocorrerá e será guiado pelo

desempenho do classificador, isto é, pela sua capacidade de reconhecer corretamente os padrões apresentados.

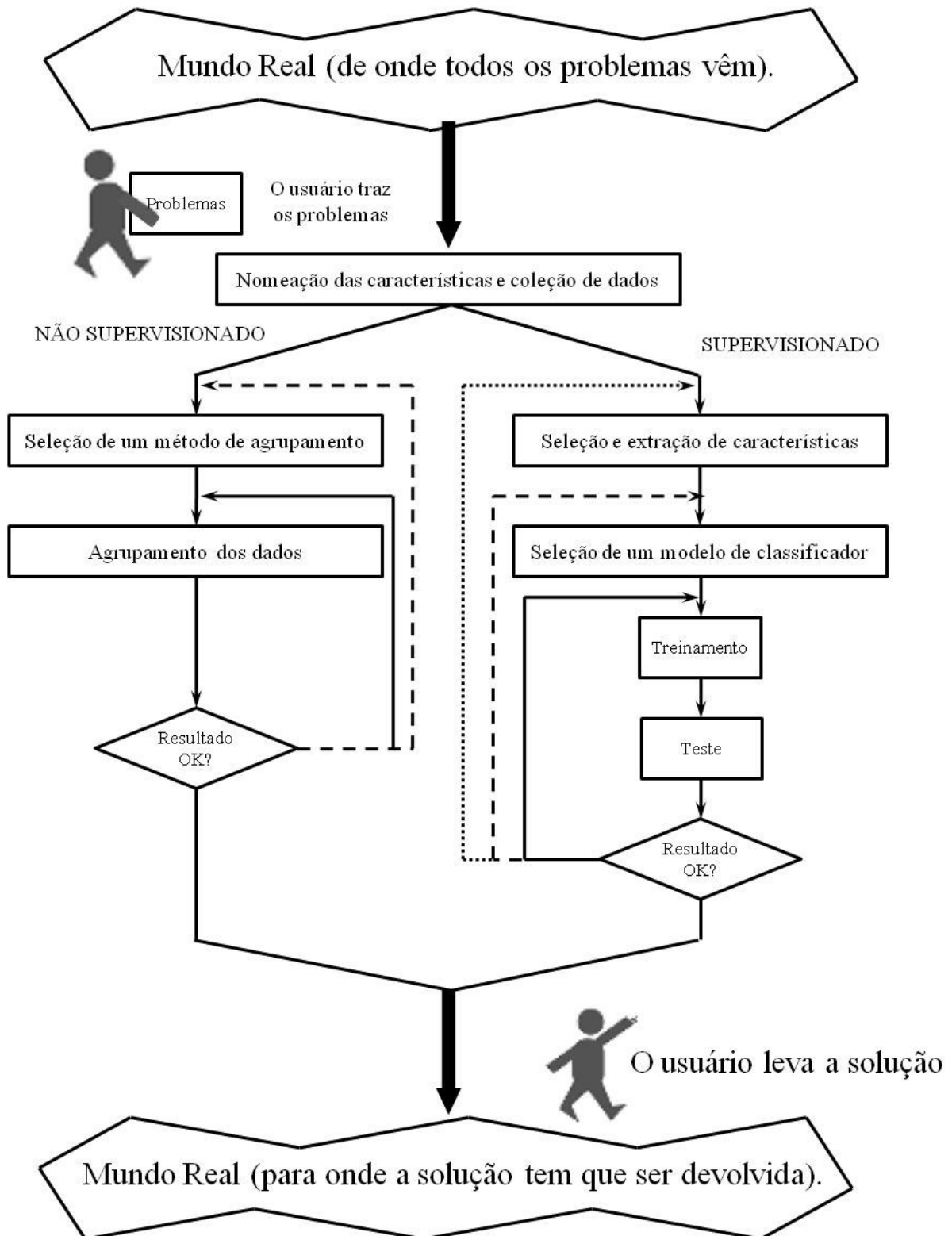


Figura 1 – O ciclo do reconhecimento de padrões (Adaptado de KUNCHEVA, 2004).

O núcleo do reconhecimento de padrão supervisionado de um modelo de classificador é formado por: seleção, treinamento e teste. Os laços que aparecem na Figura 1 na forma de linhas cheias, tracejadas e pontilhadas, que podem ser fechados em lugares diferentes. Pode-se decidir usar o mesmo modelo de classificador e refazer o treinamento somente com os parâmetros diferentes ou mudar o modelo do classificador. A seleção e a extração de características também poderão entrar no laço.

2.2 Classe e rótulos de classe e características

Intuitivamente, uma classe contém objetos similares e classes diferentes contêm os objetos que são diferentes entre si. Algumas classes têm um significado bem definido e em alguns casos são mutuamente exclusivas. Por exemplo, em uma verificação de assinatura, ela é genuína ou falsificada. Em outros problemas, as classes podem ser muito difíceis para definir, por exemplo, pode ser desejável estabelecer classes de baixo, médio e alto risco de enfarte, entretanto existem muitas dificuldades na interpretação de dados devido à variabilidade natural do objeto de estudo para poder atribuir as classes com segurança. Deve-se assumir que existam c possíveis classes no problema, rotuladas de ω_1 à ω_c , organizados como um conjunto de rótulos $\Omega = \{\omega_1, \dots, \omega_c\}$ em que cada objeto pertence apenas a uma classe.

Os objetos descritos por propriedades são conhecidos como características (também chamados de atributos). Essas características podem ser qualitativas ou quantitativas como ilustrada no diagrama da Figura 2. Se uma característica possui um grande número de valores possíveis, ela é tratada como quantitativa. Características qualitativas (categorias) são essas com pequenos números de possíveis valores com ou sem gradações. Um ramo de reconhecimento de padrões chamado de reconhecimento de padrões sintático (em oposição reconhecimento de padrões estatístico) lida exclusivamente com características qualitativas.

O reconhecimento estatístico de padrões opera com características numéricas. Esses incluem, por exemplo, altura, vazão de uma tubulação, lucro líquido de um banco nos últimos 12 meses, tamanho da folha de uma planta, etc. Os valores de características para um dado objeto são colocados em um vetor de dimensão $n \times 1 = [x_1, \dots, x_n]^T \in \mathfrak{R}^n$. O espaço real \mathfrak{R}^n é chamado de espaço de característica e cada eixo corresponde a uma característica em particular. A representação do número real $x \in \mathfrak{R}^n$ necessita de uma metodologia para

converter características qualitativas em quantitativas. Tipicamente, tais métodos são altamente subjetivos e heurísticos, por exemplo, aplicar uma prova é uma metodologia para quantificar o processo de aprendizagem do estudante. Existem outras características incomensuráveis típicas dos seres humanos que podem ser avaliadas intuitivamente, mas dificilmente explicadas. Dentre essas, pode-se citar: senso de humor, inteligência e beleza.

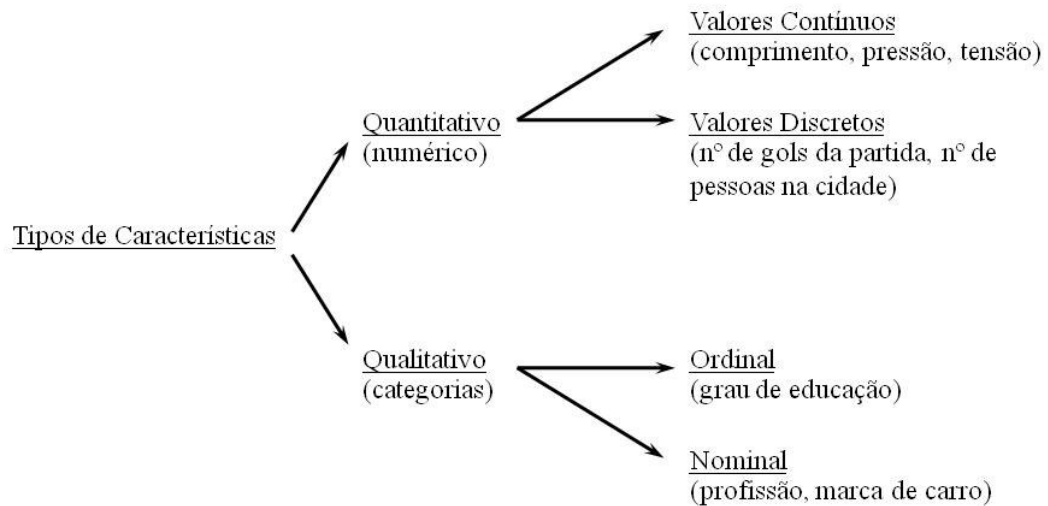


Figura 2 – Tipos de características (Adaptado de KUNCHEVA, 2004).

Às vezes um objeto pode ser representado por múltiplos subconjuntos de características. Por exemplo, usar três diferentes tipos de detecção para verificação de identidade como: face frontal, perfil da face e a voz. São realizadas as medidas específicas para extrair as características dos subconjuntos, e então, o vetor característico é composto por três sub-vetores, $x = [x^{(1)}, x^{(2)}, x^{(3)}]^T$.

2.3 Conjunto de dados

Para projetar um classificador a informação é usualmente colocada na forma de um conjunto de dados rotulados $Z = [z_1, \dots, z_n]^T$, $z_j \in \mathfrak{R}^n$. O rótulo de classe de z_j é denotado por $l(z_j) \in \Omega$, $j = 1, \dots, N$. A Figura 3 mostra um conjunto de exemplos de dígitos manuscritos os quais são rotulados em 10 classes pela máquina. Para construir um conjunto de dados, as imagens em preto e branco devem ser transformadas em vetores características. Nem sempre é fácil formular as n características que devem ser usadas no problema. No

exemplo da Figura 3 várias características discriminatórias podem ser nomeadas usando também várias transformações na imagem. Duas possíveis características são: o número de traços verticais e o número de círculos na imagem do dígito. Obter um bom conjunto de características pré-determina em grande parte o sucesso do sistema de reconhecimento de padrões.

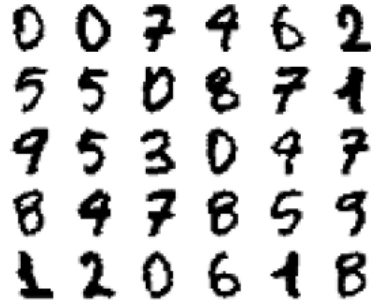


Figura 3 – Exemplos de dígitos manuscritos (Adaptado de KUNCHEVA, 2004).

2.4 Classificação, classificador e funções discriminantes

Um classificador é qualquer função conforme a Equação (1).

$$D: \mathfrak{R}^n \rightarrow \Omega \quad (1)$$

O modelo canônico de um classificador mostrado na Figura 4 considera um conjunto de c funções discriminantes $G = \{g_1(x), \dots, g_c(x)\}$, onde

$$g_i: \mathfrak{R}^n \rightarrow \mathfrak{R}, \quad i = 1, \dots, c \quad (2)$$

em que cada uma gera uma pontuação para a respectiva classe. Tipicamente (e mais naturalmente) X é rotulado na classe com mais alta pontuação. Essa escolha de pontuação máxima é chamada de *regra de associação máxima* conforme a Equação (3).

$$D(X) = \omega_i \in \Omega \Leftrightarrow g_i * (X) = \max_{i=1, \dots, c} \{g_i(X)\} \quad (3)$$

As funções discriminantes particionam o espaço de características \mathfrak{R}^n em regiões de decisões c ou regiões de classificação representada por $\mathcal{R}_1, \dots, \mathcal{R}_c$ conforme aparece na Equação (4).

$$\mathcal{R}_i = \{x | x \in \mathfrak{R}^n, g_i(x) = \max_{k=1, \dots, c} g_k(x)\}, \quad i = 1, \dots, c \quad (4)$$

onde:

- A seta mais larga significa entrada (vetor X de dimensão n);
- as saídas das caixas são os valores da função discriminante;

- $g_i(x)$ (escalares);
- a saída máxima do seletor é o rótulo de classe.

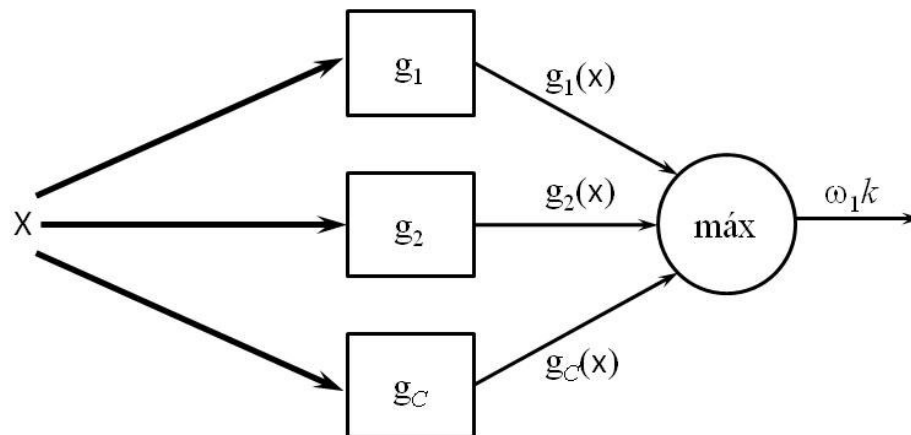


Figura 4 – Modelo canônico de um classificador (Adaptado de KUNCHEVA, 2004).

A região de decisão para a classe ω_i é o conjunto de pontos para o qual a $i^{\text{ésima}}$ função discriminante tem a mais alta pontuação. De acordo com a regra de associação máxima, todos os pontos na região de decisão \mathcal{R}_i são atribuídos na classe ω_i . As regiões de decisão são especificadas pelo classificador D , ou de forma equivalente, pela função discriminante G . Os chamados *limites de classificação* contêm os pontos para os quais os valores das funções discriminantes são iguais. Um ponto no limite pode ser atribuído para qualquer das classes limítrofes. Se uma região de decisão \mathcal{R}_i contém pontos de dados do conjunto rotulado \mathbf{Z} com classe de rótulo verdadeira $\omega_j, j \neq i$, as classes ω_i e ω_j são chamadas de sobreposição (*overlapping*). Note que, a sobreposição pode ocorrer para uma partição específica do espaço de características (definido por um certo classificador D), e pode ser eliminada se o espaço de características for particionado de outro modo. Se não existirem pontos idênticos com diferentes rótulos de classes em \mathbf{Z} , pode-se sempre particionar o espaço de características na região de classificação de forma que as classes não sobreponham. Normalmente quanto menor a superposição, melhor o classificador.

2.5 Taxonomia dos métodos de projeto do classificador

O reconhecimento de padrões estatístico fornece uma variedade de modelos de classificadores. A Figura 5 mostra uma possível taxonomia de métodos para projeto de

classificadores. Nas caixas aparecem modelos de classificadores representativos para as respectivas categorias.

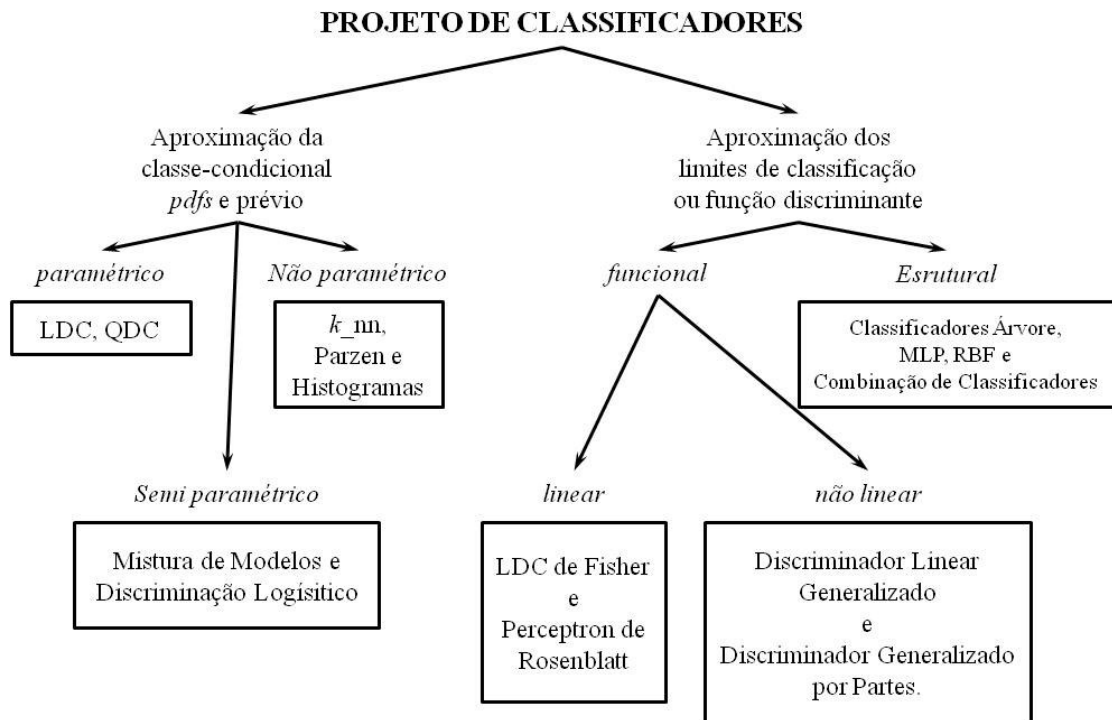


Figura 5 – Uma taxonomia de métodos para projeto de classificadores (Adaptado de KUNCHEVA, 2004).

De um lado do diagrama, encontram-se as abordagens baseadas em densidade de probabilidade (*pdf – probability density functions*). Uma vez que normalmente não se conhece a verdadeira probabilidade *a priori* $P(\omega_i)$ e nem as funções de densidade de probabilidade condicional $p(x|\omega_i)$ das classes nos problemas da vida real, pode-se somente projetar versões imperfeitas do classificador de Bayes. Uma solução é tentar estimar $P(\omega_i)$ e $p(x|\omega_i)$, $i = 1, \dots, c$ para Z , substituindo-os por $\hat{P}(x)$ e o $\hat{P}(x|\omega_i)$ para estimar a função discriminante $g_i(x) = P(\omega_i) p(x|\omega_i)$, $i = 1, \dots, c$. Ao aproximar $p(x|\omega_i)$ em função de x , os métodos para projetar classificadores dividem-se em dois grandes grupos: os paramétricos e os não paramétricos. Do outro lado do diagrama, estão os métodos de projeto de classificador que não são os métodos derivados das funções de densidade de probabilidade (*pdfs*) mas sim através da criação empírica dos limites de decisão ou função discriminante.

A distinção entre os grupos não é clara. Por exemplo, a rede *função de base radial* (RBF *radial base function*), que pertence ao grupo de aproximação da função discriminante pode ser movida para o grupo de aproximação funcional ou até para o grupo de modelagem semi-paramétrica de estimação *pdf*. Da mesma forma, o método *k_vizinhos mais próximos* (*knn*) é teoricamente ligado com a estimativa *pdf* não paramétrica e produz uma estimativa

direta da função discriminante e pode ser colocado no grupo de projeto estrutural para aproximar as funções discriminantes.

Não há consenso em uma taxonomia única e nem sobre a definição de paramétrico e não paramétrico, baseado ainda em (Kunsheva, 2004) são listados dois grupos com alguns tipos de classificadores.

O primeiro grupo cita cinco tipos de classificadores:

- probabilístico (LDC *linear discriminant classifier*, QDC *quadratic discriminant classifier* e Parzen);
- global (*multilayer perceptron* (MLP));
- local (rede neural com função de base radial (RBF))
- tipo vizinhos mais próximos (k_nn , redes neurais de aprendizagem através de quantização vetorial (LVQ *learning vector quantization*));
- por formação de regras (árvore de decisão binária e sistemas baseados em regras.)

O segundo grupo cita os seguintes classificadores

- classificadores baseados em estimativa de densidade:
 - paramétricos (LDC e QDC);
 - não paramétrico (k_nn , método *kernel*, misturas finitas, RBF).
- classificadores baseados em regressão:
 - paramétricos (regressão linear, regressão logística, MLP);
 - não paramétrico (busca de projeção, modelos aditivos).
- outros classificadores (por exemplo, baseado em protótipos: LVQ, k_nn para pequenos k).

2.6 Classificadores lineares

O tipo do classificador está ligado ao tipo da função discriminante usada. Para um conjunto de funções lineares do tipo $g_i: \mathfrak{R}^n \rightarrow \mathfrak{R}, i = 1, \dots, c$, pode ser considerado um classificador linear conforme Equação (5).

$$g_i(x) = w_{i0} + w_i^T x, \quad x, \quad w_i \in \mathfrak{R}^n, \quad w_{i0} \in \mathfrak{R} \quad (5)$$

No início dos estudos sobre reconhecimento de padrões, os classificadores lineares foram muito explorados desde 1936 com o desenvolvimento do discriminante linear de

Fischer. O classificador linear pode ser derivado como o classificador de erro mínimo (*Bayes*) para as classes normalmente distribuídas com matrizes de covariâncias iguais. Deve-se chamar esse modelo de classificador discriminante linear (LDC). O LDC é simples para calcular a partir dos dados e é razoavelmente robusto, isto é, os resultados podem ser surpreendentemente bons até quando as classes não têm distribuição normal.

Qualquer conjunto de funções discriminantes obtido por uma transformação monotônica da probabilidade *a posteriori* $P(\omega_i|x)$ constitui um conjunto ótimo em termos de mínimo erro. Pode-se formar um conjunto considerando que $P(\omega_i)$ é a probabilidade *a priori* para a classe ω_i e $p(x|\omega_i)$ é a função densidade de probabilidade condicional. Supõe-se que todas as classes são normalmente distribuídas com a média μ_i e a matriz covariância Σ_i , isto é $p(x|\omega_i) \sim N(\mu_i, \Sigma_i)$, $i = 1, \dots, c$. A Equação (6) se transforma na Equação (7).

$$g_i(x) = \log[P(\omega_i)p(x|\omega_i)], \quad i = 1, \dots, c \quad (6)$$

$$\begin{aligned} g_i(x) &= \log[P(\omega_i)] + \log \left\{ \frac{1}{(2\pi)^{n/2} \sqrt{|\Sigma_i|}} \exp \left[-\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \right] \right\} \\ &= \log[P(\omega_i)] - \frac{n}{2} \log(2\pi) - \frac{1}{2} \log|\Sigma_i| - \frac{1}{2} \log(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) \end{aligned} \quad (7)$$

Supondo que todas as matrizes covariância de classe sejam as mesmas, isto é $\Sigma_i = \Sigma$ e $p(x|\omega_i) \sim N(\mu_i, \Sigma)$. Abrindo os parênteses no último termo da Equação (7) e descartando todos os termos que não dependem de ω_i pode-se obter um novo conjunto de funções discriminantes conforme aparece na Equação (8) onde $w_{i0} \in \mathfrak{R}$ e $w_i \in \mathfrak{R}^n$ são coeficientes da função discriminante linear de g_i .

$$g_i(x) = \log[P(\omega_i)] - \frac{n}{2} \log(2\pi) - \frac{1}{2} \mu_i^T \Sigma^{-1} \mu_i + \mu_i^T \Sigma^{-1} x = w_{i0} + w_i^T x \quad (8)$$

Na realidade, as classes não são normalmente distribuídas e também não são valores verdadeiros conhecidos nem de μ_i nem de Σ_i . Pode-se ainda calcular os coeficientes de w_i e μ_i dos dados usando a estimativa da média e a matriz covariância, mas o classificador obtido não será equivalente ao classificador (*Bayes*) mínimo erro.

2.7 Classificadores não paramétricos

Nos classificadores paramétricos assume-se previamente que as classes seguem uma determinada distribuição e, teoricamente, eles são ótimos para o problema, porém, na maioria das vezes, há um desconhecimento sobre as funções de distribuição de probabilidade. Sendo assim, nos projetos não paramétricos, a função densidade de probabilidade $p(x|\omega_i)$ é estimada na vizinhança de x em \mathfrak{R}^n . A probabilidade p^* de que x está em certa região $R \subset \mathfrak{R}^n$ é mostrado na Equação (9).

$$p^* = P(x \in R) = \int_R p(u) du \quad (9)$$

Supondo que N amostras são retiradas da distribuição em questão. Usando a distribuição binomial com parâmetros (p^*, N) , a probabilidade que k amostras caíam exatamente em R é dado pela Equação (10) e o valor p^* pode ser estimado como a proporção dos pontos em R com relação ao número total de amostras como apresentado na Equação (11).

$$p_k = \binom{N}{k} (p^*)^k (1 - p^*)^{N-k} \quad (10)$$

$$p^* \approx \frac{k}{N} \quad (11)$$

Seja x , um ponto dentro da região R . Se R for suficientemente pequena, pode-se assumir que $p(u)$ é aproximadamente constante em R , p^* pode também ser aproximado como mostrado na Equação (12).

$$p^* \approx p(x) \int_R du = p(x)V_R \quad (12)$$

onde V_R é o volume de R em \mathfrak{R}^n . Juntando as Equações (11) e (12) e resolvendo em função de $p(x)$ obtém-se a Equação (13).

$$p(x) \approx \frac{k}{NV_R} \quad (13)$$

Quando N tende para infinito e a região R diminui para um ponto ($V_R \rightarrow 0$), a Equação (13) fornece o $p(x)$ exato. Esse resultado é um importante ponto de partida de vários métodos de classificação não paramétrica.

2.8 Classificadores multinomial

O classificador multinomial (conhecido também como método de histogramas) é derivado da Equação (13) e para isso, deve-se fixar N e V , e calcular k de Z . O espaço característica é dividido em compartimentos (células). Seja um compartimento B que contenha m pontos do conjunto de dados Z . Considere que m_i seja o número de pontos em B da classe ω_i , $i = 1, \dots, c$, $m = m_1 + \dots + m_c$. Da Equação (13) $p(x)$ passa a ser como na Equação (14).

$$p(x) \approx \frac{m}{NV_B} \quad (14)$$

onde V_B é o volume do compartimento e $p(x|\omega_i)$ é representado pela Equação (15).

$$p(x|\omega_i) \approx \frac{m_i}{N_i V_B} \quad (15)$$

Na Equação (15) N_i é o número de elementos da classe ω_i em Z . A probabilidade *a priori* normalmente é estimada pela Equação (16).

$$\hat{P}(\omega_i) = \frac{N_i}{N} \quad (16)$$

As probabilidades posteriores são obtidas como aparece na Equação (17).

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)} \approx \frac{\frac{m_i}{N_i V_B} \frac{N_i}{N}}{\frac{m}{NV_B}} \quad (17)$$

Daí aparece a Equação (18).

$$P(\omega_i|x) = \frac{m_i}{m} \quad (18)$$

A aproximação do classificador de Bayes é obtida utilizando os valores aproximados das probabilidades *a posteriori* para as classes como função discriminante. Assim, o compartimento é rotulado de acordo com o maior m_i , e todos os pontos em B obtém o mesmo rótulo de classe. Para classificar uma entrada x , o classificador encontra o compartimento onde x pertence e recupera o rótulo de classe do compartimento. Praticamente, este classificador é uma tabela de pesquisa (geralmente uma grande tabela) cuja precisão depende do número de compartimentos e do tamanho das amostras N .

A desvantagem deste tipo de classificador é que o número total de compartimentos cresce exponencialmente com a dimensionalidade n do problema, sendo que para M compartimentos por eixo, o número total de compartimentos será M^n . Isso requer um

conjunto de dados correspondentemente grande, caso contrário muitos compartimentos ficarão vazios ou poderão conter poucos elementos para fornecer estimativas confiáveis. Este fenômeno é conhecido como *maldição da dimensionalidade*.

2.9 Classificador parzen

Este modelo é baseado na Equação (13) onde N e V_R são fixos e k é encontrado a partir dos dados. Se $K(t), t = [t_1, \dots, t_n]^T \in \mathfrak{R}^n$ for a *função kernel* ou (janela de *Parzen*) que chega ao valor máximo quando está na origem, é não negativo e tem uma integral sobre \mathfrak{R}^n . O modelo mais simples da janela de *Parzen* tem a função mostrada na Equação (19).

$$K(t) = \begin{cases} 1, & \text{se } |t_i| \leq \frac{1}{2}, \forall i = 1, \dots, n \\ 0, & \text{caso contrário} \end{cases} \quad (19)$$

A função da Equação (19) define um hipercubo em \mathfrak{R}^n com o lado 1 centrado na origem. Para todos os pontos t dentro do hipercubo $K(t) = 1$ e para pontos em \mathfrak{R}^n no lado de fora do hipercubo, $K(t) = 0$. Se a janela estiver centrada em algum \mathbf{x} , pode-se, usando o conjunto Z para calcular a Equação (13) e levando em consideração que $V_R = 1$, a função densidade de probabilidade em \mathbf{x} calculada aproximadamente conforme a Equação (20)

$$p(\mathbf{x}) \approx \frac{k}{N} = \frac{1}{N} \sum_{j=1}^N K(Z_j - \mathbf{x}) \quad (20)$$

Essa fórmula pode ser interpretada de um modo diferente se admitir que existam vários hipercubos, cada um centrado em um ponto de Z . A aproximação é calculada como a proporção de cada hipercubo que contenha \mathbf{x} . A função núcleo (*kernel*) multidimensional centrada em $z_j \in \mathfrak{R}^n$ é usualmente expressa na forma de $(1/h^n)K((\mathbf{x} - z_j)/h)$ onde h é o parâmetro de suavização (*smoothing*) e a integral dessa função é igual a 1 conforme aparece na Equação (21).

$$\int_{\mathfrak{R}^n} \frac{1}{h^n} K\left(\frac{\mathbf{x} - z_j}{h}\right) d\mathbf{x} = 1 \quad (21)$$

Uma escolha comum da função núcleo é a *Gaussiana* multidimensional como aparece na Equação (22).

$$\frac{1}{h^n} K_G\left(\frac{\mathbf{x} - \mathbf{z}_k}{h}\right) = \frac{1}{h^n (2\pi)^{n/2} \sqrt{|S|}} \exp\left[-\frac{1}{2h^2} (\mathbf{x} - \mathbf{z}_k)^T S^{-1} (\mathbf{x} - \mathbf{z}_k)\right] \quad (22)$$

Na Equação (22), S é uma matriz covariância específica que determina a forma do *kernel*. As funções densidade de probabilidade são estimadas usando as amostras do conjunto Z pela Equação (23).

$$\hat{p}(\mathbf{x}|\omega_i) = \frac{1}{N_i} \sum_{l(z_j)=\omega_i} \frac{1}{h^n} K\left(\frac{\mathbf{x} - \mathbf{z}_j}{h}\right) \quad (23)$$

Onde N_i é o número de elementos de Z da classe ω_i . A estimativa é assintoticamente imparcial se o parâmetro de suavização h for uma função do número de amostras N_i , tal que o limite de $h(N_i)$ quando N_i tende a infinito como aparece na Equação (24).

$$\lim_{N_i \rightarrow \infty} h(N_i) = 0 \quad (24)$$

O classificador *parzen* é um modelo bastante eficiente, mas que apresenta as seguintes desvantagens:

- O classificador *parzen* precisa de todos os Z como conjunto de protótipo o que pode levar muito tempo quando o N for grande.
- A escolha de h é difícil: h pequeno conduz a aproximações ruidosas das pdfs e um h grande faz uma suavização excessiva das pdfs.

Na prática, a recomendação é tentar vários valores de h e selecionar o que produza o menor erro. Na literatura é proposto estreitar ou alargar os modelos de *parzen*. O valor real do classificador *parzen* vem do fato que este é o correspondente estatístico de muitos métodos de classificação tais como *função de base radial*, a *rede neural probabilística* e dos classificadores *fuzzy* (KUNCHEVA, 2004).

2.10 A regra dos kvizinhos mais próximos

A classificação não paramétrica é geralmente associada com a noção de *protótipo*. Deve-se pensar em um protótipo como um elemento representativo da classe. O rótulo da classe atribuído a um exemplo é baseado na similaridade deste exemplo para um ou mais protótipos. Tipicamente, a similaridade é definida em um sentido geométrico isso é, com base em certa distância. Quanto menor as distâncias, maior as semelhanças entre \mathbf{x} e o protótipo.

O classificador por histograma pode ser visto como um classificador protótipo. Pode-se colocar um protótipo no centro geométrico de cada compartimento. A entrada \mathbf{x} é considerada similar ao protótipo, se estiver no compartimento no qual esse protótipo está localizado. O rótulo da classe desse compartimento torna-se o rótulo da classe de \mathbf{x} . O classificador *parzen* também pode ser considerado um classificador protótipo. Nesse caso, os protótipos são todos os elementos de Z . Os pesos de cada classe são determinados de acordo com a soma das distâncias entre \mathbf{x} e o protótipo. As pontuações das classes são então comparadas para achar a classe com o máximo suporte (isto é, com a maior função discriminante). O exemplo clássico do classificador protótipo é o *k_vizinhos mais próximos* (*k_nn*).

2.11 k vizinhos mais próximos

Teoricamente, o *k_nn* é uma das técnicas de classificação mais elegantes e simples. Considerando $V = \{v_1, \dots, v_v\}$ seja o *conjunto de referência* rotulado contento v pontos em \mathfrak{R}^n , que na verdade são os protótipos. Os protótipos são rotulados na classe c , isto é, para qualquer $v_i \in V$, o rótulo da classe $l(v_i) \in \Omega$. No projeto clássico do vizinho mais próximo, V é a totalidade de Z . Para classificar uma entrada \mathbf{x} , os protótipos k mais próximos são obtidos a partir de V junto com seus rótulos de classes. A entrada \mathbf{x} é rotulada para a classe representada em maior número entre os *k_vizinhos mais próximos*.

Para chegar neste método de classificação fixa-se k e N da Equação (13) e varia-se V_R . Considerando a distância *Euclidiana*, seja R a hiper-esfera contendo exatamente k dos elementos do conjunto de referência V . A *função densidade de probabilidade* é aproximadamente como aparece na Equação (25).

$$p(\mathbf{x}) \approx \frac{k}{NV_R} \quad (25)$$

Considerando que k_i é o número de elementos em R da classe ω_i , a *pdf* da classe para ω_i , $i = 1, \dots, c$, aproximada em R aparece na Equação (26).

$$p(\mathbf{x}|\omega_i) \approx \frac{k_i}{N_i V_R} \quad (26)$$

Onde N_i é o número de elementos da classe em Z . Usando a Equação (16) para estimar $P(\omega_i)$, as probabilidades a posteriori são obtidas na Equação (27).

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)} \approx \frac{\frac{k_i}{N_i} \frac{N_i}{N}}{\frac{k}{NV_B}} \quad (27)$$

portanto a probabilidade *a posteriori* é a que aparece na Equação (28).

$$P(\omega_i|x) \approx \frac{k_i}{k} \quad (28)$$

O erro mínimo (Bayes) do classificador usando a aproximação da Equação (28) atribuirá \mathbf{x} à classe com a maior probabilidade *a posteriori*, isto é, a maior classe representada entre os k vizinhos mais próximos de \mathbf{x} . A região de R e o volume V_R são específicos respectivamente para cada $\mathbf{x} \in \mathfrak{X}^n$ e um conjunto de dados Z . A regra de classificação para o k_{nn} , no entanto, atribui o rótulo da classe usando somente números k_i , $i = 1, \dots, c$, de modo que o rótulo vencedor não dependa de V_R .

O k_{nn} é ótimo quando $N \rightarrow \infty$ e $V_R \rightarrow 0$. A expressão $V_R \rightarrow 0$ é equivalente ao $k \rightarrow \infty$ e $k/N \rightarrow 0$. Isto é, a taxa de erro do classificador k_{nn} $P_{k_{nn}}$ é satisfeta conforme a Equação (29).

$$\lim_{\substack{N \rightarrow \infty \\ k \rightarrow \infty \\ k/N \rightarrow 0}} P_{k_{nn}} = P_B \quad (29)$$

Onde o P_B é o erro de Bayes. Quando k é 1 (pela regra do vizinho mais próximo significa que é 1_{nn}) e $N \rightarrow \infty$, a taxa de erro $P_{1_{nn}}$ é limitado para cima com duas vezes a taxa de erro de Bayes que é mostrado na Equação (30).

$$P_{1_{nn}} \leq 2P_B \quad (30)$$

2.12 Matriz de confusão

A matriz de confusão MC é uma ferramenta de análise e avaliação do processamento dos classificadores seja multiclasse ou classe única. Devido ao extenso comprimento, a MC é mostrada em duas partes na Figura 6 (a) e (b) onde (b) é o complemento do comprimento de (a). O classificador é treinado com uma quantidade de elementos rotulados em classes e posteriormente esse classificador deve testar outra quantidade de elementos também rotulados e logo após o teste é formada a MC. Para entender as informações contidas na MC é importante conhecer as disposições numéricas como descrito a seguir: A primeira coluna da Figura 6 (a) mostra os rótulos (classes) verdadeiros; a primeira linha mostra os rótulos

(classes) nos quais os elementos testados foram estimados pelos classificadores; a última linha representa o total de elementos testados que foram estimados naquela classe, por exemplo: 344 elementos foram estimados pelo classificador como pertencentes à classe 1 sendo que 166 realmente pertencem à classe 1 e os 168 elementos restantes foram equivocadamente estimados como pertencentes à classe 1 porém, são da classe 10.

	True Labels		Estimated Labels						
	1	2	3	4	5	6	7	8	9
1	166	0	0	0	0	0	0	0	0
2	0	162	0	0	0	0	0	0	0
3	0	0	138	0	0	0	0	0	0
4	0	0	0	164	0	0	0	0	0
5	0	0	0	0	300	0	0	0	0
6	0	0	0	0	0	300	0	0	0
7	0	0	0	0	0	0	92	0	0
8	0	0	0	0	0	0	0	300	0
9	0	0	0	0	0	0	0	0	300
10	168	0	0	0	0	0	0	0	0
11	0	138	0	0	0	0	0	0	0
12	0	0	138	0	0	0	0	0	0
13	0	0	0	167	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	4	0	0
17	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	45	0	0
Totals	334	300	276	331	300	300	141	300	300

(a)

	True										Totals
	10	11	12	13	14	15	16	17	18	19	
1	134	0	0	0	0	0	0	0	0	0	300
2	0	138	0	0	0	0	0	0	0	0	300
3	0	0	162	0	0	0	0	0	0	0	300
4	0	0	0	136	0	0	0	0	0	0	300
5	0	0	0	0	0	0	0	0	0	0	300
6	0	0	0	0	0	0	0	0	0	0	300
7	0	0	0	0	0	0	82	0	0	126	300
8	0	0	0	0	0	0	0	0	0	0	300
9	0	0	0	0	0	0	0	0	0	0	300
10	132	0	0	0	0	0	0	0	0	0	300
11	0	162	0	0	0	0	0	0	0	0	300
12	0	0	162	0	0	0	0	0	0	0	300
13	0	0	0	133	0	0	0	0	0	0	300
14	0	0	0	0	300	0	0	0	0	0	300
15	0	0	0	0	0	300	0	0	0	0	300
16	0	0	0	0	0	0	296	0	0	0	300
17	0	0	0	0	0	0	0	300	0	0	300
18	0	0	0	0	0	0	0	0	300	0	300
19	0	0	0	0	0	0	0	0	0	255	300
Total	266	300	324	269	300	300	378	300	300	381	5700

(b)

Figura 6 – Exemplo de matriz de confusão (a) primeira parte de uma matriz de confusão; (b) segunda parte de uma matriz de confusão.

Na segunda linha da Figura 6 (a) aparecem os elementos pertencentes à classe 1 (que se estende à Figura 6 (b)). No final da matriz (b), na última coluna (*Totals*) aparece o total de elementos pertencentes à verdadeira classe 1 (300 elementos) que foram estimadas na seguinte forma: 166 elementos foram estimados corretamente como pertencentes à classe 1 e 134 elementos foram estimados erroneamente como pertencentes à classe 10 e assim por

diante. Na última célula da matriz, isto é, a última linha e a última coluna contêm o total de elementos testados Figura 6 (b).

3 CLASSIFICADORES DE CLASSE ÚNICA

Segundo (Tax, 2001) classificadores de classe única são desenvolvidos para resolver um tipo especial de problema de classificação de duas classes, onde existe um grande número de amostras de apenas uma das classes (*target*), enquanto a outra (*outlier*) é amostrada escassamente, ou pode ser totalmente ausente. Pode ser que exemplos desta classe sejam muito difíceis de obter, ou que para se obter as medidas dos exemplos desta classe seja muito custoso. O elemento de uma determinada classe em um classificador de classe única pode se apresentar em quatro situações:

- Na de verdadeiro positivo, o elemento em questão pertence à classe *target* e é classificado como pertencente à classe *target* nessa situação, ela não contribui com o erro porque o elemento está classificado corretamente.
- Na de verdadeiro negativo, o elemento não pertence à classe *target* e é classificado como não pertencente à classe *target* nessa situação, como a situação anterior, também não contribui com o erro porque o elemento está classificado corretamente.
- Na de falso positivo, o elemento pertence à classe *outlier* e é classificado como pertencente à classe *target* e nesse caso, o elemento é erroneamente classificado.
- Na de falso negativo, o elemento pertence à classe *target* e é classificado como pertencente à classe *outlier* nesse caso, o elemento é erroneamente classificado e contribui com a fração de rejeição (são objetos *target* classificados como *outlier*).

Em princípio, um classificador de classe única deve ser capaz de ser treinado exclusivamente com base nos exemplos do *target*. Um bom classificador de classe-única apresenta, dois tipos de erros que devem ser minimizados, a fração de falsos positivos e a fração de falsos negativos. O falso negativo é causado pela rejeição dos objetos *target* e o falso positivo é gerado pela aceitação de objetos *outlier*. Para que seja possível estabelecer um compromisso entre esses dois erros, um deles deve ser definido *a priori*. Devido à facilidade estimar o erro na classe *target*, supõe-se que o erro *target* já está definido. Este erro é também conhecido como fração de rejeição.

Os classificadores de classe única podem ser agrupados em três categorias: classificadores com *estimativa de densidade*, classificadores com *determinação de vizinhança* e classificadores *de reconstrução* (Z. YI et al., 2008).

Os classificadores que utilizam estimativa de densidade estão entre as mais simples e possuem um bom desempenho quando o número de exemplos da classe *target* é suficiente e o modelo de probabilidade encaixa muito bem com os dados obtidos. Exemplos deste tipo de classe são: o Gaussiano; Mistura de Gaussianas e Parzen (TAX, 2001).

No caso do classificador gaussiano, a classe *target* é modelada como uma distribuição Gaussiana. É usada a distância *Mahalanobis* ao invés do modelo de distribuição Gaussiana completa, para evitar as instabilidades numéricas, Equação (31) (TAX, 2006).

$$f(X) = (X - \mu)^T \Sigma^{-1} (X - \mu) \quad (31)$$

Onde: a média μ e a matriz covariância Σ são amostras estimadas.

O classificador é definido na Equação (32).

$$h(X) = \begin{cases} target & \text{if } f(X) \leq \theta, \\ outlier & \text{if } f(X) > \theta. \end{cases} \quad (32)$$

O limiar θ é ajustado de acordo com erro de *target* predeterminado.

Os classificadores por vizinhança não necessitam da estimação de distribuição completa dos objetos *target*. A classificação é feita com base na proximidade dos componentes da classe *target* utilizada no treinamento. Eles são mais vantajosos quando a amostra disponível é limitada e a exata densidade de distribuição do *target* é desconhecida. O knn é um exemplo, desta categoria de classificador. Ele é um modelo simples e avalia um objeto de teste X pelo cálculo da distância para o vizinho mais próximo no conjunto de treinamento. Esta distância é comparada com a distância deste vizinho mais próximo encontrado com distância a outros elementos do conjunto de treinamento. Quando mais de um vizinho é utilizado, a distância utilizada para comparação é dada pela média das distâncias aos k vizinhos mais próximos.

O classificador de reconstrução não se esforça para encontrar uma determinada característica (distribuição ou limite) dos dados, mas prefere modelar o processo de geração de dados. Quando o modelo de dados é construído, um pequeno erro médio de reconstrução indica qual o objeto se encaixa melhor no modelo, e, o mais provável é que ele seja da classe *target*. Os classificadores de reconstrução são robustos para generalização e ruim quando o tamanho de amostra é escasso e não apresentam um bom desempenho quando os modelos não ajustam bem aos dados. No classificador *k-Means*, os dados são descritos pelo k grupos e os centros dos grupos são colocados de tal forma que as distâncias para o centro sejam minimizadas, Equação (33).

$$\varepsilon_{k-means} = \sum_i (\min_k \|X_i - \mu_k\|^2) \quad (33)$$

3.1 Métricas para avaliação dos classificadores de classe única

Os classificadores de classe única necessitam de valores ou métricas para que eles tenham o melhor desempenho, dentro desse item estão descritos quais são as métricas e como são calculadas.

3.1.1 Cálculos dos erros

Tax, 2008, afirma que para avaliar os classificadores de classe única, primeiro é preciso determinar a porcentagem de falsos positivos e de falsos negativos. Quando se fornece apenas os objetos *target*, não se consegue estimar esses erros. Sendo assim, a fração de falsos negativos é definida a priori (fração de rejeição) para a construção do modelo (classificador). Uma vez que o modelo foi determinado, ele pode ser avaliado em um conjunto de teste e calcular a porcentagem de falsos positivos e falsos negativos.

Ainda conforme (Tax, 2008) na literatura existem duas outras medidas que são geralmente usadas: o *precision*, que é definido na Equação (34), e o *recall*, que é basicamente a taxa de verdadeiros positivos e é mostrado na Equação (35).

$$precision = \frac{\text{números de previsões } targets \text{ corretos}}{\text{número de previsões de } target} \quad (34)$$

$$recall = \frac{\text{números de previsões de } targets \text{ corretos}}{\text{números de exemplos } targets} \quad (35)$$

A partir das definições de *precision* e *recall* pode-se definir *F1* que é uma métrica de desempenho que está definida na Equação (36). O *F1* relaciona as métricas *precision* e *recall* e essas duas métricas possuem valores no intervalo entre zero e um. Considerando apenas a relação da multiplicação pela soma do *precision* e *recall* e se os valores forem o máximo, isto é, *precision* = 1 e *recall* = 1, o valor máximo dessa relação será 0,5 e com o intuito de normalizar a métrica *F1* = 1, a relação entre o numerador e o denominador é multiplicada por dois.

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \quad (36)$$

3.1.2 Área sobre a curva roc (AUC).

ROC significa (*receiver operating characteristic curve*) ou curva característica de operação do receptor. Em (Tax, 2006) comenta que um bom classificador terá sempre uma fração de falsos positivos e uma fração de falsos negativos porque o erro na classe *target* pode ser estimada relativamente bem, pois para os classificadores de classe única considera-se que o limiar seja definido antecipadamente no erro *target*. Variando o limiar e medindo o erro nos objetos *outlier* obtém-se uma curva ROC. Esta curva, conforme aparece na Figura 7, mostra como a variação de falsos positivos pela variação das frações de falsos negativos. Quanto menores forem essas frações, melhor classificador de classe única. Como é difícil comparar duas curvas ROC, normalmente se adota como métrica o valor da área sob a curva.

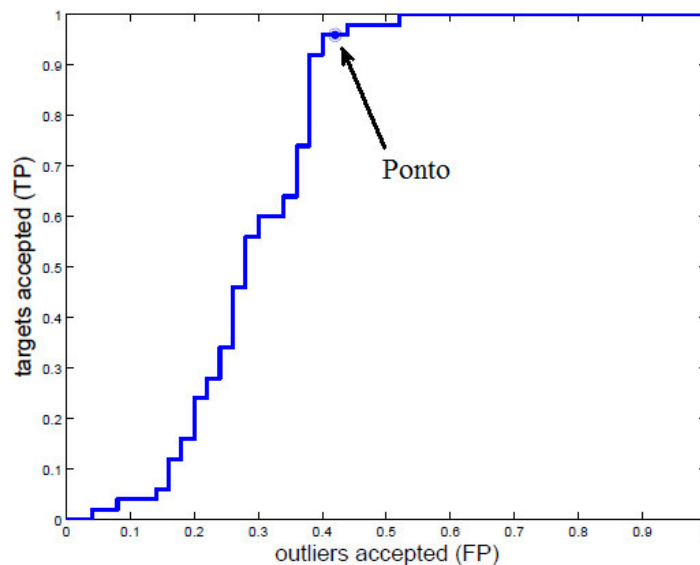


Figura 7 – Curva característica com o ponto de operação indicado pelo ponto.

4 SISTEMA PROPOSTO

Esse capítulo descreve o sistema proposto, os recursos utilizados e quais as etapas de processamento envolvidas para o desenvolvimento do trabalho.

A estrutura do sistema detector de falhas do circuito pode ser vista na Figura 8. Dependendo da complexidade do circuito, ela pode ser implementada em um BIST (*Built In Self Test*) ou em um sistema ATE (*Automatic Test Equipment*). No centro da Figura 8, tem-se um circuito linear e invariante no tempo. Ele possui m entradas e n nós observáveis (saídas). É possível definir uma matriz de $H(m,n)$, onde cada elemento, elemento $h_{i,j}$ representa a função de transferência entre a entrada i e a saída j .

O teste funcional é realizado pelo módulo de detecção de falhas (FDM) conforme aparece na Figura 8. Este módulo é responsável por controlar os multiplexadores para colocar o circuito no modo de teste e selecionar o par entrada e saída desejado. A abordagem adotada usa a resposta ao impulso, que está intimamente ligada à função de transferência do circuito e pode ser usada para caracterizar o comportamento dinâmico de um circuito linear e invariante no tempo. Se o sinal de teste de entrada for um degrau, então o sinal da saída será a resposta ao degrau. Se esta for derivada, obtém-se a resposta ao impulso na saída. O extrator de atributos recebe o sinal analógico de saída do diferenciador e tem por finalidade extrair os atributos do sinal de saída para que estes sejam apresentados às duas arquiteturas de detecção com classificadores (comitê de classe única e ao multiclasse), que serão responsáveis pela detecção de falhas.

4.1 Obtenção da resposta ao impulso

Segundo (SINGH, et al, 2004) qualquer LTI (*linear time invariant*) sistema linear invariante no tempo, pode ser completamente caracterizado pela *Resposta ao Impulso* RI chamada de $h(t)$. Onde a resposta ao impulso $h(t)$ é a saída do sistema à função impulso $\delta(t)$ como mostrado na Figura 9 e na Equação (37). A função da Equação (37) é matematicamente chamada de convolução onde $x(t)$ é o sinal de entrada, $y(t)$ é o sinal de saída e o $h(t)$ é o RI. O estado estacionário da resposta em frequência, $H(j\omega)$ de um sistema pode ser obtido pelo desempenho da transformada de Fourier na resposta ao impulso.

Ela pode ser usada para inferir os parâmetros de desempenho do circuito ao invés de medi-los diretamente.

Teoricamente, a RI de um sistema LTI pode ser derivada pela simulação de um impulso na entrada, Figura 9. No entanto, gerar uma função impulso que possa ser aceitável na prática é bastante difícil. Sendo assim, foram desenvolvidos diversos métodos para obter a resposta ao impulso, dentre os quais pode-se citar: obtenção da resposta ao impulso a partir da resposta ao degrau, testes de oscilação senoidal, convolução na frequência e técnicas de correlação cruzada (Singh, et al, 2004). Neste trabalho, o circuito a ser testado é estimulado por uma entrada degrau. A seguir, a resposta ao degrau é derivada através de um circuito diferenciador. O projeto deste circuito é muito importante na aquisição da resposta ao impulso. Ele é responsável por realizar a operação de diferenciação na faixa de frequência desejada, atenuar o ruído em alta frequência e limitar o ganho para evitar operação na faixa não linear.

$$y(t) = h(t) \otimes x(t) \quad (37)$$

$$\int_{-\infty}^{\infty} h(\tau) x(t - \tau) d\tau$$

$$\int_{-\infty}^{\infty} h(t - \tau) x(\tau) d\tau$$

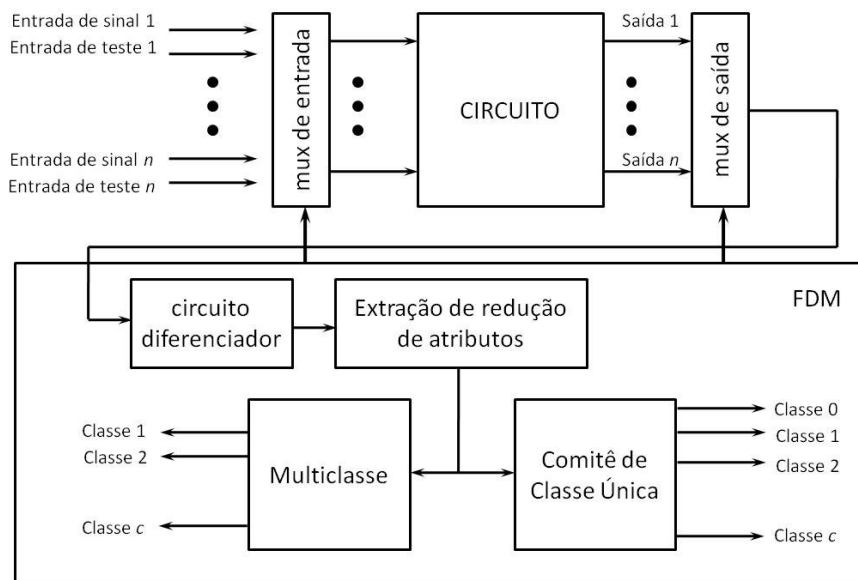


Figura 8 – Diagrama em blocos do detector de falhas



Figura 9 – Representação da resposta ao impulso de um sistema LTI.

4.2 Extração de características usando o PAA

Uma vez que a resposta ao impulso está disponível, realiza-se a extração de características desse sinal aplicando uma Aproximação Agregada por Partes (*Piecewise Aggregate Approximation – PAA*) (Keogh et al., 2000). Esta é uma técnica de redução de dimensão que é bastante simples de entender e implementar. Ela permite diferentes medidas de distância e apresenta resultados competitivos quando comparada a transformações mais sofisticadas como a decomposição por valor singular (SVD – *Singular Value Decomposition*) e as Transformadas de *Fourier* e *Wavelet*, na tarefa de indexação das séries temporais (Keogh et al., 2000). Uma série temporal (sinal) $X = x_1, \dots, x_n$ pode ser representada no espaço N , $N \leq n$, por $\bar{X} = \bar{x}_1, \dots, \bar{x}_n$, onde cada elemento pode ser calculado conforme a Equação (38).

$$\bar{x}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j \quad (38)$$

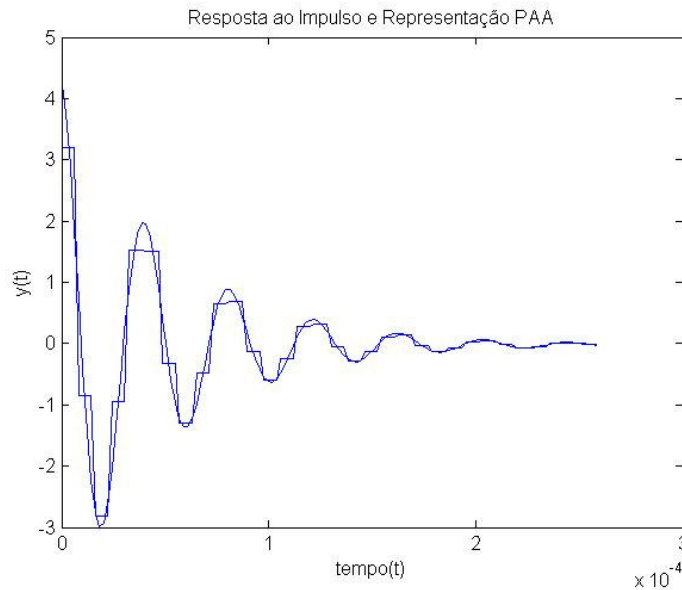


Figura 10 – Representação PAA de um sinal sobreposta ao sinal impulso original.

A Equação (38) indica que o sinal é dividido em N janelas e que o valor médio representa todos os pontos nesta janela. A Figura 10 mostra um sinal e sua respectiva representação PAA.

4.3 Arquitetura para detecção e diagnóstico de falhas usando classificadores

Existem duas arquiteturas que podem ser utilizadas no diagnóstico de circuitos utilizando classificadores. A primeira delas utiliza o classificador multiclasse que pode ser visto na Figura 11. Esta arquitetura, conforme (Fanni et al., 1993 e Salat, 2003), é bastante utilizada em problemas de diagnóstico. Este classificador possui tantas saídas quantas forem as diferentes classes que representam os possíveis comportamentos do circuito. Quando a entrada pertencer à classe de comportamento i então o *status* da saída i deve ser igual a 1 e todas as outras devem ser iguais a 0. É importante observar que apenas uma das classes definidas pode ser escolhida na classificação da entrada. Os classificadores multiclasse funcionam muito bem quando as diferentes classes de comportamento são bastante distintas, entretanto, se as classes estiverem muito próximas ou caso haja uma sobreposição entre elas, o classificador terá dificuldades de distingui-las. Outro ponto importante é a incapacidade do classificador multiclasse tratar classes de falha não previstas no seu treinamento e quando acontece tal situação, o classificador irá encaixar a falha não prevista em uma das classes existentes.

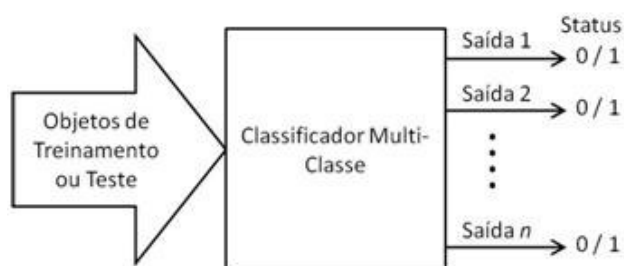


Figura 11 – Bloco do classificador multiclasse.

A outra arquitetura consiste em utilizar uma combinação ou comitê de classificadores para solucionar os problemas relatados na arquitetura multiclasse. Os classificadores do comitê são construídos usando classes de falhas predefinidas.

Os classificadores do comitê classe única como aparece na Figura 12, cujo treinamento é realizado por exemplos das classes de falhas então, para a classe com *label* 1 a saída 1 do classificador recebe o nível lógico *um* e as demais *zero*, para a classe com *label* 2 a saída 2 do classificador recebe o nível lógico *um* e as demais *zero* e assim por diante de forma que, para qualquer classe de falha com *label* n quando treinada, a saída n relativa ao *label* n recebe o nível lógico 1 e as demais recebem o nível lógico 0. O classificador de classe única,

como citado no item 3, precisa apenas dos elementos *target* sendo assim, necessita rejeitar alguns objetos *target* (falsos negativos) provocado pela taxa de rejeição e isso se torna uma desvantagem, porém ele ainda competitivo quando comparados ao classificador multiclasse. O classificador classificará um objeto na classe *zero* (classe desconhecida) ou quando este for de uma classe de falha para a qual o comitê não foi treinado. Caso mais de um classificador no comitê identificar o objeto como pertencente à sua classe, ou seja, mais de uma saída igual a *um*, o comitê vai sugerir através de códigos (representando duas ou mais classes) que esse objeto possa pertencer a uma das classes sugeridas. A utilização de classificadores de classe única em comitê é vantajosa, pois eles são treinados utilizando somente as informações pertinentes à sua classe. Se houver amostras suficientes relacionadas a uma classe em particular, basta retreinar o classificador com os elementos dessa classe.

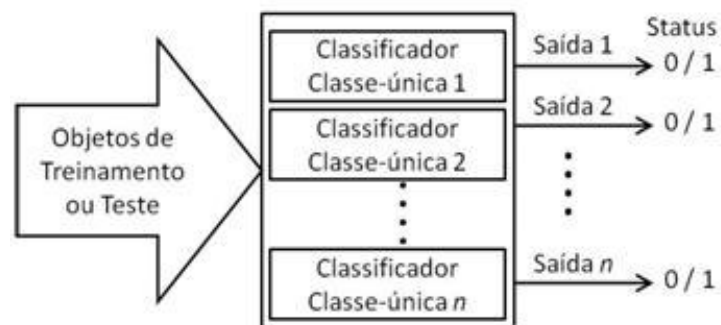


Figura 12 – Arquitetura do comitê classificador combinado classe única.

5 ESTUDOS DE CASO

Para realizar a avaliação das diferentes arquiteturas de classificadores alguns experimentos foram realizados utilizando dois circuitos diferentes: os filtros passa-banda *Salley-Key* e o Filtro Universal ou *ctsv* (*continuous-time state-variable*). Nos experimentos, comparou-se o desempenho das arquiteturas para o caso de falhas simples e de falhas múltiplas. A primeira etapa para a realização do estudo de casos foi obter os conjunto de treinamento e teste para cada um dos experimentos propostos. Os conjuntos de treinamento e testes são formados pelas representações PAA das respostas ao impulso dos circuitos.

Para que a etapa dos estudos casos descritos no parágrafo anterior pudesse ser realizada com segurança, o programa desenvolvido é capaz de gerar vários circuitos com valores aleatórios e os comportamentos destes circuitos gerados são avaliados para determinar se um determinado circuito possui falha ou não. Esta primeira classificação é feita com base na resposta em frequência do circuito. É gerada uma envoltória com os valores máximos e mínimos para os quais a resposta em frequência dos circuitos sem falha pode se encontrar. Para obter esta envoltória, são gerados aleatoriamente circuitos com os todos os componentes dentro da tolerância especificada. Com estes valores, é calculada a resposta em frequência para cada um dos circuitos e, dentre os resultados, são selecionados valores máximos e mínimos que a resposta dos circuitos pode alcançar para ser considerado sem falhas. Os circuitos em que a resposta em frequência se encontrar dentro da envoltória pré-definida serão considerados sem falha, enquanto que os circuitos fora da envoltória serão considerados com falha. Na Figura 13 é mostrado o gráfico de um exemplo de envoltória gerada para o filtro *Sallen-Key* utilizado neste trabalho.

Optou-se por utilizar uma envoltória para a resposta em frequência, pois não é suficiente dizer que, se o valor de um componente estiver fora da tolerância especificada, então o circuito apresenta uma falha. Muitas vezes, o circuito apresenta uma baixa sensibilidade à variação do valor de um componente em particular. Neste caso, mesmo que o valor do componente esteja fora da tolerância, isto pode não ser o suficiente para que a resposta em frequência do circuito se encontre fora da envoltória, logo, não se pode dizer que este circuito apresenta uma falha. Usando a envoltória, pode-se garantir que os circuitos gerados fora da tolerância especificada terão a variação necessária para fazer com que a resposta em frequência fique fora da envoltória e ter certeza que a baixa sensibilidade do

circuito a certos componentes não irá atrapalhar a classificação. As envoltórias mostrando circuitos dentro e circuitos fora aparecem nos gráficos (a) e (b) da Figura 14.

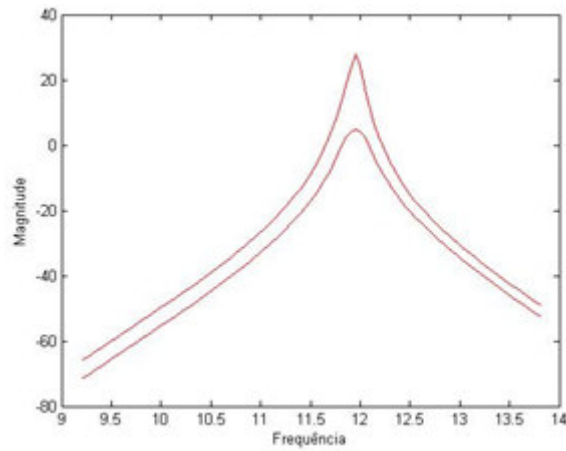


Figura 13 – Exemplo de envoltória na função de transferência do circuito *sallen-key*.

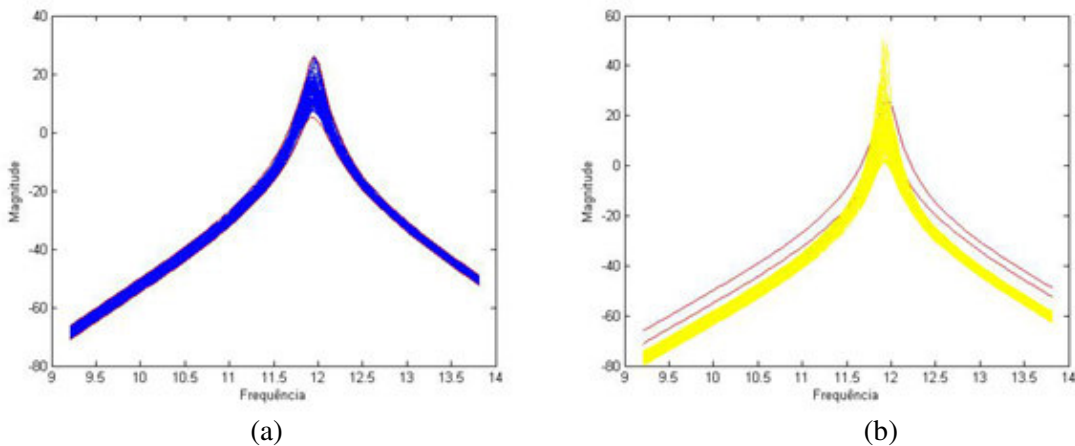


Figura 14 – Envoltórias mostrando (a) circuito dentro e (b) circuitos fora.

5.1 Sallen-Key

O circuito *Sallen-Key* conforme é mostrado na Figura 15 será usado nos dois primeiros experimentos para avaliar o desempenho das arquiteturas de classificadores. Segundo (Daryanani, 1976), é um circuito passa-banda cuja função de transferência genérica aparece na Equação (39), a função específica do circuito na Equação (40) e a resposta em frequência aparece na Figura 16. Muitos autores, como: (Albustani, 2004); (Augusto e Almeida, 2008); (Aminan e Aminan, 2001), citam esse circuito como *benchmark* para detecção de falhas.

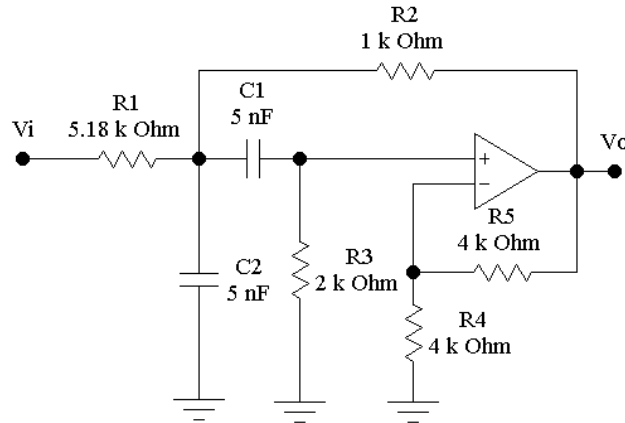


Figura 15 – Circuito Sallen-Key.

$$K = \frac{s}{s^2 + \frac{\omega_p}{Q_p} s + \omega_p^2} \quad (39)$$

$$K = \frac{7,722 \times 10^4 s}{s^2 + 3,861 \times 10^4 s + 2,386 \times 10^{10}} \quad (40)$$

5.1.1 Identificação dos componentes com falhas simples

No trabalho, foram obtidos três conjuntos para avaliar o desempenho dos classificadores. O primeiro conjunto foi o de treino que continha exemplos de circuitos sem falha e circuitos com falhas simples para os componentes indicados. Além do conjunto de treino, foram gerados dois conjuntos de teste: um com circuitos sem falha e com circuitos com falhas simples e o outro com circuitos com falha múltipla, além dos circuitos sem falha e com falha simples. Cabe destacar que estes conjuntos foram gerados de forma independente. A Tabela 1 traz o resumo das classes, relacionando a quantidade de circuitos com falhas por classe, início e término dessas classes e também relaciona o componente com falhas, por exemplo:

C1M – significa: C = capacitor, 1 = referência no circuito e M = falha paramétrica superior a 1%. Essa classe possui 300 circuitos com falhas paramétricas somente em C1 e com valores superiores a 1%.

R1m – significa: R = resistor, 1 = referência no circuito e m = falha paramétrica inferior a 1%. Essa classe possui 300 circuitos com falhas paramétricas somente em R1 e com valores inferiores a 1%.

No Fail – significa que possui 300 circuitos cujos componentes dessa classe estão dentro da tolerância, ou seja, não possui qualquer tipo de falha.

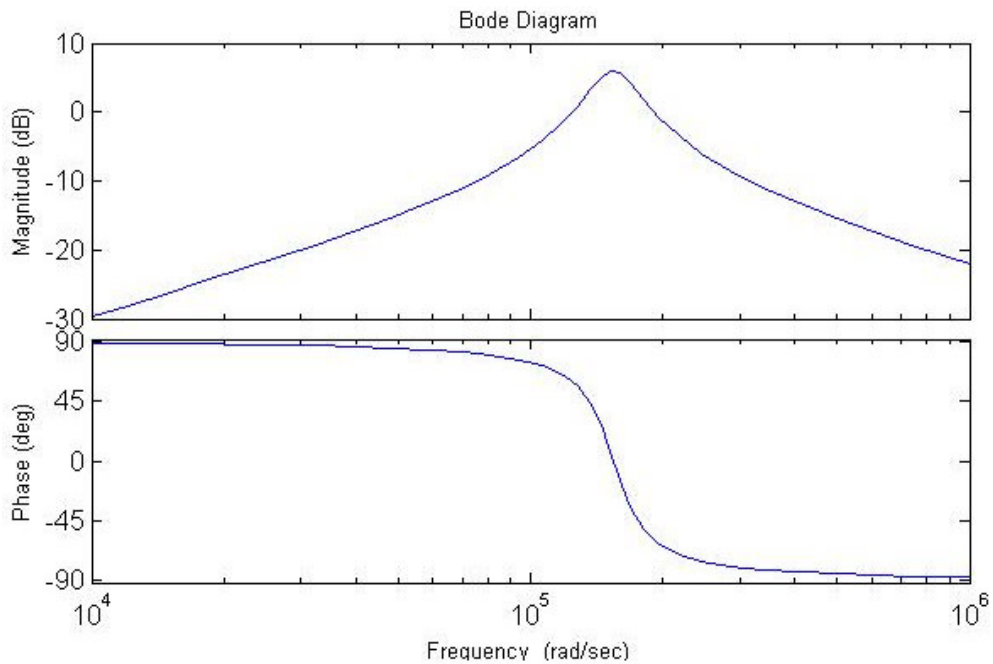


Figura 16 – resposta em frequência do circuito *sallen-key*.

A exceção para falha simples na tabela é classe 1036, pois são inseridos falhas paramétricas em dois componentes simultâneos, R1m e C1M representando circuitos com múltiplas falhas. Essa classe será usada posteriormente no item de classificação com múltiplas falhas.

5.1.1.1 Classificador multiclasse.

Para avaliar o comitê de classificadores de classe única, os conjuntos de dados passarão pelo classificador multiclasse para que os dois tipos classificadores sejam comparados e discutidos destacando suas diferenças e qualidades. O classificador multiclasse proposto é o k -vizinhos mais próximos. Para que o classificador seja eficiente é preciso determinar o valor do parâmetro que define a quantidade de vizinhos mais próximos que será utilizada.

Tabela 1 – Classes relacionados às falhas de tolerância para o circuito *sallen-key*.

Classe	Início	Fim	componente
1	1	300	C1M
2	301	600	C1m
3	601	900	C2M
4	901	1200	C2m
5	1201	1500	No Fail
6	1501	1800	R3M
7	1801	2100	R3m
8	2101	2400	R2M
9	2401	2700	R2m
10	2701	3000	R1M
11	3001	3300	R1m
1036	3301	3600	R1m e C1M

Fonte: Resultados do estudo de caso

Isto pode ser feito com auxílio da validação cruzada (Witten e Frank, 2005). O procedimento consiste em dividir conjunto de dados em certo número de partes, normalmente em 5 ou 10. Para cada uma destas partes, obtém-se o erro do classificador que foi treinado com as partes restantes. A média dos erros de classificação obtidos para cada uma das partes fornece uma estimativa do erro de generalização do classificador, e, portanto, do seu desempenho. A determinação do número de vizinhos a ser utilizado pode ser feita experimentando-se diferentes valores para o número de vizinhos mais próximos e avaliando para qual valor se obtém o melhor desempenho.

Tabela 2 – erro de validação cruzada do circuito *sallen-key*, (*knn*) por $n^{\circ} knn$.

$n^{\circ} knn$	erro
1	0
3	0
5	0
7	0,0003
9	0,0006
11	0,0009
13	0,0006
15	0,0006
17	0,0006

Fonte: Resultado do experimento

Os resultados da validação cruzada para o conjunto de treino do circuito *sallen-key*, aparecem na Tabela 2, como o parâmetro avaliado é o erro, quanto menor o erro, melhor é o parâmetro. Portanto o *knn* escolhido foi o $k-nn = 1$.

5.1.1.2 Resultados do classificador multi-classe k-nn para o circuito SK com falhas simples.

Na Tabela 3 aparecem os resultados do classificador multiclasse k -nn do circuito *sallen-key* com 10 classes de falhas simples e 1 classe sem falhas conforme mostrado na Tabela 1. A matriz de confusão pode ser vista no Anexo, Figura 22.

Tabela 3 – Resultado da k _nn para o *Sallen-Key* com falhas simples.

Classe Verdadeira	Classe Estimada	Número de Classificações Indesejadas	Erros (%)
1	1	0	0
2	2	0	0
3	3	0	0
4	4	0	0
5	5	0	0
6	6	0	0
7	7	0	0
8	8	0	0
9	9	0	0
10	10	0	0
11	11	0	0
Erro Total	—	0	0
Sugestões Erradas	—	N/A	N/A
Falha Desconhecida	—	0	0

Fonte: Resultado do experimento

Para todos os 3300 circuitos *sallen-key* testados pelo classificador multiclasse k -nn, não houve qualquer erro de classificação, o classificador acertou 100% dos defeitos simples e também todos os circuitos sem falhas.

5.1.1.3 Comitê de classificadores de Classe única.

O comitê de classificadores classe única escolhidos para serem comparados ao classificador multi-classe são: o k -vizinhos mais próximos *data description* ou k_nndd , o $gauss_dd$ e o $kmeans_dd$. A escolha dos parâmetros é uma atividade importante para um bom desempenho dos classificadores de classe única, portanto novamente, a validação cruzada será utilizada, entretanto em vez do erro de classificação, a métrica $F1$ é a escolhida para determinação do número de vizinhos através da validação cruzada. $F1$ relaciona os parâmetros *precision* e *recall* e esses parâmetros são métricas relacionadas ao acerto, portanto quanto maior $F1$ melhor o desempenho dos classificadores escolhidos. É característico do classificador de classe única ter uma fração de rejeição (*fracrej*) as taxas de rejeição utilizadas na validação cruzada dos classificadores serão: 1%, 2%, 5% e 10%.

No comitê classificador classe única knn_dd , da mesma forma que no classificador multiclasse, também é necessário que o número de vizinhos seja especificado na validação cruzada para encontrar os melhores parâmetros e serão usados os seguintes vizinhos mais próximos: 1, 3, 5, 7, 9, 11, 13, 15, 17 e 19.

Na validação cruzada para o circuito *Sallen-key*, o resultado com maior valor de $F1$ para o knn_dd foi: $knn = 1$ e a fração de rejeição = 0,01 ou 1%. A variação de $F1$ em função da fração de rejeição aparece na Tabela 4 e também no gráfico da Figura 17.

No comitê classificador classe única $gauss_dd$, os parâmetros a serem definidos são: fração de rejeição e o valor de r . o valor de r pode ser dado para acrescentar alguma regularização da matriz covariância estimada. Para validação cruzada foram estimados os seguintes valores de r : 0,01; 0,02; 0,04; 0,06; 0,08; 0,1; 0,12; 0,14; 0,16; 0,18; 0,20; 0,22; 0,24; 0,26; 0,28.

Na validação cruzada para o circuito *Sallen-key*, o resultado com maior valor de $F1$ para o $gauss_dd$ foi: a fração de rejeição = 0,01 ou 1% e o valor de $r = 0,01$. A variação de $F1$ em função da fração de rejeição aparece na Tabela 4 e também no gráfico da Figura 17.

No comitê classificador classe única $kmeans_dd$ o parâmetro a ser definido além da fração de rejeição é a quantidade de *clusters*. Na validação cruzada o número de clusters será de 1 a 15.

Na validação cruzada para o circuito *Sallen-key*, o resultado com maior valor de $F1$ para o $kmeans_dd$ foi: a fração de rejeição = 0,01 ou 1% e número de *clusters* = 13. A

variação de $F1$ em função da fração de rejeição aparece na Tabela 4 e também no gráfico da Figura 17.

5.1.1.4 Resultados do comitê de classificadores de classe única com falhas simples.

Na Tabela 5 aparece o resultado do comitê de classificadores de classe única k -nndd, $gauss_dd$ e $kmeans_dd$ do circuito *sallen-key*, os conjuntos de dados usados para treino e teste são os mesmos que para o multi-classe, sendo assim, as classes escolhidas são as mesmas e estão dispostas na Tabela 1. O conteúdo da matriz de confusão aparece no Anexo, Figura 23, Figura 24 e Figura 25.

Tabela 4 – valores de $F1$ em função da fração de rejeição (%) para knn_dd , $gauss_dd$ e $kmeans_dd$.

<i>fracrej (%)</i>	$F1$ (knn_dd)	$F1$ ($gauss_dd$)	$F1$ ($kmeans_dd$)
1	0,9930	0,9901	0,9821
2	0,9902	0,9857	0,9773
5	0,9751	0,9702	0,9581
10	0,9444	0,9421	0,9191

Fonte: Resultado do experimento

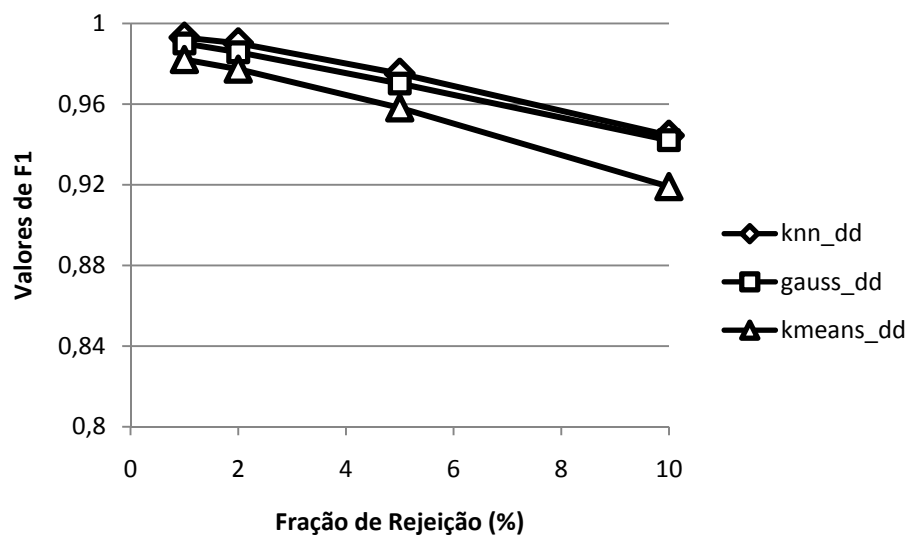


Figura 17 – gráfico dos valores de $F1$ em função da fração de rejeição (%) *sallen-key*.

Observando a Tabela 5, para todos os 3300 circuitos *sallen-key* testados pelo comitê de classificadores de classe única o que teve o melhor rendimento, ou seja, o menor erro total foi o *k-nn*, o classificador errou apenas 0,79%, ou seja, 26 erros de 3300 circuitos, porém classificou-os como erro desconhecido. Isso quer dizer que o classificador na dúvida durante a classificação ele não colocou o circuito em outra classe e sim como categoria desconhecida, portanto em caso de falhas simples, o comitê de classe única tem resultados competitivos com o classificador multiclasse, pois a porcentagem de erro foi muito baixa. A matriz de confusão está no Anexo, Figura 23.

Tabela 5 – Resultado do comitê de classificadores de classe única para SK falhas simples.

Classe Verdadeira	Classe Estimada	<i>kmm_dd</i>		<i>gauss_dd</i>		<i>kmeans_dd</i>	
		Número de Classificações Indesejadas	Erros (%)	Número de Classificações Indesejadas	Erros (%)	Número de Classificações Indesejadas	Erros (%)
1	1	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	8	2,67	4	1,33
2	2	0	0,00	0	0,00	0	0,00
	Desconhecida	6	2,00	4	1,33	9	3,00
3	3	0	0,00	0	0,00	0	0,00
	Desconhecida	4	1,33	3	1,00	8	2,67
4	4	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	3	1,00	5	1,67
5	5	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	1	0,33	5	1,67
6	6	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	4	1,33	2	0,67
7	7	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	5	1,67	3	1,00
8	8	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	5	1,67	6	2,00
9	9	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	2	0,67	3	1,00
10	10	0	0,00	0	0,00	0	0,00
	Desconhecida	5	1,67	3	1,00	10	3,33
11	11	0	0,00	0	0,00	0	0,00
	Desconhecida	3	1,00	10	3,33	13	4,33
143	C2m ou R2m	0	0,00	38	12,67	193	64,33
Erro Total	—	26	0,79	86	2,61	261	7,91
Sugestões Erradas	—	0	0,00	38	1,15	193	5,85
Falha Desconhecida	—	26	0,79	52	1,58	68	2,06

Fonte: Resultado do experimento

5.1.2 Múltiplas falhas

A fim de verificar o desempenho do comitê em comparação ao desempenho do classificador multiclasse foi criado outro conjunto de dados para o circuito *sallen-key* que além dos 3.000 circuitos com falhas simples e 300 circuitos sem falhas, foi incluído 300 circuitos com falhas múltiplas. Os circuitos foram gerados de forma que os valores da tolerância dos componentes $R1$ e $C1$ fossem alterados simultaneamente. O componente $R1$ com valores abaixo da tolerância cuja falha pertence à classe 7 e o $C1$ com valores acima da tolerância e essa falha pertence à classe 1.

5.1.2.1 Resultados classificador multiclasse k-nn para o Sallen-Key com múltiplas falhas.

A Tabela 6 mostra o resultado da classificação multiclasse k -nn para o circuito *sallen-key* com falhas múltiplas. O classificador para classificar o conjunto de teste com múltiplas falhas foi treinado com um conjunto de treino que possuía exemplos apenas de falhas simples, possuindo 11 classes, como aparece na Tabela 1, portanto qualquer circuito classificado vai pertencer a uma das onze classes. A matriz de confusão está no Anexo, Figura 26.

Como pode ser visto na Tabela 6 todos os circuitos da classe 1036 que é a classe que contém as múltiplas falhas, 60,33% são classificadas como classe 3 e 39,67% são classificadas como classe 8. É comentado no parágrafo anterior que as classes cujos componentes apresentam as falhas individualmente são as classes 1 e 7. Portanto, o classificador multiclasse errou a classificação de todos os circuitos múltiplas falhas.

5.1.2.2 Comitê de classificadores de classe única para o circuito SK para múltiplas falhas.

Da mesma forma que o classificador $knndd$ (k -vizinhos mais próximos) foi usado para classificar os circuitos do *Sallen-key* com falha simples, foi usado também para classificar os circuitos com múltiplas falhas. O conjunto de dados utilizado é o mesmo do classificador multiclasse, portanto as classes e os valores de tolerância é o da Tabela 1.

Tabela 6 – Resultado da classificação k _nn para o *Sallen-Key* com falhas múltiplas.

Classe Verdadeira	Classe Estimada	Número de Classificações Indesejadas	Erros (%)
1	1	0	0
2	2	0	0
3	3	0	0
4	4	0	0
5	5	0	0
6	6	0	0
7	7	0	0
8	8	0	0
9	9	0	0
10	10	0	0
11	11	0	0
1036	3	181	60,33
	8	119	39,67
Erro Total	—	300	10
Sugestões Erradas	—	N/A	N/A
Falha Desconhecida	—	N/A	N/A

Fonte: Resultado do experimento

Para o comitê de classe única k -nndd será usado os mesmos parâmetros que foram usados para avaliação do k -nndd falha simples no item 5.1.1.3, pois o conjunto de dados treino é o mesmo. Sendo assim, a fração de rejeição ou $fracrej = 1\%$ e vizinhos mais próximos ou $knndd = 1$.

5.1.2.3 Resultados do comitê classe única knn_dd do circuito SK para múltiplas falhas.

A Tabela 7 trás os resultados do classificador classe única para o conjunto de dados dos circuitos com múltiplas falhas. Do comitê de classificadores de classe única o que apresentou melhor desempenho de classificação levando em consideração apenas o erro total foi o classificador *knn_dd*. Nesse aspecto, apesar do comitê de classificadores de classe única ter a fração de rejeição que classifica erroneamente alguns circuitos da classe verdadeira em classe desconhecida, ele não os coloca erroneamente como se fosse falha em outro componente como faz o classificador multiclasse.

Tabela 7 – Resultado do comitê de classificadores de classe única para SK, falhas múltiplas.

Classe Verdadeira	Classe Estimada	knn_dd		Gauss_dd		kmeans_dd	
		Número de Classificações Indesejadas	Erros (%)	Número de Classificações Indesejadas	Erros (%)	Número de Classificações Indesejadas	Erros (%)
1	1	0	0,00	0	0,00	0	0,00
	Desconhecida	7	2,33	3	1,00	3	1,00
2	2	0	0,00	0	0,00	0	0,00
	Desconhecida	3	1,00	2	0,67	10	3,33
3	3	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	0	0,00	6	2,00
4	4	0	0,00	0	0,00	0	0,00
	Desconhecida	4	1,33	3	1,00	5	1,67
5	5	0	0,00	0	0,00	0	0,00
	Desconhecida	1	0,33	1	0,33	1	0,33
6	6	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	2	0,67	0	0,00
7	7	0	0,00	0	0,00	0	0,00
	Desconhecida	4	1,33	8	2,67	6	2,00
8	8	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	8	2,67	6	2,00
9	4	0	0,00	0	0,00	7	2,33
	9	0	0,00	0	0,00	0	0,00
	143 R1m ou C1m	0	0	41	13,67	95	31,67
	Desconhecida	0	0,00	3	1,00	0	0,00
10	10	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	1	0,33	3	1,00
11	11	0	0,00	0	0,00	0	0,00
	Desconhecida	6	2,00	7	2,33	4	1,33
1036	Desconhecida	0	0,00	0	0,00	0	0,00
Erro Total	—	31	0,86	79	2,19	146	4,06
Sugestões Erradas	—	0	0	41	1,14	95	2,64
Falhas Desconhecidas	—	31	0,86	38	1,06	44	1,22

Fonte: Resultado do experimento

5.2 Filtro universal

O segundo circuito a ser usado nessa dissertação para avaliar o desempenho dos classificadores é o filtro universal (*ctsv continuous-time state-variable*) conforme mostrado na Figura 18.

5.2.1 Descrição do circuito

Segundo (Kaminska et. al., 1997) o filtro universal é um circuito que incorpora os filtros passa baixas, passa banda e passa altas. A função de transferência é de um filtro de segunda ordem considerando filtro passa banda na saída (BPO) é dado pela Equação (41) e a resposta em frequência do circuito é mostrada na Figura 19.

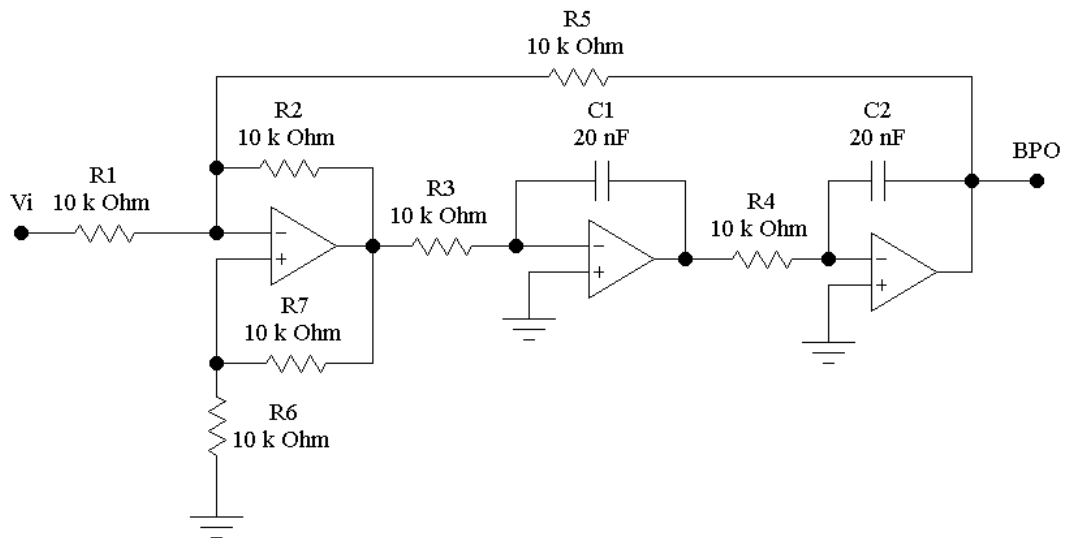


Figura 18 – circuito *benchmark ctsv (continuous-time state-variable)*

$$V_{BPO} = K \frac{\frac{s}{R_3 C_1}}{s^2 + \frac{1}{Q R_3 C_1} s + \frac{1}{R_3 C_1 R_4 C_2}} \quad (41)$$

onde K representa o ganho e Q é o fator de qualidade do filtro conforme mostra as Equações (42) e (43). A frequência central do BPO é dada pela Equação (44) e a função de transferência aparece na Equação (45).

$$K = \frac{R_2}{R_1} \approx \frac{R_5}{R_1} \quad (R_2 = R_5) \quad (42)$$

$$Q = \frac{R_6 + R_7}{(1 + 2K)R_6} \quad (43)$$

$$f_c = \frac{1}{2\pi\sqrt{R_3C_1R_4C_2}} \quad (44)$$

$$V_{BPO} = \frac{5000s}{s^2 + 4500s + 2,5 \times 10^7} \quad (45)$$

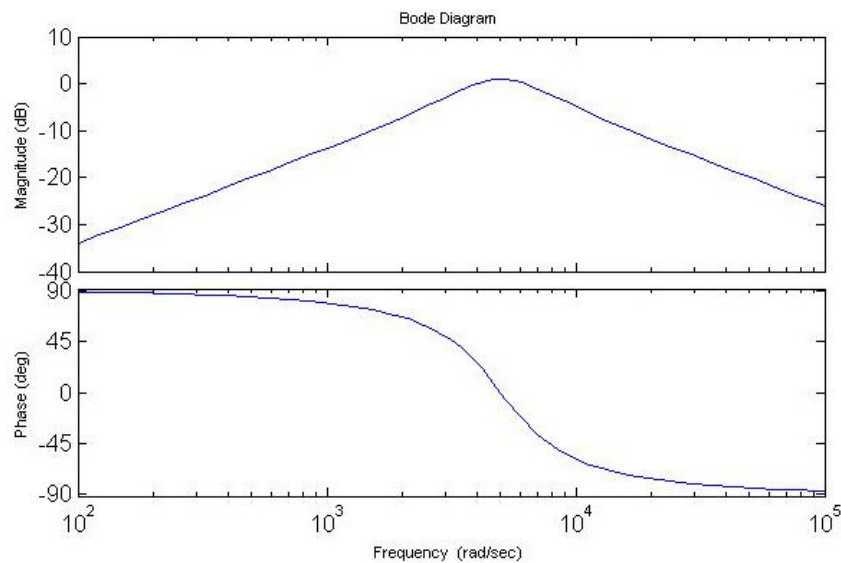


Figura 19 – resposta em frequência do circuito ctsv.

5.2.2 Identificação dos componentes normais com falhas simples e falha múltipla

A Tabela 8 mostra as 20 classes do *filtro universal* que estão representadas da seguinte forma:

- a classe 1 possui 300 circuitos com falhas no componente C1 cuja tolerância está acima de 1% e assim por diante;
- Quando o componente apresentar a letra M significa que a variação da tolerância está acima da nominal e se apresentar a letra m, significa que a tolerância está abaixo da nominal;
- A classe 5 é formada por componentes dentro da tolerância, ou seja, sem falhas;

- A classe 266259 é a codificação que representa os componentes com falhas múltiplas contendo valores de $R1$ abaixo da tolerância nominal e valores de $C1$ acima da tolerância nominal.

Para o circuito *ctsv* assim como o circuito *sallen-key* são montados três conjuntos: 1 para treino com 19 classes de falhas simples, nesse caso, a classe 266259 que é a classe de circuitos com falhas múltiplas é excluída, um conjunto de dados para teste contendo igualmente 19 classes com falhas simples para avaliar o desempenho do classificador e um conjunto de dados com 20 classes, esse último além dos 19 conjuntos de falhas simples.

A exceção para falha simples na tabela é classe 266259, pois são inseridos falhas paramétricas em dois componentes simultâneos, $R1m$ e $C1M$ representando circuitos com múltiplas falhas. Essa classe será usada posteriormente para classificação com múltiplas falhas.

Tabela 8 – classe do circuito, início, fim e componente com falha

Classe	Início	Fim	componente	Classe	Início	Fim	componente
1	1	300	C1M	11	3001	3300	R3m
2	301	600	C1m	12	3301	3600	R4M
3	601	900	C2M	13	3601	3900	R4m
4	901	1200	C2m	14	3901	4200	R5M
5	1201	1500	No Fail	15	4201	4500	R5m
6	1501	1800	R1M	16	4501	4800	R6M
7	1801	2100	R1m	17	4801	5100	R6m
8	2101	2400	R2M	18	5101	5400	R7M
9	2401	2700	R2m	19	5401	5700	R7m
10	2701	3000	R3M	266259	5701	6000	R1m e C1M

Resultado do estudo de caso

5.2.2.1 Classificador multiclasse.

Para o circuito *ctsv* será usado o classificador multiclasse *knn* que será avaliado da mesma forma como foi avaliado no circuito *sallen-key* conforme comentado no item 5.1.1.1. Para tanto, a validação cruzada será utilizada a fim de descobrir o número de vizinhos mais próximos que dará o melhor desempenho ao classificador.

Os resultados obtidos aparecem na Tabela 9, como o parâmetro avaliado é o erro, quanto menor o erro, melhor é o parâmetro. Portanto o número de vizinhos mais próximos escolhido foi o $k_{mm} = 17$.

Tabela 9 – erro de validação cruzada *ctsv* para o classificador *knn* pelo n° *knn*

<i>knn</i>	erro
1	0.2711
3	0.2732
5	0.2747
7	0.2625
9	0.2630
11	0.2577
13	0.2570
15	0.2502
17	0.2484
19	0.2582
21	0.2589
23	0.2526

Fonte: Resultado do experimento

5.2.2.2 Resultados do classificador multiclasse k-nn para o circuito *ctsv*.

Na Tabela 10 aparecem os resultados do classificador multiclasse *k-nn* do circuito *ctsv* com 19 classes de falhas simples e 1 classe sem falhas que podem ser conferidas na Tabela 8. A matriz de confusão está no Anexo, Figura 30.

Pelos resultados obtidos na Tabela 10, pode-se observar que existe uma sobreposição de classes no circuito *ctsv*, o que faz com que o classificador multiclasse tenha muitos erros de classificação. O total de erro é consideravelmente grande, ou seja, dos 5.700 circuitos classificados 1441 foram classificados em classe errada e representa 25,28 % do valor total dos circuitos.

5.2.2.3 Comitê de classificadores de Classe única

O comitê de classificadores de classe única escolhido para classificar os circuitos *ctsv* e comparados ao classificador multi-classe, igualmente ao *Sallen-Key* são: ou *k_nn*, o *gauss_dd* e o *kmeans_dd* e como foi feito para o circuito *Sallen-Key*, descrito no item 5.1.1.3, a validação cruzada será usada para encontrar os melhores parâmetros para classificar o *ctsv*. Para todos os classificadores serão usadas as taxas de rejeição de: 1%, 2%, 5% e 10%.

Tabela 10 – Resultado da classificação *k_nn* para o *ctsv* com falhas simples.

Classe Verdadeira	Classe Estimada	Número de Classificações indesejadas	Erros (%)
1	1	0	0,00
	10	134	44,67
2	2	0	0,00
	11	138	46,00
3	3	0	0,00
	12	162	54,00
4	4	0	0,00
	13	136	45,33
5	5	0	0,00
6	6	0	0,00
7	7	0	0,00
	16	82	27,33
	19	129	43,00
8	8	0	0,00
9	9	0	0,00
10	1	168	56,00
	10	0	0,00
11	2	138	46,00
	11	0	0,00
12	3	138	46,00
	12	0	0,00
13	4	167	55,67
	13	0	0,00
14	14	0	0,00
15	15	0	0,00
16	7	4	1,33
	16	0	0,00
17	17	0	0,00
18	18	0	0,00
19	7	45	15,00
	19	0	0,00
Erro total	—	1441	25,28
Sugestões Erradas	—	N/A	N/A
Falha Desconhecida	—	N/A	N/A

Fonte: Resultado do experimento

No comitê classificador classe única *knn_dd* serão usados o número de vizinhos mais próximos iguais a: 1, 3, 5, 7, 9, 11, 13, 15, 17 e 19.

Na validação cruzada para o circuito *ctsv* o resultado com maior valor de *F1*: *knn* = 1 e a fração de rejeição = 0,01 ou 1%. A variação de *F1* em função da fração de rejeição aparece na Tabela 11 e também no gráfico da Figura 20.

No comitê de classificadores de classe única *gauss_dd* para o *ctsv*, os parâmetros a serem definidos são: fração de rejeição e o valor de *r*. o valor de *r* conforme comentado em 5.1.1.3, pode ser dado para acrescentar alguma regularização da matriz covariância estimada. Para validação cruzada foram estimados os seguintes valores de *r*: 0,01; 0,02; 0,04; 0,06; 0,08; 0,1; 0,12; 0,14; 0,16; 0,18; 0,20; 0,22; 0,24; 0,26; 0,28.

Na validação cruzada para o circuito *ctsv*, o resultado com maior valor de *F1* para o *gauss_dd* foi: fração de rejeição = 0,01 e *r* = 0,02. Os valores do parâmetro *F1* em função da fração de rejeição aparecem na Tabela 11 e também no gráfico da Figura 20.

No comitê classificador classe única *kmeans_dd* para o circuito *ctsv* o parâmetro a ser definido além da fração de rejeição é a quantidade de *clusters*. Na validação cruzada o número de *clusters* será de 1 a 15.

Na validação cruzada para o circuito *ctsv*, o resultado com maior valor de *F1* para o *kmeans_dd* foi: a fração de rejeição = 0,01 ou 1%. e número de *clusters* = 14. Os valores do parâmetro *F1* em função da fração de rejeição aparecem na Tabela 11 e também no gráfico da Figura 20.

Tabela 11 – valores de *F1* em função da fração de rejeição (%).

<i>fracrej</i> (%)	<i>F1 knn_dd</i>	<i>F1 kmeans_dd</i>	<i>F1 gauss_dd</i>
1	0,9206	0,9291	0,9326
2	0,9161	0,9209	0,9283
5	0,9008	0,9011	0,9123
10	0,8752	0,8662	0,8866

5.2.2.4 Resultados

Na Tabela 12 aparece o resultado do comitê de classificadores de classe única *knn_dd*, *gauss_dd*, *kmean_dd* do circuito *ctsv*, os conjuntos de dados usados para treino e teste são os

mesmos que para o multi-classe, sendo assim, as classes são as mesmas e estão dispostas na Tabela 8. As matrizes de confusão estão no Anexo, Figura 31, Figura 32 e Figura 33.

Na avaliação dos resultados apresentados na Tabela 12 e na observação do *erro total* por comitê de classificador de classe única, o comitê classificador que teve o melhor desempenho foi *knn_dd*.

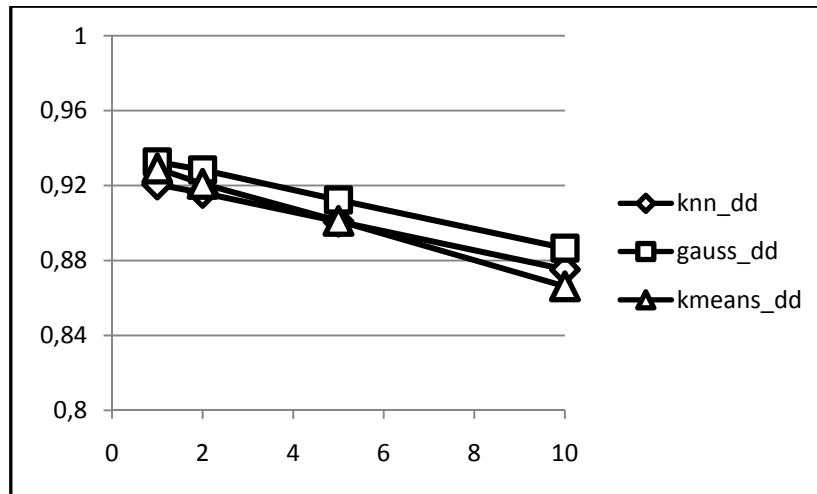


Figura 20 – gráfico dos valores de *F1* em função da fração de rejeição (%) *ctsv*.

Na comparação entre o comitê classificador classe única *knn_dd* e o multiclasse *knn*, enquanto a classificação multiclasse para o circuito *ctsv* apresenta uma porcentagem muito elevada de erros (25,28%) para o comitê de classificadores de classe única a porcentagem de erro é muito pequena, apenas 0,82% dos 5700 circuitos foram erroneamente classificados. Por exemplo, na classe 7 que possui 300 circuitos, o comitê acertou 49 circuitos, sugeriu que 86 circuitos poderiam ser da classe 7 ou classe 16 e também sugeriu para 164 circuitos da classe 7 que estes poderiam ser da classe 7 ou classe 19, Portanto o classificador de classe única não classifica errado, ao contrário, sugere uma possível classe para o circuito. O classificador multiclasse classificou erroneamente 69,33% dos circuitos da classe 7.

Tabela 12 – Resultado dos comitês classificadores classe única *ctsv* falhas simples.

Classe Verdadeira	Classe Estimada	knn_dd		gauss_dd		kmeans_dd	
		Número de Classificações indesejadas	Erros (%)	Número de Classificações indesejadas	Erros (%)	Número de Classificações indesejadas	Erros (%)
1	10	1	0,33	0	0,00	1	0,33
	1 ou 10	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	1	0,33	1	0,33
2	11	1	0,33	2	0,67	1	0,33
	2 ou 11	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	2	0,67	2	0,67
3	12	0	0,00	1	0,33	0	0,00
	3 ou 12	0	0,00	0	0,00	0	0,00
	Desconhecida	1	0,33	1	0,33	0	0,00
4	13	3	1,00	0	0,00	3	1,00
	4 ou 13	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	0	0,00	0	0,00
5	Desconhecida	3	1,00	3	1,00	3	1,00
6	Desconhecida	1	0,33	2	0,67	4	1,33
7	7 ou 16	0	0,00	0	0,00	0	0,00
	7 ou 19	0	0,00	0	0,00	0	0,00
	16	1	0,33	0	0,00	1	0,33
	19	0	0,00	0	0,00	1	0,33
8	Desconhecida	3	1,00	6	2,00	5	1,67
9	Desconhecida	5	1,67	2	0,67	3	1,00
10	1	0	0,00	0	0,00	1	0,33
	1 ou 10	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	3	1,00	2	0,67
11	2	8	2,67	0	0,00	0	0,00
	2 ou 11	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	3	1,00	3	1,00
12	3	2	0,67	2	0,67	0	0,00
	3 ou 12	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	1	0,33	1	0,33
13	4	6	2,00	1	0,33	2	0,67
	4 ou 13	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	2	0,67	0	0,00
14	Desconhecida	1	0,33	1	0,33	1	0,33
15	Desconhecida	2	0,67	3	1,00	8	2,67
16	7	0	0,00	1	0,33	3	1,00
	7 ou 16	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	2	0,67	3	1,00
17	Desconhecida	0	0,00	3	1,00	3	1,00
18	17 ou 18	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	1	0,33	1	0,33
19	7	4	1,33	4	1,33	7	2,33
	7 ou 19	0	0,00	0	0,00	0	0,00
	Desconhecida	1	0,33	2	0,67	2	0,67
Erro total	—	47	0,82	49	0,86	60	1,05
Sugestões Erradas	—	0	0,00	0	0,00	0	0,00
Falha Desconhecida	—	21	0,37	38	0,67	42	0,74

Fonte: Resultado do experimento

5.2.3 Múltiplas falhas

Para dar continuidade à avaliação do desempenho do classificador de classe única comparando-o com o desempenho do classificador multiclasse foi criado outro conjunto de dados para o circuito *ctsv* que além dos 5400 circuitos com falhas simples e 300 circuitos sem falhas, foi incluído 300 circuitos com falhas múltiplas. Os circuitos foram gerados de forma que os valores da tolerância dos componentes *R1* e *C1* fossem alterados simultaneamente. O componente *R1* com valores abaixo da tolerância cuja falha pertence à classe 7 e o *C1* com valores acima da tolerância e essa falha pertence à classe 1.

5.2.3.1 Resultados do classificador multi-classe *k*-nn para o circuito *ctsv* com múltiplas falhas.

A Tabela 13 mostra o resultado da classificação multiclasse *k*-nn para o circuito *ctsv* com falhas múltiplas. O classificador para classificar o conjunto de teste com múltiplas falhas foi treinado com o conjunto de dados treino que possui 19 classes como aparece na Tabela 8, portanto qualquer circuito classificado vai pertencer a uma das 19 classes. A matriz de confusão está no Anexo, Figura 34.

A Tabela 13 mostra que das 20 classes, (inclusive a classe com falhas múltiplas número 266259), expostas ao classificador multiclasse, 12 classes receberam circuitos classificados erroneamente. Conforme comentado no parágrafo anterior, os circuitos com múltiplas falhas foram classificados erroneamente nas classes 3, 12 e 14. No total 30,43 % dos 6.000 circuitos classificados foram errados, ou seja, quando há sobreposição de circuitos o classificador multiclasse não consegue distingui-los e classifica esses circuitos em outra classe e por outro lado, quando há um elemento novo (no caso a classe com múltiplas falhas) o classificador sempre vai classificar esse novo elemento em uma das classes as quais o classificador multiclasse foi treinado.

5.2.3.2 Comitê de classificadores de classe única para o circuito *ctsv* para múltiplas falhas.

Os comitês classificadores de classe única: *knn_dd*, *gauss_dd* e *kmeans_dd*, para classificar o conjunto de dados com falhas múltiplas para o circuito *ctsv* terá igualmente os mesmos parâmetros do classificador de falhas simples, pois o classificador sempre será treinado com o conjunto de dados treino conforme item 5.1.1.3. Sendo assim, os parâmetros a serem usados para classificar os circuitos com múltiplas falhas serão os mesmos usados para os que classificaram os circuitos com falhas simples.

Tabela 13 – Resultado da classificação k_nn para o *ctsv* com falhas múltiplas.

Classe Verdadeira	Classe Estimada	Número de Classificações indesejadas	Erros (%)
1	1	0	0,00
	10	149	49,67
2	2	0	0,00
	11	145	48,33
3	3	0	0,00
	12	155	51,67
4	4	0	0,00
	13	145	48,33
5	5	0	0,00
6	6	0	0,00
7	7	0	0,00
	16	67	18,33
	19	107	33,00
8	8	0	0,00
9	9	0	0,00
10	1	146	48,67
	10	0	0,00
11	2	161	53,67
	11	0	0,00
12	3	142	47,33
	12	0	0,00
13	4	155	51,67
	13	0	0,00
14	14	0	0,00
15	15	0	0,00
16	7	55	18,33
	16	0	0,00
17	17	0	0,00
18	18	0	0,00
19	7	99	33,00
	19	0	0,00
266259	3	145	48,33
	12	33	11,00
	14	122	40,67
Erro total	—	1826	30,43
Sugestões Erradas	—	N/A	—
Falha Desconhecida	—	N/A	—

Fonte: Resultado do experimento

5.2.3.3 Resultados do comitê classe única knn_dd do circuito ctsv para circuitos com múltiplas falhas.

A Tabela 14 traz os resultados do classificador classe única para o conjunto de dados dos circuitos com múltiplas falhas. Dos classificadores avaliados, o que teve o maior desempenho foi o knn_dd porque o erro total dele foi o menor.

Para esse classificador, devido à fração de rejeição, alguns falsos negativos são classificados como classe desconhecida. Os circuitos classificados como classe desconhecida e que pertencem a uma classe ocorrem em uma porcentagem muito baixa. Classificações erradas também aparecem muito pouco. Por exemplo, pode-se observar que dos 300 circuitos da classe 1 somente dois circuitos foram erroneamente classificados, um outro exemplo é a classe 7 o classificador classificou 42 circuitos corretamente, sugere que 87 desses circuitos podem ser da classe 7 ou da classe 16 e sugere também que 171 circuitos poderia pertencer às classes 7 ou 19.

Tabela 14 – Resultado dos comitês classificadores classe única para ctsv falhas múltiplas.

Classe Verdadeira	Classe Estimada	knn_dd		gauss_dd		kmeans_dd	
		Número de Classificações indesejadas	Erros (%)	Número de Classificações indesejadas	Erros (%)	Número de Classificações indesejadas	Erros (%)
1	10	2	0,67	1	0,33	7	2,33
	1 ou 10	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	1	0,33	1	0,33
2	2	0	0,00	0	0,00	0	0,00
	11	1	0,33	5	1,67	4	1,33
	2 ou 11	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	0	0,00	0	0,00
3	3	0	0,00	0	0,00	0	0,00
	12	0	0,00	0	0,00	3	1,00
	3 ou 12	0	0,00	0	0,00	0	0,00
	Desconhecida	1	0,33	3	1,00	0	0,00
4	4	0	0,00	0	0,00	0	0,00
	13	2	0,67	0	0,00	0	0,00
	4 ou 13	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	2	0,67	0	0,00
5	5	0	0,00	0	0,00	0	0,00
	Desconhecida	4	1,33	5	1,67	9	3,00
6	6	0	0,00	0	0,00	0	0,00
	Desconhecida	2	0,67	3	1,00	8	2,67
7	7	0	0,00	0	0,00	0	0,00
	7 ou 16	0	0,00	0	0,00	0	0,00
	7 ou 19	0	0,00	0	0,00	0	0,00
	19	0	0,00	1	0,33	2	0,67

Tabela 14 – Resultado dos comitês classificadores classe única para *ctsv* falhas múltiplas (continuação).

Classe Verdadeira	Classe Estimada	knn_dd		gauss_dd		kmeans_dd	
		Número de Classificações indesejadas	Erros (%)	Número de Classificações indesejadas	Erros (%)	Número de Classificações indesejadas	Erros (%)
8	8	0	0,00	0	0,00	0	0,00
	Desconhecida	5	1,67	5	1,67	8	2,67
9	9	0	0,00	0	0,00	0	0,00
	Desconhecida	4	1,33	2	0,67	8	2,67
10	10	0	0,00	0	0,00	0	0,00
	1 ou 10	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	0	0,00	4	1,33
11	2	3	1,00	0	0,00	5	1,67
	11	0	0,00	0	0,00	0	0,00
	2 ou 11	0	0,00	0	0,00	0	0,00
	Desconhecido	0	0,00	2	0,67	0	0,00
12	3	2	0,67	2	0,67	4	1,33
	3 ou 12	0	0,00	0	0,00	0	0,00
	Desconhecida	1	0,33	5	1,67	0	0,00
13	13	0	0,00	0	0,00	0	0,00
	4	5	1,67	4	1,33	5	1,67
	4 ou 13	0	0,00	0	0,00	0	0,00
14	14	0	0,00	0	0,00	0	0,00
	Desconhecida	4	1,33	6	2,00	7	2,33
15	15	0	0,00	0	0,00	0	0,00
	Desconhecida	3	1,00	6	2,00	7	2,33
16	16	0	0,00	0	0,00	0	0,00
	7	3	1,00	3	1,00	8	2,67
	7 ou 16	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	2	0,67	0	0,00
17	17	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	6	2,00	6	2,00
18	17	0	0,00	0	0,00	1	0,33
	18	0	0,00	0	0,00	0	0,00
	17 ou 18	0	0,00	0	0,00	0	0,00
	Desconhecida	0	0,00	1	0,33	2	0,67
19	7	4	1,33	1	0,33	0	0,00
	19	0	0,00	0	0,00	0	0,00
	7 ou 19	0	0,00	0	0,00	0	0,00
266259	3	8	2,67	4	1,33	5	1,67
	12	2	0,67	0	0,00	5	1,67
	14	53	17,67	57	19,00	47	15,67
	3 ou 12	46	15,33	46	15,33	40	13,33
	Desconhecida	0	0,00	0	0,00	0	0,00
Erro total	—	155	2,58	173	2,88	196	3,27
Sugestões Erradas	—	46	0,77	107	1,78	97	1,62

Fonte: Resultado do experimento

Esse comportamento característico do classificador, quando há sobreposição de falhas, faz com que ele não cometa erros de classificação. O comportamento do comitê de classificadores de classe única quando não há treinamento para um determinado tipo de classe pode ser analisado tomando por base a classe 266259. Ele fez as seguintes classificações com os 300 circuitos: classificou 191 circuitos como classe desconhecida, errou 2,67% dos 300 circuitos classificando-os como classe 3, errou 2,67% dos circuitos classificando esses

circuitos como classe 12 e apresentou um erro de 17,67% na classificação desses circuitos multifalhas como classe 14. O comitê de classe única classificando o circuito *ctsv* errou no total 2,58% de todos os circuitos classificados.

Comparando o comitê de classe única com o classificador multiclasse, quando o critério é erro de classificação, o comitê de classe única é muito superior ao classificador multiclasse. O erro total do classificador multiclasse é de 30,43% enquanto que o erro total do comitê de classe única é de 2,58%. As matrizes de confusão estão em Anexo, Figura 35, Figura 36 e Figura 37.

5.2.4 Resultados obtidos pela análise de sinais.

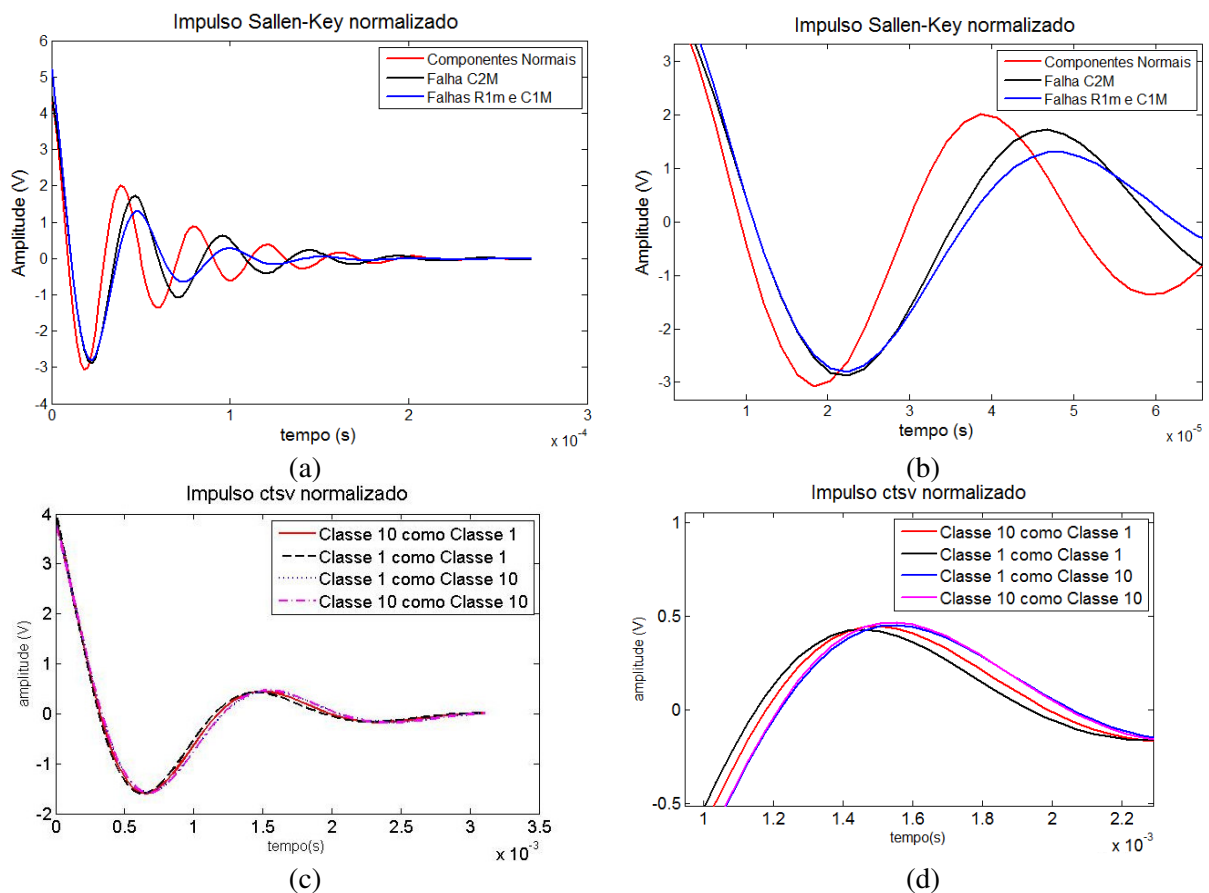
Comparando o comitê de classe única com o classificador multiclasse, quando o critério é erro de classificação, o comitê de classe única apresenta melhores resultados que o classificador multiclasse. O erro total do classificador multiclasse é de 30,43% enquanto o erro total do comitê classe única é de 2,58%.

Pode-se observar que o classificador multiclasse deve, de qualquer forma, classificar um circuito pertencente ao conjunto de dados até que ele pertença a uma classe não treinada. A Figura 21 (a) mostra o sinal de um circuito com múltiplas falhas que foi classificado como pertencente à classe 3. Comparando esse sinal com outros dois sinais pertencentes à classe 3 (os mais parecidos encontrados dentro da classe 3) percebe-se que não existe tanta semelhança entre o sinal do circuito da classe 1036 com os dois sinais dos circuitos 3, isso significa que dentro de todas as classes disponíveis para o classificador, o objeto vai ser classificado na classe que mais se assemelhar, mesmo que o objeto não pertença àquela classe, o que caracteriza erro de classificação.

O circuito *ctsv* tem a resposta de impulso com transiente mais amortecido do que a do *sallen-key*, essa diferença pode ser visto nas formas de onda da Figura 21 (a) com a (c) e a (e). A característica de resposta do impulso para o circuito *ctsv* aumenta a capacidade do circuito de ser menos sensível aos valores de tolerância dos componentes. Por conta desse comportamento, o classificador multiclasse se confunde e erra ao classificar circuito que tenham as mesmas características. No caso de circuitos com falhas simples observa-se que existe sobreposições. Por exemplo, na matriz de confusão ocorreram vários erros de classificação, porém foram analisados os seguintes circuitos:

- da classe 1 que foi classificado como classe 10;
- da classe 1 que foi classificado como classe 1;
- da classe 10 que foram classificados como classe 1; e
- da classe 10 que foram classificados como classe 10.

O resultado gráfico como o da Figura 21 (c) são as sobreposições dos sinais que estão bem próximas e faz com que o classificador se confunda mesmo quando todas as classes tenham sido treinadas e no caso do classificador multiclasse, ele deve escolher uma das classes treinadas. No caso do comitê de classificadores de classe única, indica que há uma sobreposição e não classifica o objeto, mas sugere quais classes ele pode pertencer. Essa sugestão pode ser visto na matriz de confusão na Figura 30. No caso de circuitos com múltiplas falhas o classificador multiclasse, da mesma forma que na classificação dos circuitos *sallen-key*, classifica esses circuitos nas classes mais afins. O comitê de classe única classifica os circuitos com falhas múltiplas como classe desconhecida ou sugere algumas classes em que esses circuitos podem ser classificados e até que não sejam as classes certas, o classificador não os classifica em qualquer uma das classes, vide a Figura 21 (e).



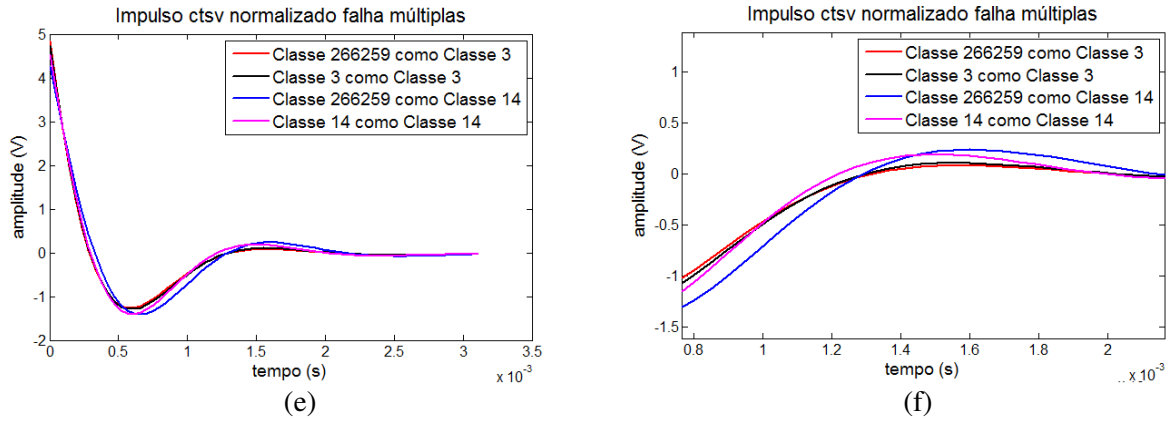


Figura 21 – Gráficos de resposta ao impulso de: (a) Sallen-key múltiplas falhas, (b) zoom de (a), (c) ctsv falhas simples, (d) zoom de (c), (e) ctsv múltiplas falhas, (f) zoom de (e).

6 CONCLUSÃO

O desenvolvimento de estratégias de teste para detectar e diagnosticar falhas em circuitos analógicos e de sinais mistos é uma tarefa complexa. Existem muitos fatores que contribuem para aumentar a dificuldade no teste destes circuitos tais como: a dificuldade de medir correntes sem alterar as conexões, a falta de bons modelos de falha, a falta de um padrão para projeto de circuitos analógicos com vistas a testabilidade e a crescente importância das falhas temporais. Além disso, os métodos clássicos necessitam de grande poder computacional se a identificação de parâmetros for utilizada ou de um grande número de simulações no caso de um dicionário de falhas.

Esse desafio tem estimulado o desenvolvimento de ferramentas que buscam facilitar os procedimentos de detecção de falhas. Em particular o uso de técnicas de Inteligência Computacional tem sido amplamente empregado, sobretudo através da utilização de classificadores para identificação de componentes defeituosos. Uma das arquiteturas mais utilizadas é o classificador multiclasse. Nesta arquitetura, cada saída do classificador representa uma classe de falha. Ela possui duas grandes deficiências: a primeira é a dificuldade de tratar classes de falha que se sobrepõem e a segunda se refere ao fato de que o classificador multiclasse só pode indicar classes de falha para as quais ele tiver sido treinado.

Este trabalho apresentou um sistema de detecção de falhas para circuitos lineares baseada em comitê de classificadores de classe única. A utilização de comitê de classe única tem o objetivo de atenuar as deficiências encontradas na detecção de falhas usando classificadores multiclasse. Neste caso, cada classe de falha é representada por um classificador de classe única que necessita apenas dos exemplos da própria classe para ser treinado.

Foram realizados quatro experimentos envolvendo circuitos que são bastante utilizados na literatura em problemas de detecção de falhas: o filtro passa-banda *Sallen-Key* e o filtro universal. Observou-se que quando o conjunto de testes apresenta apenas classes de falha para qual o classificador foi treinado e estas não apresentam sobreposição, o comitê de classificadores de classe única apresenta um desempenho bastante competitivo, mas ligeiramente inferior do que um classificador multiclasse. Isto acontece porque, o classificador de classe única é treinado para rejeitar uma pequena parcela dos elementos da classe para qual ele foi treinado. Quando o conjunto apresenta superposição de classes, o comitê de classe única consegue identificar esta sobreposição, fornecendo possíveis sugestões

para a falha. Quando o conjunto apresenta uma falha desconhecida, o comitê de classe única consegue, dependendo do tipo de falha, indicar a falha desconhecida.

Uma característica importante do classificador de classe única é que ele necessita apenas de exemplos da classe para o qual ele se destina e também é capaz de identificar falhas para o qual não foi treinado. Isto permite que, além dessa capacidade de identificação, à medida que sejam obtidos mais exemplos de uma classe, somente este classificador precisa ser retreinado.

As propostas para trabalhos futuros no desenvolvimento desta pesquisa estão listadas a seguir.

- A experimentação de novas técnicas de pré-processamento com vistas a melhorar o desempenho na detecção de falhas. Em particular, a utilização da Aproximação Adaptativa por partes (APCA - Adaptive Piecewise Constant Approximation) (KEOGH et al., 2001).
- A aplicação de técnicas de redução de dimensão específicas para classificadores de classe única, principalmente aquelas baseadas em preservação local e coerência de agrupamento (VILLALBA E CUNNINGHAM, 2007).
- A extensão do sistema proposto para incluir diferentes sinais de teste nas entradas do circuito como o objetivo de aumentar a capacidade de detecção e também poder lidar como circuitos não lineares.

REFERÊNCIA

ALBUSTANI HASAN. **Modelling Methods for Testability Analysis of Analog Integrated Circuits Based on Pole-Zero Analysis**. Deutschland: Der Fakultät für Ingenieurwissenschaften der Universität Duisburg-Essen, 2004.

AMINIAN, M.; AMINIAN, F. Neural network based analog circuit fault diagnosis using wavelet transform as preprocessor, **IEEE Trans. Circuits Syst.** 47 (2) (2000) 151–156, 2000

AMINIAN, M.; AMINIAN, F.; A comprehensive examination of neural network architectures for analog fault diagnosis, in: Proceedings of the IJCNN '01, **International Joint Conference on Neural Networks 2001**, III, pp. 2304–2309, 2001.

AMINIAN F., AMINIAN M., COLLINS H.W., Analog fault diagnosis of actual circuits using neural networks, **IEEE Trans. Instrum. Meas.** 15 (3) 544–550, 2002.

AUGUSTO. J. A. S., ALMEIDA, C. B.; A Tool for Single-Fault Diagnosis in Linear Analog Circuits with Tolerance Using the T-Vector Approach. **VLSI Design**. Volume 2008, Article ID 630951, 8 pages.

BANDLER J. W., SALAMA A. E.: Fault Diagnosis of Analog Circuits. **Proceedings of the IEEE**, Vol. 73 N° 8, 1279 – 1325, August 1985.

CATELANI M., A. Fort, Soft fault detection and isolation in analog circuits: some results and a comparison between a fuzzy approach and radial basis function networks, **IEEE Trans. Instrum. Meas.** 51 (2) 196–202, 2002.

CLAASEN, T.A.C.M.: **System on a Chip: Changing IC Design Today and in the Future**. IEEE, Micro, 23(3), 20-26 Mai-June, 2003.

COTA, É. F.; CARRO, L.; LUBASZEWSKI, M.; **A Method to Diagnose Faults in Linear Analog Circuits Using an Adaptive Tester**. Departamento de Engenharia Elétrica Universidade Federal do Rio Grande do Sul - Porto Alegre – BRASIL

CZAJA, ZBIGNIEW: A fault diagnosis algorithm of analog circuits based on node-voltage relation. **12th IMEKO TC1 & TC7 Joint Symposium on Man Science & Measurement**. Annecy, France, p. 297 – 304. September, 2008

CZAJA Zbigniew: A method of fault diagnosis of analog parts of electronic embedded systems with tolerances. **Science Direct Journal Measurement 42**. Poland, p 903 915, 2009.

DARYANANI GOBIND; **Principle of active network synthesis and design**. New York: John Wiley & Sons, 1976.

DUDA, R. O.; HART, P. E.; STORK D. G.: **Pattern Classification**. New York: Wiley-Interscience, 2^a ed. 2002.

DURAN O., ALTHOEFER K., SENEVIRATNE L. D. Automated Pipe Defect Detection and Categorization Using Camera/Laser-Based Profiler and Artificial Neural Network. **IEEE Transactions on Automation Science and Engineering**. Vol. 4. N° 1. p. 118-126, January 2007.

DUHAMEL, P.; RAULT, J. C.: Automatic Test Generation Techniques for Analog Circuits and Systems: a Review. **IEEE, Transactions on Circuits and Systems, CAS26** (7). July – 1979.

FANNI, A., GIUA, A., MARCHESI, M., MONTISCI, A.: A Neural Network Diagnosis Approach for Analog Circuits. Applied Intelligence, **Journal of Applied Intelligence**. 1 - 20, January 1999.

FANNI, A., GIUA, A., SANDOLI, E., Neural networks for multiple fault diagnosis in analog circuits, in: **Proc. 1993 IEEE Int.** 1993.

FENTON, W. G., MCGINNITY, T. M., MAGUIRE, L. P.: Fault diagnosis of electronic systems using intelligent techniques: a review. **IEEE, Transactions on Systems, Man and Cybernetics – Part C**, 31(3), 269–281, August 2001.

JUNIOR, M. P. P.; **Combinação de Múltiplos Classificadores para Identificação de Materiais em Imagens Ruidosas**. Brasil: Programa de Pós-Graduação em Ciência de Computação da Universidade Federal de São Carlos, 2004.

KAMINSKA B., ARABI K., BELL I., GOTETI P., HUERTAS J.L., KIM B., RUEDA A., SOMA M.; Analog and Mixed-Signal Benchmark Circuits. **TTTC Mixed-Signal Testing Technical Activity Committee, IEEE International Test Conference**. 1997, Paper 8.3 p. 183 – 190.

KEOGH, E., CHAKRABARTI, K., PAZZANI, M., MEHROTRA, S. Locally adaptive dimensionality reduction for indexing large time series databases. **In proceedings of ACM SIGMOD Conference on Management of Data**, May. pp 151-162. 2001.

KEOGH, E.; CHAKRABARTI, K.; PAZZANI, M.; MEHROTRA, S.. Dimensionality reduction for fast similarity search in large time series databases. **J Knowl Inf Syst** 3(3):263–286, (2000)

KUNCHEVA, I. LUNDIMILA; **Combining Pattern Classifier Methods and Algorithms**. New Jersey: Wiley-Interscience, 2004.

LIU, D.; STARZYK, J. A.: **A generalized fault diagnosis in dynamic analog circuits**. School of Electrical Engineering and Computer Science, p. 19. Ohio University, Athens, OH, 45701, USA, 2002.

LIU, H.; CHEN, G.; JIANG, S.; SONG, G.: A Survey of Feature Extraction Approaches in Analog Circuit Fault Diagnosis. **IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application**, p. 676 – 680, China, 2008.

MANDERS, E. J.; NARASIMHAN, S.; BISWAS, G.; MOSTERMAN, P.J.: A combined qualitative/quantitative approach for fault isolation in continuous dynamic systems. **In SafeProcess 2000**, volume 1, pages 1074–1079, Budapest, Hungary, June 2000.

MUSTAPHA S., BOZENA K.: Analog Circuit Fault Diagnosis Based on Sensitivity Computation and Functional Testing. **IEEE design & test of computers**, 30 -39, March 1992.

NETO, LUIS BIONDI; **Aprendizado não Supervisionado**. Programa de Pós-Graduação em Engenharia Eletrônica da Universidade Estadual do Rio de Janeiro, 2006.

SALAT, R.; OSOWSKI, S. **Analog Filter Diagnosis Using Support Vector Machine**, in: **ECCTD'03**, Krakow, 2003, III, pp. 421 – 424.

SANTIAGO D., CUNNINGHAM VILLALBA , CUNNINGHAM PÁDRAIG; An evaluation of dimension reduction techniques for one-class classification. **Artificial Intelligence Review** Volume 27, Number 4, 273-294, DOI: 10.1007/s10462-008-9082-5

SINGH, A.; PATEL, C.; PLUSQUELLIC, J.; **On-Chip Impulse Response Generation for Analog and Mixed-Signal Testing**. itc2004, analog testing.

SPINA R., UPADHYAYA S.; Linear circuit fault diagnosis using neuromorphic analyzer. **IEEE Trans. Circuits Syst. II** 44, 1997.

TAX, D.M.J. **dd tools 1.5.0 A Matlab Toolbox for Data Description, Outlier and Novelty Detection**, Delft University of Technology, 2006.

TAX, D.M.J. **PRTtools4 A Matlab Toolbox for Pattern Recognition**, Delft University of Technology, 2007.

VILLALBA, S.; CUNNINGHAM, P.; An Evaluation of Dimension Reduction Techniques for One-Class Classification, **Proceedings of 18th Irish Conference on Artificial Intelligence and Cognitive Science**, S. J. Delany, M. Madden (eds), pp171- 180, 2002.

WITTEN, I. H.; FRANK. E., **Data Mining: Practical Machine Learning Tools and Techniques**, San Francisco: Morgan Kaufmann, 2 edition ,June , 2005

Z. Yi et al; Work on Defect and Fault Tolerance in VLSI Systems, **Measurement 41 371–380 379**, 1993, pp. 303–310, 1993.

YI, Z.; XUEYE W.; HAIFENG J.: One-class classifier based on SBT for analog circuit fault diagnosis. **Science Direct Measurement**. Measurement 48. 371 – 380, March 2007.

ANEXO A

True Labels	Estimated Labels											Totals	
	1	2	3	4	5	6	7	8	9	10	11		
1	300	0	0	0	0	0	0	0	0	0	0	0	300
2	0	300	0	0	0	0	0	0	0	0	0	0	300
3	0	0	300	0	0	0	0	0	0	0	0	0	300
4	0	0	0	300	0	0	0	0	0	0	0	0	300
5	0	0	0	0	300	0	0	0	0	0	0	0	300
6	0	0	0	0	0	300	0	0	0	0	0	0	300
7	0	0	0	0	0	0	300	0	0	0	0	0	300
8	0	0	0	0	0	0	0	300	0	0	0	0	300
9	0	0	0	0	0	0	0	0	300	0	0	0	300
10	0	0	0	0	0	0	0	0	0	300	0	0	300
11	0	0	0	0	0	0	0	0	0	0	300	0	300
Totals	300	300	300	300	300	300	300	300	300	300	300	300	3300

Figura 22 – Matriz confusão (MC) o classificador multiclasse knn SK falhas simples

True Labels	Estimated Labels											Totals	
	0	1	2	3	4	5	6	7	8	9	10		11
1	2	298	0	0	0	0	0	0	0	0	0	0	300
2	6	0	294	0	0	0	0	0	0	0	0	0	300
3	4	0	0	296	0	0	0	0	0	0	0	0	300
4	0	0	0	0	300	0	0	0	0	0	0	0	300
5	2	0	0	0	0	298	0	0	0	0	0	0	300
6	0	0	0	0	0	0	300	0	0	0	0	0	300
7	0	0	0	0	0	0	0	300	0	0	0	0	300
8	2	0	0	0	0	0	0	0	298	0	0	0	300
9	2	0	0	0	0	0	0	0	0	298	0	0	300
10	5	0	0	0	0	0	0	0	0	0	295	0	300
11	3	0	0	0	0	0	0	0	0	0	0	297	300
Totals	26	298	294	296	300	298	300	300	298	298	295	297	3300

Figura 23 – MC classificador classe única knn_dd SK falhas simples

True Labels	Estimated Labels														Totals
	0	1	2	3	4	5	6	7	8	9	10	11	143		
1	8	292	0	0	0	0	0	0	0	0	0	0	0	0	300
2	4	0	296	0	0	0	0	0	0	0	0	0	0	0	300
3	3	0	0	297	0	0	0	0	0	0	0	0	0	0	300
4	3	0	0	0	297	0	0	0	0	0	0	0	0	0	300
5	1	0	0	0	0	299	0	0	0	0	0	0	0	0	300
6	4	0	0	0	0	0	296	0	0	0	0	0	0	0	300
7	5	0	0	0	0	0	0	295	0	0	0	0	0	0	300
8	5	0	0	0	0	0	0	0	295	0	0	0	0	0	300
9	2	0	0	0	0	0	0	0	0	260	0	0	0	38	300
10	7	0	0	0	0	0	0	0	0	0	293	0	0	0	300
11	10	0	0	0	0	0	0	0	0	0	0	290	0	0	300
Totals	52	292	296	297	297	299	296	295	295	260	293	290	38	3300	

Figura 24 – MC classificador classe única gauss_dd SK falhas simples

True Labels	Estimated Labels														Totals
	0	1	2	3	4	5	6	7	8	9	10	11	143		
1	4	296	0	0	0	0	0	0	0	0	0	0	0	0	300
2	9	0	291	0	0	0	0	0	0	0	0	0	0	0	300
3	8	0	0	292	0	0	0	0	0	0	0	0	0	0	300
4	5	0	0	0	295	0	0	0	0	0	0	0	0	0	300
5	5	0	0	0	0	295	0	0	0	0	0	0	0	0	300
6	2	0	0	0	0	0	298	0	0	0	0	0	0	0	300
7	3	0	0	0	0	0	0	297	0	0	0	0	0	0	300
8	6	0	0	0	0	0	0	0	294	0	0	0	0	0	300
9	3	0	0	0	3	0	0	0	0	101	0	0	193	0	300
10	10	0	0	0	0	0	0	0	0	0	290	0	0	0	300
11	13	0	0	0	0	0	0	0	0	0	0	287	0	0	300
Totals	68	296	291	292	298	295	298	297	294	101	290	287	193	3300	

Figura 25 – MC classificador classe única kmeans_dd SK falhas simples

True Labels	Estimated Labels											Totals	
	1	2	3	4	5	6	7	8	9	10	11		
1	300	0	0	0	0	0	0	0	0	0	0	0	300
2	0	300	0	0	0	0	0	0	0	0	0	0	300
3	0	0	300	0	0	0	0	0	0	0	0	0	300
4	0	0	0	300	0	0	0	0	0	0	0	0	300
5	0	0	0	0	300	0	0	0	0	0	0	0	300
6	0	0	0	0	0	300	0	0	0	0	0	0	300
7	0	0	0	0	0	0	300	0	0	0	0	0	300
8	0	0	0	0	0	0	0	300	0	0	0	0	300
9	0	0	0	0	0	0	0	0	300	0	0	0	300
10	0	0	0	0	0	0	0	0	0	300	0	0	300
11	0	0	0	0	0	0	0	0	0	0	300	0	300
1036	0	0	181	0	0	0	0	119	0	0	0	0	300
Totals	300	300	481	300	300	300	300	419	300	300	300	300	3600

Figura 26 – MC classificador multiclasse knn SK múltiplas falhas.

True Labels	Estimated Labels											Totals	
	0	1	2	3	4	5	6	7	8	9	10		11
1	7	293	0	0	0	0	0	0	0	0	0	0	300
2	3	0	297	0	0	0	0	0	0	0	0	0	300
3	2	0	0	298	0	0	0	0	0	0	0	0	300
4	4	0	0	0	296	0	0	0	0	0	0	0	300
5	1	0	0	0	0	299	0	0	0	0	0	0	300
6	0	0	0	0	0	0	300	0	0	0	0	0	300
7	4	0	0	0	0	0	0	296	0	0	0	0	300
8	2	0	0	0	0	0	0	0	298	0	0	0	300
9	0	0	0	0	0	0	0	0	0	300	0	0	300
10	2	0	0	0	0	0	0	0	0	0	298	0	300
11	6	0	0	0	0	0	0	0	0	0	0	294	300
1036	300	0	0	0	0	0	0	0	0	0	0	0	300
Totals	331	293	297	298	296	299	300	296	298	300	298	294	3600

Figura 27 – MC classificador classe única knn_dd SK múltiplas falhas.

True Labels	Estimated Labels													Totals
	0	1	2	3	4	5	6	7	8	9	10	11	143	
1	3	297	0	0	0	0	0	0	0	0	0	0	0	300
2	2	0	298	0	0	0	0	0	0	0	0	0	0	300
3	0	0	0	300	0	0	0	0	0	0	0	0	0	300
4	3	0	0	0	297	0	0	0	0	0	0	0	0	300
5	1	0	0	0	0	299	0	0	0	0	0	0	0	300
6	2	0	0	0	0	0	298	0	0	0	0	0	0	300
7	8	0	0	0	0	0	0	292	0	0	0	0	0	300
8	8	0	0	0	0	0	0	0	292	0	0	0	0	300
9	3	0	0	0	0	0	0	0	0	256	0	0	41	300
10	1	0	0	0	0	0	0	0	0	0	299	0	0	300
11	7	0	0	0	0	0	0	0	0	0	0	293	0	300
1036	300	0	0	0	0	0	0	0	0	0	0	0	0	300
Totals	338	297	298	300	297	299	298	292	292	256	299	293	41	3600

Figura 28 – MC classificador classe única gauss_dd SK múltiplas falhas.

True Labels	Estimated Labels													Totals
	0	1	2	3	4	5	6	7	8	9	10	11	143	
1	3	297	0	0	0	0	0	0	0	0	0	0	0	300
2	10	0	290	0	0	0	0	0	0	0	0	0	0	300
3	6	0	0	294	0	0	0	0	0	0	0	0	0	300
4	5	0	0	0	295	0	0	0	0	0	0	0	0	300
5	1	0	0	0	0	299	0	0	0	0	0	0	0	300
6	0	0	0	0	0	0	300	0	0	0	0	0	0	300
7	6	0	0	0	0	0	0	294	0	0	0	0	0	300
8	6	0	0	0	0	0	0	0	294	0	0	0	0	300
9	0	0	0	0	7	0	0	0	0	198	0	0	95	300
10	3	0	0	0	0	0	0	0	0	0	297	0	0	300
11	4	0	0	0	0	0	0	0	0	0	0	296	0	300
1036	300	0	0	0	0	0	0	0	0	0	0	0	0	300
Totals	344	297	290	294	302	299	300	294	294	198	297	296	95	3600

Figura 29 – MC para o classificador classe única kmeans_dd SK múltiplas falhas.

	True Labels		Estimated Labels						
	1	2	3	4	5	6	7	8	9
1	166	0	0	0	0	0	0	0	0
2	0	162	0	0	0	0	0	0	0
3	0	0	138	0	0	0	0	0	0
4	0	0	0	164	0	0	0	0	0
5	0	0	0	0	300	0	0	0	0
6	0	0	0	0	0	300	0	0	0
7	0	0	0	0	0	0	92	0	0
8	0	0	0	0	0	0	0	300	0
9	0	0	0	0	0	0	0	0	300
10	168	0	0	0	0	0	0	0	0
11	0	138	0	0	0	0	0	0	0
12	0	0	138	0	0	0	0	0	0
13	0	0	0	167	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	4	0	0
17	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	45	0	0
Totals	334	300	276	331	300	300	141	300	300

(a)

	True										Totals
	10	11	12	13	14	15	16	17	18	19	
1	134	0	0	0	0	0	0	0	0	0	300
2	0	138	0	0	0	0	0	0	0	0	300
3	0	0	162	0	0	0	0	0	0	0	300
4	0	0	0	136	0	0	0	0	0	0	300
5	0	0	0	0	0	0	0	0	0	0	300
6	0	0	0	0	0	0	0	0	0	0	300
7	0	0	0	0	0	0	82	0	0	126	300
8	0	0	0	0	0	0	0	0	0	0	300
9	0	0	0	0	0	0	0	0	0	0	300
10	132	0	0	0	0	0	0	0	0	0	300
11	0	162	0	0	0	0	0	0	0	0	300
12	0	0	162	0	0	0	0	0	0	0	300
13	0	0	0	133	0	0	0	0	0	0	300
14	0	0	0	0	300	0	0	0	0	0	300
15	0	0	0	0	0	300	0	0	0	0	300
16	0	0	0	0	0	0	296	0	0	0	300
17	0	0	0	0	0	0	0	300	0	0	300
18	0	0	0	0	0	0	0	0	300	0	300
19	0	0	0	0	0	0	0	0	0	255	300
Total	266	300	324	269	300	300	378	300	300	381	5700

(b)

Figura 30 – (a) e (b) MC classificador multiclasse *knn ctsv* falhas simples.

True Labels	Estimated Labels													
	0	2	3	4	5	6	7	8	9	10	11	13	14	15
1	0	0	0	0	0	0	0	0	0	1	0	0	0	0
2	0	5	0	0	0	0	0	0	0	0	1	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	0	0	3	0	0	0	0	0	0	0	3	0	0
5	3	0	0	0	297	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	299	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	49	0	0	0	0	0	0	0
8	3	0	0	0	0	0	0	297	0	0	0	0	0	0
9	5	0	0	0	0	0	0	0	295	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	4	0	0	0	0
11	0	8	0	0	0	0	0	0	0	0	1	0	0	0
12	0	0	2	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	6	0	0	0	0	0	0	0	2	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	299	0
15	2	0	0	0	0	0	0	0	0	0	0	0	0	298
16	2	0	0	0	0	0	3	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	1	0	0	0	0	0	4	0	0	0	0	0	0	0
Totals	21	13	2	9	297	299	56	297	295	5	2	5	299	298

(a)

True Labels	16	17	18	19	25	4116	4123	32851	65683	131347	262675
1	0	0	0	0	0	0	0	0	0	0	299
2	0	0	0	0	0	0	0	0	0	294	0
3	0	0	0	0	0	0	0	0	299	0	0
4	0	0	0	0	0	0	0	292	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	164	86	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	296
11	0	0	0	0	0	0	0	0	0	291	0
12	0	0	0	0	0	0	0	0	298	0	0
13	0	0	0	0	0	0	0	292	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0
16	17	0	0	0	0	0	278	0	0	0	0
17	0	300	0	0	0	0	0	0	0	0	0
18	0	0	299	0	1	0	0	0	0	0	0
19	0	0	0	22	0	273	0	0	0	0	0
Totals	18	300	299	22	1	437	364	584	597	585	595

(b)

Figura 31 – (a) e (b) MC classificador classe única knn_dd ctsv falhas simples

True Labels	Estimated Labels												
	0	1	3	4	5	6	7	8	9	10	11	12	14
1	1	1	0	0	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	2	0	0
3	1	0	1	0	0	0	0	0	0	0	0	1	0
4	0	0	0	3	0	0	0	0	0	0	0	0	0
5	3	0	0	0	297	0	0	0	0	0	0	0	0
6	2	0	0	0	0	298	0	0	0	0	0	0	0
7	0	0	0	0	0	0	44	0	0	0	0	0	0
8	6	0	0	0	0	0	0	294	0	0	0	0	0
9	2	0	0	0	0	0	0	0	298	0	0	0	0
10	3	0	0	0	0	0	0	0	0	1	0	0	0
11	3	0	0	0	0	0	0	0	0	0	0	0	0
12	1	0	2	0	0	0	0	0	0	0	0	0	0
13	2	0	0	1	0	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	299
15	3	0	0	0	0	0	0	0	0	0	0	0	0
16	2	0	0	0	0	0	1	0	0	0	0	0	0
17	3	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0	0
19	2	0	0	0	0	0	4	0	0	0	0	0	0
Totals	38	1	3	4	297	298	49	294	298	1	2	1	299

(a)

True Labels	Estimated Labels											Totals
	15	16	17	18	19	4116	4123	32851	65683	131347	262675	
1	0	0	0	0	0	0	0	0	0	0	298	300
2	0	0	0	0	0	0	0	0	0	296	0	300
3	0	0	0	0	0	0	0	0	297	0	0	300
4	0	0	0	0	0	0	0	297	0	0	0	300
5	0	0	0	0	0	0	0	0	0	0	0	300
6	0	0	0	0	0	0	0	0	0	0	0	300
7	0	0	0	0	0	164	92	0	0	0	0	300
8	0	0	0	0	0	0	0	0	0	0	0	300
9	0	0	0	0	0	0	0	0	0	0	0	300
10	0	0	0	0	0	0	0	0	0	0	296	300
11	0	0	0	0	0	0	0	0	0	297	0	300
12	0	0	0	0	0	0	0	0	297	0	0	300
13	0	0	0	0	0	0	0	297	0	0	0	300
14	0	0	0	0	0	0	0	0	0	0	0	300
15	297	0	0	0	0	0	0	0	0	0	0	300
16	0	7	0	0	0	0	290	0	0	0	0	300
17	0	0	297	0	0	0	0	0	0	0	0	300
18	0	0	0	299	0	0	0	0	0	0	0	300
19	0	0	0	0	25	269	0	0	0	0	0	300
Totals	297	7	297	299	25	433	382	594	594	593	594	5700

(b)

Figura 32 – (a) e (b) MC classificador classe única gauss_dd ctsv falhas simples

True Labels	Estimated Labels													
	0	1	3	4	5	6	7	8	9	10	11	13	14	
1	1	0	0	0	0	0	0	0	0	1	0	0	0	
2	2	0	0	0	0	0	0	0	0	0	1	0	0	
3	0	0	4	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	2	0	0	0	0	0	0	0	3	0	
5	3	0	0	0	297	0	0	0	0	0	0	0	0	
6	4	0	0	0	0	296	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	46	0	0	0	0	0	0	
8	5	0	0	0	0	0	0	295	0	0	0	0	0	
9	3	0	0	0	0	0	0	0	297	0	0	0	0	
10	2	1	0	0	0	0	0	0	0	0	0	0	0	
11	3	0	0	0	0	0	0	0	0	0	4	0	0	
12	1	0	0	0	0	0	0	0	0	0	0	0	0	
13	0	0	0	2	0	0	0	0	0	0	0	5	0	
14	1	0	0	0	0	0	0	0	0	0	0	0	299	
15	8	0	0	0	0	0	0	0	0	0	0	0	0	
16	3	0	0	0	0	0	3	0	0	0	0	0	0	
17	3	0	0	0	0	0	0	0	0	0	0	0	0	
18	1	0	0	0	0	0	0	0	0	0	0	0	0	
19	2	0	0	0	0	0	7	0	0	0	0	0	0	
Totals	42	1	4	4	297	296	56	295	297	1	5	8	299	

(a)

True Labels	Estimated Labels												Totals
	15	16	17	18	19	25	4116	4123	32851	65683	131347	262675	
1	0	0	0	0	0	0	0	0	0	0	0	298	300
2	0	0	0	0	0	0	0	0	0	0	297	0	300
3	0	0	0	0	0	0	0	0	0	296	0	0	300
4	0	0	0	0	0	0	0	0	295	0	0	0	300
5	0	0	0	0	0	0	0	0	0	0	0	0	300
6	0	0	0	0	0	0	0	0	0	0	0	0	300
7	0	1	0	0	1	0	165	87	0	0	0	0	300
8	0	0	0	0	0	0	0	0	0	0	0	0	300
9	0	0	0	0	0	0	0	0	0	0	0	0	300
10	0	0	0	0	0	0	0	0	0	0	0	297	300
11	0	0	0	0	0	0	0	0	0	0	293	0	300
12	0	0	0	0	0	0	0	0	0	299	0	0	300
13	0	0	0	0	0	0	0	0	293	0	0	0	300
14	0	0	0	0	0	0	0	0	0	0	0	0	300
15	292	0	0	0	0	0	0	0	0	0	0	0	300
16	0	17	0	0	0	0	0	277	0	0	0	0	300
17	0	0	297	0	0	0	0	0	0	0	0	0	300
18	0	0	0	241	0	58	0	0	0	0	0	0	300
19	0	0	0	0	28	0	263	0	0	0	0	0	300
Totals	292	18	297	241	29	58	428	364	588	595	590	595	5700

(b)

Figura 33 – (a) e (b) MC classificador classe única *kmeans_dd csv* falhas simples.

True Labels	Estimated Labels								
	1	2	3	4	5	6	7	8	9
1	151	0	0	0	0	0	0	0	0
2	0	155	0	0	0	0	0	0	0
3	0	0	145	0	0	0	0	0	0
4	0	0	0	155	0	0	0	0	0
5	0	0	0	0	300	0	0	0	0
6	0	0	0	0	0	300	0	0	0
7	0	0	0	0	0	0	126	0	0
8	0	0	0	0	0	0	0	300	0
9	0	0	0	0	0	0	0	0	300
10	146	0	0	0	0	0	0	0	0
11	0	161	0	0	0	0	0	0	0
12	0	0	142	0	0	0	0	0	0
13	0	0	0	155	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	55	0	0
17	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	99	0	0
266259	0	0	145	0	0	0	0	0	0
Totals	297	316	432	310	300	300	280	300	300

(a)

True Labels	10	11	12	13	14	15	16	17	18	19	Totals
1	149	0	0	0	0	0	0	0	0	0	300
2	0	145	0	0	0	0	0	0	0	0	300
3	0	0	155	0	0	0	0	0	0	0	300
4	0	0	0	145	0	0	0	0	0	0	300
5	0	0	0	0	0	0	0	0	0	0	300
6	0	0	0	0	0	0	0	0	0	0	300
7	0	0	0	0	0	0	67	0	0	107	300
8	0	0	0	0	0	0	0	0	0	0	300
9	0	0	0	0	0	0	0	0	0	0	300
10	154	0	0	0	0	0	0	0	0	0	300
11	0	139	0	0	0	0	0	0	0	0	300
12	0	0	158	0	0	0	0	0	0	0	300
13	0	0	0	145	0	0	0	0	0	0	300
14	0	0	0	0	300	0	0	0	0	0	300
15	0	0	0	0	0	300	0	0	0	0	300
16	0	0	0	0	0	0	245	0	0	0	300
17	0	0	0	0	0	0	0	300	0	0	300
18	0	0	0	0	0	0	0	0	300	0	300
19	0	0	0	0	0	0	0	0	0	201	300
266259	0	0	33	0	122	0	0	0	0	0	300
Totals	303	284	346	290	422	300	312	300	300	308	6000

(b)

Figura 34 – (a) e (b) MC classificador multiclasse knn circuito ctsv falhas múltiplas

True Labels	Estimated Labels											
	0	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	2	0	0
2	0	6	0	0	0	0	0	0	0	0	1	0
3	1	0	4	0	0	0	0	0	0	0	0	0
4	0	0	0	7	0	0	0	0	0	0	0	0
5	4	0	0	0	296	0	0	0	0	0	0	0
6	2	0	0	0	0	298	0	0	0	0	0	0
7	0	0	0	0	0	0	42	0	0	0	0	0
8	5	0	0	0	0	0	0	295	0	0	0	0
9	4	0	0	0	0	0	0	0	296	0	0	0
10	0	0	0	0	0	0	0	0	0	3	0	0
11	0	3	0	0	0	0	0	0	0	0	0	0
12	1	0	2	0	0	0	0	0	0	0	0	0
13	0	0	0	5	0	0	0	0	0	0	0	0
14	4	0	0	0	0	0	0	0	0	0	0	0
15	3	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	3	0	0	0	0	0
17	1	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	4	0	0	0	0	0
266259	191	0	8	0	0	0	0	0	0	0	0	2
Totals	216	9	14	12	296	298	49	295	296	5	1	2

(a)

True Labels	13	14	15	16	17	18	19	25	4116	4123	32851	65683	131347	262675	Totals
1	0	0	0	0	0	0	0	0	0	0	0	0	0	298	300
2	0	0	0	0	0	0	0	0	0	0	0	0	293	0	300
3	0	0	0	0	0	0	0	0	0	0	0	295	0	0	300
4	2	0	0	0	0	0	0	0	0	0	291	0	0	0	300
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300
7	0	0	0	0	0	0	0	0	171	87	0	0	0	0	300
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300
10	0	0	0	0	0	0	0	0	0	0	0	0	0	297	300
11	0	0	0	0	0	0	0	0	0	0	0	0	297	0	300
12	0	0	0	0	0	0	0	0	0	0	0	297	0	0	300
13	4	0	0	0	0	0	0	0	0	0	291	0	0	0	300
14	0	296	0	0	0	0	0	0	0	0	0	0	0	0	300
15	0	0	297	0	0	0	0	0	0	0	0	0	0	0	300
16	0	0	0	30	0	0	0	0	0	267	0	0	0	0	300
17	0	0	0	0	299	0	0	0	0	0	0	0	0	0	300
18	0	0	0	0	0	299	0	1	0	0	0	0	0	0	300
19	0	0	0	0	0	0	24	0	272	0	0	0	0	0	300
266259	0	53	0	0	0	0	0	0	0	0	0	46	0	0	300
Totals	6	349	297	30	299	299	24	1	443	354	582	638	590	595	6000

(b)

Figura 35 – (a) e (b) MC classificador classe única knn_dd ctsv falhas múltiplas.

True Labels	Estimated Labels											
	0	3	4	5	6	7	8	9	10	11	14	15
1	1	0	0	0	0	0	0	0	1	0	0	0
2	2	0	0	0	0	0	0	0	0	5	0	0
3	3	4	0	0	0	0	0	0	0	0	0	0
4	2	0	3	0	0	0	0	0	0	0	0	0
5	5	0	0	295	0	0	0	0	0	0	0	0
6	3	0	0	0	297	0	0	0	0	0	0	0
7	0	0	0	0	0	40	0	0	0	0	0	0
8	5	0	0	0	0	0	295	0	0	0	0	0
9	2	0	0	0	0	0	0	298	0	0	0	0
10	0	0	0	0	0	0	0	0	2	0	0	0
11	2	0	0	0	0	0	0	0	0	2	0	0
12	5	2	0	0	0	0	0	0	0	0	0	0
13	0	0	4	0	0	0	0	0	0	0	0	0
14	6	0	0	0	0	0	0	0	0	0	294	0
15	6	0	0	0	0	0	0	0	0	0	0	294
16	2	0	0	0	0	3	0	0	0	0	0	0
17	6	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	1	0	0	0	0	0	0
266259	193	4	0	0	0	0	0	0	0	0	57	0
Totals	244	10	7	295	297	44	295	298	3	7	351	294

(a)

True Labels	Estimated Labels										Totals
	16	17	18	19	4116	4123	32851	65683	131347	262675	
1	0	0	0	0	0	0	0	0	0	298	300
2	0	0	0	0	0	0	0	0	293	0	300
3	0	0	0	0	0	0	0	293	0	0	300
4	0	0	0	0	0	0	295	0	0	0	300
5	0	0	0	0	0	0	0	0	0	0	300
6	0	0	0	0	0	0	0	0	0	0	300
7	0	0	0	1	168	91	0	0	0	0	300
8	0	0	0	0	0	0	0	0	0	0	300
9	0	0	0	0	0	0	0	0	0	0	300
10	0	0	0	0	0	0	0	0	0	298	300
11	0	0	0	0	0	0	0	0	296	0	300
12	0	0	0	0	0	0	0	293	0	0	300
13	0	0	0	0	0	0	296	0	0	0	300
14	0	0	0	0	0	0	0	0	0	0	300
15	0	0	0	0	0	0	0	0	0	0	300
16	8	0	0	0	0	287	0	0	0	0	300
17	0	294	0	0	0	0	0	0	0	0	300
18	0	0	299	0	0	0	0	0	0	0	300
19	0	0	0	32	267	0	0	0	0	0	300
266259	0	0	0	0	0	0	0	46	0	0	300
Totals	8	294	299	33	435	378	591	632	589	596	6000

(b)

Figura 36 – (a) e (b) MC classificador classe única gauss_dd csv falhas múltiplas.

True Labels	Estimated Labels													
	0	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1	0	0	0	0	0	0	0	0	7	0	0	0	0
2	0	10	0	0	0	0	0	0	0	0	4	0	0	0
3	0	0	4	0	0	0	0	0	0	0	0	3	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0
5	9	0	0	0	291	0	0	0	0	0	0	0	0	0
6	8	0	0	0	0	292	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	50	0	0	0	0	0	0	0
8	8	0	0	0	0	0	0	292	0	0	0	0	0	0
9	8	0	0	0	0	0	0	0	292	0	0	0	0	0
10	4	0	0	0	0	0	0	0	0	4	0	0	0	0
11	0	5	0	0	0	0	0	0	0	0	13	0	0	0
12	2	0	4	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	5	0	0	0	0	0	0	0	0	2	0
14	7	0	0	0	0	0	0	0	0	0	0	0	0	293
15	7	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	8	0	0	0	0	0	0	0
17	6	0	0	0	0	0	0	0	0	0	0	0	0	0
18	2	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0
266259	203	0	5	0	0	0	0	0	0	0	5	0	0	47
Totals	265	15	13	6	291	292	58	292	292	11	17	8	2	340

(a)

True Labels	Estimated Labels											Totals	
	15	16	17	18	19	25	4116	4123	32851	65683	131347		262675
1	0	0	0	0	0	0	0	0	0	0	0	292	300
2	0	0	0	0	0	0	0	0	0	0	286	0	300
3	0	0	0	0	0	0	0	0	0	293	0	0	300
4	0	0	0	0	0	0	0	0	299	0	0	0	300
5	0	0	0	0	0	0	0	0	0	0	0	0	300
6	0	0	0	0	0	0	0	0	0	0	0	0	300
7	0	0	0	0	2	0	164	84	0	0	0	0	300
8	0	0	0	0	0	0	0	0	0	0	0	0	300
9	0	0	0	0	0	0	0	0	0	0	0	0	300
10	0	0	0	0	0	0	0	0	0	0	0	292	300
11	0	0	0	0	0	0	0	0	0	0	282	0	300
12	0	0	0	0	0	0	0	0	0	294	0	0	300
13	0	0	0	0	0	0	0	0	293	0	0	0	300
14	0	0	0	0	0	0	0	0	0	0	0	0	300
15	293	0	0	0	0	0	0	0	0	0	0	0	300
16	0	17	0	0	0	0	0	275	0	0	0	0	300
17	0	0	294	0	0	0	0	0	0	0	0	0	300
18	0	0	1	287	0	10	0	0	0	0	0	0	300
19	0	0	0	0	39	0	261	0	0	0	0	0	300
266259	0	0	0	0	0	0	0	0	0	40	0	0	300
Totals	293	17	295	287	41	10	425	359	592	627	568	584	6000

(b)

Figura 37 – (a) e (b) MC classificador classe única *kmeans_dd csv* falhas múltiplas.