



**Universidade do Estado do Rio de Janeiro**

Centro de Tecnologia e Ciência

Faculdade de Engenharia

Clayton Roberto de Araújo Moraes


**Detecção de falhas em circuitos eletrônicos através de Comitê  
de classificadores de classe única utilizando técnicas de  
redução de dimensão e representações de dissimilaridade**

Rio de Janeiro

2017

Clayton Roberto de Araújo Moraes

**Detecção de falhas em circuitos eletrônicos através de Comitê de classificadores de classe única utilizando técnicas de redução de dimensão e representações de dissimilaridade**



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

2017

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

M827 Moraes, Clayton Roberto de Araújo.  
Detecção de falhas em circuitos eletrônicos através de  
Comitê de classificadores de classe única utilizando técnicas de  
redução de dimensão e reapresentações de dissimilaridade /  
Clayton Roberto de Araújo Moraes. – 2017.  
232f.

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

1. Engenharia Eletrônica. 2. Comitês de classificadores -  
Dissertações. 3. Representação de dados - Dissertações. 4.  
Circuitos eletrônicos - Análise - Dissertações. I. Amaral, Jorge  
Luís Machado do. II. Universidade do Estado do Rio de Janeiro.  
III. Título.

CDU 004.21

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

---

Assinatura

---

Data

Clayton Roberto de Araújo Moraes

**Detecção de falhas em circuitos eletrônicos através de Comitê de classificadores de classe única utilizando técnicas de redução de dimensão e representações de dissimilaridade**

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:

Banca Examinadora:

---

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

---

Prof. Dr. Ricardo Tanscheit  
Dep. de Engenharia Elétrica, PUC - Rio

---

Prof. Dr. Eduardo Costa da Silva  
Dep. de Engenharia Elétrica, PUC – Rio

---

Prof. Dr. Michel Pompeu Tcheou  
Faculdade de Engenharia – UERJ

Rio de Janeiro

2017

## DEDICATÓRIA

Dedico este trabalho à minha família, a todos os meus amigos de UERJ e da vida e à minha querida Universidade, que hoje encontra-se em triste estado de deterioração por todos os problemas políticos que foram gerados ao nosso estado por ganância e corrupção. Espero que este trabalho possa contribuir para perpetuar a vida e o contínuo sucesso da minha amada UERJ.

## AGRADECIMENTOS

Agradeço a Deus, por toda a força e capacidade que ele tem me oferecido para que eu conseguisse percorrer todo o caminho até aqui.

Agradeço a meus pais, Roberto e Rosângela, por todo o suporte, e auxílio que têm me oferecido com a ajuda de Deus e muito suor e sacrifícios realizados para que eu pudesse estar aonde estou neste momento, muito obrigado por cada esforço em meu nome.

Agradeço a meus irmãos, Kelly e Cassiano, por todo o auxílio prestado quando necessitava e pela amizade verdadeira que tenho em vós.

Agradeço à minha namorada Priscilla, por todo o companheirismo, amizade e carinho que tem me oferecido e tentando ser sempre o meu ponto de equilíbrio para me levar às minhas vitórias, toda a paciência e compreensão em entender minha ausência nos momentos de necessidade, embora estava a suportar a saudade mútua de um para o outro. Obrigado por ser uma cúmplice com o qual sempre posso contar e de fazer sacrifícios por mim.

Agradeço ao meu orientador Jorge, por todo o auxílio, suporte e mão amiga sempre estendida, com muita paciência e zelo pelo meu desenvolvimento, sempre à disposição para ajudar a crescer.

Agradeço ao meu companheiro de pesquisa, Lucas, por todo o trabalho que realizamos, trazendo crescimento de um para o outro, com total empenho e dedicação.

Agradeço à toda equipe do LARISA, pela grande amizade formada, todo o auxílio e disponibilidade e oportunidade.

Agradeço também aos meus amigos de UERJ, de mestrado, de vida, de seja onde for, os que fiz nesta última caminhada, os que já caminhavam comigo e todos que acompanharam essa minha trajetória em total expectativa e torcem pelo meu sucesso.

Agradeço também à UERJ, por mais uma oportunidade oferecida e estar sempre de portas abertas.

Peço desculpas a todos no qual me ausentei neste tempo.

É muito melhor lançar-se em busca de conquistas grandiosas,  
mesmo expondo-se ao fracasso,  
do que alinhar-se com os pobres de espírito,  
que nem gozam muito nem sofrem muito,  
porque vivem numa penumbra cinzenta,  
onde não conhecem nem vitória, nem derrota.

*Theodore Roosevelt*

## RESUMO

MORAES, C.R.A. *Detecção de falhas em circuitos eletrônicos através de Comitê de classificadores de classe única utilizando técnicas de redução de dimensão e representações de dissimilaridade*. 2017. 232f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

Esta dissertação apresenta um modelo de detecção de falhas em circuitos analógicos e mistos, realizada a partir de classificadores de classe única, além da representação dos dados de entrada por métodos de extração de assinaturas, redução de dimensão e dissimilaridade. A análise será realizada com diferentes medidas de qualidade com a proposta de verificar qual o melhor modelo para diferentes aplicações.

Palavras-chave: Aprendizado de máquinas; Classe única; Redução de dimensão; Dissimilaridade; Extração de atributos; Representação de dados; Classificação.



## ABSTRACT

MORAES, C.R.A. *Electronic circuit fault detection using one-class classifiers committee with dimension reduction and dissimilarity representation*. 2017. 232f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

This work presents an analog and mixed circuits fault detection model, performed by one-class classifiers, as well as data representation by methods of signature extraction, dimension reduction and dissimilarity. This analysis will be performed with different quality metrics in order to verify the best model for different applications.

Keywords: Machine learning; One class classifier; Dimension reduction; Dissimilarity; Attribute extraction; Data representation; Classification.

## LISTA DE ILUSTRAÇÕES

Figura 1: Método de detecção em múltiplos pontos de resposta.....	24
Figura 2: Área sob a curva ROC (AUC) .....	35
Figura 3: Fluxograma do Sistema Detector de Falhas .....	51
Figura 4: Ilustração simplificada do procedimento de classificação.....	55
Figura 5: Classificadores multiclasse e comitê de classe única .....	56
Figura 6: Ilustração da sobreposição de classes em CCUs .....	57
Figura 7: Esquema de obtenção dos conjuntos de dados .....	60
Figura 8: Circuito Sallen Key.....	64
Figura 9: Desempenho para o Sallen-key com o sinal PRBS .....	66
Figura 10: Curvas de sobreposição, classes 6 e 8, Sallen Key .....	69
Figura 11: Desempenho por representação de dados, classificador Gauss, SK .....	70
Figura 12: Desempenho por representação de dados, classificador knn, SK.....	71
Figura 13: Desempenho por representação de dados, classificador kmeans, SK.....	72
Figura 14: Desempenho por representação de dados, classificador Parzen, SK .....	73
Figura 15: Curva de desempenho para os melhores métodos, circuito Sallen-key .....	74
Figura 16: Desempenho dos classificadores para falhas não previstas com sinal PRBS, Sallen-key.....	78
Figura 17: Desempenho multiclasse x CCUs, Sallen-Key .....	80
Figura 18: Filtro Universal (CTSV) .....	81
Figura 19: Desempenho para o CTSV no PRBS.....	84
Figura 20: Curvas de resposta das classes sobrepostas, CTSV .....	86
Figura 21: Desempenho por representação de dados, classificador Gauss, CTSV ....	88
Figura 22: Desempenho por representação de dados, classificador knn, CTSV .....	90
Figura 23: Desempenho por representação de dados, classificador kmeans, CTSV ..	92
Figura 24: Desempenho por representação de dados, classificador Parzen, CTSV ...	94
Figura 25: Curvas de desempenho dos melhores métodos de representação, CTSV.	95
Figura 26: Representação das respostas do circuito CTSV em dissimilaridade .....	98
Figura 27: Desempenho dos classificadores para falhas não previstas com sinal PRBS, CTSV .....	101
Figura 28: Desempenho multiclasse x CCUs, CTSV.....	104
Figura 29: Circuito Biquad .....	105
Figura 30: Desempenho para o Biquad no PRBS .....	108
Figura 31: Sobreposição de classes, classes 7 e 11, Biquad.....	110

Figura 32: Sobreposição de classes, classes 8 e 12, Biquad.....	111
Figura 33: Desempenho por representação de dados, classificador Gauss, Biquad.	112
Figura 34: Desempenho por representação de dados, classificador knn, Biquad .....	113
Figura 35: Desempenho por representação de dados, classificador kmeans, Biquad .....	115
Figura 36: Desempenho por representação de dados, classificador Parzen, Biquad	117
Figura 37: Curva de desempenho para os melhores métodos, circuito Biquad .....	118
Figura 38: Desempenho dos classificadores para falhas não previstas com sinal PRBS, Biquad .....	122
Figura 39: Desempenho multiclasse x CCUs, Biquad .....	124
Figura 40: Circuito Retificador Não-Linear .....	125
Figura 41: Desempenho para o Retificador não-linear no PRBS .....	127
Figura 42: Sobreposição de classes, classes 5 e 11, NLR.....	129
Figura 43: Sobreposição de classes, classes 6 e 11, NLR.....	130
Figura 44: Sobreposição de classes. Classes 5, 6 e 11, circuito NLR.....	130
Figura 45: Desempenho por representação de dados, classificador Gauss, NLR.....	132
Figura 46: Desempenho por representação de dados, classificador knn, NLR .....	134
Figura 47: Desempenho por representação de dados, classificador kmeans, NLR...	135
Figura 48: Desempenho por representação de dados, classificador Parzen, NLR....	137
Figura 49: Curva de desempenho para os melhores métodos, circuito NLR.....	138
Figura 50: Desempenho dos classificadores para falhas não previstas com sinal PRBS, NLR .....	142
Figura 51: Desempenho multiclasse x CCUs, NLR.....	144

## LISTA DE TABELAS

Tabela 1: Classificadores de densidade e suas funções .....	30
Tabela 2: Tabela auxiliar de descrição da figura 6 .....	57
Tabela 3: Tabela de relação nº de classes X nº componentes .....	63
Tabela 4: Componentes com falha e suas respectivas classes, Sallen-Key .....	65
Tabela 5: Resultado de classificação multiclasse, Sallen-key .....	65
Tabela 6: Distribuição dos dados para o melhor classificador do Sallen-key para a curva PRBS.....	67
Tabela 7: Distribuição dos dados para o melhor classificador na sua melhor representação, Sallen-key .....	75
Tabela 8: Tabela de falhas não previstas, Sallen-Key.....	76
Tabela 9: Resultados do multiclasse para falhas não previstas, Sallen-Key .....	77
Tabela 10: Resultados da classificação dos CCUs no sinal PRBS para falhas não previstas, Sallen-key.....	78
Tabela 11: Componentes com falha e suas respectivas classes, CTSV .....	82
Tabela 12: Resultado de classificação multiclasse, CTSV .....	83
Tabela 13: Distribuição dos dados para o melhor classificador do CTSV para a curva PRBS.....	85
Tabela 14: Distribuição dos dados para o melhor classificador do CTSV no melhor método de representação de dados.....	96
Tabela 15: Representação das classes não previstas, CTSV .....	99
Tabela 16: Resultado de classificação multiclasse para falhas não previstas, CTSV	100
Tabela 17: Resultados para o melhor classificador do CTSV na curva PRBS para falhas não previstas .....	102
Tabela 18: Componentes com falha e suas respectivas classes, Biquad.....	106
Tabela 19: Resultado de classificação multiclasse, Biquad.....	107
Tabela 20: Distribuição dos dados para o melhor classificador do Biquad para a curva PRBS.....	109
Tabela 21: Distribuição dos dados para o melhor classificador do Biquad no melhor método de representação de dados.....	119
Tabela 22: Componentes com falha e suas respectivas classes, Biquad.....	120
Tabela 23: Resultado de classificação multiclasse para falhas não previstas, Biquad .....	121

Tabela 24: Distribuição dos dados para o melhor classificador do Biquad para a curva PRBS.....	122
Tabela 25: Componentes com falha e suas respectivas classes, NLR .....	126
Tabela 26: Resultado de classificação multiclasse, NLR .....	127
Tabela 27: Distribuição dos dados para o melhor classificador do NLR para a curva PRBS.....	128
Tabela 28: Distribuição dos dados para o melhor classificador na melhor sua melhor representação, NLR.....	139
Tabela 29: Componentes com falha e suas respectivas classes para classes não previstas, NLR .....	140
Tabela 30: Resultado de classificação multiclasse para falhas não previstas, NLR..	140
Tabela 31: Resultados para o melhor classificador do CTSV na a curva PRBS para falhas não previstas .....	142

## LISTA DE SIGLAS E ABREVIÇÕES

acc – Acurácia

APCA – Método de aproximação adaptativa constante por partes

Biquad/Biq – Circuito filtro passa-altas ativo Biquad

CCU – Classificador de classe única

CTSV – Circuito filtro universal

Diseuc – Dissimilaridade euclidiana

Dislog – Dissimilaridade lógica

DR – Técnica de redução de dimensão

FN – Elementos falso negativos

FP – Elementos falso positivos

Gauss – Classificador gaussiano

*kmeans* – Classificador das k-médias mais próximas

*knn* – Classificador por vizinhança

*Parzen* – Classificador de janela de Parzen

*knnk* – Classificador multiclasse por vizinhança

LDA – Análise discriminante linear

LPP – Projeções de preservação de Localidade

N – Elementos negativos

NLR – Retificador não-linear

P – Elementos positivos

PAA – Método de aproximação agregada por partes

PCA – Análise das componentes principais

PP – componente passivo e paramétrico

prec – Precisão (*precision*)

SK – Circuito Sallen-key

TP – Elementos positivos verdadeiros

TN – Elementos negativos verdadeiros

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	16
<b>1. DESAFIOS NA DETECÇÃO E DIAGNÓSTICO</b> .....	20
<b>1.1. Conceito Básico</b> .....	21
<b>1.2. Tipos de Falhas e suas Características</b> .....	22
<b>1.3. Etapas de Detecção e Detecção de Falhas a partir de Múltiplos Pontos de Resposta</b> .....	23
<b>2. CLASSIFICADORES DE CLASSE ÚNICA</b> .....	26
<b>2.1. Características de Classificadores de Classe Única</b> .....	27
<b>2.2. Tipos de Classificadores de Classe Única</b> .....	29
2.2.1. <i>Classificadores por estimativa de densidade</i> .....	29
2.2.2. <i>Classificadores por vizinhança</i> .....	30
2.2.3. <i>Classificadores por reconstrução</i> .....	31
<b>2.3. Métricas para avaliação de classificadores de classe única</b> .....	32
<b>3. MÉTODOS DE REPRESENTAÇÃO DE ATRIBUTOS</b> .....	36
<b>3.1. Métodos de Extração de Assinaturas</b> .....	36
3.1.1. <i>Método de Aproximação Agregada por Partes (Piecewise Aggregate Approximation – PAA)</i> .....	37
<b>3.2. Métodos de Redução de Dimensão</b> .....	38
3.2.1. <i>Análise das Componentes Principais (PCA/KPCA)</i> .....	38
3.2.2. <i>Análise Discriminante Linear (LDA/KLDA)</i> .....	42
3.2.3. <i>Projeções de Preservação de Localidade (LPP)</i> .....	44
<b>4. REPRESENTAÇÃO DOS DADOS POR DISSIMILARIDADE</b> .....	47
<b>4.1. Métricas para representação por dissimilaridade</b> .....	48
<b>5. SISTEMA PROPOSTO</b> .....	51
<b>5.1. Excitação e resposta do circuito através de uma entrada PRBS</b> ..	52
<b>5.2. Extração de características</b> .....	53
<b>5.3. Classificação dos dados a partir de um comitê de classe única</b> ..	54
<b>6. ESTUDO DE CASO</b> .....	59
<b>6.1. Circuito Sallen-Key</b> .....	64
6.1.1. <i>Desempenho</i> .....	64
6.1.2. <i>Sobreposição de classes</i> .....	68
6.1.3. <i>Influência da representação de dados na sobreposição</i> .....	69

6.1.4.	<i>Desempenho para a representação de dados</i>	74
6.1.5.	<i>Desempenho para classes não previstas</i>	76
6.1.6.	<i>Comparação com o multiclasse</i>	79
6.2.	<i>Filtro Universal (CTSV)</i>	81
6.2.1.	<i>Desempenho</i>	81
6.2.2.	<i>Sobreposição de classes</i>	85
6.2.3.	<i>Influência da representação de dados na sobreposição</i>	88
6.2.4.	<i>Desempenho para a representação de dados</i>	95
6.2.5.	<i>Desempenho para classes não previstas</i>	99
6.2.6.	<i>Comparação com o multiclasse</i>	103
6.3.	<i>Filtro Biquad</i>	105
6.3.1.	<i>Desempenho</i>	105
6.3.2.	<i>Sobreposição de classes</i>	110
6.3.3.	<i>Influência da representação de dados na sobreposição</i>	111
6.3.4.	<i>Desempenho para a representação de dados</i>	118
6.3.5.	<i>Desempenho para classes não previstas</i>	119
6.3.6.	<i>Comparação com o multiclasse</i>	123
6.4.	<i>Retificador Não-Linear</i>	125
6.4.1.	<i>Desempenho</i>	125
6.4.2.	<i>Sobreposição de classes</i>	129
6.4.3.	<i>Influência da representação de dados na sobreposição</i>	131
6.4.4.	<i>Desempenho para a representação de dados</i>	138
6.4.5.	<i>Desempenho para classes não previstas</i>	139
6.4.6.	<i>Comparação com o multiclasse</i>	143
7.	<b>CONCLUSÃO</b>	145
	<b>REFERÊNCIAS</b>	148
	<b>ANEXO I – TABELA DE RESULTADOS DE CADA CIRCUITO</b>	154
	<b>ANEXO II – CURVAS DE DESEMPENHO PARA TODAS AS DETECÇÕES REALIZADAS</b>	162
	<b>ANEXO III – TABELAS DE CLASSIFICAÇÃO DE FALHAS NÃO PREVISTAS</b>	182
	<b>ANEXO IV – TABELAS DE AJUSTE DOS PARÂMETROS</b>	198
	<b>ANEXO V – TABELA DE PARÂMETROS DE ENTRADA PARA VALIDAÇÃO CRUZADA</b>	200



<b>ANEXO VI – CURVAS DE SOBREPOSIÇÃO DE CLASSES COM REPRESENTAÇÃO DE DADOS.....</b>	<b>201</b>
---	------------

## INTRODUÇÃO

Desenvolver estratégias de teste para detecção e diagnóstico de falhas em circuitos analógicos e mistos é uma tarefa bastante desafiadora, que tem encorajado diversas pesquisas, devido ao aumento do número de aplicações onde estes circuitos são utilizados e seu alto custo de verificação. Muitas áreas, como as de telecomunicações, multimídia e biomédica demandam um bom desempenho para aplicações em circuitos de alta frequência, baixo ruído e baixa potência de operação, que somente podem ser alcançados através de circuitos integrados analógicos e de sinais mistos. Portanto, obter uma estratégia para detectar e diagnosticar falhas nesses circuitos é muito importante (ALBUSTANI, H., 2004). Anteriormente, um circuito integrado era apenas mais um dos componentes de um sistema; no entanto, agora o próprio circuito integrado é o sistema inteiro (Soc. – *System on a chip*). Neste nível de integração, aumentaram-se as dificuldades de se realizar o diagnóstico de tais circuitos, dentre as quais destacam-se: a falta de bons modelos de falhas, falta de um padrão de teste e o aumento da importância das falhas variantes no tempo. Além disso, também são aspectos relevantes: a dificuldade em distinguir falhas a partir de um único sinal de saída, a determinação de classes de defeitos a partir de seus componentes, a dificuldade de se trabalhar com falhas catastróficas, a influência da sensibilidade do componente na resposta de diagnóstico do circuito e a atual restrição de diagnóstico para apenas circuitos com resposta de frequência bem definida. Com isso, a estratégia de testes para detecção e diagnóstico de falhas ainda é severamente dependente de mão-de-obra qualificada e que compreenda as características do circuito, o que faz da detecção e a identificação de falhas ainda um processo interativo, custoso e que consome bastante tempo. Um estudo na área de detecção e diagnóstico (FENTON, ET. AL., 2001) mostrou que, nas últimas décadas, uma boa quantidade de pesquisa em diagnósticos de falhas tem sido direcionada ao desenvolvimento de técnicas que facilitem o diagnóstico que, embora importantes progressos tenham sido obtidos, essas novas tecnologias não têm sido amplamente aceitas. Dessa forma, têm sido conduzidas pesquisas a fim de investigar novos paradigmas e elaborar novos métodos de diagnóstico.

Atualmente, técnicas de inteligência computacional têm sido amplamente usadas em diagnóstico de falhas. Baseadas na construção de modelos ou uso de classificadores, estas técnicas têm seu desempenho afetado pela total dependência da qualidade do modelo utilizado, que torna-se mais difícil de ser obtido com precisão conforme aumenta a sua complexidade. Por outro lado, os classificadores, buscam comportamentos específicos de falhas, onde somente casos restritos apresentam tal comportamento de falha. As falhas ocorridas apresentam na maioria dos casos comportamentos superpostos, o que dificulta uma correta classificação. Além disso, classificadores multiclasse ficam vulneráveis ao analisar padrões de falha não apresentados durante o treinamento. Outro grande problema na detecção de falhas é a classificação com um grande número de entradas, pois, normalmente, a detecção é realizada pela análise da resposta do circuito no tempo, que é um modelo complexo de dados, aumentando a complexidade do classificador e interferindo em sua eficiência.

A fim de tornar possível trabalhar com falhas não previstas no treinamento, classificadores de classe única (TAX, D.M., 2001) podem ser treinados para resolver problemas de classificação binária, onde apenas uma das classes é bem conhecida, a partir da organização de um comitê de classificadores.

Acerca da superposição, uma alternativa seria a observação do circuito em vários pontos de saída. Dessa forma, embora ocorresse a superposição dos sinais de saída do circuito para falhas distintas, as informações adicionais da observação dos outros pontos ajudariam a evitar a superposição. O custo disso seria o aumento do número de entradas do classificador. Neste caso, técnicas de representação dos dados podem ser utilizadas a partir de extração de atributos e redução de dimensão.

Métodos de extração de assinaturas e redução de dimensão são utilizados para lidar com o problema gerado pelo grande número de entradas em sistemas muito complexos de aquisição e análise de dados. Embora suas finalidades sejam iguais, operam distintamente: na extração de assinaturas busca-se obter um novo conjunto de entradas a partir de alguma informação comum a todas as entradas originais. Por exemplo, podemos gerar este novo conjunto de entradas a partir de momentos invariantes do sinal, como a média

e o desvio padrão do sinal original. Já na redução de dimensão, o objetivo é observar os dados em um espaço diferente do original, com dimensionalidade reduzida, capaz de modelar satisfatoriamente as entradas originais do sistema. Por sua vez, a dissimilaridade é o método mais recente para a representação de dados, onde é realizada a troca dos atributos originais por novos atributos que potencializem a diferenciação entre cada elemento de cada classe.

Os métodos mais utilizados de pré-processamento são o PAA (*Piecewise Aggregate Approximation*) (KEOGH, E., PAZZANI, M., 2000), o APCA (*Adaptive Piecewise Constant Approximation*) (KEOGH, E., ET. AL., 2001), a Transformada de Fourier e as transformadas *Wavelet* (SPYRONASIOS, A.D., ET. AL., 2011). Quanto aos métodos de redução de dimensão, são bastante utilizados: PCA/KPCA (*Principal Component Analysis*), LDA (*Linear Discriminant Analysis*) (XIAO, Y., FENG, L., 2012), Preservação de Projeções de Localidade (LPP) e autoespaço do Laplaciano (*Laplacian Eigenmaps*).

Neste trabalho, serão desenvolvidos e estudados vários métodos para detecção de falhas em circuitos analógicos e mistos em geral, baseados em comitês de classificadores de classe única (CCUs). Em (LOMBARDI, A., 2011), a resposta ao impulso do circuito pré-processada com o método de aproximação agregada por partes (PAA) é utilizada como entrada em um comitê de classe única na detecção. Neste trabalho, será utilizado o mesmo método para a extração de assinaturas. Além disso, métodos de redução de dimensão e de dissimilaridade serão estudados, utilizando as técnicas de Análise Discriminante Linear (LDA), Análise de Componentes Principais (PCA) e Preservação de Projeções de Localidade (LPP) para a redução de dimensão e também combinando os métodos de extração de assinaturas e de redução de dimensão. Técnicas de representação em dissimilaridade também serão utilizadas, buscando uma melhor discriminação entre os elementos de cada respectiva classe através de dois métodos distintos de representação por dissimilaridade: a dissimilaridade euclidiana que é a técnica convencional deste método e em uma nova forma de medição desenvolvida neste trabalho, chamada dissimilaridade lógica. Para avaliação do circuito, um sinal pseudoaleatório será utilizado como entrada do circuito, a fim de excitá-lo em

um amplo espectro de frequência na detecção. Além disso, será extraída a resposta de cada circuito em vários pontos de sua topologia e realizada a junção entre cada sinal de resposta, gerando assim uma resposta combinada do circuito, a fim de se resolver o problema da semelhança entre dados de diferentes classes, causa da sobreposição de classes na detecção de falhas nos circuitos. Os circuitos avaliados neste trabalho são: o filtro passa-banda Sallen-Key, o Filtro Universal (CTSV), o Filtro Ativo Passa-alta de 4ª Ordem (Biquad) e o Retificador não-linear.

## 1. DESAFIOS NA DETECÇÃO E DIAGNÓSTICO

O diagnóstico de falhas constitui-se em um grande desafio frente à crescente complexidade dos circuitos analógicos e de sinais mistos. Em primeiro lugar, os circuitos analógicos não possuem um padrão de saída pré-determinado como em circuitos digitais, havendo para cada circuito características próprias de sinais, operações e restrições. Isto explica o pequeno número de modelos de falhas para estes tipos de circuitos, ao contrário que ocorre nos circuitos digitais. O segundo ponto é que, dependendo do tipo de detecção realizado, há dificuldade de se acessar o atributo desejado sem alterar as características do circuito. Por exemplo, existe uma dificuldade de medir correntes sem alterar as conexões originais do circuito (BANDLER, J. W., SALAMA, A. E., 1985). Outra razão para esta dificuldade está no fato de que um circuito analógico tem seus parâmetros definidos a partir dos componentes nele utilizados. Dessa forma, o número de combinações de falhas, ou de comportamentos normais pode ser muito grande, o que torna o uso de métodos determinísticos pouco eficazes.

Além disso, os circuitos analógicos podem apresentar efeitos não lineares e aleatórios, como ruídos, cujos valores dos parâmetros podem sofrer grandes variações e gerar comportamento imprevisível. Isto faz com que os métodos determinísticos sejam ineficientes nestes casos (SLAMANI, M., KAMINSKA, B. 1992). Além disso, a distribuição estatística das falhas normalmente não é conhecida com precisão suficiente, o que dificulta o uso de métodos probabilísticos. Outro desafio é a utilização dos equipamentos de teste automático (*ATE – Automatic Test Equipment*) para a detecção, devido à complexidade de alguns circuitos analógicos e mistos e o grande volume de dados que deve ser armazenado no equipamento de testes para abranger todas as classes de falhas para o circuito. Os mesmos problemas de grande demanda computacional e de dados são observados para o método de dicionário de falhas, que exige um grande poder computacional para a abrangência de todas as classes de falhas (FANNI ET AL., 1999), mesmo considerando o grande avanço tecnológico da computação.

Nas últimas décadas, a pesquisa na área de diagnóstico de falhas concentrou-se em desenvolver ferramentas que simplifiquem o processo de

diagnóstico (FENTON, ET. AL., 2001). As principais ferramentas utilizadas têm sido métodos de inteligência computacional para o diagnóstico. Mesmo tendo ocorrido progressos significativos, estas novas tecnologias ainda não são amplamente aceitas. Tais técnicas são normalmente baseadas na utilização de classificadores para a distinção entre as falhas, ao extrair atributos dos sinais de tensão e corrente do circuito quando submetido à alguma condição específica. Na sequência, submete-se o classificador a um treinamento e, avalia-se o desempenho da classificação na detecção das falhas no circuito. Entretanto, esta abordagem exige que as classes de falha sejam cuidadosamente escolhidas (CZAJA, Z., 2008). Uma escolha incorreta fará com que o classificador multiclasse tenha um desempenho ruim por não ser capaz de indicar as falhas corretamente (LOMBARDI, A., 2011). Além disso, este tipo de classificador só é capaz de indicar classes de falha para as quais ele tenha sido treinado, caso contrário é necessário que se efetue o treinamento da classe desconhecida antes da análise para não haver a classificação incorreta da falha.

### **1.1. Conceito Básico**

O termo *falha* é citado em trabalhos de diversas áreas. Em (WEBER, T.S., 2002) é chamada de falha a causa física de um erro, que precede a existência do defeito, que é o estado de desvio de especificação de um componente ou sistema, no caso da computação.

Para o caso deste trabalho, definimos *falha* como o comportamento anormal ou o defeito em um componente, sistema ou dispositivo que, pode levar a um mau funcionamento. Ou seja, a *falha* é a diminuição parcial ou total da capacidade de um sistema de desempenhar a função para o qual é projetado, por um certo período de tempo ou permanentemente (LOMBARDI, A., 2011).

## **1.2. Tipos de Falhas e suas Características**

Os tipos de falhas que um circuito pode apresentar são bastante abrangentes. As falhas podem ser determinadas pelo seu tipo de desvio, por sua duração, pela quantidade de dispositivos a apresentar falha ao mesmo tempo, à sua observabilidade e à sua distinção.

Quanto ao grau de desvio, uma falha é dita paramétrica quando a sua ocorrência está relacionada a um desvio do parâmetro no tempo, não alterando a topologia do circuito. Quando a falha possui um desvio extremo de parâmetro, ela é então chamada de falha catastrófica. Isto porque este tipo de falha está relacionado à perda de componentes do sistema, alterando a sua estrutura topológica, assim como a sua própria função de transferência (LUO, H., ET. AL., 2012). Por exemplo, em circuitos elétricos, circuito aberto e curto-circuito são falhas catastróficas (DUHAMEL, P., RAULT, J., 1979).

Quanto ao número de falhas simultâneas que ocorrem em um circuito, a falha é dita simples quando sua ocorrência é única, havendo a mudança paramétrica de um único componente. Por outro lado, a falha é dita múltipla quando, em sua ocorrência, há a mudança paramétrica de mais de um componente simultaneamente. Quanto à relação entre falhas, duas ou mais falhas são independentes se não existem relações de causa e efeito entre suas ocorrências, caso contrário, elas são dependentes (LOMBARDI, A., 2011).

Com relação à sua duração, uma falha é dita intermitente quando a sua ocorrência existe por um certo período de tempo aleatório e imprevisível, alternando-se entre o comportamento normal e anormal. Por outro lado, quando a ocorrência da falha é contínua ao longo do tempo, sendo o reparo do componente defeituoso a única solução para a sua interrupção, a falha é dita permanente. O termo transiente é também usado por alguns autores para classificar falhas causadas por mudanças temporárias no ambiente onde o circuito está operando. Por sua vez, o termo incipiente é utilizado para nomear falhas que evoluem gradualmente, tornando-se mais severas ao longo do tempo (MANDERS, E. J., ET. AL., 2000; SILVA, R., 2012).

Por fim, quanto à sua observabilidade, duas ou mais falhas são classificadas como mascaráveis quando suas ocorrências simultâneas ou



progressivas podem compensar seus efeitos, tornando o sistema livre de erros sob certas condições (LOMBARDI, A., 2011). Quando uma falha apresenta um efeito sobressalente à sua ocorrência no circuito, isolada ou simultânea, ela é chamada de falha dominante. As falhas são chamadas de indistintas quando os seus efeitos afetam o circuito de modo que, seus efeitos sejam distinguíveis entre si. Por fim, as falhas inconfundíveis são aquelas cujo efeitos são atribuídos a uma única e determinada condição, podendo ser reveláveis em certas condições (detectáveis), ou não-detectáveis, caso contrário.

### ***1.3. Etapas de Detecção e Detecção de Falhas a partir de Múltiplos Pontos de Resposta***

Várias etapas podem ser consideradas para a detecção de falhas em circuitos. Em (DUHAMEL, P., RAULT, J., 1979), considera-se que a detecção de uma falha em um circuito consiste em três etapas: avaliação, localização e identificação.

Avaliar uma falha em um circuito significa definir se o comportamento do circuito corresponde ou não ao seu comportamento normal. Após a avaliação, a localização é a etapa onde se procura encontrar o componente que está fazendo com que o circuito funcione de forma anormal. Por sua vez, a identificação corresponde a determinar qual o valor que o componente assumiu.

A etapa da localização tem uma acurácia reduzida quando o circuito está sujeito a falhas múltiplas. O circuito é um sistema cujo bom funcionamento normalmente não depende de todos os componentes do mesmo modo, ou seja, alguns componentes são mais importantes para o bom funcionamento do circuito do que outros. Dessa forma, tem-se que a saída do circuito possui maior sensibilidade à variação de alguns componentes específicos. Dois comportamentos então podem acontecer:

- O circuito continuar a funcionar adequadamente para algum componente com falha paramétrica (fora dos limites de tolerância); ou
- Uma falha pode se confundir com outra (superposição de falhas).

O primeiro caso é facilmente resolvido se for considerado que não existe falha no circuito, mesmo que um dado componente esteja fora dos limites de tolerância, o que de fato não é um equívoco visto que, somente existe a falha em um circuito quando existir um comportamento anormal de seu funcionamento.

No segundo caso, a situação torna-se mais complexa, pois na ocorrência de sobreposição, a localização do componente torna-se difícil e imprecisa, principalmente no caso em que o componente defeituoso é aquele de menor sensibilidade. O resultado é que, dependendo da saída a ser observada, a localização precisa de um único componente torna-se inviável e, da mesma forma, a determinação de uma única falha isolada.

Neste trabalho, a alternativa adotada para tentar resolver o problema de sobreposição de falhas é a utilização de entrada de dados do classificador para a saída do circuito em mais de um ponto de teste. Em outras palavras, mais de uma resposta do circuito será obtida em vários pontos de avaliação ao longo de sua arquitetura, e então serão combinadas entre si, formando um grande conjunto de dados de entrada para o classificador. A figura 1 descreve a diferença da extração de resposta em um único ponto, para a extração aqui utilizada, em múltiplos pontos de extração.

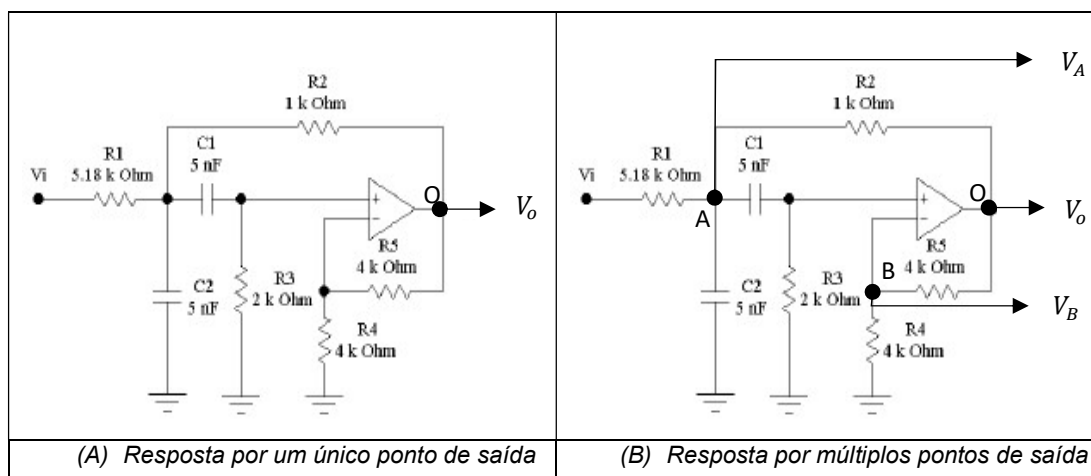


Figura 1: Método de detecção em múltiplos pontos de resposta

Ao se obterem as tensões  $V_A$  e  $V_B$  além de  $V_{out}$ , temos as respectivas respostas do circuito nestes pontos, além de suas funções de transferência  $H_A$ ,  $H_B$ , e  $H_{out}$ , distintas entre si para respectivos pontos A, B e Out. Sendo

cada ponto observado diferentemente no circuito, suas respostas e funções de transferência serão distintas. Espera-se então que cada resposta tenha maior sensibilidade a um determinado componente do sistema. Se isto acontecer, então possivelmente será resolvido o problema de sobreposição de falhas, já que as respostas obtidas entre duas classes de falhas, caso fossem iguais em um dos pontos de observação poderiam ser distintas em outro ponto e, assim, permitiria a correta identificação.

Apesar da possível melhora, esta técnica de extração tem a desvantagem de que aumenta o volume dos dados proporcionalmente ao número de pontos de observação utilizados, gerando um aumento do número de entradas para o classificador. Neste caso, técnicas de representação de dados tornam-se importantes para realizar um tratamento adequado de extração de características e de redução de volume.

## 2. CLASSIFICADORES DE CLASSE ÚNICA

Classificadores de classe única são modelos de classificadores desenvolvidos para resolver um problema especial de classificação, onde somente existe o conhecimento do comportamento de uma das classes, enquanto qualquer comportamento de outra classe é desconhecido (TAX, D.M., 2001). Quando é preciso que um classificador separe um determinado conjunto em classes, muitas vezes o conjunto de dados disponível não possui exemplos em quantidade suficiente para representar de forma satisfatória todas as classes. Considere o seguinte exemplo: um conjunto formado com base no estado de funcionamento de circuitos analógicos e suponha que um robô seja projetado para separar apenas os circuitos em boas condições. O objetivo é determinar quando um circuito está com defeito ou não.

Para realizar esta tarefa, primeiramente, necessitamos definir o que significa o circuito estar defeituoso. Assume-se simplificadamente que defeito significa a incapacidade de realização da tarefa para a qual o circuito é destinado, de forma satisfatória e sem anomalias em sua saída. Considera-se também que os circuitos avaliados podem assumir dois estados possíveis: com ou sem defeito. Até esse ponto não existem desafios por conta da tarefa simples de se determinar se um circuito tem ou não defeito, mas observe agora que, para classificar um circuito como defeituoso ou não, dois conjuntos de informações são necessários: informações sobre como o circuito comporta-se sem defeito e, como comporta-se com defeito. O primeiro caso possui informação abundante, pois o circuito passa quase toda sua vida útil em funcionamento normal. Já no segundo caso, a quantidade de informações disponíveis sobre o circuito com defeito é limitada, pois ocorre com pouca frequência em sistemas bem projetados e normalmente o circuito é substituído ao apresentar algum defeito. Portanto, para que seja possível classificar corretamente o estado (classe) do circuito por meio de classificadores tradicionais, é necessário conhecer todas as classes a serem classificadas. Isto significa, por exemplo, que é necessário conhecer todos os possíveis comportamentos de um circuito defeituoso para classificar o estado do circuito corretamente, o que não é viável.

Agora, suponha que o robô projetado para determinar se um determinado circuito está com defeito ou não seja reprogramado e só precise informar quando o circuito não está com defeito. Logo, tudo o que o robô precisa é das informações já conhecidas sobre o funcionamento normal do circuito. Por exclusão, o que não corresponder ao funcionamento normal, será classificado como comportamento desconhecido. A capacidade de determinar com precisão quando o circuito não está com defeito será mais simples de realizar do que a sua capacidade de dizer se o mesmo tem ou não defeito, pois toda a informação necessária para inferir o estado do circuito está disponível, ao contrário do outro caso. A partir deste conceito, a classificação por classe única é definida. Classificadores de classe única então, são treinados para conhecer uma determinada classe de dados definida como a classe *target*, e rejeitar qualquer informação divergente do conjunto *target*, classificando os dados rejeitados como desconhecidos (*outlier*).

Como no classificador de classe única, toda a informação necessária para a classificação está disponível, decide-se por definir o comportamento da classe conhecida e ignorar qualquer comportamento divergente da classe desejada. Esta é uma opção mais vantajosa do que tentar conhecer o comportamento de todas as classes na detecção de falhas em circuitos.

### **2.1. Características de Classificadores de Classe Única**

Como definido anteriormente e segundo (TAX, D.M., 2001), os classificadores de classe única são desenvolvidos para resolver um problema especial de classificação onde existe uma quantidade adequada de amostras de uma única classe (*target*), enquanto qualquer outra classe (*outlier*) é deficiente ou ausente. Ao se utilizar um classificador de classe única, existem quatro alternativas de saída possível para um determinado elemento (LOMBARDI, A., 2011):

- Verdadeiro positivo: quando o dado classificado como *target* de fato pertence à classe *target*. Logo, a classificação foi correta, não havendo contribuição para o erro;

- Verdadeiro negativo: quando o dado classificado como *outlier*, de fato, não pertence à classe *target*. Neste caso, o classificador também realiza uma classificação correta, não havendo erro de classificação para o elemento;
- Falso positivo: quando o dado classificado como *target*, na verdade, é um dado de *outlier*. Neste caso, há um erro na classificação do elemento;
- Falso negativo: quando o dado classificado como *outlier*, na verdade, pertence à classe *target*. Há então uma rejeição incorreta do elemento pelo classificador, contribuindo para a fração de rejeição do classificador.

Em um classificador de classe única, apenas os objetos da classe *target* são de interesse para classificação e, assim, ele depende apenas de suas amostras de *target* para treinamento. Um bom classificador de classe única deve ser capaz de obter a maior quantidade possível de verdadeiros positivos e negativos, o que proporciona aumento da sua acurácia e deve obter o menor o número de falsos positivos e falsos negativos possível, reduzindo o seu erro de classificação. Durante o seu treinamento, quando um classificador de classe única determina um dado como *target*, este dado contribui para definir a fronteira de decisão do classificador. Como somente o comportamento da classe *target* é conhecido, é possível que se tenham dados *outlier* com comportamento próximo à fronteira de decisão, aumentando as chances de que um elemento *target* seja classificado como *outlier* (falso negativo) ou que um elemento *outlier* seja classificado como *target* (falso positivo), acarretando em erro do classificador. Para minimizar o erro, os classificadores de classe única dispõem de um parâmetro, chamado de fração de rejeição, que pré-determina a parcela de falso negativo que o classificador propositalmente terá. Ou seja, determinar-se a priori uma fração de elementos *target* que serão considerados como *outliers*, para que seja possível estabelecer uma fronteira de decisão que minimize o erro de detecção de *outliers* enquanto a taxa de falso negativo é mantida.

## 2.2. Tipos de Classificadores de Classe Única

Quanto aos tipos de classificadores de classe única, existem três diferentes categorias que serão abordadas neste trabalho: classificadores por estimativa de densidade, classificadores por determinação de vizinhança e classificadores de reconstrução (YI, Z., ET. AL., 2008).

### 2.2.1. Classificadores por estimativa de densidade:

Estes são os tipos de classificadores são caracterizados por serem simples e apresentarem bom desempenho nos casos em que o conjunto de dados de *target* é quantitativamente satisfatório e pode ser bem modelado a partir do modelo de probabilidade utilizado (LOMBARDI, A., 2011). São também os tipos de classificadores mais utilizados. Alguns exemplos de classe única por estimativa de densidade são os classificadores Gaussianos, mistura de gaussianas (KALPANA, P., GUNAVATHI, K., 2007) e *Parzen* (TAX, D.M., 2001).

Um classificador de estimativa de densidade funciona da seguinte forma: considere  $X$  como sendo o conjunto de dados de treinamento. O conjunto será tratado como uma variável aleatória com distribuição  $f$ , onde:

$$\begin{aligned} f: \mathbb{R} &\rightarrow Im(f) \\ X &\rightarrow f(X) \end{aligned} \quad (1)$$

Sendo a taxa de rejeição  $thr$  um parâmetro do classificador, a imagem de  $f$  será uma medida de distância cujo valor tende ao mínimo para os dados de *target*, e a fronteira de rejeição calculada no treinamento e ajustada pelos parâmetros internos do classificador, que são normalmente a média e a variância para esta classe de classificadores. A partir da taxa de rejeição, a fronteira de classificação será um valor  $\theta$  de erro, obtido durante o treinamento do classificador. A classificação dos dados  $h(X)$  é então definida como:

$$h(X) = \begin{cases} target, & se f(X) \geq \theta \\ outlier, & se f(X) < \theta \end{cases} \quad (2)$$

A função de densidade de probabilidade  $f(X)$  depende do tipo de classificador utilizado. A tabela 1 mostra as funções  $f$  para diferentes classificadores de densidade de probabilidade e a região de imagem da função (KALPANA, P., GUNAVATHI, K., 2007; TAX, D.M., 2001):

Tabela 1: Classificadores de densidade e suas funções:

Classificador	função $f(X)$	Região de $Im(f)$
Gaussiano	$(X - \mu)^T \Sigma^{-1} (X - \mu)$	$[0, +\infty)$
Parzen	$\frac{1}{N(\sqrt{2\pi}h)^D} \sum_{i=1}^N e^{\left[-\frac{1}{2} \left(\frac{\ X - X_i\ }{h}\right)^2\right]}$	$[0, +\infty)$

### 2.2.2. Classificadores por vizinhança

Esta classe de classificadores determina a qualidade dos dados em função da sua proximidade dos dados com os dados da classe *target*. Assim, não é necessário modelar a distribuição de densidade dos dados. Essa característica torna este classificador interessante quando a amostra de dados é insuficiente e a distribuição de densidade é desconhecida. O *knn* é o método mais utilizado deste tipo de classificador. No *knn*, admita que  $X$  é o conjunto de dados e  $x_i$  é um determinado dado deste conjunto para o qual se deseja determinar se é *target* ou *outlier*. Para tal fim, calcula-se a distância  $d_{ik}$  mínima, sendo  $d_{ik}$  a distância entre o dado  $x_i$  e todas as possíveis combinações de  $k$  pontos até  $x_i$ :

$$d_{ik} = \min_{k=1, \dots, n} \|x_i - x_k\| \quad (3)$$

Como  $d_{ik}$  é a distância mínima entre o dado  $x_i$  e todos os pontos tomados em  $k$  combinações, o resultado é que o valor mínimo de  $d_{ik}$  será



obtido somente para os  $k$ -pontos de distância mínima para  $x_i$ . Resumindo,  $d_{ik}$  será mínimo se, e somente se, os  $k$  pontos distantes de  $x_i$  sejam os pontos mais próximos do mesmo. Esses pontos são chamados de  $k$ -vizinhos de  $x_i$ .

Durante o treinamento do knn para classe única, sendo  $X$  o conjunto de dados contendo  $T$  dados de *target*, é obtida a máxima distância média possível entre todos os dados de *target* para os seus  $k$ -vizinhos mais próximos ( $D$ ). Este parâmetro é utilizado como taxa de rejeição do classificador de classe única. Assim, ao avaliar um dado pelo knn obtemos a seguinte função de decisão:

$$h(X) = \begin{cases} target, se d_{ik} \leq D \\ outlier, se d_{ik} > D \end{cases} \quad (4)$$

Observe que a função de transformação é a norma-2 de subconjuntos de  $k+1$  elementos do conjunto de *target* ao invés de uma função de densidade de probabilidade.

### 2.2.3. Classificadores por reconstrução

O método de classificação por reconstrução consiste em modelar o processo de geração dos dados, antes da separação das classes. Quando o modelo de dados é construído, o menor erro médio de reconstrução indica qual objeto se encaixa melhor no modelo, sendo altamente provável que ele seja da classe *target* (LOMBARDI, A., 2011). Este tipo de classificador é robusto para generalização e ruim quando o tamanho de amostra é escasso ou quando os modelos não conseguem ajustar bem aos dados (LOMBARDI, A., 2011). Deste tipo de classificador, o mais utilizado é o *kmeans* em que, a determinação das classes pode ser obtida segundo as equações 5 e 6:

$$h(x_i) = \begin{cases} 1, se \varepsilon_{x_i} \leq \varepsilon_{max} (target) \\ 0, se \varepsilon_{x_i} > \varepsilon_{max} (outlier) \end{cases} \quad (5)$$

O erro mínimo entre o ponto e a classe é dado pela distância entre o ponto e o centroide do conjunto:

$$\varepsilon_{x_i} = \sum_i \min_k \|x_i - \mu\|^2 \quad (6)$$

O erro máximo é o limiar referente ao centro do conjunto de dados de *target* e é determinado pela equação abaixo:

$$\varepsilon_{max} = \|\mu \pm x_{target}\|^2 \quad (7)$$

Onde:

- $\mu$  = valor médio entre os pontos da classe;
- $x_{target}$  = elemento pertinente ao conjunto de dados da classe-alvo.

### **2.3. Métricas para avaliação de classificadores de classe única**

Conforme dito anteriormente, a avaliação de um classificador de classe única é baseada no tipo de resposta que ele pode obter para um determinado dado. Existem quatro possíveis respostas que um classificador pode atribuir a um dado: positivo verdadeiro, negativo verdadeiro, falso positivo e falso negativo. Nos dois primeiros casos, a avaliação da resposta é obtida através do ajuste do classificador durante seu treinamento e a sua comparação com os dados de teste. Já para os dois últimos casos de resposta, é necessário conhecer de antemão a taxa de falsos positivos e de falsos negativos do classificador (TAX, D.M., 2001). Como o classificador não tem conhecimento sobre a qualidade do dado até o treinamento, ele sozinho não é capaz de estimar o seu próprio erro. Por isso, a fração de rejeição é utilizada, pré-determinando a mínima taxa de falsos negativos a ser obtida e gerando assim a fronteira de decisão do classificador.

Uma vez determinado o modelo, a sua avaliação pode ser obtida a partir da taxa de falsos positivos e negativos para um conjunto de teste (LOMBARDI, A., 2011). Por sua vez, a visualização das taxas de acertos e erros pode ser feita por meio de uma matriz de confusão:

$$Confmat = \begin{array}{c|cc} & \text{Real} & \text{Estimado} \\ \hline & & P & N \\ \hline P & TP & FN \\ N & FP & TN \end{array} \quad (8)$$

A matriz de confusão é uma representação ilustrativa de desempenho de um classificador. Em um classificador de classe única, ela é uma matriz 2 x 2, onde a diagonal principal representa os acertos do classificador enquanto a diagonal secundária representa os erros. Para um bom classificador de classe única, a matriz de confusão tende a ser uma matriz 2 x 2 diagonal.

A partir da matriz de confusão, são estabelecidas as relações métricas para avaliação do desempenho do classificador (POWERS, D.M., 2011).

- *Precision*: é a métrica utilizada para identificar a porcentagem de todos os objetos classificados como *target* que realmente pertencem a esta classe.

$$precision = \frac{TP}{TP + FP} \quad (9)$$

- *Recall* (Sensibilidade): informa a porcentagem dos objetos que são verdadeiramente da classe *target* que foram corretamente identificados pelo classificador.

$$recall = \frac{TP}{TP + FN} \quad (10)$$

- *Acurácia*: é uma métrica de observação global dos dados. Ela é utilizada para determinar a capacidade de o classificador identificar corretamente qualquer dado a ele submetido. Em outras palavras, a

acurácia é a probabilidade de que o classificador acerte em sua classificação:

$$acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

- F1 (média harmônica de precisão e sensibilidade): é uma medida utilizada para determinar o desempenho de um classificador. Considerando apenas o *precision* e o *recall*, o F1 é obtido através da média ponderada entre essas duas medidas, de modo que,  $F1 = 1$  é máximo quando ambas as medidas são máximas ( $Prec. = Recall = 1$ ) e  $F1 = 0$  é mínimo quando ambas as medidas são mínimas ( $Prec. = Recall = 0$ ). Como as medidas de *precision* e *recall* estão relacionadas à capacidade de identificar os dados de classe corretamente, F1 consequentemente avalia a capacidade de que o classificador acerte.

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \quad (12)$$

- Taxa de falso positivo (*Fall-out*): esta é uma medida para a avaliação do erro do classificador. Como o próprio nome sugere, ela é baseada na taxa de alarmes falsos que o classificador apresentou.

$$fallout = \frac{FP}{FP + TN} \quad (13)$$

- Taxa de falso negativo (Erro de rejeição): esta medida está relacionada à taxa de dados incorretamente rejeitados. Como a taxa mínima de falsos negativos é pré-determinada pela fração de rejeição, para um bom classificador, o erro de rejeição se aproxima da fração de rejeição.

$$miss = \frac{FN}{TP + FN} \quad (14)$$

- Área sob a curva ROC (LOMBARDI, A., 2011): ROC significa curva característica de operação do receptor (*receiver operating characteristic curve*), a ROC é uma curva em função da relação entre a taxa de positivos verdadeiros (*TP*) e falsos positivos (*FP*). A figura 2 apresenta um exemplo de uma curva ROC:

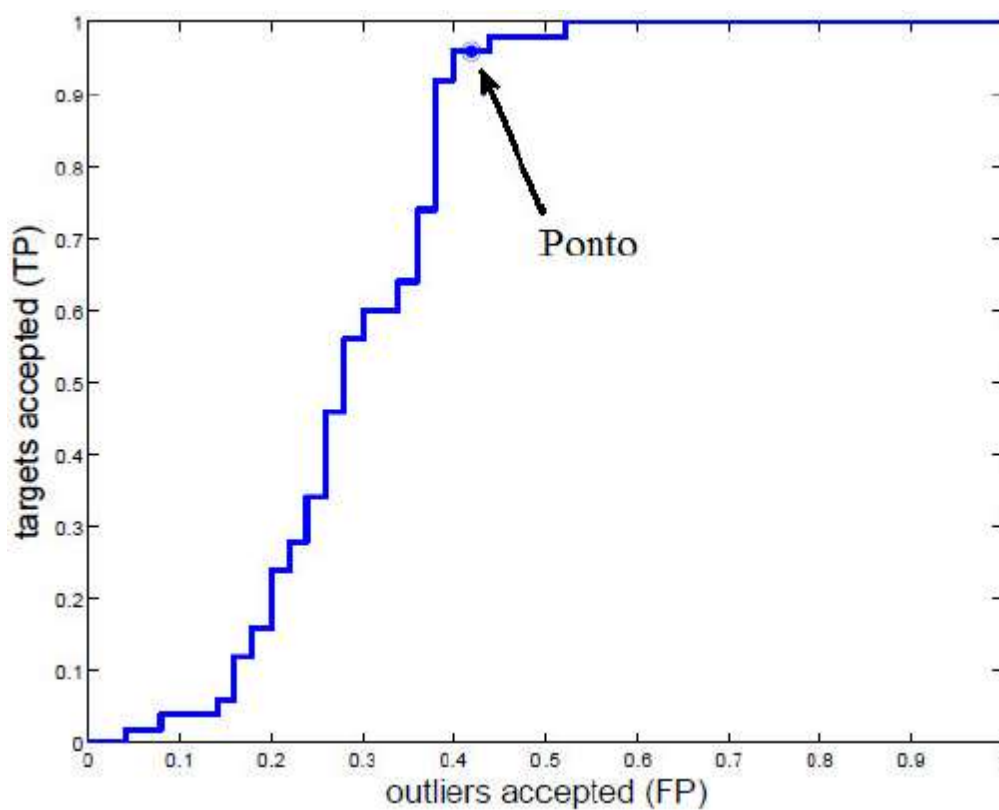


Figura 2: Área sob a curva ROC (AUC)

A abcissa representa a taxa de falsos positivos obtida pelo classificador enquanto a ordenada representa a taxa de positivos verdadeiros obtidos. O limiar de desempenho é determinado pela equação:  $recall = 1 - fall-out$ . Para um bom classificador, a curva ROC cresce até o seu valor máximo antes da fronteira do limiar.

### 3. MÉTODOS DE REPRESENTAÇÃO DE ATRIBUTOS

Uma das grandes dificuldades encontradas na implementação de algoritmos de aprendizado de máquinas está em analisar conjuntos de dados de grande dimensão. O problema é que um dado com mais informação exige um classificador mais complexo para analisá-lo, e um classificador mais complexo demanda um conjunto maior de dados para uma boa generalização. Nestes casos, técnicas que reduzam o número de atributos mas mantenham as informações mais relevantes são muito úteis em algoritmos de aprendizado de máquinas em geral, e em particular para os classificadores de classe única. Dentro do estudo de métodos de representação de dados, existem duas principais categorias distintas de realização dessa tarefa: métodos de extração de assinaturas e técnicas de redução de dimensão. Para este trabalho, métodos de extração de atributos de ambas categorias serão utilizados. Abaixo uma apresentação sucinta sobre cada tipo de método será realizada.

#### 3.1. Métodos de Extração de Assinaturas

Estes métodos realizam a redução do tamanho dos dados a partir da extração de um ou mais atributos do conjunto de dados original, sendo frequentemente utilizado como um método de pré-processamento. O classificador não analisa o conjunto de dados original, mas sim um novo conjunto gerado a partir dos atributos desejados do mesmo. O atributo pode ser qualquer característica que dê alguma “identidade” ao dado observado. Por exemplo, pode-se usar uma avaliação estatística, como: média, moda ou frequência, entre outros. Alguns dos métodos de pré-processamento mais estudados são: método de Aproximação agregada por partes (PAA – *Piecewise Aggregate Approximation*) (KEOGH, E., PAZZANI, M., 2000), Método de aproximação Constante Adaptativa (APCA – *Adaptive Piecewise Constant Approximation*) (KEOGH, E., ET. AL., 2001), o método de decomposição em componentes de Fourier (*Fourier Transform*) e o método de *Wavelets* (*Wavelets Transformation*) (SPYRONASIOS, A.D., ET. AL., 2011).

### 3.1.1. Método de Aproximação Agregada por Partes (*Piecewise Aggregate Approximation – PAA*)

O método de Aproximação Agregada por Partes (*Piecewise Aggregate Approximation – PAA*) é um método de redução de dados a partir da extração de atributos, que tem mostrado resultados bastante satisfatórios em aplicações de classificação de séries temporais. Além do bom desempenho, o PAA possui também as seguintes características: fácil implementação e compreensão, baixa demanda de processamento para ser implementado e resultados iguais ou superiores aos de outros métodos (KEOGH, E., PAZZANI, M., 2000).

Suponha que o sinal  $X = x_1, \dots, x_n$ , represente uma série temporal. Pode-se obter uma nova representação em um espaço  $N$ ,  $N \leq n$ , por  $Z = z_1, \dots, z_N$ , onde cada elemento é calculado conforme a equação (15) abaixo:

$$z_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j \quad (15)$$

Logo, cada elemento  $z_i$  corresponde ao valor médio do  $i$ -ésimo intervalo dos  $N$  intervalos da série temporal.

Visto que o número de amostras é reduzido a um valor fixo, a redução realizada pelo PAA depende apenas de seus parâmetros de entrada. Entretanto, para que esta representação ainda mantenha a informação relevante do sinal original, o seu erro de reconstrução deve ser baixo. Assim, existe um compromisso entre a redução do tamanho do sinal e a manutenção da informação presente, o que deve ser levado em conta na determinação também dos parâmetros de entrada.

A redução de dados depende unicamente do parâmetro de número de segmentos utilizado pelo PAA. O erro de reconstrução será proporcionalmente menor quanto mais próximo de  $n$  for o número de segmentos  $N$  do conjunto gerado pelo PAA. Para o problema de detecção de

falhas, é possível determinar o valor do número de segmentos a partir do qual o erro de reconstrução deixa de diminuir de forma significativa com o aumento do número de segmentos (AMARAL, J., 2011).

### **3.2. Métodos de Redução de Dimensão**

Este tipo de abordagem para a representação dos dados não busca extrair atributos específicos dos dados, como no pré-processamento, e sim transformar os dados, de forma que a nova representação tenha menor dimensão, mas ainda retenha informação relevante. Dentre as técnicas que são utilizadas, a mudança do espaço de observação dos dados é a mais utilizada. As técnicas de redução são partes fundamentais na compressão de dados e no aprendizado de máquinas. Inúmeras técnicas de redução de dimensão poderiam ser mencionadas aqui. As mais utilizadas e conhecidas atualmente são: Análise das componentes principais (*Principal Component Analysis – PCA*) (XIAO, Y., HE, Y., 2011; SCHÖLKOPF, B., ET. AL., 1997), Análise Discriminante Linear (*Linear Discriminant Analysis – LDA*) (MARTINEZ, A. M., YOU, D., ET. AL., 2011; PARK, C. H., PARK, H., 2005), Mapa de Autovalores do Laplaciano (*Laplacian Eigenmap*) (BELKIN, M., NIYOGI, P., 2003), Preservação de Projeções Locais (*Locality Preserving Projections – LPP*) (HE, X., NIYOGI, P., 2003), Decomposição em Valores Singulares (*Singular Value Decomposition – SVD*) (PARK, C. H., PARK, H., 2005), etc. Neste trabalho, foram utilizadas as seguintes técnicas:

#### **3.2.1. Análise das Componentes Principais (PCA/KPCA):**

Nesta técnica, efetua-se a observação do conjunto de dados em um novo espaço, onde este novo conjunto de dados não é linearmente correlacionado. Este novo espaço é definido a partir dos autovalores e autovetores da matriz de covariância dos dados.

Definição: Seja o momento de amostras de treinamento um mapeamento do conjunto tal que:



$$\begin{aligned} \mathbb{R}^l &\rightarrow \mathbb{R}^l \\ x &\rightarrow \varphi(x) \end{aligned} \quad (16)$$

Também se supõe que:

$$\sum_{k=1}^l \varphi(x_k) = 0 \quad (17)$$

Fazendo a covariância das funções no espaço mapeado em  $x$ , obtém-se:

$$\bar{C} = \frac{1}{n} \sum_{i=1}^n \varphi(x_i) \cdot \varphi(x_i)^T \quad (18)$$

Temos então que  $\bar{C}$  é a matriz de covariância de  $\varphi(x)$ :

$$\bar{C} = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix}, c_{ij} = \varphi(x_i) \cdot \varphi(x_j)^T \quad (19)$$

Para obter o espaço mínimo, é preciso encontrar os autovalores e autovetores de  $\bar{C}$ , tal que:

$$\begin{cases} \lambda v = \bar{C} v \\ \lambda \in F \setminus \{0\} \end{cases} \quad (20)$$

Utilizando a definição da covariância no espaço mapeado  $\varphi(x)$ , temos que:

$$\lambda \cdot v = \bar{C} \cdot v = \left[ \frac{1}{n} \sum_{i=1}^n \varphi(x_i) \cdot \varphi(x_i)^T \right] \cdot v \quad (21)$$

O conjunto da interseção de todas as soluções de  $v$  está no espaço nulo de  $\varphi(x)$ . Assim, podemos considerar a equivalência do sistema:

$$\lambda.(\varphi(x_k).v) = (\varphi(x_k).\bar{C}.v) \quad (22)$$

Dessa forma, os autovetores da equação (22) geram um autoespaço, linearmente independente de  $\lambda\{\varphi(x)\}$ .

Podemos escrever  $v$  como a combinação linear das componentes de  $\varphi(x)$ :

$$v = \sum_{i=1}^n \alpha_i. \varphi(x_i) \quad (23)$$

Definindo conseqüentemente, uma matriz  $K$ , tal que:

$$k_{ij} = \varphi(x_i). \varphi(x_j) \quad (24)$$

Substituindo-se na equação (22) a forma  $v$ , da equação (23) e o produto  $\bar{C}.v$ , da equação (21), obtém-se:

$$\lambda. \left( \varphi(x_k). \sum_{i=1}^n \alpha_i. \varphi(x_i) \right) = \left( \varphi(x_k). \left[ \frac{1}{n} \sum_{i=1}^n \varphi(x_i). \varphi(x_i)^T \right]. \sum_{i=1}^n \alpha_i. \varphi(x_i) \right) \quad (25)$$

$$\lambda. \left( \sum_{i=1}^n \alpha_i. \varphi(x_i). \varphi(x_k) \right) = \left( \left[ \frac{1}{n} \sum_{i=1}^n \varphi(x_i). \varphi(x_i)^T \right]. \sum_{i=1}^n \alpha_i. \varphi(x_i). \varphi(x_k) \right) \quad (26)$$

Por sua vez, a equação (26) pode ser escrita de forma compacta como:

$$\lambda K \alpha = \frac{1}{n} K^2 \alpha \quad (27)$$

Multiplicando à esquerda na equação (27)  $nK^{-1}$  em ambos os lados, obtém-se:

$$n\lambda\alpha = K\alpha \quad (28)$$

Onde:

- $K$  = matriz Kernel;
- $\alpha$  = vetor coluna dos coeficientes das componentes de  $v$ ;

O vetor dos coeficientes  $\alpha$  é chamado de *vetor das componentes principais*. Ele contém as informações que definem o espaço  $F \rightarrow \varphi(x)$ . Da equação (28) é possível mostrar que qualquer solução adicional da equação não gerará diferença na expansão, podendo assim ser desprezada. Os vetores das componentes principais definem o espaço vetorial que descreve  $F$  de maior dimensão. Portanto, qualquer adição de vetores será linearmente dependente dos vetores gerados pelos autovalores, não adicionando qualquer informação além daquela já existente.

Se, no caso anterior, for efetuada a normalização dos vetores de coeficientes dos autovalores não nulos, obtém-se uma normalização aos vetores  $Im(F)$ , descrita pela equação (29), a qual é obtida a partir das equações anteriores (21), (22) e (28):

$$\sum_{\substack{i,j=1 \\ i \neq j}}^n \alpha_i^k \cdot \alpha_j^k \cdot (\varphi(x_i) \cdot \varphi(x_j)) = (\alpha^k \cdot K \alpha^k) = \lambda_k (\alpha^k \cdot K \alpha^k) = 1 \quad (29)$$

Para a determinação das componentes do vetor de coeficientes, projeta-se a imagem de um ponto de teste da função nos autovetores em  $F$ :

$$(v_k \cdot \varphi(x_j)) = \sum_{i=1}^n \alpha_i^k \cdot (\varphi(x_i) \cdot \varphi(x_j)) \quad (30)$$

A obtenção das componentes principais pode ser realizada sem a necessidade de se conhecer a função  $\varphi(x)$ , desde que o seu produto interno seja conhecido. Isso leva às componentes Kernel:

$$\left( v_k \cdot \varphi(x_j) \right) = \sum_{i=1}^n \alpha_i^k \cdot k_{ij} \quad (31)$$

Dependendo da escolha da Kernel, é possível mostrar que existe uma função  $\varphi$  gerada por algum produto com  $F$ , de tal forma que:

$$K = \varphi \cdot \varphi_F \quad (32)$$

Para  $K = (x \cdot y)$ , temos um Kernel linear e assim a função PCA definida. No caso do KPCA, o Kernel não é linear e passa-se a trabalhar com uma transformação não-linear no espaço observado, elevando a sua dimensão. A não linearidade dos dados passa a ser observada de forma linearmente independente neste espaço transformado, quando necessário.

### 3.2.2. *Análise Discriminante Linear (LDA/KLDA)*

No caso do LDA, assim como nos outros métodos, o espaço original é projetado para outro espaço de melhor observação. No entanto, há uma diferença crucial: enquanto nos métodos PCA o conjunto original é transportado para um autoespaço gerado pelos autovetores da matriz de covariância, de tal modo que, os dados não estejam linearmente correlacionados. No LDA busca-se o espaço, sendo ele autoespaço gerado do espaço original ou não (normalmente não é), cuja projeção do conjunto original para este espaço forneça o maior grau de separabilidade entre as classes distintas que o mesmo possui, este espaço de maior separação é chamado de discriminante.

Para a obtenção do discriminante é utilizado o critério definido por meio da equação (33), conhecida como critério de Fisher, onde a relação entre o determinante da matriz inter-classes e intra-classes é maximizada.

$$F_c = \max \frac{|S_b|}{|S_w|} \quad (33)$$

Ao satisfazer o critério de Fisher, o conjunto é projetado para o espaço discriminante linearmente independente e os dados serão separados por uma função linear. A base vetorial que maximiza o critério de Fisher pode ser determinada resolvendo o problema de autovalores e autovetores dado pela equação 34:

$$(S_w^{-1} \cdot S_b) \Phi = \Lambda \Phi \quad (34)$$

Onde:

- $S_w$  = matriz intra-classes;
- $S_b$  = matriz inter-classes;
- $\Phi$  = vetor de base do conjunto discriminante;
- $\Lambda$  = matriz dos autovalores do conjunto discriminante.

A matriz  $S_w$  representa a relação entre a quantidade de informação disponível e de informação necessária para discriminar o conjunto. A obtenção do discriminante é dada pela seguinte condição:

$$\det(S_w) \neq 0 \quad (35)$$

Caso a matriz  $S_w$  seja singular, ou seja, a quantidade de amostras de exemplo seja menor que a quantidade de atributos que definem o conjunto, a matriz  $S_w$  não será invertível e a solução da equação não poderá ser obtida diretamente. Apesar disso, existem várias técnicas para se obter uma resposta para a equação: o uso de um subespaço PCA (SWETS & WENG, 1996) ou a regularização direta da matriz sem passar por um subespaço PCA. Em (THOMAS, G., FEITOSA, 2005) propuseram a utilização de uma função

densidade de probabilidade que maximize a entropia do sistema através das médias e das matrizes de covariância de  $S_w$  disponíveis nas amostras, estimando as informações indisponíveis originalmente. O método é conhecido como Análise Discriminante Linear de máxima incerteza (*Maximum uncertainty Linear Discriminant Analysis – MLDA*). Outra possibilidade é a solução do problema de singularidade de  $S_w$  obtendo a sua pseudo-inversa  $S_w^\dagger$  por decomposição em valores singulares (*Singular Values Decomposition – SVD*).

### 3.2.3. Projeções de Preservação de Localidade (LPP)

O LPP aparece como uma alternativa ao método de Análise de Componentes Principais, tradicionalmente conhecido como um dos métodos mais eficazes para o tratamento de dados quanto à redução de dimensão. Enquanto no clássico PCA os dados são projetados em um subespaço onde os novos atributos estão ordenados segundo sua variância, o LPP busca a redução de dimensão preservando de forma ótima a estrutura de vizinhança do conjunto de dados (HE, X., NIYOGI, P., 2003).

Suponha um conjunto de dados  $X$ , de dimensão  $n$ . A técnica de redução prevê que exista uma matriz de transformação  $A$  que leve o conjunto  $X$  para um novo espaço  $Y$ , de dimensão  $k$ , tal que:

$$\begin{cases} Y = A^T \cdot X \\ k \ll n \end{cases} \quad (36)$$

Quando  $X$  é um conjunto não-linear, o mapeamento de  $X$  para  $Y$  a partir da matriz  $A$  não é possível para  $A$  linear. No caso do LPP, a matriz de transformação  $A$  é obtida ao se efetuar uma aproximação linear de um mapa de autovalores do Laplaciano de  $X$  ( $\nabla^2 X$ ).

Primeiramente, um gráfico de adjacência para todos os pares de elementos  $x_i$  e  $x_j$  do conjunto  $X$  é obtido, utilizando duas métricas de vizinhança: vizinhança- $\epsilon$  e vizinhos mais próximos. O nível de adjacência entre os dados determinará a métrica utilizada para definir os pesos dos elementos. Na sequência, é obtida uma matriz esparsa  $W$ , de tamanho  $n \times n$ ,

sendo  $n$  a dimensão do conjunto  $X$  que representa os pesos do nível de adjacência entre os elementos de  $X$ . Cada elemento  $w_{ij}$  pode ser visto então como o nível de proximidade entre os elementos  $x_i$  e  $x_j$ , e será definido pela função:

$$w_{ij} = \begin{cases} e^{-\frac{\|x_i - x_j\|^2}{t}}, & \text{se } \|x_i - x_j\|^2 < \varepsilon \\ 1, & \text{se } \|x_i - x_p\|^2 \leq \|x_j - x_p\|^2, \forall p = 1, \dots, k \\ 0, & \text{c. c.} \end{cases} \quad (37)$$

Existem três principais alternativas de métrica de peso que são utilizadas. No primeiro caso, a métrica de peso é a regra da vizinhança em  $\varepsilon$  ( $\varepsilon$ -neighborhood), definida através de uma hiperesfera de raio  $\varepsilon$  como região de vizinhança. No segundo caso, é utilizada como métrica o método dos  $k$  vizinhos mais próximos ( $knn$ ), onde a vizinhança é definida pelo número  $k$  de vizinhos entre os dados analisados. O terceiro caso representa a não adjacência entre os dados  $x_i$  e  $x_j$ . Uma vez obtida a matriz de pesos  $W$ , o mapa dos autovalores e autovetores de  $X$  é obtido por meio da equação de decomposição:

$$XLX^T \cdot \mathbf{v} = XAX^T \cdot \mathbf{v} \quad (38)$$

Onde:

$$A = \lambda \cdot D, \lambda \in \mathbb{R}$$

Sendo  $D$  a matriz diagonal cujo valores não-nulos representam a soma numérica entre os elementos das colunas de  $W$ , e  $L$  a matriz laplaciana definida pela equação (39):

$$L = D - W \quad (39)$$

A solução do problema de autovalores de  $X$  é então obtida, o que nos fornece as componentes do autovetor  $v$  correspondente, para cada autovalor de  $X$ . Com isso, obtemos o autoespaço gerado de  $X$ .

Sendo  $\{\lambda\} = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$  o conjunto de todos os autovalores de  $X$ , o número de autovalores de  $X$  é  $k$ . Consequentemente, o conjunto de todos os autovetores gerados  $\{v\} = \{v_1, v_2, \dots, v_k\}$  terá dimensão de  $n \times k$ , sendo  $n$  o número de componentes do vetor definido pelo número de elementos de amostras  $x_i$ . Fazendo então do autoespaço gerado a matriz de transformação do conjunto:

$$A = [v_1 \quad v_2 \quad \dots \quad v_k] = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1k} \\ v_{21} & v_{22} & \dots & v_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nk} \end{bmatrix} \quad (40)$$

Sendo:

$v_{ij}$  = i-ésimo elemento do j-ésimo vetor  $v_j$

$i = 1, 2, \dots, n$  e  $j = 1, 2, \dots, k$

Conhecendo a matriz de transformação,  $Y$  é obtido por meio da equação, com a dimensão  $k \times c$ :

$$\begin{cases} Y = A^T \cdot X \\ \dim(Y) = k \times c \end{cases} \quad (41)$$



#### 4. REPRESENTAÇÃO DOS DADOS POR DISSIMILARIDADE

A representação por dissimilaridade é uma alternativa a utilização de atributos. A dissimilaridade entre um par de objetos constitui uma forma natural de representação. Ela pode ser considerada uma forma de representação mais completa que o uso de atributos (GRILL-SPECTOR, K., ET. AL., 1999). Espaços vetoriais definidos por dissimilaridades entre pares de objetos tais como imagens e séries temporais, representam uma abordagem inteligente para diminuir a separação existente entre o reconhecimento de padrões estrutural e o estatístico. Além disso, é importante mencionar que a dissimilaridade também pode ser usada em conjunto com os atributos (DUIN, R.P., PEKALSKA, E., 2009). Também é possível adotar uma opção das abordagens com *kernels*, que são baseadas em similaridades. Medidas de dissimilaridade são mais universais do que *kernels*, pois os espaços vetoriais baseados em *kernels* têm que atender às *condições de Mercer* (SCHOLKOPF, B., SMOLA, A., 2002; SHAWE-TAYLOR, J., CRISTIANINI, N., 2004). Tais condições devem ser atendidas para que a implementação de classificadores, tais como as máquinas de vetores de suporte, seja possível nos espaços de *kernels* relacionados.

A abordagem por dissimilaridade tem a vantagem de que qualquer medida pode ser usada com qualquer classificador implementado em espaços vetoriais. Outro benefício é que a representação por dissimilaridade permite que o conhecimento do especialista seja usado de forma diferente. Ao invés de indicar atributos, o especialista pode fornecer uma medida de similaridade entre cada par de objetos. Naturalmente, quando os atributos estão acessíveis, a distância entre vetores de atributos pode ser usada como uma medida de dissimilaridade. Entretanto, outras medidas comparando objetos inteiros podem ser consideradas, como será feito neste trabalho, em que utiliza-se duas métricas diferentes para dissimilaridade: a distância euclidiana tradicional e a distância lógica, método de avaliação especialista desenvolvido neste trabalho.

Um espaço de classificação baseado em dissimilaridade poder ser definido da seguinte forma (DUIN, R.P., ET. AL., 2010):

1. Obtenha as distâncias entre todos os pares de objetos no conjunto de treinamento  $X = \{o_1, \dots, o_n\}$  como uma matriz de dissimilaridade  $D(n \times n)$ .  $D_{ij}$  é a distância entre o  $i$ -ésimo e o  $j$ -ésimo objetos.
2. Defina o espaço de dissimilaridade como um espaço vetorial  $X$  por  $X = D$ . Portanto, o  $i$ -ésimo objeto é representado por um vetor de dissimilaridade de valores  $D_{ij}$ , onde  $j = 1, \dots, n$ .
3. Treine os classificadores usando os objetos de treinamento representados no espaço de dissimilaridade.
4. Os objetos do conjunto de teste são representados no espaço de dissimilaridade pelas suas distâncias aos objetos no conjunto de treinamento. A seguir, eles são fornecidos como entradas do classificador treinado neste espaço, para obtenção da classificação.

#### **4.1. Métricas para representação por dissimilaridade**

Como dito anteriormente, a representação por dissimilaridade utiliza-se da dissimilaridade entre pares de objetos. Sendo assim, é necessário que a métrica de dissimilaridade seja bem escolhida para que haja uma boa representação entre os dados. Neste caso, medidas de distância entre os elementos são as mais adequadas, por possuírem a característica de serem simétricas e positivas. A medida mais utilizada é a norma-2 (distância euclidiana), no entanto outras medidas podem também ser utilizadas, como: norma-n, função sigmoide e funções radiais. Neste trabalho, duas medidas de distância são utilizadas como funções de comparação: a norma-2 e uma medida de distância elaborada para lidar com problemas de classificação e comparação de séries, chamada de distância lógica.

Na distância euclidiana, sendo  $X$  o conjunto de treinamento e  $Y$  o conjunto a ser representado, a comparação entre os dados independe dos dados em si e de suas qualidades, somente sendo necessário que  $X$  e  $Y$  estejam, a princípio, representados no mesmo espaço. Utilizando-se a distância euclidiana entre  $X$  e  $Y$  como função de comparação, ter-se-á como

entrada as posições de  $X$  e  $y$  dentro do mesmo espaço. Logo, seja  $X = \{x_1, \dots, x_m\}$  e  $Y = \{y_1, \dots, y_k\}$ , onde  $x_i, y_j \in \mathbb{R}^n, \forall i = 1, \dots, m$  e  $j = 1, \dots, k$ . O resultado da função de comparação entre os dados é mostrada pela equação (42) da norma-2 abaixo:

$$D_{ij} = \|x_i - y_j\|^2 = \sqrt{(x_{i1} - y_{j1})^2 + (x_{i2} - y_{j2})^2 + \dots + (x_{in} - y_{jn})^2} \quad (42)$$

Por sua vez, a distância lógica, ou distância booleana, é uma métrica desenvolvida para potencializar a comparação entre dois dados quaisquer. Diferentemente do caso euclidiano ou de normas-n, ou outras medidas onde os atributos dos dados tenham que representar uma posição no espaço, o método aqui proposto serve para um atributo de qualquer natureza, desde que os dados a serem avaliados possuam os mesmos atributos, vide adiante.

De forma geral, seja  $X = \{x_1, \dots, x_m\}$  e  $Y = \{y_1, \dots, y_k\}$ , onde  $x_i, y_j \in \mathbb{R}^n, \forall i = 1, \dots, m$  e  $j = 1, \dots, k$ . A função de comparação entre um dado  $x_i$  e  $y_j$  será regida pela equação (43) de distância lógica abaixo:

$$D_{ij} = \frac{1}{n} \sum_{p=1}^n d_{ij}^{(p)} \quad (43)$$

Onde:

$$d_{ij}^{(p)} = \begin{cases} 0, & |x_{ip} - y_{jp}| \leq \varepsilon \\ 1, & |x_{ip} - y_{jp}| > \varepsilon \end{cases}, p = 1, \dots, n \quad (44)$$

A primeira equação (eq. 43) é a representação do objeto gerado através da distância lógica, obtido a partir do valor médio dos atributos obtidos da segunda equação (eq. 44). Se a diferença entre os atributos estiver dentro de uma região definida por  $\varepsilon$ , supõe-se que não há diferença entre os atributos ( $d_{ij}^{(p)} = 0$ ). Caso contrário, assume-se que há diferença entre os atributos dos dados e faz-se  $d_{ij}^{(p)} = 1$ .

O resultado é que existe um vetor  $\mathbf{D}_{ij}$  ordenado de  $n$  elementos binários  $d_{ij}^{(p)}$ , sendo a sua forma determinada pelo resultado da interação entre os dois dados  $x_i$  e  $y_j$  através da função de dissimilaridade adotada (eq. 45).

$$\mathbf{D}_{ij} = [d_{ij}^{(1)}, d_{ij}^{(2)}, \dots, d_{ij}^{(n)}] \quad (45)$$

A métrica lógica é obtida extraíndo o valor médio dos elementos de cada vetor  $\mathbf{D}_{ij}$  e utilizando-os como atributos para o novo conjunto de dados. Sendo assim, a representação por dissimilaridade lógica será determinada por um conjunto de dados de tamanho  $m \times k$ , onde  $m$  representa o número de dados e  $k$  o número de atributos de cada dado, tal que:

$$D_{ij} = \begin{cases} D_{ij} = 0, & \text{se } i = j \\ D_{ij} \rightarrow 0, & \text{se } y_j \rightarrow x_i \\ D_{ij} \rightarrow 1, & \text{se } y_j \neq x_i \end{cases} \quad (46)$$

Em outras palavras, o valor da distância lógica entre dois dados é sempre maior ou igual a zero e menor ou igual a 1. Além disso, ela diminui quanto maior for a semelhança entre os dados e aumenta caso contrário. Por essas razões, este método apresenta grande vantagem quando se deseja comparar dados entre si, além do que a região de operação dos dados é limitada, diferentemente do caso euclidiano.

## 5. SISTEMA PROPOSTO

Este capítulo irá abordar o desenvolvimento do sistema proposto, sua implementação, os recursos utilizados e suas etapas de processamento.

O fluxograma do sistema de detecção de falhas de circuitos pode ser visto na figura 3. Ele pode ser implementado em um sistema BIST (*Built In Self Test*) ou em um sistema ATE (*Automatic Test Equipment*).

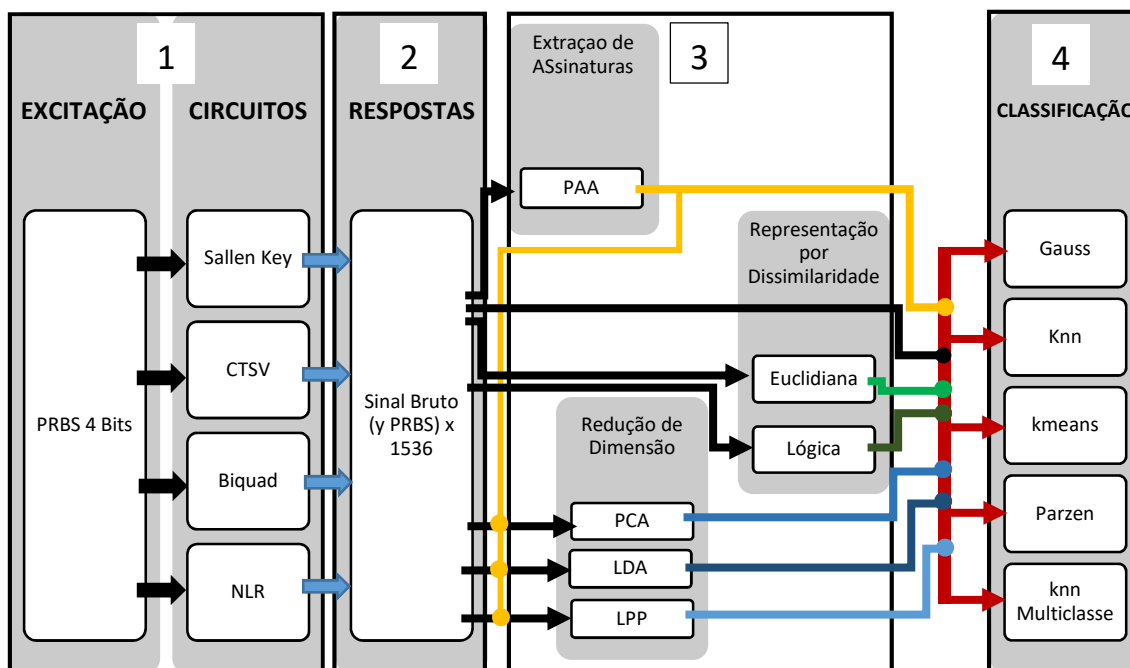


Figura 3: Fluxograma do Sistema Detector de Falhas

O funcionamento do detector é dividido em basicamente quatro etapas: Excitação, Resposta, extração de atributos e classificação, sendo as duas primeiras etapas relacionadas ao circuito avaliado a terceira etapa, à representação dos dados obtidos, e a quarta etapa, ao processo de detecção em si. Este sistema é estruturalmente semelhante ao sistema adotado por (LOMBARDI, A., 2011) e (AMARAL, J.L., 2011) em seus estudos de detecção de falhas em circuitos. Entretanto, este trabalho implementa alterações nas etapas 1 (excitação) e 3 (extração de atributos).

### **5.1. Excitação e resposta do circuito através de uma entrada pseudoaleatória**

A excitação é uma função de entrada  $f(t)$  aplicada a um circuito tal que, sendo este um circuito linear e invariante no tempo (CLIT), o sinal de resposta em seus terminais de saída com a função de excitação será:

$$y(t) = h(t) * f(t) = \int_0^t h(\tau) \cdot f(t - \tau) \cdot d\tau \quad (47)$$

Sendo:

- $y(t)$ : a resposta de saída do circuito;
- $h(t)$ : função característica do circuito.

Esta resposta da saída do circuito à uma excitação é dividida em dois regimes de resposta: o regime transitório e o regime permanente. No primeiro caso, a resposta dada existe somente no início ou cessar da excitação, desaparecendo após um certo período, quando inicia-se o regime permanente de resposta do circuito, sendo este, periódico. Para o sistema proposto de detecção de falhas em circuitos, o sinal de entrada deve excitar o circuito de forma a fornecer uma resposta que permita uma maior distinção entre os sinais de saída que representem falhas, beneficiando a classificação. Por isso, assim como no caso de (LOMBARDI, A., 2011) e (AL-QUTAYRI, M. A., SHEPHERD, P. R., 1992), será considerada somente a resposta transitória do circuito à excitação utilizada.

As principais funções de excitação são funções simples cuja resposta dos CLIT são de fácil obtenção e compreensão teórica, como: o degrau unitário, e funções alternadas senoidais e funções quadradas. (LOMBARDI, A., 2011) obtém a resposta ao impulso através da derivada da resposta do circuito ao degrau para a classificação pós extração de assinaturas. Esta estratégia é interessante, pois o sinal de resposta do circuito para o impulso é equivalente à resposta dada por sua própria função de transferência,

característica única do circuito. Além do mais, qualquer SLIT pode teoricamente ser completamente caracterizado pela sua resposta ao impulso (SINGH ET. AL., 2004).

Apesar dos bons resultados obtidos para a detecção, a excitação por meio de sinais básicos impõe dificuldades para o impulso unitário, que necessita de um estágio derivador, o que pode aumentar a propagação de ruídos no sinal de saída. Já quando usamos sinais senoidais, a resposta do circuito no domínio da frequência é limitada. Dos estudos de resposta em frequência de Fourier, é conhecido que as respostas às excitações de sistemas lineares invariantes no tempo são combinações lineares de respostas do sistema às várias componentes de frequências múltiplas do sinal de entrada. Logo, quanto maior a riqueza de componentes de Fourier no sinal de entrada, maior será a exposição do circuito a diferentes componentes de frequência, o que é vantajoso quando desejamos detectar falhas, pois submetemos os circuitos a diferentes condições. Portanto, para efetuar análise e comparações entre CLITs como no caso da detecção de falhas paramétricas, utilizar sinais de excitação não-determinísticos é teoricamente mais vantajoso do que utilizar sinais determinísticos. Por essa razão, neste trabalho será adotado um sinal pseudoaleatório de 4 bits (300 $\mu$ s não-determinísticos) de excitação.

As primeiras duas etapas do sistema de detecção são então realizadas ao excitar um circuito por uma entrada pseudoaleatória de 4 bits, obtendo a sua resposta em várias faixas de frequência.

## **5.2. Extração de características**

Obtida a resposta do circuito à excitação pseudoaleatória, os dados de resposta possuem grande volume de informação disponível o que torna o processo de detecção menos eficiente e mais demorado, visto que o classificador precisa analisar um maior número de informações para inferir resultados, o que irá levar a um aumento de complexidade. Os métodos de extrações de características de PAA (KEOGH, E., PAZZANI, M., 2000) para extração de assinaturas, de PCA (XIAO, Y., HE, Y., 2011), LDA (XIAO, Y., FENG, L., 2012), LPP (HE, X., NIYOGI, P., 2003) para redução de dimensão

e de representação por dissimilaridade (DUIN, R.P., ET. AL., 2010; KEOGH, E., PAZZANI, M., 2000) explicados nos capítulos 3 e 4 são empregados na etapa 3 do sistema proposto.

Uma peculiaridade deste trabalho é a proposta da combinação entre esses diferentes métodos de extração de características. Esse estudo é representado no fluxograma da figura 3. Para cada circuito analisado são obtidos 10 diferentes conjuntos de dados a partir da mesma resposta à excitação PRBS4. Estes conjuntos são utilizados independentemente para a detecção. O primeiro conjunto é obtido sem nenhum tratamento de extração, mantendo a característica do sinal PRBS original; o segundo conjunto é obtido ao realizar a extração de atributos do sinal de resposta ao PRBS com um PAA; os terceiro, quarto e quinto conjuntos são obtidos ao se realizar a redução de dimensão nos sinais de resposta à PRBS com os métodos PCA, LDA e LPP respectivamente; os conjuntos 6, 7 e 8 são obtidos ao se realizar a extração dos sinais de resposta ao PRBS com o PAA e na sequência é aplicada a redução de dimensão com os métodos utilizados e; por último são obtidos os conjuntos 9 e 10 a partir da representação de dissimilaridade, com a distância euclidiana padrão e com a distância lógica, proposta neste trabalho, respectivamente. Cada um destes conjuntos será utilizado como entrada dos classificadores na etapa 4.

### ***5.3. Classificação dos dados a partir de um comitê de classe única***

A detecção é realmente realizada nesta etapa do processo. O subsistema consiste basicamente em um classificador e do conjunto de dados pós-extração de características. Antes da real execução da detecção, os dados de resposta do circuito são utilizados para o treinamento do classificador, obtendo o ajuste da sua fronteira de decisão. A seguir, é realizada a detecção para um circuito qualquer, de topologia equivalente aos circuitos treinados. Nesse caso, a saída do classificador corresponderá à saída do sistema de detecção, acusando um determinado defeito ou não no circuito conforme a pertinência do dado obtido à fronteira de decisão do classificador. A figura 4 ilustra este subsistema:



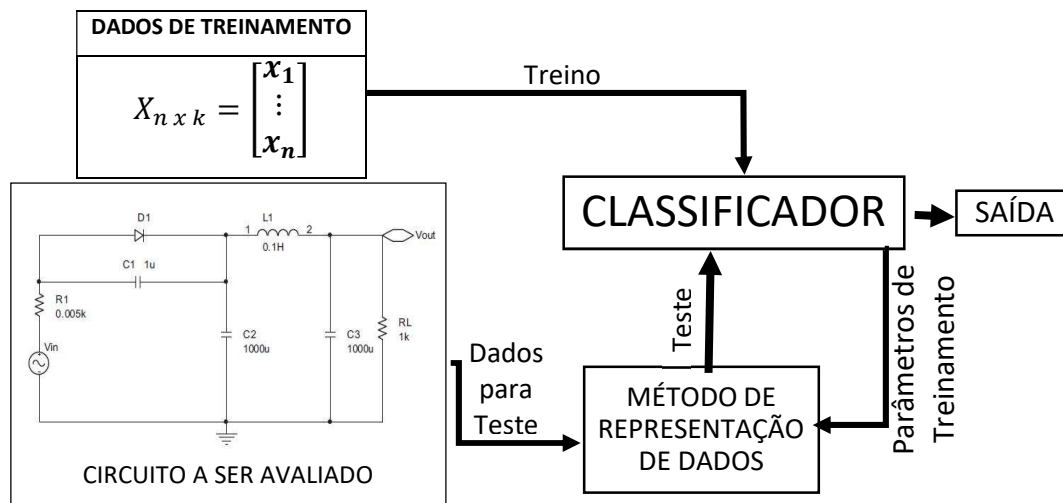


Figura 4: Ilustração simplificada do procedimento de classificação

Caso deseje-se apenas definir se o circuito apresenta falha ou não, o classificador somente irá necessitar somente de uma única classe (LI, X., XIE, Y., 2013), não havendo a identificação da falha neste caso. No sistema proposto, é desejado que a detecção não só determine a existência ou não de falha nos circuitos, como também seja capaz de identificá-las (ou seja, a especificidade). Neste caso, mais classes serão necessárias, havendo  $N$  classes de saída para  $N$  falhas do circuito.

Existem, então, duas possíveis arquiteturas para o classificador na detecção e identificação de falhas paramétricas: a primeira delas é a utilização de um classificador multiclasse (Fig. 5A) e a outra arquitetura consiste em utilizar um comitê de classificadores de classe única (Fig. 5B). Para o primeiro caso, este classificador possui um número de classes igual ao número de saídas, que neste caso representam as  $N$  possíveis falhas paramétricas do circuito estudado. No segundo caso, quando a saída do circuito corresponder a um sinal da classe  $i$ , então a saída  $i$  mudará o seu estado de falso ( $C_i = 0$ ) para verdadeiro ( $C_i = 1$ ) enquanto todas as outras ( $N - 1$ ) classes deverão adquirir o estado de falso ( $C_j = 0, \forall j \neq i$ ).

Como dito no capítulo 2, a arquitetura de classificadores multiclasse é bastante eficiente quando existe um conjunto de dados bem definido. Em outras palavras, os classificadores multiclasse são ótimos quando os dados das diferentes classes são bem distintos entre si. No caso em que dados entre duas ou mais classes têm pouca diferenciação entre si, as fronteiras de

decisão das classes envolvidas tornam-se próximas e, conseqüentemente, o classificador fica significativamente suscetível à classificação incorreta. Outro ponto importante de problemas de classificadores multiclasse é a sua incapacidade de lidar com dados não-previstos durante o treinamento, o que leva a dois resultados importantes. O primeiro ponto é que este classificador sempre classificará incorretamente classes não-previstas, pois necessita entregar algum resultado para qualquer vetor de atributos aplicado em sua entrada, respondendo equivocadamente a falha não prevista como uma das classes existentes (LOMBARDI, A., 2011). Em segundo lugar, necessitará sempre realizar um novo treinamento de todas as suas classes a cada vez que uma nova classe ter de ser adicionada ou retirada.

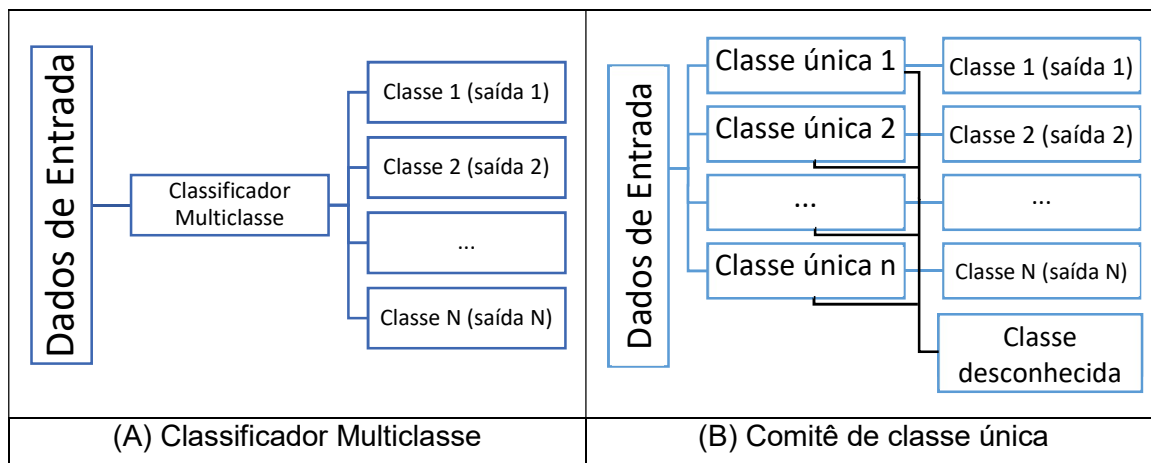


Figura 5: Classificadores multiclasse (A) e comitê de classe única (B)

No caso da arquitetura de comitê de classe única, serão utilizados  $N$  classificadores de classe única no lugar das  $N$  classes de um único classificador multiclasse, para detecção das falhas. Cada classificador  $i$  será um especialista para a detecção dos dados de falhas pertencentes à classe  $i$ . Os classificadores do comitê serão construídos usando os mesmos dados de treinamento de falhas predefinidas para o treinamento de um classificador multiclasse, com a diferença que para uma falha/classe  $i$ , apenas os seus dados serão definidos como *target* e todos os outros dados das outras  $(N - 1)$  falhas serão classificados como desconhecidos para o classificador  $i$ , o mesmo ocorrendo para todos os outros  $(N - 1)$  classificadores e seus respectivos dados de falhas.

Como cada dado de falha está relacionado exclusivamente a um único classificador especialista, isso torna os conjuntos de falhas e as fronteiras de decisão independentes entre si para um mesmo espaço de observação. Consequentemente, torna-se o detector mais simples, reduz o problema encontrado com relação à proximidade de dados de classes distintas e atenua-se a incapacidade de lidar com falhas não previstas. Uma das desvantagens do comitê de classe única deve-se ao fato de que os classificadores de classe única têm que rejeitar uma parcela dos dados previstos para formarem sua fronteira de decisão (fração de rejeição), como explicado no capítulo 2, sempre existindo um erro mínimo de classificação. Mesmo assim, eles são competitivos quando comparados aos classificadores multiclasse. Outra desvantagem é que, no caso dos comitês, ao lidar com dados próximos a  $k$  fronteiras de decisão e sendo cada classificador independente do outro, todos podem classificar o dado como pertencente à sua respectiva classe de falhas (figura 6). Isto não necessariamente representa um erro, pois se o dado pertence a uma das classes que acusaram pertinência, então ele classificou corretamente para uma das classes. E seu erro para as outras é devido à proximidade do dado com as várias fronteiras de decisão, além da proximidade das várias fronteiras de decisão entre si. Neste caso, dizemos que o comitê acusou uma das possíveis classes da qual o dado pertence.

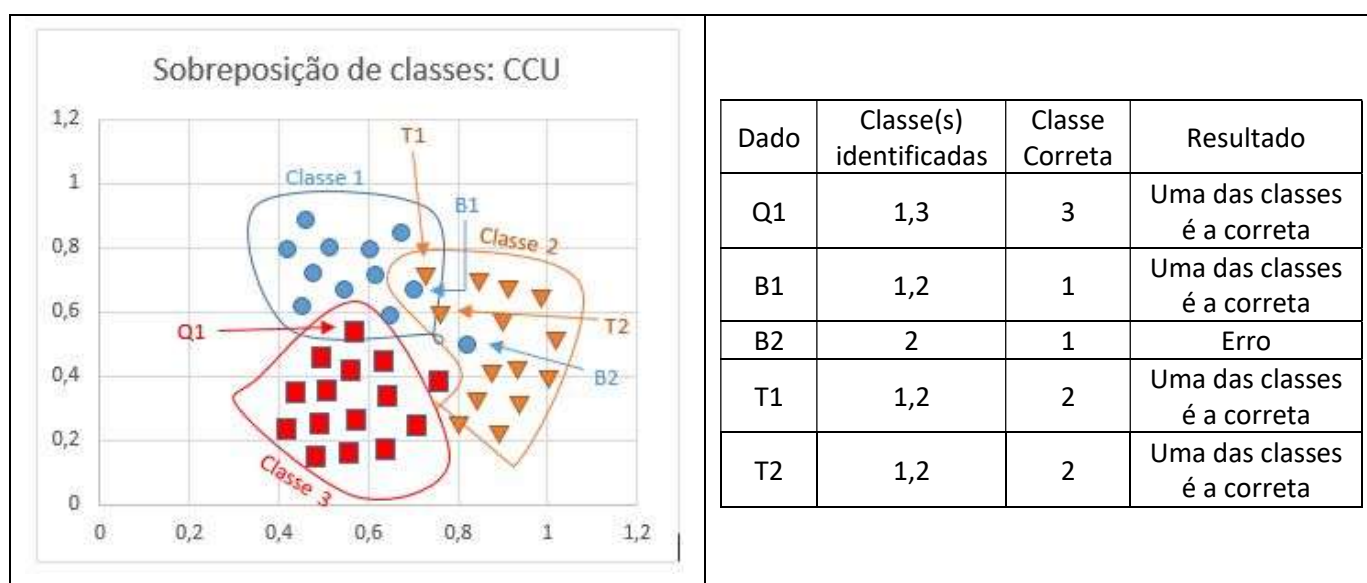


Figura 6: Ilustração da sobreposição de classes em CCUs, e Tabela 2 descritiva da figura 6

Após a realização da classificação pelo comitê de classe única, a falha existente ou não é acusada e identificada, através da combinação obtida do conjunto de saídas dos classificadores de classe única especialistas.

## 6. ESTUDO DE CASO

A avaliação das diferentes arquiteturas de classificadores na detecção de falhas foi realizada em quatro circuitos analógicos diferentes: O filtro passa-banda Sallen-Key, o filtro universal (CTSV – *Continuous Time State-variable*), o filtro passa-alta ativo Biquad (BIQUAD – *Four Opamp Biquad Highpass Filter*) e Retificador não-linear. Os procedimentos para a realização das avaliações são aqueles explicados no capítulo 5, os quais requerem a obtenção dos conjuntos de treinamento e teste.

Os conjuntos de treinamento e teste são obtidos em duas etapas diferentes. Primeiramente, o circuito a ser avaliado é modelado na plataforma de simulação do LTSpice para avaliação, simulação e modelagem de circuitos eletrônicos, onde são realizadas  $K$  simulações por circuito, para cada conjunto de treino e teste e sendo  $2.K$  o número total de simulações realizadas. A geração dos dados de cada classe é realizada através da varredura dos parâmetros associados à falha correspondente, sendo cada parâmetro do circuito definido como o valor de cada componente do circuito fora das suas especificações. Logo, os parâmetros de cada classe têm como referência para a existência da falha os dados fora do intervalo de tolerância do componente associado. Essa varredura é realizada por meio de um ensaio de Monte-Carlo do circuito, variando o valor do parâmetro (componente) da classe correspondente na região especificada em um total de 300 ensaios, tendo-se 300 dados por cada classe do circuito e um número total de classes de cada circuito de  $N = \frac{K}{300}$  classes.

Na segunda etapa é realizada a transferência dos dados obtidos nas simulações do LTSpice para a plataforma MATLAB. A transferência é feita realizando-se a conversão dos dados de falhas para o formato do MATLAB, para cada conjunto de treino e teste. Cada um destes conjuntos é obtido em simulações independentes, existindo um arquivo de dados para cada (respectivo) conjunto, sendo necessário repetir o procedimento para cada um separadamente.

Ao se obter os dados das simulações das respostas à excitação pseudoaleatória de 4bits, implementadas no LTSpice, transfere-se as

mesmas para o MATLAB. A resposta à entrada pseudoaleatória será chamada de sinal bruto. Durante a transferência entre as plataformas são gerados novos conjuntos para todos os métodos de extração de atributos que serão utilizados neste trabalho, a partir dos conjuntos de dados de sinais brutos. Nos casos em que se realiza PAA, a configuração do parâmetro deste método é feito anteriormente à validação por sua curva de erro de reconstrução. Nos demais casos, os novos conjuntos são obtidos com uma configuração de parâmetros realizada por validação cruzada de 10 pastas para cada classificador, realizada com os dados de sinal PRBS de treinamento de cada circuito. Para os conjuntos de teste destes métodos de extração de atributos, são utilizados os parâmetros obtidos da validação cruzada dos dados de treinamento de cada caso correspondente. A figura 7 descreve este procedimento:

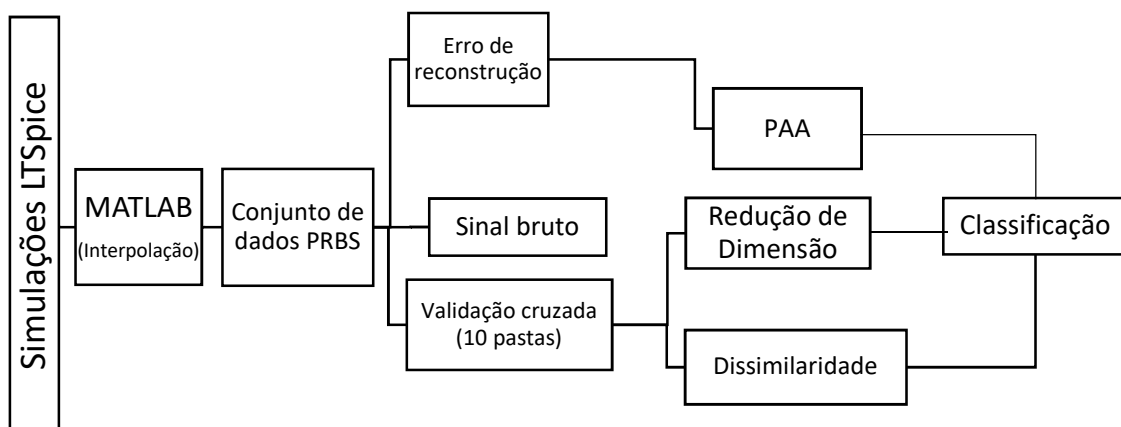


Figura 7: Esquema de obtenção dos conjuntos de dados

Quanto à identificação das falhas em seus componentes, elas podem ser divididas, basicamente, em três categorias:

1. As falhas podem ser simples e ocorrerem em apenas um componente por vez;
2. Podem ser múltiplas e mais de um componente apresentar falha simultaneamente; ou
3. Pode não haver falha em nenhum dos componentes (comportamento normal).

A falha paramétrica em cada componente, por sua vez, está relacionada a duas possíveis características independentes: o componente pode estar com valor acima do limite de tolerância especificado, com valor abaixo do limite de tolerância especificado. (LOMBARDI, A., 2011) e (AMARAL, J., 2011), consideram cada uma destas características como uma classe independente da outra. Neste trabalho, também, cada característica determinará uma classe diferente de falhas, pois tais características refletem diferentes respostas do circuito em falha.

Logo, para cada componente passivo e paramétrico<sup>1</sup>, haverá duas classes de falhas associadas, sendo o número de falhas simples definido pelo dobro do número de componentes deste tipo no circuito:

$$F_{SingleFail} = 2 \cdot K \quad (48)$$

Com relação à quantidade de falhas múltiplas ( $F_{MultiFail}$ ), a possibilidade de sua existência no circuito depende do número de componentes que ele possui. Definindo-se  $K$  o número de componentes passivos e paramétricos de um circuito  $C$ , o número  $i$  de componentes a apresentar defeito simultaneamente neste circuito, para  $i = 2, 3, \dots, k$  componentes, estando ainda cada componente associado a duas classes de falhas distintas, é definido pela combinação de todas as possíveis e distintas falhas. Assim o número de falhas múltiplas é igual à soma de todas as combinações de falhas nos componentes, para  $i = 2, \dots, k$  componentes, com falhas simultâneas:

$$F_{MultiFail} = 2 \cdot \sum_{i=2}^K A_i^K \quad (49)$$

Por sua vez, para o terceiro caso aqui considerado, a não existência de falha está relacionada a uma situação específica: quando todos os

---

<sup>1</sup> Passivo e paramétrico (PP): todo componente passivo em um circuito cujo funcionamento é definido por um parâmetro variável em intervalos. Ex.: resistores, capacitores e indutores são passivos e paramétricos. Um diodo não é passivo e paramétrico, pois funciona como uma chave (circuito aberto ou curto circuito), o que permite somente existirem falhas catastróficas em seu caso.

componentes do circuito tiverem seus valores dentro dos limites normais de funcionamento. Logo:

$$F_{NoFai} = 1 \quad (50)$$

Isso permite que se defina o número total  $F$  de classes de falhas de um circuito, como:

$$F = 2 \cdot \left( K + \sum_{i=2}^K A_i^K \right) + 1 \quad (51)$$

Da equação acima, vê-se que, mesmo para pequenos valores de  $K$ , o número de classes  $F$  é grande, principalmente devido às falhas simultâneas. No entanto, o comportamento destas não é independente e tende a acompanhar o comportamento anormal de um dos componentes. Devido a essa característica, neste trabalho focar-se-á nas classes apenas para as falhas simples dos circuitos, reduzindo o número de classes de falhas de cada circuito para:

$$F = 2 \cdot K + 1 \quad (52)$$

Dessa forma, por meio da equação (52), tem-se o número de classes de falhas a ser utilizado para cada circuito, os quais são apresentados na tabela 3. Por fim, é importante dizer que somente os componentes passivos e paramétricos são considerados para a geração de classes, e nem todos eles são analisados em todos os circuitos:



*Tabela 3: Tabela de relação nº de classes X nº componentes*

Circuito	Número de Componentes PP	Número de classes de falha simples total	Número de classes de falhas utilizado
Sallen-Key	7	15	11
CTSV	9	19	19
Biquad	12	25	13
Retificador não-Linear	5	11	11

Conhecendo-se o número de classes a ser utilizado para cada circuito, pode-se então iniciar o estudo de cada caso abordado neste trabalho.

## 6.1. Circuito Sallen-Key

### 6.1.1. Desempenho

Este é um circuito passa-banda ativo de 2ª Ordem, projetado neste trabalho com 1 amplificador operacional como o coração do circuito, 5 resistores e 2 capacitores, a qual é a configuração mais básica deste tipo de arquitetura. A sua arquitetura é mostrada na figura 8 e sua função de transferência é definida na equação (53). Este é o principal circuito utilizado no estudo de detecção de falhas, devido à sua baixa sensibilidade aos parâmetros passivos do circuito já tendo sido previamente avaliado por (AMARAL, J.L., 2011; LOMBARDI, A., 2011; LI, X., XIE, Y., 2013; XIAO, Y., HE, Y., 2011; XIAO, Y., FENG, L., 2012).

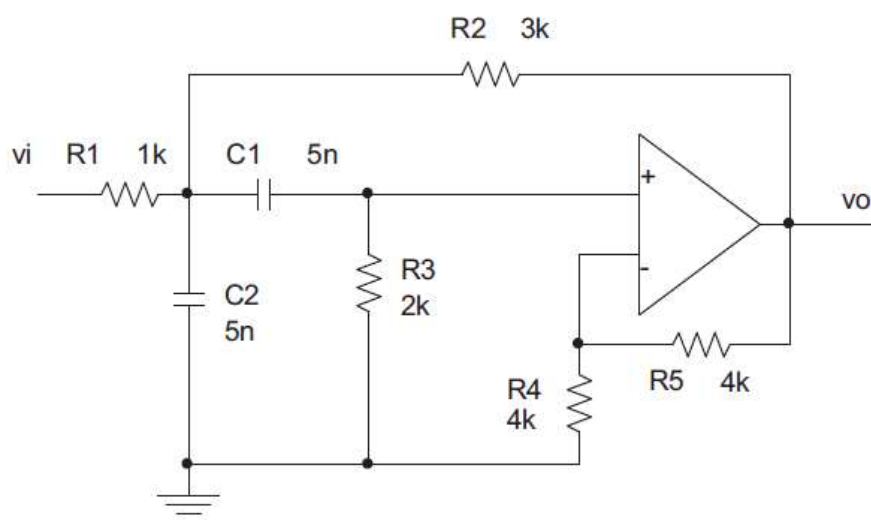


Figura 8: Circuito Sallen Key

$$\frac{V_o}{V_i} = \frac{s}{s^2 + \frac{\omega_p}{Q_p} s + \omega_p^2} \quad (53)$$

A tabela 4 abaixo identifica os componentes a serem analisados e suas respectivas classes durante a classificação:

Tabela 4: Componentes com falha e suas respectivas classes, Sallen-Key

Componente	Falha Associada	Respectiva Classe
R1	R1+	1
	R1-	2
R2	R2+	3
	R2-	4
R3	R3+	5
	R3-	6
C1	C1+	7
	C1-	8
C2	C2+	9
	C2-	10
NoFail		11

Um classificador multiclasse *knn* foi utilizado para fins de comparação e análise dos resultados fornecidos do classificador de classe única, como demonstrado na figura 3 no capítulo 5. A tabela 5 adiante mostra tais resultados:

Tabela 5: Resultado de classificação multiclasse, Sallen-key

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de Erro
1	300	100,00%	0	0,00%
2	300	100,00%	0	0,00%
3	300	100,00%	0	0,00%
4	300	100,00%	0	0,00%
5	300	100,00%	0	0,00%
6	300	100,00%	0	0,00%
7	300	100,00%	0	0,00%
8	300	100,00%	0	0,00%
9	300	100,00%	0	0,00%
10	300	100,00%	0	0,00%

11	300	100,00%	0	0,00%
Total:	3.300	100,00%	0	0,00%

Na figura 9, apresentam-se os resultados dos comitês de classe única na classificação do circuito Sallen-Key. Neste caso, o classificador que apresentou a melhor desempenho foi o *kmeans*. Dois conjuntos de resultados são apresentados de um mesmo classificador: seu desempenho desconsiderando todas as sobreposições em cor escura e em cor clara e desempenho considerando os dados sobrepostos em até 2 classes. Neste caso, o gráfico analisado mostra os resultados para o conjunto de dados de sinal bruto, ou seja, dados de resposta à excitação ao PRBS. O resultado do desempenho do classificador multiclasse (*knnc*) também foi inserido para fins de comparação, encontrando-se em cor distinta da utilizada para os classificadores de classe única. Maiores detalhes e resultados de desempenho de cada conjunto de dados pré-processado seguem nos anexos I e II.

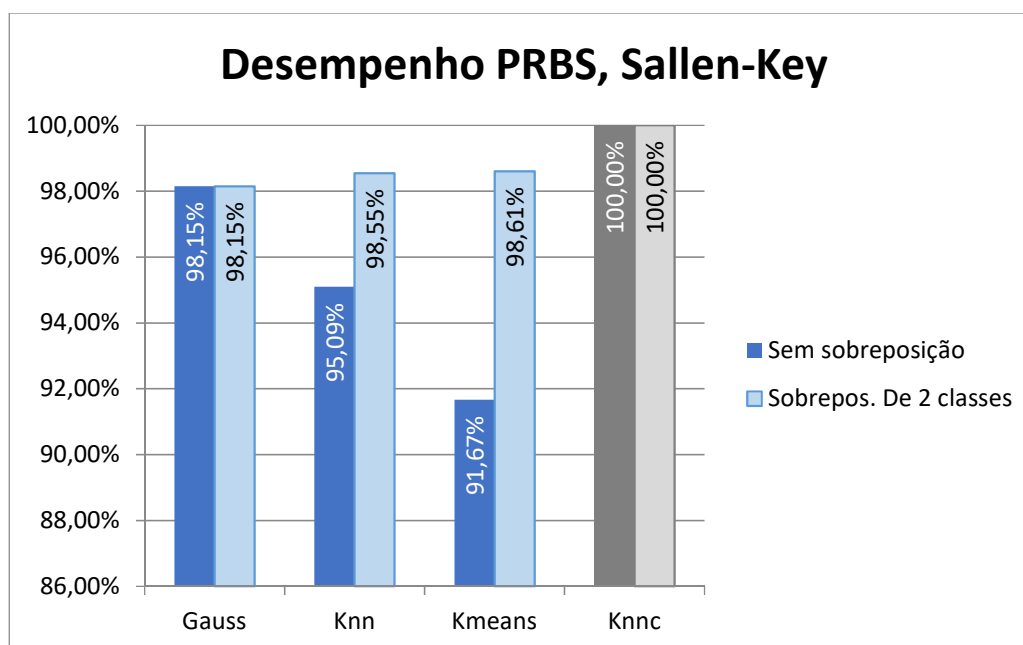


Figura 9: Desempenho para o Sallen-key com o sinal PRBS (Parzen não classificou)

A tabela 6 mostra os resultados do classificador com o melhor desempenho anterior, com relação à distribuição dos dados em cada classe. Comparado com o desempenho do multiclasse, pode-se dizer que há a equivalência entre os dois classificadores, pois o CCU apresenta uma rejeição mínima, o que em nosso caso são 3 circuitos de cada classe, totalizando 33 circuitos propositalmente rejeitados, fato que ainda é superado para as classes 6 e 11. Em segundo lugar, nenhum dos dados não classificado foi erroneamente colocado em outra classe, como aconteceria no caso do multiclasse, e sim foram rejeitados como dados desconhecidos, o que não representa erro do classificador (visto que ele não classificou).

*Tabela 6: Distribuição dos dados para o melhor classificador do Sallen-key para a curva PRBS*

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de Desconhecidos
1	295	95,67%	0	0,00%	5	1,67%
2	298	97,67%	0	0,00%	2	0,67%
3	299	98,33%	0	0,00%	1	0,33%
4	295	97,00%	0	0,00%	5	1,67%
5	297	97,00%	0	0,00%	3	1,00%
6	280	99,33%	2	0,67%	1	0,33%
7	296	99,00%	0	0,00%	4	1,33%
8	78	26,00%	3	1,00%	7	2,33%
9	294	97,67%	0	0,00%	6	2,00%
10	296	98,67%	0	0,00%	4	1,33%
11	297	99,33%	0	0,00%	3	1,00%
51 (6 e 8)	229	38,17%	0	0,00%	0	0,00%
Total:	3.254	98,61%	5	0,15%	41	1,24%

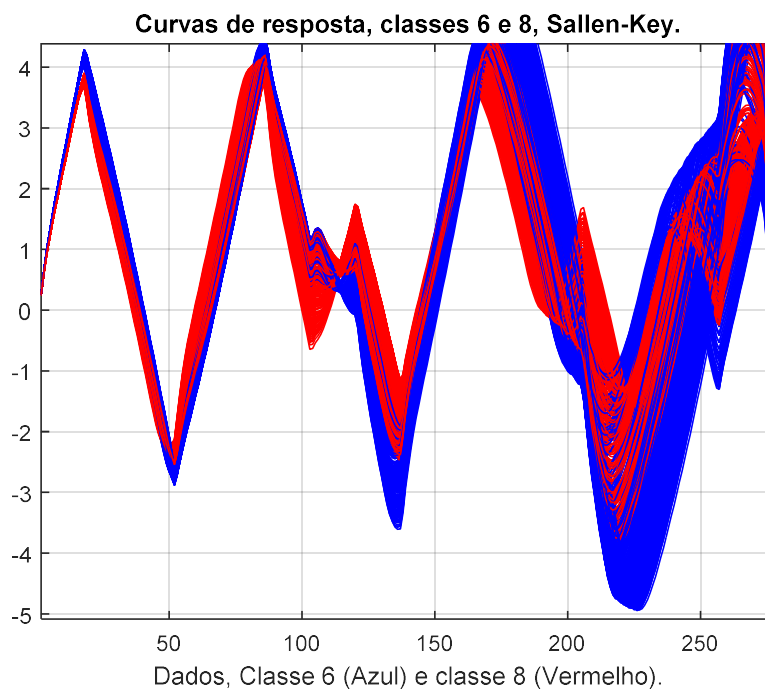
O Sallen-Key apresenta além da robustez quanto às variações de parâmetros, variações bem definidas entre os diferentes parâmetros de sua topologia, o que é benéfico para este trabalho que busca identificar as diferenças entre as variações dos diversos componentes, característica que será evidenciada como importante na análise do próximo circuito. Mesmo para

o conjunto de dados formado somente pela excitação básica, já se demonstra suficiente para efetuar a classificação das falhas simples no Sallen-key, seja para o classificador multiclasse ou para o comitê de classe única. Quanto à observação do desempenho entre os dois tipos de classificadores, pode ser observado pelas tabelas 4 e 5 e pela figura 9 que o desempenho de classificação entre ambos é praticamente equivalente do ponto de vista qualitativo. Quantitativamente, o classificador multiclasse tem uma mínima superioridade frente aos CCUs para este circuito. Esse fato está diretamente relacionado à limitação do classificador de classe única quanto a sua fração de rejeição (necessária para a implementação e execução deste tipo de classificador). Entretanto, os CCUs podem lidar com o problema de classes não previstas, enquanto o multiclasse não é capaz disto.

### **6.1.2. Sobreposição de classes**

O circuito Sallen-Key não apresentou muitos problemas com sobreposições de classes. Este é um circuito que possui uma variação bem definida entre os seus parâmetros e grande robustez, o que o faz ter uma classes bem definidas.

Os únicos pares de classes em sobreposição a surgirem em uma classificação no Sallen-Key foram as sobreposições entre as classes 6 e 8. Na figura 10, é possível observar a disposição destas classes entre si.



*Figura 10: Curvas de sobreposição, classes 6 e 8, Sallen Key*

A sobreposição entre as classes 6 e 8 deste circuito ocorre somente em alguns pontos das curvas de resposta, porém, mantendo uma região mínima de fronteira entre elas, existindo somente uma proximidade muito grande entre os dados das classes. Este tipo de sobreposição permite a discriminação entre as classes pelos classificadores e, não ocorre na maioria das classificações deste circuito, a classificação incorreta dos dados.

### **6.1.3. Influência da representação de dados na sobreposição**

A fim de analisar o comportamento de cada classificador e a influência do tratamento dos dados de entrada para classificação com os métodos de representação de dados, foi observado o comportamento de cada um para cada tipo diferente de representação de dados realizada. A figura 11 mostra o desempenho do classificador gaussiano no circuito Sallen-key, para cada método utilizado para representar os dados.

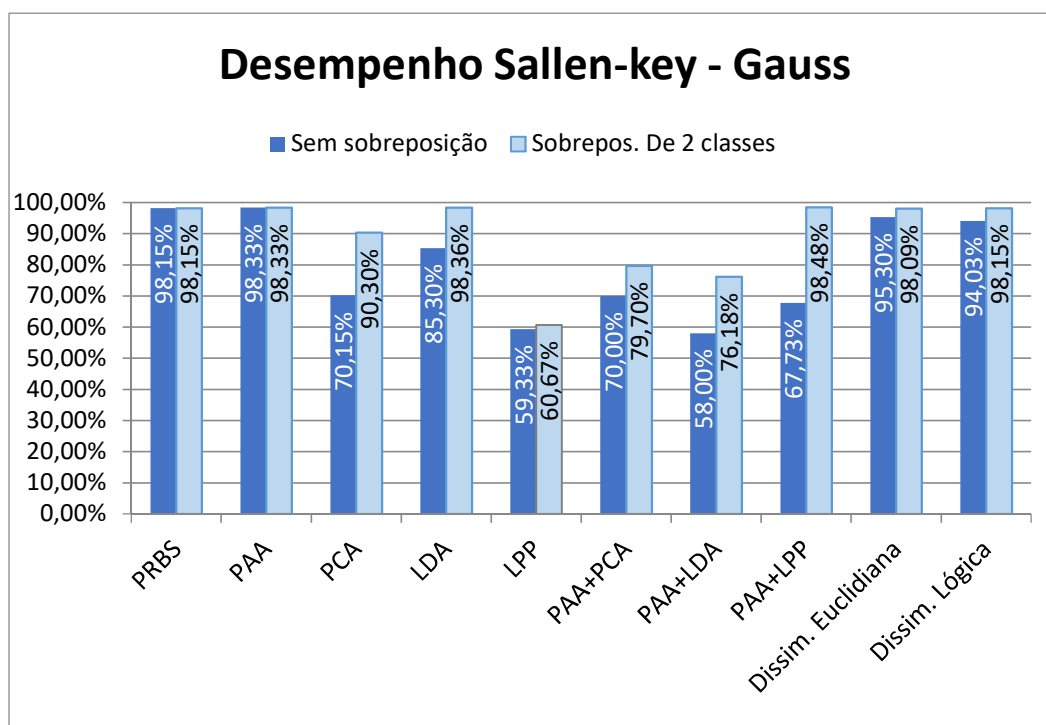


Figura 11: Desempenho por representação de dados, classificador Gauss, Sallen-Key

Para o método gaussiano, o Sallen-key apresentou um bom desempenho para a maior parte dos métodos de representação apresentados, até mesmo no caso do sinal bruto PRBS. O método PAA foi o que apresentou melhor desempenho, como mencionado anteriormente, tanto sem quanto com sobreposição. Graças à robustez apresentada por este circuito, a classificação apresenta bom desempenho para os casos onde a representação dos dados é feita pela extração de atributos, o que explica os melhores desempenhos do PAA, da dissimilaridade euclidiana e da dissimilaridade lógica. No caso dos métodos de redução de dimensão, tem-se um desempenho inferior pelo fato de reorganizarem os dados, interferindo no modelo estatístico obtido pelo Gauss.

Conforme indicado na figura 12, para o caso do *knn*, o PAA ainda se mostra o melhor método de representação para os dados com sobreposição, mas apresenta desempenho inferior ao do sinal bruto para o caso sem sobreposição. Para os casos de redução de dimensão, não houve um bom desempenho, salvo o LDA. Para o caso das representações por dissimilaridade, somente a euclidiana obteve um desempenho satisfatório.



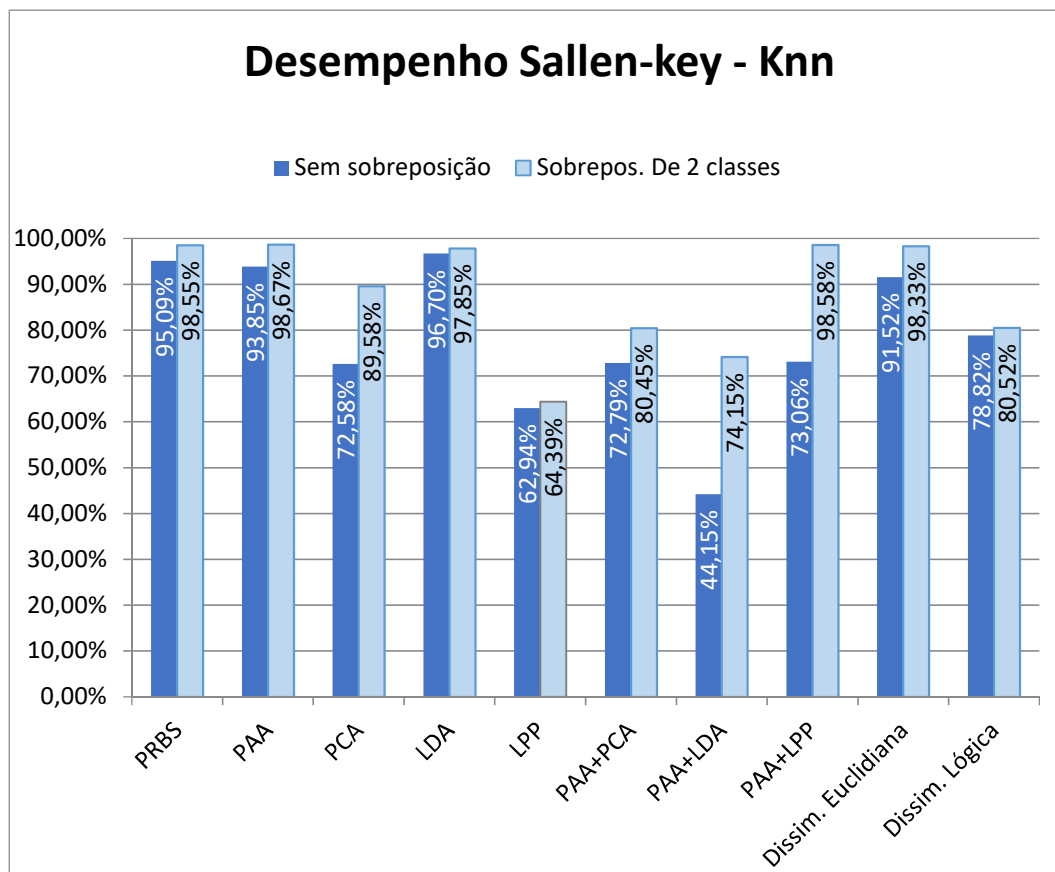


Figura 12: Desempenho por representação de dados, classificador knn, Sallen-Key

Para o caso do classificador *kmeans* no circuito Sallen-Key, há uma certa perda de dados em sobreposição, apresentando um desempenho 6,94% inferior entre o caso sem sobreposição para o caso com sobreposição da classificação do sinal bruto, no caso da extração de assinaturas, o PAA se equipara com a classificação bruta, ficando somente 1,55% inferior a representação pelo sinal bruto sem sobreposição de classes, e apenas 0,03% inferior no caso com sobreposição.

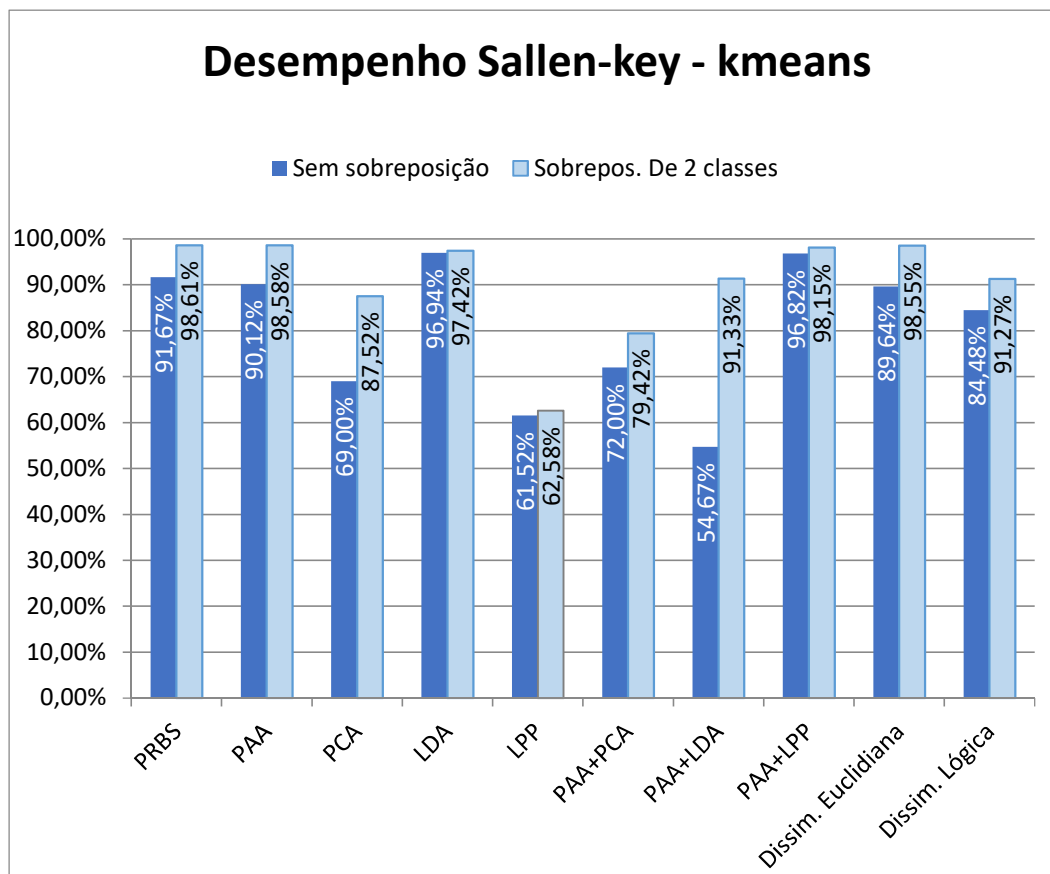


Figura 13: Desempenho por representação de dados, classificador kmeans, Sallen-Key

Um dos métodos que se sobressaíram está na redução de dimensão. Embora a maioria dos métodos desta categoria tenha apresentado fraco desempenho neste circuito, o método LDA mostrou ser uma exceção neste caso. Além de ter apresentado desempenho equivalente aos dos melhores métodos no caso com sobreposição. Ele também apresentou o melhor desempenho para o caso sem sobreposição, tendo somente 0,8% de diferença na sua classificação entre os dois casos. Para o caso da combinação entre a extração de assinaturas com redução de dimensão, somente para o caso do PAA+LPP houve um desempenho satisfatório, tendo um bom desempenho para ambos os casos de classificação. Já no caso da representação por dissimilaridade, ambos os casos apresentaram desempenhos bons, mas um desempenho menor para os casos sem sobreposição ao compará-los com o mesmo caso para o sinal bruto, com uma diferença entre os dois casos de 8,91% para a dissimilaridade euclidiana e de 6,79% para a dissimilaridade lógica.

Para o *Parzen*, houve grande variação entre os resultados: para o sinal bruto, o classificador não consegue realizar a classificação, o mesmo acontecendo na dissimilaridade euclidiana. O método do PAA consegue manter o bom desempenho, apresenta um dos melhores resultados, e mantém o mesmo resultado entre os dois casos de sobreposição. Os métodos de redução de dimensão também apresentaram um fraco resultado para este classificador, à exceção do LDA para o caso com sobreposição. A combinação dos métodos PAA+DR mostrou ter sido benéfica neste caso, pois dois dos três casos mostraram um bom desempenho com relação a este classificador, com exceção da representação PAA+LPP, que no caso sem sobreposição teve um fraco desempenho.

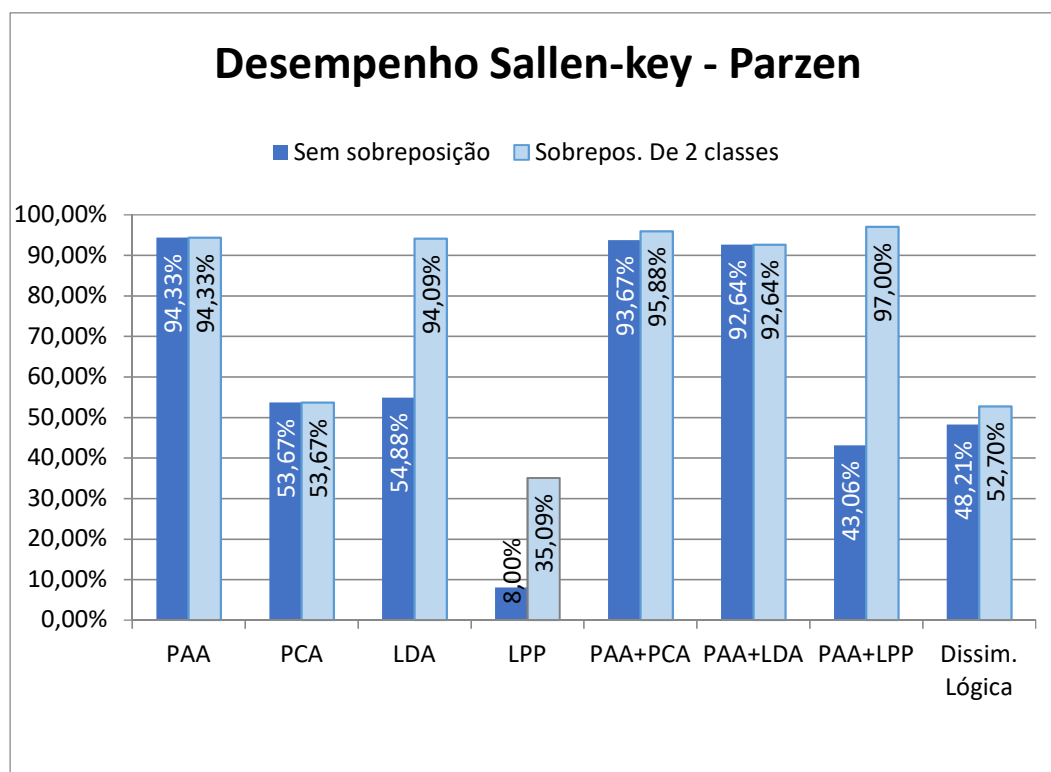


Figura 14: Desempenho por representação de dados, classificador Parzen, Sallen-Key

Nos casos das representações por dissimilaridade, ambas as técnicas apresentaram resultados insatisfatórios no *Parzen* para o Sallen-key. Não sendo possível a classificação dos dados tratados com a dissimilaridade euclidiana e apresentando um desempenho de aproximadamente 50% para a dissimilaridade lógica.

#### 6.1.4. Desempenho para a representação de dados

Além da avaliação do circuito realizada para o sinal PRBS, também foi verificado o comportamento deste circuito quando são aplicadas técnicas de representação de dados, com o objetivo de potencializar a diferenciação entre as diferentes classes. Na figura 15 é possível observar os resultados de desempenho de cada classificador para o método de representatividade que obteve o melhor desempenho com o mesmo.

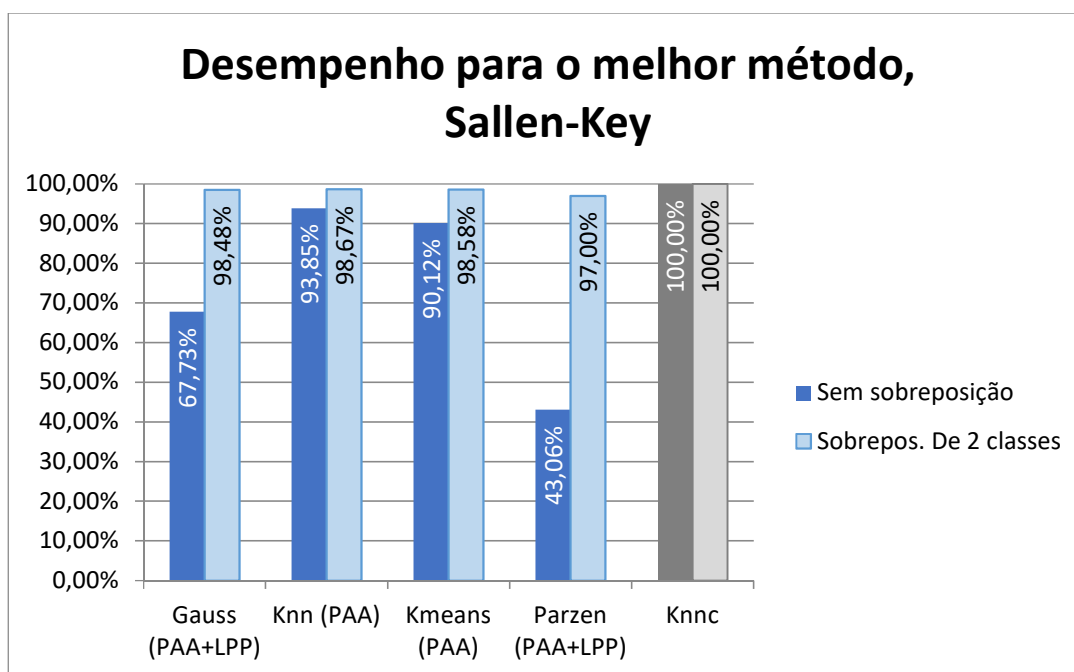


Figura 15: Curva de desempenho para os melhores métodos, circuito Sallen-key

O melhor classificador foi o *knn* para a representação de dados em PAA. A tabela 7 mostra a distribuição dos dados nas classes para o melhor classificador com a melhor representação de dados para o circuito Sallen-key.

Tabela 7: Distribuição dos dados para o melhor classificador na melhor sua melhor representação, Sallen-key

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de Desconhecidos
1	291	97,00%	0	0,00%	9	3,00%
2	295	98,33%	0	0,00%	5	1,67%
3	299	99,67%	0	0,00%	1	0,33%
4	298	97,00%	0	0,00%	2	0,67%
5	294	98,00%	0	0,00%	6	2,00%
6	299	99,67%	4	1,33%	1	0,33%
7	300	100,00%	0	0,00%	0	0,00%
8	135	45,00%	0	0,00%	2	0,67%
9	295	98,33%	0	0,00%	5	1,67%
10	294	98,00%	0	0,00%	6	2,00%
11	297	99,00%	0	0,00%	3	1,00%
51 (6 e 8)	159	26,50%	0	0,00%	0	0,00%
Total:	3.256	98,67%	4	0,12%	40	1,21%

Da tabela 7 acima é possível constatar pouca diferença de desempenho, comparando ao desempenho com o sinal bruto. É possível ver uma leve melhora de desempenho do *knn* quanto à situação em que não são aceitas quaisquer sobreposições, indicando uma possível afinidade do método de representação em questão para o classificador quanto a melhorar a discriminação dos dados de diferentes classes. O classificador multiclasse não mostrou diferença de desempenho entre o sinal bruto PRBS e o tratamento do sinal neste circuito, repetindo o desempenho máximo apresentado para o primeiro caso. Isto é justificado pelas características de estabilidade, poucas similaridades e nenhuma sobreposição de classes das respostas deste circuito.

### 6.1.5. Desempenho para classes não previstas

Nesta parte foi estudado como cada tipo de classificador se comportaria frente a um problema imprevisto do circuito. Para esta parte do estudo, um novo conjunto de dados de falhas foi gerado para avaliação com todas as 11 classes estudadas até então, 4 classes de falhas múltiplas do Sallen-key, e mais 3 novas classes de falhas desconhecidas. A relação entre componentes e classes previstas e não previstas é apresentada na tabela 8. Os classificadores serão avaliados com o treinamento somente das 11 classes de falhas anteriores.

Tabela 8: Tabela de falhas não previstas, Sallen-Key

Componente	Falha Associada	Respectiva Classe
R1	R1+	1
	R1-	2
R2	R2+	3
	R2-	4
R3	R3+	5
	R3-	6
C1	C1+	7
	C1-	8
C2	C2+	9
	C2-	10
NoFail		11
M1	(R1+, C1+)	12
	(R1-, C1-)	13
M2	(R1+, C1-)	14
	(R1-, C1+)	15
X1		16
X2		17
X3		18

Neste experimento, somente o sinal PRBS foi utilizado. Na tabela 9 abaixo são apresentados os resultados obtidos pelo classificador multiclasse para a avaliação de classes imprevistas. Como pode ser visto, a avaliação errada deste classificador é inevitável pelo fato de não lidar com dados fora do treinamento.

*Tabela 9: Resultados do multiclasse para falhas não previstas, Sallen-Key*

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de Erro
1	300	100,00%	0	0,00%
2	300	100,00%	0	0,00%
3	300	100,00%	0	0,00%
4	300	100,00%	0	0,00%
5	300	100,00%	0	0,00%
6	300	100,00%	0	0,00%
7	300	100,00%	0	0,00%
8	300	100,00%	0	0,00%
9	300	100,00%	0	0,00%
10	300	100,00%	0	0,00%
11	300	100,00%	0	0,00%
12	0	0,00%	300	100,00%
13	0	0,00%	300	100,00%
14	0	0,00%	300	100,00%
15	0	0,00%	300	100,00%
16	0	0,00%	300	100,00%
17	0	0,00%	300	100,00%
18	0	0,00%	300	100,00%
Total:	3.300	61,11%	2.100	38,89%

Adiante, são apresentados os gráficos de desempenho de cada classificador para a avaliação com falhas imprevistas no circuito Sallen-key (figura 16). Sendo os CCUs capazes de lidar com as classes não previstas, seu desempenho é mantido enquanto o desempenho do multiclasse diminui.

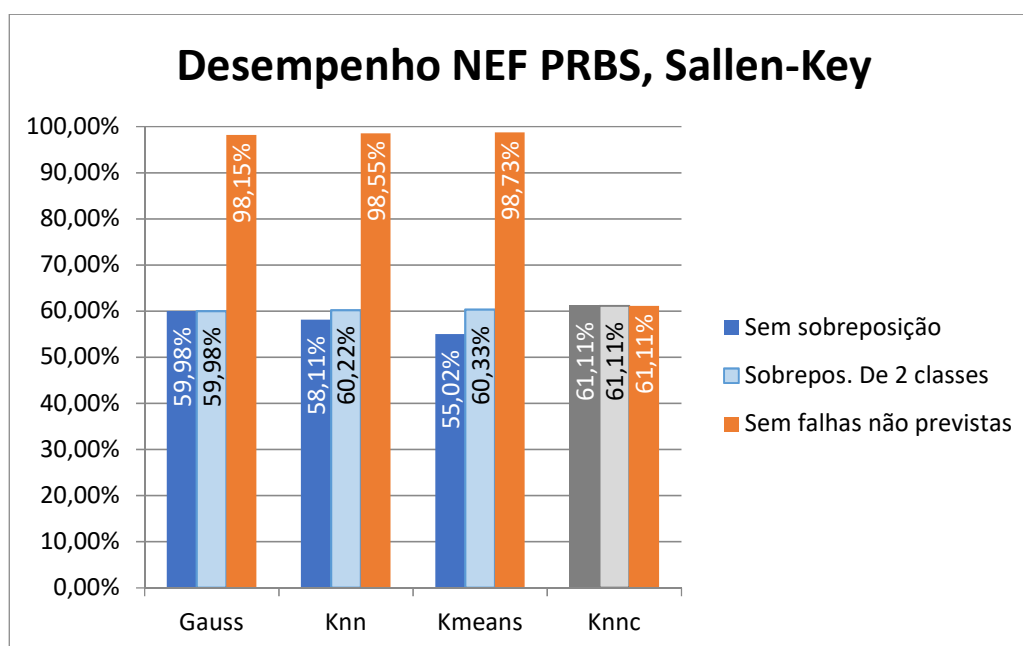


Figura 16: Desempenho dos classificadores para falhas não previstas com sinal PRBS, Sallen-key (Parzen não classificou)

Para este caso, o classificador que obteve o melhor desempenho foi o *kmeans*. A tabela 10 mostra os resultados detalhados deste classificador com falhas não previstas.

Tabela 10: Resultados da Classificação dos CCUs no sinal PRBS para falhas não previstas, Sallen-key

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de Desconhecidos
1	295	98,33%	0	0,00%	5	1,67%
2	298	99,33%	0	0,00%	2	0,67%
3	299	99,67%	0	0,00%	1	0,33%
4	298	99,33%	0	0,00%	2	0,67%
5	295	98,33%	0	0,00%	5	1,67%
6	257	85,67%	0	0,00%	2	0,67%
7	297	99,00%	0	0,00%	3	1,00%
8	44	14,67%	4	1,33%	6	2,00%
9	295	98,33%	0	0,00%	5	1,67%
10	297	99,00%	0	0,00%	3	1,00%
11	296	98,67%	0	0,00%	4	1,33%



51 (6 e 8)	287	95,67%	0	0,00%	0	0,00%
12	0	0,00%	0	0,00%	300	100,00%
13	0	0,00%	0	0,00%	300	100,00%
14	0	0,00%	0	0,00%	300	100,00%
15	0	0,00%	0	0,00%	300	100,00%
16	0	0,00%	0	0,00%	300	100,00%
17	0	0,00%	0	0,00%	300	100,00%
18	0	0,00%	0	0,00%	300	100,00%
Total:	3.258	60,33%	4	0,07%	2.138	39,59%
Total não rejeitado	3.258	98,73%	4	0,12%		

Observando os resultados obtidos para o multiclasse e os CCUs nas tabelas 9 e 10 e na figura 16, comprova-se a incapacidade do classificador multiclasse em classificar falhas não previstas. Em contrapartida, os CCUs mantêm seus desempenhos quando definem os dados não previstos como falhas desconhecidas e assim obtêm a superioridade para a classificação com dados desconhecidos neste circuito.

#### **6.1.6. Comparação com o multiclasse**

No classificador multiclasse houve um desempenho excelente para todos os tipos de representação de dados na classificação do circuito Sallen-key; onde o pior caso, no LPP, houve um erro de apenas 0,55% na classificação, ou 18,15 de 3300 circuitos avaliados. Para o caso do Sallen-Key, nenhum resultado do multiclasse foi inferior ao resultado do comitê de classe única na respectiva representação dos dados. O que mostra que, para o caso de circuitos com classes de falhas bem definidas e com pouca sobreposição, o desempenho do multiclasse é excelente e não pode ser superado pelo comitê de classe única.

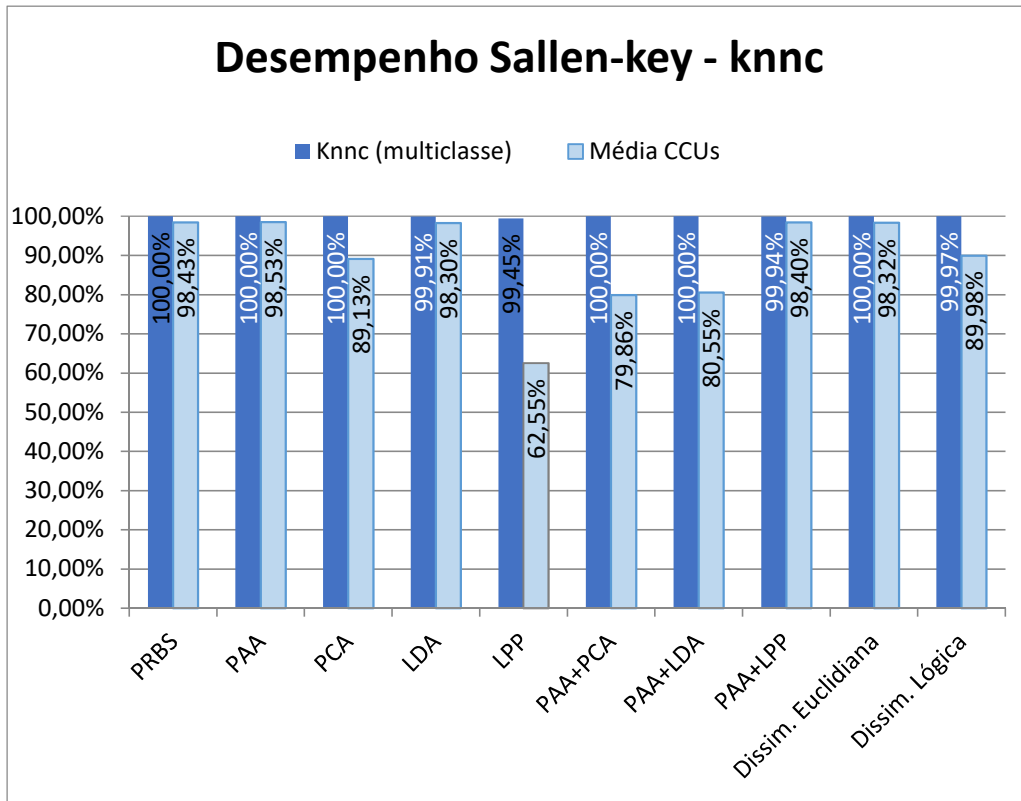


Figura 17: Desempenho multiclasse x CCUs, Sallen-Key

## 6.2. Filtro Universal (CTSV)

### 6.2.1. Desempenho

Este é um circuito projetado com 3 amplificadores operacionais em sua parte ativa, 7 resistores e 2 capacitores, totalizando 9 componentes passivos e 12 componentes no total. A sua arquitetura é mostrada na figura 18 e sua função de transferência é definida na equação (54). Depois do Sallen-Key, este é um dos principais circuitos utilizados no estudo de detecção de falhas pelos mesmos motivos do Sallen-Key. Este circuito também é utilizado como benchmark no trabalho realizado por (KALPANA, P., GUNAVATHI, K., 2009).

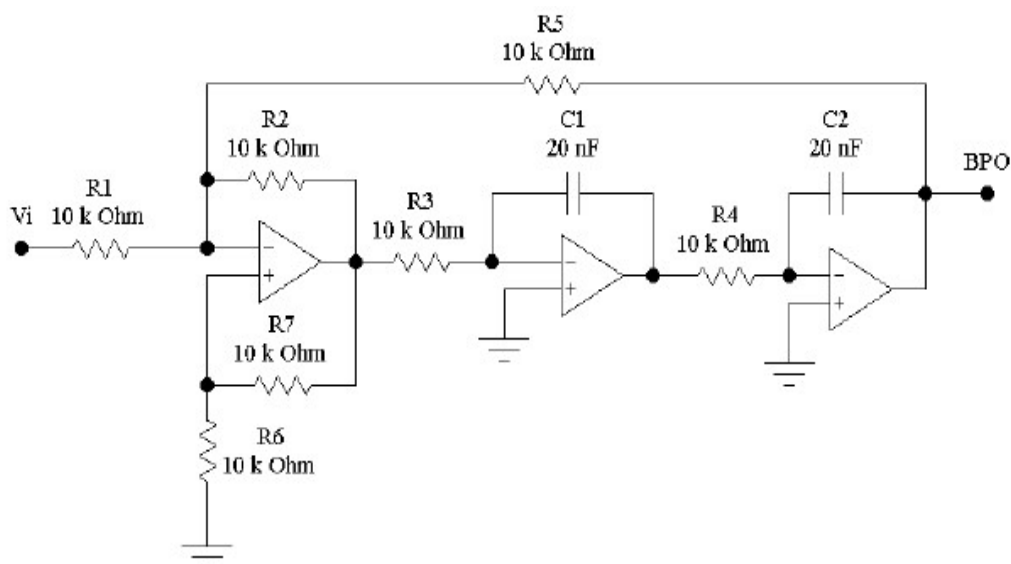


Figura 18: Filtro Universal (CTSV)

$$V_{BPO} = K \cdot \frac{\frac{s}{R_3 C_1}}{s^2 + \frac{1}{Q R_3 C_1} s + \frac{1}{R_3 C_1 R_4 C_2}} \quad (54)$$

O circuito CTSV apresentou muita estabilidade e similaridade nas suas respostas em relação às variações dos parâmetros do circuito, o que dificulta a detecção das falhas para estes parâmetros e, por esta razão, foram

analisadas as classes de falhas de todos os componentes, totalizando 19 classes de falhas:

*Tabela 11: Componentes com falha e suas respectivas classes, CTSV*

Componente	Falha Associada	Respectiva Classe
R1	R1+	1
	R1-	2
R2	R2+	3
	R2-	4
R3	R3+	5
	R3-	6
R4	R4+	7
	R4-	8
R5	R5+	9
	R5-	10
R6	R6+	11
	R6-	12
R7	R7+	13
	R7-	14
C1	C1+	15
	C1-	16
C2	C2+	17
	C2-	18
NoFail		19

Assim como no Sallen-Key, um classificador multiclasse *knn* foi utilizado para fins de comparação e análise dos resultados fornecidos pelo classificador de classe única. A tabela 12 mostra os resultados obtidos pelo classificador multiclasse para a detecção de falhas no circuito CTSV.

Tabela 12: Resultado de classificação multiclasse, CTSV

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de Erro
1	300	100,00%	0	0,00%
2	300	100,00%	0	0,00%
3	300	100,00%	0	0,00%
4	300	100,00%	0	0,00%
5	190	63,33%	113	37,67%
6	169	56,33%	126	42,00%
7	147	49,00%	139	46,33%
8	160	53,33%	106	35,33%
9	300	100,00%	0	0,00%
10	300	100,00%	0	0,00%
11	300	100,00%	0	0,00%
12	300	100,00%	0	0,00%
13	300	100,00%	0	0,00%
14	300	100,00%	0	0,00%
15	187	62,33%	110	36,67%
16	174	58,00%	131	43,67%
17	161	53,67%	153	51,00%
18	194	64,67%	140	46,67%
19	300	100,00%	0	0,00%
Total:	4.682	82,14%	1.018	17,86%

Os resultados dos comitês de classe única na classificação do circuito CTSV são mostrados adiante na figura 19. Os classificadores que apresentaram melhor desempenho foram o gaussiano, no caso sem sobreposição, e o *knn* para o caso com sobreposição. Os resultados mostrados referem-se ao desempenho na classificação para a sobreposição de até duas classes. O gráfico analisado mostra os resultados para o conjunto de dados de sinal bruto, ou seja, dados de resposta à excitação ao PRBS.

É perceptível o baixo desempenho na classificação do comitê de classe única no circuito CTSV, nos casos sem a sobreposição de 2 classes. Isso se deve à existência de classes extremamente sobrepostas neste circuito, algo que não pode ser bem percebido na comparação de desempenho entre os comitês e o multiclasse, pois no segundo caso a diferença de desempenho é menor do que nos CCUs. Isto acontece com menor severidade para o multiclasse porque este tipo de classificador não aceita sobreposições e precisa entregar um único resultado, o que neste caso acaba por inferir corretamente a maior parte das classes, embora de forma totalmente sorteada e, ainda assim, seu desempenho também é claramente prejudicado. Além disso, a análise mostra um melhor desempenho dos CCUs frente ao multiclasse para esses casos onde existem a sobreposição de classes, já que os CCUs permitem a existência da sobreposição, enquanto o multiclasse não.

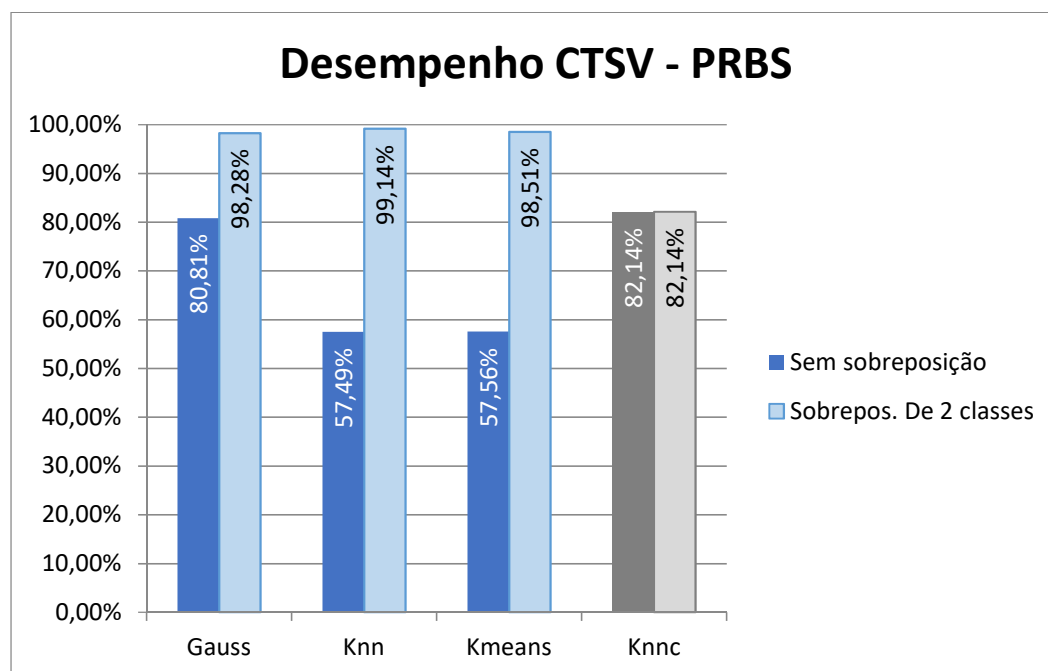


Figura 19: Desempenho para o CTSV no PRBS (o classificador Parzen não conseguiu classificar)

O classificador que obteve o melhor resultado neste caso foi o *knn*. A tabela 13 a seguir mostra os resultados do classificador com o melhor desempenho anterior com relação à distribuição dos dados de cada classe. É possível observar uma forte correlação entre os grupos de classes (5,15),

(6,16), (7,17) e (8,18), indicando que a possível confusão de classes está entre estes pares.

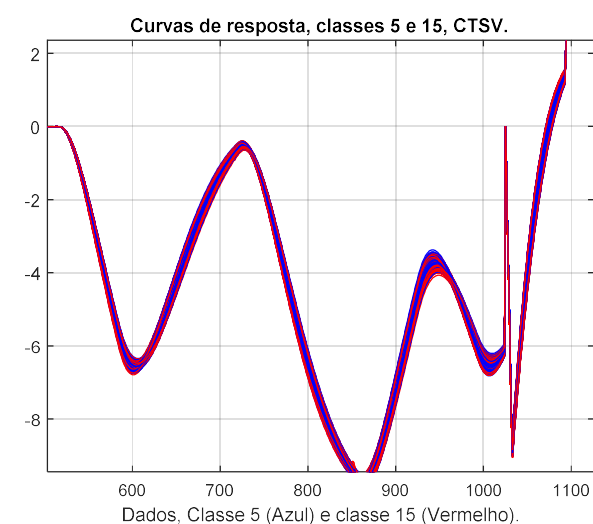
*Tabela 13: Distribuição dos dados para o melhor classificador do CTSV para a curva PRBS*

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de desconhecidos
1	298	99,33%	0	0,00%	2	0,67%
2	299	99,67%	0	0,00%	1	0,33%
3	293	97,67%	0	0,00%	7	2,33%
4	299	99,67%	0	0,00%	1	0,33%
5	0	0,00%	0	0,00%	0	0,00%
6	0	0,00%	0	0,00%	0	0,00%
7	0	0,00%	1	0,33%	0	0,00%
8	0	0,00%	2	0,67%	1	0,33%
9	295	98,33%	0	0,00%	5	1,67%
10	300	100,00%	0	0,00%	0	0,00%
11	295	98,33%	0	0,00%	5	1,67%
12	295	98,33%	0	0,00%	5	1,67%
13	298	99,33%	0	0,00%	2	0,67%
14	295	49,17%	0	0,00%	5	1,67%
15	2	0,33%	0	0,00%	1	0,33%
16	5	1,67%	2	0,67%	0	0,00%
17	6	2,00%	1	0,33%	2	0,67%
18	0	0,00%	1	0,33%	2	0,67%
19	297	99,00%	0	0,00%	3	1,00%
16419 (5 e 15)	597	99,50%	0	0,00%	0	0,00%
8219 (6 e 16)	593	98,83%	0	0,00%	0	0,00%
4119 (7 e 17)	590	98,33%	0	0,00%	0	0,00%
2069 (8 e 18)	594	99,00%	0	0,00%	0	0,00%
Total:	5.651	99,14%	7	0,12%	42	0,74%

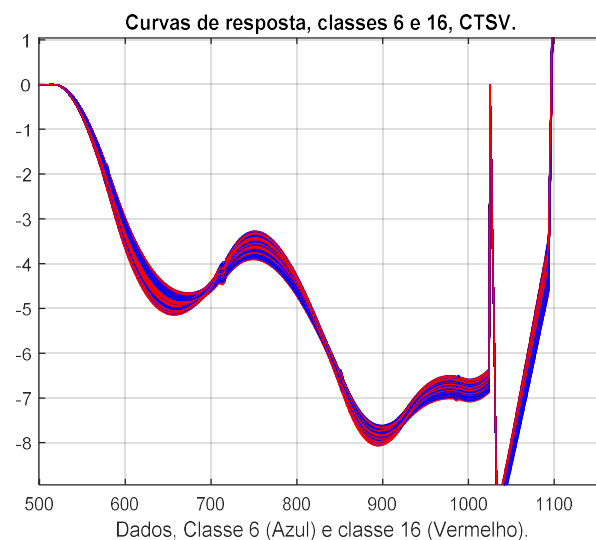
### **6.2.2. Sobreposição de classes**

Da última tabela, é possível notar claramente que os dados entre as classes 5 e 15, 6 e 16, 7 e 17 e 8 e 18 são os responsáveis pelo desempenho inferior ao se classificar sem a tolerância de sobreposição. Os gráficos das

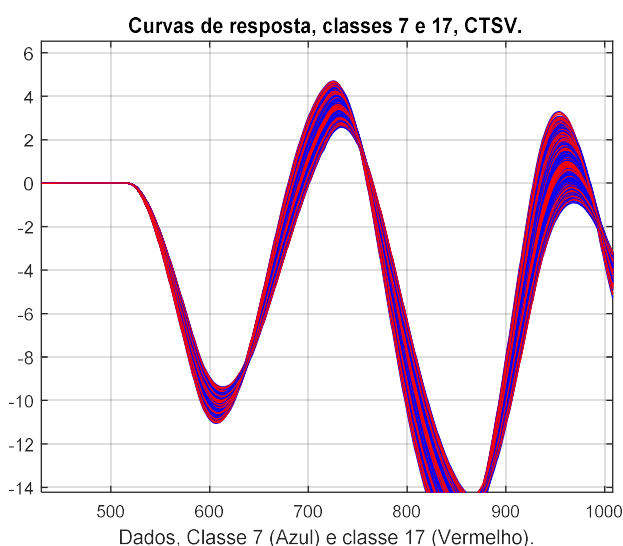
figuras 20 confirmam esta afirmação. Eles mostram todas as curvas das trezentas respostas dos dados para o circuito CTSV nas respectivas classes identificadas e seus correspondentes de sobreposição, no mesmo gráfico. Estes gráficos foram feitos baseados nas informações obtidas anteriormente e, como pode ser observado, a diferença entre os elementos de cada par de dados é praticamente nula.



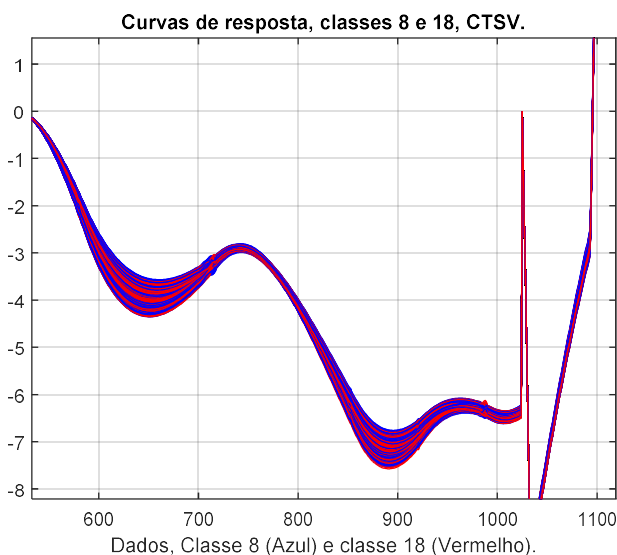
(A) Sobreposição entre as classes 5 e 15



(B) Sobreposição entre as classes 6 e 16



(C) Sobreposição entre as classes 7 e 17



(D) Sobreposição entre as classes 8 e 18

Figura 20: Curvas de resposta das classes sobrepostas, CTSV

Dessa forma, observa-se que não há como distinguir os respectivos pares de classes de classes 5 entre si, sendo válida a união entre as classes



ou, em nosso caso, a tolerância da sobreposição de 2 classes no comitê de classe única, que se mostra superior à classificação do multiclasse, com um desempenho de 99,21% para o melhor classificador contra 82,14% do multiclasse (figura 19). Isso equivale a dizer que, quando tolerada a sobreposição de até 2 classes, o comitê *knn* erra apenas 45 classificações entre 5700, enquanto no multiclasse temos a classificação incorreta de 1018 dados para este circuito, ou uma média de 2,37 dados incorretamente classificados para cada 300 dados inseridos por classe no CCU do *knn*, contra 53,58 dados incorretos para cada 300 dados inseridos por classe no *knn* multiclasse. Isto mostra o melhor desempenho do classe única frente a um multiclasse para a detecção de falhas em circuitos para os casos de sobreposição entre classes mais severas.

### 6.2.3. Influência da representação de dados na sobreposição

Assim como no caso do Sallen-key, foi observado o comportamento de cada classificador para cada tipo diferente de método de representação de dados avaliado. A figura 21 mostra o desempenho do classificador gaussiano no circuito CTSV para cada método de representação de dados utilizado.

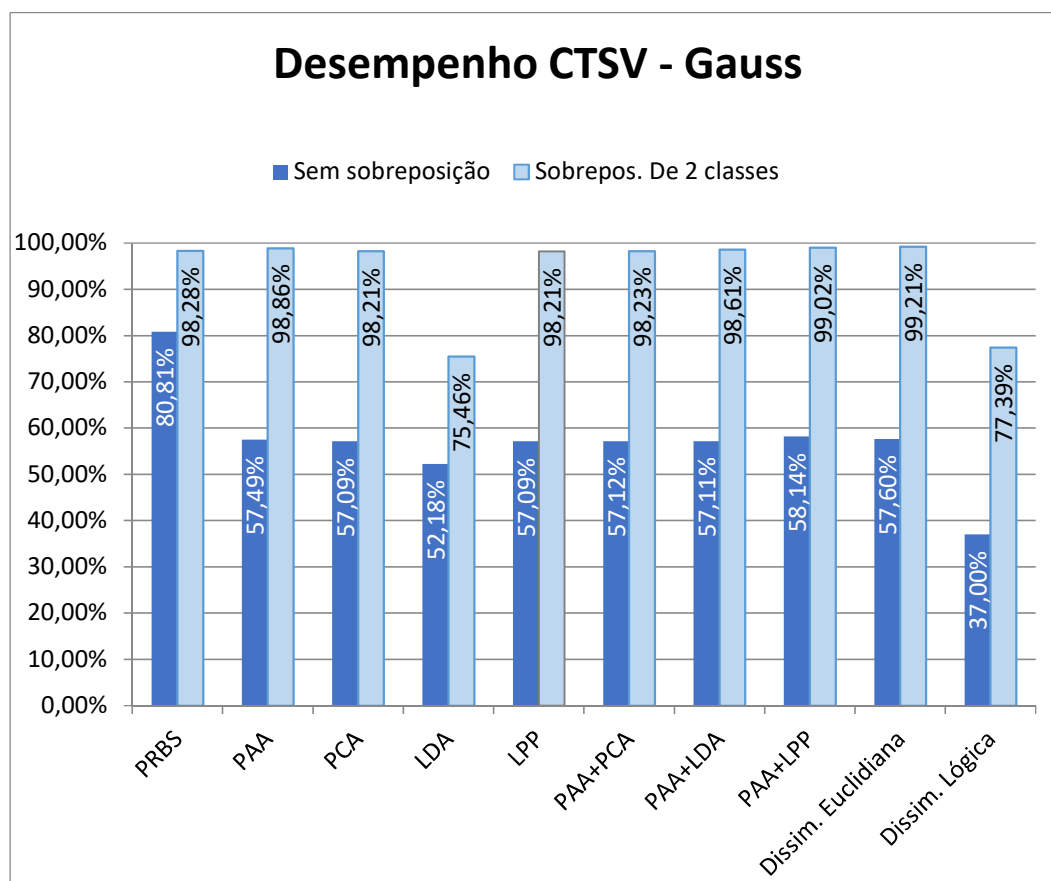


Figura 21: Desempenho por representação de dados, classificador Gauss, CTSV

Com exceção do método de dissimilaridade lógica e do LDA, todos os métodos de representação obtiveram resultados semelhantes de classificação para o classificador gaussiano no CTSV, tanto nas comparações entre os casos sem sobreposição quanto entre os casos com sobreposição. Para o sinal bruto PRBS, houve a melhor classificação geral entre os CCUs para casos sem a sobreposição de classes, com 80,81% de acerto. Já quando é considerada a sobreposição, o desempenho obtido no sinal bruto foi de 98,28% para este classificador.

No caso da representação pelo PAA, o desempenho foi próximo ao apresentado pela maioria dos outros casos, com exceção do *gauss\_PRBS*, sendo 23,32% inferior ao sinal bruto sem sobreposição e 0,58% superior ao sinal bruto com sobreposição.

Para o caso dos métodos de redução de dimensão, o desempenho pelos métodos PCA e LPP foi similar aos obtidos com os outros métodos, em ambos os casos de classificação. A representação por PCA foi 23,72% inferior ao sinal bruto no caso sem sobreposição e praticamente equivalente ao sinal bruto no caso com sobreposição, sendo apenas 0,07% inferior ao sinal bruto. Para o caso do LPP, o classificador gaussiano obteve um resultado idêntico ao PCA, para o caso sem sobreposição, e 0,02% maior que o PCA no caso com sobreposição. Por sua vez, o LDA apresentou um desempenho inferior aos métodos de redução anteriores, sendo 28,63% inferior ao desempenho obtido com o sinal bruto, no caso sem sobreposição, e 22,82% inferior em relação ao caso com sobreposição.

No caso da combinação entre os métodos PAA+DR, os resultados foram equivalentes a maioria dos casos sem e com sobreposição do circuito CTSV. Para o PAA+PCA, a diferença para o sinal bruto foi de 23,69% sem sobreposição, no PAA+LDA, essa diferença foi de 23,70% e para o PAA+LPP essa diferença foi de 22,67%. Para os mesmos métodos de representação aplicados ao caso com sobreposição, foi praticamente equivalente para o PAA+PCA, 0,33% para o PAA+LDA e de 0,74% para o PAA+LPP, com relação ao sinal bruto.

Para as representações por dissimilaridade, apenas a dissimilaridade euclidiana teve um resultado satisfatório. O seu resultado foi equivalente ao obtido com a maioria dos outros métodos, com uma diferença de 23,21% para o caso sem sobreposição e de 0,93% para o caso com sobreposição, com relação ao sinal bruto, sendo este o caso que apresentou o melhor desempenho com sobreposição. A dissimilaridade lógica apresentou o pior desempenho para o caso sem sobreposição, sendo o seu desempenho 43,81% inferior ao do sinal bruto neste caso. Para o caso com sobreposição, apresentou o segundo pior resultado, à frente apenas do LDA, com uma diferença de 20,89% para o desempenho do sinal bruto.

No caso do *knn* (figura 22), com exceção do LDA e da combinação PAA+LPP, todos os métodos apresentaram desempenho similar e que apresentaram os melhores resultados para cada caso. Para o sinal bruto, o desempenho obtido foi de 57,49%, no caso sem sobreposição, e 99,14% para o caso com sobreposição. No método do PAA, houve a equivalência na classificação com relação aos resultados no sinal bruto, tendo um pequeno aumento de 0,07% para o caso sem sobreposição e de 0,05% no caso com sobreposição.

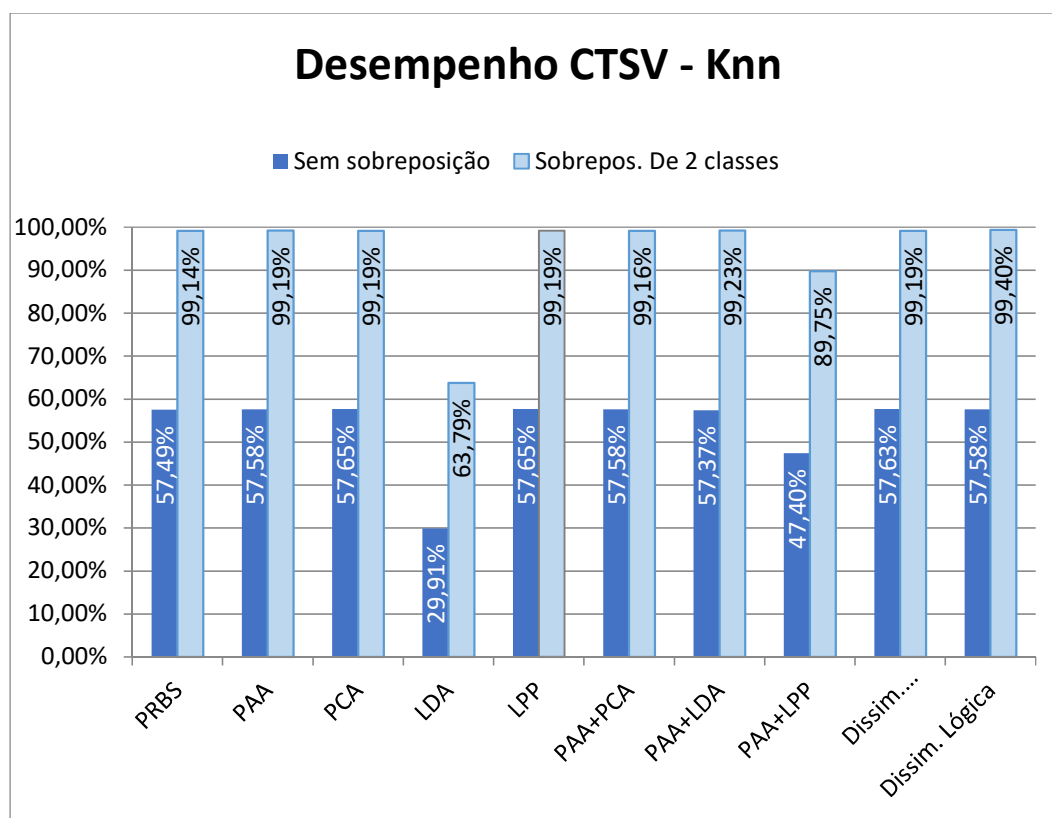


Figura 22: Desempenho por representação de dados, classificador *knn*, CTSV

Nos métodos de redução de dimensão, os métodos PCA e LPP tiveram os mesmos resultados para os dois casos de sobreposição, com um aumento de 0,16% do desempenho sem sobreposição e de 0,05% nos casos com sobreposição, o LDA não obteve novamente um resultado satisfatório, aumentando o número de sobreposições em 27,58% e tendo um desempenho 35,35% inferior no caso com sobreposição com relação ao sinal bruto.

Para os métodos combinados PAA+DR, não houve diferença na combinação PAA+PCA com o método PAA e PCA individualmente, para o

caso do desempenho sem sobreposições, neste caso foi apenas 0,07% superior ao sinal bruto e com sobreposição foi praticamente equivalente ao sinal bruto. Para o caso da combinação PAA+LDA, houve uma redução de 0,12% para o sinal bruto e 0,21% para o desempenho no PAA, mas uma melhora do desempenho com relação ao método LDA individualmente em 27,46%, no caso do desempenho sem sobreposição. Para o desempenho com sobreposição, o PAA+LDA obteve um desempenho 0,09% superior ao desempenho com o sinal bruto, sendo praticamente equivalente ao desempenho do PAA, e uma melhora de 35,44% com relação ao LDA individualmente. Os métodos PAA+LPP não obtiveram uma potencialização de desempenho neste caso, tendo apresentado um desempenho aproximadamente 10% inferior ao do sinal bruto e aos métodos PAA e LPP individualmente, em ambos os casos de sobreposição.

Nos métodos de representação por dissimilaridade, os resultados foram satisfatórios em ambos os casos. Para a dissimilaridade euclidiana, houve um aumento de 0,14% de desempenho para o caso sem sobreposição, e um resultado equivalente no caso com sobreposição, comparado aos resultados obtidos com o PRBS. Na dissimilaridade euclidiana, o resultado obtido para o caso sem sobreposição foi 0,09% superior ao mesmo caso no sinal bruto e com sobreposição o desempenho obtido com este classificador foi o melhor entre todos os métodos avaliados, sendo 0,26% superior à classificação do sinal bruto, com 99,40% de desempenho. Isso representa um erro de apenas 0,78 circuitos por classe, ou em outras palavras, 0,78 circuitos errados a cada 300 circuitos analisados.

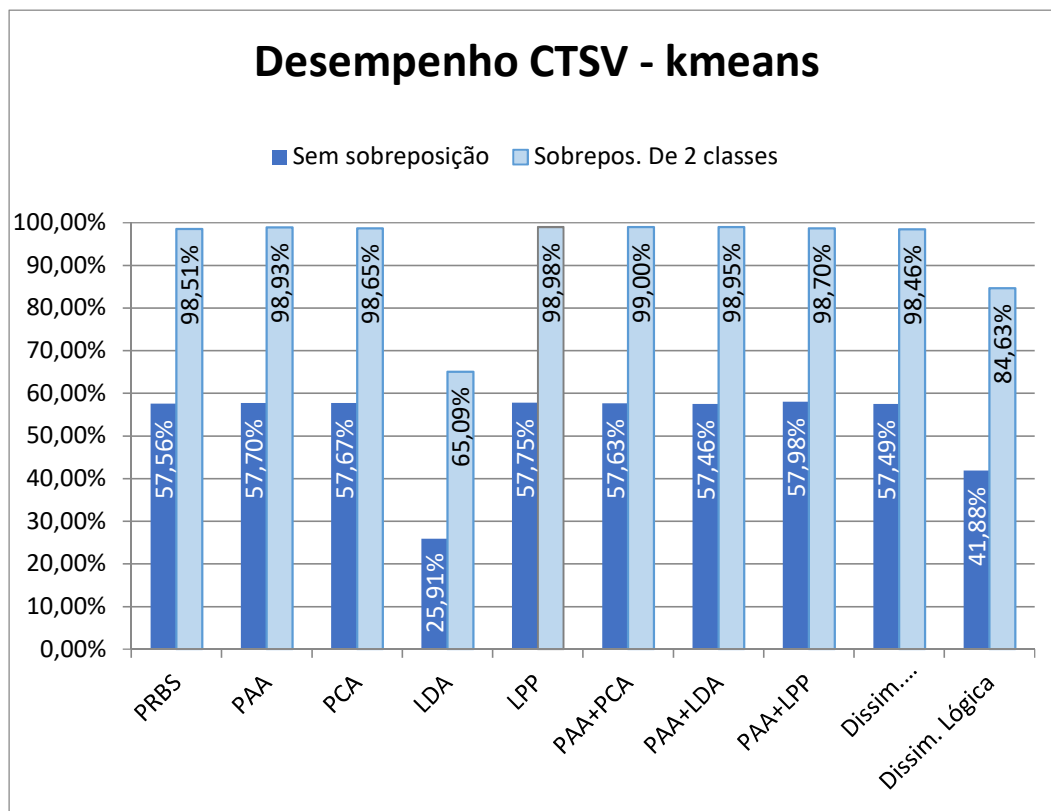


Figura 23: Desempenho por representação de dados, classificador *kmeans*, CTSV

Para o classificador de reconstrução *kmeans* (figura 23), assim como no caso do *knn* e do *Gauss*, a maior parte dos resultados foi equivalente, com exceção do LDA e, e da dissimilaridade lógica, em ambos os casos de sobreposição. No PAA, obteve-se uma classificação sem sobreposição superior em 0,14% ao sinal bruto PRBS e uma diferença de 0,42% da classificação PAA para o sinal bruto com sobreposição.

Nos métodos de redução de dimensão, a representação por PCA apresenta uma melhora de classificação de 0,11% para os casos sem sobreposição e, para o caso com sobreposição, o PCA teve uma melhora de 0,14% com relação à utilização do sinal bruto. A representação pelo método LPP apresentou um aumento de desempenho em 0,19% com relação ao sinal bruto na classificação sem sobreposições e, para o caso com sobreposições, obteve um resultado 0,47% superior à classificação com sinal bruto. O método de redução pelo LDA não obteve resultados satisfatórios também para este classificador, o que indica que este pode não ser um bom método para a representação de dados das classes deste circuito. Para o *kmeans*, o método LDA obteve um desempenho 31,65% inferior ao sinal bruto sem sobreposição

e 33,42% inferior ao sinal bruto no caso com sobreposição. Por sua vez, para os dos métodos de representação combinados, todos os casos obtiveram um desempenho satisfatório, em ambos os casos de sobreposição, no PAA+PCA obtivemos um desempenho 0,07% superior ao sinal bruto na classificação sem sobreposição e 0,49% superior ao sinal bruto na classificação com sobreposição. No caso do PAA+LDA, há uma perda no desempenho para o caso sem sobreposição, sendo 0,1% inferior ao sinal bruto, no caso da classificação com sobreposição, o desempenho obtido foi 0,44% superior ao desempenho com o sinal bruto. A combinação PAA+LPP teve o melhor desempenho no caso sem sobreposição, com 0,42% de melhora no desempenho com relação à classificação com o sinal PRBS, representando a detecção correta de até 24 mais falhas. No caso com sobreposição, teve uma melhora de 0,19% no desempenho com relação ao sinal bruto.

Para os casos das representações por dissimilaridade, somente para a dissimilaridade euclidiana foram obtidos bons resultados. Nesta representação, o desempenho foi inferior à classificação com o sinal bruto em 0,07%, sem sobreposição, e apenas 0,05% a menos, considerando as sobreposições. Na dissimilaridade lógica, houve uma perda de desempenho para este classificador, apresentando um desempenho 15,68% inferior para a classificação sem sobreposição e 13,88% inferior para a classificação com sobreposição, quando comparado ao desempenho obtido na classificação com o sinal bruto.

Para o classificador de *Parzen* (figura 24), obteve-se um desempenho insatisfatório para vários métodos de representação de dados, a começar pelo próprio sinal bruto, este classificador não foi capaz de distinguir quaisquer classes para a classificação no sinal PRBS. Ao utilizar a extração de assinaturas PAA, o a classificação consegue ser realizada, obtendo um desempenho de 57,42% para o caso sem sobreposição e 98,95% para o caso com sobreposição. Nos métodos de redução de dimensão, o LDA novamente apresenta resultados insatisfatórios para este circuito. No caso do PCA e do LPP, ambos apresentam os mesmos resultados tanto para a classificação sem sobreposições quanto com sobreposições, com um desempenho de 57,26% para o primeiro caso e 97,84% para o segundo caso. Na representação pela combinação de métodos, o *Parzen* consegue classificar

satisfatoriamente dois dos três casos: o PAA+PCA e o PAA+LDA, mostrando um bom desempenho ao se combinar os métodos. No PAA+PCA, o desempenho foi de 57,46% para a classificação sem sobreposições e de 99,05% para a classificação com sobreposições, em ambos os casos. Estes foram os melhores resultados obtidos para este classificador, tendo sido melhor que o PAA em 0,04% e 0,1%, e melhor que o PCA em 0,2% e 1,21%. No PAA+LDA, os resultados obtidos foram de 52,60% e 98,86% para a classificação sem e com sobreposição, respectivamente, sendo inferior ao desempenho do PAA em 5,36% e 0,09% e superior ao desempenho do LDA em 51,09% e 62,88%. Para o caso da combinação PAA+LPP, os resultados foram insatisfatórios.

Para os métodos de representação por dissimilaridade, o método de dissimilaridade euclidiana não pôde ser classificado no *Parzen* também neste circuito, o que pode indicar a incompatibilidade deste método para este classificador. Já a dissimilaridade lógica, obteve desempenho de 54,16% na classificação sem sobreposição e de 96,26% na classificação com sobreposição.

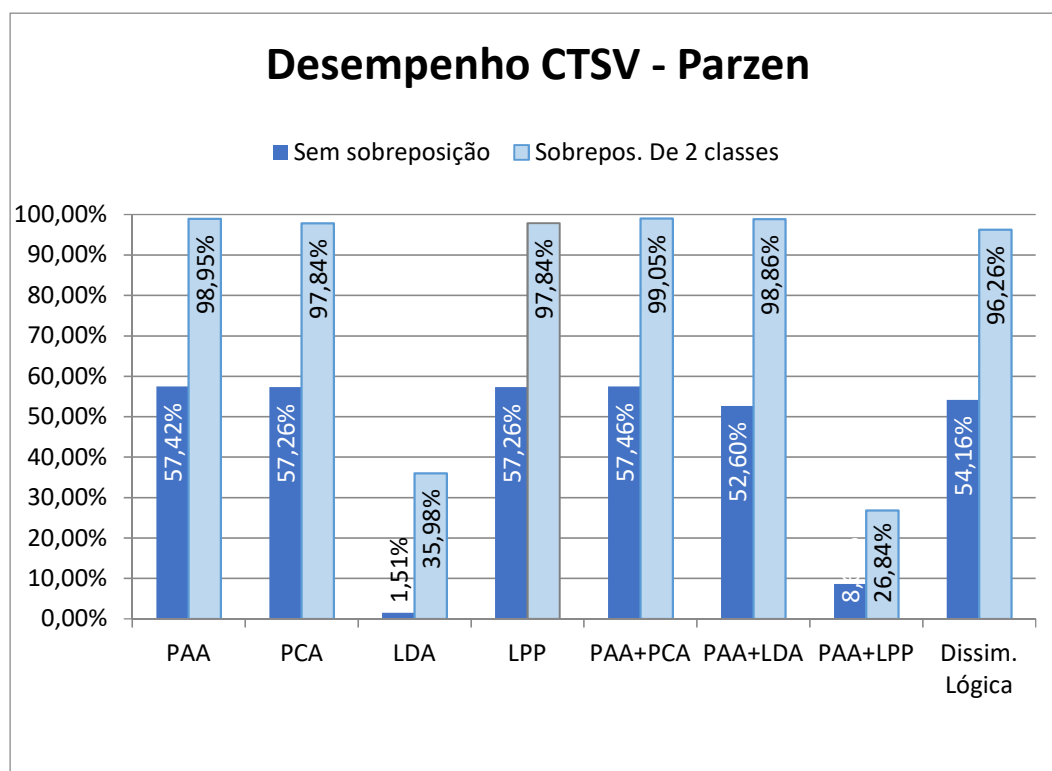


Figura 24: Desempenho por representação de dados, classificador Parzen, CTSV



#### 6.2.4. Desempenho para a representação de dados

A análise seguinte teve por objetivo verificar se, executando tratamento nos dados com métodos de redução de dimensão e extração de assinaturas, seria possível obter uma maior discriminação dos dados, tendo uma melhor distinção entre as classes como consequência e assim, resolvendo o problema de sobreposição de classes. Esta etapa foi de extrema importância neste circuito, pois permite que se verifique se as classes em sobreposição deste circuito são ou não separáveis utilizando métodos de representação. Caso seja possível a separação das classes deste circuito possivelmente para os outros três circuitos testados neste trabalho seja possível a separação de qualquer sobreposição de classes, através destes métodos.

A figura 25 a seguir mostra os resultados de desempenho de classificação de resposta do CTSV pós extração de atributos.

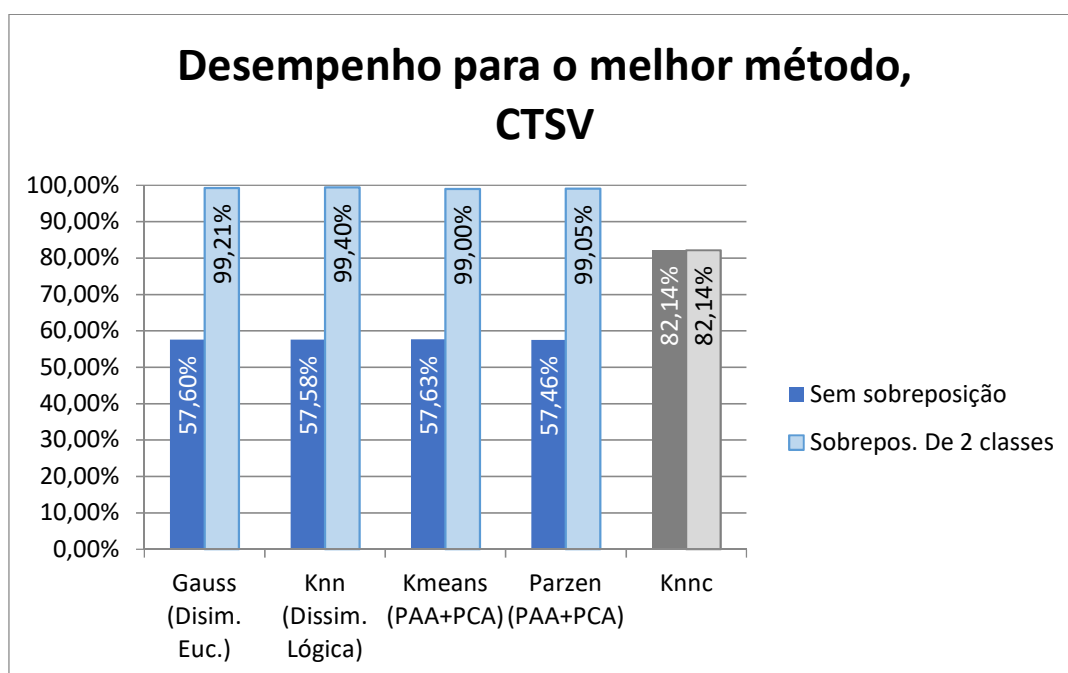


Figura 25: Curvas de desempenho dos melhores métodos de representação, CTSV

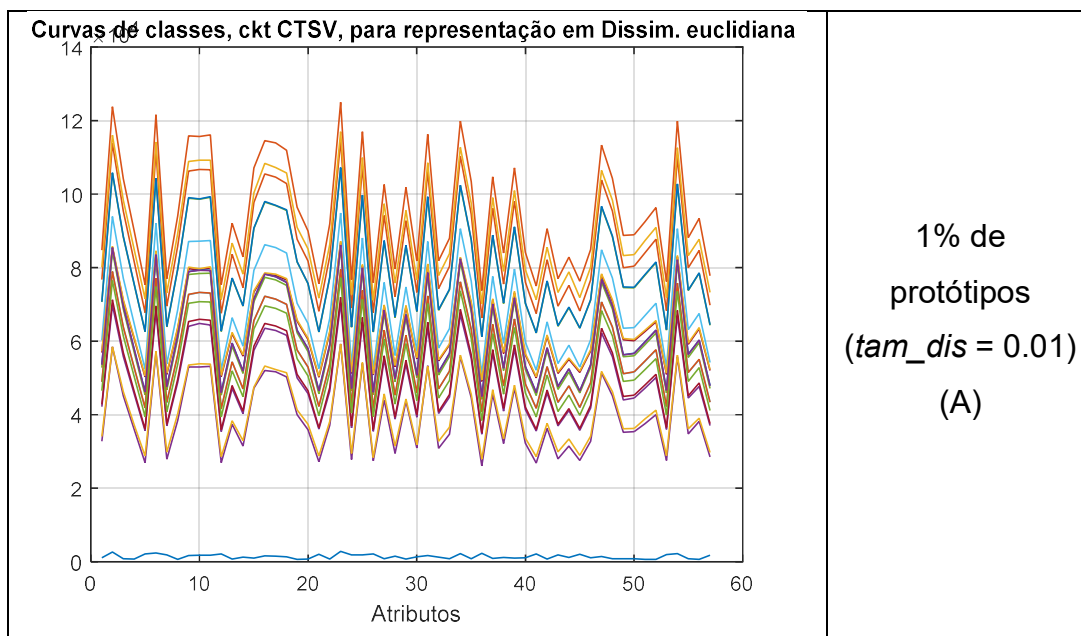
No caso deste circuito, o classificador que obteve o melhor resultado com a representação de dados foi o *knn* com a dissimilaridade lógica. A tabela 14 mostra o desempenho por classe deste classificador para a melhor representação de dados identificada.

*Tabela 14: Distribuição dos dados para o melhor classificador do CTSV no melhor método de representação de dados*

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de desconhecidos
1	300	100,00%	0	0,00%	0	0,00%
2	298	99,33%	0	0,00%	2	0,67%
3	297	99,00%	0	0,00%	3	1,00%
4	300	100,00%	0	0,00%	0	0,00%
5	0	0,00%	1	0,33%	0	0,00%
6	0	0,00%	0	0,00%	0	0,00%
7	1	0,33%	0	0,00%	0	0,00%
8	1	0,33%	0	0,00%	2	0,67%
9	297	99,00%	0	0,00%	3	1,00%
10	300	100,00%	0	0,00%	0	0,00%
11	298	99,33%	0	0,00%	2	0,67%
12	298	99,33%	0	0,00%	2	0,67%
13	295	98,33%	0	0,00%	5	1,67%
14	291	48,50%	0	0,00%	9	3,00%
15	0	0,00%	0	0,00%	0	0,00%
16	3	1,00%	4	1,33%	0	0,00%
17	3	1,00%	0	0,00%	0	0,00%
18	0	0,00%	1	0,33%	0	0,00%
19	300	100,00%	0	0,00%	0	0,00%
16419 (5 e 15)	599	99,83%	0	0,00%	0	0,00%
8219 (6 e 16)	593	98,83%	0	0,00%	0	0,00%
4119 (7 e 17)	596	98,33%	0	0,00%	0	0,00%
2069 (8 e 18)	596	99,00%	0	0,00%	0	0,00%
Total:	5.666	99,40%	6	0,11%	28	0,49%

As curvas de desempenho da figura 25 e a tabela 14 demonstram o potencial dos métodos de representação de dados para os problemas de classificação. Repare como, embora para os classificadores de classe única tenha havido uma indiferença ou redução de desempenho com relação a classificação sem sobreposição, o desempenho com sobreposição dispara e alcança níveis muito satisfatórios de desempenho.

Nos métodos de classe única, houve uma perda de desempenho no caso gaussiano, considerando a sua resposta sem sobreposição, a indiferença do caso sem sobreposição para os métodos *knn*, *kmeans* e *Parzen*, e uma pequena melhor desempenho de todos os métodos citados, considerando as sobreposições, comparadas às suas respectivas respostas para o sinal PRBS (figura 19). A possível perda de desempenho do método de *Gauss* para o caso sem sobreposição é atribuída ao fato do desempenho deste tipo de classificador (estimativa de densidade) estar relacionado com a qualidade do modelo estatístico dos dados a ele submetidos. Os métodos de representação por dissimilaridade são capazes de oferecer bons modelos, mas isso também vai depender do número de atributos que serão analisados. Observe os exemplos da figura 26: No lado esquerdo temos a representação dos dados em dissimilaridade euclidiana do CTSV que foi aplicado ao classificador gaussiano, com somente 1% do número de respostas PRBS ao CTSV de atributos, no lado direito, temos os mesmos dados para a mesma representação, mas com o número de atributos extraídos de 30% do número de respostas PRBS ao CTSV. Com o número maior de amostras, temos maiores variações entre os atributos, conseqüentemente, uma maior probabilidade de separação entre as classes. Resumindo, quanto maior o número de amostras, melhor o modelo.



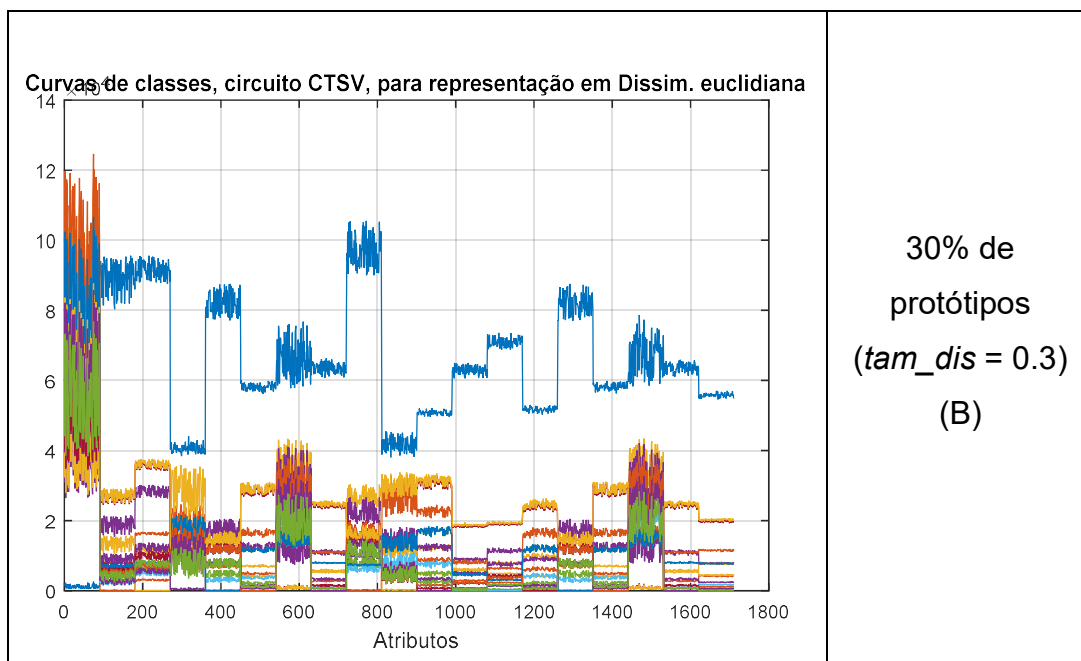


Figura 26: Representação das respostas do circuito CTSV em dissimilaridade Euclidiana, para 1% de protótipos (A) e 30% de protótipos (B)

Além disso, a sobreposição muito acentuada entre os dados das classes analisadas dificulta o processo de classificação para todos os casos, o que explica a indiferença da classificação sem sobreposições tanto para o *Gauss* quanto para os classificadores *knn*, *kmeans* e *parzen*. Os gráficos de sobreposição de classes do CTSV encontrados no anexo VI mostram os dados após cada método de representação no CTSV. Ao analisá-lo, será observado que, para qualquer método, a proximidade entre as classes sobrepostas continua muito grande. Isso mostra que em alguns casos, certas sobreposições são muito difíceis de se resolver ou não podem ser resolvidas, além da união entre as classes. Considerando a sobreposição, os métodos de representação dos dados conseguem devolver o mesmo nível de resposta da classificação com o sinal bruto, mas utilizando um número menor de informação, cumprindo plenamente vossa função para o circuito CTSV.

### 6.2.5. Desempenho para classes não previstas

Para o circuito CTSV, cada classificador será submetido ao conjunto de dados com 7 classes não previstas para avaliação, sendo 4 conjuntos de falhas múltiplas do CTSV, e 3 classes de falhas totalmente desconhecidas para este circuito. A tabela 15 identifica cada classe de falha não prevista, assim como as já previstas.

*Tabela 15: Representação das classes não previstas, CTSV*

Componente	Falha Associada	Respectiva Classe
R1	R1+	1
	R1-	2
R2	R2+	3
	R2-	4
R3	R3+	5
	R3-	6
R4	R4+	7
	R4-	8
R5	R5+	9
	R5-	10
R6	R6+	11
	R6-	12
R7	R7+	13
	R7-	14
C1	C1+	15
	C1-	16
C2	C2+	17
	C2-	18
NoFail		19
M1	(R3+, C1+)	20
	(R3-, C1-)	21
M2	(R3+, C1-)	22

	(R3-, C1+)	23
	X1	24
	X2	25
	X3	26

Os resultados obtidos para o classificador multiclasse são mostrados na tabela 16. Novamente, o número de dados equivocadamente classificados aumenta proporcionalmente ao número de dados de classes não previstas aplicados para avaliação neste tipo de classificador. O que demonstra que nenhum dado não-treinado pode ser aplicado ao multiclasse.

*Tabela 16: Resultado de classificação multiclasse para falhas não previstas, CTSV*

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de Erro
1	300	100,00%	0	0,00%
2	300	100,00%	0	0,00%
3	300	100,00%	0	0,00%
4	300	100,00%	0	0,00%
5	190	63,33%	113	37,67%
6	169	56,33%	126	42,00%
7	147	49,00%	139	46,33%
8	160	53,33%	106	35,33%
9	300	100,00%	0	0,00%
10	300	100,00%	0	0,00%
11	300	100,00%	0	0,00%
12	300	100,00%	0	0,00%
13	300	100,00%	0	0,00%
14	300	100,00%	0	0,00%
15	187	62,33%	110	36,67%
16	174	58,00%	131	43,67%
17	161	53,67%	153	51,00%

18	194	64,67%	140	46,67%
19	300	100,00%	0	0,00%
20	0	0,00%	300	100,00%
21	0	0,00%	300	100,00%
22	0	0,00%	300	100,00%
23	0	0,00%	300	100,00%
24	0	0,00%	300	100,00%
25	0	0,00%	300	100,00%
26	0	0,00%	300	100,00%
Total:	4.682	60,03%	3.118	39,97%

Adiante, a figura 27 apresenta os gráficos de desempenho de cada classificador para a avaliação com falhas imprevistas no circuito CTSV. Novamente, os CCUs mantêm o desempenho por serem capazes de lidar com as classes não previstas, enquanto o desempenho do multiclasse, diminui.

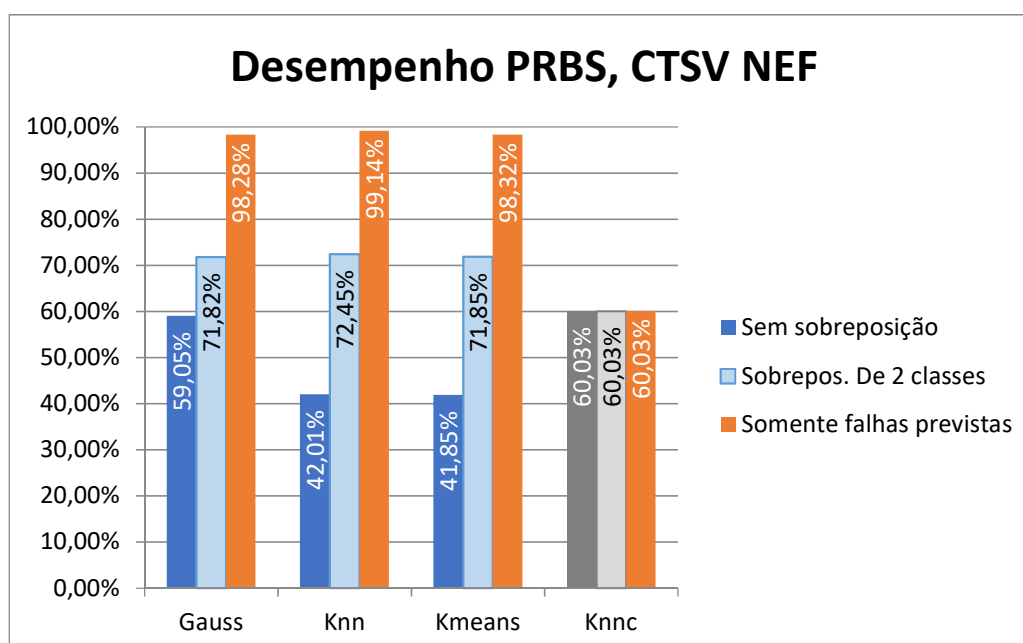


Figura 27: Desempenho dos classificadores para falhas não previstas com sinal PRBS, CTSV (Parzen não classificou)

Para este caso, o classificador que obteve o melhor desempenho foi o *knn*. A tabela 17 mostra os resultados detalhados deste classificador para o tratamento das falhas não previstas.

*Tabela 17: Resultados para o melhor classificador do CTSV na curva PRBS para falhas não previstas*

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de desconhecidos
1	298	99,33%	0	0,00%	2	0,67%
2	299	99,67%	0	0,00%	1	0,33%
3	293	97,67%	0	0,00%	7	2,33%
4	299	99,67%	0	0,00%	1	0,33%
5	0	0,00%	0	0,00%	0	0,00%
6	0	0,00%	0	0,00%	0	0,00%
7	0	0,00%	1	0,33%	0	0,00%
8	0	0,00%	2	0,67%	1	0,33%
9	295	98,33%	0	0,00%	5	1,67%
10	300	100,00%	0	0,00%	0	0,00%
11	295	98,33%	0	0,00%	5	1,67%
12	295	98,33%	0	0,00%	5	1,67%
13	298	99,33%	0	0,00%	2	0,67%
14	295	49,17%	0	0,00%	5	1,67%
15	2	0,33%	0	0,00%	1	0,33%
16	5	1,67%	2	0,67%	0	0,00%
17	6	2,00%	1	0,33%	2	0,67%
18	0	0,00%	1	0,33%	2	0,67%
19	297	99,00%	0	0,00%	3	1,00%
16419 (5 e 15)	597	99,50%	0	0,00%	0	0,00%
8219 (6 e 16)	593	98,83%	0	0,00%	0	0,00%
4119 (7 e 17)	590	98,33%	0	0,00%	0	0,00%
2069 (8 e 18)	594	99,00%	0	0,00%	0	0,00%
20	0	0,00%	0	0,00%	300	100,00%
21	0	0,00%	0	0,00%	300	100,00%
22	0	0,00%	0	0,00%	300	100,00%
23	0	0,00%	0	0,00%	300	100,00%
24	0	0,00%	0	0,00%	300	100,00%
25	0	0,00%	0	0,00%	300	100,00%



26	0	0,00%	0	0,00%	300	100,00%
Total:	5.651	72,45%	7	0,09%	2.142	27,46%
Total não rejeitado	5.651	99,14%	7	0,12%		

Assim como no caso do circuito Sallen-Key, os CCUs foram estáveis na classificação quando submetidos a dados não previstos no treinamento, enquanto o multiclasse acaba por inferir resultados equivocados para as falhas. É também interessante observar que, as falhas múltiplas utilizadas eram entre dois componentes cujas falhas estavam entrando em sobreposição para o caso das falhas simples. Ainda assim, não houve qualquer similaridade entre as falhas múltiplas e as falhas destes componentes, o que indica uma sensibilidade razoável para as falhas entre estes componentes ou uma correlação entre eles, o que de fato existe.

#### **6.2.6. Comparação com o multiclasse**

No caso da classificação multiclasse, todos os casos foram satisfatórios para o classificador multiclasse, considerando que este tipo de classificador nem sempre consegue lidar com as sobreposições de classes, acertando ou errando na sua escolha para o dado. Para o sinal PRBS, a classificação do circuito CTSV ficou em 82,14%, inferior a todos os casos de classe única em PRBS, com exceção do caso do *Parzen*.

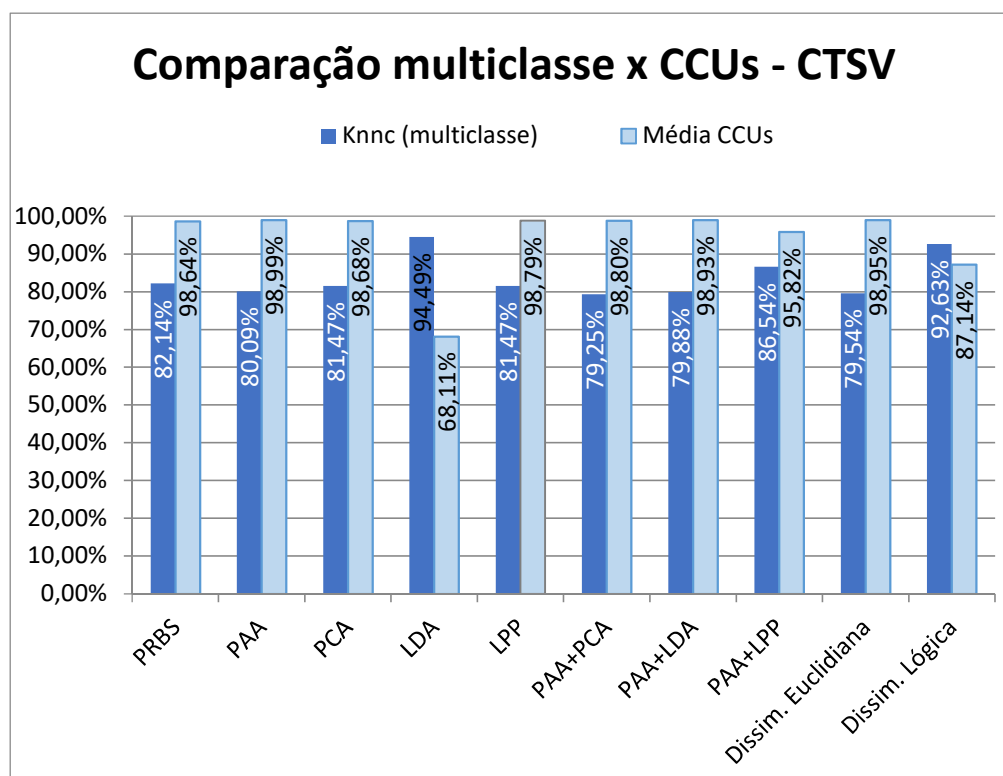


Figura 28: Desempenho multiclasse x CCUs, CTSV

Somente em dois dos dez métodos diferentes de representação de dados houve uma superioridade do classificador multiclasse com relação aos CCUs: no caso da representação em LDA e na representação com a dissimilaridade lógica. Os CCUs, além de terem obtido o melhor desempenho para 80% dos casos avaliados, tiveram um desempenho consideravelmente superior ao multiclasse neste circuito, sendo 16,5% superior para o sinal bruto. Para o melhor dos casos no multiclasse, à exceção dos métodos de representação onde o multiclasse foi superior, a diferença do comitê de classe única para o multiclasse é de 9,28% (PAA+LPP) e para o pior dos casos essa diferença chega a 19,55% (PAA+PCA). Ou seja, os classificadores de classes única identificam corretamente entre 529 a 1115 falhas a mais do que o multiclasse neste circuito.

### 6.3. Filtro Biquad

#### 6.3.1. Desempenho

Este é um circuito passa-altas ativo de 2ª Ordem, o seu projeto básico consiste em 4 amplificadores operacionais de parte ativa do circuito, 10 resistores e 2 capacitores, totalizando 12 componentes passivos e 16 componentes no total. A sua arquitetura é mostrada na figura 29 e sua função de transferência é definida na equação (55). Este circuito também é bastante utilizado como benchmark no estudo de detecção de falhas por sua baixa sensibilidade à variação dos parâmetros passivos do circuito, tendo sido avaliado nos estudos de (XIAO, Y., HE, Y., 2011; XIAO, Y., FENG, L., 2012).

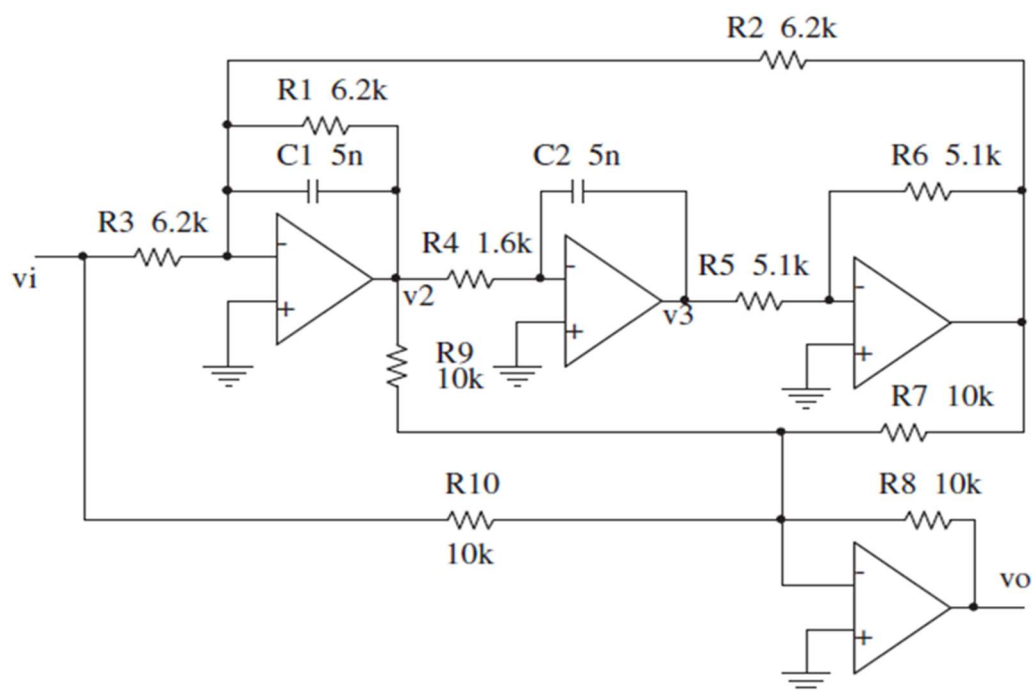


Figura 29: Circuito Biquad

$$V_{BPO} = K \cdot \frac{\frac{s}{R_3 C_1}}{s^2 + \frac{1}{Q R_3 C_1} s + \frac{1}{R_3 C_1 R_4 C_2}} \quad (55)$$

Assim como no primeiro circuito analisado neste trabalho, apesar deste circuito apresentar um total de até 25 falhas simples, o comportamento de

alguns componentes não foi considerado, analisando somente 13 destas falhas. A tabela 18 determina quais os componentes analisados e suas respectivas classes durante a classificação:

*Tabela 18: Componentes com falha e suas respectivas classes, Biquad*

Componente	Falha Associada	Respectiva Classe
R1	R1+	1
	R1-	2
R2	R2+	3
	R2-	4
R3	R3+	5
	R3-	6
R4	R4+	7
	R4-	8
C1	C1+	9
	C1-	10
C2	C2+	11
	C2-	12
NoFail		13

Como mencionado anteriormente, um classificador multiclasse *knn* foi utilizado para fins de comparação e análise dos resultados fornecidos do classificador de classe única. A tabela 19 mostra os resultados obtidos pelo classificador multiclasse para a detecção das falhas no circuito Biquad:

Tabela 19: Resultado de classificação multiclasse, Biquad

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de Erro
1	300	100,00%	0	0,00%
2	300	100,00%	0	0,00%
3	300	100,00%	0	0,00%
4	300	100,00%	0	0,00%
5	300	100,00%	0	0,00%
6	300	100,00%	0	0,00%
7	300	100,00%	0	0,00%
8	205	68,33%	95	31,67%
9	300	100,00%	0	0,00%
10	300	100,00%	0	0,00%
11	299	99,67%	1	0,33%
12	210	70,00%	90	30,00%
13	300	100,00%	0	0,00%
Total:	3.714	95,23%	186	4,77%

A figura 30 apresenta os resultados dos comitês de classe única na classificação do circuito Biquad. Neste caso, o classificador que apresentou o melhor desempenho foi o *Gauss*, para o caso sem sobreposição, e o *knn*, com sobreposição. Os resultados mostrados referem-se ao desempenho na classificação sem tolerância de sobreposição de classes e com tolerância para a sobreposição de até duas classes. Neste caso, os gráficos na figura 30 mostram os resultados para o conjunto de dados de sinal bruto, ou seja, para os dados de resposta à excitação do PRBS.

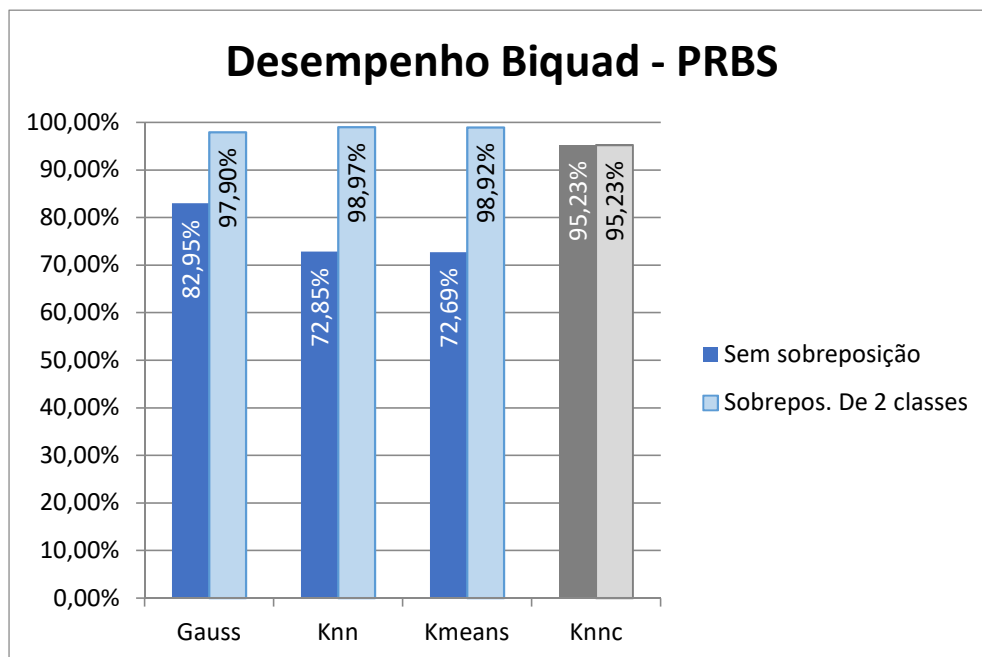


Figura 30: Desempenho para o Biquad no PRBS (O classificador parzen não conseguiu realizar a classificação neste caso)

A tabela 20 mostra os resultados do classificador com o melhor desempenho anterior, com relação à distribuição dos dados em cada classe.

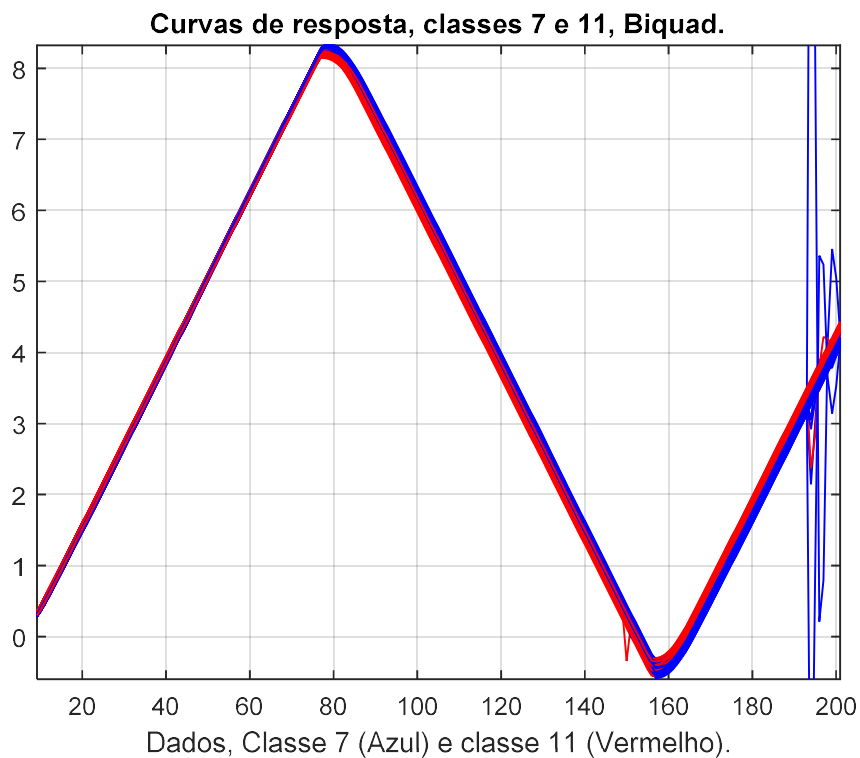
Tabela 20: Distribuição dos dados para o melhor classificador do Biquad para a curva PRBS

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de Desconhecidos
1	298	99,33%	0	0,00%	2	0,67%
2	299	99,67%	0	0,00%	1	0,33%
3	293	97,67%	0	0,00%	7	2,33%
4	299	99,67%	0	0,00%	1	0,33%
5	299	99,67%	0	0,00%	1	0,33%
6	300	100,00%	0	0,00%	0	0,00%
7	162	54,00%	0	0,00%	0	0,00%
8	0	0,00%	0	0,00%	3	1,00%
9	296	98,67%	0	0,00%	4	1,33%
10	295	98,33%	0	0,00%	5	1,67%
11	0	0,00%	5	1,67%	2	0,67%
12	0	0,00%	0	0,00%	2	0,67%
13	300	100,00%	0	0,00%	0	0,00%
47 (8 e 12)	595	99,17%	0	0,00%	0	0,00%
81 (7 e 11)	416	69,33%	0	0,00%	0	0,00%
78 (7 e 13)	8	1,33%	0	0,00%	0	0,00%
82 (7, 11 e 13)	0	0,00%	7	1,17%	0	0,00%
Total:	3.860	98,97%	12	0,31%	28	0,72%

Como pode ser visto, este circuito também possui classes com dados similares, embora o fenômeno não tenha a mesma gravidade do caso do filtro CTSV. Como consequência, o desempenho dos CCUs para os casos sem sobreposição de classes fica de 20 a 30% inferior ao desempenho considerando a sobreposição de classes. No caso do multiclasse, observa-se o mesmo comportamento dos circuitos anteriores, onde nota-se um desempenho superior do multiclasse para os casos sem sobreposição e para os casos com sobreposição, dos CCUs.

### 6.3.2. Sobreposição de classes

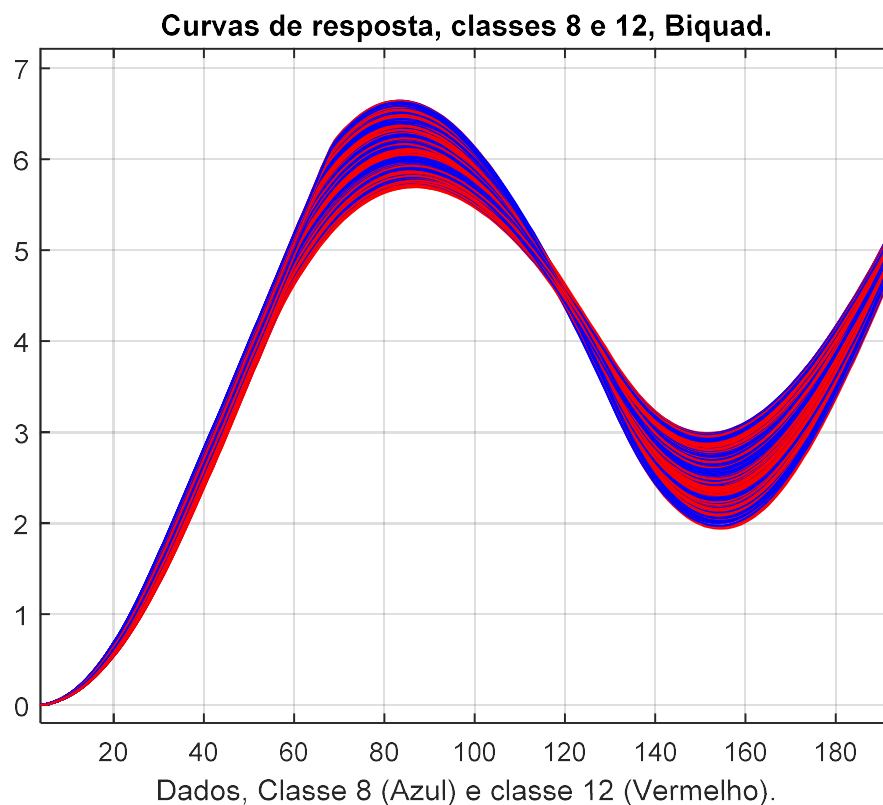
Da tabela 20, pode ser percebido que neste circuito também há problemas com a sobreposição de classes. As classes no qual ocorrem o problema são os pares 8 e 12 e 7 com 11. Nas figuras 31 e 32 pode-se ver o grau de severidade entre as sobreposições destas classes entre si.



*Figura 31: Sobreposição de classes, classes 7 e 11, Biquad*

As classes 7 e 11 apresentam resposta similar ao longo de todos os atributos. Entretanto, é possível observar somente uma extrema proximidade entre este par de classes, existindo mínimas sobreposições entre elas.





*Figura 32: Sobreposição de classes, classes 8 e 12, Biquad*

Já nas classes 8 e 12, é possível observar que alguns dados de uma classe estão de fato inseridos na região de dados da outra, havendo um problema maior de sobreposição, sendo este o principal responsável pela perda de 20% de desempenho da classificação sem sobreposição para este circuito.

### **6.3.3. Influência da representação de dados na sobreposição**

Assim como nos circuitos anteriores, aqui também foi observado o comportamento de cada classificador para cada tipo diferente de representação de dados realizada. A figura 33 mostra o desempenho do classificador gaussiano no circuito Biquad para cada método de representação de dados utilizado.

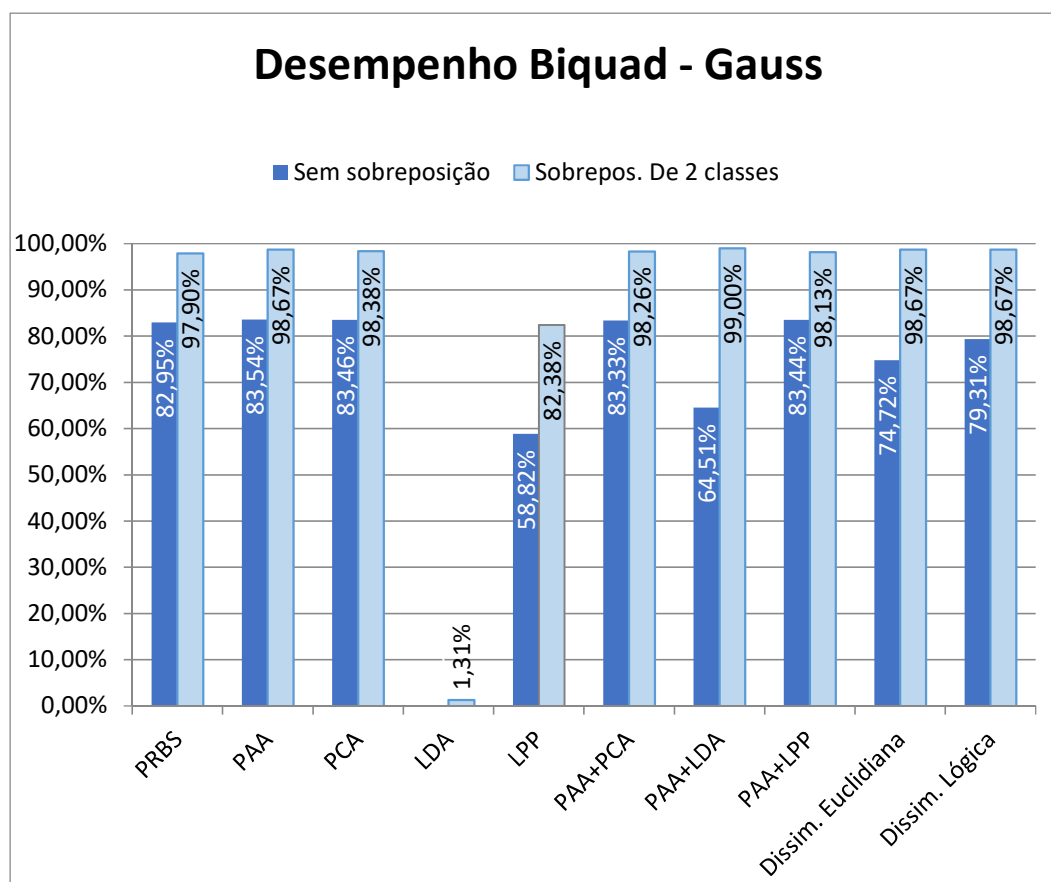


Figura 33: Desempenho por representação de dados, classificador Gauss, Biquad

Para o método gaussiano, o Biquad apresentou um bom desempenho para a maior parte dos métodos de representação apresentados, até mesmo no caso do sinal bruto PRBS, o que mostra a capacidade da excitação em PRBS neste circuito. Nos métodos de extração de assinaturas, o método PAA apresentou um ótimo desempenho, sendo superior à classificação no sinal bruto nos dois casos de classificação abordados. Nos métodos de redução de dimensão, o método do PCA manteve o desempenho equivalente ao PAA e ao sinal bruto. No caso do LPP, houve uma redução no desempenho obtido no classificador e para o LDA, o classificador teve um desempenho extremamente insatisfatório, não conseguindo detectar a maior parte dos dados. Este fato ocorrido no LDA não foi exclusividade do classificador gaussiano, o que será mostrado na análise dos classificadores seguintes. Apesar de somente o PCA ter mostrado um desempenho equiparável com os outros métodos na redução de dimensão, a combinação entre os métodos de PAA e redução de dimensão surtiu um efeito positivo para o caso deste

classificador. No caso do PAA+PCA houve a equivalência de desempenho com os casos anteriores, no caso do PAA+LDA houve substancial melhora no método de redução após a representação PAA ser aplicada no sinal. O desempenho sem sobreposição subiu para 64,51% e com sobreposição o desempenho se equiparou com os métodos anteriores, chegando a 99%, a qual foi o melhor desempenho obtido neste classificador, para este circuito. O LPP, antes com bom desempenho, mas inferior aos desempenhos dos métodos que obtiveram os melhores resultados, consegue alcançar o desempenho dos outros métodos, após o tratamento do sinal bruto com o PAA ser aplicado anteriormente à esta redução. As dissimilaridades tiveram desempenho equivalente aos métodos anteriores, com um desempenho levemente inferior, somente nos casos sem sobreposição.

No caso do *knn* aplicado ao circuito Biquad, o desempenho obtido por todos os métodos de representação foi inferior, nos casos sem sobreposição, quando comparados com o *Gauss*. Na extração de assinaturas, o PAA mostrou novamente um desempenho satisfatório, sendo superior ao sinal bruto no caso sem sobreposição e equivalente no caso com sobreposição.

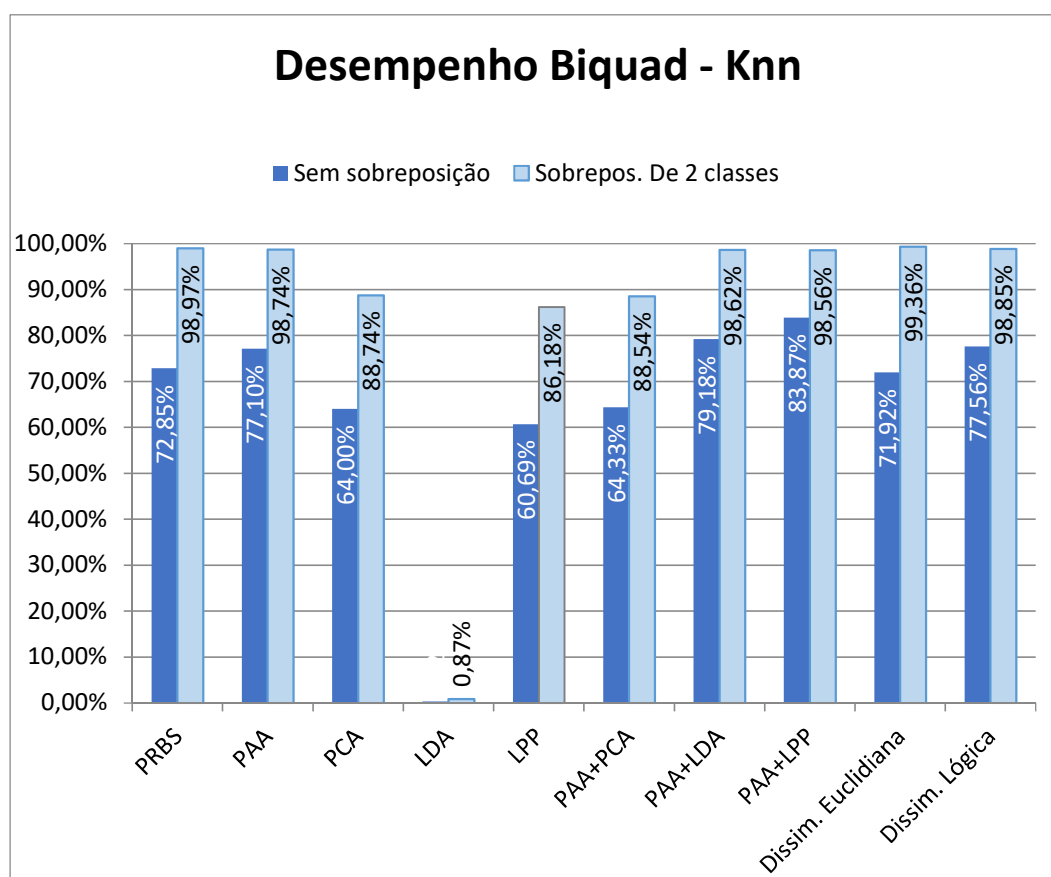


Figura 34: Desempenho por representação de dados, classificador knn, Biquad

Nos métodos de redução de dimensão, houve um desempenho inferior ao *Gauss*, houve uma pequena melhora na representação do LPP, mas uma perda de 10% na representação do PCA. O LDA apresentou desempenho insatisfatório para este classificador, assim como no *Gauss* para este circuito. A combinação dos métodos de PAA e DR mostrou uma melhora no desempenho do classificador novamente, onde que o desempenho do PAA combinado com o LDA passa a ser superior ao desempenho do PAA individualmente e torna a utilização do LDA satisfatória. A combinação PAA+LDA também teve desempenho superior à classificação com o sinal bruto, considerando o caso sem sobreposição, e equivalente para o caso com sobreposição. Na combinação PAA+LPP, também tivemos um desempenho satisfatório, tendo esta representação obtido o melhor resultado no caso sem sobreposição e um desempenho equivalente ao sinal bruto, com sobreposição. Nos métodos de dissimilaridade, obtivemos um desempenho cerca de 2% inferior a classificação pelo *Gauss*, para o sinal bruto. A dissimilaridade euclidiana apresentou um resultado 0,93% inferior a classificação por sinal bruto no caso sem sobreposição e na dissimilaridade lógica o desempenho foi superior ao sinal bruto em 4,71%. No caso da classificação com sobreposição, ambos os casos de dissimilaridade apresentaram desempenho satisfatórios, tendo a dissimilaridade euclidiana obtido o melhor desempenho geral neste classificador, com 99,36% de acurácia e uma superioridade de 0,39% em relação ao sinal bruto. No caso da dissimilaridade lógica, o desempenho foi praticamente igual ao desempenho com o sinal bruto.

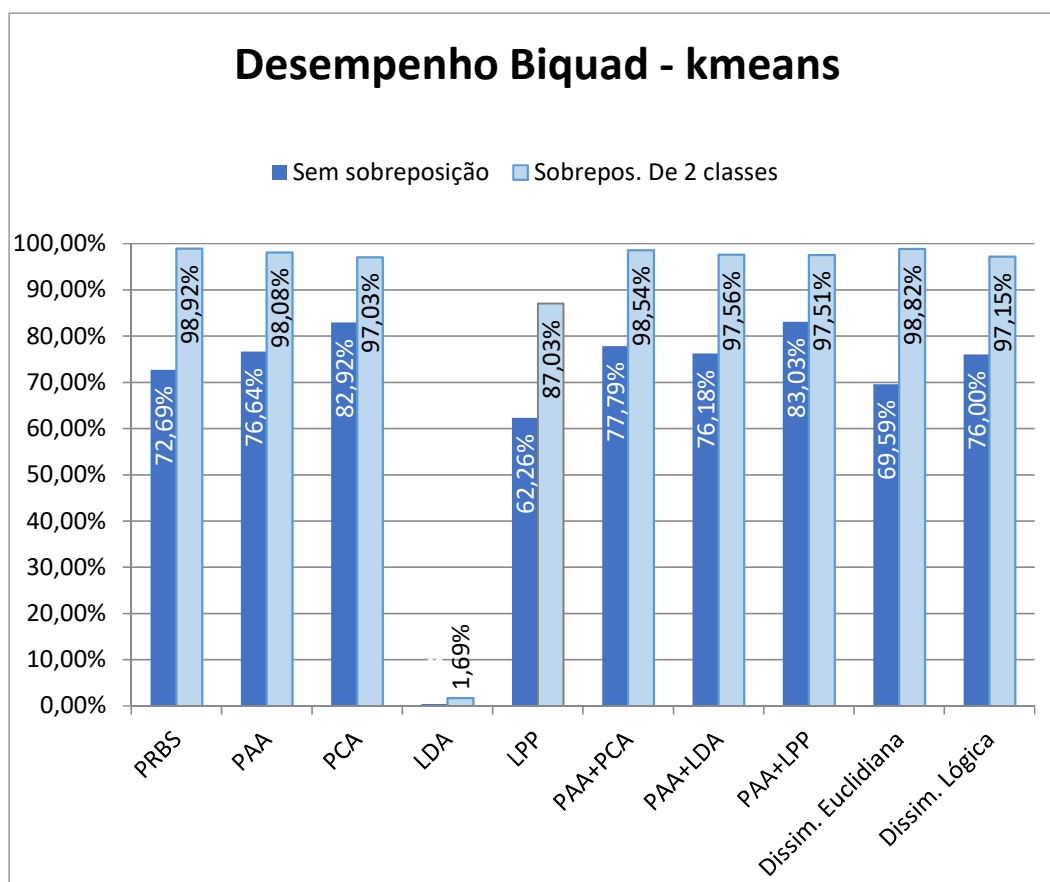


Figura 35: Desempenho por representação de dados, classificador kmeans, Biquad

Para o classificador de reconstrução *kmeans* (figura 35), o sinal bruto apresentou um desempenho equivalente ao do *knn*, para os casos sem sobreposição, e uma desempenho satisfatória para o caso com sobreposição. No caso deste classificador, à exceção das representações em LDA e LPP, todas as outras representações obtiveram um desempenho equivalente à classificação com o sinal bruto, para os casos com sobreposição, e superior ou equivalente para os casos sem sobreposição. No PAA, obteve-se uma classificação sem sobreposição superior em 3,95% ao sinal bruto PRBS e uma diferença de 0,84% da classificação PAA para o sinal bruto com sobreposição.

Em relação aos métodos de redução de dimensão, tem-se que o PCA apresenta um bom desempenho para este tipo de classificador, atingindo uma melhora de classificação de 10,23% para os casos sem sobreposição, ou seja, esta representação reduziu o número de sobreposição de classes de até 399 circuitos neste classificador. Por outro lado, para o caso com sobreposição o

PCA teve um desempenho 1,89% abaixo do obtido com o sinal bruto. A representação pelo método LPP apresentou um resultado de classificação cerca de 10% inferior a classificação com sinal bruto, em ambos os casos de classificação. No caso da redução pelo LDA, a classificação foi totalmente insatisfatória para ambos os casos de classificação. No caso dos métodos de representação combinados, todos os casos avaliados obtiveram um desempenho satisfatório, em ambos os casos de classificação, no PAA+LDA obtivemos um desempenho 5,1% superior ao sinal bruto na classificação sem sobreposição e 0,38% inferior ao sinal bruto na classificação com sobreposição, no caso do PAA+LDA, a diferença é de 3,49% de melhora no caso sem sobreposição e desempenho menor em 1,36% do sinal bruto para o caso com sobreposição. A combinação PAA+LPP teve a melhor desempenho no caso sem sobreposição, com 83,03% de desempenho contra os 72,69% obtidos na classificação com o sinal bruto, esta é uma melhora de 10,34%, que representa menos 403 casos de circuitos sobrepostos, agora corretamente classificados. Na classificação com sobreposição, o desempenho foi 1,41% inferior à classificação feita a partir do sinal bruto.

Nos casos das representações por dissimilaridade, ambos os casos mostraram desempenho satisfatório. No caso da dissimilaridade euclidiana, temos um desempenho inferior a classificação com o sinal bruto em 3,1% e uma classificação praticamente equivalente para a classificação com o sinal bruto, com uma diferença de apenas 0,1% inferior no caso da dissimilaridade euclidiana, considerando as sobreposições. Na dissimilaridade lógica, verificou-se uma melhora no desempenho do classificador para o caso sem sobreposição, apresentando uma melhora de 3,31% para este método de representação. No caso da classificação considerando as sobreposições, obteve-se uma diferença de 1,77% a menos na acurácia da dissimilaridade lógica com relação a classificação com o sinal bruto.

Para o classificador de *Parzen* (fig. 36), foi obtido um desempenho insatisfatório para vários métodos de representação de dados, começar pelo próprio sinal bruto. Este classificador não foi capaz de distinguir quaisquer classes para a classificação no sinal PRBS. Ao utilizar a extração de assinaturas PAA, o classificador consegue então classificar, obtendo um desempenho de 75% para os casos sem sobreposição e 98,64% para os

casos com sobreposição. Nos métodos de redução de dimensão, tanto o LDA quanto o LPP falham com este classificador neste circuito. O PCA apresenta desempenho praticamente equivalente ao do PAA, sendo superior em 0,85% no caso sem sobreposição e inferior em 1,08% no caso com sobreposição. Na representação pela combinação de métodos, a classificação pelo *parzen* consegue ser realizada com todos os métodos de redução de dimensão, obtendo o desempenho menos satisfatório para a combinação PAA+PCA, que obteve um desempenho baixo para os casos sem sobreposição de 56,56%, e um desempenho razoável de 87,79% para a classificação com sobreposição. Para o caso dos métodos de redução LDA e LPP, que sozinhos não obtiveram resultados, a combinação dos mesmos com o PAA funcionou positivamente mais uma vez. No caso do PAA+LDA, o classificador obteve um desempenho de 77,79% para os casos sem sobreposição e 98,54% para o caso com sobreposição, o que para o primeiro caso melhora em 2,79% o desempenho do PAA individualmente. No caso da combinação PAA+LPP, o desempenho obtido no caso sem sobreposição foi de 77,33%, com uma melhora de 2,33% para o PAA individualmente, e no caso com sobreposição a desempenho foi de 98,74%, sendo praticamente equivalente ao PAA individualmente.

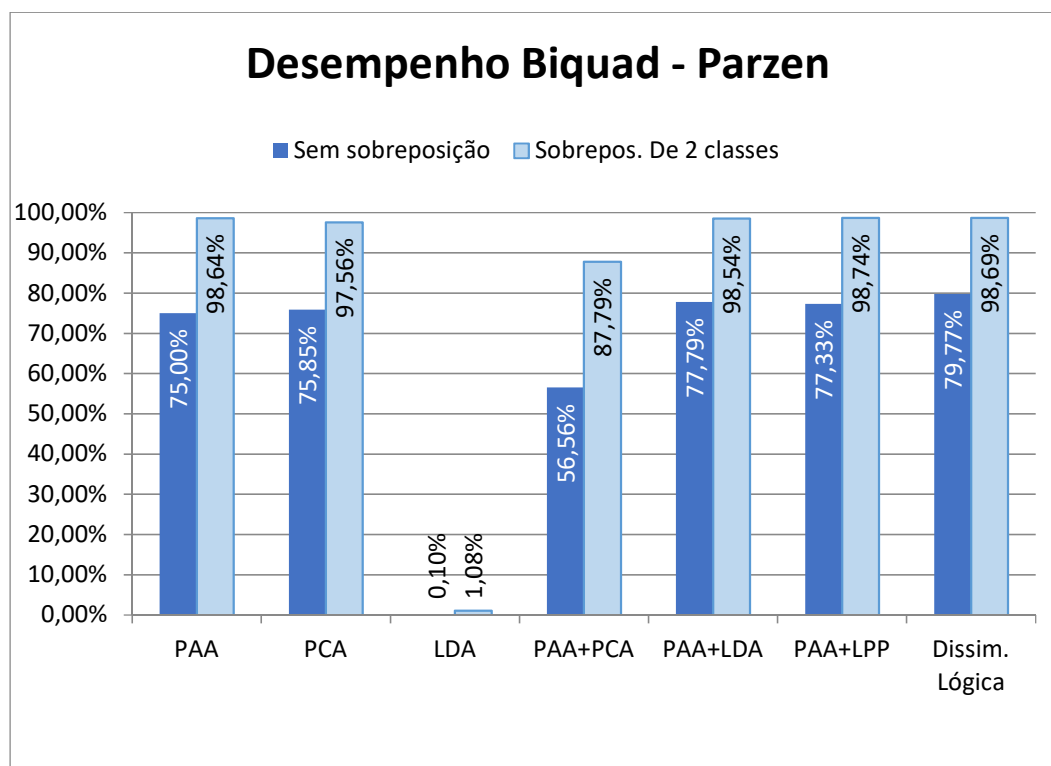


Figura 36: Desempenho por representação de dados, classificador Parzen, Biquad

Para os métodos de representação por dissimilaridade, o método de dissimilaridade euclidiana não pôde ser classificado no *Parzen*. Por outro lado, dissimilaridade lógica apresentou um desempenho de 79,77% na classificação sem sobreposição e de 98,69% na classificação com sobreposição.

#### 6.3.4. Desempenho para a representação de dados

A seguir foi realizado o tratamento dos dados do circuito pelos métodos de representação de dados apresentados. A figura 37 mostra o desempenho de cada classificador para o método que obteve o melhor resultado com o mesmo. No caso do filtro Biquad, os melhores desempenhos apresentados foram para os métodos de representação com PAA ou de dissimilaridade.

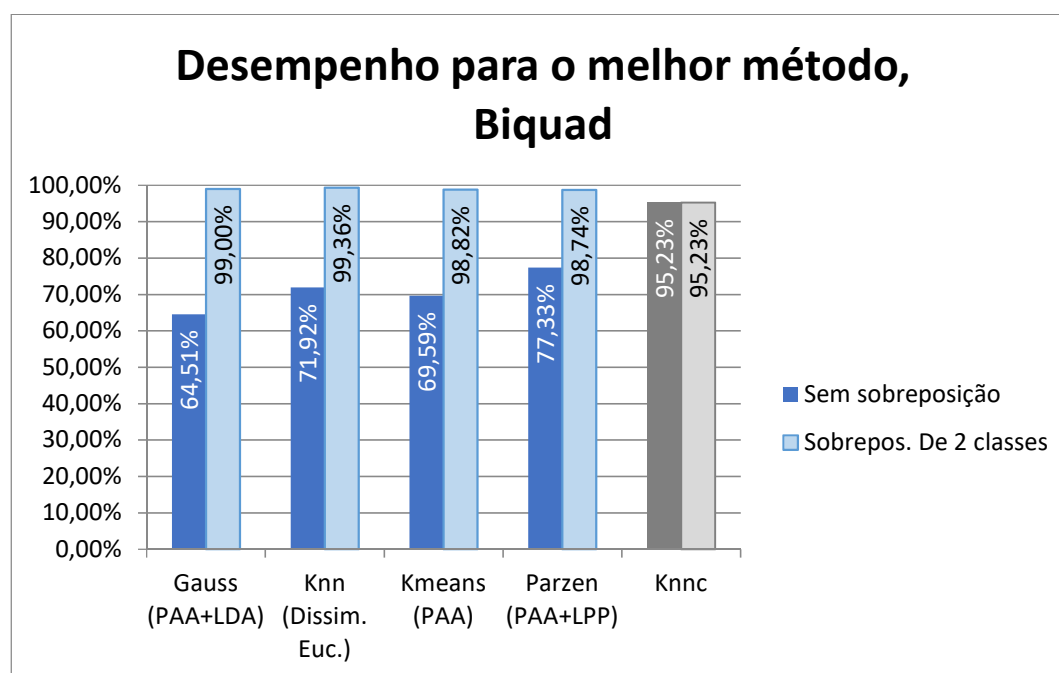


Figura 37: Curva de desempenho para os melhores métodos, circuito Biquad

Neste caso, o classificador que obteve o melhor desempenho foi o *knn*. Adiante, na tabela 21 encontram-se os resultados em detalhes deste classificador, com seu melhor método de representação, para o Biquad.



*Tabela 21: Distribuição dos dados para o melhor classificador do Biquad no melhor método de representação de dados*

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de desconhecidos
1	297	99,00%	0	0,00%	3	1,00%
2	300	100,00%	0	0,00%	0	0,00%
3	298	99,33%	0	0,00%	2	2,67%
4	300	100,00%	0	0,00%	0	0,00%
5	299	99,67%	0	0,00%	1	0,33%
6	299	99,67%	0	0,00%	1	0,33%
7	80	26,67%	1	0,33%	0	0,00%
8	0	0,00%	1	0,33%	0	0,00%
9	291	97,00%	0	0,00%	9	3,00%
10	296	98,67%	0	0,00%	4	1,33%
11	41	13,67%	0	0,00%	3	1,00%
12	4	1,33%	0	0,00%	0	0,00%
13	300	100,00%	0	0,00%	0	0,00%
47 (8 e 12)	595	99,17%	0	0,00%	0	0,00%
81 (7 e 11)	475	79,17%	0	0,00%	0	0,00%
Total:	3.875	99,36%	2	0,05%	23	0,59%

Aqui, novamente os métodos de dissimilaridade e combinados PAA+DR assumem os melhores resultados. Da tabela 21 pode ser visto que, em muitas vezes, o classificador chega a quebrar o limite de fração de rejeição, o que mostra a capacidade dos CCUs potencializada com o uso dos métodos de representação de dados, principalmente nas representações por dissimilaridade. No caso, a combinação trouxe uma potencialização até mesmo para o caso do multiclasse, que diminuiu a diferença de desempenho para os classificadores classe única com o uso da representação em dissimilaridade lógica (anexo I-C).

### **6.3.5. Desempenho para classes não previstas**

Aqui foi avaliado o comportamento dos classificadores frente a classes imprevistas no circuito Biquad. Nesta parte foi estudado como cada tipo de classificador se comportaria frente a um problema imprevisto deste circuito. Cada classificador será submetido a um conjunto de dados com 7 classes de

falhas não previstas no treinamento, sendo 4 delas, falhas múltiplas do circuito Biquad, e 3 falhas totalmente imprevistas, conforme observado na tabela 22.

*Tabela 22: Componentes com falha e suas respectivas classes, Biquad*

Componente	Falha Associada	Respectiva Classe
R1	R1+	1
	R1-	2
R2	R2+	3
	R2-	4
R3	R3+	5
	R3-	6
R4	R4+	7
	R4-	8
C1	C1+	9
	C1-	10
C2	C2+	11
	C2-	12
NoFail		13
M1	(R4+, C2+)	14
	(R4-, C2-)	15
M2	(R4+, C2-)	16
	(R4-, C2+)	17
X1		18
X2		19
X3		20

Os resultados obtidos para o classificador multiclasse são mostrados na tabela 23. O classificador multiclasse previsivelmente falha ao classificar as falhas não previstas.

Tabela 23: Resultado de classificação multiclasse para falhas não previstas, Biquad

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de Erro
1	300	100,00%	0	0,00%
2	300	100,00%	0	0,00%
3	300	100,00%	0	0,00%
4	300	100,00%	0	0,00%
5	300	100,00%	0	0,00%
6	300	100,00%	0	0,00%
7	300	100,00%	0	0,00%
8	205	68,33%	95	31,67%
9	300	100,00%	0	0,00%
10	300	100,00%	0	0,00%
11	299	99,67%	1	0,33%
12	210	70,00%	90	30,00%
13	300	100,00%	0	0,00%
14	0	0,00%	300	100,00%
15	0	0,00%	300	100,00%
16	0	0,00%	300	100,00%
17	0	0,00%	300	100,00%
18	0	0,00%	300	100,00%
19	0	0,00%	300	100,00%
20	0	0,00%	300	100,00%
Total:	3.714	61,90%	2.286	38,10%

Na figura 38 são exibidos os gráficos de desempenho de cada classificador, para a avaliação com falhas imprevistas no circuito Biquad. Novamente aqui, quase todas as falhas não previstas são classificadas como falhas desconhecidas para os CCUs, o que de fato o faz acertar na classificação.

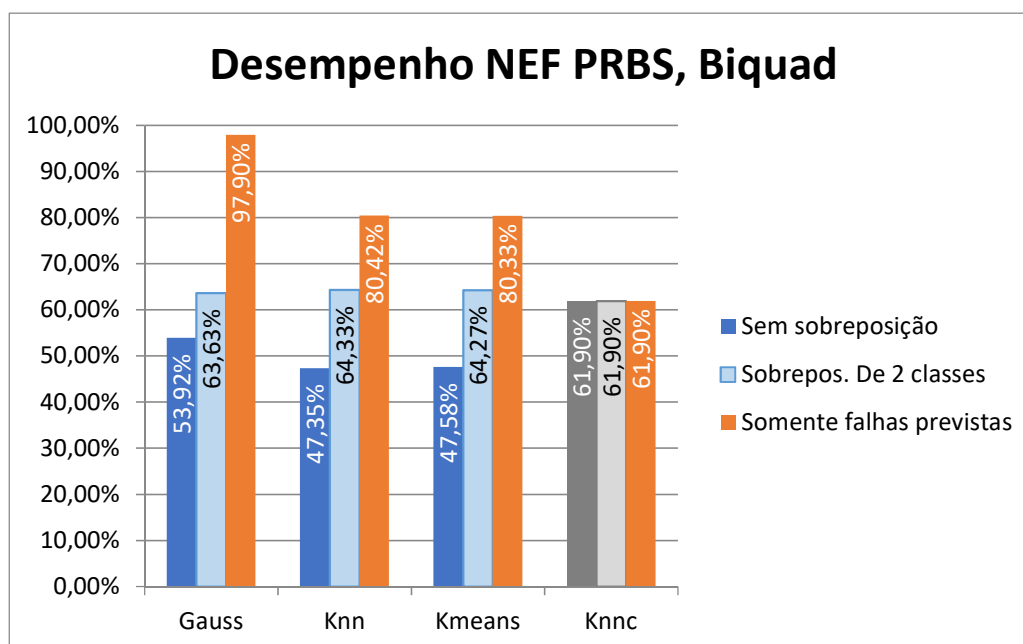


Figura 38: Desempenho dos classificadores para falhas não previstas com sinal PRBS, Biquad (Parzen não classificou)

Para este caso, o classificador que obteve o melhor desempenho foi o gaussiano. A tabela 24 mostra os resultados detalhados deste classificador para o tratamento das falhas não previstas.

Tabela 24: Distribuição dos dados para o melhor classificador do Biquad para a curva PRBS

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de Desconhecidos
1	297	99,00%	0	0,00%	3	1,00%
2	291	97,00%	0	0,00%	9	3,00%
3	291	97,00%	0	0,00%	9	3,00%
4	299	99,67%	0	0,00%	1	0,33%
5	299	99,67%	0	0,00%	1	0,33%
6	296	98,67%	0	0,00%	4	1,33%
7	291	97,00%	0	0,00%	9	3,00%
8	1	0,33%	0	0,00%	6	2,00%
9	296	98,67%	0	0,00%	4	1,33%
10	288	96,00%	0	0,00%	12	4,00%
11	287	95,67%	0	0,00%	13	4,33%
12	4	1,33%	0	0,00%	6	2,00%
13	295	98,33%	0	0,00%	5	1,67%
47 (8 e 12)	583	97,17%	0	0,00%	0	0,00%

81 (7 e 11)	0	0,00%	0	0,00%	0	0,00%
14	0	0,00%	0	0,00%	300	100,00%
15	0	0,00%	0	0,00%	300	100,00%
16	0	0,00%	0	0,00%	300	100,00%
17	0	0,00%	8	2,67%	292	97,33%
18	0	0,00%	0	0,00%	300	0,00%
19	0	0,00%	0	0,00%	300	0,00%
20	0	0,00%	0	0,00%	300	0,00%
Total:	3.818	63,63%	8	0,13%	2.174	36,23%
Total não rejeitado	3.818	97,90%	8	0,21%		

Comparando o desempenho para avaliar as falhas não previstas dos CCUs e do multiclasse, os CCUs novamente conseguem obter o melhor desempenho.

### **6.3.6. Comparação com o multiclasse**

No caso da classificação multiclasse, todos os casos, com exceção da representação pelo LDA foram satisfatórios, visto que este tipo de classificador pode não lidar com sobreposições de classes, acertando ou errando na sua escolha para o dado aleatoriamente. Para o sinal PRBS, a classificação do circuito Biquad ficou em 95,23%, inferior a todos os casos de classe única em PRBS, com exceção do caso do *Parzen*, onde não foi possível efetuar a classificação.

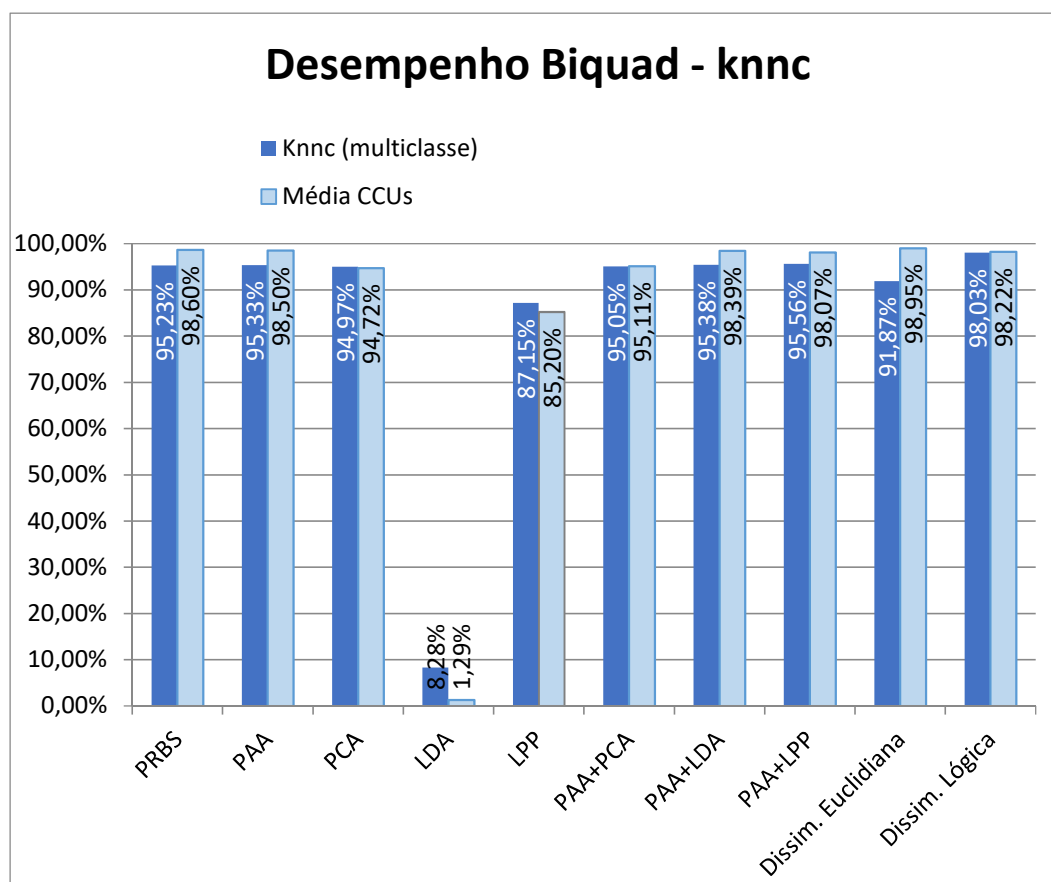


Figura 39: Desempenho multiclasse x CCUs, Biquad

. É interessante notar que, o classificador multiclasse apresenta desempenho superior aos CCUs somente em 3 dos 10 casos de classificação para diferentes métodos de representação neste circuito e, sendo a diferença de desempenho para o comitê de classe única inferior em 3,36% no sinal bruto PRBS. Apesar da superioridade dos CCUs em 70% dos casos comparados, essa diferença não é expressiva como ocorre no circuito CTSV. Isto se deve ao fato que as sobreposições entre classes neste circuito são menos severas que no CTSV, o que conseqüentemente facilita a classificação correta pelo multiclasse.

## 6.4. Retificador Não-Linear

### 6.4.1. Desempenho

Dos 4 circuitos analisados neste trabalho, o retificador não-linear é o único que não é um circuito de filtro e, provavelmente, o circuito mais sensível às variações dos seus parâmetros. A sua arquitetura pode ser vista na figura 40. Não possui componentes ativos e é composta por um único diodo retificador, três capacitores, um indutor e uma resistência\*\*. (XIAO, Y., FENG, L., 2012) utilizam este circuito como *benchmark* em seu trabalho com redução de dimensão para a detecção de falhas em circuitos, apesar de sua utilização em benchmark não ser muito usual neste tipo de problema, em virtude da sua maior sensibilidade.

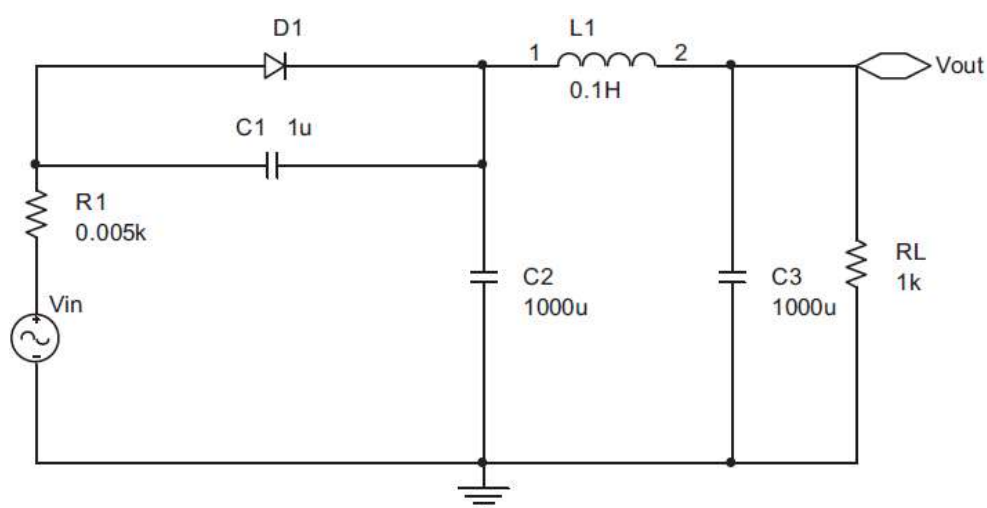


Figura 40: Circuito Retificador Não-Linear \*

\* a consideração da resistência  $R1$  deste circuito pode ser opcional, visto que ela modela a resistência interna da fonte do sinal de entrada e não, necessariamente, um componente do sistema de retificação.

\*\* Como  $RL$  é uma carga, não é propriamente parte integrante do circuito e pode apresentar qualquer característica, por este motivo, não foi considerada como parte do circuito. No entanto, o valor de  $RL$  e as suas características influenciam no tipo de resposta apresentado pelo retificador.

No caso deste circuito, todos os seus componentes principais foram considerados para a definição das classes de falhas, totalizando 11 classes, conforme observado na tabela 25

*Tabela 25: Componentes com falha e suas respectivas classes, NLR*

Componente	Falha Associada	Respectiva Classe
R1	R1+	1
	R1-	2
L1	L1+	3
	L1-	4
C1	C1+	5
	C1-	6
C2	C2+	7
	C2-	8
C3	C3+	9
	C3-	10
NoFail		11

Um classificador multiclasse *knn* foi utilizado para fins de comparação e análise dos resultados fornecidos do classificador de classe única assim como feito nos casos anteriores. A tabela 26 mostra os resultados obtidos pelo classificador multiclasse, para a detecção das falhas no circuito retificador não-linear:



Tabela 26: Resultado de classificação multiclasse, NLR

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de Erro
1	300	100,00%	0	0,00%
2	300	100,00%	0	0,00%
3	300	100,00%	0	0,00%
4	300	100,00%	0	0,00%
5	300	100,00%	0	0,00%
6	300	100,00%	0	0,00%
7	300	100,00%	0	0,00%
8	300	100,00%	0	0,00%
9	300	100,00%	0	0,00%
10	300	100,00%	0	0,00%
11	300	100,00%	0	0,00%
Total:	3.300	100,00%	0	0,00%

Na figura 41 adiante temos os resultados dos comitês de classe única na classificação do circuito retificador não-linear. Neste caso, o classificador que apresentou a melhor desempenho foi o *Gauss*.

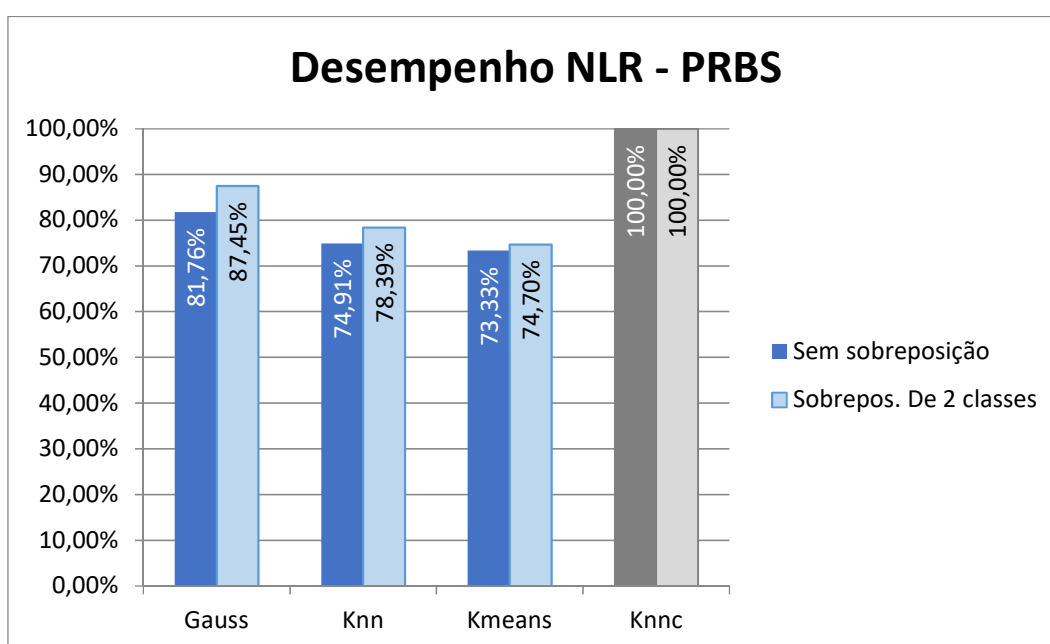


Figura 41: Desempenho para o Retificador não-linear no PRBS (o Parzen não classificou)

A tabela 27 mostra os resultados do classificador com o melhor desempenho, em relação à distribuição dos dados em cada classe. Neste caso, o classificador que teve o melhor desempenho foi o *Gauss*. Comparado com o desempenho do multiclasse, observa-se que os CCUs apresentaram desempenho inferior, mesmo para a classificação com sobreposição podendo ser um indício de uma sobreposição severa entre classes. Os dados da tabela 27 a seguir comprovam esta suspeita.

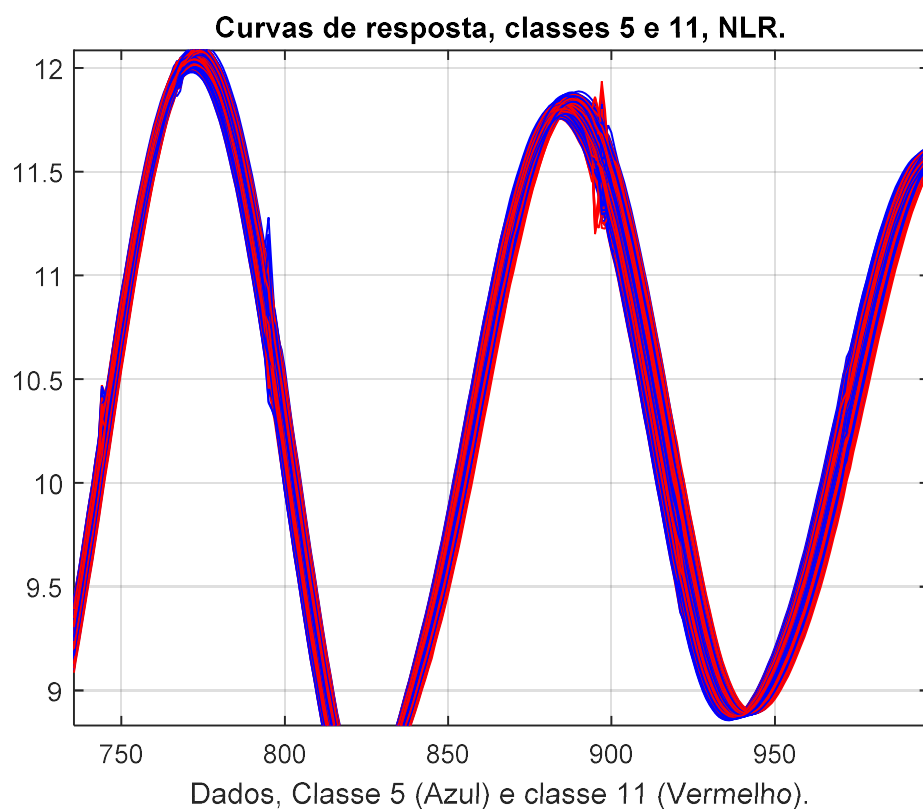
*Tabela 27: Distribuição dos dados para o melhor classificador do NLR para a curva PRBS*

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de rejeição
1	297	99,00%	0	0,00%	3	1,00%
2	297	99,00%	0	0,00%	3	1,00%
3	297	99,00%	0	0,00%	3	1,00%
4	297	99,00%	0	0,00%	3	1,00%
5	297	99,00%	0	0,00%	3	1,00%
6	14	4,67%	0	0,00%	3	1,00%
7	297	99,00%	0	0,00%	3	1,00%
8	297	99,00%	0	0,00%	3	1,00%
9	297	99,00%	0	0,00%	3	1,00%
10	297	99,00%	0	0,00%	3	1,00%
11	11	3,67%	0	0,00%	3	1,00%
76 (5 e 11)	31	5,17%	0	0,00%	0	0,00%
44 (6 e 11)	157	26,17%	0	0,00%	0	0,00%
108 (5, 6 e 11)	0	0,00%	381	42,33%	0	0,00%
Total:	2.886	87,45%	381	11,55%	33	1,00%

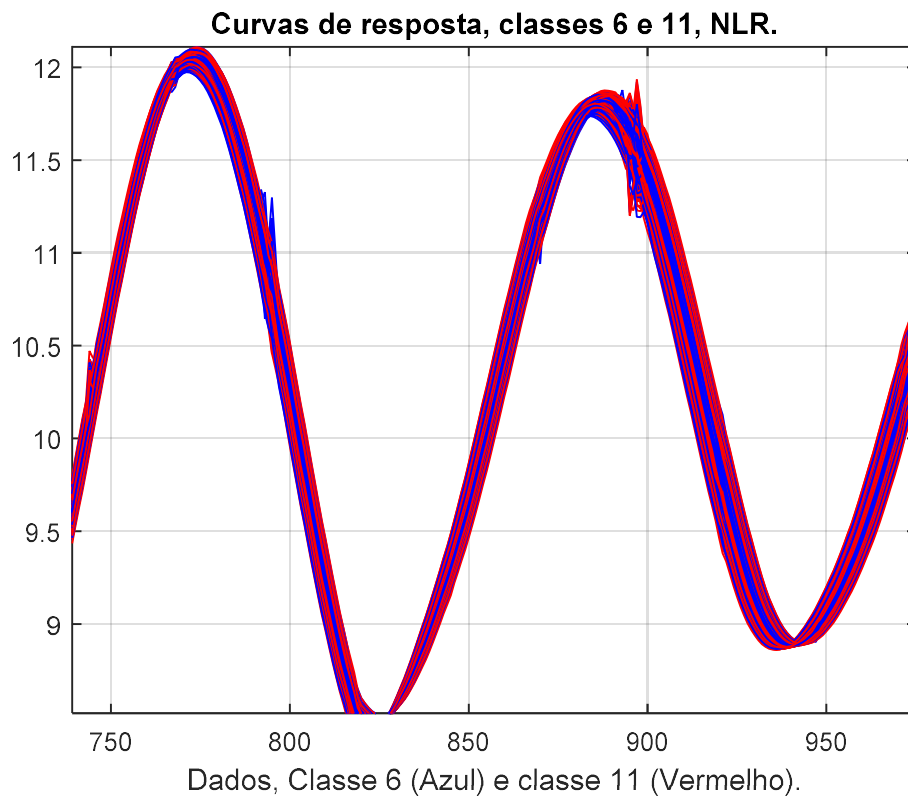
Podemos ver claramente na tabela 27 que o problema se encontra na sobreposição entre as classes 5, 6 e 11 e sendo o desempenho do classificador perfeito fora destas classes. Dessa forma, na sequência analisa-se a severidade desta sobreposição.

#### 6.4.2. Sobreposição de classes

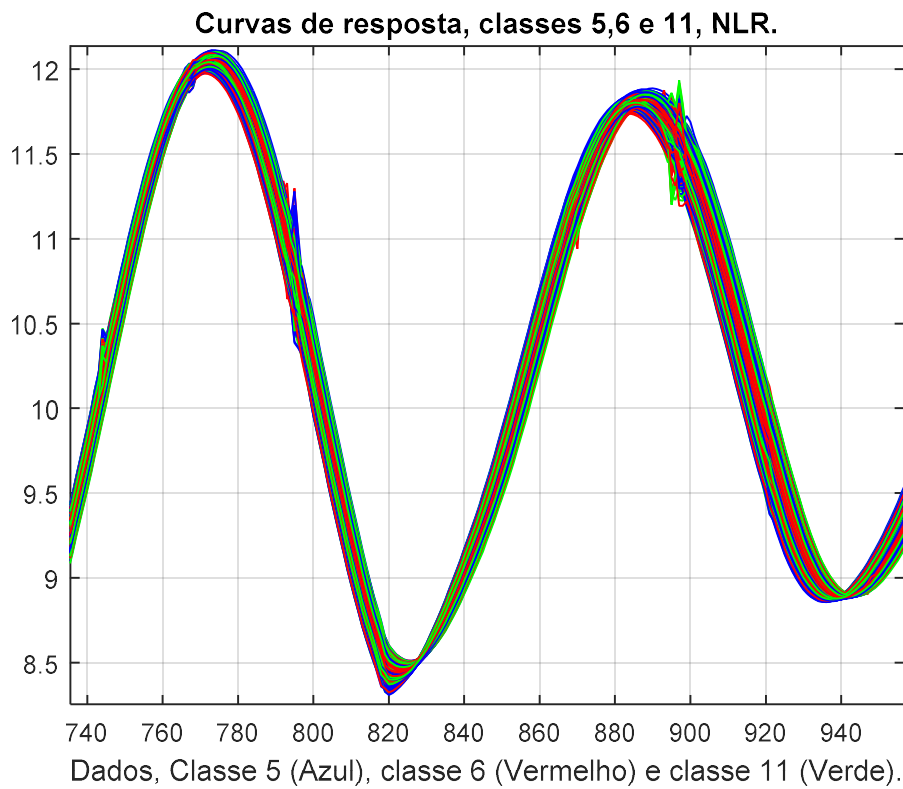
Este circuito sofre com um grande problema de sobreposição de classes. Como observado nos resultados anteriores, as classes 5, 6 e 11 parecem apresentar os mesmos dados de respostas e isso é confirmado pela análise das figuras 42 a 44, que apresentam as curvas de respostas entre as classes 5, 6 e 11. Os resultados apresentados nos gráficos indicam que os dados, de fato, não podem ser facilmente separados, caso isto seja possível.



*Figura 42: Sobreposição de classes, classes 5 e 11, NLR*



*Figura 43: Sobreposição de classes, classes 6 e 11, NLR*



*Figura 44: Sobreposição de classes. Classes 5, 6 e 11, circuito NLR*

É interessante observar que esta sobreposição ocorre em classes muito específicas: as classes 5 e 6 são as classes de falhas paramétricas superior e inferior do capacitor C1 do circuito retificador respectivamente, enquanto que a classe 11 é a classe de funcionamento normal do circuito. Ou seja, a confusão entre essas três classes e o nível com que elas estão sobrepostas indicam claramente que, o capacitor C1 tem um peso muito pequeno na variação da resposta do circuito, ao ponto de que, uma grande variação paramétrica neste componente não faria a menor diferença na resposta de saída do circuito, que continuaria a apresentar comportamento normal.

#### ***6.4.3. Influência da representação de dados na sobreposição***

Assim como nos três circuitos anteriores, o comportamento de cada classificador foi analisado para cada tipo diferente de método de representação de dados realizado. Existe uma importância nesta análise deste circuito assim como no caso do CTSV, onde deseja-se saber se somente o tratamento dos dados com os métodos de representação são capazes de melhorar a sobreposição ocorrida nas respostas deste circuito. O gráfico da figura 45 mostra o desempenho do classificador gaussiano no retificador não-linear com cada método utilizado para representar os dados.

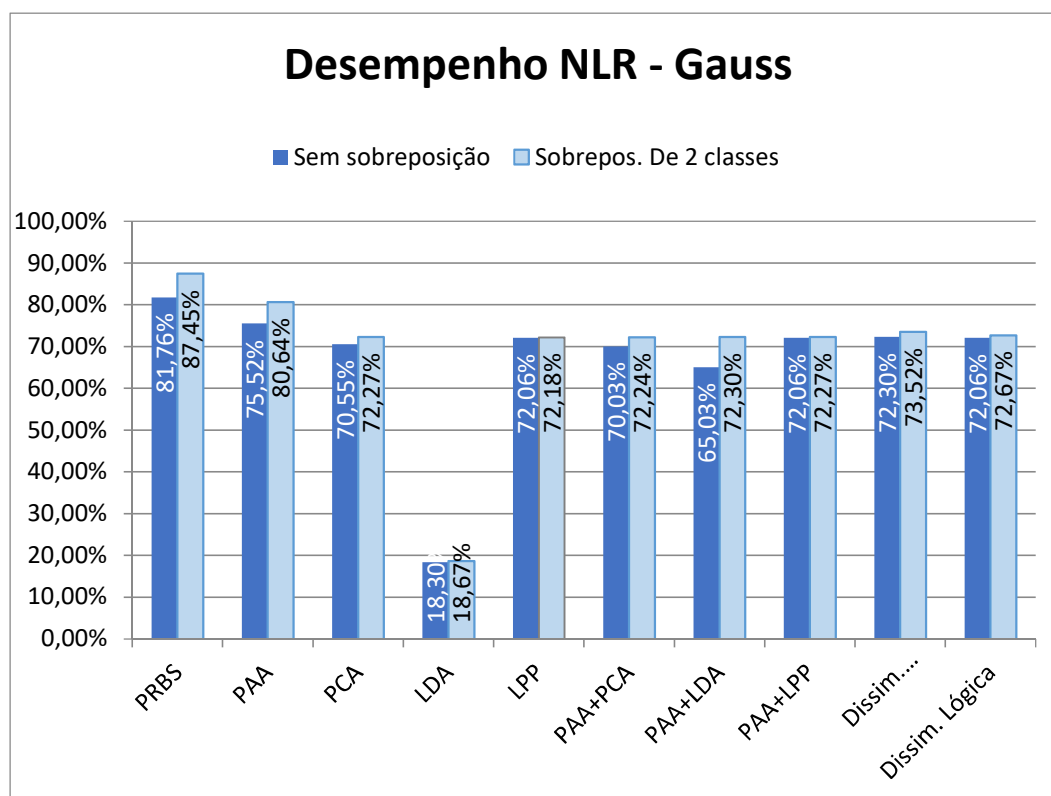


Figura 45: Desempenho por representação de dados, classificador Gauss, NLR

Com exceção dos métodos LDA e para o sinal bruto, todos os métodos de representação obtiveram resultados semelhantes de classificação para o classificador gaussiano no NLR, tanto no caso sem sobreposição quanto no caso com sobreposição. No sinal bruto PRBS, houve a melhor classificação geral nos CCUs em ambos os casos de sobreposição, com 81,76% de acerto para a classificação sem sobreposição e, quando considerada a sobreposição, o desempenho obtido no sinal bruto foi de 87,45% para este classificador.

A representação pelo PAA foi o método de representação que apresentou o melhor desempenho no NLR. Porém, não teve desempenho equivalente ao sinal bruto, sendo 6,24% inferior ao sinal bruto sem sobreposição e 6,81% inferior ao sinal bruto com sobreposição.

Para o caso dos métodos de redução de dimensão, o desempenho entre o PCA e LPP foi praticamente equivalente no caso com sobreposição. Por sua vez, no caso sem sobreposição houve uma pequena superioridade do LPP, em 1,51%. O desempenho de ambos foi inferior ao do sinal bruto para

os dois casos de sobreposição. O LDA não apresentou desempenho satisfatório.

No caso da combinação entre os métodos PAA+DR, os resultados foram equivalentes para os casos com sobreposição do circuito NLR e variados nos casos sem sobreposição. Para o PAA+PCA, a diferença para o sinal bruto foi de 11,73% sem sobreposição, no PAA+LDA, essa diferença foi a maior da combinação PAA+DR, com 16,73% de diferença, e para o PAA+LPP essa diferença foi de 9,7%. O PAA+LPP foi o método que apresentou a menor diferença entre o caso sem sobreposição e com sobreposição, com uma diferença de apenas 0,21%. Em nenhum dos casos, o desempenho foi superior ao obtido com o sinal bruto.

Para as representações por dissimilaridade, os resultados foram equivalentes aos casos anteriores. Na euclidiana, a diferença para o sinal bruto foi de 9,46%, para o caso sem sobreposição, e de 13,93%, para o caso com sobreposição. A dissimilaridade lógica apresentou uma diferença de 9,6% e 14,78%, com relação aos respectivos casos do sinal bruto, e uma diferença entre casos de sobreposição de 0,61%.

No caso do *knn*, com exceção do LDA, todos os métodos apresentaram desempenho similar (entre 70 e 80%), prejudicado pela sobreposição de dados entre as classes 5,6 e 11. Para o sinal bruto, o desempenho foi de 74,91% no caso sem sobreposição e 78,39% para o caso com sobreposição, sendo novamente os melhores casos de classificação. No método do PAA, a diferença para o sinal bruto foi de 2,7% para o caso sem sobreposição e de 5,94% no caso com sobreposição.

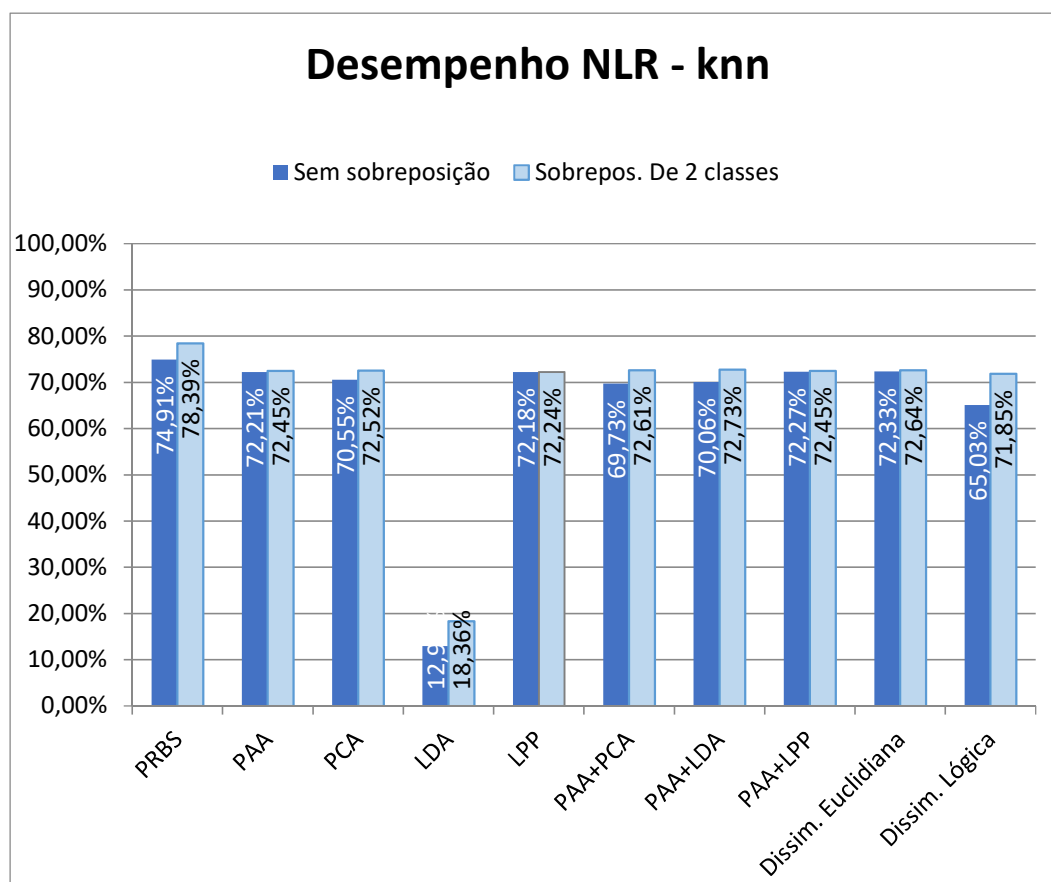


Figura 46: Desempenho por representação de dados, classificador knn, NLR

Para os métodos de redução de dimensão, os métodos PCA e LPP tiveram resultados equivalentes com sobreposição, sendo cerca de 5,82% inferiores ao com o sinal bruto neste caso. Sem sobreposição, o LPP obteve um melhor resultado, sendo 1,63% superior ao PCA e 2,73% inferior ao sinal bruto. O LDA não obteve novamente um resultado satisfatório.

Para os métodos combinados PAA+DR, os três casos apresentaram desempenhos similares entre si e equivalentes em relação aos métodos individuais, exceto o PAA+LDA com relação ao LDA. O menor desempenho sem sobreposição (PAA+ PCA) foi 5,18% inferior ao sinal bruto e o maior (PAA+LPP) foi 2,18% inferior.

Os métodos de representação por dissimilaridade apresentaram resultados satisfatórios em ambos os casos, mas ainda inferiores ao desempenho do sinal bruto. Para a dissimilaridade euclidiana, a diferença para o sinal PRBS foi de 2,58% sem sobreposição e de 5,75% com sobreposição. Na dissimilaridade lógica, o resultado obtido para o caso sem



sobreposição foi 9,88% inferior ao sinal bruto e para o caso com sobreposição foi 6,54% inferior ao mesmo caso no sinal bruto.

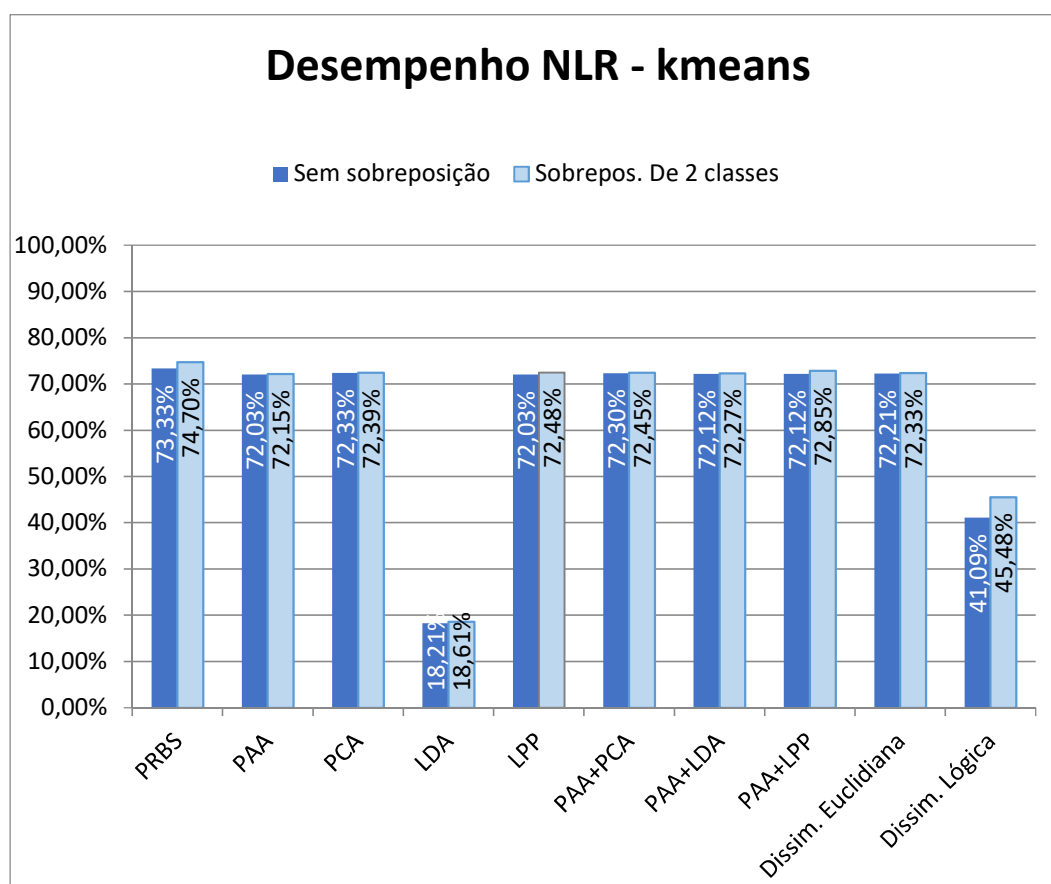


Figura 47: Desempenho por representação de dados, classificador *kmeans*, NLR

Para o classificador de reconstrução *kmeans* (fig. 47), assim como no caso do *knn* e do *Gauss*, a maior parte dos resultados foi equivalente, com exceção do LDA e da dissimilaridade lógica, para ambos os casos de sobreposição e com o sinal bruto PRBS obtendo os melhores resultados. No PAA, obtivemos uma classificação sem sobreposição 1,3% inferior ao sinal bruto PRBS e uma diferença de 2,55% da classificação PAA para o sinal bruto com sobreposição. Por sua vez, a diferença entre estes dois casos foi de 0,12%.

Nos métodos de redução de dimensão, a representação por PCA apresentou uma diferença entre os dois casos de 0,06% e um desempenho inferior ao sinal bruto em 1,00% e 2,31% respectivamente sem e com sobreposição. A representação pelo método LPP teve uma diferença de

desempenho de 0,45% entre seus casos de sobreposição e teve desempenho inferior ao sinal bruto em 1,00% e 2,22%, sem e com sobreposição. O método de redução pelo LDA não obteve resultados satisfatórios também para este classificador.

No caso dos métodos de representação combinados, o PAA+PCA obteve uma diferença entre desempenhos de 0,15% entre os casos de sobreposição considerados, sendo comparável ao obtido pelo PAA e inferior ao obtido pelo PCA, mas superior ao obtido pelo sinal bruto. O PAA+LDA apresentou diferença entre desempenhos de 0,15% também, enquanto o LPP apresentou diferença entre desempenhos de 0,73%. Todos os casos apresentaram um desempenho inferior ao sinal bruto similar entre 2% e 3% para ambos os casos.

Para os casos das representações por dissimilaridade, na dissimilaridade euclidiana obteve-se uma diferença entre os dois casos de sobreposição de 0,12%. Nesta representação, o desempenho foi inferior à classificação com o sinal bruto em 1,12% e 2,4% sem e com sobreposição respectivamente. Na dissimilaridade lógica, não se obteve bons resultados com este classificador no NLR, para esta representação, a diferença entre os dois casos de sobreposição foi de 4,39%. Nesta representação, o desempenho foi inferior à classificação com o sinal bruto em 32,24% e 29,22% sem e com sobreposição respectivamente.

Para o classificador de *Parzen* (fig. 48), assim como em outros circuitos, foram obtidos desempenhos insatisfatórios para vários métodos de representação de dados: no sinal bruto PRBS, no LDA e também, neste circuito, no LPP.

No PAA, a diferença entre os casos sem e com sobreposição foi de 0,27%. Nos métodos de redução de dimensão, somente o PCA apresentou desempenho comparável com os outros métodos. Nesta representação, a diferença entre os casos de sobreposição fica em 1,51%.

Para os métodos combinados, o PAA+PCA apresentou uma diferença entre desempenhos de 0,15%, o PAA+LDA apresentou uma diferença de 0,24%, com um desempenho equivalente aos outros métodos, contra o desempenho insatisfatório do LDA individualmente. O mesmo ocorre no LPP que, quando combinado com o PAA, é capaz de obter resultados para este

classificador neste circuito. Para este método combinado de representação, a diferença entre desempenhos foi de 0,12%.

Na dissimilaridade euclidiana, não houve diferença entre o desempenho na classificação sem sobreposição e a classificação com sobreposição. Nesta representação também, o classificador obteve o melhor resultado entre os CCUs para este circuito, com 99,64% de acertos, o equivalente a 3288 de 3300 circuitos com suas falhas corretamente classificadas.

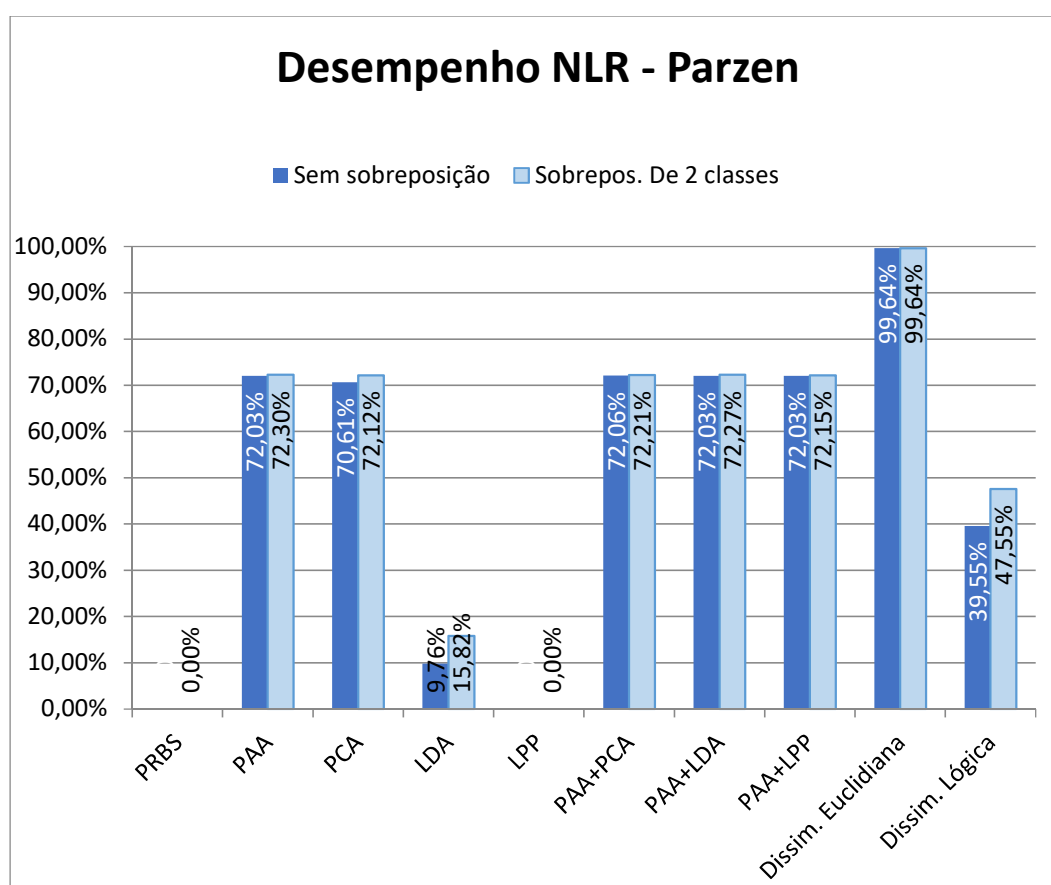


Figura 48: Desempenho por representação de dados, classificador Parzen, NLR

É importante lembrar também que, o fraco desempenho dos CCUs neste circuito ocorre devido à sobreposição severa de classes causada pelo capacitor C1, que praticamente não possui influência paramétrica no circuito, o que faz as suas classes de falhas se confundirem com a classe sem falhas no circuito. Ainda assim, os métodos de representação conseguiram reduzir a diferença entre desempenhos dos casos com e sem sobreposição.

#### 6.4.4. Desempenho para a representação de dados

Além da avaliação do circuito realizada para o sinal PRBS, também foi verificado o comportamento deste circuito quando são aplicadas técnicas de representação de dados, com o objetivo de potencializar a diferenciação entre as diferentes classes. Na figura 49 é possível observar os resultados de desempenho de cada classificador para o método de representatividade que obteve o melhor desempenho com o mesmo.

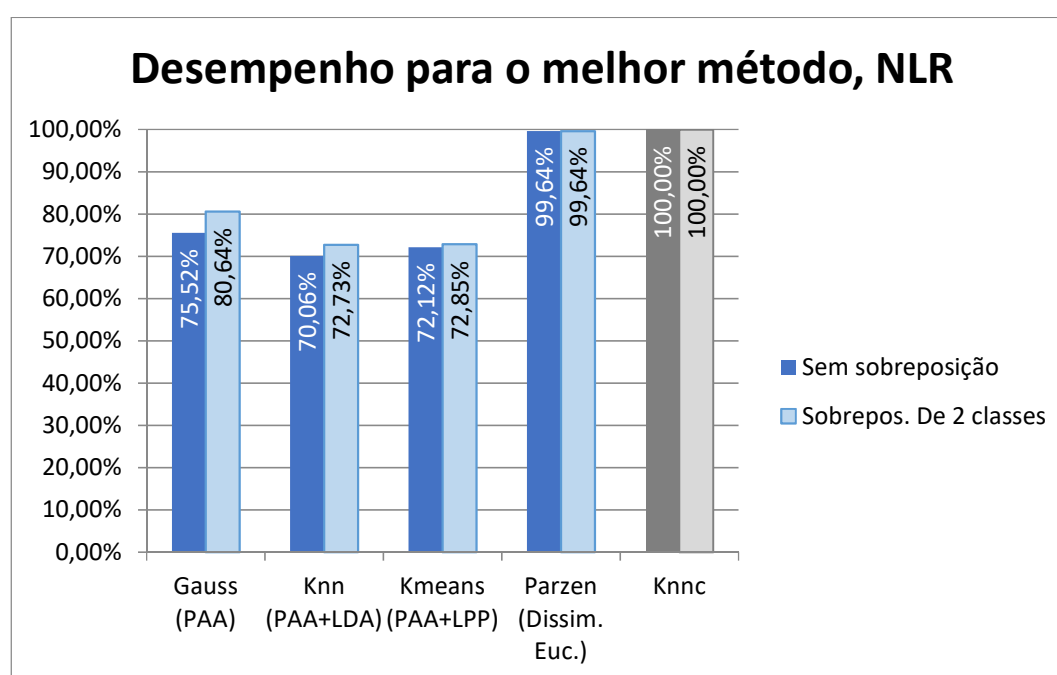


Figura 49: Curva de desempenho para os melhores métodos, circuito NLR

Como pode ser observado, o desempenho da utilização de um método de representação de dados para a classificação neste circuito tende a melhorar o desempenho do classificador, os métodos PAA+DR foram os que obtiveram os melhores resultados neste circuito. O desempenho dos CCUs só não pôde ser melhor devido à perda na sobreposição entre as classes de C1 e as respostas dos casos sem falhas, exceto no *Parzen*. Por sua vez, o *Parzen* foi o classificador com o melhor desempenho com a representação de dados. A tabela 28 mostra o desempenho detalhado deste classificador na determinada representação.

Tabela 28: Distribuição dos dados para o melhor classificador na melhor sua melhor representação, NLR.

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de rejeição
1	300	100,00%	0	0,00%	0	0,00%
2	299	99,67%	0	0,00%	1	0,33%
3	298	99,33%	0	0,00%	2	0,67%
4	297	99,00%	0	0,00%	3	1,00%
5	300	100,00%	0	0,00%	0	0,00%
6	300	100,00%	0	0,00%	0	0,00%
7	299	99,67%	0	0,00%	1	0,33%
8	297	99,00%	0	0,00%	3	1,00%
9	298	99,33%	0	0,00%	2	0,67%
10	300	100,00%	0	0,00%	0	0,00%
11	300	100,00%	0	0,00%	0	0,00%
Total:	3.288	99,64%	0	0,00%	12	0,36%

#### **6.4.5. Desempenho para classes não previstas**

Aqui foi avaliado o desempenho dos classificadores frente ao último circuito de estudo deste trabalho. É um pouco mais interessante a avaliação deste circuito para este tipo de problema do que os demais circuitos, pelo fato de ser o único circuito do estudo que não é um filtro, além de ser o único circuito passivo deste trabalho, abrangendo uma categoria diferente de circuitos que é pouco utilizada para este tipo de estudo. Além disso, temos neste circuito uma sobreposição de falhas causada por baixa sensibilidade do circuito a um componente, visto no subcapítulo 6.4.3, o que torna esta avaliação ainda mais interessante.

Nesta parte, foi estudado como cada tipo de classificador se comportaria frente a um problema imprevisto no retificador não-linear. Cada classificador será submetido ao conjunto de dados com 7 classes não previstas para avaliação, sendo 4 classes referentes a falhas múltiplas e 3 classes totalmente desconhecidas para o NLR. As respectivas classes podem ser vistas na tabela 29.

Tabela 29: Componentes com falha e suas respectivas classes para classes não previstas, NLR

Componente	Falha Associada	Respectiva Classe
R1	R1+	1
	R1-	2
L1	L1+	3
	L1-	4
C1	C1+	5
	C1-	6
C2	C2+	7
	C2-	8
C3	C3+	9
	C3-	10
NoFail		11
M1	(C1+, C2+)	12
	(C1-, C2-)	13
M2	(C1+, C2-)	14
	(C1-, C2+)	15
X1		16
X2		17
X3		18

Os resultados obtidos para o classificador multiclasse são apresentados na tabela 30. O classificador multiclasse ainda demonstra robustez e bons resultados para os dados treinados, mas total deficiência para lidar com as falhas imprevistas.

Tabela 30: Resultado de classificação multiclasse para falhas não previstas, NLR

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de Erro
1	300	100,00%	0	0,00%
2	300	100,00%	0	0,00%
3	300	100,00%	0	0,00%

4	300	100,00%	0	0,00%
5	300	100,00%	0	0,00%
6	300	100,00%	0	0,00%
7	300	100,00%	0	0,00%
8	300	100,00%	0	0,00%
9	300	100,00%	0	0,00%
10	300	100,00%	0	0,00%
11	300	100,00%	0	0,00%
12	0	0,00%	300	100,00%
13	0	0,00%	300	100,00%
14	0	0,00%	300	100,00%
15	0	0,00%	300	100,00%
16	0	0,00%	300	100,00%
17	0	0,00%	300	100,00%
18	0	0,00%	300	100,00%
Total:	3.300	61,11%	2.100	38,89%

Na figura 50 são apresentados os gráficos de desempenho de cada classificador para a avaliação com falhas imprevistas no circuito NLR. Os CCUs demonstram novamente bons resultados, havendo falhas imprevistas ou não.

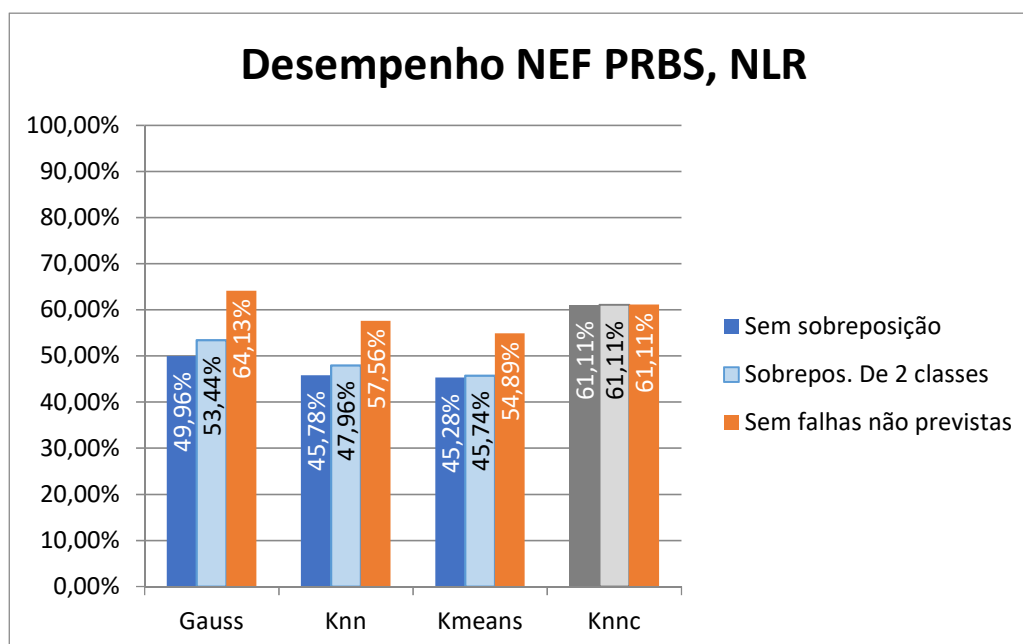


Figura 50: Desempenho dos classificadores para falhas não previstas com sinal PRBS, NLR (Parzen não classificou)

Para este caso, o classificador que obteve o melhor desempenho foi o gaussiano. A tabela 31 mostra os resultados detalhados deste classificador para o tratamento das falhas não previstas.

Tabela 31: Resultados para o melhor classificador do CTSV na curva PRBS para falhas não previstas

Classe	Dados Corretos	Taxa de acertos	Dados Incorretos	Taxa de erros	Desconhecidos	Taxa de desconhecidos
1	297	99,00%	0	0,00%	3	1,00%
2	297	99,00%	0	0,00%	3	1,00%
3	297	99,00%	0	0,00%	3	1,00%
4	297	99,00%	0	0,00%	3	1,00%
5	297	99,00%	0	0,00%	3	1,00%
6	14	4,67%	0	0,00%	3	1,00%
7	297	99,00%	0	0,00%	3	1,00%
8	297	99,00%	0	0,00%	3	1,00%
9	297	99,00%	0	0,00%	3	1,00%
10	297	99,00%	0	0,00%	3	1,00%
11	11	3,67%	0	0,00%	3	1,00%
44 (5 e 11)	157	26,17%	0	0,00%	0	0,00%



76 (6 e 11)	31	5,17%	0	0,00%	0	0,00%
108 (5, 6 e 11)	0	0,00%	381	42,33%	0	0,00%
12	0	0,00%	131	43,67%	169	56,33%
13	0	0,00%	297	99,00%	3	1,00%
14	0	0,00%	166	55,33%	134	44,67%
15	0	0,00%	292	97,33%	8	2,67%
16	0	0,00%	0	0,00%	300	0,00%
17	0	0,00%	0	0,00%	300	100,00%
18	0	0,00%	0	0,00%	300	100,00%
Total:	2.886	53,44%	1.267	23,46%	1.247	23,09%
Total não rejeitado	2.886	64,13%	1.267	28,16%		

Para este caso, os CCUs apresentaram um desempenho equivalente ao classificador multiclasse. Apesar dos resultados não terem apresentado a indiferença às falhas não previstas como nos outros circuitos, é importante ressaltar que a perda adicional causada pelas falhas não previstas estão relacionadas às classes não previstas 12, 13, 14 e 15, propositalmente escolhidas como classes de falhas múltiplas com o capacitor C1, que foi mostrado no subcapítulo 6.4.2 a baixíssima sensibilidade do circuito NLR às variações paramétricas deste componente. Estes resultados então, ratificam os resultados obtidos no subcapítulo 6.4.2.

#### **6.4.6. Comparação com o multiclasse**

Aqui foi realizada a comparação entre os desempenhos dos CCUS e do multiclasse para cada tipo de representação de dados.

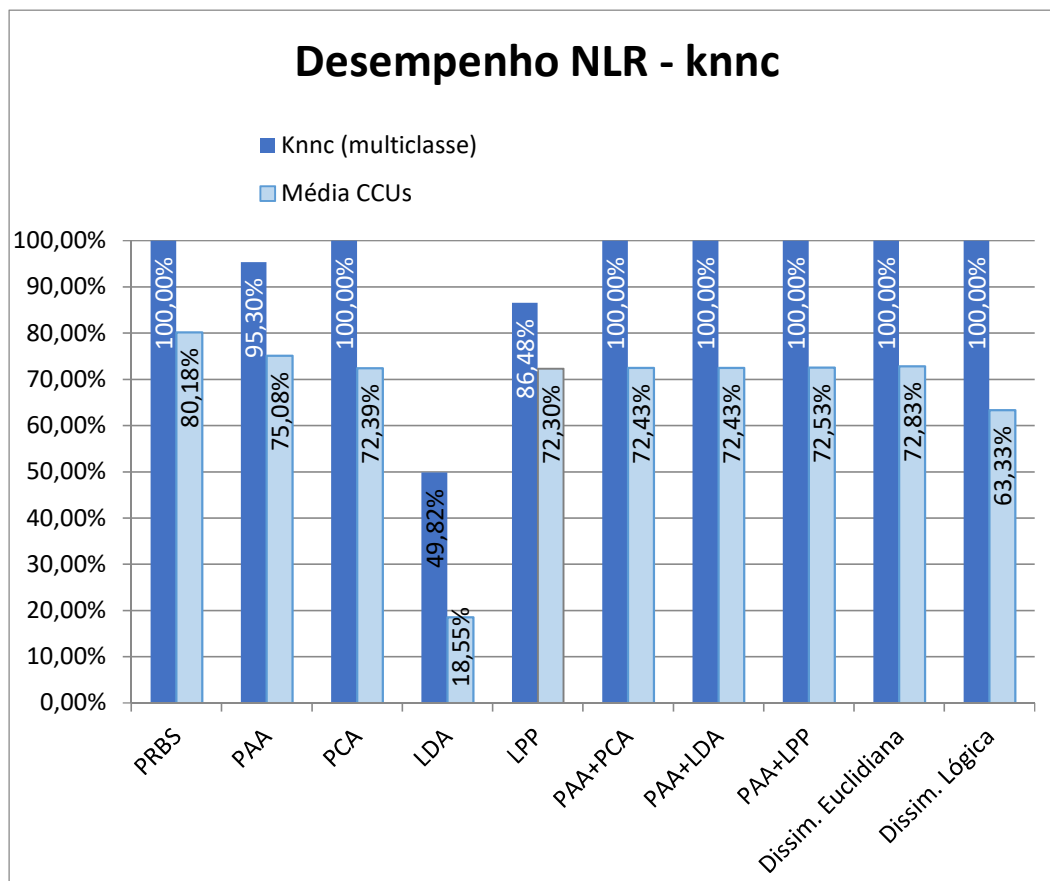


Figura 51: Desempenho multiclasse x CCUs, NLR

O classificador multiclasse obteve o máximo de desempenho em 7 dos 10 métodos de representação para este circuito, mesmo havendo o problema entre as classes 5,6 e 11. Como existe a sobreposição para os CCUs, a classificação com estes classificadores fica prejudicada neste circuito, pois o tipo de sobreposição ocorrido neste circuito envolve 3 classes, sendo uma delas a classe sem falhas. Com isso, todos os dez métodos diferentes de representação de dados tiveram uma superioridade do classificador multiclasse com relação aos CCUs.