



**DOUGLAS ROBERTO FERNANDES DAMASCENO**

**CLASSIFICADORES NÃO INTRUSIVOS DE CARGAS  
ELÉTRICAS INDUSTRIAIS UTILIZANDO TÉCNICAS DE  
INTELIGÊNCIA COMPUTACIONAL**

**LAVRAS – MG**

**2023**

**DOUGLAS ROBERTO FERNANDES DAMASCENO**

**CLASSIFICADORES NÃO INTRUSIVOS DE CARGAS ELÉTRICAS INDUSTRIAIS  
UTILIZANDO TÉCNICAS DE INTELIGÊNCIA COMPUTACIONAL**

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do programa de Pós-graduação em Engenharia de Sistemas e Automação, para obtenção do título de Mestre.

Prof. Dr. Wilian Soares Lacerda

Orientador

**LAVRAS – MG**

**2023**

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca  
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Damasceno, Douglas Roberto Fernandes.

Classificadores não intrusivos de cargas elétricas industriais  
utilizando técnicas de inteligência computacional / Douglas Roberto  
Fernandes Damasceno. - 2023.

153 p. : il.

Orientador(a): Wilian Soares Lacerda.

Dissertação (mestrado acadêmico) - Universidade Federal de  
Lavras, 2023.

Bibliografia.

1. Monitoramento de Carga Não Intrusivo. 2. Perceptron de  
Multicamadas. 3. Métodos de clusterização Fuzzy. I. Lacerda,  
Wilian Soares. II. Título.

**DOUGLAS ROBERTO FERNANDES DAMASCENO**

**CLASSIFICADORES NÃO INTRUSIVOS DE CARGAS ELÉTRICAS INDUSTRIAIS  
UTILIZANDO TÉCNICAS DE INTELIGÊNCIA COMPUTACIONAL  
NON-INTRUSIVE CLASSIFIERS OF INDUSTRIAL ELECTRIC LOADS USING  
COMPUTATIONAL INTELLIGENCE TECHNIQUES**

Dissertação apresentada à Universidade Federal de Lavras como parte das exigências do programa de Pós-graduação em Engenharia de Sistemas e Automação, para obtenção do título de Mestre.

APROVADA em 04 de abril de 2023.

Prof. Dr. Danton Diego Ferreira                      UFLA  
Profa. Dra. Lane Maria Rabelo Baccarini      UFSJ

  
Prof. Dr. Wilian Soares Lacerda  
Orientador

**LAVRAS – MG  
2023**

*À minha família, agradeço por todo o apoio desde o momento em que fui aprovado, e a Deus,  
por me fortalecer nos momentos mais difíceis da minha vida.*

## AGRADECIMENTOS

Sou grato a Deus por todas as bênçãos em minha vida, em especial à minha mãe, **Maria Helena Fernandes Damasceno**, por todo o amor, apoio e incentivo em minha jornada.

À minha família, em especial minha esposa **Mariana Jaqueline de Paula Ferreira** e meus filhos **Davi Ferreira Damasceno** e **Lucas Ferreira Damasceno**, agradeço pelo apoio nos momentos mais difíceis e por serem a razão de todo o meu esforço.

Ao **Dr. Wilian Soares Lacerda** pela amizade e por me orientar, estando sempre disponível durante o mestrado.

Ao **Dr. Danton Diego Ferreira**, à **Dra. Lane Maria Rabelo Baccarini**, ao **Dr. Bruno Henrique Groenner Barbosa** e ao **Dr. Paulo Cezar Monteiro Lamim** por aceitarem serem membros banca.

A todos que de alguma forma colaboraram para que eu conseguisse concluir o mestrado.

*"Porque Deus é quem efetua em vós tanto o querer como o realizar, segundo a sua boa vontade."  
(Filipenses 2:13)*

## RESUMO

O aprimoramento da gestão de energia tem exigido a realização de tarefas fundamentais como o monitoramento de cargas elétricas, em virtude da atual conjuntura econômica e das crescentes tendências ecológicas. Este trabalho apresenta um método de identificação e classificação de cinco cargas industriais de uma linha de produção de amortecedores automobilísticos, sendo: uma prensa válvula, uma dosadora de óleo, um teste de tração, um dinamômetro e uma roladora. Para coleta dos dados de treinamento dos classificadores propostos, as cargas foram acionadas individualmente e os dados do sinal da corrente elétrica foram obtidos por meio da técnica de Monitoramento de Cargas Não Intrusivo. Como métodos de classificação, foram implementados os seguintes algoritmos de aprendizagem de máquina: Redes Neurais Artificiais (RNA) do tipo Perceptron de Multicamadas (*Multilayer Perceptron*, MLP), as Máquinas de Vetores de Suporte (*Support Vector Machines*, SVM) e também os métodos de clusterização fuzzy K-Means (KM), Fuzzy C-Means (FCM) e Gustafson-Kessel (GK). Com a finalidade de se obter os principais parâmetros do MLP e das SVMs, aplicou-se três técnicas de otimização, sendo a Otimização por Enxame de Partículas (*Particle Swarm Optimization*, PSO), a Evolução Diferencial (*Differential Evolution*, DE) e o Otimizador Lobo Cizento (*Grey Wolf Optimizer*, GWO). Já nos métodos de clusterização, para determinar o número eficiente de clusters, utilizou-se os índices de validação Critério de Xie-Beni (*Xie-Beni criterion*, XB), Entropia de Classificação (*Classification Entropy*, CE), Índice de Partição (*Partition Index*, SC) e Índice de Dunn (*Dunn's Index*, DI) para cada método proposto. O melhor classificador obtido, comparando-se os classificadores MLP e as SVMs, foi o MLP-PSO o qual apresentou entre as principais métricas de desempenho a precisão de 0,9242, acurácia de 0,9139, F1-score de 0,9140 e o coeficiente Kappa de 0,8926 demonstrando a eficácia do classificador para a identificação e classificação das cargas da linha de produção. Em relação aos métodos de clusterização, destacou-se o GK, o qual apresentou precisão de 0,8025, acurácia 0,7816, F1-score 0,7745 e coeficiente Kappa 0,7216, sendo estes valores abaixo do esperado.

**Palavras-chave:** Monitoramento de Carga Não Intrusivo. Perceptron de Multicamadas. Máquinas de Vetores de Suporte. Algoritmos de otimização. Métodos de clusterização Fuzzy.

## ABSTRACT

Improving energy management has required performing fundamental tasks such as monitoring electrical loads, due to the current economic situation and growing ecological trends. This work presents a method of identification and classification of five industrial loads of a automotive shock absorber production line, namely: a valve press, an oil doser, a traction test, a dynamometer and a roller. In order to collect the training data of the proposed classifiers, the loads were triggered individually and the electrical current signal data were obtained through the Non-Intrusive Load Monitoring technique. As classification methods, the following machine learning algorithms were implemented: Artificial Neural Networks (ANN) of the Multilayer Perceptron (MLP), Support Vector Machines (SVM) and also the fuzzy clustering methods K-Means (KM), Fuzzy C-Means (FCM) and Gustafson-Kessel (GK). In order to obtain the main parameters of the MLP and SVMs, three optimization techniques were applied, namely Particle Swarm Optimization (PSO), Differential Evolution (DE) and the Gray Wolf Optimizer (GWO). As for the clustering methods, to determine the efficient number of clusters, the validation indices Xie-Beni Criterion (XB), Classification Entropy (CE), Partition Index (SC) and Dunn Index (DI) for each proposed method. The best classifier obtained, comparing the MLP classifiers and the SVMs, was the MLP-PSO, which presented among the main performance metrics a precision of 0.9556, F1-score of 0.9478, accuracy of 0.9474 and the Kappa coefficient of 0.9345 demonstrating the effectiveness of the classifier. Regarding the clustering methods, the GK stood out, which presented precision of 0.8472, accuracy 0.8378, F1-score 0.8398 and Kappa coefficient 0.7991, these values being lower than expected, and therefore not being applicable for classification of loads.

**Keywords:** Non-intrusive Load Monitoring. Multilayer Perceptron. Support Vector Machines. Optimization algorithms. Fuzzy clustering methods.

## LISTA DE FIGURAS

Figura 1.1 – Aplicações da digitalização no setor elétrico. . . . .	18
Figura 3.1 – Sistema de monitoramento de cargas intrusivo. . . . .	30
Figura 3.2 – Sistema de monitoramento de cargas não intrusivo. . . . .	30
Figura 3.3 – Neurônio do tipo Perceptron. . . . .	33
Figura 3.4 – Rede com múltiplas camadas. . . . .	35
Figura 3.5 – Direção de dois fluxos básicos de sinais no MLP. . . . .	36
Figura 3.6 – Diagrama de bloco da aprendizagem supervisionada. . . . .	37
Figura 3.7 – Diagrama de bloco da aprendizagem não supervisionada. . . . .	37
Figura 3.8 – Fluxo de sinal destacando os detalhes da saída do neurônio $j$ . . . . .	38
Figura 3.9 – Mapa do SVM dos espaços de entradas em um hiperplano dimensional. . . . .	41
Figura 3.10 – Exemplo de separação de duas classes utilizando o produto interno do tipo polinomial. . . . .	42
Figura 4.1 – Fluxograma da primeira etapa. . . . .	58
Figura 4.2 – Fluxograma do processo de treinamento e análise dos resultados para o MLP e SVM. . . . .	59
Figura 4.3 – Fluxograma do processo de treinamento e análise dos resultados para os métodos de clusterização. . . . .	60
Figura 4.4 – Analisador de qualidade de energia <i>KYORITSI KEW 6310</i> . . . . .	61
Figura 4.5 – Diagrama do analisador de qualidade de energia. . . . .	62
Figura 4.6 – Instalação do aparelho no painel elétrico da linha de produção. . . . .	63
Figura 4.7 – Diagrama de conexão. . . . .	63
Figura 4.8 – Conexão dos plugues e braçadeiras nos barramentos de alimentação. . . . .	64
Figura 4.9 – K-fold para $k = 4$ . . . . .	68
Figura 5.1 – Amostra da corrente no momento em que a C1 inicia a produção do produto. . . . .	74
Figura 5.2 – DFT da corrente das cinco cargas em análise. . . . .	75
Figura 1 – Matriz de confusão MLP-PSO partição 1. . . . .	117
Figura 2 – Matriz de confusão MLP-PSO partição 2. . . . .	118
Figura 3 – Matriz de confusão MLP-PSO partição 3. . . . .	119
Figura 4 – Matriz de confusão MLP-PSO partição 4. . . . .	120
Figura 5 – Matriz de confusão MLP-DE partição 1. . . . .	121
Figura 6 – Matriz de confusão MLP-DE partição 2. . . . .	122

Figura 7 – Matriz de confusão MLP-DE partição 3. . . . .	123
Figura 8 – Matriz de confusão MLP-DE partição 4. . . . .	124
Figura 9 – Matriz de confusão MLP-GWO partição 1. . . . .	125
Figura 10 – Matriz de confusão MLP-GWO partição 2. . . . .	126
Figura 11 – Matriz de confusão MLP-GWO partição 3. . . . .	127
Figura 12 – Matriz de confusão MLP-GWO partição 4. . . . .	128
Figura 13 – Matriz de confusão SVM-PSO partição 1. . . . .	129
Figura 14 – Matriz de confusão SVM-PSO partição 2. . . . .	130
Figura 15 – Matriz de confusão SVM-PSO partição 3. . . . .	131
Figura 16 – Matriz de confusão SVM-PSO partição 4. . . . .	132
Figura 17 – Matriz de confusão SVM-DE partição 1. . . . .	133
Figura 18 – Matriz de confusão SVM-DE partição 2. . . . .	134
Figura 19 – Matriz de confusão SVM-DE partição 3. . . . .	135
Figura 20 – Matriz de confusão SVM-DE partição 4. . . . .	136
Figura 21 – Matriz de confusão SVM-GWO partição 1. . . . .	137
Figura 22 – Matriz de confusão SVM-GWO partição 2. . . . .	138
Figura 23 – Matriz de confusão SVM-GWO partição 3. . . . .	139
Figura 24 – Matriz de confusão SVM-GWO partição 4. . . . .	140
Figura 25 – Matriz de confusão do método KM obtida na partição 1 com 24 clusters. . .	142
Figura 26 – Matriz de confusão do método KM obtida na partição 2 com 20 clusters. . .	143
Figura 27 – Matriz de confusão do método KM obtida na partição 3 com 25 clusters. . .	144
Figura 28 – Matriz de confusão do método KM obtida na partição 4 com 22 clusters. . .	145
Figura 29 – Matriz de confusão do método FCM obtida na partição 1 com 24 clusters. . .	146
Figura 30 – Matriz de confusão do método FCM obtida na partição 2 com 21 clusters. . .	147
Figura 31 – Matriz de confusão do método FCM obtida na partição 3 com 24 clusters. . .	148
Figura 32 – Matriz de confusão do método FCM obtida na partição 4 com 22 clusters. . .	149
Figura 33 – Matriz de confusão do método GK obtida na partição 1 com 23 clusters. . .	150
Figura 34 – Matriz de confusão do método GK obtida na partição 2 com 23 clusters. . .	151
Figura 35 – Matriz de confusão do método GK obtida para a partição 3 com 25 clusters. .	152
Figura 36 – Matriz de confusão do método GK obtida na partição 4 com 24 clusters. . .	153

## LISTA DE TABELAS

Tabela 2.1 – Trabalhos referentes à gestão de energia. . . . .	28
Tabela 3.1 – Matriz de confusão. . . . .	53
Tabela 3.2 – Níveis de concordância do índice Kappa. . . . .	56
Tabela 4.1 – Classes (C) para as cargas industriais. . . . .	60
Tabela 4.2 – Quantidade de formas de onda de um período de 60Hz utilizadas na DFT para cada classe. . . . .	66
Tabela 4.3 – Hiperparâmetros das técnicas de otimização PSO, DE e GWO. . . . .	70
Tabela 5.1 – Índices e harmônicos obtidos ao aplicar o FDR. . . . .	76
Tabela 5.2 – Métricas dos classificadores MLP-PSO. . . . .	77
Tabela 5.3 – Métricas dos classificadores MLP-DE. . . . .	78
Tabela 5.4 – Métricas dos classificadores MLP-GWO. . . . .	79
Tabela 5.5 – Média das métricas dos classificadores MLP. . . . .	80
Tabela 5.6 – Métricas dos classificadores SVM-PSO. . . . .	81
Tabela 5.7 – Métricas dos classificadores SVM-DE. . . . .	81
Tabela 5.8 – Métricas dos classificadores SVM-GWO. . . . .	82
Tabela 5.9 – Métricas dos classificadores SVM. . . . .	83
Tabela 5.10 – Métricas dos melhores classificadores MLP e SVM. . . . .	84
Tabela 5.11 – Valores dos índices de validação do método KM obtidos na partição 1. . . . .	85
Tabela 5.12 – Acertos e erros do classificador do método KM obtidos na partição 1 com 24 clusters. . . . .	86
Tabela 5.13 – Valores dos índices de validação do método KM obtidos na partição 2. . . . .	86
Tabela 5.14 – Métricas de desempenho do método KM obtidas na partição 2 com 20 clusters. . . . .	86
Tabela 5.15 – Valores dos índices de validação do método KM obtidos na partição 3. . . . .	87
Tabela 5.16 – Métricas de desempenho do método KM obtidas na partição 2 com 25 clusters. . . . .	87
Tabela 5.17 – Valores dos índices de validação do método KM obtidos na partição 4. . . . .	88
Tabela 5.18 – Métricas de desempenho do método KM obtidas na partição 4 com 22 clusters. . . . .	88
Tabela 5.19 – Métricas dos classificadores KM. . . . .	89
Tabela 5.20 – Valores dos índices de validação do método FCM obtidos na partição 1. . . . .	90
Tabela 5.21 – Métricas de desempenho do método FCM obtidas na partição 1 com 24 clusters. . . . .	90
Tabela 5.22 – Valores dos índices de validação do método FCM obtidos na partição 2. . . . .	91

Tabela 5.23 – Métricas de desempenho do método FCM obtidas na partição 3 com 21 clusters. . . . .	91
Tabela 5.24 – Valores dos índices de validação do método FCM obtidos na partição 3. . .	92
Tabela 5.25 – Métricas de desempenho do método FCM obtidas na partição 3 com 24 clusters. . . . .	92
Tabela 5.26 – Valores dos índices de validação do método FCM obtidos na partição 4. . .	93
Tabela 5.27 – Métricas de desempenho do método FCM obtidas na partição 4 com 22 clusters. . . . .	93
Tabela 5.28 – Métricas dos classificadores FCM. . . . .	93
Tabela 5.29 – Valores dos índices de validação do método GK obtidos na partição 1. . . .	94
Tabela 5.30 – Métricas de desempenho do método GK obtidas na partição 1 com 23 clusters.	94
Tabela 5.31 – Valores dos índices de validação do método GK obtidos na partição 2. . . .	95
Tabela 5.32 – Métricas de desempenho do método GK obtidas na partição 2 com 23 clusters.	95
Tabela 5.33 – Métricas de desempenho do método GK obtidas na partição 3 com 25 clusters.	95
Tabela 5.34 – Valores dos índices de validação do método GK obtidos na partição 3. . . .	96
Tabela 5.35 – Métricas de desempenho do método GK obtidas na partição 4 com 24 clusters.	96
Tabela 5.36 – Valores dos índices de validação do método GK obtidos na partição 4. . . .	97
Tabela 5.37 – Métricas dos classificadores GK. . . . .	97
Tabela 5.38 – Métricas dos classificadores KM, GK e FCM. . . . .	98
Tabela 5.39 – Métricas dos melhores classificadores das técnicas de aprendizado supervisionado e não supervisionado. . . . .	98
Tabela 1 – Acertos e erros do classificador MLP-PSO obtidos na partição 1. . . . .	120
Tabela 2 – Acertos e erros do classificador MLP-PSO obtidos na partição 2. . . . .	120
Tabela 3 – Acertos e erros do classificador MLP-PSO obtidos na partição 3. . . . .	121
Tabela 4 – Acertos e erros do classificador MLP-PSO obtidos na partição 4. . . . .	121
Tabela 5 – Acertos e erros do classificador MLP-DE obtidos na partição 1. . . . .	124
Tabela 6 – Acertos e erros do classificador MLP-DE obtidos na partição 2. . . . .	124
Tabela 7 – Acertos e erros do classificador MLP-DE obtidos na partição 3. . . . .	125
Tabela 8 – Acertos e erros do classificador MLP-DE obtidos na partição 4. . . . .	125
Tabela 9 – Acertos e erros do classificador MLP-GWO obtidos na partição 1. . . . .	128
Tabela 10 – Acertos e erros do classificador MLP-GWO obtidos na partição 2. . . . .	128
Tabela 11 – Acertos e erros do classificador MLP-GWO obtidos na partição 3. . . . .	129

Tabela 12 – Acertos e erros do classificador MLP-GWO obtidos na partição 4. . . . .	129
Tabela 13 – Acertos e erros do classificador SVM-PSO obtidos na partição 1. . . . .	132
Tabela 14 – Acertos e erros do classificador SVM-PSO obtidos na partição 2. . . . .	132
Tabela 15 – Acertos e erros do classificador SVM-PSO obtidos na partição 3. . . . .	133
Tabela 16 – Acertos e erros do classificador SVM-PSO obtidos na partição 4. . . . .	133
Tabela 17 – Acertos e erros do classificador SVM-DE obtidos na partição 1. . . . .	136
Tabela 18 – Acertos e erros do classificador SVM-DE obtidos na partição 2. . . . .	136
Tabela 19 – Acertos e erros do classificador SVM-DE obtidos na partição 3. . . . .	137
Tabela 20 – Acertos e erros do classificador SVM-DE obtidos na partição 4. . . . .	137
Tabela 21 – Acertos e erros do classificador SVM-GWO obtidos na partição 1. . . . .	140
Tabela 22 – Acertos e erros do classificador SVM-GWO obtidos na partição 2. . . . .	140
Tabela 23 – Acertos e erros do classificador SVM-GWO obtidos na partição 3. . . . .	141
Tabela 24 – Acertos e erros do classificador SVM-GWO obtidos na partição 4. . . . .	141
Tabela 25 – Acertos e erros do método KM obtidos na partição 1 com 24 clusters. . . .	142
Tabela 26 – Acertos e erros do classificador do método KM obtidos na partição 2 com 20 clusters. . . . .	143
Tabela 27 – Acertos e erros do classificador do método KM obtidos na partição 3 com 25 clusters. . . . .	144
Tabela 28 – Acertos e erros do classificador do método KM obtidos na partição 4 com 22 clusters. . . . .	145
Tabela 29 – Acertos e erros do classificador do método FCM obtidos na partição 1 com 24 clusters. . . . .	146
Tabela 30 – Acertos e erros do classificador do método FCM obtidos na partição 2 com 21 clusters. . . . .	147
Tabela 31 – Acertos e erros do classificador do método FCM obtidos na partição 3 com 24 clusters. . . . .	148
Tabela 32 – Acertos e erros do classificador do método FCM obtidos na partição 4 com 22 clusters. . . . .	149
Tabela 33 – Acertos e erros do classificador do método GK obtidos na partição 1 com 23 clusters. . . . .	150
Tabela 34 – Acertos e erros do classificador do método GK obtidos na partição 2 com 23 clusters. . . . .	151

Tabela 35 – Acertos e erros do classificador do método GK obtidos na partição 3 com 25 clusters. . . . .	152
Tabela 36 – Acertos e erros do classificador do método GK obtidos na partição 4 com 24 clusters. . . . .	153

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>1.1</b>	<b>Objetivo Geral</b>	<b>20</b>
<b>1.2</b>	<b>Objetivos Específicos</b>	<b>20</b>
<b>1.3</b>	<b>Motivação</b>	<b>20</b>
<b>1.4</b>	<b>Organização do documento</b>	<b>21</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>22</b>
<b>3</b>	<b>REFERENCIAL TEÓRICO</b>	<b>29</b>
<b>3.1</b>	<b>Monitoramento de cargas intrusivo e não intrusivo</b>	<b>29</b>
<b>3.2</b>	<b>Algoritmos de pré-processamento de dados</b>	<b>30</b>
<b>3.2.1</b>	<b>Normalização dos dados</b>	<b>31</b>
<b>3.3</b>	<b>Seleção de Características</b>	<b>32</b>
<b>3.3.1</b>	<b>Razão Discriminante de Fisher</b>	<b>32</b>
<b>3.4</b>	<b>Redes Neurais Artificiais</b>	<b>33</b>
<b>3.4.1</b>	<b>Perceptron de Multicamadas</b>	<b>34</b>
<b>3.4.2</b>	<b>Processo de aprendizagem</b>	<b>35</b>
<b>3.4.2.1</b>	<b>Aprendizagem supervisionada</b>	<b>36</b>
<b>3.4.2.2</b>	<b>Aprendizagem não supervisionada</b>	<b>37</b>
<b>3.4.3</b>	<b>Algoritmo de aprendizagem – <i>backpropagation</i></b>	<b>38</b>
<b>3.5</b>	<b>Máquinas de Vetores de Suporte</b>	<b>40</b>
<b>3.6</b>	<b>Sistemas Fuzzy</b>	<b>42</b>
<b>3.7</b>	<b>Clusterização</b>	<b>44</b>
<b>3.8</b>	<b>Clusterização Fuzzy</b>	<b>45</b>
<b>3.8.1</b>	<b>K-Means</b>	<b>45</b>
<b>3.8.2</b>	<b>Fuzzy C-Means</b>	<b>46</b>
<b>3.8.3</b>	<b>Gustafson-Kessel</b>	<b>48</b>
<b>3.9</b>	<b>Método de validação — clusterização</b>	<b>49</b>
<b>3.9.1</b>	<b>Crítério de Xie-Beni</b>	<b>49</b>
<b>3.9.2</b>	<b>Índice de Partição</b>	<b>49</b>
<b>3.9.3</b>	<b>Entropia de Classificação</b>	<b>50</b>
<b>3.9.4</b>	<b>Índice de Dunn</b>	<b>50</b>
<b>3.10</b>	<b>Técnicas de otimização meta-heurística</b>	<b>50</b>

3.10.1	Otimização de Enxame de Partículas . . . . .	51
3.10.2	Otimizador Lobo Cinzento . . . . .	51
3.10.3	Evolução Diferencial . . . . .	52
3.11	Métodos de medição de desempenho dos sistemas de classificação . . . . .	52
3.11.1	Matriz de confusão . . . . .	52
3.11.2	Acurácia . . . . .	53
3.11.3	Sensibilidade . . . . .	53
3.11.4	Precisão . . . . .	53
3.11.5	Especificidade . . . . .	54
3.11.6	F1-score . . . . .	54
3.11.6.1	Macro F1-score . . . . .	54
3.11.6.2	Micro F1-score . . . . .	55
3.11.7	Coefficiente de concordância Kappa . . . . .	55
4	<b>METODOLOGIA . . . . .</b>	<b>57</b>
4.1	Aquisição dos dados . . . . .	59
4.2	Base de Dados . . . . .	65
4.3	Pré-processamento dos dados . . . . .	65
4.4	Seleção de características . . . . .	66
4.5	Normalização dos dados . . . . .	66
4.6	Validação cruzada . . . . .	67
4.6.1	Método K-fold . . . . .	67
4.7	Redes Neurais Artificiais . . . . .	68
4.7.1	RNA do tipo MLP . . . . .	68
4.7.2	Técnicas de otimização PSO, DE e GWO aplicadas na topologia e treinamento da RNA . . . . .	69
4.7.3	Máquina de Vetores de Suporte . . . . .	71
4.7.4	Técnicas de otimização PSO, DE e GWO aplicadas na topologia e treinamento da SVM . . . . .	71
4.7.5	Métodos de clusterização KM, FCM e GK . . . . .	72
4.8	Métricas de desempenho . . . . .	72
5	<b>RESULTADOS E DISCUSSÕES . . . . .</b>	<b>74</b>
5.1	Aprendizado supervisionado - MLP e SVM . . . . .	76

5.1.1	<b>MLP-PSO</b> . . . . .	77
5.1.2	<b>MLP-DE</b> . . . . .	78
5.1.3	<b>MLP-GWO</b> . . . . .	78
5.1.4	<b>Resultados dos classificadores MLP</b> . . . . .	79
5.1.5	<b>SVM-PSO</b> . . . . .	80
5.1.6	<b>SVM-DE</b> . . . . .	81
5.1.7	<b>SVM-GWO</b> . . . . .	82
5.1.8	<b>Resultados dos classificadores SVM</b> . . . . .	83
5.1.9	<b>Resultados dos classificadores MLP e PSO</b> . . . . .	83
5.2	<b>Aprendizagem não-supervisionado — Clusterização Fuzzy</b> . . . . .	84
5.2.1	<b>K-Means</b> . . . . .	84
5.2.2	<b>Fuzzy C-Means</b> . . . . .	89
5.2.3	<b>Gustafson-Kessel</b> . . . . .	92
5.2.4	<b>Resultados dos métodos KM, FCM e GK</b> . . . . .	98
5.2.5	<b>Resultados das técnicas de aprendizado supervisionado e não-supervisionado.</b>	98
6	<b>CONCLUSÃO</b> . . . . .	100
6.1	<b>Propostas de continuidade</b> . . . . .	101
	<b>REFERÊNCIAS</b> . . . . .	103
	<b>APENDICE A – Código-fonte MLP-GWO</b> . . . . .	109
	<b>APENDICE B – Figuras e tabelas dos classificadores do método de aprendi- zagem supervisionado</b> . . . . .	117
	<b>APENDICE C – Figuras e tabelas dos classificadores do método de aprendi- zagem não-supervisionado</b> . . . . .	142

## 1 INTRODUÇÃO

Nas últimas décadas, houve um aumento significativo na população mundial, assim como na expectativa de vida. Com a rápida urbanização, a população urbana já representa 55% da população total global a partir de 2018. Estima-se que até 2050, cerca de 68,4% da população mundial estará vivendo em áreas urbanas (KIM et al., 2021).

Com a rápida urbanização e com o aumento explosivo na demanda de energia, o impacto causado, foi, a priori, o aumento do consumo de combustíveis fósseis. Nas cidades, estes impactos são motivos de preocupação, devido aos danos ocasionados, tanto no meio ambiente quanto na saúde, não só dos seres humanos, mas de todas as forma de vida.

Em todo o mundo, nos últimos anos, bilhões de toneladas de dióxido de carbono e mais de 120 milhões de toneladas de metano, os dois principais gases de efeito estufa, foram emitidas anualmente na atmosfera a partir da produção e queima de combustíveis fósseis para energia e transporte (PERERA; NADEAU, 2022). As grandes cidades ocupam apenas 5% do território do planeta e correspondem ao uso de 75% de combustíveis fósseis (KIM et al., 2021).

Conforme o SEEG (Sistema de Estimativas de Emissões de Gases do Efeito Estufa), o Brasil emitiu 2,42 bilhões de toneladas brutas de  $CO_2$  (uma unidade que engloba todos os gases do efeito estufa) em 2021. O dado representa um aumento de pouco mais de 12% em relação à quantia registrada em 2020. Destaca-se que em 2003, o Brasil bateu 3,02 bilhões de toneladas brutas de gás carbônico (SEEG, 2022).

Devido ao uso de combustíveis fósseis, a temperatura média da superfície da Terra aumentou cerca de 1,1 °C em relação aos tempos pré-industriais. Especialistas do Painel Intergovernamental sobre Mudanças Climáticas alertam que ações urgentes são necessárias para limitar o aquecimento global a 1,5 °C, com o objetivo de minimizar as consequências mais catastróficas.

Entre as consequências catastróficas do aquecimento global pode se mencionar: o degelo, o aumento do nível dos oceanos, a desertificação, a alteração do regime das chuvas, inundações e a redução da biodiversidade. Vale destacar, por exemplo, os efeitos na saúde humana. Grande parte dos seres humanos vivem nos trópicos e não estão adaptados ao forte calor e às radiações solares nocivas, sendo fatores de riscos de diversas doenças, como alergias, infecções e doenças cardiorrespiratórias, entre outras. Isso acontece porque o aumento da temperatura pode causar mudanças na flora e, conseqüentemente, uma produção maior de pólen e quadros de alergia entre populações, favorecer a propagação de parasitas e vetores de doenças, como mosquitos que

transmitem a dengue e a malária, e ainda colaborar para o aumento de casos de stress térmico e crises de asma.

Um conceito que está crescendo e se tornando uma necessidade são as cidades inteligentes (*smart cities*). A criação das cidades inteligentes é uma estratégia para mitigar os problemas do crescimento da população urbana, acompanhado dos aumentos de emissões de gases nocivos (MOHANTY; CHOPPALI; KOUGIANOS, 2016).

Apesar do custo considerável para sua implantação, ao serem estabelecidas poderão reduzir o consumo de energia e de água, emissões de gases, resíduos urbanos, entre outros. Assim, o desenvolvimento de novas tecnologias para a economia de energia é de extrema importância. Como é sabido, o consumo consciente de energia elétrica deveria ser uma prática diária, já que colabora para o desenvolvimento sustentável do planeta e ainda contribui para reduzir os impactos ambientais negativos.

No Brasil, o Ministério de Minas e Energia (MME) está atuando para intervir diretamente no setor energético. Em 16 de dezembro de 2020, o MME aprovou o Plano Nacional de Energia 2050 (PNE) que foi elaborado pela Empresa de Pesquisa Energética (EPE). A PNE é um conjunto de estudos que dão suporte a estratégias de longo prazo do governo em relação à expansão do setor energético (BRASIL; EPE, 2020).

Segundo o PNE, a atividade industrial brasileira corresponde a 50% do consumo total de energia do país e aproximadamente 1/5 do PIB brasileiro. Deste modo, em 2050 não apenas a expansão da atividade econômica do setor desempenha papel fundamental no crescimento do consumo de energia, como também a expansão territorial deverá ser considerada. Conforme (BRASIL; EPE, 2020), “há oportunidades para aproveitamento de potenciais relevantes de efficientização no consumo de energia e redução de emissões de gases de efeito estufa (GEE)”.

Pelo fato do consumo de energia estar aumentando e a projeção de consumo ser relativamente preocupante, o desenvolvimento em tecnologias para a gestão energética são viáveis e necessárias. As empresas de gerenciamento de energia buscam cada vez mais investir em qualidade de energia e confiabilidade no fornecimento. Assim, uma alocação racional e redução no consumo de energia são os principais objetivos destas empresas.

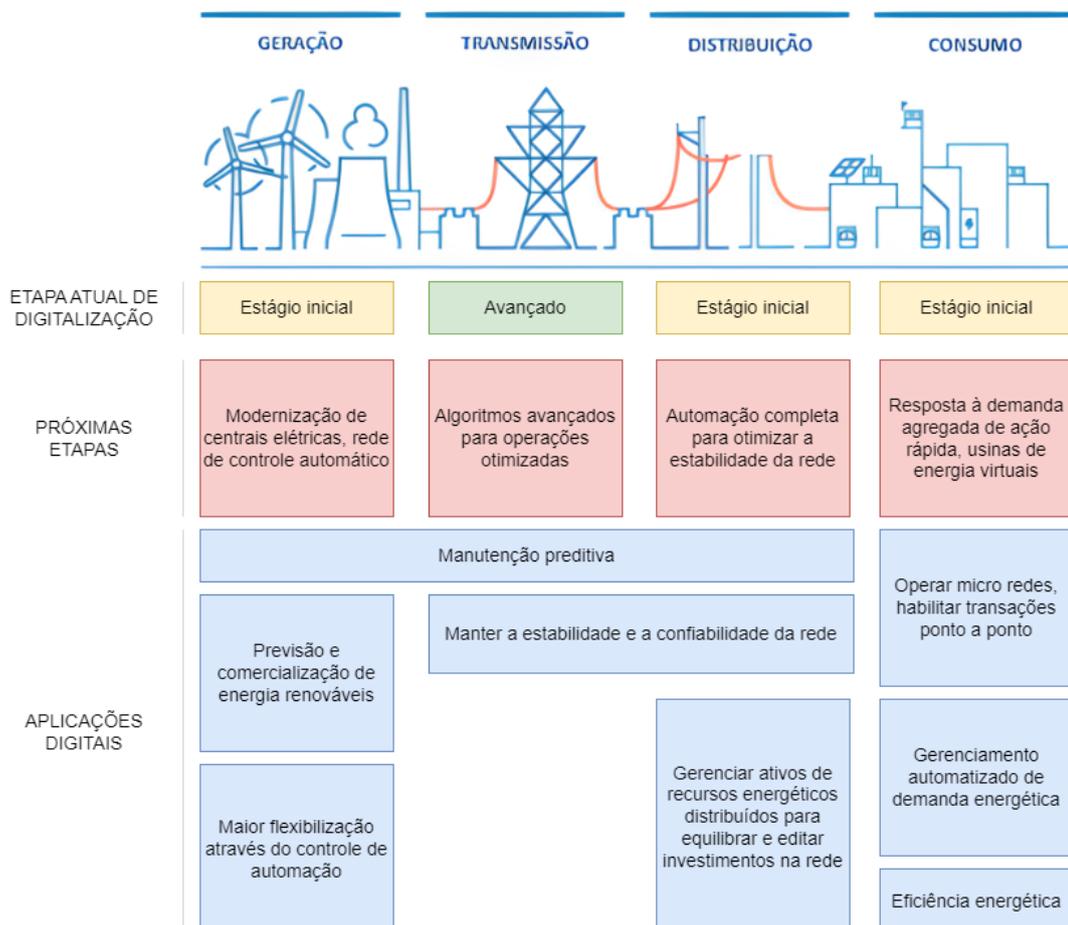
A digitalização na produção vem crescendo nos últimos anos, por meio do avanço da automação e da difusão das Tecnologias de Informação e Comunicação (TIC), associada à evolução da conectividade, coleta e análise de excesso de dados. Segundo a PNE, são necessárias medidas cabíveis para tornar o consumo de energia mais eficiente, os quais podem ser listados:

- Substituição de fontes energéticas;
- *Retrofit* das instalações existentes;
- Inovações em processos e produtos com menor conteúdo energético;
- Inserção de digitalização das operações industriais;
- Gerenciamento energético.

Com a digitalização, medidores inteligentes podem melhorar a confiabilidade e eficiência operacional, gerenciando o perfil de consumo e respondendo à variação na demanda. Além disso, há oportunidades em termos de negócios, estrutura tarifária e preços mais eficientes. O setor elétrico já está aplicando a digitalização em suas estruturas.

Na Figura 1.1, as etapas atuais e futuras da digitalização no setor elétrico são apresentadas. O setor de consumo ainda está em estágios iniciais, mas é esperado que o gerenciamento automatizado de demanda energética, com a eficiência energética, se torne necessário em etapas futuras, demandando investimentos em Pesquisa, Desenvolvimento e Inovação (PD&I).

Figura 1.1 – Aplicações da digitalização no setor elétrico.



Fonte: (BRASIL; EPE, 2020).

Atualmente, existem incentivos para o uso eficiente de energia nas indústrias, entre as quais podem ser citados (BRASIL; EPE, 2020):

- Plano Nacional de Eficiência Energética (PNEF): indica ações e diretrizes específicas direcionadas à promoção da eficiência energética em vários setores da economia, incluindo o setor industrial brasileiro, orientando políticas públicas voltadas para esse fim;
- Programa Nacional de Conservação de Energia (PROCEL): executado pela Eletrobras, no âmbito do Programa de Aplicação de Recursos do PROCEL (PAR/PROCEL), são direcionados recursos para aplicação em projetos que realizem investimentos em estudos, capacitação e programas de eficiência energética neste setor;
- Programa de Eficiência Energética (PEE/ANEEL): conduzido pela Agência Nacional de Energia Elétrica (ANEEL), o PEE destina verba das distribuidoras para aplicação em projetos de eficiência energética, orientado aplicação para os segmentos mais representativos no consumo de eletricidade no Brasil, o que reconhece a relevância da indústria brasileira nesse aspecto.

A base para o gerenciamento energético constitui-se em três principais processos, sendo: analisar os dados de consumo, discriminá-los e a partir dos resultados o consumidor poderá gerenciar o comportamento do consumo de energia (HUA et al., 2021). O monitoramento de cargas é o princípio do gerenciamento, sendo que a partir dos dados coletados é possível identificar e definir quais cargas estão fora do padrão de consumo, podendo ser ocasionados por mau funcionamento, identificação de falhas na rede elétrica, roubo de energia, entre outros.

Deste modo, diversos pesquisadores implementaram metodologias e paradigmas da inteligência computacional para o monitoramento e classificação das cargas, possibilitando o gerenciamento do consumo energético dos setores residenciais e industriais. Entre diversas técnicas utilizadas destacam-se as Máquinas de Vetor de Suporte (*Support Vector Machines*, SVM), as Redes Neurais Artificiais (RNA), floresta aleatória (*random forest*), k vizinhos mais próximos (*k-nearest neighbor*, KNN), métodos de clusterização, entre outros. Estes são utilizados devido a padrões existentes no sistema que facilitam sua implementação.

## 1.1 Objetivo Geral

O objetivo do trabalho é investigar o uso de métodos de inteligência computacional na identificação e classificação de cargas elétricas industriais.

## 1.2 Objetivos Específicos

Entre os objetivos específicos, pode-se destacar:

- Pesquisas de trabalhos relacionados a área de identificação e classificação de cargas elétricas industriais;
- Coleta e organização de um banco de dados de amostras instantâneas de correntes de cargas industriais;
- Implementação, testes e avaliações das técnicas de classificação MLP e SVM e dos métodos de clusterização KM, FCM e GK;
- Implementação e teste das técnicas meta-heurísticas PSO, DE e GWO para definição dos parâmetros básicos dos classificadores MLP e SVM;
- Implementação e teste de métodos de validação XB, SC, CE e DI para definição da quantidade de clusters nos métodos de clusterização.

## 1.3 Motivação

Para a realização da gestão energética eficientemente, é necessário o desenvolvimento de tecnologias capazes de monitorar em tempo real o consumo de energia, para isso, é premente identificar e classificar as cargas elétricas. Assim, os classificadores propostos conseguem realizar o monitoramento em tempo real de cada carga, podendo futuramente realizar a gestão energética da linha, possibilitando as seguintes melhorias em destaque:

- Realização de intervenções para redução de energia, gerando retorno sustentável e econômico para a indústria;
- Detecção de falhas na rede e nas cargas, tornando a manutenção preventiva e a produção mais eficientes com a redução de paradas não programadas.

Outra motivação em destaque refere-se aos investimentos futuros na digitalização do setor elétrico e em Pesquisa, Desenvolvimento e Inovação (PD&I), buscando a eficiência energética para um futuro sustentável.

Por fim, na literatura existem poucos trabalhos que abordam classificadores de cargas industriais em relação aos trabalhos de classificadores de cargas residenciais, ressaltando a importância do trabalho proposto.

#### **1.4 Organização do documento**

O presente trabalho encontra-se dividido em 5 capítulos. No capítulo 1 foram apresentados a introdução, os objetivos e a contextualização do trabalho e os trabalhos relacionados. No capítulo 2 são abordados os conceitos teóricos. Em seguida, no capítulo 3 são apresentados os materiais e métodos necessários para a realização do trabalho. No capítulo 4 expõem-se as análises e resultados e, por fim, no capítulo 5 encontra-se a conclusão.

## 2 REVISÃO BIBLIOGRÁFICA

A abordagem de monitoramento de cargas não intrusivas foi proposto inicialmente por Hart (1992) para determinar o consumo de energia elétrica de aparelhos residenciais individuais, por meio das análises da corrente e tensão total da carga, mensurado entre a carga e a fonte de energia. A partir das análises, são obtidas a potência ativa ( $\Delta P$ ) e a potência reativa ( $\Delta Q$ ) e analisa-se a faixa de valores para determinados aparelhos. Porém, este método apresenta limitações como o uso em aparelhos com variações contínuas, pequenos aparelhos e aparelhos sempre ativos não devem ser escolhidos para a abordagem proposta.

Ao decorrer dos anos novas técnicas e métodos foram propostos visando realizar e controlar o gerenciamento de energia. No trabalho proposto por Cominola et al. (2017), utilizou-se a técnica *Hybrid Signature-based Iterative Disaggregation* (HSID) para o Monitoramento de Carga Não Intrusivo (*Non-intrusive Load Monitoring*, NILM) baseada na combinação dos *Factorial Hidden Markov Models* (FHMM). Este fornece uma aproximação inicial das trajetórias de uso, com a *Iterative Subsequence Dynamic Time Warping* (ISDTW) que processa as trajetórias de uso para corresponder ao padrão de consumo de energia típico de cada aparelho. Como resultado, o algoritmo conseguiu identificar múltiplos aparelhos operacionais.

O trabalho proposto por Nguyen et al. (2017) investigou o processo de monitorar energia chamado *Non-Intrusive Appliance Load Monitoring* (NIALM) baseado em *System on Chip* (SoC) utilizando *Field-Programmable Gate Array* (FPGA) em tempo real. Como resultado, obteve-se uma precisão de 80% na identificação dos principais eletrodomésticos na rede analisada.

Os autores Bouhouras et al. (2017) propuseram um método para a formulação de assinatura de cargas, implementado em tempo real por meio da técnica NILM. Contou-se com os valores do primeiro, terceiro e quinto harmônico para realizar a assinatura das cargas de uma residência de baixa tensão. Os resultados demonstraram que o maior harmônico da corrente facilitou o desempenho na identificação. Para a formulação da assinatura, utilizou-se a combinação do conteúdo do harmônico mais intenso, o ângulo da fase do maior harmônico. Segundo os autores, futuramente a soma dos harmônicos da corrente também serão analisados juntamente para se obter uma formulação mais eficiente.

O trabalho apresentado por Tabatabaei, Dick e Xu (2017) relata uma investigação aprofundada de algoritmos de classificação *multi-label* para identificação dos aparelhos em sinal de energia, e relata que essa classe de algoritmo recebe pouca atenção na literatura, embora seja um ajuste natural ao problema de identificação. Os autores exploraram dois métodos di-

ferentes de classificação *multi-label*, sendo o *framework RANdom k-labELsets* (RAkEL) e o algoritmo derivado do popular *k-Nearest Neighbor* (MLkNN), aplicando-os no conjunto de dados no domínio do tempo e ao domínio wavelet. Ao comparar o método com os existentes na literatura, conclui-se que o método apresentado é competitivo, sendo um classificador plausível para abordagens NILM.

Em (ZHU et al., 2017) foi proposto um novo método de monitoramento de evento aprimorado baseado em somas acumuladas (CUSUM), visando um melhor desempenho de detecção, especialmente em aparelhos domésticos. Além disso, introduz um nível de confiança como critério e a eficiência do algoritmo *bootstrapping*, reduzindo os erros de detecções. Os resultados mostram que a nova abordagem atinge uma maior precisão em comparação ao método tradicional NILM.

O método FHMM também foi utilizado por Raiker et al. (2018) visando identificar cargas elétricas por meio do NILM. Em complemento, foi desenvolvido um aplicativo para Smartphone para visualização dos resultados. Como resultado foi identificado o consumo total de energia, o qual pode ser acompanhado pelo aplicativo desenvolvido.

O método proposto em (GARCIA et al., 2018) apresenta um esquema de identificação de cargas não intrusiva usando o algoritmo *kernel-Adaline* baseado em estratégia de regressão não linear em um conjunto de cargas RLC com comutação caótica. Não foi realizada a comparação do desempenho do método proposto com diferentes estratégias de identificação de cargas devido à falta de ferramentas de avaliações sólidas e por se tratar de uma novidade.

O trabalho proposto por Baets et al. (2018) utilizou o NILM para detecção de aparelhos ativos em uma residência, com o consumo de energia. Para isto, utilizou-se dois métodos de classificação, sendo a floresta aleatória e a rede neural convolucional. Como resultado, obteve-se  $F_{macro}$  de 85,3% e 88% respectivamente. A validação consistiu em analisar a corrente e a tensão de um único aparelho.

Em (BONFIGLI et al., 2018) foi proposto um algoritmo NILM baseado em *Deep Neural Networks* (DNNs) permitindo o detalhamento de consumo para os usuários. Em adição, foi realizada uma comparação exaustiva dos trabalhos de referência no campo dos *Hidden Markov Models* (HMM). Ao aplicar o método em cenários com ruídos e sem, os resultados superaram os métodos comparados.

O autor Lima (2018) propôs um classificador de cargas acionadas individual e simultaneamente. Para isso, utilizou-se um banco de dados com cinco cargas, compondo cinco classes

diferentes. Para extrair parâmetros das assinaturas dos equipamentos elétricos, foi utilizado o método de Estatística de Ordem Superior e foram implementados os classificadores Máquina de Vetor de Suporte, Redes Neurais e Floresta Aleatória. Como resultado, obteve-se de acurácia média 99,8%, 98,85% e 98,95% para o classificador Máquina de Vetor de Suporte, Redes Neurais e Floresta Aleatória respectivamente.

Os autores He et al. (2018) apresentam duas abordagens de processamento de sinal grafo *Graph Signal Processing* (GSP) baseado em NILM. A primeira abordagem refere-se à minimização da variação total do grafo, procurando um sinal de grafo suave, e a segunda aborda um minimizador de variação total do grafo tendo como ponto de partida um refinamento via *simulated annealing*. A abordagem NILM baseada em GSP tem por objetivo discorrer sobre a grande sobrecarga de treinamento, associada à complexidade dos métodos tradicionais baseados em grafo por meio de uma nova abordagem baseada em eventos. Foi apresentado como resultado que o método proposto realiza o treinamento em um curto período e o quão rápido este pode ser implementado. Os métodos propostos podem funcionar com medidores inteligentes convencionais, com requisições a cada 10 segundos.

Os métodos de monitoramento semi intrusivos *Semi-Intrusive Load Monitoring* (SILM) e NILM são amplamente utilizados para reconhecimento de aparelhos em operação. Para se obter uma precisão considerável utilizando o método NILM, um maior custo computacional poderá ser exigido ao se comparar com uma solução SILM. Deste modo, Xu et al. (2018) propuseram um método de construção de rede classificatória para NILM/SILM. O método proposto auxilia na tomada de decisão, selecionando diferentes tipos de classificadores NILM e SILM em conjunto de forma otimizada. Um problema de otimização combinatória é estabelecido e um modelo do tipo árvore foi utilizado. Como resultado, um modelo mais preciso em NILM ou SILM pode ter um custo maior, devido à complexidade, em relação a uma rede de classificadores múltiplos.

O trabalho proposto por Welikala et al. (2019) apresenta uma implementação prática do monitoramento de carga não intrusivo em tempo real *Real-Time Non-Intrusive Load Monitoring* (RT-NILM). A maioria das técnicas NILM são validadas em ambiente em que não há variação na tensão de alimentação. Assim, foi implementado o algoritmo RT-NILM com objetivo de suprir severas flutuações. O método foi validado em um equipamento residencial em que ocorreram as flutuações de tensão e o algoritmo apresentou uma precisão média de 92,5%.

Piscitelli, Brandi e Capozzoli (2019) apresenta um processo de classificação dos clientes de um fornecedor italiano de energia, por meio do método não intrusivo que não utiliza atributos

preditivos de dados em campo. Para isso, analisou-se os dados das contas de energia mensais e informações adicionais de 114 clientes. Os perfis representativos foram agrupados utilizando o algoritmo de clusterização e árvore de decisão. Como resultado, o modelo alcançou uma precisão geral de 80%.

Du et al. (2019) apresentam um algoritmo de clusterização *Affinity Propagation* (AP) para NILM, o qual é melhorado conforme o modelo gráfico de fator e a teoria de propagação. Os resultados experimentais mostram que esse algoritmo melhorado identifica combinações de classe de aparelhos domésticos. Segundo o método, a empresa de energia pode usar os dados do equipamento monitorado para estabelecer um monitoramento e previsão do consumo elétrico domiciliar.

O trabalho apresentado em (CUÑADO; LINSANGAN, 2019) expôs um medidor inteligente de classificação de eletrodomésticos, com um banco de dados contendo cinco classes. Uma abordagem de aprendizado supervisionado usando o classificador *k-Nearest Neighbors* (KNN) foi utilizado na classificação dos aparelhos. Como resultado, obteve-se a acurácia de 95,55% na classificação das classes dos aparelhos.

Em (C; KUMAR; K, 2019) foi proposto um medidor de energia único para monitorar o estado de um conjunto de aparelhos utilizando a técnica NILM. Para isso, utilizou-se o MATLAB/Simulink para simular matematicamente cada aparelho e as técnicas supervisionadas de treinamento, sendo a árvore de decisão e o kNN. Os resultados obtidos foram demonstrados por meio da matriz de confusão e mostra que o algoritmo kNN teve uma melhor acurácia em relação à árvore de decisão.

De acordo com De Baets et al. (2019), existe um método capaz de identificar aparelhos eletrodomésticos automatizadamente que até o momento não haviam sido identificados baseado em NILM. Para isso, os aparelhos são mapeados pela trajetória da tensão e corrente (VI) e mapeados para um espaço de características recém aprendido criado por uma rede neural siamesa, de modo que as amostras do mesmo dispositivo formam clusters. Como resultado, ao utilizar a avaliação comparativa *PLAID* e *WHITED* mostrou-se uma  $F_{macro}$  de 0,90 e 0,85, respectivamente.

Em (GOMES; PEREIRA, 2020) foi proposto a aplicação da função de perda de quantil de pinball para guiar uma rede neural profunda para NILM. Foi apresentado um conjunto de métricas de desempenho complementares para estimativa de energia. A arquitetura proposta promove conceitos como redes neurais de convolução e redes neurais recorrentes. Os resultados

apresentados demonstraram a superioridade de identificação do sistema proposto, mostrando que os desempenhos obtidos usando a função de perda de quantil de pinball são superiores aos obtidos usando o erro quadrático médio.

Na proposição de Fang et al. (2020) há um novo método NILM, o qual aproveita os avanços na aprendizagem estatística, que até o momento não havia sido aplicado neste domínio. Para tanto, são propostos três estágios, sendo:

- Primeiro: uma abordagem baseada em aprendizagem não paramétrico Bayesiano para extração de estados de aparelhos;
- Segundo: a tecnologia de sobre amostragem de minoria sintética para aumento de dados e mitigação do grande desequilíbrio em eventos de comutação;
- Terceiro: as redes de memória de curto prazo para classificação da situação para cada aparelho. Utilizou-se um esquema de geração de amostras diferenciais, o qual reduziu a complexidade das tarefas da rede específica dos aparelhos e a velocidade obtida no treinamento da rede preservaram um desempenho robusto.

Schirmer, Mporas e Paraskevas (2020), no que lhe concerne, apresenta uma arquitetura de desagregação de energia usando algoritmo de correspondência elástica. A arquitetura descrita pelo autor, utiliza banco de dados de assinaturas de consumo de energia de referência, comparando-os com os quadros de consumo de energia de entrada usando a correspondência dos modelos. Para tanto, foi proposto cinco algoritmos de correspondência elástica diferentes. Os resultados experimentais demonstram que os algoritmos podem ser usados para desagregação de energia, com ênfase na *Minimum Variance Matching* (MVM) que obteve 87,58% de acurácia.

Já Corrêa e Castro (2020), propuseram um sistema automatizado baseado em rede neural artificial para realizar algumas tarefas do NILM. Uma rede neural empilhada foi desenvolvida para extrair características do sinal de energia dos aparelhos para identificar quais estão em operação durante um determinado tempo. Essa informação é usada para identificação de aparelhos individuais por meio do sinal agregado total, e o consumo é calculado por integração numérica. Como resultado, o sistema teve uma precisão de 87,7% e uma F-score de 0,879, demonstrando uma eficiência considerável ao ser comparada com outros estudos realizados anteriormente.

Em (JIA et al., 2021) propõe um NILM visando monitoramento do consumo de energia de aparelhos domésticos, principalmente resistivos. Para isso, utilizou-se a reconstrução de imagem de uma trajetória da tensão e corrente (VI) como entrada de uma rede neural convolucional

*Convolutional Neural Network* (CNN) para a classificação dos aparelhos, em particular, aparelhos resistivos. Os resultados obtidos pelo autor, houve um excelente desempenho no método de reconstrução de imagem VI, com uma  $F_{macro}$  de 100% para aparelhos residenciais com forma de ondas similares.

Em (MONTEIRO et al., 2021) propõe o NILM aplicado em tempo real que considera os transientes eletromagnéticos, analisando o desempenho entre a rede neural convolucional *Long Short-Term Memory* (LSTM) e a rede neural *Multilayer Perceptron* (MLP). Nos casos estudados, encontrou-se uma CNN 1D com um desempenho melhor para os casos considerados e um sinal de corrente elétrica adequado para o NILM. Foram utilizados filtros adaptáveis para a extração automática, obtendo o comportamento da carga, os transientes eletromagnéticos, entre outros, o qual ocasionou uma diminuição no custo do pré-processamento dos dados. Os resultados demonstraram que a CNN apresenta um melhor desempenho para usar em casos de NILM.

Völker et al. (2021) propuseram um *framework* para coleção de dados para NILM. Esse é composto de hardware e software necessários para rotular e registrar semi automaticamente os dados. O hardware é composto por um medidor inteligente com múltiplos soquetes, o qual registra os dados de consumo. A rotulagem é realizada na etapa de pós-processamento semi automática, reduzindo o esforço em 72% em comparação com uma abordagem manual. O conjunto de dados apresenta ininterrupto tempo de sincronização e agregação individuais de tensão e corrente com rótulos de transição de estados distintos por um total de 101 dias.

O trabalho proposto por Azizi, Beheshti e Bolouki (2021) aborda um novo algoritmo de classificação baseado em NILM que atenua problemas como: ruídos e *spikes* do sinal de potência, *overshoots* nos tempos de transição, valores próximos de consumo por diferentes aparelhos e indisponibilidade de um conjunto de dados considerável para treinamento. Neste caso, o algoritmo proposto filtra os sinais de energia, extrai características específicas dos aparelhos e rotula com alta precisão cada evento detectado. Os resultados aqui obtidos demonstram uma alta precisão do algoritmo proposto, o qual foi validado usando a avaliação do *Residential Energy Disaggregation Dataset* (REDD).

Salem, Sayed-Mouchaweh e Tagina (2021) propuseram um NILM utilizando *infinite Factorial Hidden Markov Model Constrained on Contextual features* (iFHMMCC). Esse modelo foi utilizado para aliviar a complexidade computacional e tornar o modelo mais tratável do que algoritmos baseados em *Bayesian NonParametric* (BNP). Como resultado, obteve-se uma melhoria na precisão de identificação, como também na complexidade computacional.

Em (HUA et al., 2021) é proposto um NILM cujo objetivo é identificar múltiplos aparelhos quando comutados simultaneamente. Esse é um modelo *Mixed Linear Integer Programming* (MILP) em que 0-1 indicam se os aparelhos foram comutados no processo do evento. Utilizou-se o algoritmo genérico para a seleção dos parâmetros de somas acumuladas (CUSUM). Como resultado, o algoritmo obteve uma  $F1_{score}$  de 0,92. Na detecção de eventos, o algoritmo genético de classificação não dominado II é usado para selecionar os parâmetros (CUSUM) como limite e o comprimento das janelas deslizantes. Como resultados, o CUSUM proposto tem uma precisão de 92% na detecção de eventos.

A Tabela 2.1 apresenta um resumo dos autores citados e suas variáveis estudadas.

Tabela 2.1 – Trabalhos referentes à gestão de energia.

Referência	Ano	Variáveis estudadas
(HART, 1992)	1992	Monitoramento de cargas não intrusivo
(COMINOLA et al., 2017)	2017	Desagregação iterativa baseada em assinatura híbrida para algoritmo NILM
(NGUYEN et al., 2017)	2017	Monitoramento não intrusivo de carga de aparelho NIALM baseado em System on Chip
(BOUHOURLAS et al., 2017)	2017	Método para a formulação de assinatura de cargas eficientes
(TABATABAEI; DICK; XU, 2017)	2017	Algoritmos de classificação <i>multi-label</i> para identificação de aparelhos
(ZHU et al., 2017)	2017	Método de monitoramento de eventos aprimorados baseado em CUSUM
(RAIKER et al., 2018)	2018	Aplicação de Modelos Fatoriais de Markov Ocultos em NILM
(GARCIA et al., 2018)	2018	Desagregação de cargas não intrusivo usando o algoritmo kernel-Adaline
(BAETS et al., 2018)	2018	NILM para detecção de aparelhos ativos em uma residência
(BONFIGLI et al., 2018)	2018	Algoritmo NILM baseado em redes neurais profundas
(LIMA, 2018)	2018	Classificador de cargas acionadas individual e simultaneamente
(HE et al., 2018)	2018	GSP baseado em NILM
(XU et al., 2018)	2018	Rede classificatória para NILM/SILM
(WELIKALA et al., 2019)	2019	Implementação prática de um monitoramento de carga não intrusivo em tempo real
(PISCITELLI; BRANDI; CAPOZZOLI, 2019)	2019	Processo de classificação de clientes não intrusivo
(DU et al., 2019)	2019	Algoritmo clustering AP para NILM
(CUÑADO; LINSANGAN, 2019)	2019	Medidor inteligente de classificação de eletrodomésticos
(C; KUMAR; K, 2019)	2019	Medidor de energia único para monitorar o estado de um conjunto de aparelhos
(De Baets et al., 2019)	2019	Método capaz de identificar aparelhos eletrodomésticos automatizadamente
(GOMES; PEREIRA, 2020)	2020	Aplicação da função de perda de pinball para quantis para guiar uma rede neural profunda para NILM
(FANG et al., 2020)	2020	Novo método NILM baseada em aprendizagem não paramétrico Bayesiano
(REDDY; GARG; PUDI, 2020)	2020	NILM híbrido, utilizando o estado estacionário e o transitório para identificação das cargas
(SCHIRMER; MPORAS; PARASKEVAS, 2020)	2020	Arquitetura de desagregação de energia usando algoritmo de correspondência elástica
(CORRÊA; CASTRO, 2020)	2020	sistema automatizado baseado em rede neural artificial para realizar algumas tarefas do NILM
(JIA et al., 2021)	2021	Monitoramento do consumo de energia de aparelhos domésticos, principalmente resistivos
(MONTEIRO et al., 2021)	2021	NILM aplicado em tempo real que considera os transientes eletromagnéticos
(VöLKER et al., 2021)	2021	Framework para coleção de dados para NILM
(AZIZI; BEHESHTI; BOLOUKI, 2021)	2021	Algoritmo de classificação baseado em NILM
(SALEM; SAYED-MOUCHAWEH; TAGINA, 2021)	2021	NILM utilizando Modelo Fatorial Oculto de Markov infinito restringido em Recursos Contextuais
(HUA et al., 2021)	2021	Identificação de múltiplos aparelhos quando comutados simultaneamente utilizando NILM

Fonte: Do autor (2021).

### 3 REFERENCIAL TEÓRICO

O processo de classificação de cargas elétricas, em geral, consiste em diversos passos. Deste modo, alguns conceitos importantes da literatura são apresentados a fim de proporcionar um melhor entendimento do trabalho proposto.

#### 3.1 Monitoramento de cargas intrusivo e não intrusivo

Existem duas abordagens para o monitoramento de cargas, o método intrusivo, também conhecido como Monitoramento de Cargas Intrusivo (*Intrusive Load Monitoring*, ILM) e o método não intrusivo, também conhecido como Monitoramento de Cargas Não Intrusivas (*Non-intrusive Load Monitoring*, NILM). O ILM é baseado em sensores distribuídos, enquanto o NILM utiliza apenas um ponto de sensoriamento (ABUBAKAR et al., 2017). Segundo (RIDI; GISLER; HENNEBERT, 2014), os métodos podem ser definidos como:

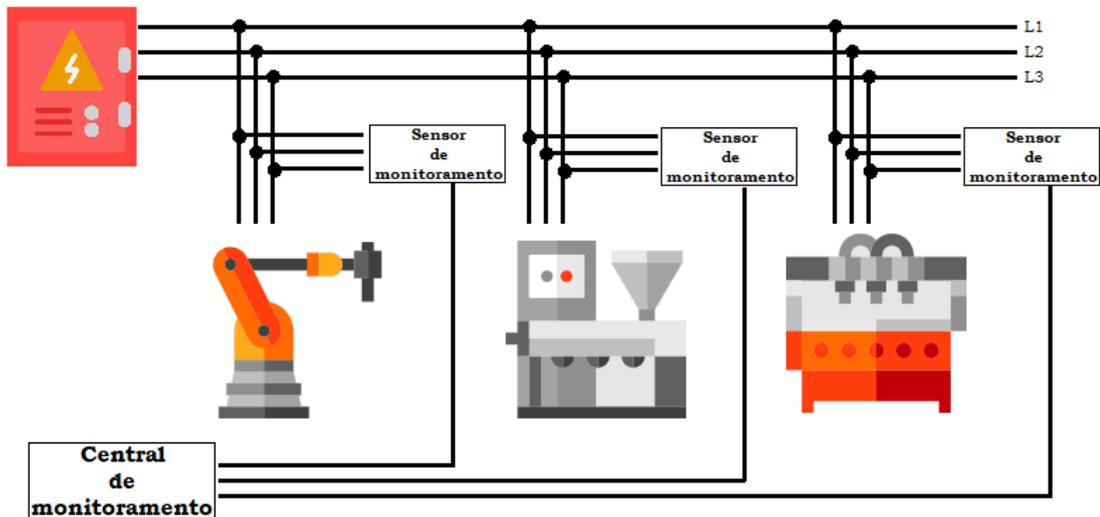
- **Monitoramento de Cargas Intrusivo:** consiste em realizar o monitoramento do consumo de uma ou de um grupo de cargas usando um aparelho de medição de baixo custo. Este monitoramento é dominado intrusivo, pois a medição é realizada no interior do local da carga em que se deseja monitorar, geralmente perto da carga, como ilustra a Figura 3.1;
- **Monitoramento de Cargas Não Intrusivo:** consiste em realizar o monitoramento do consumo utilizando um medidor inteligente que, geralmente, é instalado no painel de distribuição de energia. O termo não intrusivo significa não haver equipamentos extras instalados no local que se deseja monitorar, apenas o medidor inteligente. No NILM as assinaturas das cargas são sobrepostas, sendo necessário a separação para se obter a assinatura de cada carga monitorada. Esta operação é denominada desagregação total do consumo de energia. O sistema é ilustrado na Figura 3.2.

Ambos os métodos possuem vantagens e desvantagens ao serem relacionados. Comparando as vantagens do método NILM em relação ao ILM, pode-se destacar a utilização de apenas um sensor, tornando a instalação e a aquisição dos dados simples. O método ILM possui medidores de baixo custo, porém à medida que se aumenta as cargas monitoradas, aumenta-se o custo linearmente.

Em relação às vantagens do método ILM em relação ao NILM pode-se destacar que as informações adquiridas do sensor são melhores, pelo fato de apresentar a assinatura apenas da

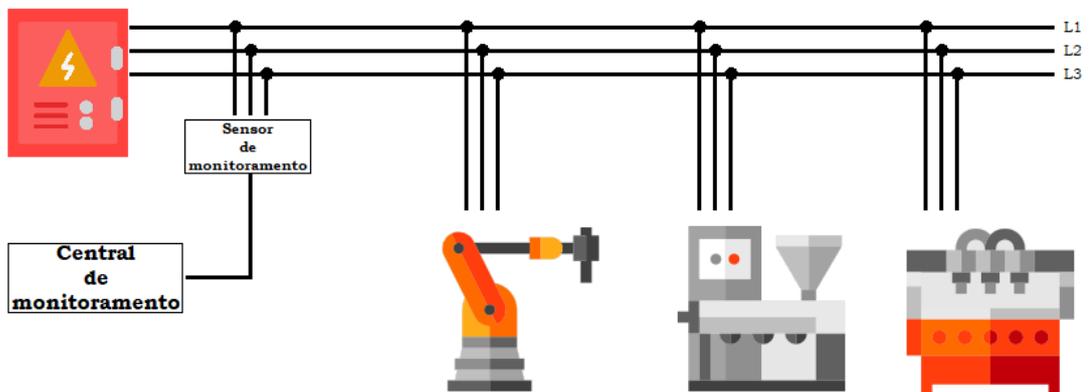
carga em que o sensor está instalado. Geralmente, as cargas que demandam baixa potência, ou cargas em posição de espera são difíceis de serem identificadas com NILM (RIDI; GISLER; HENNEBERT, 2014).

Figura 3.1 – Sistema de monitoramento de cargas intrusivo.



Fonte: Do autor (2023).

Figura 3.2 – Sistema de monitoramento de cargas não intrusivo.



Fonte: Do autor (2023).

### 3.2 Algoritmos de pré-processamento de dados

O processamento de dados diz respeito ao processo de extração de variáveis e informações de medições. Esta é uma das partes mais importantes de um sistema de reconhecimento de padrões, que por vez tem por objetivo caracterizar e distinguir diferentes classes.

Várias técnicas de processamento de sinal são utilizadas para extração de variáveis, as quais são destacadas por (KHOKHAR et al., 2015), sendo a Transformada de Fourier (*Fourier*

*Transform*, FT), Transformada Wavelet (*Wavelet Transform*, WT), Filtro de Kalman (*Kalman Filter*, KF), Transformada de Hilbert (*Hilbert Transform*, HT), Filtro Adaptativo (*Adaptive Filter*, AF), entre outros.

Geralmente, as variáveis extraídas do processamento de sinal são usadas como entradas no sistema de classificação, ao invés do uso da forma de onda do sinal, no caso da grandeza elétrica corrente. Isto ocorre pelo fato da contaminação de um sinal por outros indesejados ou por ruído ser um problema muito comum, inviabilizando sua aplicação direta no classificador. Deste modo, é necessário um processamento de baixa complexidade e confiável, pois a seleção de um conjunto de variáveis adequadas é um passo importante para a robustez do classificador.

Em técnicas de processamento é desejável que as variáveis selecionadas possam caracterizar e distinguir classes e, além disso, que elas não sejam correlacionadas e existam em pequeno número. Outros problemas que podem ser relevantes são aqueles relacionados à estabilidade numérica, insensibilidade ao ruído, e a interpretação física do problema (MORAVEJ; PAZOKI; ABDOOS, 2011).

### 3.2.1 Normalização dos dados

A normalização por variância é uma técnica de pré-processamento de dados amplamente utilizada em aplicações de aprendizado de máquina e análise de dados para remover as diferenças de escala entre as variáveis de um conjunto de dados. Segundo Müller e Klinkenberg (2017), essa técnica consiste em calcular a variância de cada variável e depois dividir todas as observações da variável pelo desvio padrão da mesma. Dessa forma, as variáveis são redimensionadas de modo que todas tenham a mesma variância. Na Equação 3.1 é apresentada a normalização por variância.

$$x_{norm} = \frac{x - \mu}{\sigma} \quad (3.1)$$

onde  $x$  é o valor original da variável,  $\mu$  é a média da variável e  $\sigma$  é o desvio padrão da variável. A variável  $x_{norm}$  é o valor normalizado da variável  $x$ .

De acordo com Chollet (2018) e Ghosh e Chakraborty (2019), a normalização por variância é uma das técnicas de pré-processamento de dados mais importantes para o treinamento de redes neurais artificiais. Essa técnica ajuda a reduzir a influência de variáveis com grandes diferenças de escala e a melhorar a convergência dos algoritmos de otimização utilizados no

treinamento das redes. Além disso, a normalização por variância pode ajudar a aumentar a precisão e a estabilidade dos modelos de aprendizado de máquina.

### 3.3 Seleção de Características

Ao projetar um classificador, vários fatores devem ser analisados, os quais se destacam a complexidade computacional e a capacidade de generalização. Assim, realizar a seleção de características é uma parte importante para se otimizar o desempenho do classificador e a capacidade de generalização, o qual tem por objetivo reduzir o vetor de características para um mínimo suficiente.

Por este fato, duas características podem carregar informações importantes quando tratadas separadamente, entretanto, ao serem combinadas em um vetor, haverá um pequeno ganho devido à alta correlação mútua, porém, aumenta-se a complexidade sem ganho considerável. Deste modo, a seleção de característica tem por objetivo selecionar as características mais importantes, reduzindo o seu número e, em simultâneo, manter o máximo de informação discriminatória da classe (THEODORIDIS; KOUTROUMBAS, 2008).

#### 3.3.1 Razão Discriminante de Fisher

A Razão Discriminante de Fisher (*Fisher's Discriminant Ratio*, FDR) é um dos métodos eficientes para realizar a seleção de características, reduzindo a dimensão do vetor de entrada  $\mathbf{x}$ . Supondo dois pontos de amostra em um espaço de dados de dimensão  $d$ , deseja-se encontrar uma linha no espaço original tal que os dados projetados na linha podem ser separados por algum ponto na linha. Desta forma, quanto maior o quadrado da diferença entre as médias de dois pontos de amostra projetados fora, e quanto menor a dispersão na classe, melhor será a linha esperada (WANG et al., 2011).

Assim, a projeção de Fisher visa maximizar a dispersão entre os valores médios de duas classes, enquanto minimiza a variação de cada classe (SHARMA; YADAV; SINGH, 2016). A função de custo do critério da FDR, usada como ferramenta de seleção de parâmetros para múltiplas classes, é dada pela Equação 3.2.

$$FDR = \sum_i^M \sum_{j \neq i}^M \frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2} \quad (3.2)$$

onde  $\mu$  e  $\sigma$  representam a média e a variância e os subscritos  $i$  e  $j$  referem-se a média e variância correspondentes às características sob investigação para as classes  $\omega_i$  e  $\omega_j$  respectivamente (THEODORIDIS; KOUTROUMBAS, 2008).

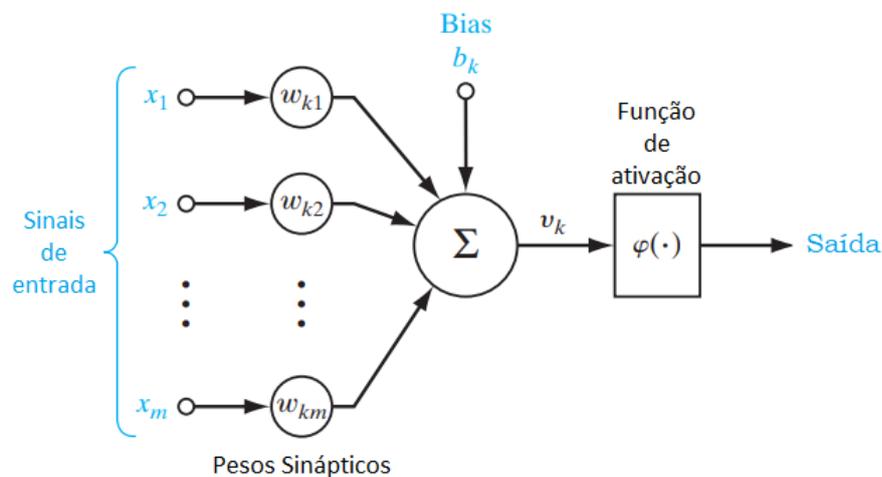
### 3.4 Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA) foram desenvolvidas como generalizações de modelos matemáticos das redes neurais biológicas. As RNAs são capazes de generalizar informações após o treinamento, o que as tornou amplamente utilizadas e destacadas na comunidade acadêmica.

Uma primeira onda de interesse em redes neurais, também conhecida como modelos conexionistas, surgiu após a introdução do primeiro neurônio artificial proposto por McCulloch e Pitts (1943). No entanto, Rosenblatt (1957) propôs um modelo de neurônio artificial, amplamente utilizado até hoje, denominado *Perceptron*.

O neurônio apresentado na Figura 3.3 é a unidade básica de uma rede neural. Os sinais de entrada  $x_1, x_2, \dots, x_l$  são aplicados nos nós de entrada da rede. Em seguida, cada um é multiplicado pelos pesos correspondentes  $w_{k1}, w_{k2}, \dots, w_{km}$ , conhecidos como pesos sinápticos ou simplesmente sinapses. O bias é dado por  $b_k$ . Os produtos são somados, produzindo um sinal de ativação  $v_k$ . Este sinal entra em um dispositivo não linear, chamado função de ativação  $\varphi(\cdot)$ , que fornece o sinal de saída (axônio).

Figura 3.3 – Neurônio do tipo Perceptron.



Fonte: Adaptado de (HAYKIN, 2009)

A função que representa este neurônio é dada na Equação 3.6.

$$y_k = \varphi\left(\sum_{i=1}^m \omega_{ki}x_i + b_k\right) \quad (3.3)$$

A função de ativação ( $\varphi(\cdot)$ ) define a saída do neurônio em termos de  $v$ . A seguir são demonstrados três tipos básicos de funções de ativação (HAYKIN, 2009).

- Função de Limiar ou função de Heaviside, sendo:

$$\varphi(v) = \begin{cases} 1, & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases} \quad (3.4)$$

- Função Sigmoidal, sendo:

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (3.5)$$

onde  $a$  é o parâmetro de inclinação da função.

- Função Tangente Hiperbólica, sendo:

$$\varphi(v) = \tanh(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} \quad (3.6)$$

- Função ReLU, sendo:

$$\varphi(v) = \max(0, v) \quad (3.7)$$

As redes de uma única camada de pesos adaptativos apresentam uma série de limitações importantes em termos da gama de funções que podem representar. Para permitir mapeamentos gerais, é necessário considerar transformações sucessivas da rede. Deste modo, é apresentado a rede do tipo Perceptron de Multicamadas (*Multilayer Perceptron*, MLP) que permite mapeamentos gerais.

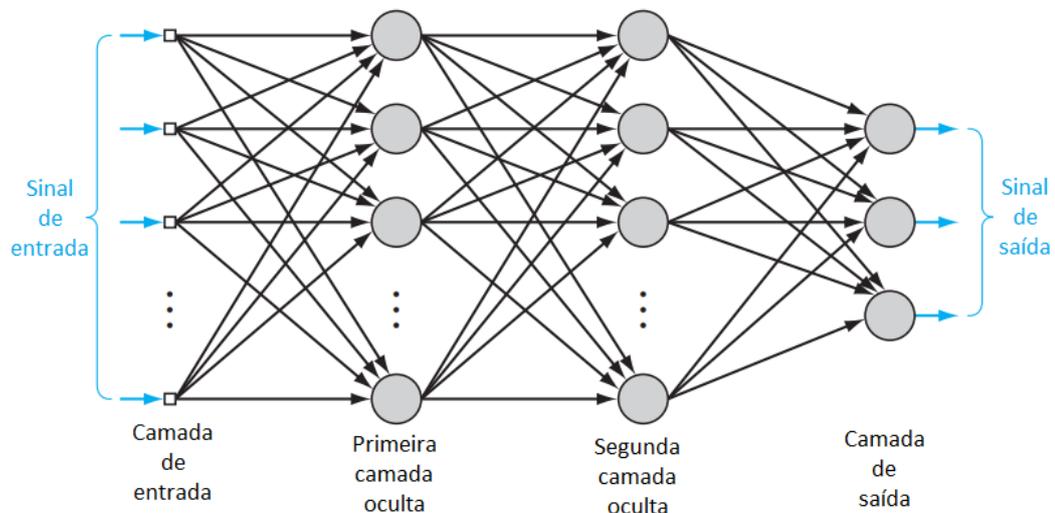
### 3.4.1 Perceptron de Multicamadas

A RNA do tipo MLP é uma poderosa rede de múltiplas camadas *feedforward*, ou seja, processa os dados em uma direção, sendo do nó de entrada para o nó de saída. Essas redes oferecem uma estrutura para a representação de funções não lineares entre um conjunto de entrada e um conjunto de saída. Os três pontos que destacam as redes RNA do tipo MLP são (HAYKIN, 2009):

- O modelo de cada neurônio da rede inclui uma função de ativação não linear que é diferenciável;
- A rede contém uma ou mais camadas ocultas do nó de entrada e saída;
- A rede exibe um alto grau de conectividade, cuja extensão é determinada pelos pesos da rede.

A Figura 3.4 ilustra uma rede com múltiplas camadas em que há uma camada de entrada, duas camadas ocultas e uma camada de saída. Na Figura 3.5 encontra-se os dois tipos de fluxos da rede MLP. O sinal da função é um sinal de entrada que chega na extremidade da rede e se propaga para frente, neurônio por neurônio, chegando na extremidade de saída. O sinal de erro se origina na saída do neurônio da rede e propaga para trás, camada por camada.

Figura 3.4 – Rede com múltiplas camadas.

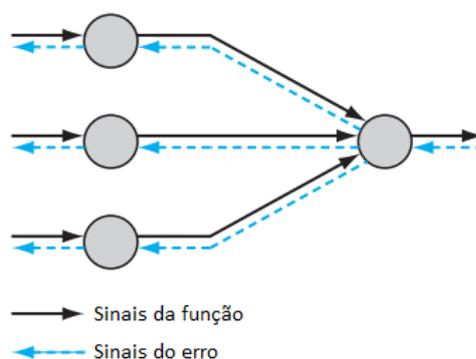


Fonte: Adaptado de (HAYKIN, 2009).

### 3.4.2 Processo de aprendizagem

Existem diferentes maneiras de o ser humano aprender no meio que o cerca, assim ocorre com as redes neurais. De acordo com (HAYKIN, 2009) em um sentido amplo pode-se caracterizar os processos de aprendizagem em que a rede aprende com um professor e sem professor.

Figura 3.5 – Direção de dois fluxos básicos de sinais no MLP.



Fonte: Adaptado de (HAYKIN, 2009).

O processo de aprendizagem sem professor pode ser categorizado em processo de aprendizagem não supervisionada que é paralelo ao aprendizado do ser humano (HAYKIN, 2009).

### 3.4.2.1 Aprendizagem supervisionada

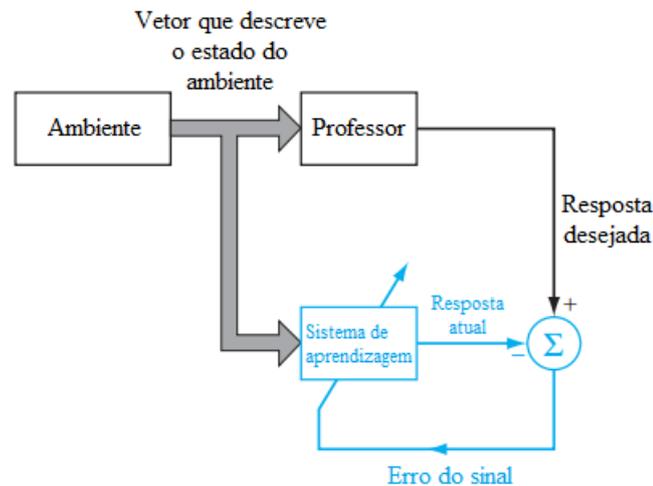
A aprendizagem com professor é referenciada como aprendizagem supervisionada. A Figura 3.6 mostra o bloco que ilustra esta aprendizagem. O professor tem conhecimento do ambiente, representado por um conjunto de entradas e saídas. Porém, o ambiente é desconhecido para a rede neural.

Suponha que o professor e a rede neural estejam ambos expostos a um vetor de treinamento extraído de um mesmo ambiente. Pelo fato de o professor ter conhecimento prévio, o professor consegue fornecer à rede uma resposta desejada.

Deste modo, com a resposta atual da rede e a resposta desejada fornecida pelo professor é possível obter o erro. Assim, os parâmetros da rede são ajustados sob a influência combinada do vetor de treinamento e do erro (HAYKIN, 2009).

Desta maneira, o conhecimento do professor é transferido para a rede por meio do treinamento. Quando os pesos sinápticos não variam, pode-se dispensar o professor e a rede poderá lidar com o ambiente por si própria.

Figura 3.6 – Diagrama de bloco da aprendizagem supervisionada.



Fonte: Adaptado de (HAYKIN, 2009).

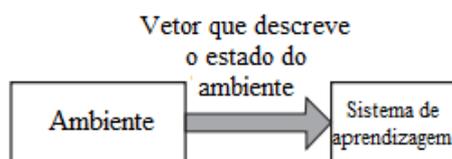
### 3.4.2.2 Aprendizagem não supervisionada

No processo de aprendizagem não supervisionada não há o professor para supervisionar o processo de aprendizagem. A Figura 3.7 mostra o bloco que ilustra esta aprendizagem.

Pelo fato de não existir um professor, a provisão é realizada por meio da análise do comportamento estatístico dos dados em que os parâmetros são otimizados conforme a análise.

Uma vez que o sistema esteja sintonizado para as regularidades estatísticas dos dados de entrada, este desenvolve a capacidade de formar representações internas para codificar características da entrada (HAYKIN, 2009).

Figura 3.7 – Diagrama de bloco da aprendizagem não supervisionada.



Fonte: Adaptado de (HAYKIN, 2009).

Pelo fato das RNAs serem caracterizadas pelas funções de ativação não lineares e diferenciáveis, há um algoritmo de aprendizagem supervisionada popular e computacionalmente eficiente, chamado *backpropagation*, o qual será detalhado a seguir (BISHOP, 1995).

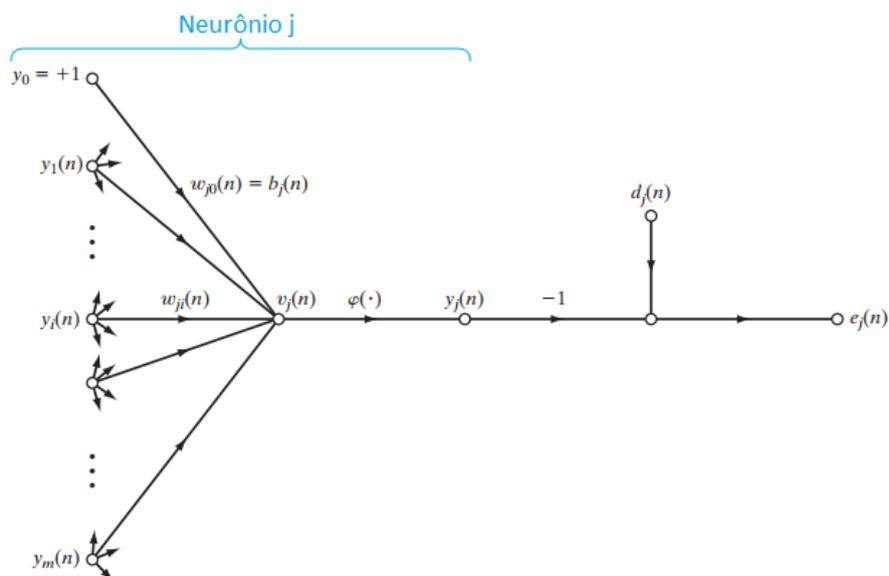
### 3.4.3 Algoritmo de aprendizagem – *backpropagation*

A maioria dos algoritmos de treinamento abrange um procedimento iterativo para minimização do erro, com ajustes nos pesos feito em uma sequência de etapas. Estas podem ser especificadas em duas, sendo (BISHOP, 1995):

- Primeira etapa: são avaliadas as derivadas da função erro em relação aos pesos. Nesta etapa, os erros são propagados para trás na rede, sendo definido como retropropagação do erro ou *error backpropagation* especificamente para descrever a avaliação das derivadas.
- Segunda etapa: as derivadas são então usadas para calcular os ajustes a serem feitos nos pesos.

O algoritmo *backpropagation* fornece uma solução computacional eficiente para avaliar as derivadas da função erro. Para descrevê-la, considere Figura 3.8, que mostra o neurônio  $j$  sendo alimentado por um conjunto de sinais de funções de entrada.

Figura 3.8 – Fluxo de sinal destacando os detalhes da saída do neurônio  $j$ .



Fonte: Adaptado de (HAYKIN, 2009).

O sinal de ativação  $v_j(n)$  produzido na entrada da função de ativação do neurônio  $j$  é demonstrado na Equação 3.8.

$$v_j(n) = \sum_{i=0}^m \omega_{ji}(n) y_i(n) \quad (3.8)$$

onde  $m$  é o número total de entradas (excluindo o bias), os pesos sinápticos  $\omega_{j0}$  corresponde ao bias  $b_j$  aplicado no neurônio  $j$  (HAYKIN, 2009). A função de ativação de  $y_j(n)$  que está na saída no neurônio  $j$  na interação  $n$  é apresentada na Equação 3.9.

$$y_j(n) = \varphi_j(v_j(n)) \quad (3.9)$$

O algoritmo aplica uma correção na regra delta  $\Delta\omega_{ji}(n)$  no peso sináptico  $\omega_{ji}(n)$ , sendo proporcional a derivada parcial  $\frac{\partial \xi}{\partial \omega_{ji}}(n)$ . Conforme a regra da cadeia, expressa-se o gradiente conforme a Equação 3.10 (HAYKIN, 2009).

$$\frac{\partial \xi(n)}{\partial \omega_{ji}(n)} = \frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial \omega_{ji}(n)} \quad (3.10)$$

onde  $\xi(n)$  é a energia do erro instantânea, apresentada na Equação 3.11 (HAYKIN, 2009).

$$\xi_j(n) = \frac{1}{2} e_j^2(n) \quad (3.11)$$

Derivando ambos os lados na Equação 3.11, tem-se

$$\frac{\partial \xi(n)}{\partial e_j(n)} = e_j(n) \quad (3.12)$$

Como o erro do sinal é demonstrado na Equação 3.13, em que  $d_j(n)$  é o  $j$ -ésimo elemento do vetor desejado e  $y_j$  é o  $j$ -ésimo elemento do vetor de saída, ao derivar ambos os lados em relação a  $y_j$  o resultado é apresentado na Equação 3.14.

$$e_j(n) = d_j(n) - y_j(n) \quad (3.13)$$

$$\frac{\partial e_j(n)}{\partial y_j(n)} = -1 \quad (3.14)$$

A derivada da Equação 3.9 em relação a  $v_j$  é apresentada na Equação 3.15.

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'(v_j(n)) \quad (3.15)$$

onde o apóstrofo significa a derivada da função. Por fim, diferenciando a Equação 3.8 em relação a  $\omega_{ji}(n)$ , tem-se

$$\frac{\partial v_j(n)}{\partial \omega_{ji}(n)} = y_i(n) \quad (3.16)$$

Alterando as equações 3.12, 3.14, 3.15, 3.16 em 3.10 é apresentada na Equação 3.17.

$$\frac{\partial \xi(n)}{\partial \omega_{ji}(n)} = -e_j(n) \phi'(v_j(n)) y_i(n) \quad (3.17)$$

Deste modo, a correção  $\Delta \omega_{ji}(n)$  aplicado no peso sináptico  $\omega_{ji}(n)$  é definido pela regra da cadeia, conforme a Equação 3.18.

$$\Delta \omega_{ji}(n) = -\eta \frac{\partial \xi(n)}{\partial \omega_{ji}(n)} \quad (3.18)$$

onde  $\eta$  é o parâmetro taxa de aprendizado do algoritmo *backpropagation*.

Quando uma rede neural é empregada, certas configurações devem ser consideradas e devidamente configuradas. Estas dependem diretamente da complexidade do problema, tamanho do conjunto de dados, número de neurônios, meta de erro e aplicação (regressão ou classificação) (ABRAHAM, 2005).

### 3.5 Máquinas de Vetores de Suporte

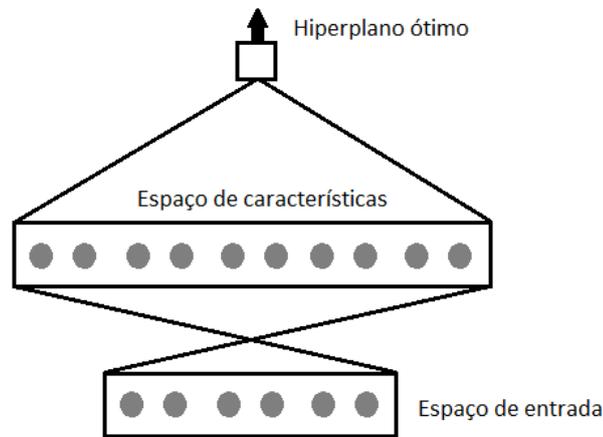
O algoritmo Máquinas de Vetores de Suporte (*Support Vector Machines*, SVM) é amplamente utilizado em diversas áreas da ciência. O algoritmo foi proposto por (VAPNIK, 1995) que se baseia na teoria de aprendizagem estatística, apresentando um excelente desempenho de generalização (YAN; JIA, 2018).

Segundo (VAPNIK, 1995), a SVM segue a ideia de mapear os vetores de entrada  $\mathbf{x}$  em um espaço de características de alta dimensão  $\mathbf{Z}$  por meio de um mapeamento não linear, escolhido inicialmente, como demonstrado na Figura 3.9.

A SVM baseia-se na criação de um função que melhor separa as entradas em classes, buscando alocar limites na distância máxima possível de uma linha de referência, plano ou hiperplano, dependendo da dimensão das variáveis de entrada. Quanto maior a distância, menor será o erro de generalização do classificador (JARAMILLO et al., 2020).

Dado um vetor de treino  $x_i \in R^n$ ,  $i = 1, \dots, l$ , onde  $i$  representa o número de entradas, e um vetor de saída  $\mathbf{y} \in R^l$ , sendo que para duas classes,  $y_i \in -1, 1$  (CHANG; LIN, 2011), o algoritmo resolve o seguinte problema de otimização da Equação (3.19).

Figura 3.9 – Mapa do SVM dos espaços de entradas em um hiperplano dimensional.



Fonte: Adaptado de (VAPNIK, 1995).

$$\begin{aligned}
 \min_{\omega, b, \xi} \quad & \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \xi_i \\
 \text{subject to} \quad & y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i \\
 & \xi_i \geq 0, i = 1, \dots, l
 \end{aligned} \tag{3.19}$$

onde  $\phi(\mathbf{x}_i)$  mapeia  $\mathbf{x}_i$  em um hiperplano e  $C > 0$  é o parâmetro de otimização. Devido à possibilidade de haver um problema de grandes dimensionalidades da variável vetorial  $\mathbf{w}$ , geralmente resolve-se o problema da Equação (3.20).

$$\begin{aligned}
 \min_{\alpha} \quad & \frac{1}{2} \alpha^T Q \alpha - \mathbf{e}^T \alpha \\
 \text{subject to} \quad & \mathbf{y}^T \alpha = 0 \\
 & 0 \leq \alpha_i \leq C, i = 1, \dots, l
 \end{aligned} \tag{3.20}$$

onde  $\mathbf{e} = [1, \dots, 1]^T$  é um vetor de um em todas as posições,  $Q$  é uma matriz semi definida positiva  $l$  por  $l$ ,  $Q_{ij} \equiv y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$ , e  $K(\mathbf{x}_i, \mathbf{x}_j) \equiv \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$  é a função *kernel*, que pode ser linear, polinomial, gaussiana, sigmoidal e função de base radial.

Após o problema da Equação (3.20) ser resolvido, usando a relação primal dual, o ótimo  $\mathbf{w}$  satisfaz a Equação (3.21).

$$\mathbf{w} = \sum_{i=1}^l y_i \alpha_i \phi(\mathbf{x}_i) \quad (3.21)$$

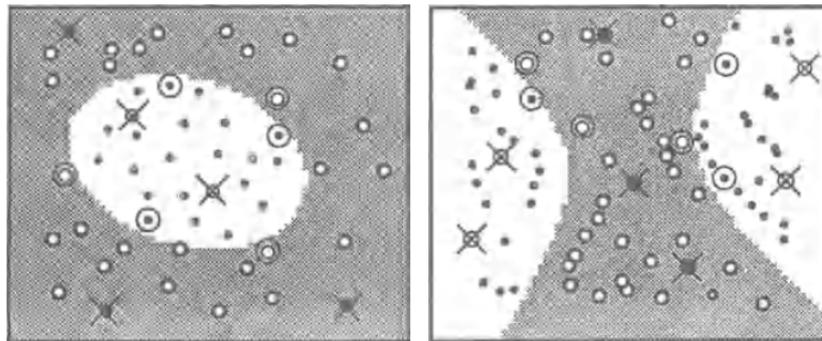
A função de decisão encontra-se na Equação (3.22).

$$\text{sgn}(\mathbf{w}^T \phi(\mathbf{x}) + b) = \text{sgn}\left(\sum_{i=1}^l y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\right) \quad (3.22)$$

Algumas informações são armazenados no modelo de previsão, os quais podem ser citados o  $y_i$  e  $\alpha_i \forall_i$ , o  $b$ , o mapa das saídas, os vetores de suporte, e outras informações como os parâmetros da *kernel* (CHANG; LIN, 2011).

Na Figura 3.10 encontram-se duas classes, representadas por bolas pretas e brancas. A fronteira de decisão foi construída usando-se um produto interno do tipo polinomial com  $d = 2$ . Observa-se que no exemplo não foi possível separar sem erros. Estes são representados por  $X$  e o vetor de suporte é representado pelo duplo círculos (VAPNIK, 1995).

Figura 3.10 – Exemplo de separação de duas classes utilizando o produto interno do tipo polinomial.



Fonte: (VAPNIK, 1995).

### 3.6 Sistemas Fuzzy

A lógica fuzzy, proposta por Lotfi Zadeh no início dos anos 1960, representa uma abordagem para lidar com informações vagas, ambíguas e imprecisas, permitindo o processamento e aprendizado a partir de dados. Diferentemente da lógica booleana tradicional, que opera apenas com valores 0 e 1, a lógica fuzzy utiliza valores reais entre 0 e 1, indicando o grau de relevância ou associação (*Membership*) que cada elemento possui em um determinado conjunto. Isso permite uma representação mais fiel e flexível das incertezas presentes em muitos problemas reais.

De acordo com Zadeh (1965):

Seja  $X$  um espaço matemático, com um elemento genérico de  $X$  denotado por  $x$ . Assim,  $X = \{x\}$ . Um conjunto fuzzy (classe)  $A$  em  $X$  é caracterizado por uma função de pertinência  $f_A(x)$  que associa a cada ponto em  $X$  um número real no intervalo  $[0, 1]$ .

A lógica fuzzy, ao contrário dos sistemas clássicos, é um conjunto ordenado de instruções que após a sua aplicação fornecerá uma solução aproximada para um dado problema. Esta, permite representar nos algoritmos a forma como funciona o raciocínio humano, o qual desempenha um papel essencial na capacidade de tomar decisões racionais em um ambiente de incerteza e imprecisão.

Essa capacidade depende, no que lhe concerne, da capacidade de inferir uma resposta aproximada para uma pergunta baseada em um estoque de conhecimento que é inexato, incompleto ou não totalmente confiável (ZADEH, 1988).

Existem duas principais razões que os sistemas clássicos não conseguem lidar com problemas incertos e imprecisos, sendo (ZADEH, 1988):

- Não fornece um sistema para representar o significado das proposições expressas em linguagem natural quando estes são imprecisos;
- Nos casos em que o significado das proposições podem ser representadas simbolicamente em uma linguagem de representação, por exemplo, uma rede semântica ou um gráfico de dependência conceitual, e não há mecanismo de inferência.

Uma característica relevante da lógica fuzzy é o fato de conter como casos especiais não apenas os sistemas lógicos bivalorados e multivalorados clássicos, mas também a teoria da probabilidade e probabilística lógica.

Em um conjunto fuzzy as fronteiras não são nitidamente definidas, possibilitando que determinados dados possam pertencer, com um determinado grau de pertinência, a conjuntos diferentes. Deste modo, na lógica fuzzy tudo, inclusive a verdade, é uma questão de grau de pertinência.

Um silogismo fuzzy básico na lógica fuzzy relevante para as regras de combinação em sistemas especialistas é o silogismo interseccional/produto, sendo um silogismo que serve como uma regra de inferência para quantificar proposições. Este silogismo pode ser expresso como a regra de inferência demonstrada na Equação 3.23 (ZADEH, 1988).

$$\begin{aligned}
Q_1 \quad & A's \text{ são } B's \\
Q_2 \quad & (A \text{ e } B)'s \text{ são } C's \\
(Q_1 \otimes Q_2) \quad & A's \text{ são } (B \text{ e } C)'s
\end{aligned} \tag{3.23}$$

### 3.7 Clusterização

Clusterização é uma abordagem de análise de dados que auxilia na resolução de problemas que necessitam de classificação. O objetivo é subdividir um conjunto de dados em subconjuntos (*clusters*). Estes subconjuntos possuem alto grau de associação entre seus elementos e fraca associação ou fraco grau de associação referente aos elementos dos demais subconjuntos.

Geralmente, os conjuntos são subdivididos por especialistas humanos que classificam dados manualmente, porém, esta classificação pode ser influenciada por conhecimentos prévios e inviável para grandes conjuntos de dados.

Desse modo, a utilização de um algoritmo de classificação é indicado para classificação dos conjuntos de dados. Dentre os algoritmos de aprendizagem, o não supervisionado é amplamente utilizado por não possuir conhecimento do valor de saída para determinadas entradas, visando encontrar similaridade entre os dados de entrada. Assim, os padrões são identificados a partir dos dados de entrada (MOTA; DAMASCENO; LEITE, 2018 apud LEITE et al., 2012).

Dado um conjunto de dados X, o algoritmo de clusterização atribui rotulação aos objetos de um determinado subconjunto no conjunto X. Isso é possível por meio do uso de uma medida de distância entre os objetos. A partir da medida determinam-se os clusters, particionando o conjunto de objetos em  $c$  clusters, tornando os objetos do subconjunto, definido pela medida de distância, altamente associados. As principais medidas de distância usadas nos algoritmos são as normas Manhattan (1 norma), euclidiana (2 normas), Chebyshev ( $\infty$ -norma) e a distância de Mahalanobis (MOTA; DAMASCENO; LEITE, 2018). No último caso é necessário o cálculo de uma matriz de covariância. O tempo de computação cresce quadraticamente com a quantidade de variáveis.

### 3.8 Clusterização Fuzzy

A clusterização pode seguir uma abordagem clássica ou fuzzy. Na abordagem clássica, cada elemento pertence unicamente a uma determinada classe. Já na abordagem fuzzy, cada elemento pode pertencer a várias classes com diferentes valores de pertinência.

Os algoritmos K-Means (KM), Fuzzy C-Means (FCM), Gustafson-Kessel (GK), possuem aspectos em comum referentes à estrutura do algoritmo, sendo:

- Inicialização aleatória dos clusters e configuração dos parâmetros iniciais;
- Para cada amostra na base de dados, calcular a distância para cada cluster e associar ao que estiver mais próximo;
- Atualizar a matriz de partição, o centro de cada cluster, e parâmetros associados.

Os passos b e c são repetidos até o algoritmo convergir conforme um critério adotado.

Tem-se como distinção dos algoritmos o método para calcular a distância entre cluster e amostra. Os algoritmos KM e FCM utilizam a distância Euclidiana, e o algoritmo GK utiliza a distância de Mahalanobis.

#### 3.8.1 K-Means

O objetivo do algoritmo de clusterização KM é particionar um conjunto de dados  $X$  em  $c$  clusters. O algoritmo é baseado na minimização da função objetivo demonstrada na Equação 3.24, em relação a  $U$ , uma  $K$ -partição fuzzy do conjunto de dados, e a  $V$ , os centroides (protótipos) (GATH; GEVA, 1989).

$$J_q(U, V) = \sum_{j=1}^N \sum_{i=1}^K (u_{ij})^q d^2(X_j, V_i); K \leq N \quad (3.24)$$

onde  $q$  é um número real maior que 1,  $X_j$  é a  $j$ -ésimo vetor de recursos de  $m$ -dimensional,  $V_i$  é o centroide do  $i$ -ésimo cluster,  $u_{ij}$  é o grau de pertinência de  $X_j$  no  $i$ -ésimo cluster,  $d^2(X_j, V_i)$  é qualquer métrica de produto interno (distância entre  $X_j$  e  $V_i$ ),  $N$  é o número de pontos de dados e  $K$  é o número de clusters.

A partição fuzzy é realizada por uma otimização iterativa da Eq.3.24 conforme os seguintes passos (GATH; GEVA, 1989).

- Escolha primária dos centroides  $V_i$ ;
- Cálculo do grau de pertinência de todos os vetores de recursos em todos os clusters, conforme a Equação 3.25;

$$u_{ij} = \frac{\left| \frac{1}{d^2(X_j, V_i)} \right|^{1/(q-1)}}{\sum_{k=1}^K \left| \frac{1}{d^2(X_j, V_k)} \right|^{1/q-1}} \quad (3.25)$$

- Cálculo dos novos centroides  $\hat{V}_i$ , Equação 3.26;

$$\hat{V}_i = \frac{\sum_{j=1}^N (u_{ij})^q X_j}{\sum_{j=1}^N (u_{ij})^q} \quad (3.26)$$

- Atualização dos graus de pertinência  $u_{ij}$  para  $\hat{u}_{ij}$  conforme 3.25;
- Se  $\max_{ij} [|u_{ij} - \hat{u}_{ij}|] < \varepsilon$ , ocorre a parada, caso contrário é calculado o novo centroide, sendo  $\varepsilon$  o critério de parada com valor entre 0 e 1.

O cálculo do grau de pertinência  $u_{ij}$  depende da definição da distância calculada, conforme a Equação 3.27. Para os casos em que  $A$  é igual a matriz identidade, a distância é Euclidiana (GATH; GEVA, 1989).

$$d^2(X_j, V_i) = (X_j - V_i)^T A (X_j - V_i) \quad (3.27)$$

### 3.8.2 Fuzzy C-Means

Vários critérios de clusterização foram propostos para identificar partições fuzzy  $c$  em uma determinada base de dados  $X$ . Destes, o mais corriqueiro está associado ao erro dos mínimos quadrados generalizados (BEZDEK; EHRLICH; FULL, 1984), conforme a Equação 3.28.

$$J_m(U, V; X) = \sum_{k=1}^N \sum_{i=1}^C (u_{ik})^m \|x_k - v_i\|_A^2 \quad (3.28)$$

sendo:

- $X \{x_1, x_2, \dots, x_N\} \subset \mathbb{R}^n$ ;
- $c$  Números de clusters em  $X$  ( $2 \leq c < n$ );

- N Número de amostras;
- m Expoente de ponderação ( $1 \leq m \leq \infty$ );
- U Partições fuzzy de X;
- V ( $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c$ ) é o vetor dos centros dos clusters;
- $v_i (v_{i1}, v_{i2}, \dots, v_{in})$  centro do cluster;
- $\| \cdot \|_A$  Norma matricial induzida  $A \in \mathbb{R}^n$ ;
- A Matriz de partição (n x n).

A matriz de partição A é a identidade no FCM, induzindo clusters esféricos. A distância ao quadrado entre  $x_k$  e  $v_i$  na Equação 3.28 é dada pelo produto interno da diferença, apresentada na Equação 3.29.

$$d_{ik}^2 = \|x_k - v_i\|_A^2 = (x_k - v_i)^T A (x_k - v_i) \quad (3.29)$$

O peso do erro quadrático está associado a  $(u_{ik})^m$  por 3.28, elevado ao m-ésimo expoente de ponderação da pertinência de  $x_k$  no cluster i. Os centros dos clusters são obtidos a partir de:

$$v_i = \frac{\sum_{k=1}^N (u_{ki})^m x_k}{\sum_{k=1}^N (u_{ki})^m} \quad (3.30)$$

Seja uma matriz de partição  $U \in \mathbb{R}^{c \times N}$  constituídas de elementos  $u_{ik}$  conforme:

$$U = \begin{pmatrix} u_{11} & \dots & u_{1c} \\ \vdots & \ddots & \vdots \\ u_{N1} & \dots & u_{Nc} \end{pmatrix} \quad (3.31)$$

Sendo a partição de X,  $u_{ki} \in [0, 1] \forall k, i$ ;  $\sum_{i=1}^c u_{ki} = 1 \forall k$ ; e  $0 < \sum_{i=1}^N u_{ki} < N \forall i$ . Os elementos da matriz de partição são dados por:

$$u_{ki} = \frac{1}{\sum_{j=1}^c \left( \frac{d_{ki}}{d_{kj}} \right)^{\frac{1}{m-1}}} \quad (3.32)$$

Assim, o valor de 3.28 é uma medida da variância total entre  $x_k$  e  $v_i$ , que deve ser minimizado. Os centros dos clusters convergem para um determinado ponto no espaço dos dados

quando não há mais variação significativa da matriz de partição observada, ou seja  $\|\Delta U\| < \varepsilon$ , em que  $\varepsilon$  é um limiar (MOTA; DAMASCENO; LEITE, 2018).

### 3.8.3 Gustafson-Kessel

O algoritmo de clusterização GK foi proposto por Gustafson e Kessel (1978). O algoritmo GK é uma variante do FCM empregando uma norma de distância adaptativa (uma matriz de covariância), a fim de detectar grupos de diferentes formas geométricas em um conjunto de dados (EGRIOGLU et al., 2011).

GK segue os mesmos passos do FCM, porém há uma diferença na distância entre a  $k$ -ésima amostra de dados e o  $i$ -ésimo centro, sendo baseada em uma matriz adaptativa independente  $A_i$ , conhecida como a distância de Mahalanobis, apresentada na Equação 3.33.

$$D_{k_i A_i}^2 = (x_i - v_i)^T A_i (x_i - v_i) \quad (3.33)$$

Matrizes  $A_i$  permitem que os clusters se adaptem à estrutura topológica dos dados. A obtenção dos centros dos clusters é dada para cada interação  $l = 1, 2, \dots$ , calculada conforme a Equação 3.34 (GUSTAFSON; KESSEL, 1978).

$$v_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{k_i}^{l-1})^m x_k}{\sum_{k=1}^N (\mu_{k_i}^{l-1})^m}, i = 1, \dots, c \quad (3.34)$$

As matrizes de covariância são apresentadas na Equação 3.35.

$$A_i^{(l)} = \rho_i \det(F_i) \left(\frac{1}{n}\right) F_i^{-1} \quad (3.35)$$

Em que  $F_i$  é apresentado na Equação 3.36.

$$F_i = \frac{\sum_{k=1}^N (\mu_{k_i}^{l-1})^m (x_k - v_i^{(l)})(x_k - v_i^{(l)})^T}{\sum_{k=1}^N (\mu_{k_i}^{l-1})^m}, i = 1, \dots, c \quad (3.36)$$

Por fim, atualiza-se a matriz de partição de  $X$  de acordo com 3.31 e 3.32. O critério de parada é  $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$ .

Devido à complexidade de se decidir qual o melhor número de clusters, é necessário a aplicação de índices de validação que auxiliam na escolha da quantidade de clusters. A seguir são listados os índices utilizados neste trabalho.

### 3.9 Método de validação — clusterização

Um índice de validação é geralmente utilizado para definir a qualidade de partições, buscando uma escolha eficiente do número de grupos. Diversos métodos de validação são propostos na literatura, entre eles estão: Critério de Xie-Beni (*Xie-Beni criterion*, XB), Entropia de Classificação (*Classification Entropy*, CE), Índice de Partição (*Partition Index*, SC) e Índice de Dunn (*Dunn's Index*, DI), entre outros.

#### 3.9.1 Critério de Xie-Beni

O índice de validação Xie-Beni (XB) é útil para validação do número de clusters, sendo definido como (XIE; BENI, 1991):

$$XB = \frac{\sum_{i=1}^c \sum_{k=1}^n (u_{ik})^m \|x_k - v_i\|_A^2}{N \min_{\forall k,i} \|x_k - v_i\|_A^2} \quad (3.37)$$

O critério tem por finalidade encontrar o valor mínimo, demonstrando o quão compactos e bem separados são os clusters.

#### 3.9.2 Índice de Partição

O Índice de Partição é a razão da soma da compactação e separação dos clusters, consistindo em uma soma de medidas de validade de cluster individual normalizado por meio da divisão pela cardinalidade difusa de cada cluster (BENSAID et al., 1996). O índice é obtido pela Equação 3.38.

$$SC = \sum_{i=1}^c \frac{\sum_{k=1}^N (u_{ik})^m \|x_k - v_i\|_2}{N_i \sum_{k=1}^c \|x_k - v_i\|_2} \quad (3.38)$$

O Índice de Partição é útil ao comparar diferentes partições com números iguais de clusters. A minimização do valor de SC sugere a melhor partição.

### 3.9.3 Entropia de Classificação

O índice de Entropia de Classificação (*Classification Entropy*, CE) mede a imprecisão da partição do cluster apenas. O índice é obtido pela Equação 3.39.

$$CE = -\frac{1}{N} \sum_{i=1}^c \sum_{j=1}^N \mu_{ij} \log(\mu_{ij}) \quad (3.39)$$

A maximização do valor de CE sugere o melhor número de cluster.

### 3.9.4 Índice de Dunn

Como todos os índices citados, o Índice de Dunn (*Dunn's Index*, DI) tem por objetivo identificar o conjunto de clusters que são compactos, com uma pequena variância entre os membros do cluster e bem separados, em que as médias dos clusters estão suficientemente distantes. A maximização do valor de DI sugere a melhor partição.

O DI é obtido pela Equação 3.40. A principal desvantagem do DI é computacional, pois o cálculo torna-se computacionalmente muito expansivo à medida que c e N aumentam.

$$DI = \min_{i \in c} \left\{ \min_{i \in c, i \neq j} \left\{ \frac{\min_{i \in C_i, y \in C_j} d(x, y)}{\max_{k \in c} \left\{ \min_{i \in C_i} d(x, y) \right\}} \right\} \right\} \quad (3.40)$$

### 3.10 Técnicas de otimização meta-heurística

A Computação Evolucionária (CE) é um sub-campo da Inteligência Computacional (IC) que consiste em várias técnicas para solucionar problemas complexos em diferentes ambientes. A CE trata de um conjunto de ferramentas computacionais aplicadas em problemas de otimização. Essas ferramentas têm por objetivo reproduzir os comportamentos das espécies da natureza em modelos computacionais.

Na CE, as técnicas de otimização meta-heurísticas tornaram-se populares nas últimas décadas. Entre as técnicas, podem ser destacadas a Otimização de Enxame de Partículas (*Particle Swarm Optimization*, PSO), Evolução Diferencial (*Differential Evolution*, DE) e Otimizador Lobo Cinzento (*Grey Wolf Optimizer*, GWO). Estas são bastantes conhecidos tanto no meio dos cientistas da computação quanto pelos cientistas de diversas áreas.

Uma questão levantada é porque as técnicas meta-heurísticas tem se tornado notavelmente comum, e a resposta é por quatro razões principais: simplicidade, flexibilidade, mecanismo livre de derivação e o fato de evitarem um ótimo local (MIRJALILI; MIRJALILI; LEWIS, 2014).

### 3.10.1 Otimização de Enxame de Partículas

A Otimização de Enxame de Partículas (*Particle Swarm Optimization*, PSO) é um algoritmo baseado em enxames de partículas que tem por inspiração o comportamento social de um bando (KENNEDY; EBERHART, 1995). O Algoritmo consiste em uma série de entidades simples, as partículas, colocadas no espaço de busca e cada partícula avalia a função objetivo.

A partir disso, cada partícula determina seu movimento por meio do espaço de busca com combinações de alguns aspectos de seu local atual, com os aspectos do melhor local de um ou de mais membros do enxame. A partir da realização do movimento de todas as partículas, a próxima interação ocorre.

Cada indivíduo do enxame é composto por três vetores de D-dimensões, sendo D a dimensionalidade do espaço de busca. O primeiro vetor é a posição atual  $\bar{x}_i$ , o segundo é a melhor posição  $\bar{p}_i$  e o terceiro é a velocidade que se move a partícula  $\bar{v}_i$ .

O PSO possui como principais vantagens a simples implementação, insensível à mudança de escala das variáveis, poucos parâmetros a serem definidos e adaptável a computadores paralelos. Entretanto, tem por desvantagem a lentidão no ajuste fino da solução.

### 3.10.2 Otimizador Lobo Cinzento

O Otimizador do Lobo Cinzento (*Grey Wolf Optimizer*, GWO) é um algoritmo relativamente novo no grupo de algoritmos de enxame para resolver problemas de otimização contínua e reais. Ele se baseia no comportamento social dos lobos cinzentos, que vivem em alcateias compostas por cerca de 5 a 11 lobos. Para caçar e manter a disciplina, os lobos se organizam em quatro grupos (MIRJALILI; MIRJALILI; LEWIS, 2014). O GWO é o único algoritmo de enxame baseado na hierarquia de liderança (GUPTA; DEEP, 2019).

O primeiro grupo consiste no lobo líder chamado de alfa ( $\alpha$ ), responsável pelas decisões cruciais da alcateia. Os lobos do segundo grupo é o beta ( $\beta$ ), sendo este subordinado ao alfa e o ajuda nas tomadas de decisões como caçar ou em outras atividades. O terceiro grupo conhecido

como lobo ômega ( $\omega$ ) sempre tem que se submeter aos lobos dominantes e são os últimos lobos a comer. Por fim, o último grupo é chamado de subordinado ou delta  $\delta$ , que se submetem aos lobos alfas e betas, mas eles dominam o ômega.

Além da hierarquia social dos lobos, a caça em grupo é outro comportamento social interessante dos lobos cinzentos, em que segundo (MURO et al., 2011) as principais fases da caça consiste em: perseguir e se aproximar da presa, rodear a presa e atacá-la.

### 3.10.3 Evolução Diferencial

A Evolução Diferencial (*Differential Evolution*, DE) é um modelo matemático simples de um grande e naturalmente complexo processo de evolução, pertencente aos algoritmos de otimização evolucionários.

A DE tem por característica ser autoajustável, por deduzir as informações de perturbação entre os vetores que compõem a própria população. Este recurso produz perturbações razoavelmente grandes no início, porém quando a população está se aproximando do ótimo, as distâncias ficam menores, o que permite que a DE conduza uma busca refinada para o ótimo.

O algoritmo DE é controlado apenas por três variáveis, sendo o tamanho da população,  $N$  o fator de escala de mutação  $F$  e a constante de *crossover*  $CR$  (PRICE, 1997).

## 3.11 Métodos de medição de desempenho dos sistemas de classificação

Os métodos de medição de desempenho dos sistemas de classificação são úteis quando o objetivo é avaliar e comparar diferentes modelos de classificação ou técnicas de aprendizagem de máquinas. As métricas são indispensáveis para testar a capacidade de qualquer classificador que auxilie na comparação do desempenho de modelos diferentes ou para observar o comportamento do mesmo modelo para a realização dos ajustes dos seus parâmetros.

### 3.11.1 Matriz de confusão

Uma das métricas muito utilizadas em aprendizagem de máquina é a matriz de confusão. A partir da matriz é possível obter métricas de desempenho do classificador, pelo fato de ela

conter todas as informações importantes referente ao desempenho do classificador (GRANDINI; BAGLI; VISANI, 2020).

A matriz de confusão é uma tabela cruzada que registra o número de ocorrências entre dois avaliadores, representadas as instâncias das classes reais nas colunas e as instâncias classificadas pelo classificador conforme a Tabela 3.1.

Tabela 3.1 – Matriz de confusão.

		Valor Real	
		Positivo	Negativo
Valor Previsto	Positivo	TP (Verdadeiro Positivo)	FN (Falso Negativo)
	Negativo	FP (Falso Positivo)	TN (Verdadeiro Negativo)

Fonte: Do autor (2022).

### 3.11.2 Acurácia

A acurácia é considerada uma das métricas mais simples e importantes. Ela avalia o percentual de acertos, mostrado na Equação 3.41.

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.41)$$

### 3.11.3 Sensibilidade

A sensibilidade é a capacidade de um modelo para detectar todas as amostras positivas. Esta métrica é definida pela razão entre o verdadeiro positivo sobre o total de positivos, conforme a Equação 3.42.

$$Sensibilidade = \frac{TP}{TP + FN} \quad (3.42)$$

### 3.11.4 Precisão

A precisão avalia a quantidade de verdadeiros positivos sobre a soma de todos os valores positivos, a qual é apresentada na Equação 3.43.

$$Precisão = \frac{TP}{TP + FP} \quad (3.43)$$

### 3.11.5 Especificidade

A especificidade avalia a capacidade do método de detectar resultados negativos, sendo apresentada na Equação 3.44.

$$Especificidade = \frac{TN}{TN + FP} \quad (3.44)$$

### 3.11.6 F1-score

O F1-score avalia o desempenho do modelo por meio da matriz de confusão, associando a precisão juntamente com a sensibilidade sob o conceito de média harmônica (GRANDINI; BAGLI; VISANI, 2020), a qual é apresentada na Equação 3.45.

$$F1 - score = 2 \left( \frac{precisão \cdot sensibilidade}{precisão + sensibilidade} \right) \quad (3.45)$$

A Equação 3.45 pode ser interpretada como a média ponderada entre a precisão e a sensibilidade em que o melhor valor ocorre quando atinge 1 e o pior valor atinge 0.

As métricas de precisão e sensibilidade podem se referir a classificações binárias quanto classificações de multi classes. No caso binário, considera-se apenas a classe Positiva, enquanto no caso multi classe considera-se todas as classes uma a uma e, conseqüentemente, todas as entradas da matriz de confusão (GRANDINI; BAGLI; VISANI, 2020).

No caso de multi classes, é necessário que as medidas de precisão e sensibilidade sejam medidas de multi classes. Essas métricas podem ter duas especificações diferentes, dando origem a duas métricas diferentes, sendo Micro F1-Score e Macro F1-Score.

#### 3.11.6.1 Macro F1-score

Para o cálculo da Macro F1-score, necessariamente deve-se calcular a Macro precisão e sensibilidade. Estas são calculadas tomando a precisão média para cada classe predita e a sensibilidade média para cada classe real (GRANDINI; BAGLI; VISANI, 2020).

Por este fato, a abordagem Macro considera todas as classes como elementos básicos do cálculo, em que cada classe tem o mesmo peso na média, não havendo distinção entre as classes.

A precisão e a sensibilidade para cada classe utilizam-se as Equações 3.43 e 3.42. A Macro precisão e sensibilidade podem ser simplificadas calculando-se a média aritmética (GRANDINI; BAGLI; VISANI, 2020). Deste modo, a Macro F1-score é apresentada na Equação 3.46.

$$MacroF1 - scoreMédia = 2\left(\frac{MacroPrecisãoMédia \cdot MacroSensibilidadeMédia}{MacroPrecisãoMédia + MacroSensibilidadeMédia}\right) \quad (3.46)$$

### 3.11.6.2 Micro F1-score

Como ocorrido com a Macro F1-score, para realizar o cálculo da Micro F1-score é necessário os cálculos da Micro precisão e Micro sensibilidade. No caso da Micro média, considera-se todas as unidades juntas, sem considerar possíveis diferenças entre as classes. A Micro F1-score é apresentada na Equação 3.47.

$$MicroF1 - scoreMédia = \frac{\sum_{k=1}^K TP_k}{TotalGeral} \quad (3.47)$$

onde o Total Geral é a soma dos valores das linhas e das colunas da matriz de confusão.

### 3.11.7 Coeficiente de concordância Kappa

O coeficiente de concordância Kappa, ou simplesmente índice Kappa, foi proposto por (COHEN, 1960). Esta métrica tem suas origens no campo da psicologia sendo adequado à aprendizagem de máquina.

O índice baseia-se na ideia de medir a concordância entre os rótulos previstos e verdadeiros, considerados duas variáveis categóricas aleatórias (GRANDINI; BAGLI; VISANI, 2020).

Cohen (1960) avaliou a classificação de dois avaliadores, ou seja, previsão do modelo e distribuição real para encontrar uma medida de concordância entre eles, ou melhor, procurou um método estatístico que se fornece o grau de concordância entre dois ou mais conjuntos de medidas (GRANDINI; BAGLI; VISANI, 2020). Matematicamente, o índice Kappa é calculado conforme a Equação 3.48.

$$k = \frac{p_o - p_e}{1 - p_e} \quad (3.48)$$

onde  $p_o$  é a proporção de concordância observada,  $p_e$  é a taxa hipotética de aceitação. Quando há concordância total entre os dois conjuntos de dados, o índice Kappa será igual a 1.

Em (LANDIS; KOCH, 1977) foi proposta a classificação dos diferentes níveis de concordância conforme a Tabela 3.2.

Tabela 3.2 – Níveis de concordância do índice Kappa.

Coefficiente Kappa	Nível de concordância
<0	Não existe
0 - 0.20	Pouca
0.21 - 0.40	Justa
0.41 - 0.60	Moderada
0.61 - 0,80	Substancial
0.81 - 1.00	Quase perfeita

Fonte: (LANDIS; KOCH, 1977).

## 4 METODOLOGIA

Este capítulo tem como premissa a descrição da implementação e testes dos classificadores Perceptron de Multicamadas (*MultiLayer Perceptron*, MLP), Máquina de Vetores de Suporte (*Support vector machine*, SVM) juntamente com os métodos de clusterização K-means (KM), Fuzzy c-means (FCM) e Gustafson-Kessel (GK) para identificação das cargas de uma linha de produção.

Para isso, foi utilizado um computador Dell<sup>1</sup> com as seguintes especificações:

- Processador core i5 11<sup>a</sup> geração;
- Memória de 8 GB;
- Armazenamento de 256 GB SSD;
- Windows 10 Home;
- Software MATLAB<sup>2</sup> R2011.

O software MATLAB<sup>2</sup> foi utilizado com a finalidade de se implementar os classificadores propostos. Para isso, foi aplicada a Toolbox ANN para o MLP e funções nativas do software para a SVM. Já a implementação dos métodos de clusterização, a *Fuzzy Clustering and Data Analysis Toolbox* (FCDAT) foi adotada, a qual é proposta por Balasko, Abonyi e Feil (2005).

A implementação dos classificadores foi dividida em três etapas apresentadas a seguir.

- Primeira etapa: Ênfase na aquisição, pré-processamento dos dados e a seleção de características;
- Segunda etapa: implementação dos classificadores MLP e a SVM com a aplicação das técnicas meta-heurísticas;
- Terceira etapa: implementação dos métodos de clusterização.

Na primeira etapa, a aquisição dos dados por meio de um medidor instalado na linha de produção pelo método de Monitoramento de Carga Não Intrusivo (*Non-intrusive Load Monitoring*, NILM), em que as cargas foram acionadas uma de cada vez. Neste estágio é gerada uma base de dados com os dados coletados.

---

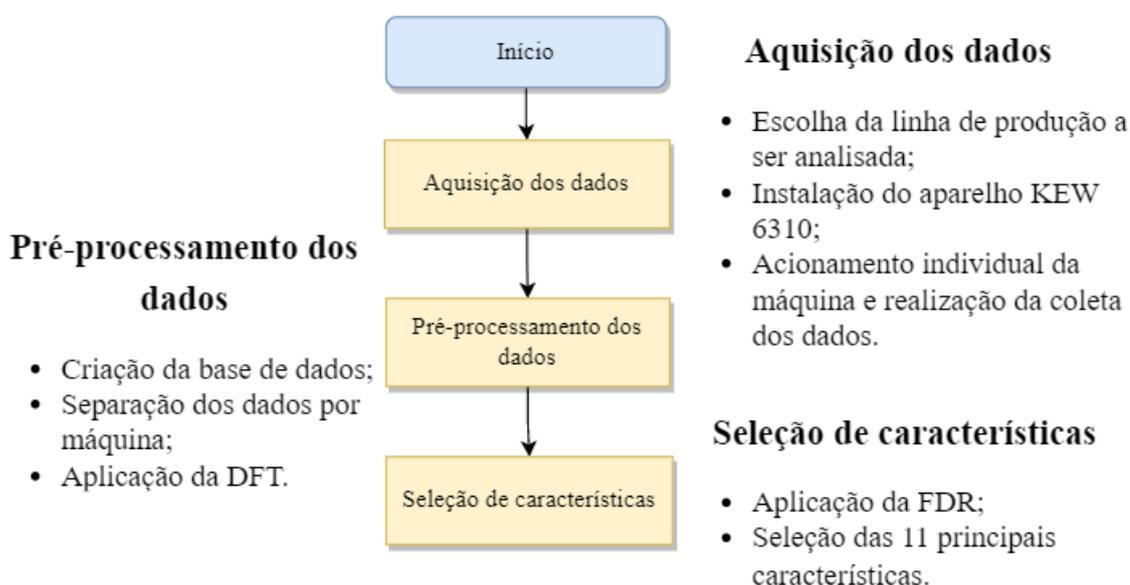
<sup>1</sup> [www.dell.com](http://www.dell.com)

<sup>2</sup> [www.mathworks.com](http://www.mathworks.com)

Após a aquisição, realizou-se o pré-processamento dos dados em que se aplicou a Transformada Discreta de Fourier (*Discrete Fourier Transform*, DFT) a fim de preparar, organizar e estruturar os dados, sendo uma etapa importante por determinar a qualidade final dos dados.

Por fim, foi realizada a seleção das características por meio da Razão Discriminante de Fisher (*Fisher's Discriminant Ratio*, FDR) cujo objetivo é reduzir a dimensionalidade e conservar as informações intrínsecas dos dados. Na Figura 4.1 encontra-se o fluxograma da primeira etapa.

Figura 4.1 – Fluxograma da primeira etapa.

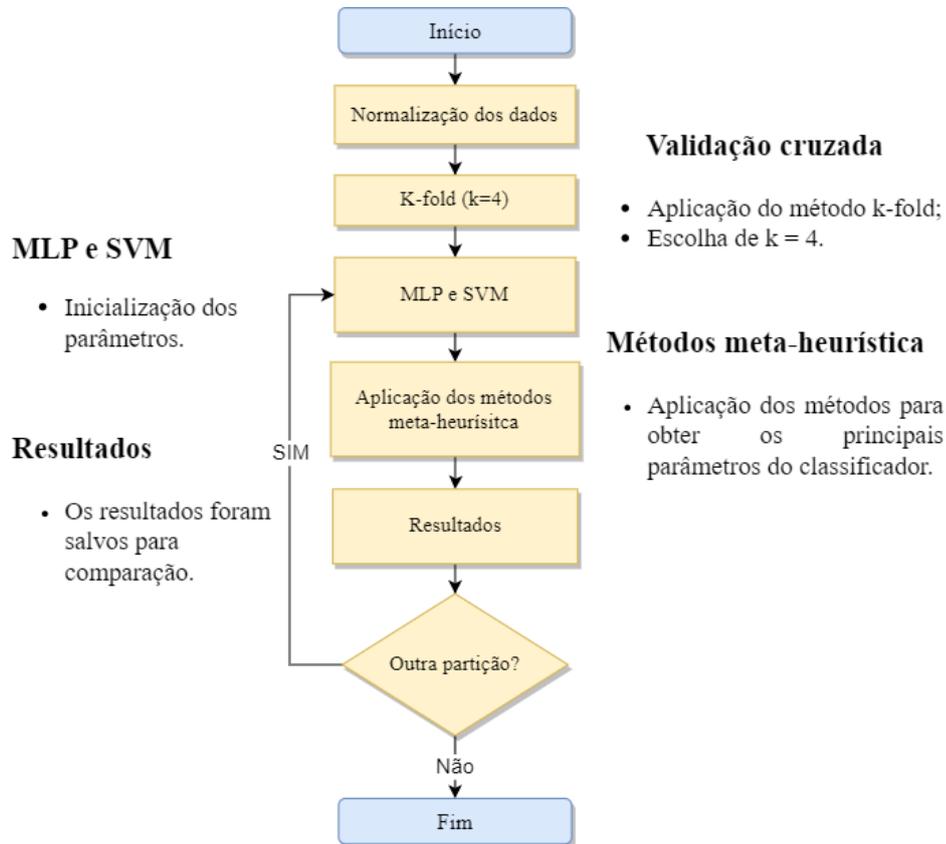


Fonte: Do autor (2022).

Na segunda etapa do projeto, foram aplicadas técnicas de aprendizado de máquina usando os classificadores MLP e SVM nos dados coletados e processados na primeira etapa. Antes da aplicação dos algoritmos, os dados foram normalizados para remover as diferenças de escala entre as variáveis. Além disso, técnicas meta-heurísticas foram utilizadas para explorar espaços de busca de forma estratégica. As etapas de aplicação foram semelhantes para ambos os classificadores, a fim de permitir a comparação dos resultados, conforme ilustrado na Figura 4.2.

A terceira e última etapa do trabalho consiste na aplicação de técnicas de clusterização para agrupar dados similares sem a necessidade de supervisão. Para isso, os dados foram normalizados e em seguida foram aplicados índices de validação cruzada para determinar a melhor quantidade de clusters. Após essa etapa, os clusters foram rotulados com base na distância entre os dados de entrada e os clusters, sendo que o cluster com a menor distância recebeu o rótulo correspondente à saída definida na base de dados. Por fim, os resultados foram obtidos e avaliados.

Figura 4.2 – Fluxograma do processo de treinamento e análise dos resultados para o MLP e SVM.



Fonte: Do autor (2022).

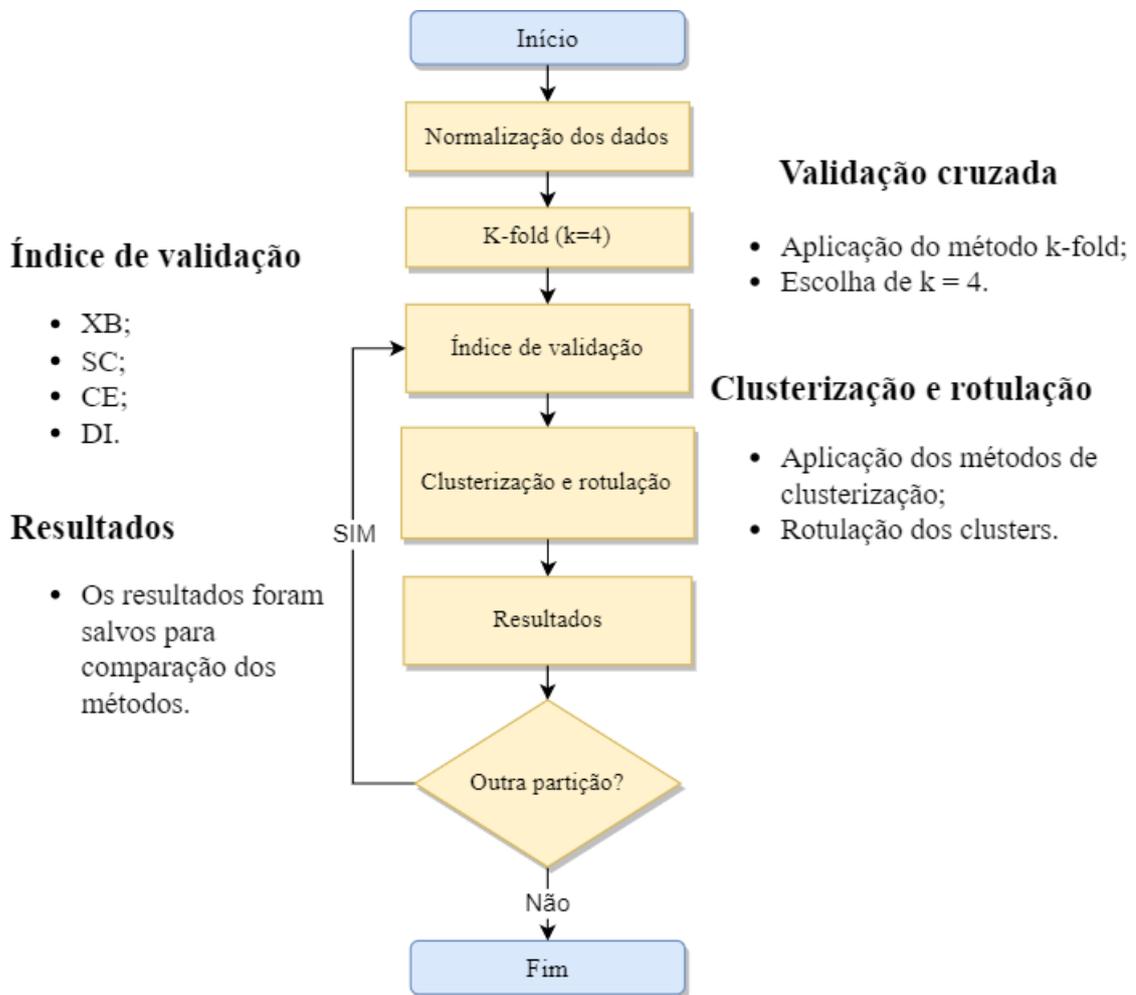
Embora o problema de monitoramento de cargas não intrusivo seja um problema supervisionado, foram empregados métodos de clusterização com técnicas de aprendizado não supervisionado para fins de comparação dos resultados na prática. Assim, o fluxograma dessa etapa, apresentado na Figura 4.3, é semelhante ao fluxograma dos classificadores MLP e SVM.

Após ilustrado passo a passo, a seguir serão demonstrados os detalhes de cada etapa.

#### 4.1 Aquisição dos dados

Visando projetar classificadores de cargas industriais, foi primeiramente escolhida uma linha de produção com o intuito de realizar a aquisição dos dados. Deste modo, foi definida uma linha de produção de amortecedores automobilístico. Esta escolha foi baseada na disponibilidade das linhas de produção, pelo fato da necessidade das máquinas estarem ligadas e existir produto a ser montado para a realização da aquisição. A linha é composta por cinco máquinas elétricas, que serão listadas por classes, conforme a Tabela 4.1.

Figura 4.3 – Fluxograma do processo de treinamento e análise dos resultados para os métodos de clusterização.



Fonte: Do autor (2022).

Tabela 4.1 – Classes (C) para as cargas industriais.

Classe	Carga
C1	Prensa Válvula
C2	Dosadora de óleo
C3	Teste de tração
C4	Dinamômetro
C5	Roladora

Fonte: Do autor (2021).

Destaca-se entre as principais características similares das máquinas a alimentação trifásica de 440V, frequência de 60Hz, além de possuírem motores de indução e sistemas pneumáticos. No entanto, informações detalhadas como potência, corrente nominal e outras não puderam ser obtidas devido à restrição de acesso às informações. Na linha de produção, as

principais informações incluem a rede de alimentação trifásica, tensão de 440V, frequência de 60Hz e corrente máxima de 200A, conforme especificado no quadro elétrico principal.

Em seguida, foi definido o analisador de qualidade de energia *KYORITSI KEW 6310*<sup>1</sup> para a realização da aquisição dos dados, pelo fato de estar disponível e ser um medidor utilizado na empresa. A frequência de amostragem do analisador é de 7,68 kHz.

Na Figura 4.4 encontra-se o analisador utilizado e na Figura 4.5 é ilustrado o diagrama de seus principais componentes, os quais serão listados a seguir.

- *Current Input*: braçadeira para medir a corrente;
- *Voltage Input*: plug para medir a tensão;
- USB: conexão com o computador via USB;
- *Digital Output*: caso necessite de uma saída digital;
- *Analogue Input*: entrada analógica;
- CF Card: cartão em que são salvos os dados coletados pelo medidor.

Figura 4.4 – Analisador de qualidade de energia *KYORITSI KEW 6310*.



Fonte: Do autor (2021).

Para a realização da aquisição dos dados, foram primeiramente configurados os parâmetros do aparelho, sendo a tensão de 440V, corrente máxima de 200A e frequência de 60Hz. Após realizada a configuração, definiu-se quais grandezas elétricas seriam analisadas e coletadas.

<sup>1</sup> <https://www.kew-ltd.co.jp>



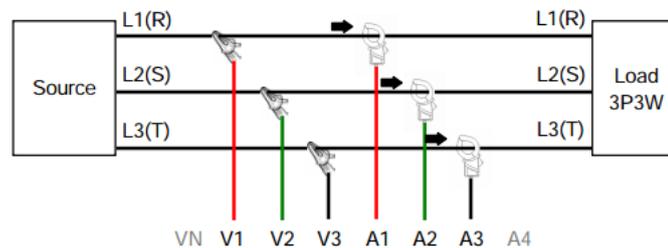
A instalação do analisador foi a mesma apresentada no diagrama proposto em (KYORITSU, 1992), no qual os plugues e as braçadeiras foram conectados nas fases seguindo as cores conforme a Figura 4.7. Na Figura 4.8 é ilustrada a ligação real do analisador no barramento principal do painel elétrico da linha.

Figura 4.6 – Instalação do aparelho no painel elétrico da linha de produção.



Fonte: Do autor (2021)

Figura 4.7 – Diagrama de conexão.



Fonte: (KYORITSU, 1992).

O processo de instalação do medidor e a aquisição dos dados ocorreu durante o terceiro turno da produção, iniciando a instalação do medidor às 22h. Após a instalação, foi necessário aguardar a pausa na linha para o jantar, que ocorreu às 23h15 para iniciar de fato a coleta dos dados. Este processo durou aproximadamente 30min.

Existem duas opções de armazenamento dos dados disponíveis no analisador, sendo o armazenamento por meio de um cabo USB conectado a um computador ou por meio de um

cartão de memória CF Card. No caso em questão, o método de armazenamento utilizado foi o CF Card, que permite o armazenamento e fornecimento dos dados em dois arquivos no formato (.csv). Esses arquivos foram disponibilizados pelo medidor e podem ser acessados por meio do link: <[https://drive.google.com/drive/folders/1zA78RSDoCleHL2Xa6mMxr5GMhTn7ThJW?usp=share\\_link](https://drive.google.com/drive/folders/1zA78RSDoCleHL2Xa6mMxr5GMhTn7ThJW?usp=share_link)>.

Figura 4.8 – Conexão dos plugues e braçadeiras nos barramentos de alimentação.



Fonte: Do autor (2021).

Por fim, após as configurações e instalação do medidor, iniciou-se a aquisição dos dados. Ressalta-se que durante a aquisição todas as máquinas elétricas (cargas) encontravam-se energizadas, porém, sem produção. Por este fato, a corrente mensurada não é zero quando todas as cargas não estão produzindo.

Para analisar individualmente as cargas, cada uma delas foram acionadas separadamente. Esse procedimento foi adotado porque a produção foi retomada antes do horário previsto, impossibilitando a obtenção de dados para acionamentos simultâneos. É importante ressaltar que o acionamento individual das cargas também é comum na linha de produção, já que geralmente um operador é responsável por operar as cinco máquinas, uma de cada vez.

Assim, com todas as cargas energizadas, porém, sem produção, posicionou-se o produto na primeira carga (C1). Após o posicionamento, foi registrado o tempo, com o auxílio do relógio do medidor, a fim de registrar o tempo em que C1 encontrava-se em repouso. Em seguida, acionou-se o *start* da carga e iniciou-se a produção. Novamente, o tempo foi registrado visando obter o tempo de transição em que a carga encontrava-se em repouso e o início do ciclo de produção. Por fim, ao término do ciclo, foi registrado o tempo, o qual possibilitou obter o intervalo de tempo em que C1 encontrava-se em repouso, o início e o fim de seu ciclo.

Com a realização da aquisição dos dados conforme descrito, foi possível ter conhecimento dos tempos exatos para a realização da separação dos dados posteriormente. Este procedimento

foi realizado para as demais cargas e assim foi possível obter uma base de dados detalhada a seguir.

## **4.2 Base de Dados**

Após a etapa de aquisição dos dados, foi necessário analisá-los para a construção de uma base de dados sólida para a realização do pré-processamento dos dados. Para isso, foi utilizado o segundo arquivo em que se encontram os valores instantâneos da corrente elétrica.

O analisador fornece 256 amostras com periodicidade de aquisição a cada 10 segundos, em que o horário da aquisição é registrado. Em cada amostra há dois períodos de ondas de 60Hz, ou seja, um período de onda encontra-se em 128 amostras. Deste modo, a frequência de amostragem é de 7,68 kHz.

Com base nos dados fornecidos e com os intervalos de tempo registrados na aquisição, foi possível realizar a separação dos dados para cada classe manualmente. Para isso, foi importado o arquivo para o software MATLAB (após esta etapa todos os procedimentos foram realizados no software) e efetuou-se a separação dos dados comparando o horário de aquisição e os intervalos registrados.

## **4.3 Pré-processamento dos dados**

Após a separação dos dados por classe, realizou-se o pré-processamento para maximizar os resultados dos classificadores. Dessa forma, aplicou-se a DFT que permite a decomposição do sinal temporal em frequências para cada classe. A aplicação da DFT se justifica pela possibilidade de decompor um sinal em suas componentes de frequência, evidenciando quais frequências são predominantes no sinal e em que intensidade. A Tabela 4.2 apresenta a quantidade de aquisições em que a DFT foi aplicada para cada classe.

Após a aplicação da DFT obteve-se 65 valores de frequências, cada uma com sua amplitude e fase correspondente, deste modo, haveriam 65 atributos de entrada no classificador. Pelo fato deste número ser elevado, o qual eleva a complexidade do classificador, foi necessário realizar a seleção das principais características dos sinais. Para isso, utilizou-se a FDR.

Tabela 4.2 – Quantidade de formas de onda de um período de 60Hz utilizadas na DFT para cada classe.

Classe	Formas de onda
1	26
2	34
3	32
4	31
5	25

Fonte: Do autor (2021).

#### 4.4 Seleção de características

O método FDR é uma abordagem matemática facilmente implementável em computadores e a seleção das características mais relevantes depende da média e variância dos dados de cada classe. Neste estudo, foi aplicado o método FDR aos dados de amplitude obtidos pela DFT, selecionando 11 harmônicos como as principais características de entrada. O número de características foi definido como 11 com base em testes empíricos. Após a seleção das características, os dados foram divididos em conjuntos de treinamento e teste, como detalhado na seção seguinte.

#### 4.5 Normalização dos dados

A normalização por variância foi empregada como uma técnica de pré-processamento de dados fundamental para a preparação adequada dos dados antes do treinamento do modelo de aprendizado de máquina. Essa abordagem possibilitou o redimensionamento das variáveis, garantindo que todas as variáveis tivessem a mesma variância e, conseqüentemente, eliminando as diferenças de escala entre elas. Além disso, a normalização por variância contribuiu para minimizar a influência de variáveis com grandes diferenças de escala, bem como a melhorar a convergência dos algoritmos de otimização empregados no treinamento das redes.

Ressalta-se que os dados foram normalizados após a seleção das principais características, visando a melhorar a precisão e a estabilidade do modelo, removendo diferenças de escala que podem afetar a sua capacidade de generalização. Vale ressaltar que a aplicação da normalização por variância se mostrou eficaz em aumentar a precisão e a estabilidade dos modelos de aprendizado de máquina desenvolvidos neste estudo.

## 4.6 Validação cruzada

Os métodos de validação são usados para verificar o desempenho do classificador durante e após o seu treinamento. A validação cruzada é um método de validação em que os dados são divididos em partições para avaliar o desempenho dos modelos de aprendizagem de máquina. Este método é amplamente utilizado para verificar a capacidade do modelo de generalização, permitindo que o modelo seja treinado e testado com diferentes conjuntos de dados, a fim de avaliar sua capacidade de prever novos dados não vistos durante o treinamento. Em geral, quanto maior a precisão do modelo durante a validação cruzada, maior a sua capacidade de generalização e desempenho.

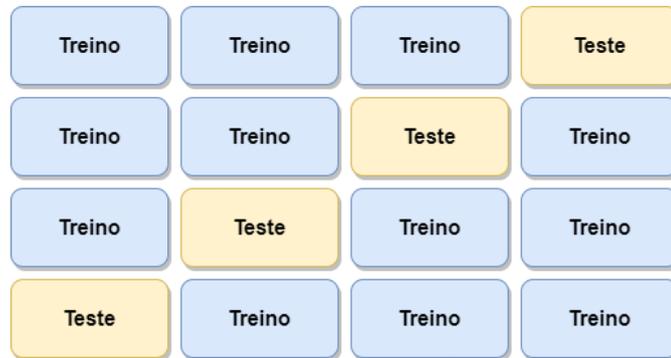
Nesta etapa, será apresentado o método de validação cruzada  $k$ -fold, sendo  $k = 4$  implementado tanto para os classificadores MLP e a SVM quanto para os métodos de clusterização K-Means (KM), Fuzzy  $c$ -means (FCM) e Gustafson-Kessel (GK). Este método foi utilizado para avaliar a eficácia e o desempenho dos classificadores, pelo fato de permitir que o modelo seja avaliado em várias configurações diferentes de treinamento e teste, ajudando a evitar problemas como sobreajuste ou subajuste do modelo.

### 4.6.1 Método K-fold

O método K-fold consiste em dividir o conjunto de dados em  $k$  subconjuntos mutuamente exclusivos e com tamanhos iguais. Deste modo, foi definido  $k = 4$  pelo fato de ser suficiente pela quantidade de amostras por classe.

Na Figura 4.9 é ilustrado a partição do conjunto de dados em subconjuntos aplicando o método. Nas linhas encontram-se as partições dos dados e na colunas os subconjuntos, sendo que  $\frac{3}{4}$  dos dados foram utilizados para treinamento do classificador e  $\frac{1}{4}$  dos dados foram aplicados para teste e obtenção das métricas de desempenho do mesmo.

Para a realização da partição, inicialmente os dados foram embaralhados aleatoriamente visando aleatoriedade dos dados nas partições. Devido a este fato, foi necessário certificar que o subconjunto de testes possuía dados de todas as classes, a fim de evitar teste incompletos, assim, utilizou-se o  $k$ -fold estratificado. Após o particionamento do conjunto de dados, iniciou-se a estimação dos parâmetros dos classificadores, o qual é apresentado na seção a seguir.

Figura 4.9 – K-fold para  $k = 4$ .

Fonte: Do autor (2022).

## 4.7 Redes Neurais Artificiais

Devido à complexidade do problema, e a dimensão dos dados, foi utilizado o Perceptron de Multicamadas (*MultiLayer Perceptron*, MLP) em que consiste em um tipo de RNA com camadas ocultas, e que soluciona problemas não linearmente separáveis. Após definido o tipo da rede a ser utilizada, foram definidos os seus parâmetros para a realização do treinamento e teste da mesma que serão demonstrados a seguir.

### 4.7.1 RNA do tipo MLP

Foram realizados diversos testes empíricos com o objetivo de encontrar os parâmetros ideais para o classificador MLP. Para iniciar, foi configurada uma rede com duas camadas ocultas, visto que essa configuração é eficiente na resolução de problemas não lineares, com no máximo 20 neurônios em cada camada. Os principais parâmetros iniciais definidos para o MLP foram:

- Função de treinamento da rede: retropropagação do gradiente conjugado aliado com a atualização de Polak-Ribière (*traincgp*);
- Função a ser minimizada: erro médio quadrático;
- Taxa de aprendizagem: 0,01;
- Erro máximo:  $10^{-6}$ ;
- Épocas: 1000;

- Máximo de falhas de validação: 200;
- Gradiente mínimo de desempenho:  $1^{-6}$ ;
- 2 camadas escondidas com no máximo 20 neurônios;
- Camada de entrada e saída com 5 neurônios;
- Função de ativação na camada escondida e na saída: tangente hiperbólica.

Após definir os principais parâmetros da rede, o treinamento foi iniciado e, ao final, os dados de teste foram aplicados para obter as métricas de desempenho da rede. No entanto, os resultados iniciais não foram satisfatórios, e novos testes empíricos foram realizados para variar a taxa de aprendizado e a quantidade de neurônios, visando maximizar os resultados. No entanto, os resultados continuaram abaixo do esperado.

Durante os testes empíricos, a variação manual da quantidade de neurônios e da taxa de aprendizado foi exaustiva. Com relação à quantidade de neurônios, existem 400 combinações possíveis, uma vez que há a possibilidade de usar de 1 a 20 neurônios em cada camada oculta, tornando inviável a realização manual de todas elas. Com relação à taxa de aprendizado, foi observado que certos valores levaram ao treinamento divergir e piorar os resultados, conforme destacado por Zhao et al. (2019), que afirmam que "uma taxa de aprendizado inadequada resultará em uma redução na velocidade de convergência ou levará a uma oscilação na convergência".

Pelo fato de a escolha da taxa de aprendizado ser importante para a convergência do treinamento e por existir 400 combinações possíveis de neurônios, que são inviáveis realizar manualmente, foram aplicadas técnicas meta-heurísticas para otimização do problema. As meta-heurísticas aplicadas foram Otimização por Enxame de Partículas (*Particle Swarm Optimization*, PSO), Evolução Diferencial (*Differential Evolution*, DE) e Otimizador Lobo Cinzento (*Grey Wolf Optimizer*, GWO). A seguir são mostradas as definições dos hiperparâmetros de cada técnica.

#### **4.7.2 Técnicas de otimização PSO, DE e GWO aplicadas na topologia e treinamento da RNA**

Os algoritmos PSO, DE e GWO foram aplicados visando encontrar o melhor valor da taxa de aprendizado e a melhor quantidade de neurônios para as 2 camadas ocultas. Os demais parâmetros da rede foram mantidos os definidos inicialmente.

Com o auxílio das técnicas, foi possível aumentar a quantidade de neurônios de 20 para 30, existindo 900 combinações possíveis. Em relação à taxa de aprendizado, definiu-se o intervalo de 0,001 — 0,95. Todos os valores descritos foram obtidos por meio de testes empíricos. A seguir, encontram-se os parâmetros básicos utilizados nas três técnicas meta-heurísticas.

- População: 100 indivíduos;
- Quantidade de neurônios possíveis em cada camada: 1 - 30;
- Taxa de aprendizado: 0,001 — 0,95;
- Interações: 25;
- Critério de parada: 25 interações.

Nota-se que o critério de parada definido foi de 25 interações. Esse foi definido objetivando comparar as três técnicas. Após a definição dos parâmetros básicos, foram definidos os hiperparâmetros de cada técnica conforme a Tabela 4.3.

Tabela 4.3 – Hiperparâmetros das técnicas de otimização PSO, DE e GWO.

Técnica	Hiperparâmetro	Valores
PSO	Taxa de aprendizado 1	2,05
	Taxa de aprendizado 2	2,05
	Taxa de amortecimento	0,97
	Fc	0,73
DE	Fator de escala inferior	0,1
	Fator de escala superior	0,85
	F	0,9
	CR	0,25
GWO	a (diminuição linear)	2 a 0
	$r_1$ e $r_2$	[0,1]

Na técnica PSO, os hiperparâmetros foram definidos tendo como base os valores propostos por Clerc e Kennedy (2002) que demonstraram ser os melhores valores a serem aplicados em problemas de otimização. Para as demais técnicas, foram realizados testes empíricos para se obter os melhores valores.

Após definidas as configurações iniciais, foram realizados os treinamentos das redes. Estes foram divididos em três partes, devido à aplicação das três técnicas meta-heurísticas PSO, DE e GWO. Definiu-se os valores -1 e 1 para a saída da rede, sendo 1 para a carga em produção

e -1 para carga energizada, porém, sem produção. Os resultados são apresentados na seção de resultados e discussões. A seguir é demonstrado as configurações dos parâmetros da Máquina de Vetores de Suporte (*Support Vectors Machine*, SVM).

#### 4.7.3 Máquina de Vetores de Suporte

Foi utilizada a função *fitcecoc* do MATLAB para implementar e treinar a SVM, que ajusta modelos de múltiplas classes para Máquinas de Vetores de Suporte ou outros classificadores, conforme os parâmetros definidos. Assim como na RNA, a SVM requer a definição de parâmetros antes do treinamento da máquina. Para comparar os resultados da MLP e da SVM, foram aplicadas as mesmas técnicas meta-heurísticas utilizadas na MLP.

A seguir, serão apresentados os hiperparâmetros de cada técnica.

#### 4.7.4 Técnicas de otimização PSO, DE e GWO aplicadas na topologia e treinamento da SVM

A fim de comparação, manteve-se os mesmos parâmetros básicos das técnicas meta-heurísticas definidos no MLP, sendo: uma população de 100 indivíduos, 25 interações, e 25 interações como critério de parada. Os demais parâmetros foram obtidos por meio de testes empíricos, os quais são listados a seguir.

- Proporção esperada de *outliers*: 0,0001 — 0,95;
- *Score transform*: *doublelogit*, *ismax*, *sign*, *symmetric*, *symmetriclogit* e *logit*;
- *Kernel function*: gaussiana, rbf, linear e polinomial;
- solucionador: ISDA e SMO;
- Grau do polinômio (em caso polinomial): 1 - 12.

Os hiperparâmetros das três técnicas foram definidos conforme foi apresentado na Tabela 4.3. Após definido os parâmetros da SVM e os hiperparâmetros das técnicas meta-heurísticas, iniciou-se o treinamento e teste. Estes são apresentados em resultados e discussões. Após a implementação da SVM, iniciaram-se as configurações dos parâmetros básicos dos métodos de clusterização que serão apresentados a seguir.

#### 4.7.5 Métodos de clusterização KM, FCM e GK

Nos métodos de clusterização, foi adotado o método de validação cruzada *k-fold*, em que o valor de *k* foi definido como 4, mantendo-se o mesmo valor utilizado nos métodos supervisionados. Essa escolha foi realizada com o intuito de comparar os resultados obtidos nos métodos supervisionados e não-supervisionados, garantindo uma análise mais consistente.

Após definir o método de validação cruzada, para cada partição, foi necessário determinar a quantidade de clusters. Devido à dificuldade de se obter a quantidade manualmente, foram utilizados métodos de validação que fornecem índices de validação para auxiliar na escolha do número ideal de clusters. Os métodos de validação utilizados foram:

- Xie-Beni (*Xie-Beni criterion*, XB);
- Índice de Partição (*Partition Index*, SC);
- Entropia de classificação (*Classification Entropy*, CE);
- Índice de Dunn (*Dunn's Index*, DI).

Pelo fato de existirem 5 classes, definiu-se 5 clusters o mínimo e 25 o máximo. O valor máximo foi limitado em 25 pelo fato que nos testes empíricos realizados, ao aumentar o número de clusters, os índices de validação a partir do 25 começaram a não influenciar na escolha dos clusters, o que aumentava apenas a complexidade do problema.

É importante ressaltar que é possível haver mais clusters do que as classes definidas, o que significa que mais de um cluster pode ser rotulado para a mesma classe. Após a aplicação dos índices de validação e a obtenção do melhor número de clusters, foram rotulados os clusters e os dados de teste foram aplicados. Os resultados são apresentados na seção de resultados e discussões. Em seguida, são apresentadas as métricas de desempenho obtidas após a aplicação dos dados de teste nos classificadores.

#### 4.8 Métricas de desempenho

Com a finalidade de se encontrar o melhor classificador e o melhor método de clusterização, foi necessário gerar matrizes de confusão para cada partição proposto no método de validação. Com o auxílio das matrizes, foi possível obter as seguintes métricas de desempenho para análise dos resultados.

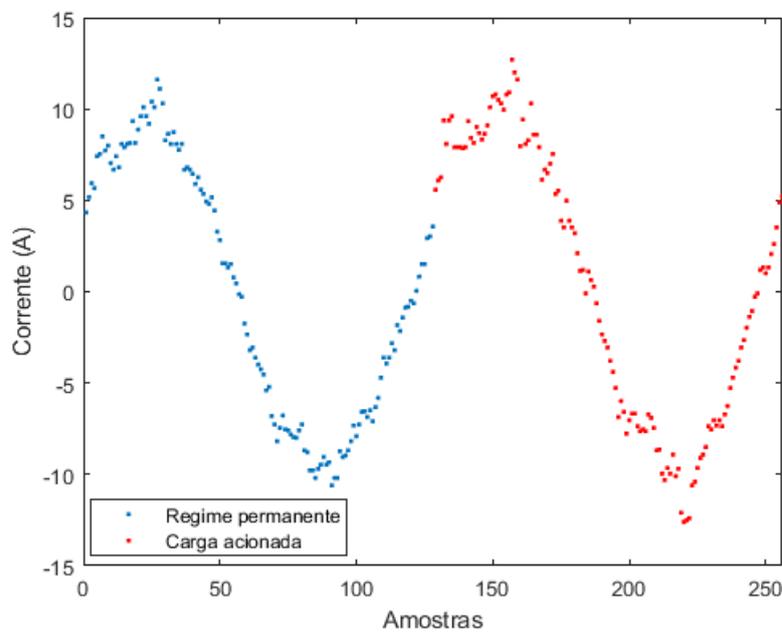
- Precisão;
- Sensibilidade;
- Especificidade;
- Acurácia;
- F1-score;
- Coeficiente Kappa.

Os resultados são apresentados e discutidos na seção a seguir, em que as métricas são analisadas e comparadas.

## 5 RESULTADOS E DISCUSSÕES

Após a realização da separação dos dados comparando o horário de aquisição e os intervalos registrado, os valores instantâneos da corrente em um período de aquisição da C1 é apresentada na Figura 5.1. É relevante ressaltar que na parte azul da figura representa o período em que a carga estava em regime permanente, enquanto na parte vermelha representa o início do ciclo de produção, ou seja, no momento em que a carga foi acionada.

Figura 5.1 – Amostra da corrente no momento em que a C1 inicia a produção do produto.



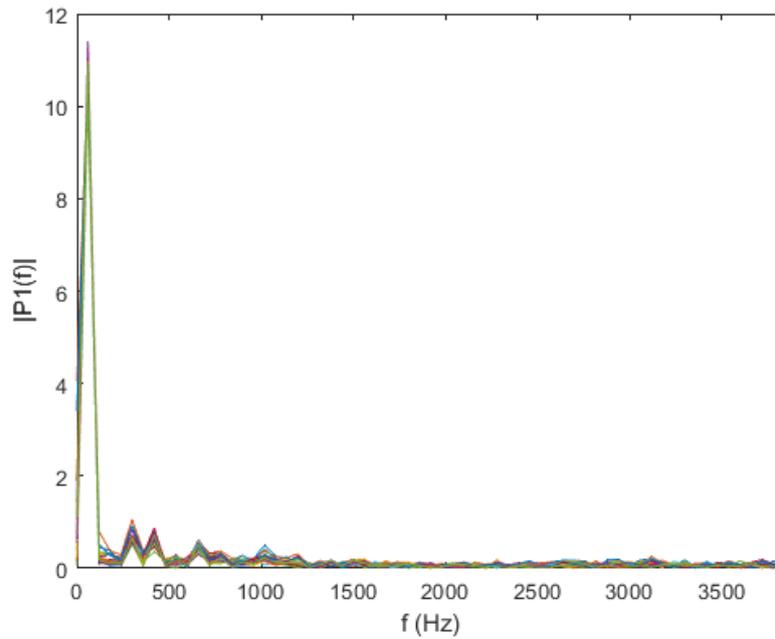
Fonte: Do autor (2021).

Após a realização da Transformada de Fourier Discreta (TFD) nos sinais de corrente das cinco classes, foram selecionadas as principais características utilizando o FDR. A Tabela 5.1 apresenta os harmônicos selecionados, enquanto a Figura 5.2 exibe a TFD do sinal de corrente da C1 com zoom aplicado para facilitar a visualização.

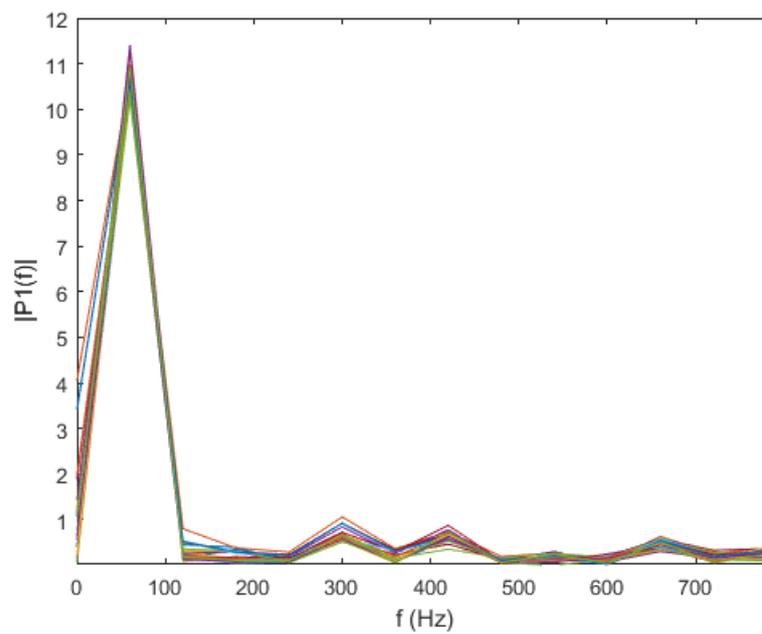
Após a seleção dos principais harmônicos, os dados foram divididos em conjuntos de treino e teste para a aplicação dos classificadores e obtenção das métricas de desempenho. Foram utilizadas duas categorias de classificadores, com base na técnica de aprendizado: supervisionado para os classificadores Perceptron de Multicamadas (*Multilayer Perceptron*, MLP) e Máquina de Vetores de Suporte (*Support Vector Machines*, SVM), e não supervisionado para os métodos de clusterização.

As técnicas de classificação apresentadas a seguir foram avaliadas com base nas métricas de desempenho, visando selecionar a melhor técnica para a classificação do estado operacional de motores elétricos.

Figura 5.2 – DFT da corrente das cinco cargas em análise.



(a) C1.



(b) C2.

Fonte: Do autor (2021).

Tabela 5.1 – Índices e harmônicos obtidos ao aplicar o FDR.

Harmônico	Frequência (Hz)
Fundamental	60
3	180
5	300
7	420
10	600
13	780
25	1500
29	1740
31	1860
47	2820
52	3120

Fonte: Do autor (2021).

### 5.1 Aprendizado supervisionado - MLP e SVM

Os classificadores MLP e a SVM foram divididos em três. A seguir são demonstrados os classificadores.

- MLP
  - MLP-PSO;
  - MLP-DE;
  - MLP-GWO.
- SVM
  - SVM-PSO;
  - SVM-DE;
  - SVM-GWO.

A seguir são apresentados os resultados dos classificadores. O código-fonte utilizados na implementação do classificador MLP-GWO encontra-se no Apêndice A.

### 5.1.1 MLP-PSO

Foram obtidas as matrizes de confusão dos classificadores após o treinamento da rede e a aplicação dos dados de teste em cada partição. As matrizes de confusão dos classificadores, juntamente com as tabelas de acertos e erros de cada classificador (variando o conjunto de treinamento e teste), são apresentadas no Apêndice B. A partir das matrizes de confusão, as principais métricas foram calculadas e estão apresentadas na Tabela 5.2.

Tabela 5.2 – Métricas dos classificadores MLP-PSO.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,9114	0,8917	0,9726	0,8919	0,8875	0,8613
2	<b>0,9556</b>	<b>0,9444</b>	<b>0,9864</b>	<b>0,9474</b>	<b>0,9478</b>	<b>0,9345</b>
3	0,8981	0,8929	0,9726	0,8919	0,8916	0,8647
4	0,9386	0,9214	0,9804	0,9231	0,9246	0,9053
Média	0,9259	0,9126	0,9780	0,9136	0,9129	0,8914
$\sigma$	$\pm 0,0225$	$\pm 0,0219$	$\pm 0,0058$	$\pm 0,0233$	$\pm 0,0248$	$\pm 0,0303$

Fonte: Do autor (2022).

Observa-se na Tabela 5.2 que o melhor classificador foi obtido na partição 2, em que os valores estão destacados em negrito. A seguir, são listados os parâmetros da rede obtidos pelo PSO para o melhor classificador.

- Camada escondida 1: 15 neurônios;
- Camada escondida 2: 12 neurônios;
- Taxa de aprendizagem: 0,5984.

Nota-se na Tabela 5.2 que os valores obtidos na partição 2 foram satisfatórios, ao apresentarem valores acima de 0,93 para todos os índices, fato que ressalta a eficiência do classificador.

Analisando as médias e os desvios padrão das métricas, observa-se que os valores médios estão acima de 0,8914, e os desvios padrão estão abaixo de 0,0303, demonstrando valores satisfatórios, enfatizando a eficiência e a generalização da rede.

Ressalta-se que o coeficiente Kappa apresentou o menor valor entre as médias, sendo de 0,8914. Entretanto, por seu valor estar contido no intervalo 0,81 – 1, comprova-se um resultado satisfatório no geral, pois de acordo com Landis e Koch (1977), os valores contidos nesta faixa apresentam um grau de concordância quase perfeito.

### 5.1.2 MLP-DE

No Apêndice B, são apresentadas as matrizes de confusão dos quatro classificadores MLP-DE, obtidas após o treinamento e a aplicação dos dados de teste. As tabelas de acertos e erros dos classificadores também são apresentadas no mesmo Apêndice. Utilizando essas tabelas, foram calculadas as principais métricas, que estão demonstradas na Tabela 5.3.

Tabela 5.3 – Métricas dos classificadores MLP-DE.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,9264	0,9131	0,9800	0,9211	0,9177	0,8981
2	0,8850	0,8742	0,9670	0,8684	0,8709	0,8380
3	0,8836	0,8714	0,9660	0,8649	0,8672	0,8338
4	<b>0,9492</b>	<b>0,9528</b>	<b>0,9871</b>	<b>0,9474</b>	<b>0,9478</b>	<b>0,9346</b>
Média	0,9110	0,9029	0,9750	0,9004	0,9009	0,8761
$\sigma$	$\pm 0,0280$	$\pm 0,0332$	$\pm 0,0089$	$\pm 0,0351$	$\pm 0,0336$	$\pm 0,0423$

Fonte: Do autor (2022).

Observa-se na Tabela 5.3 que o melhor classificador foi obtido na partição 4, em que os valores estão destacados em negrito. A seguir, são listados os parâmetros da rede obtido pelo DE para o melhor classificador.

- Camada escondida 1: 14 neurônios;
- Camada escondida 2: 30 neurônios;
- Taxa de aprendizagem: 0,4963.

Observa-se na Tabela 5.3 que nas partições 2 e 3 os classificadores apresentaram valores abaixo de 0,9 em todas as métricas, exceto na especificidade, ocasionando aumento no desvio padrão. Entretanto, o valor do maior desvio foi de 0,0423, sendo seu valor aceitável.

Destaca-se, como ocorrido no MLP-PSO, o valor médio do coeficiente Kappa, sendo de 0,8761, apresentou o menor valor entre as métricas. Contudo, seu valor é satisfatório por apresentar um grau de concordância quase perfeito.

### 5.1.3 MLP-GWO

O Apêndice B contém as matrizes de confusão dos quatro classificadores MLP-GWO, que foram obtidas após o treinamento da rede e a aplicação dos dados de teste. Além disso,

as tabelas de acertos e erros dos classificadores também são disponibilizadas no Apêndice B. A partir dessas tabelas, foram calculadas as principais métricas, apresentadas de forma clara e organizada na Tabela 5.3.

Tabela 5.4 – Métricas dos classificadores MLP-GWO.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,9270	0,9167	0,9791	0,9167	0,9143	0,8938
2	<b>0,9400</b>	<b>0,9214</b>	<b>0,9804</b>	<b>0,9231</b>	<b>0,9233</b>	<b>0,9042</b>
3	0,9270	<b>0,9214</b>	0,9802	0,9211	0,9228	0,9030
4	0,9028	0,9028	0,9739	0,8947	0,8956	0,8695
Média	0,9242	0,9156	0,9784	0,9139	0,9140	0,8926
$\sigma$	$\pm 0,0135$	$\pm 0,0076$	$\pm 0,0026$	$\pm 0,0113$	$\pm 0,0112$	$\pm 0,0139$

Fonte: Do autor (2022).

Observa-se na Tabela 5.4 que o melhor classificador foi obtido na partição 2, em que os valores estão destacados em negrito. A seguir, são listados os parâmetros da rede obtidos pelo GWO para o melhor classificador.

- Camada escondida 1: 12 neurônios;
- Camada escondida 2: 26 neurônios;
- Taxa de aprendizagem: 0,2473.

Nota-se na Tabela 5.4 que a sensibilidade destacada na partição 3 foi a mesma obtida na partição 2, porém nas demais métricas o classificador da partição 2 foi superior. Observa-se que a menor média foi apresentada pelo coeficiente Kappa, fato este, ocorrido nos classificadores MLP-PSO e MLP-DE, cujo valor é de 0,8926, sendo satisfatório pelo fato de apresentar um grau de concordância quase perfeito.

Outro ponto importante, é o baixo valor de desvio padrão em relação aos classificadores MLP-PSO e MLP-DE, cujo valor é de 0,0139, enfatizando a generalização da rede.

#### 5.1.4 Resultados dos classificadores MLP

Antes de prosseguir com a análise e comparação dos resultados obtidos pelos melhores classificadores, é importante ressaltar que as classes 1, 2 e 3 e as classes 4 e 5 apresentam maior incidência de erros entre si, conforme as matrizes de confusão das classes estão disponíveis

no Apêndice B. Esse fato se deve às características físicas e mecânicas semelhantes, tornando a tarefa de classificação das cargas mais desafiadora. Vale ressaltar que, por questões de confidencialidade, não é possível fornecer detalhes específicos sobre as características físicas e mecânicas das cargas.

Após apresentados os resultados dos classificadores individualmente, são analisados e comparados os resultados entre os melhores classificadores MLP. Na Tabela 5.5 encontram-se as médias das métricas obtidas para cada classificador proposto, em que as melhores estão destacadas em negrito.

Nota-se que o melhor classificador foi o MLP-GWO, pelo fato de apresentar as melhores métricas em relação aos demais. Destaca-se que na métrica precisão o MLP-PSO foi melhor, entretanto, nas demais métricas MLP-GWO foi melhor, demonstrando ser o melhor.

Tabela 5.5 – Média das métricas dos classificadores MLP.

Método	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
MLP-PSO	<b>0,9259</b>	0,9126	0,9780	0,9136	0,9129	0,8914
MLP-DE	0,9110	0,9029	0,9750	0,9004	0,9009	0,8761
MLP-GWO	0,9242	<b>0,9156</b>	<b>0,9784</b>	<b>0,9139</b>	<b>0,9140</b>	<b>0,8926</b>

Fonte: Do autor (2022).

### 5.1.5 SVM-PSO

Após a definição dos parâmetros iniciais da SVM e do PSO, os treinamentos foram iniciados. Foi aplicado o método de validação k-fold com 4 partições, gerando assim 4 classificadores SVM-PSO. Para avaliar o desempenho dos classificadores, os dados de testes foram aplicados e as principais métricas foram obtidas.

Além das matrizes de confusão dos classificadores, o Apêndice B também apresenta tabelas de acertos e erros correspondentes a cada um deles. Essas tabelas foram utilizadas para o cálculo das métricas apresentadas na Tabela 5.6.

Observando a tabela, nota-se que o melhor classificador foi obtido na partição 3, cujos valores estão destacados em negrito. Os parâmetros obtidos pelo PSO para o melhor classificador são listados a seguir.

- *Outlier Fraction*: 0,4890;

- *Score Transform*: symmetric;
- *Kernel function*: gaussiana;
- *Solver*: ISDA.

Tabela 5.6 – Métricas dos classificadores SVM-PSO.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,8878	0,8607	0,9675	0,8718	0,8601	0,8292
2	0,9000	0,8742	0,9669	0,8718	0,8799	0,8476
3	<b>0,9092</b>	<b>0,8944</b>	<b>0,9719</b>	<b>0,8919</b>	<b>0,8985</b>	<b>0,8709</b>
4	0,8981	0,8917	<b>0,9719</b>	0,8889	0,8911	0,8632
Média	0,8988	0,8803	0,9696	0,8811	0,8824	0,8527
$\sigma$	$\pm 0,0076$	$\pm 0,0137$	$\pm 0,0024$	$\pm 0,0094$	$\pm 0,0145$	$\pm 0,0160$

Fonte: Do autor (2022).

Nota-se na Tabela 5.6 que os valores médios estão próximos dos valores do melhor classificador, demonstrando um desvio padrão baixo, evidenciando o bom desempenho da SVM-PSO. Refente a menor média, sendo de 0,8527, obtida no coeficiente Kappa, confirma-se um resultado satisfatório, pelo fato de apresentar um grau de concordância quase perfeito.

### 5.1.6 SVM-DE

No Apêndice B, além das matrizes de confusão dos classificadores, há também as tabelas de acertos e erros correspondentes a cada das Máquinas de vetores de Suporte SVM-DE. Essas tabelas foram utilizadas para o cálculo das principais métricas apresentadas na Tabela 5.7.

Tabela 5.7 – Métricas dos classificadores SVM-DE.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	<b>0,9333</b>	<b>0,8833</b>	<b>0,9724</b>	<b>0,8919</b>	<b>0,8914</b>	<b>0,8650</b>
2	0,8779	0,8595	0,9670	0,8684	0,8605	0,8282
3	0,8655	0,8444	0,9579	0,8378	0,8427	0,8025
4	0,8705	0,8456	0,9609	0,8462	0,8500	0,8113
Média	0,8868	0,8582	0,9645	0,8611	0,8612	0,8267
$\sigma$	$\pm 0,0272$	$\pm 0,0157$	$\pm 0,0056$	$\pm 0,0210$	$\pm 0,0186$	$\pm 0,0239$

Fonte: Do autor (2022).

Observa-se na Tabela 5.7 que o melhor classificador foi obtido na partição 1, em que seus valores estão destacados em negrito. A seguir, são listados os parâmetros obtidos pelo DE para o melhor classificador.

- *Outlier Fraction*: 0,3823;
- *Score Transform*: *symmetriclogit*;
- *Kernel function*: *rbf*;
- *Solver*: SMO;

Nota-se na Tabela 5.7 que os desvios padrão das métricas foram maiores em relação ao SVM-PSO, devido aos valores da partição 1 serem maiores em relação as demais, entretanto, são valores abaixo de 0,0239, os quais são aceitáveis. Em relação à menor média, apresentada no coeficiente Kappa, sendo de 0,8267 apresenta um grau de concordância quase perfeito.

### 5.1.7 SVM-GWO

No Apêndice B são demonstradas as matrizes de confusão obtidas para cada partição após o treinamento e teste, além das tabelas de acertos e erros dos classificadores. Estas tabelas foram utilizadas para se obter as principais métricas, as quais estão demonstradas na Tabela 5.8.

Tabela 5.8 – Métricas dos classificadores SVM-GWO.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,8914	0,8671	0,9666	0,8684	0,8629	0,8308
2	0,9000	0,8714	0,9673	0,8718	0,8716	0,8400
3	<b>0,9429</b>	<b>0,9528</b>	<b>0,9871</b>	<b>0,9460</b>	<b>0,9441</b>	<b>0,9308</b>
4	0,8731	0,8631	0,9662	0,8649	0,8623	0,8288
Média	0,9018	0,8886	0,9718	0,8878	0,8852	0,8576
$\sigma$	$\pm 0,0256$	$\pm 0,0372$	$\pm 0,0088$	$\pm 0,0337$	$\pm 0,0342$	$\pm 0,0425$

Fonte: Do autor (2022).

Observa-se na Tabela 5.8 que o melhor classificador foi obtido na partição 3, em que seus valores estão destacados em negrito. A seguir são listados os parâmetros obtidos pelo GWO para o melhor classificador.

- *Outlier Fraction*: 0,8725;

- *Score Transform*: logit;
- *Kernel function*: rbf;
- *Solver*: SMO;

Nota-se na Tabela 5.8 que o desvio padrão foi maior em relação a SVM-PSO e a SVM-DE, sendo de 0,0425. Isto ocorreu devido aos valores da partição 3 estarem maiores em relação aos demais, entretanto, por ser valores baixos o tornam aceitáveis. Em relação à menor média, apresentada no coeficiente Kappa, nota-se que seu valor foi de 0,8576, demonstrando um grau de concordância quase perfeito.

### 5.1.8 Resultados dos classificadores SVM

As matrizes de confusão dos classificadores SVM, apresentadas no Apêndice B, revelam um comportamento similar às matrizes dos classificadores MLP. É possível observar que as classes 1, 2 e 3, e as classes 4 e 5, apresentam maiores erros entre si, evidenciando a complexidade da tarefa de classificação das cargas.

Na Tabela 5.9 encontram-se os valores médios dos melhores classificadores. Ao analisá-los, nota-se que o classificador SVM-GWO apresentou o melhor resultado em relação aos demais. Ressalta-se que, mesmo a SVM-GWO apresentando o maior desvio padrão em relação aos demais, seus valores finais são satisfatórios, enfatizando ser o melhor classificador.

Tabela 5.9 – Métricas dos classificadores SVM.

Método	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
SVM-PSO	0,8988	0,8803	0,9696	0,8811	0,8824	0,8527
SVM-DE	0,8868	0,8582	0,9645	0,8611	0,8612	0,8267
SVM-GWO	<b>0,9018</b>	<b>0,8886</b>	<b>0,9718</b>	<b>0,8878</b>	<b>0,8852</b>	<b>0,8576</b>

Fonte: Do autor (2022).

### 5.1.9 Resultados dos classificadores MLP e PSO

Após analisados os resultados dos classificadores MLP e SVM, comparar-se-á os resultados escolhendo-se o melhor para o método de aprendizagem supervisionado. Na Tabela 5.10 são

apresentadas os valores médios dos melhores classificadores, os quais foram o MLP-GWO e a SVM-GWO, e os melhores valores estão destacados em negrito.

Analisando os resultados, observa-se que o MLP-GWO apresentou as melhores métricas, demonstrando ser o melhor classificador para a técnica de aprendizagem supervisionada para a classificação de cargas industriais.

Tabela 5.10 – Métricas dos melhores classificadores MLP e SVM.

Método	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
MLP-GWO	<b>0,9242</b>	<b>0,9156</b>	<b>0,9784</b>	<b>0,9139</b>	<b>0,9140</b>	<b>0,8926</b>
SVM-GWO	0,9018	0,8886	0,9718	0,8878	0,8852	0,8576

Fonte: Do autor (2022).

## 5.2 Aprendizagem não-supervisionado — Clusterização Fuzzy

Foram utilizados três métodos de clusterização Fuzzy, sendo K-Means (KM), o Fuzzy C-Means (FCM) e o Gustafson-Kessel (GK) para classificar cargas industriais. Esses métodos são uma abordagem de aprendizado não-supervisionado que agrupam conjuntos em classes com base em padrões, regularidades e correlações.

Para comparar os métodos, foram realizados os mesmos procedimentos para cada um deles, utilizando o método de validação k-fold ( $k = 4$ ) e aplicando os índices de validação Critério de Xie-Beni (*Xie-Beni criterion*, XB), Entropia de Classificação (*Classification Entropy*, CE), Índice de Partição (*Partition Index*, SC) e Índice de Dunn (*Dunn's Index*, DI). Em seguida, os clusters foram rotulados e os resultados dos métodos propostos foram apresentados.

### 5.2.1 K-Means

Os índices da partição 1 estão na Tabela 5.11. Os valores em negrito indicam a melhor quantidade de clusters. O índice CE não foi mostrado para o KM, pois é dependente dos graus de pertinência.

Ao avaliar os índices XB, SC e DI, percebe-se que o valor ideal para XB é aquele que minimiza a função, enquanto para SC e DI é o valor que maximiza suas funções. Como resultado, foi necessário utilizar diferentes quantidades de clusters para encontrar o melhor classificador: 5 clusters para SC, 17 clusters para XB e 24 clusters para DI.

Com base nesses valores, foram posicionados os clusters no espaço dos dados, rotulados e testados com os dados de teste para medir o desempenho do classificador. Essa abordagem permitiu avaliar diferentes números de clusters e selecionar aquele que proporcionou o melhor desempenho do modelo. No Apêndice C, é apresentada a matriz de confusão juntamente com os acertos e erros do melhor classificador obtido com 24 clusters. Na Tabela 5.12, são demonstradas as métricas de desempenho que serão posteriormente analisadas.

Após obtido o classificador na partição 1, iniciou-se a busca do classificador para a partição 2. Primeiramente, obteve-se a quantidade de clusters por meio dos índices de validação apresentados na Tabela 5.13, em que os melhores valores estão destacados em negrito. Nota-se que foram obtidos 20 clusters em XB e DI e 7 clusters em DI, deste modo, foram utilizados 7 e 20 clusters.

Tabela 5.11 – Valores dos índices de validação do método KM obtidos na partição 1.

Clusters	XB	SC	DI
5	1,8846	<b>0,1303</b>	0,1329
6	1,7505	0,1239	0,1662
7	1,8167	0,1230	0,1540
8	1,7925	0,1195	0,1639
9	1,7996	0,1017	0,1767
10	1,7772	0,0816	0,2007
11	1,8875	0,1094	0,1260
12	1,6928	0,0812	0,1752
13	1,6012	0,0903	0,1359
14	1,7797	0,0820	0,1543
15	1,5369	0,0760	0,1574
16	1,5931	0,0844	0,1735
17	<b>1,4640</b>	0,0991	0,1741
18	1,6132	0,0860	0,1528
19	1,5112	0,0829	0,1976
20	1,4668	0,0830	0,1425
21	1,6143	0,0948	0,2059
22	1,5903	0,0942	0,1826
23	1,5690	0,0670	0,1474
24	1,5026	0,0885	<b>0,2067</b>
25	1,5596	0,0751	0,1681

Fonte: Do autor (2022).

No Apêndice C, são apresentadas a matriz de confusão, os acertos e erros do melhor classificador obtido com 20 clusters. Na Tabela 5.14, são apresentadas as métricas de desempenho que serão posteriormente analisadas.

Tabela 5.12 – Acertos e erros do classificador do método KM obtidos na partição 1 com 24 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,7476	0,7381	0,9350	0,7368	0,7343	0,6692

Fonte: Do autor (2022).

Tabela 5.13 – Valores dos índices de validação do método KM obtidos na partição 2.

Clusters	XB	SC	DI
5	1,7380	0,1265	0,1390
6	1,7367	0,1278	0,1480
7	1,8188	<b>0,1357</b>	0,1292
8	1,8143	0,1155	0,1167
9	1,7814	0,1295	0,1477
10	1,8050	0,1126	0,1913
11	1,7853	0,1100	0,1479
12	1,7445	0,1003	0,1529
13	1,5023	0,0878	0,1937
14	1,6013	0,1119	0,2123
15	1,7146	0,1021	0,1601
16	1,5570	0,0965	0,1267
17	1,6782	0,0931	0,2100
18	1,6202	0,0933	0,1608
19	1,5698	0,0875	0,1397
20	<b>1,4866</b>	0,0828	<b>0,2151</b>
21	1,5046	0,0769	0,1485
22	1,5957	0,0870	0,1769
23	1,4963	0,0838	0,1744
24	1,6776	0,0889	0,1610
25	1,5029	0,0794	0,1380

Fonte: Do autor (2022).

Tabela 5.14 – Métricas de desempenho do método KM obtidas na partição 2 com 20 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
2	0,7400	0,7056	0,9273	0,7027	0,6917	0,6205

Fonte: Do autor (2022).

Na partição 3, os valores dos índices são apresentados na Tabela 5.15, em que os melhores valores estão destacados em negrito. Nota-se que foram obtidos 25 clusters em XB e DI e 7 clusters em SC, deste modo, foram escolhidos 7 e 25 clusters e iniciou-se o posicionamento e a rotulação dos clusters e aplicou-se os dados de testes visando obter as métricas de desempenho.

Tabela 5.15 – Valores dos índices de validação do método KM obtidos na partição 3.

Clusters	XB	SC	DI
5	1,7151	0,1253	0,1463
6	1,7360	<b>0,1700</b>	0,1779
7	1,7191	0,1283	0,1295
8	1,6204	0,1193	0,1713
9	1,6253	0,1138	0,1575
10	1,5853	0,1362	0,1277
11	1,6028	0,1086	0,1619
12	1,5986	0,0949	0,1697
13	1,8470	0,1068	0,1610
14	1,5531	0,0827	0,1980
15	1,5371	0,1060	0,1276
16	1,5701	0,0951	0,1978
17	1,5246	0,0935	0,1304
18	1,5212	0,1009	0,1747
19	1,6138	0,0844	0,1978
20	1,6589	0,0780	0,1340
21	1,5089	0,0960	0,1296
22	1,4968	0,0921	0,1389
23	1,4829	0,1048	0,1499
24	1,4796	0,0839	0,1389
25	<b>1,4606</b>	0,0726	<b>0,2080</b>

Fonte: Do autor (2022).

No Apêndice C, são apresentadas a matriz de confusão, os acertos e erros do melhor classificador obtido com 25 clusters. Na Tabela 5.16, são apresentadas as métricas de desempenho que serão posteriormente analisadas.

Tabela 5.16 – Métricas de desempenho do método KM obtidas na partição 2 com 25 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
3	0,7873	0,7698	0,9401	0,7632	0,7662	0,7071

Fonte: Do autor (2022).

Por fim, na partição 4 os valores dos índices de validação são apresentados na Tabela 5.17, em que os melhores valores estão destacados em negrito. Nota-se que a quantidade de clusters foi de 15 em XB, 5 em SC e 22 em DI. Assim, foram escolhidos 5, 15 e 22 clusters e iniciou-se o posicionamento e a rotulação dos cluster e logo em seguida foram aplicados os dados de teste.

No Apêndice C, são apresentadas a matriz de confusão, os acertos e erros do melhor classificador obtido com 22 clusters. Na Tabela 5.18, são apresentadas as métricas de desem-

penho que serão posteriormente analisadas. Os resultados obtidos nas diferentes partições são apresentados na Tabela 5.19, destacando-se os valores em negrito como os melhores. A partição que obteve o melhor classificador foi a terceira, utilizando 25 clusters. Os desvios padrão ficaram abaixo de 0,0308, o que é considerado satisfatório.

Quanto às médias, o coeficiente Kappa apresentou o menor valor, de 0,6640, indicando uma concordância boa de acordo com a classificação proposta por Landis e Koch (1977). Os resultados serão agrupados por método de clusterização e posteriormente comparados para determinar o melhor classificador.

Tabela 5.17 – Valores dos índices de validação do método KM obtidos na partição 4.

Clusters	XB	SC	DI
5	1,6782	<b>0,1813</b>	0,1415
6	1,6002	0,1424	0,1132
7	1,6391	0,1279	0,1266
8	1,6082	0,1220	0,2013
9	1,9168	0,1026	0,1531
10	1,6823	0,1186	0,1375
11	1,6391	0,1020	0,1972
12	1,5004	0,1336	0,1375
13	1,5510	0,1062	0,1889
14	1,5418	0,1112	0,1782
15	<b>1,4931</b>	0,0884	0,1610
16	1,6686	0,0910	0,1484
17	1,5298	0,0881	0,1536
18	1,5595	0,0928	0,1726
19	1,5682	0,0889	0,2291
20	1,5701	0,0881	0,1463
21	1,6039	0,1110	0,1639
22	1,5810	0,0843	<b>0,2771</b>
23	1,5349	0,0871	0,1585
24	1,5304	0,0883	0,1437
25	1,5598	0,0883	0,1905

Fonte: Do autor (2022).

Tabela 5.18 – Métricas de desempenho do método KM obtidas na partição 4 com 22 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
4	0,7488	0,7429	0,9338	0,7368	0,7191	0,6591

Fonte: Do autor (2022).

Tabela 5.19 – Métricas dos classificadores KM.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,7476	0,7381	0,9350	0,7368	0,7343	0,6692
2	0,7400	0,7056	0,9273	0,7027	0,6917	0,6205
3	<b>0,7873</b>	<b>0,7698</b>	<b>0,9401</b>	<b>0,7632</b>	<b>0,7662</b>	<b>0,7071</b>
4	0,7488	0,7429	0,9338	0,7368	0,7191	0,6591
Média	0,7559	0,7391	0,9340	0,7349	0,7278	0,6640
$\sigma$	$\pm 0,0184$	$\pm 0,0228$	$\pm 0,0046$	$\pm 0,0215$	$\pm 0,0269$	$\pm 0,0308$

Fonte: Do autor (2022).

### 5.2.2 Fuzzy C-Means

Na Tabela 5.20 são apresentados os valores dos índices de validação obtidos na partição 1, em que os melhores valores encontram-se em negrito. Observa-se que o índice de validação CE existe no método FCM, e seu melhor índice é aquele que maximiza sua função. Os números dos melhores clusters são de 19, 24, 9 e 21 para XB, SC, CE e DI respectivamente.

Após a rotulação dos clusters e a realização da classificação, foi obtida a matriz de confusão, bem como os acertos e erros do melhor classificador com 24 clusters, apresentados no Apêndice C. As métricas de desempenho são apresentadas na Tabela 5.21.

Após obtido o classificador na partição 1, iniciou-se a busca do classificador na partição 2. Na Tabela 5.22 são apresentados os valores dos índices de validação, em que os melhores valores encontram-se em negrito. Observa-se que as quantidades de clusters obtidos foram de 21, 22, 7 e 16 para XB, SC, CE e DI respectivamente.

No Apêndice C, é apresentada a matriz de confusão juntamente com os acertos e erros do melhor classificador obtido com 21 clusters. As métricas de desempenho são apresentadas na Tabela 5.23.

Na partição 3, os valores dos índices encontram-se na Tabela 5.24, em que os melhores valores estão destacados em negrito. Verifica-se que as melhores quantidades de clusters foram de 25 em XB, 24 em SC e DI e 11 em CE.

No Apêndice C, é apresentada a matriz de confusão juntamente com os acertos e erros do melhor classificador obtido com 24 clusters. As métricas de desempenho são apresentadas na Tabela 5.25.

Tabela 5.20 – Valores dos índices de validação do método FCM obtidos na partição 1.

Clusters	XB	SC	CE	DI
5	1,7562	0,6252	0,0581	0,2138
6	1,6695	0,5998	0,0507	0,2138
7	1,6310	0,5468	0,0460	0,2279
8	1,5585	0,5091	0,0608	0,2461
9	1,4724	0,5685	<b>0,0637</b>	0,2068
10	1,5164	0,4143	0,0372	0,2372
11	1,4149	0,5319	0,0473	0,1841
12	1,4697	0,5521	0,0513	0,2059
13	1,4697	0,4465	0,0283	0,2467
14	1,4180	0,4350	0,0279	0,1748
15	1,4424	0,4371	0,0264	0,1883
16	1,4314	0,3885	0,0246	0,1887
17	1,4020	0,3552	0,0418	0,1936
18	1,3773	0,3327	0,0174	0,2451
19	<b>1,3543</b>	0,3319	0,0249	0,2266
20	1,3969	0,3439	0,0286	0,1954
21	1,3625	0,2554	0,0325	<b>0,2531</b>
22	1,3747	0,2731	0,0183	0,2266
23	1,4767	0,2732	0,0205	0,2370
24	1,3836	<b>0,2539</b>	0,0309	0,2030
25	1,3755	0,2628	0,0182	0,2299

Fonte: Do autor (2022).

Tabela 5.21 – Métricas de desempenho do método FCM obtidas na partição 1 com 24 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,7167	0,7028	0,9273	0,7105	0,6960	0,6249

Fonte: Do autor (2022).

Por fim, os valores dos índices na partição 4 encontram-se na Tabela 5.26, em que os melhores valores estão destacados em negrito. Identifica-se que as melhores quantidades de clusters foram de 22, 23, 7 e 10 para XB, SC, CE e DI respectivamente.

A matriz de confusão, bem como os acertos e erros do melhor classificador obtido com 22 clusters, são apresentados no Apêndice C. As métricas de desempenho são demonstradas na Tabela 5.27.

Visando facilitar a análise dos resultados, as métricas obtidas são apresentadas na Tabela 5.28 para cada partição, em que os melhores valores estão destacados em negrito, observando-se que o melhor classificador foi obtido na partição 3 com 24 clusters.

Tabela 5.22 – Valores dos índices de validação do método FCM obtidos na partição 2.

Clusters	XB	SC	CE	DI
5	1,7489	0,6153	0,0604	0,2080
6	1,6111	0,6429	0,0721	0,1770
7	1,7486	0,6655	<b>0,0829</b>	0,1107
8	1,7173	0,6849	0,0671	0,1576
9	1,6502	0,5531	0,0565	0,1569
10	1,6002	0,5196	0,0507	0,1737
11	1,5310	0,4130	0,0563	0,1470
12	1,6099	0,5512	0,0467	0,2127
13	1,5463	0,3971	0,0521	0,1583
14	1,4784	0,4594	0,0607	0,1698
15	1,5556	0,4128	0,0275	0,2040
16	1,6904	0,3939	0,0281	<b>0,3111</b>
17	1,5201	0,4242	0,0354	0,2447
18	1,4860	0,3925	0,0389	0,1910
19	1,5750	0,3625	0,0335	0,1552
20	1,5363	0,3763	0,0289	0,2333
21	<b>1,4352</b>	0,3942	0,0279	0,1485
22	1,4918	<b>0,2214</b>	0,0259	0,2389
23	1,5197	0,2573	0,0427	0,2594
24	1,5848	0,3445	0,0294	0,2048
25	1,4855	0,3413	0,0311	0,1589

Fonte: Do autor (2022).

Tabela 5.23 – Métricas de desempenho do método FCM obtidas na partição 3 com 21 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
2	0,7843	0,7643	0,9404	0,7568	0,7571	0,6976

Fonte: Do autor (2022).

Outro ponto a evidenciar refere-se aos desvios padrão e a menor média. Em relação aos desvios padrão, nota-se que foram abaixo de 0,0404, sendo valores aceitáveis. Em relação à menor média, obtida no coeficiente Kappa, seu valor foi de 0,6923, correspondendo a uma concordância boa.

Tabela 5.24 – Valores dos índices de validação do método FCM obtidos na partição 3.

Clusters	XB	SC	CE	DI
5	1,6431	0,7093	0,0505	0,1300
6	1,6665	0,6341	0,0462	0,1300
7	1,5783	0,6593	0,0453	0,1300
8	1,6135	0,6052	0,0417	0,1774
9	1,7837	0,5875	0,0528	0,1777
10	1,6344	0,4455	0,0534	0,1489
11	1,5393	0,4199	<b>0,0595</b>	0,1806
12	1,5857	0,5620	0,0492	0,1687
13	1,4798	0,6600	0,0446	0,1699
14	1,5505	0,3838	0,0464	0,1952
15	1,5831	0,3448	0,0305	0,2398
16	1,4587	0,3367	0,0349	0,2062
17	1,4649	0,3131	0,0330	0,2234
18	1,4399	0,3858	0,0467	0,2634
19	1,4551	0,3557	0,0369	0,1646
20	1,4953	0,3460	0,0269	0,2091
21	1,3891	0,2674	0,0234	0,2628
22	1,3934	0,3299	0,0302	0,2370
23	1,4559	0,3817	0,0195	0,1692
24	1,4304	<b>0,2019</b>	0,0346	<b>0,3057</b>
25	<b>1,3588</b>	0,3048	0,0228	0,1975

Fonte: Do autor (2022).

Tabela 5.25 – Métricas de desempenho do método FCM obtidas na partição 3 com 24 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
3	0,8067	0,7964	0,9483	0,7895	0,7764	0,7267

Fonte: Do autor (2022).

### 5.2.3 Gustafson-Kessel

Na Tabela 5.29 são apresentados os valores dos índices de validação obtidos na partição 1, em que os melhores valores estão destacados em negrito. Nota-se que as melhores quantidades de clusters foram de 9, 23, 17 e 25 para XB, SC, CE e DI respectivamente.

A matriz de confusão, bem como os acertos e erros do melhor classificador obtido com 23 clusters, são apresentados no Apêndice C. As métricas de desempenho são demonstradas na Tabela 5.30.

Tabela 5.26 – Valores dos índices de validação do método FCM obtidos na partição 4.

Clusters	XB	SC	CE	DI
5	1,6788	0,6485	0,0556	0,1277
6	1,6505	0,6281	0,0437	0,1277
7	1,5869	0,6362	<b>0,0624</b>	0,1463
8	1,5706	0,6202	0,0543	0,2082
9	1,5053	0,5105	0,0597	0,1780
10	1,5403	0,4666	0,0475	<b>0,2397</b>
11	1,4884	0,4785	0,0607	0,1845
12	1,6397	0,4713	0,0586	0,0987
13	1,4447	0,3793	0,0530	0,2103
14	1,4425	0,3354	0,0465	0,2007
15	1,4078	0,3665	0,0566	0,1756
16	1,4929	0,3947	0,0389	0,1707
17	1,4427	0,3589	0,0409	0,1485
18	1,5016	0,3787	0,0462	0,2007
19	1,4394	0,3086	0,0313	0,2085
20	1,5227	0,3038	0,0303	0,2007
21	1,4535	0,3321	0,0299	0,1955
22	<b>1,3819</b>	0,3436	0,0286	0,2007
23	1,4646	<b>0,2563</b>	0,0309	0,2007
24	1,5576	0,2583	0,0352	0,1555
25	1,4385	0,2757	0,0291	0,2007

Fonte: Do autor (2022).

Tabela 5.27 – Métricas de desempenho do método FCM obtidas na partição 4 com 22 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
4	0,7984	0,7750	0,9468	0,7895	0,7712	0,7199

Fonte: Do autor (2022).

Tabela 5.28 – Métricas dos classificadores FCM.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,7167	0,7028	0,9273	0,7105	0,6960	0,6249
2	0,7843	0,7643	0,9404	0,7568	0,7571	0,6976
3	<b>0,8067</b>	<b>0,7964</b>	<b>0,9483</b>	<b>0,7895</b>	<b>0,7764</b>	<b>0,7267</b>
4	0,7984	0,7750	0,9468	<b>0,7895</b>	0,7712	0,7199
Média	0,7765	0,7596	0,9407	0,7616	0,7502	0,6923
$\sigma$	$\pm 0,0355$	$\pm 0,0348$	$\pm 0,0083$	$\pm 0,0323$	$\pm 0,0321$	$\pm 0,0404$

Fonte: Do autor (2022).

Tabela 5.29 – Valores dos índices de validação do método GK obtidos na partição 1.

Clusters	XB	SC	CE	DI
5	0,6398	0,4277	1,2420	0,1482
6	0,6386	0,2593	1,3319	0,1042
7	0,6419	0,2459	1,4173	0,1484
8	0,6310	0,2256	1,5149	0,1576
9	<b>0,6243</b>	0,2134	1,5203	0,1662
10	0,6706	0,1876	1,5138	0,1356
11	0,6783	0,1836	1,5655	0,1812
12	0,6853	0,1767	1,5831	0,1458
13	0,6896	0,1670	1,6189	0,1064
14	0,6796	0,1566	1,6261	0,1562
15	0,7077	0,1509	1,6366	0,0935
16	0,7084	0,1480	1,5981	0,1691
17	0,6993	0,1317	<b>1,6766</b>	0,1777
18	0,7518	0,1491	1,6436	0,1925
19	0,7547	0,1313	1,6135	0,1880
20	0,7239	0,1247	1,6714	0,1916
21	0,7397	0,1216	1,6039	0,1962
22	0,7411	0,1158	1,6211	0,2092
23	0,7543	<b>0,1067</b>	1,6215	0,1880
24	0,7753	0,1196	1,5919	0,1813
25	0,7534	0,1135	1,5856	<b>0,2448</b>

Fonte: Do autor (2022).

Tabela 5.30 – Métricas de desempenho do método GK obtidas na partição 1 com 23 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,8393	0,8472	0,9597	0,8378	0,8398	0,7992

Fonte: Do autor (2022).

A Tabela 5.31 apresenta os índices de validação obtidos na partição 2, destacando em negrito os melhores valores. As quantidades ideais de clusters foram 12, 24, 18 e 23 para os índices XB, SC, CE e DI, respectivamente.

A matriz de confusão, bem como os acertos e erros do melhor classificador obtido com 23 clusters, são apresentados no Apêndice C. As métricas de desempenho são demonstradas na Tabela 5.32.

Na partição 3, os valores dos índices de validação encontram-se na Tabela 5.34, em que os melhores valores estão destacados em negrito. Observa-se que as melhores quantidades de cluster foram 7 em XB, 25 em SC e 20 em CE e DI.

Tabela 5.31 – Valores dos índices de validação do método GK obtidos na partição 2.

Clusters	XB	SC	CE	DI
5	0,6767	0,2944	1,2359	0,0478
6	0,6425	0,2962	1,3443	0,0890
7	0,6222	0,2163	1,4295	0,1021
8	0,6319	0,2008	1,4720	0,1266
9	0,6264	0,1792	1,4978	0,1266
10	0,6367	0,1732	1,5118	0,1266
11	0,6335	0,1777	1,5876	0,1093
12	<b>0,6215</b>	0,1560	1,5849	0,1399
13	0,6335	0,1559	1,6322	0,1341
14	0,6526	0,1559	1,6027	0,1301
15	0,6703	0,1471	1,6443	0,1462
16	0,6820	0,1324	1,6413	0,1535
17	0,7177	0,1312	1,6371	0,1535
18	0,7297	0,1270	<b>1,6505</b>	0,1474
19	0,7339	0,1258	1,5882	0,1479
20	0,7805	0,1272	1,5955	0,1436
21	0,7645	0,1178	1,5964	0,0885
22	0,7691	0,1158	1,6121	0,1436
23	0,7978	0,1120	1,5352	<b>0,2111</b>
24	0,7492	<b>0,1039</b>	1,5513	0,0657
25	0,7814	0,1092	1,5636	0,1411

Fonte: Do autor (2022).

Tabela 5.32 – Métricas de desempenho do método GK obtidas na partição 2 com 23 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
2	0,8505	0,7921	0,9486	0,7949	0,7826	0,7345

Fonte: Do autor (2022).

No Apêndice C, é possível encontrar a matriz de confusão do melhor classificador obtido com 25 clusters, juntamente com os acertos e erros. As métricas de desempenho desse classificador estão apresentadas na Tabela 5.33.

Tabela 5.33 – Métricas de desempenho do método GK obtidas na partição 3 com 25 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
3	0,7533	0,7492	0,9337	0,7368	0,7322	0,6686

Fonte: Do autor (2022).

Na Tabela 5.36 estão os resultados dos índices de validação na partição 4, com os valores ótimos em negrito. Observa-se que a quantidade ideal de clusters foi de 20 em XB e CE, e 24 em SC e DI. No Apêndice C, encontra-se a matriz de confusão do melhor classificador com 24 clusters, juntamente com os acertos e erros. As métricas desse classificador estão na Tabela 5.35.

Tabela 5.34 – Valores dos índices de validação do método GK obtidos na partição 3.

Clusters	XB	SC	CE	DI
5	0,7115	0,3620	1,1622	0,1133
6	0,6749	0,2005	1,2740	0,1472
7	<b>0,6350</b>	0,1866	1,3901	0,1246
8	0,6440	0,1805	1,3976	0,1806
9	0,6530	0,1637	1,4386	0,1962
10	0,6589	0,1588	1,4466	0,1962
11	0,7003	0,1521	1,4502	0,1129
12	0,7196	0,1569	1,5245	0,1474
13	0,6863	0,1284	1,5307	0,1704
14	0,6379	0,1232	1,6139	0,1972
15	0,7220	0,1333	1,5507	0,1474
16	0,7017	0,1459	1,6057	0,1474
17	0,7301	0,1241	1,5590	0,1474
18	0,6714	0,1159	1,6111	0,1441
19	0,6686	0,1157	1,6472	0,1933
20	0,6862	0,1089	<b>1,6912</b>	<b>0,2053</b>
21	0,6575	0,1116	1,6364	0,1645
22	0,7056	0,1114	1,6210	0,1675
23	0,7026	0,1062	1,6528	0,1614
24	0,6787	0,1093	1,6209	0,1939
25	0,6905	<b>0,1050</b>	1,6274	0,1939

Fonte: Do autor (2022).

Tabela 5.35 – Métricas de desempenho do método GK obtidas na partição 4 com 24 clusters.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
4	0,7670	0,7500	0,9400	0,7568	0,7433	0,6840

Fonte: Do autor (2022).

Para facilitar a análise dos resultados, a Tabela 5.37 apresenta as métricas obtidas para cada partição, sendo os melhores valores destacados em negrito. Observa-se que, em geral, as métricas na partição 1 foram superiores em relação aos demais classificadores, com exceção da precisão, que apresentou um valor superior na partição 2. No entanto, as diferenças entre os valores são relativamente baixas quando comparadas com as demais métricas.

Tabela 5.36 – Valores dos índices de validação do método GK obtidos na partição 4.

Clusters	XB	SC	CE	DI
5	0,6923	0,3255	1,2092	0,1341
6	0,7096	0,2562	1,3072	0,1685
7	0,7294	0,1991	1,3727	0,1993
8	0,7121	0,1973	1,4229	0,1643
9	0,7051	0,1781	1,4898	0,1266
10	0,7051	0,1743	1,5069	0,1266
11	0,6866	0,1517	1,5776	0,1422
12	0,6865	0,1475	1,6021	0,1292
13	0,7277	0,1384	1,5693	0,1239
14	0,7202	0,1336	1,5561	0,1622
15	0,6803	0,1216	1,6128	0,1811
16	0,6956	0,1270	1,5791	0,1761
17	0,6876	0,1267	1,6953	0,1025
18	0,7095	0,1194	1,6408	0,1735
19	0,6734	0,1156	1,6886	0,2048
20	<b>0,6628</b>	0,1135	<b>1,7106</b>	0,1456
21	0,7075	0,1127	1,6035	0,1604
22	0,7133	0,1201	1,6434	0,2064
23	0,6956	0,1099	1,6168	0,1735
24	0,7533	<b>0,1037</b>	1,5840	<b>0,2090</b>
25	0,7281	0,1070	1,5812	0,1735

Fonte: Do autor (2022).

Tabela 5.37 – Métricas dos classificadores GK.

Partição	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
1	0,8393	<b>0,8472</b>	<b>0,9597</b>	<b>0,8378</b>	<b>0,8398</b>	<b>0,7992</b>
2	<b>0,8505</b>	0,7921	0,9486	0,7949	0,7826	0,7345
3	0,7533	0,7492	0,9337	0,7368	0,7322	0,6686
4	0,7670	0,7500	0,9400	0,7568	0,7433	0,6840
Média	0,8025	0,7846	0,9455	0,7816	0,7745	0,7216
$\sigma$	$\pm 0,0428$	$\pm 0,0401$	$\pm 0,0098$	$\pm 0,0386$	$\pm 0,0421$	$\pm 0,0510$

Fonte: Do autor (2022).

Assim, o classificador obtido na partição 1, com a utilização de 23 clusters, apresentou o melhor desempenho. Cabe ressaltar que, apesar do coeficiente Kappa apresentar o menor valor médio, de 0,7216, ainda assim indica uma concordância boa.

### 5.2.4 Resultados dos métodos KM, FCM e GK

Os melhores resultados dos métodos de clusterização encontram-se na Tabela 5.38, em que os melhores valores estão destacados em negrito. Observa-se que o método GK mostrou ser o melhor classificador para as cargas industriais, pelo fato de apresentar as melhores métricas em relação aos demais. Ressalta-se que a melhor quantidade de clusters foi de 23, demonstrando que 25 clusters são suficientes para análise dos índices de validação para se obter a melhor quantidade.

Tabela 5.38 – Métricas dos classificadores KM, GK e FCM.

Método	Precisão	Sensibilidade	Especificidade	ACC	F1-score	Kappa	Clusters
KM	0,7559	0,7391	0,9340	0,7349	0,7278	0,6640	25
FCM	0,7765	0,7596	0,9407	0,7616	0,7502	0,6923	24
GK	<b>0,8025</b>	<b>0,7846</b>	<b>0,9455</b>	<b>0,7816</b>	<b>0,7745</b>	<b>0,7216</b>	<b>24</b>

Fonte: Do autor (2022).

### 5.2.5 Resultados das técnicas de aprendizado supervisionado e não-supervisionado.

Após encontrar os melhores classificadores para as técnicas de aprendizagem supervisionado e não-supervisionado, foi realizada a comparação dos resultados. Na Tabela 5.39 encontram-se os valores obtidos dos melhores classificadores para as duas técnicas, em que os melhores estão destacados em negrito.

Tabela 5.39 – Métricas dos melhores classificadores das técnicas de aprendizado supervisionado e não supervisionado.

Método	Precisão	Sensibilidade	Especificidade	Acurácia	F1-score	Kappa
MLP-GWO	<b>0,9242</b>	<b>0,9156</b>	<b>0,9784</b>	<b>0,9139</b>	<b>0,9140</b>	<b>0,8926</b>
GK	0,8025	0,7846	0,9455	0,7816	0,7745	0,7216

Fonte: Do autor (2022).

Observa-se que o classificador MLP-GWO apresentou as melhores métricas em relação ao método GK. Isso se deve principalmente à técnica de aprendizado utilizada. Na técnica de aprendizado supervisionado, o algoritmo aprende com resultados previamente definidos e ajusta seus pesos com base no erro calculado na saída do classificador. Por outro lado, na técnica de

aprendizado não supervisionado, o algoritmo não possui conhecimento prévio dos resultados e deve associar os padrões dos dados de entrada fornecidos.

No entanto, é importante destacar que, apesar das métricas melhores apresentadas pela técnica supervisionada, o custo computacional para realizar o treinamento é maior em comparação aos métodos propostos que utilizam a técnica não supervisionada.

## 6 CONCLUSÃO

Neste trabalho, foram desenvolvidos classificadores de cargas industriais visando classificar as cargas que estavam em processo de fabricação do produto enquanto as demais estavam energizadas, mas não estavam produzindo. Para essa finalidade, foram implementados os classificadores Perceptron de Multicamadas (*MultiLayer Perceptron*, MLP) e a Máquina de Vetores de Suporte (*Support Vectors Machine*, SVM) aplicando-se a técnica de aprendizagem supervisionado e os métodos de clusterização K-Means (KM), Fuzzy C-Means (FCM), Gustafson-Kessel (GK) aplicando-se a técnica de aprendizagem não-supervisionado.

Devido à complexidade de definir os parâmetros essenciais do MLP e da SVM, foram utilizadas as meta-heurísticas Otimização de Enxame de Partículas (*Particle Swarm Optimization*, PSO), Evolução Diferencial (*Differential Evolution*, DE) e Otimizador Lobo Cinzento (*Grey Wolf Optimizer*, GWO) a fim de maximizar os resultados. Como resultado, o MLP-GWO foi o classificador que apresentou as melhores métricas de desempenho, as quais destacaram-se a precisão com  $0,9242 \pm 0,0135$ , acurácia com  $0,9139 \pm 0,0113$ , F1-score com  $0,9140 \pm 0,0112$  e o coeficiente Kappa com  $0,8926 \pm 0,0139$ . Observa-se que os resultados são satisfatórios, destacando o coeficiente Kappa que apresentou um grau de concordância quase perfeito.

Com relação aos métodos de clusterização, foi aplicado os índices de validação Critério de Xie-Beni (XB), Classificação Entropia (CE), Índice de Partição (IP) e o Índice de Dunn (ID) visando obter a melhor quantidade de clusters, sendo 25, 24 e 23 para os métodos KM, FCM e GK respectivamente. Como resultado, o método GK foi o classificador que apresentou as melhores métricas, as quais destacam-se a precisão com  $0,8025 \pm 0,0428$ , acurácia com  $0,7816 \pm 0,0386$ , F1-score com  $0,7745 \pm 0,0421$  e coeficiente Kappa com  $0,7216 \pm 0,0510$ . Destaca-se que o coeficiente Kappa apresentou uma concordância boa.

Com base nos dados analisados, é possível concluir que o MLP-GWO é adequado para ser utilizado na linha de produção de amortecedores da indústria em questão, com o objetivo de classificar cargas industriais, devido aos resultados significativos obtidos em suas métricas de desempenho. Embora as métricas de desempenho do classificador GK sejam inferiores às do MLP-GWO, é importante ressaltar que o custo computacional do GK na fase do projeto é menor em relação ao MLP-GWO. Essa informação é relevante para situações em que é necessário realizar retreinamentos frequentes nos classificadores em tempo real, pois a resposta do classificador GK será mais rápida do que a do MLP-GWO.

Ao utilizar a classificação de cargas industriais na linha de produção, será possível gerenciá-la de maneira mais eficiente. Os classificadores obtidos permitirão identificar os equipamentos ativos na produção, ou seja, aqueles que estão fabricando o produto. Com isso, será possível obter informações sobre o consumo de energia de cada carga, bem como a frequência com que cada carga é acionada diariamente.

As informações obtidas por meio da classificação de cargas industriais serão de grande utilidade para estimar o consumo diário de cada carga e identificar possíveis problemas, como sobrecarga de produção ou erros operacionais. Com a frequência de acionamento dos equipamentos, será possível identificar se um equipamento está realizando retrabalho durante a produção e quais são as causas, pois ao identificar o problema será evitado desgaste excessivo do equipamento e desperdício de energia. Esses dados são fundamentais para um gerenciamento eficiente da linha de produção, permitindo aumentar a produtividade e reduzir custos desnecessários.

Em resumo, a aplicação da classificação de cargas industriais na linha de produção permitirá uma melhor gestão dos recursos, reduzindo o consumo de energia desnecessário e aumentando a eficiência operacional da empresa.

Com essas informações, será possível realizar treinamentos de capacitação aos operadores, a fim de minimizar paradas não programadas na linha de produção, que podem resultar em prejuízos financeiros para a empresa. Em resumo, a classificação de cargas industriais permitirá um gerenciamento mais eficiente da linha de produção, aumentando a produtividade e reduzindo custos desnecessários.

Com base nos fatos apresentados, a classificação das cargas na linha de produção será uma ferramenta fundamental para o planejamento, a conscientização e o conhecimento do consumo diário de energia, bem como dos erros operacionais que podem resultar em prejuízos financeiros para a empresa. Isso permitirá um gerenciamento mais eficiente da linha de produção, garantindo uma maior produtividade e reduzindo custos desnecessários. Em resumo, a classificação das cargas será uma ferramenta essencial para a otimização do processo produtivo da empresa.

## **6.1 Propostas de continuidade**

Espera-se que a discussão aqui estabelecida seja uma contribuição para que novas pesquisas e tecnologias referente ao gerenciamento de uma linha de produção sejam fomentadas, por

ser por meio delas, que novas alternativas serão criadas, corroborando-se, assim, com o Planeta e melhorando a produtividade industrial.

Como propostas de continuidade estão:

- Construção de uma nova base de dados com acionamento de duas ou mais cargas simultaneamente;
- Analisar o 10° e o 52° harmônico obtido na seleção de características;
- Fazer comparação do custo computacional entre as técnicas de treinamento supervisionado e não-supervisionado;
- Realizar testes estatísticos das médias obtidas;
- Embarcar o sistema para patentear um equipamento de gerenciamento de linha de produção;
- Implementar um software que emita relatórios diariamente com o auxílio do sistema embarcado;
- Implementar as meta-heurísticas PSO, DE e GWO nos métodos de clusterização.

## REFERÊNCIAS

- ABRAHAM, A. Artificial neural networks. In: HANDBOOK. **Measuring System Design**. John Wiley Sons, Ltd, 2005. v. 17, cap. 129. ISBN 9780471497394. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/0471497398.mm421>>.
- ABUBAKAR, I. et al. Application of load monitoring in appliances' energy management – a review. **Renewable and Sustainable Energy Reviews**, v. 67, p. 235–245, 2017. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S136403211630555X>>.
- AZIZI, E.; BEHESHTI, M. T. H.; BOLOUKI, S. Event matching classification method for non-intrusive load monitoring. **Sustainability**, v. 13, n. 2, 2021. ISSN 2071-1050. Disponível em: <<https://www.mdpi.com/2071-1050/13/2/693>>.
- BAETS, L. D. et al. Vi-based appliance classification using aggregated power consumption data. In: 2018 IEEE INTERNATIONAL CONFERENCE. **Smart Computing (SMARTCOMP)**. Taormina, Italy, 2018. p. 179–186.
- BALASKO, B.; ABONYI, J.; FEIL, B. Fuzzy clustering and data analysis toolbox. **Department of Process Engineering, University of Veszprem, Veszprem**, 2005.
- BENSAID, A. et al. Validity-guided (re)clustering with applications to image segmentation. **IEEE Transactions on Fuzzy Systems**, v. 4, n. 2, p. 112–123, 1996.
- BEZDEK, J. C.; EHRLICH, R.; FULL, W. Fcm: The fuzzy c-means clustering algorithm. **Computers & Geosciences**, v. 10, n. 2, p. 191–203, 1984. ISSN 0098-3004. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0098300484900207>>.
- BISHOP, C. **Neural networks for pattern recognition**. USA: Oxford University Press, Inc., 1995.
- BONFIGLI, R. et al. Denoising autoencoders for non-intrusive load monitoring: Improvements and comparative evaluation. **Energy and Buildings**, v. 158, p. 1461–1474, 2018. ISSN 0378-7788. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378778817314457>>.
- BOUHOUBAS, A. S. et al. Load signature formulation for non-intrusive load monitoring based on current measurements. **Energies**, v. 10, n. 4, 2017. ISSN 1996-1073. Disponível em: <<https://www.mdpi.com/1996-1073/10/4/538>>.
- BRASIL, M. d. M. e. E.; EPE. **Plano Nacional de Energia 2050**. Brasília, 2020. Disponível em: <<https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-227/topico-563/Relatorio%20Final%20do%20PNE%202050.pdf>>. Acesso em: 19 mar. 2022.
- C, S.; KUMAR, M.; K, I. Design and implementation of non-intrusive load monitoring using machine learning algorithm for appliance monitoring. In: 2019 IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT TECHNIQUES IN CONTROL. **Optimization and Signal Processing (INCOS)**. Tamilnadu, India, 2019. p. 1–6.
- CHANG, C.-C.; LIN, C.-J. Libsvm: A library for support vector machines. Association for Computing Machinery, New York, NY, USA, v. 2, n. 3, 2011. ISSN 2157-6904. Disponível em: <<https://doi.org/10.1145/1961189.1961199>>.

CHOLLET, F. **Deep Learning with Python**. Shelter Island, NY: Manning Publications Co., 2018.

CLERC, M.; KENNEDY, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 1, p. 58–73, 2002.

COHEN, J. A coefficient of agreement for nominal scales. **Educational and Psychological Measurement**, v. 20, n. 1, p. 37–46, 1960. Disponível em: <<https://doi.org/10.1177/001316446002000104>>.

COMINOLA, A. et al. A hybrid signature-based iterative disaggregation algorithm for non-intrusive load monitoring. **Applied Energy**, v. 185, p. 331–344, 2017. ISSN 0306-2619. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S030626191631488X>>.

CORRÊA, S. d. J. d. C.; CASTRO, A. R. G. Automated non-intrusive load monitoring system using stacked neural networks and numerical integration. **IEEE Access**, v. 8, p. 210566–210581, 2020.

CUÑADO, J. R.; LINSANGAN, N. B. A supervised learning approach to appliance classification based on power consumption traces analysis. **IOP Conference Series: Materials Science and Engineering**, IOP Publishing, v. 517, p. 012011, apr 2019. Disponível em: <<https://doi.org/10.1088/1757-899x/517/1/012011>>.

De Baets, L. et al. Detection of unidentified appliances in non-intrusive load monitoring using siamese neural networks. **International Journal of Electrical Power & Energy Systems**, v. 104, p. 645–653, 2019. ISSN 0142-0615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0142061517331253>>.

DU, S. et al. Multi-pattern data mining and recognition of primary electric appliances from single non-intrusive load monitoring data. **Energies**, v. 12, n. 6, 2019. ISSN 1996-1073. Disponível em: <<https://www.mdpi.com/1996-1073/12/6/992>>.

EGRIOGLU, E. et al. Fuzzy time series forecasting method based on gustafson–kessel fuzzy clustering. **Expert Systems with Applications**, v. 38, n. 8, p. 10355–10357, 2011. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417411002582>>.

FANG, Z. et al. Nonintrusive appliance identification with appliance-specific networks. **IEEE Transactions on Industry Applications**, v. 56, n. 4, p. 3443–3452, 2020.

GARCIA, P. et al. Non intrusive load monitoring in chaotic switched networks. In: 2018 IEEE. **Third Ecuador Technical Chapters Meeting (ETCM)**. Cuenca, Ecuador, 2018. p. 1–5.

GATH, I.; GEVA, A. Unsupervised optimal fuzzy clustering. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 11, n. 7, p. 773–780, 1989.

GHOSH, A. K.; CHAKRABORTY, S. A comparative study on data normalization techniques for feature extraction in human activity recognition using wearable sensor data. **Journal of Ambient Intelligence and Humanized Computing**, Springer, v. 10, n. 4, p. 1419–1435, 2019.

GOMES, E.; PEREIRA, L. Pb-nilm: Pinball guided deep non-intrusive load monitoring. **IEEE Access**, v. 8, p. 48386–48398, 2020.

GRANDINI, M.; BAGLI, E.; VISANI, G. **Metrics for Multi-Class Classification: an Overview**. arXiv, 2020. Disponível em: <<https://arxiv.org/abs/2008.05756>>.

GUPTA, S.; DEEP, K. A novel random walk grey wolf optimizer. **Swarm and Evolutionary Computation**, v. 44, p. 101–112, 2019. ISSN 2210-6502. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2210650217305333>>.

GUSTAFSON, D. E.; KESSEL, W. C. Fuzzy clustering with a fuzzy covariance matrix. In: 1978 IEEE CONFERENCE. **Decision and Control including the 17th Symposium on Adaptive Processes**. San Diego, CA, USA, 1978. p. 761–766.

HART, G. Nonintrusive appliance load monitoring. **Proceedings of the IEEE**, v. 80, n. 12, p. 1870–1891, 1992.

HAYKIN, S. S. **Neural networks and learning machines**. Third. Upper Saddle River, NJ: Pearson Education, 2009.

HE, K. et al. Non-intrusive load disaggregation using graph signal processing. **IEEE Transactions on Smart Grid**, v. 9, n. 3, p. 1739–1747, 2018.

HUA, D. et al. Simultaneous disaggregation of multiple appliances based on non-intrusive load monitoring. **Electric Power Systems Research**, v. 193, p. 106887, 2021. ISSN 0378-7796. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0378779620306854>>.

JARAMILLO, A. F. M. et al. Non-intrusive load monitoring algorithm for pv identification in the residential sector. In: 2020 31ST IRISH. **Signals and Systems Conference (ISSC)**. Letterkenny, Ireland, 2020. p. 1–6.

JIA, D. et al. Non-intrusive load identification using reconstructed voltage–current images. **IEEE Access**, v. 9, p. 77349–77358, 2021.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: TEST. **Proceedings of ICNN'95 - International Conference on Neural Networks**. Perth, WA, Australia, 1995. v. 4, p. 1942–1948 vol.4.

KHOKHAR, S. et al. A comprehensive overview on signal processing and artificial intelligence techniques applications in classification of power quality disturbances. **Renewable and Sustainable Energy Reviews**, v. 51, p. 1650–1663, 2015. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1364032115007157>>.

KIM, H. et al. A systematic review of the smart energy conservation system: From smart homes to sustainable smart cities. **Renewable and Sustainable Energy Reviews**, v. 140, p. 110755, 2021. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1364032121000502>>.

KYORITSU, E. I. **KEW 6310**. 1992. Disponível em: <[https://www.kew-ltd.co.jp/files/en/manual/6310\\_IM\\_92-1880B\\_E\\_L.pdf](https://www.kew-ltd.co.jp/files/en/manual/6310_IM_92-1880B_E_L.pdf)>.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. **Biometrics**, [Wiley, International Biometric Society], v. 33, n. 1, p. 159–174, 1977. ISSN 0006341X, 15410420. Disponível em: <<http://www.jstor.org/stable/2529310>>.

LEITE, D. et al. Evolving fuzzy granular modeling from nonstationary fuzzy data streams. **Evolving Systems**, v. 3, 2012. ISSN 1868-6486. Disponível em: <<https://doi.org/10.1007/s12530-012-9050-9>>.

LIMA, R. B. d. **Classificador não invasivo de cargas elétricas residenciais com acionamento simultâneo**. Dissertação (Mestrado) — Universidade Federal de Lavras, <http://repositorio.ufla.br/jspui/handle/1/31740>, 2018. Acesso em: 03 jun. 2021.

MIRJALILI, S.; MIRJALILI, S. M.; LEWIS, A. Grey wolf optimizer. **Advances in Engineering Software**, v. 69, p. 46–61, 2014. ISSN 0965-9978. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0965997813001853>>.

MOHANTY, S. P.; CHOPPALI, U.; KOUIGIANOS, E. Everything you wanted to know about smart cities: The internet of things is the backbone. **IEEE Consumer Electronics Magazine**, v. 5, n. 3, p. 60–70, 2016.

MONTEIRO, R. et al. Non-intrusive load monitoring using artificial intelligence classifiers: Performance analysis of machine learning techniques. **Electric Power Systems Research**, v. 198, p. 107347, 2021. ISSN 0378-7796. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S037877962100328X>>.

MORAVEJ, Z.; PAZOKI, M.; ABDOOS, A. A. **Application of Signal Processing in Power Quality Monitoring**. Croácia: Power Quality Monitoring, Analysis and Enhancement, 2011. ISBN 978-953-307-330-9.

MOTA, V. C.; DAMASCENO, F. A.; LEITE, D. F. Fuzzy clustering and fuzzy validity measures for knowledge discovery and decision making in agricultural engineering. **Computers and Electronics in Agriculture**, v. 150, p. 118–124, 2018. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169918300632>>.

MÜLLER, A. C.; KLINKENBERG, R. **An Introduction to Machine Learning**. Cham: Springer International Publishing, 2017.

MURO, C. et al. Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations. **Behavioural Processes**, v. 88, n. 3, p. 192–197, 2011. ISSN 0376-6357. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0376635711001884>>.

NGUYEN, T. K. et al. Development of a real-time non-intrusive appliance load monitoring system: An application level model. **International Journal of Electrical Power & Energy Systems**, v. 90, p. 168–180, 2017. ISSN 0142-0615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0142061516311437>>.

PERERA, F.; NADEAU, K. Climate change, fossil-fuel pollution, and children's health. **New England Journal of Medicine**, Mass Medical Soc, v. 386, n. 24, p. 2303–2314, 2022.

PISCITELLI, M. S.; BRANDI, S.; CAPOZZOLI, A. Recognition and classification of typical load profiles in buildings with non-intrusive learning approach. **Applied Energy**, v. 255, p. 113727, 2019. ISSN 0306-2619. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S030626191931414X>>.

PRICE, K. Differential evolution vs. the functions of the 2/sup nd/ iceo. In: 1997 IEEE INTERNATIONAL CONFERENCE. **Evolutionary Computation (ICEC '97)**. Indianapolis, IN, USA, 1997. p. 153–157.

RAIKER, G. A. et al. Approach to non-intrusive load monitoring using factorial hidden markov model. In: 2018 IEEE 13TH INTERNATIONAL CONFERENCE. **Industrial and Information Systems (ICIIS)**. Rupnagar, India, 2018. p. 381–386.

REDDY, R.; GARG, V.; PUDI, V. A feature fusion technique for improved non-intrusive load monitoring. **Energy Informatics**, v. 3, 2020. ISSN 2520-8942. Disponível em: <<https://doi.org/10.1186/s42162-020-00112-w>>.

RIDI, A.; GISLER, C.; HENNEBERT, J. A survey on intrusive load monitoring for appliance recognition. In: 2014 22ND INTERNATIONAL CONFERENCE. **Pattern Recognition**. Stockholm, Sweden, 2014. p. 3702–3707.

SALEM, H.; SAYED-MOUCHAWEH, M.; TAGINA, M. Unsupervised bayesian non parametric approach for non-intrusive load monitoring based on time of usage. **Neurocomputing**, v. 435, p. 239–252, 2021. ISSN 0925-2312. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0925231220320087>>.

SCHIRMER, P. A.; MPORAS, I.; PARASKEVAS, M. Energy disaggregation using elastic matching algorithms. **Entropy**, v. 22, n. 1, 2020. ISSN 1099-4300. Disponível em: <<https://www.mdpi.com/1099-4300/22/1/71>>.

SEEG. **Emissões do Brasil têm maior alta em 19 anos**. 2022. Disponível em: <<https://seeg.eco.br/imprensa>>.

SHARMA, L.; YADAV, D. K.; SINGH, A. Fisher's linear discriminant ratio based threshold for moving human detection in thermal video. **Infrared Physics Technology**, v. 78, p. 118–128, 2016. ISSN 1350-4495. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1350449516301621>>.

TABATABAEI, S. M.; DICK, S.; XU, W. Toward non-intrusive load monitoring via multi-label classification. **IEEE Transactions on Smart Grid**, v. 8, n. 1, p. 26–40, 2017.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition, Fourth Edition**. 4th. ed. USA: Academic Press, Inc., 2008. ISBN 1597492728.

VAPNIK, V. N. Constructing learning algorithms. In: BOOK. **The Nature of Statistical Learning Theory**. New York, NY: Springer New York, 1995. p. 119–166. ISBN 978-1-4757-2440-0. Disponível em: <[https://doi.org/10.1007/978-1-4757-2440-0\\_6](https://doi.org/10.1007/978-1-4757-2440-0_6)>.

VÖLKER, B. et al. A framework to generate and label datasets for non-intrusive load monitoring. **Energies**, v. 14, n. 1, 2021. ISSN 1996-1073. Disponível em: <<https://www.mdpi.com/1996-1073/14/1/75>>.

WANG, S. et al. A feature selection method based on improved fisher's discriminant ratio for text sentiment classification. **Expert Systems with Applications**, v. 38, n. 7, p. 8696–8702, 2011. ISSN 0957-4174. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957417411000972>>.

WELIKALA, S. et al. Implementation of a robust real-time non-intrusive load monitoring solution. **Applied Energy**, v. 238, p. 1519–1529, 2019. ISSN 0306-2619. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0306261919301849>>.

XIE, X. L.; BENI, G. A validity measure for fuzzy clustering. **IEEE Transactions on pattern analysis and machine intelligence**, v. 13, n. 8, p. 841–847, 1991.

XU, F. et al. Classifier economics of semi-intrusive load monitoring. **International Journal of Electrical Power & Energy Systems**, v. 103, p. 224–232, 2018. ISSN 0142-0615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0142061518305908>>.

YAN, X.; JIA, M. A novel optimized svm classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing. **Neurocomputing**, v. 313, p. 47–64, 2018. ISSN 0925-2312. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S092523121830523X>>.

ZADEH, L. Fuzzy logic. **Computer**, v. 21, n. 4, p. 83–93, 1988.

ZADEH, L. A. **Fuzzy sets**. United States: Information and control, 1965. 338–353 p.

ZHAO, H. et al. Research on a learning rate with energy index in deep learning. **Neural Networks**, v. 110, p. 225–231, 2019. ISSN 0893-6080. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S089360801830340X>>.

ZHU, Z. et al. A novel approach for event detection in non-intrusive load monitoring. In: 2017 IEEE CONFERENCE. **Energy Internet and Energy System Integration (EI2)**. Beijing, China, 2017. p. 1–5.

**APÊNDICE A – Código-fonte MLP-GWO**

```
clear all
close all
clc

%% Dados
load ../data
dataNorm.X = data(:,1:end-1);
auxData = clust_normalize(dataNorm,'var');
data = [auxData.X data(:,end)];
data = shuffle(data);

%% Kfold
kfold=4;
ind= crossvalind('Kfold',data(:,end),kfold);

%% RNA parametros
epochs=1000;
max_fail=200;
performFcn = 'mse';
transferFcn1 = 'tansig';
transferFcn2 = 'tansig';
transferFcn3 = 'tansig';
min_grad = 1e-6;
trainFcn = 'traincgp';

%% GWO
Max_iter=25;
SearchAgents_no=100;
lb=[1 1 0.001];
ub=[30 30 0.95];
dim=3;

%% Kfold
```

```

for k = 1:kfold
    test = (ind == k);
    treino = ~test;

    % Dados de treino e teste
    Test_data = data(test,:);
    Train_data = data(treino,:);

    % Conversao dos dados de saida de decimal para binario -1 e 1
    Test_data_yd=decimal2binary(Test_data);
    Train_data_yd=decimal2binary(Train_data);

    % Remocao da coluna com os dados de saida
    Test_data=Test_data(:,1:end-1);
    Train_data=Train_data(:,1:end-1);

    % Inicializacao de beta, beta, delta_pos
    Alpha_pos = zeros(1,dim);
    Alpha_score(1:6) = -inf;
    Beta_pos = zeros(1,dim);
    Beta_score(1:6) = -inf;
    Delta_pos = zeros(1,dim);
    Delta_score(1:6) = -inf;

    % Inicializacao das posicoes dos agentes
    Positions=initialization(SearchAgents_no,dim,ub,lb);
    Convergence_curve=zeros(1,Max_iter);

    inter=1;
    acc=0;

    while inter <= Max_iter
        for i=1:size(Positions,1)

```

```

% Devolve os agentes de busca que ultrapassam
% os limites do espaço de busca
Flag4ub=Positions(i,:)>ub;
Flag4lb=Positions(i,:)<lb;
Positions(i,:) = (Positions(i,:).* ...
    (~(Flag4ub+Flag4lb)))+ub.*Flag4ub+lb.*Flag4lb;

% Camada escondida com os valores arredondados
Positions(i,1:2)=round(Positions(i,1:2));
hiddenLayer = Positions(i,1:2);
net = feedforwardnet(hiddenLayer);
net = configure(net,Test_data',Test_data_yd');

% Ajusta parametros para treinamento:
net.trainFcn = trainFcn;
net.performFcn = performFcn;
net.trainParam.epochs = epochs;
net.trainParam.lr = Positions(i,3);
net.trainParam.goal = 0;
net.trainParam.showWindow = false;
net.layers{1}.transferFcn = transferFcn1;
net.layers{2}.transferFcn = transferFcn2;
net.layers{3}.transferFcn = transferFcn3;
net.trainParam.min_grad = min_grad;
net.trainParam.max_fail = max_fail;

% Inicializa a rede neural com valores de pesos e bias
    aleatorios
net = init(net);

% Treino da rede neural
[net, tr] = train(net,Train_data',Train_data_yd');

% Simulacao da rede neural

```

```
Ysaida = sim(net,Test_data');

Ysaida(Ysaida <= 0) = -1;
Ysaida(Ysaida > 0) = 1;

% Saida convertida em decimal
decimalTest{k} = binary2decimal(Test_data_yd);
decimalYsaida{k} = binary2decimal(Ysaida');

% Metricas de desempenho
C = confusionmat(decimalTest{k},decimalYsaida{k});
stats = statsOfMeasure(C, 0);
metrics = stats.metrics;
precision = metrics(1);
sensitivity = metrics(2);
specificity = metrics(3);
accuracy = metrics(4);
FMeasure = metrics(5);
kappa = metrics(6);

alphaPrecision = Alpha_score(1);
alphaSensitivity = Alpha_score(2);
alphaSpecificity = Alpha_score(3);
alphaAccuracy = Alpha_score(4);
alphaFMeasure = Alpha_score(5);
alphaKappa = Alpha_score(6);

updateAlphaPositions = precision > alphaPrecision && ...
    sensitivity > alphaSensitivity && ...
    specificity > alphaSpecificity && ...
    accuracy > alphaAccuracy && ...
    FMeasure > alphaFMeasure && ...
    kappa > alphaKappa;
```

```

% Update Alpha, Beta, and Delta
if updateAlphaPositions
    Alpha_score = metrics; % Atualizacao de alpha
    Alpha_pos = Positions(i,:);
    Best_pos{k}=Positions(i,:);
    AlphaNet{k}=net;
    AlphaBest{k}=metrics;
end

betaPrecision = Beta_score(1);
betaSensitivity = Beta_score(2);
betaSpecificity = Beta_score(3);
betaAccuracy = Beta_score(4);
betaFMeasure = Beta_score(5);
betaKappa = Beta_score(6);

updateBetaPositions = precision > betaPrecision && ...
    sensitivity > betaSensitivity && ...
    specificity > betaSpecificity && ...
    accuracy > betaAccuracy && ...
    FMeasure > betaFMeasure && ...
    kappa > betaKappa;

if ~updateAlphaPositions && updateBetaPositions
    Beta_score = metrics; % Atualizacao de beta
    Beta_pos = Positions(i,:);
    BetaNet = net;
end

deltaPrecision = Delta_score(1);
deltaSensitivity = Delta_score(2);
deltaSpecificity = Delta_score(3);
deltaAccuracy = Delta_score(4);

```

```

deltaFMeasure = Delta_score(5);
deltaKappa = Delta_score(6);

updateDeltaPositions = precision > deltaPrecision && ...
    sensitivity > deltaSensitivity && ...
    specificity > deltaSpecificity && ...
    accuracy > deltaAccuracy && ...
    FMeasure > deltaFMeasure && ...
    kappa > deltaKappa;

if ~updateAlphaPositions && ~updateBetaPositions &&
    updateDeltaPositions
    Delta_score = metrics; % Atualizacao de delta
    Delta_pos = Positions(i,:);
end
end

a = 2-inter*((2)/Max_iter); % decrementa de 2 a 0

% Atualizacao das posicoes
for pos1 = 1:size(Positions,1)
    for pos2 = 1:size(Positions,2)

        r1 = rand(); % r1 numero aleatorio em [0,1]
        r2 = rand(); % r2 numero aleatorio em [0,1]

        A1 = 2*a*r1-a; % Equacao (3.3)
        C1 = 2*r2; % Equacao (3.4)

        % Equacao (3.5) - part 1
        D_alpha = abs(C1*Alpha_pos(pos2)-Positions(pos1,pos2)
            );

        % Equacao (3.6) - part 1

```

```

X1 = Alpha_pos(pos2)-A1*D_alpha;

r1 = rand();
r2 = rand();

A2 = 2*a*r1-a; % Equacao (3.3)
C2 = 2*r2; % Equacao (3.4)

% Equacao (3.5)-part 2
D_beta = abs(C2*Beta_pos(pos2)-Positions(pos1,pos2));

% Equacao (3.6)-part 2
X2 = Beta_pos(pos2)-A2*D_beta;

r1 = rand();
r2 = rand();

A3 = 2*a*r1-a; % Equacao (3.3)
C3 = 2*r2; % Equacao (3.4)

% Equacao (3.5)-part 3
D_delta = abs(C3*Delta_pos(pos2)-Positions(pos1,pos2)
);
X3 = Delta_pos(pos2)-A3*D_delta; % Equacao (3.5)-part
3

Positions(pos1,pos2) = (X1+X2+X3)/3;% Equacao (3.7)

end
end

inter=inter+1;
end

```

```
        Bests{k}=Best;  
end  
  
save AlphaNet{k}  
save AlphaBest{k}  
save Best_pos{k}  
save Bests{k}  
save decimalTest{k}  
save decimalYsaida{k}
```

**APÊNDICE B – Figuras e tabelas dos classificadores do método de aprendizagem supervisionado**

Figura 1 – Matriz de confusão MLP-PSO partição 1.

**Matriz de confusão**

Classe de saída	1	6 16.2%	0 0.0%	1 2.7%	0 0.0%	0 0.0%	85.7% 14.3%
	2	0 0.0%	9 24.3%	1 2.7%	0 0.0%	0 0.0%	90.0% 10.0%
	3	0 0.0%	0 0.0%	5 13.5%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	5 13.5%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	1 2.7%	1 2.7%	8 21.6%	80.0% 20.0%
			100% 0.0%	100% 0.0%	62.5% 37.5%	83.3% 16.7%	100% 0.0%
		Classe alvo					

Fonte: Do autor (2022).

Figura 2 – Matriz de confusão MLP-PSO partição 2.

**Matriz de confusão**

Classe de saída	1	5 13.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	1 2.6%	8 21.1%	0 0.0%	0 0.0%	0 0.0%	88.9% 11.1%
	3	0 0.0%	1 2.6%	8 21.1%	0 0.0%	0 0.0%	88.9% 11.1%
	4	0 0.0%	0 0.0%	0 0.0%	7 18.4%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 21.1%	100% 0.0%
			83.3% 16.7%	88.9% 11.1%	100% 0.0%	100% 0.0%	100% 0.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 3 – Matriz de confusão MLP-PSO partição 3.

**Matriz de confusão**

Classe de saída	1	5 13.5%	1 2.7%	0 0.0%	0 0.0%	0 0.0%	83.3% 16.7%
	2	1 2.7%	6 16.2%	0 0.0%	0 0.0%	0 0.0%	85.7% 14.3%
	3	1 2.7%	1 2.7%	8 21.6%	0 0.0%	0 0.0%	80.0% 20.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 16.2%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 21.6%	100% 0.0%
			71.4% 28.6%	75.0% 25.0%	100% 0.0%	100% 0.0%	100% 0.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 4 – Matriz de confusão MLP-PSO partição 4.

**Matriz de confusão**

Classe de saída	1	6 15.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	1 2.6%	9 23.1%	1 2.6%	0 0.0%	0 0.0%	81.8% 18.2%
	3	0 0.0%	0 0.0%	7 17.9%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	7 17.9%	1 2.6%	87.5% 12.5%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 17.9%	100% 0.0%
			85.7% 14.3%	100% 0.0%	87.5% 12.5%	100% 0.0%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 1 – Acertos e erros do classificador MLP-PSO obtidos na partição 1.

Classe	VP	FP	FN	VN
1	6	1	0	30
2	9	1	0	27
3	5	0	3	29
4	5	0	1	31
5	8	2	0	27

Fonte: Do autor (2022).

Tabela 2 – Acertos e erros do classificador MLP-PSO obtidos na partição 2.

Classe	VP	FP	FN	VN
1	5	0	1	32
2	8	1	1	28
3	8	1	0	29
4	7	0	0	31
5	8	0	0	30

Fonte: Do autor (2022).

Tabela 3 – Acertos e erros do classificador MLP-PSO obtidos na partição 3.

Classe	VP	FP	FN	VN
1	5	1	2	29
2	6	1	2	28
3	8	2	0	27
4	6	0	0	31
5	8	0	0	29

Fonte: Do autor (2022).

Tabela 4 – Acertos e erros do classificador MLP-PSO obtidos na partição 4.

Classe	VP	FP	FN	VN
1	6	0	1	32
2	9	2	0	28
3	7	0	1	31
4	7	1	0	31
5	7	0	1	31

Fonte: Do autor (2022).

Figura 5 – Matriz de confusão MLP-DE partição 1.

**Matriz de confusão**

Classe de saída	1	5 13.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	1 2.6%	9 23.7%	0 0.0%	0 0.0%	0 0.0%	90.0% 10.0%
	3	0 0.0%	0 0.0%	8 21.1%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	1 2.6%	85.7% 14.3%
	5	0 0.0%	0 0.0%	0 0.0%	1 2.6%	7 18.4%	87.5% 12.5%
			83.3% 16.7%	100% 0.0%	100% 0.0%	85.7% 14.3%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 6 – Matriz de confusão MLP-DE partição 2.

**Matriz de confusão**

Classe de saída	1	6 15.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	1 2.6%	8 21.1%	1 2.6%	0 0.0%	0 0.0%	80.0% 20.0%
	3	0 0.0%	1 2.6%	7 18.4%	0 0.0%	0 0.0%	87.5% 12.5%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	2 5.3%	75.0% 25.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 15.8%	100% 0.0%
			85.7% 14.3%	88.9% 11.1%	87.5% 12.5%	100% 0.0%	75.0% 25.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 7 – Matriz de confusão MLP-DE partição 3.

**Matriz de confusão**

Classe de saída	1	6 16.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	8 21.6%	3 8.1%	0 0.0%	0 0.0%	72.7% 27.3%
	3	1 2.7%	0 0.0%	5 13.5%	0 0.0%	0 0.0%	83.3% 16.7%
	4	0 0.0%	0 0.0%	0 0.0%	6 16.2%	1 2.7%	85.7% 14.3%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 18.9%	100% 0.0%
			85.7% 14.3%	100% 0.0%	62.5% 37.5%	100% 0.0%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 8 – Matriz de confusão MLP-DE partição 4.

**Matriz de confusão**

Classe de saída	1	6 15.8%	1 2.6%	0 0.0%	0 0.0%	0 0.0%	85.7% 14.3%
	2	0 0.0%	8 21.1%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	7 18.4%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	7 18.4%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	1 2.6%	0 0.0%	8 21.1%	88.9% 11.1%
			100% 0.0%	88.9% 11.1%	87.5% 12.5%	100% 0.0%	100% 0.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 5 – Acertos e erros do classificador MLP-DE obtidos na partição 1.

Classe	VP	FP	FN	VN
1	5	0	1	32
2	9	1	0	28
3	8	0	0	30
4	6	1	1	30
5	7	1	1	29

Fonte: Do autor (2022).

Tabela 6 – Acertos e erros do classificador MLP-DE obtidos na partição 2.

Classe	VP	FP	FN	VN
1	6	0	1	31
2	8	2	1	27
3	7	1	1	29
4	6	2	0	30
5	6	0	2	30

Fonte: Do autor (2022).

Tabela 7 – Acertos e erros do classificador MLP-DE obtidos na partição 3.

Classe	VP	FP	FN	VN
1	6	0	1	30
2	8	3	0	26
3	5	1	3	28
4	6	1	0	30
5	7	0	1	29

Fonte: Do autor (2022).

Tabela 8 – Acertos e erros do classificador MLP-DE obtidos na partição 4.

Classe	VP	FP	FN	VN
1	6	1	0	31
2	8	0	1	29
3	7	0	1	30
4	7	0	0	31
5	8	1	0	29

Fonte: Do autor (2022).

Figura 9 – Matriz de confusão MLP-GWO partição 1.

**Matriz de confusão**

Classe de saída	1	6 16.7%	0 0.0%	1 2.8%	0 0.0%	0 0.0%	85.7% 14.3%
	2	0 0.0%	8 22.2%	1 2.8%	0 0.0%	0 0.0%	88.9% 11.1%
	3	0 0.0%	0 0.0%	6 16.7%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	5 13.9%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	1 2.8%	8 22.2%	88.9% 11.1%
			100% 0.0%	100% 0.0%	75.0% 25.0%	83.3% 16.7%	100% 0.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 10 – Matriz de confusão MLP-GWO partição 2.

**Matriz de confusão**

Classe de saída	1	6 15.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	9 23.1%	1 2.6%	0 0.0%	0 0.0%	90.0% 10.0%
	3	0 0.0%	0 0.0%	6 15.4%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	7 17.9%	0 0.0%	100% 0.0%
	5	1 2.6%	0 0.0%	1 2.6%	0 0.0%	8 20.5%	80.0% 20.0%
			85.7% 14.3%	100% 0.0%	75.0% 25.0%	100% 0.0%	100% 0.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 11 – Matriz de confusão MLP-GWO partição 3.

**Matriz de confusão**

Classe de saída	1	6 15.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	9 23.7%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	7 18.4%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	1 2.6%	85.7% 14.3%
	5	0 0.0%	0 0.0%	1 2.6%	1 2.6%	7 18.4%	77.8% 22.2%
			100% 0.0%	100% 0.0%	87.5% 12.5%	85.7% 14.3%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 12 – Matriz de confusão MLP-GWO partição 4.

**Matriz de confusão**

Classe de saída	1	7 18.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	8 21.1%	1 2.6%	0 0.0%	0 0.0%	88.9% 11.1%
	3	0 0.0%	1 2.6%	7 18.4%	0 0.0%	0 0.0%	87.5% 12.5%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	2 5.3%	75.0% 25.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 15.8%	100% 0.0%
			100% 0.0%	88.9% 11.1%	87.5% 12.5%	100% 0.0%	75.0% 25.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 9 – Acertos e erros do classificador MLP-GWO obtidos na partição 1.

Classe	VP	FP	FN	VN
1	6	1	0	29
2	8	1	0	27
3	6	0	2	28
4	5	0	1	30
5	8	1	0	27

Fonte: Do autor (2022).

Tabela 10 – Acertos e erros do classificador MLP-GWO obtidos na partição 2.

Classe	VP	FP	FN	VN
1	6	0	1	32
2	9	1	0	29
3	6	0	2	31
4	7	0	0	32
5	8	2	0	29

Fonte: Do autor (2022).

Tabela 11 – Acertos e erros do classificador MLP-GWO obtidos na partição 3.

Classe	VP	FP	FN	VN
1	6	0	0	32
2	9	0	0	29
3	7	0	1	30
4	6	1	1	30
5	7	2	1	28

Fonte: Do autor (2022).

Tabela 12 – Acertos e erros do classificador MLP-GWO obtidos na partição 4.

Classe	VP	FP	FN	VN
1	7	0	0	31
2	8	1	1	28
3	7	1	1	29
4	6	2	0	30
5	6	0	2	30

Fonte: Do autor (2022).

Figura 13 – Matriz de confusão SVM-PSO partição 1.

**Confusion Matrix**

Output Class	1	4 10.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	2 5.1%	9 23.1%	0 0.0%	0 0.0%	0 0.0%	81.8% 18.2%
	3	1 2.6%	0 0.0%	8 20.5%	0 0.0%	0 0.0%	88.9% 11.1%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.4%	1 2.6%	85.7% 14.3%
	5	0 0.0%	0 0.0%	0 0.0%	1 2.6%	7 17.9%	87.5% 12.5%
			57.1% 42.9%	100% 0.0%	100% 0.0%	85.7% 14.3%	87.5% 12.5%
		1	2	3	4	5	
		Target Class					

Fonte: Do autor (2022).

Figura 14 – Matriz de confusão SVM-PSO partição 2.

**Confusion Matrix**

Output Class	1	6 15.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	1 2.6%	8 20.5%	3 7.7%	0 0.0%	0 0.0%	66.7% 33.3%
	3	0 0.0%	1 2.6%	5 12.8%	0 0.0%	0 0.0%	83.3% 16.7%
	4	0 0.0%	0 0.0%	0 0.0%	7 17.9%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 20.5%	100% 0.0%
			85.7% 14.3%	88.9% 11.1%	62.5% 37.5%	100% 0.0%	100% 0.0%
		1	2	3	4	5	
		Target Class					

Fonte: Do autor (2022).

Figura 15 – Matriz de confusão SVM-PSO partição 3.

**Confusion Matrix**

Output Class	1	6 16.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	8 21.6%	2 5.4%	0 0.0%	0 0.0%	80.0% 20.0%
	3	0 0.0%	1 2.7%	6 16.2%	0 0.0%	0 0.0%	85.7% 14.3%
	4	0 0.0%	0 0.0%	0 0.0%	5 13.5%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	1 2.7%	8 21.6%	88.9% 11.1%
			100% 0.0%	88.9% 11.1%	75.0% 25.0%	83.3% 16.7%	100% 0.0%
		1	2	3	4	5	
		Target Class					

Fonte: Do autor (2022).

Figura 16 – Matriz de confusão SVM-PSO partição 4.

**Confusion Matrix**

Output Class	1	5 13.9%	0 0.0%	1 2.8%	0 0.0%	0 0.0%	83.3% 16.7%
	2	1 2.8%	8 22.2%	1 2.8%	0 0.0%	0 0.0%	80.0% 20.0%
	3	0 0.0%	0 0.0%	6 16.7%	0 0.0%	1 2.8%	85.7% 14.3%
	4	0 0.0%	0 0.0%	0 0.0%	6 16.7%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 19.4%	100% 0.0%
			83.3% 16.7%	100% 0.0%	75.0% 25.0%	100% 0.0%	87.5% 12.5%
		1	2	3	4	5	
		<b>Target Class</b>					

Fonte: Do autor (2022).

Tabela 13 – Acertos e erros do classificador SVM-PSO obtidos na partição 1.

Classe	VP	FP	FN	VN
1	4	0	3	32
2	9	2	0	28
3	8	1	0	30
4	6	1	1	31
5	7	1	1	30

Fonte: Do autor (2022).

Tabela 14 – Acertos e erros do classificador SVM-PSO obtidos na partição 2.

Classe	VP	FP	FN	VN
1	6	0	1	32
2	8	4	1	26
3	5	1	3	30
4	7	0	0	32
5	8	0	0	31

Fonte: Do autor (2022).

Tabela 15 – Acertos e erros do classificador SVM-PSO obtidos na partição 3.

Classe	VP	FP	FN	VN
1	6	0	0	31
2	8	2	1	26
3	6	1	2	28
4	5	0	1	31
5	8	1	0	28

Fonte: Do autor (2022).

Tabela 16 – Acertos e erros do classificador SVM-PSO obtidos na partição 4.

Classe	VP	FP	FN	VN
1	5	1	1	29
2	8	2	0	26
3	6	1	2	27
4	6	0	0	30
5	7	0	1	28

Fonte: Do autor (2022).

Figura 17 – Matriz de confusão SVM-DE partição 1.

**Matriz de confusão**

Classe de saída	1	4 10.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	2 5.4%	8 21.6%	2 5.4%	0 0.0%	0 0.0%	66.7% 33.3%
	3	0 0.0%	0 0.0%	6 16.2%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	7 18.9%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	8 21.6%	100% 0.0%
			66.7% 33.3%	100% 0.0%	75.0% 25.0%	100% 0.0%	100% 0.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 18 – Matriz de confusão SVM-DE partição 2.

**Matriz de confusão**

Classe de saída	1	5 13.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	2 5.3%	9 23.7%	0 0.0%	0 0.0%	0 0.0%	81.8% 18.2%
	3	0 0.0%	0 0.0%	8 21.1%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	5 13.2%	2 5.3%	71.4% 28.6%
	5	0 0.0%	0 0.0%	0 0.0%	1 2.6%	6 15.8%	85.7% 14.3%
			71.4% 28.6%	100% 0.0%	100% 0.0%	83.3% 16.7%	75.0% 25.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 19 – Matriz de confusão SVM-DE partição 3.

**Matriz de confusão**

Classe de saída	1	6 16.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	8 21.6%	3 8.1%	0 0.0%	0 0.0%	72.7% 27.3%
	3	0 0.0%	1 2.7%	4 10.8%	0 0.0%	0 0.0%	80.0% 20.0%
	4	0 0.0%	0 0.0%	0 0.0%	5 13.5%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	1 2.7%	1 2.7%	8 21.6%	80.0% 20.0%
			100% 0.0%	88.9% 11.1%	50.0% 50.0%	83.3% 16.7%	100% 0.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 20 – Matriz de confusão SVM-DE partição 4.

**Matriz de confusão**

Classe de saída	1	5 12.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	1 2.6%	8 20.5%	2 5.1%	0 0.0%	0 0.0%	72.7% 27.3%
	3	1 2.6%	1 2.6%	6 15.4%	0 0.0%	0 0.0%	75.0% 25.0%
	4	0 0.0%	0 0.0%	0 0.0%	7 17.9%	1 2.6%	87.5% 12.5%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 17.9%	100% 0.0%
			71.4% 28.6%	88.9% 11.1%	75.0% 25.0%	100% 0.0%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 17 – Acertos e erros do classificador SVM-DE obtidos na partição 1.

Classe	VP	FP	FN	VN
1	4	0	2	31
2	8	4	0	25
3	6	0	2	29
4	7	0	0	30
5	8	0	0	29

Fonte: Do autor (2022).

Tabela 18 – Acertos e erros do classificador SVM-DE obtidos na partição 2.

Classe	VP	FP	FN	VN
1	5	0	2	31
2	9	2	0	27
3	8	0	0	30
4	5	2	1	30
5	6	1	2	29

Fonte: Do autor (2022).

Tabela 19 – Acertos e erros do classificador SVM-DE obtidos na partição 3.

Classe	VP	FP	FN	VN
1	6	0	0	31
2	8	3	1	25
3	4	1	4	28
4	5	0	1	31
5	8	2	0	27

Fonte: Do autor (2022).

Tabela 20 – Acertos e erros do classificador SVM-DE obtidos na partição 4.

Classe	VP	FP	FN	VN
1	5	0	2	32
2	8	3	1	27
3	6	2	2	29
4	7	1	0	31
5	7	0	1	31

Fonte: Do autor (2022).

Figura 21 – Matriz de confusão SVM-GWO partição 1.

**Matriz de confusão**

Classe de saída	1	4 10.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	2 5.3%	8 21.1%	0 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
	3	1 2.6%	1 2.6%	8 21.1%	0 0.0%	0 0.0%	80.0% 20.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	1 2.6%	85.7% 14.3%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 18.4%	100% 0.0%
			57.1% 42.9%	88.9% 11.1%	100% 0.0%	100% 0.0%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 22 – Matriz de confusão SVM-GWO partição 2.

**Matriz de confusão**

Classe de saída	1	6 15.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	1 2.6%	9 23.1%	2 5.1%	0 0.0%	0 0.0%	75.0% 25.0%
	3	0 0.0%	0 0.0%	5 12.8%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	7 17.9%	1 2.6%	87.5% 12.5%
	5	0 0.0%	0 0.0%	1 2.6%	0 0.0%	7 17.9%	87.5% 12.5%
			85.7% 14.3%	100% 0.0%	62.5% 37.5%	100% 0.0%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 23 – Matriz de confusão SVM-GWO partição 3.

**Matriz de confusão**

Classe de saída	1	6 16.2%	1 2.7%	0 0.0%	0 0.0%	0 0.0%	85.7% 14.3%
	2	0 0.0%	8 21.6%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	8 21.6%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 16.2%	1 2.7%	85.7% 14.3%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 18.9%	100% 0.0%
			100% 0.0%	88.9% 11.1%	100% 0.0%	100% 0.0%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Figura 24 – Matriz de confusão SVM-GWO partição 4.

**Matriz de confusão**

Classe de saída	1	5 13.5%	0 0.0%	1 2.7%	0 0.0%	0 0.0%	83.3% 16.7%
	2	1 2.7%	8 21.6%	1 2.7%	0 0.0%	0 0.0%	80.0% 20.0%
	3	0 0.0%	0 0.0%	6 16.2%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 16.2%	1 2.7%	85.7% 14.3%
	5	0 0.0%	0 0.0%	0 0.0%	1 2.7%	7 18.9%	87.5% 12.5%
			83.3% 16.7%	100% 0.0%	75.0% 25.0%	85.7% 14.3%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 21 – Acertos e erros do classificador SVM-GWO obtidos na partição 1.

Classe	VP	FP	FN	VN
1	4	0	3	31
2	8	2	1	27
3	8	2	0	28
4	6	1	0	31
5	7	0	1	30

Fonte: Do autor (2022).

Tabela 22 – Acertos e erros do classificador SVM-GWO obtidos na partição 2.

Classe	VP	FP	FN	VN
1	6	0	1	32
2	9	3	0	27
3	5	0	3	31
4	7	1	0	31
5	7	1	1	30

Fonte: Do autor (2022).

Tabela 23 – Acertos e erros do classificador SVM-GWO obtidos na partição 3.

Classe	VP	FP	FN	VN
1	6	1	0	30
2	8	0	1	28
3	8	0	0	29
4	6	1	0	30
5	7	0	1	29

Fonte: Do autor (2022).

Tabela 24 – Acertos e erros do classificador SVM-GWO obtidos na partição 4.

Classe	VP	FP	FN	VN
1	5	1	1	30
2	8	2	0	27
3	6	0	2	29
4	6	1	1	29
5	7	1	1	28

Fonte: Do autor (2022).

**APÊNDICE C – Figuras e tabelas dos classificadores do método de aprendizagem não-supervisionado**

Figura 25 – Matriz de confusão do método KM obtida na partição 1 com 24 clusters.

**Matriz de confusão**

Classe de saída	1	4 10.5%	2 5.3%	1 2.6%	0 0.0%	0 0.0%	57.1% 42.9%
	2	0 0.0%	6 15.8%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	2 5.3%	1 2.6%	6 15.8%	0 0.0%	0 0.0%	66.7% 33.3%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	2 5.3%	75.0% 25.0%
	5	0 0.0%	0 0.0%	1 2.6%	1 2.6%	6 15.8%	75.0% 25.0%
			66.7% 33.3%	66.7% 33.3%	75.0% 25.0%	85.7% 14.3%	75.0% 25.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 25 – Acertos e erros do método KM obtidos na partição 1 com 24 clusters.

Classe	VP	FP	FN	VN
1	4	3	2	29
2	6	0	3	29
3	6	3	2	27
4	6	2	1	29
5	6	2	2	28

Fonte: Do autor (2022).

Figura 26 – Matriz de confusão do método KM obtida na partição 2 com 20 clusters.

**Matriz de confusão**

Classe de saída	1	3 8.1%	1 2.7%	2 5.4%	0 0.0%	0 0.0%	50.0% 50.0%
	2	0 0.0%	7 18.9%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	3 8.1%	1 2.7%	6 16.2%	0 0.0%	0 0.0%	60.0% 40.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 16.2%	4 10.8%	60.0% 40.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 10.8%	100% 0.0%
			50.0% 50.0%	77.8% 22.2%	75.0% 25.0%	100% 0.0%	50.0% 50.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 26 – Acertos e erros do classificador do método KM obtidos na partição 2 com 20 clusters.

Classe	VP	FP	FN	VN
1	3	3	3	28
2	7	0	2	28
3	6	4	2	25
4	6	4	0	27
5	4	0	4	29

Fonte: Do autor (2022).

Figura 27 – Matriz de confusão do método KM obtida na partição 3 com 25 clusters.

**Matriz de confusão**

Classe de saída	1	4 10.5%	0 0.0%	1 2.6%	0 0.0%	0 0.0%	80.0% 20.0%
	2	3 7.9%	7 18.4%	1 2.6%	0 0.0%	0 0.0%	63.6% 36.4%
	3	0 0.0%	2 5.3%	6 15.8%	0 0.0%	0 0.0%	75.0% 25.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	2 5.3%	75.0% 25.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 15.8%	100% 0.0%
			57.1% 42.9%	77.8% 22.2%	75.0% 25.0%	100% 0.0%	75.0% 25.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 27 – Acertos e erros do classificador do método KM obtidos na partição 3 com 25 clusters.

Classe	VP	FP	FN	VN
1	4	1	3	30
2	7	4	2	25
3	6	2	2	28
4	6	2	0	30
5	6	0	2	30

Fonte: Do autor (2022).

Figura 28 – Matriz de confusão do método KM obtida na partição 4 com 22 clusters.

**Matriz de confusão**

Classe de saída	1	6 15.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	8 21.1%	3 7.9%	0 0.0%	0 0.0%	72.7% 27.3%
	3	1 2.6%	0 0.0%	2 5.3%	0 0.0%	0 0.0%	66.7% 33.3%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	2 5.3%	75.0% 25.0%
	5	0 0.0%	0 0.0%	3 7.9%	1 2.6%	6 15.8%	60.0% 40.0%
			85.7% 14.3%	100% 0.0%	25.0% 75.0%	85.7% 14.3%	75.0% 25.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 28 – Acertos e erros do classificador do método KM obtidos na partição 4 com 22 clusters.

Classe	VP	FP	FN	VN
1	6	0	1	31
2	8	3	0	27
3	2	1	6	29
4	6	2	1	29
5	6	4	2	26

Fonte: Do autor (2022).

Figura 29 – Matriz de confusão do método FCM obtida na partição 1 com 24 clusters.

**Matriz de confusão**

Classe de saída	1	3 7.9%	0 0.0%	1 2.6%	0 0.0%	0 0.0%	75.0% 25.0%
	2	0 0.0%	8 21.1%	2 5.3%	0 0.0%	0 0.0%	80.0% 20.0%
	3	3 7.9%	1 2.6%	4 10.5%	0 0.0%	0 0.0%	50.0% 50.0%
	4	0 0.0%	0 0.0%	0 0.0%	7 18.4%	3 7.9%	70.0% 30.0%
	5	0 0.0%	0 0.0%	1 2.6%	0 0.0%	5 13.2%	83.3% 16.7%
			50.0% 50.0%	88.9% 11.1%	50.0% 50.0%	100% 0.0%	62.5% 37.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 29 – Acertos e erros do classificador do método FCM obtidos na partição 1 com 24 clusters.

Classe	VP	FP	FN	VN
1	3	1	3	31
2	8	2	1	27
3	4	4	4	26
4	7	3	0	28
5	5	1	3	29

Fonte: Do autor (2022).

Figura 30 – Matriz de confusão do método FCM obtida na partição 2 com 21 clusters.

**Matriz de confusão**

Classe de saída	1	4 10.8%	2 5.4%	1 2.7%	0 0.0%	0 0.0%	57.1% 42.9%
	2	1 2.7%	6 16.2%	1 2.7%	0 0.0%	0 0.0%	75.0% 25.0%
	3	0 0.0%	0 0.0%	6 16.2%	0 0.0%	0 0.0%	100% 0.0%
	4	2 5.4%	0 0.0%	0 0.0%	6 16.2%	2 5.4%	60.0% 40.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 16.2%	100% 0.0%
			57.1% 42.9%	75.0% 25.0%	75.0% 25.0%	100% 0.0%	75.0% 25.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 30 – Acertos e erros do classificador do método FCM obtidos na partição 2 com 21 clusters.

Classe	VP	FP	FN	VN
1	4	3	3	27
2	6	2	2	27
3	6	0	2	29
4	6	4	0	27
5	6	0	2	29

Fonte: Do autor (2022).

Figura 31 – Matriz de confusão do método FCM obtida na partição 3 com 24 clusters.

**Matriz de confusão**

Classe de saída	1	6 15.8%	0 0.0%	3 7.9%	0 0.0%	0 0.0%	66.7% 33.3%
	2	0 0.0%	9 23.7%	1 2.6%	0 0.0%	0 0.0%	90.0% 10.0%
	3	1 2.6%	0 0.0%	4 10.5%	0 0.0%	0 0.0%	80.0% 20.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	3 7.9%	66.7% 33.3%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 13.2%	100% 0.0%
			85.7% 14.3%	100% 0.0%	50.0% 50.0%	100% 0.0%	62.5% 37.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 31 – Acertos e erros do classificador do método FCM obtidos na partição 3 com 24 clusters.

Classe	VP	FP	FN	VN
1	6	3	1	28
2	9	1	0	28
3	4	1	4	29
4	6	3	0	29
5	5	0	3	30

Fonte: Do autor (2022).

Figura 32 – Matriz de confusão do método FCM obtida na partição 4 com 22 clusters.

**Matriz de confusão**

Classe de saída	1	3 7.9%	0 0.0%	1 2.6%	0 0.0%	0 0.0%	75.0% 25.0%
	2	1 2.6%	9 23.7%	2 5.3%	0 0.0%	0 0.0%	75.0% 25.0%
	3	2 5.3%	0 0.0%	5 13.2%	0 0.0%	0 0.0%	71.4% 28.6%
	4	0 0.0%	0 0.0%	0 0.0%	7 18.4%	2 5.3%	77.8% 22.2%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6 15.8%	100% 0.0%
			50.0% 50.0%	100% 0.0%	62.5% 37.5%	100% 0.0%	75.0% 25.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 32 – Acertos e erros do classificador do método FCM obtidos na partição 4 com 22 clusters.

Classe	VP	FP	FN	VN
1	3	1	3	31
2	9	3	0	26
3	5	2	3	28
4	7	2	0	29
5	6	0	2	30

Fonte: Do autor (2022).

Figura 33 – Matriz de confusão do método GK obtida na partição 1 com 23 clusters.

**Matriz de confusão**

Classe de saída	1	5 13.5%	1 2.7%	1 2.7%	0 0.0%	0 0.0%	71.4% 28.6%
	2	0 0.0%	7 18.9%	1 2.7%	0 0.0%	0 0.0%	87.5% 12.5%
	3	1 2.7%	1 2.7%	6 16.2%	0 0.0%	0 0.0%	75.0% 25.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 16.2%	1 2.7%	85.7% 14.3%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 18.9%	100% 0.0%
			83.3% 16.7%	77.8% 22.2%	75.0% 25.0%	100% 0.0%	87.5% 12.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 33 – Acertos e erros do classificador do método GK obtidos na partição 1 com 23 clusters.

Classe	VP	FP	FN	VN
1	5	2	1	29
2	7	1	2	27
3	6	2	2	27
4	6	1	0	30
5	7	0	1	29

Fonte: Do autor (2022).

Figura 34 – Matriz de confusão do método GK obtida na partição 2 com 23 clusters.

**Matriz de confusão**

Classe de saída	1	4 10.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	3 7.7%	8 20.5%	0 0.0%	0 0.0%	0 0.0%	72.7% 27.3%
	3	0 0.0%	1 2.6%	8 20.5%	0 0.0%	0 0.0%	88.9% 11.1%
	4	0 0.0%	0 0.0%	0 0.0%	7 17.9%	4 10.3%	63.6% 36.4%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 10.3%	100% 0.0%
			57.1% 42.9%	88.9% 11.1%	100% 0.0%	100% 0.0%	50.0% 50.0%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 34 – Acertos e erros do classificador do método GK obtidos na partição 2 com 23 clusters.

Classe	VP	FP	FN	VN
1	4	0	3	32
2	8	3	1	27
3	8	1	0	30
4	7	4	0	28
5	4	0	4	31

Fonte: Do autor (2022).

Figura 35 – Matriz de confusão do método GK obtida para a partição 3 com 25 clusters.

**Matriz de confusão**

Classe de saída	1	6 15.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	8 21.1%	4 10.5%	0 0.0%	0 0.0%	66.7% 33.3%
	3	1 2.6%	1 2.6%	3 7.9%	0 0.0%	0 0.0%	60.0% 40.0%
	4	0 0.0%	0 0.0%	0 0.0%	6 15.8%	3 7.9%	66.7% 33.3%
	5	0 0.0%	0 0.0%	1 2.6%	0 0.0%	5 13.2%	83.3% 16.7%
			85.7% 14.3%	88.9% 11.1%	37.5% 62.5%	100% 0.0%	62.5% 37.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 35 – Acertos e erros do classificador do método GK obtidos na partição 3 com 25 clusters.

Classe	VP	FP	FN	VN
1	6	0	1	31
2	8	4	1	25
3	3	2	5	28
4	6	3	0	29
5	5	1	3	29

Fonte: Do autor (2022).

Figura 36 – Matriz de confusão do método GK obtida na partição 4 com 24 clusters.

**Matriz de confusão**

Classe de saída	1	3 8.1%	1 2.7%	2 5.4%	0 0.0%	0 0.0%	50.0% 50.0%
	2	2 5.4%	7 18.9%	0 0.0%	0 0.0%	0 0.0%	77.8% 22.2%
	3	1 2.7%	0 0.0%	6 16.2%	0 0.0%	0 0.0%	85.7% 14.3%
	4	0 0.0%	0 0.0%	0 0.0%	7 18.9%	3 8.1%	70.0% 30.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 13.5%	100% 0.0%
			50.0% 50.0%	87.5% 12.5%	75.0% 25.0%	100% 0.0%	62.5% 37.5%
		1	2	3	4	5	
		Classe alvo					

Fonte: Do autor (2022).

Tabela 36 – Acertos e erros do classificador do método GK obtidos na partição 4 com 24 clusters.

Classe	VP	FP	FN	VN
1	3	3	3	28
2	7	2	1	27
3	6	1	2	28
4	7	3	0	27
5	5	0	3	29

Fonte: Do autor (2022).