

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM  
ENGENHARIA DE PRODUÇÃO E SISTEMAS**

**DAVID GABRIEL DE BARROS FRANCO**

**OTIMIZAÇÃO DE REDES NEURAS ARTIFICIAIS POR MEIO DE  
META-HEURÍSTICAS APLICADA A PROBLEMAS DE CLASSIFICAÇÃO**

**CURITIBA**

**2019**

**DAVID GABRIEL DE BARROS FRANCO**

**OTIMIZAÇÃO DE REDES NEURAS ARTIFICIAIS POR MEIO DE  
META-HEURÍSTICAS APLICADA A PROBLEMAS DE CLASSIFICAÇÃO**

Tese apresentada ao Curso de Pós-Graduação em Engenharia de Produção e Sistemas da Pontifícia Universidade Católica do Paraná, como requisito parcial à obtenção do título de doutor em Engenharia de Produção.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Maria Teresinha Arns Steiner

**CURITIBA**

**2019**

Dados da Catalogação na Publicação  
Pontifícia Universidade Católica do Paraná  
Sistema Integrado de Bibliotecas – SIBI/PUCPR  
Biblioteca Central  
Luci Eduarda Wielganczuk – CRB 9/1118

F825o  
2019 Franco, David Gabriel de Barros  
Otimização de redes neurais artificiais por meio de meta-heurísticas aplicada a problemas de classificação / David Gabriel de Barros Franco ; orientadora: Maria Teresinha Arns Steiner. – 2019.  
101 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2019  
Bibliografia: f. 85-101

1. Redes neurais (Computação). 2. Reconhecimento de padrões.  
3. Heurística. 4. Otimização combinatória. I. Steiner, Maria Teresinha Arns.  
II. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Engenharia de Produção e Sistemas. III. Título.

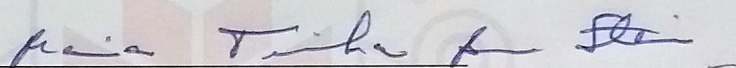
CDD 22. ed. – 006.32

## TERMO DE APROVAÇÃO

### David Gabriel de Barros Franco

#### OTIMIZAÇÃO DE REDES NEURAIS ARTIFICIAIS POR MEIO DE META-HEURÍSTICAS APLICADA A PROBLEMAS DE CLASSIFICAÇÃO.

Tese aprovada como requisito parcial para obtenção do grau de Doutor no Curso de Doutorado em Engenharia de Produção e Sistemas, Programa de Pós-Graduação em Engenharia de Produção e Sistemas, da Escola Politécnica da Pontifícia Universidade Católica do Paraná, pela seguinte banca examinadora:

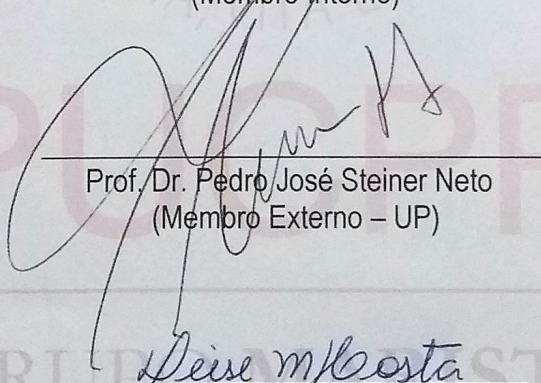


Presidente da Banca

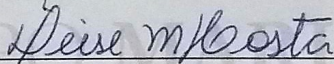
Prof.ª. Dra. Maria Teresinha Arns Steiner  
(Orientadora)



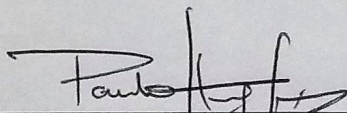
Prof. Dr. Roberto Zanetti Freire  
(Membro Interno)



Prof. Dr. Pedro José Steiner Neto  
(Membro Externo – UP)



Prof.ª. Dra. Deise Maria Bertholdi Costa  
(Membro Externo – UFPR)



Prof. Dr. Paulo Henrique Siqueira  
(Membro Externo – UFPR)

Curitiba, 29 de janeiro de 2019.

## **AGRADECIMENTOS**

Primeiramente e acima de todas as coisas, agradeço a Deus, por guiar-nos.

Agradeço à minha esposa e companheira de todas as horas Idelma por sempre ser paciente e sábia, e me apoiar e aconselhar em todas as situações; também à minha mãe, Maria Rita, e à minha irmã, Marília, por estarem presentes em todas as ocasiões, alegres e difíceis, e a toda a família que sempre nos apoiou.

Agradeço imensamente à minha professora e orientadora (desde o TCC) Maria Teresinha, por sua dedicação ao ensino e sabedoria, sempre amável e disposta a ajudar, e a todos os professores que sempre me transmitiram conhecimento e valores éticos e morais durante toda a minha vida.

Agradeço também aos professores membros da banca pelo tempo dedicado ao exame deste trabalho e por suas contribuições inestimáveis.

*“De tudo o que se tem ouvido, a suma é:  
Teme a Deus e guarda os seus mandamentos; porque isto é o dever de todo homem.”*

*Salomão, Século X a.C.*

## RESUMO

As Redes Neurais Artificiais (RNAs) são consideradas sistemas não-lineares e seu processo de treinamento pode ser visto como um problema de otimização. Uma vez que os pesos sinápticos (estrutura onde o aprendizado é armazenado) ótimos são difíceis de serem encontrados por métodos analíticos, faz-se necessário o uso de métodos iterativos de otimização local ou global. Os algoritmos de treinamento baseados no gradiente são amplamente utilizados devido à sua eficácia, porém, são de lenta convergência e, muitas vezes, não conseguem “fugir” de mínimos locais. Tem sido demonstrado que o desempenho da RNA depende de diversos fatores, entre eles as condições iniciais (os pesos sinápticos iniciais) da rede. A apropriada inicialização desses pesos pode contribuir com a melhoria do desempenho da rede e muitos métodos têm sido propostos para esse fim. O objetivo principal do presente trabalho é propor uma nova técnica de otimização dos pesos iniciais das RNAs *Multilayer Perceptron (MLP)* por meio das meta-heurísticas (MHs) *Differential Evolution (DE)*, *Genetic Algorithms (GA)* e *Particle Swarm Optimization (PSO)*. Estas RNAs serão posteriormente treinadas com o algoritmo *Levenberg-Marquardt (LM)*. Estas técnicas híbridas serão denominadas, respectivamente, *DE-LM*, *GA-LM* e *PSO-LM* ou, genericamente, *MH-LM*. Estas técnicas híbridas aqui propostas foram aplicadas, ineditamente, a um problema real de classificação de áreas abandonadas para implantação de usinas de geração de energia solar. Em termos de Erro Quadrático Médio (*MSE*), considerando apenas o conjunto de teste (50% dos dados), a técnica *MH-LM* obteve uma redução média de aproximadamente 20% em relação ao algoritmo *LM*: *GA-LM* obteve uma redução de 22,9%, *PSO-LM* alcançou uma redução de 19,6% e *DE-LM* reduziu o *MSE* em 17,2%. Também foram testados outros seis conjuntos de dados da literatura, nos quais a estratégia *MH-LM* se mostrou superior ao algoritmo *LM* e a outras cinco técnicas de classificação: *Linear Discriminant Analysis (LDA)*, *Naive Bayes (NB)*, *Decision Trees (DT)*, *Support Vector Machines (SVM)* e *Random Forest (RF)*. Com isso mostra-se o desempenho promissor da técnica proposta, que poderá ser explorada futuramente em outras aplicações e com outras MHs.

Palavras-chave: Redes Neurais Artificiais. Reconhecimento de Padrões. Análise de Agrupamentos. Meta-heurísticas.

## ABSTRACT

Artificial Neural Networks (ANNs) are considered non-linear systems and their training process can be seen as an optimization problem. Since the optimal synaptic weights (structure where learning is stored) are difficult to find by analytical methods, it is necessary to use iterative methods of local or global optimization. Gradient-based training algorithms are widely used because of their effectiveness; however, they are slow to converge and often fail to "escape" from local minima. It has been demonstrated that the performance of ANN depends on several factors, including initial conditions (the initial synaptic weights) of the network. Appropriate initialization of these weights can contribute to the improvement of network performance and many methods have been proposed for this purpose. The main objective of the present work is to propose a new technique to optimize the Multilayer Perceptron (MLP) ANN initial weights through the Differential Evolution (DE), Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) metaheuristics (MHs). These ANNs will be further trained with the Levenberg-Marquardt (LM) algorithm. These hybrid techniques will be named, respectively, DE-LM, GA-LM and PSO-LM or, generally, MH-LM. These hybrid techniques proposed here were applied, unquestioningly, to a real problem of classification of abandoned areas for the implantation of solar power plants. In terms of Mean Squared Error (MSE), considering only the test set (50% of the dataset), the MH-LM technique obtained an average reduction of approximately 20% in relation to the LM algorithm: GA-LM obtained a reduction of 22.9%, PSO-LM reached a reduction of 19.6% and DE-LM reduced the MSE by 17.2%. Another six sets of literature data were tested in which the MH-LM strategy was superior to the LM algorithm and to five other classification techniques: Linear Discriminant Analysis (LDA), Naive Bayes (NB), Decision Trees (DT), Support Vector Machines (SVM) and Random Forest (RF). This shows the promising performance of the proposed technique, which can be explored in other applications and other MHs in the future.

Key-words: Artificial Neural Networks. Pattern Recognition. Cluster Analysis. Metaheuristics.



## LISTA DE ILUSTRAÇÕES

Figura 2.1 – Processo <i>KDD</i> .....	21
Figura 2.2 – As subdivisões da área “Inteligência de Máquina” .....	25
Figura 2.3 – Classificação das técnicas de otimização .....	26
Figura 2.4 – Formalização do algoritmo <i>DE</i> .....	29
Figura 2.5 – Formalização do algoritmo <i>GA</i> .....	30
Figura 2.6 – Formalização do algoritmo <i>PSO</i> .....	32
Figura 2.7 – Formalização do algoritmo <i>FCM</i> .....	34
Figura 2.8 – Modelo de neurônio artificial .....	35
Figura 2.9 – Superfície de erro para neurônios lineares .....	38
Figura 2.10 – Superfície de erro para neurônios não-lineares .....	39
Figura 2.11 – Autores mais produtivos (2010-2018) .....	45
Figura 2.12 – Palavras-chave mais citadas (2010-2018) .....	46
Figura 2.13 – Publicações por países (2010-2018).....	47
Figura 2.14 – Mapa das publicações (2010-2018) .....	47
Figura 2.15 – Meta-heurísticas mais citadas entre 2010-2018.....	49
Figura 3.1 – Metodologia utilizada.....	55
Figura 3.2 – Exemplo de áreas abandonadas.....	57
Figura 3.3 – Evolução no uso do solo ao longo dos anos (hectares) .....	58
Figura 3.4 – Consumo de energia no mundo ( $10^{18}$ Joules).....	59
Figura 3.5 – Áreas consideradas no trabalho.....	60
Figura 3.6 – Resultado do agrupamento por meio do <i>HFCM</i> .....	62
Figura 3.7 – Algoritmo <i>HFCM</i> .....	63
Figura 3.8 – Algoritmo <i>MH-MLP</i> .....	65
Figura 3.9 – Esquematização da técnica <i>MH-MLP</i> .....	65
Figura 4.1 – Comparativo: Meta-heurísticas x Número de iterações.....	67
Figura 4.2 – Comparativo entre a acurácia e o balanceamento .....	76
Figura 4.3 – Melhorias atingidas pela técnica proposta ( <i>MH-LM</i> ) .....	78
Figura 4.4 – Capacidade de generalização da técnica proposta ( <i>MH-LM</i> ).....	78
Figura 5.1 – Projetos em funcionamento.....	82
Figura 5.2 – Geração de energia renovável no mundo .....	83

Figura 5.3 – Geração de empregos em energia renovável no mundo .....	84
Figura 5.4 – Investimento total em energia renovável (em bilhões de US\$) .....	84

## LISTA DE TABELAS

Tabela 2.1 – Algoritmos de agrupamento .....	23
Tabela 2.2 – Métricas de validação .....	24
Tabela 2.3 – Algoritmos de otimização inspirados na natureza .....	48
Tabela 3.1 – Análise de correlação .....	61
Tabela 4.1 – Resultados para o conjunto <i>Solar clustering</i> .....	68
Tabela 4.2 – Resultados com o conjunto <i>Aggregation</i> .....	68
Tabela 4.3 – Resultados com o conjunto <i>Compound</i> .....	69
Tabela 4.4 – Resultados com o conjunto <i>D31</i> .....	69
Tabela 4.5 – Resultados com o conjunto <i>T4.8k</i> .....	69
Tabela 4.6 – Resultados com o conjunto <i>Credit Card</i> .....	70
Tabela 4.7 – Resultados com o conjunto <i>Wine Quality</i> .....	70
Tabela 4.8 – Redução de tempo alcançada por <i>HFCM</i> .....	70
Tabela 4.9 – Resultados para o conjunto <i>Solar classification</i> com <i>LM</i> .....	72
Tabela 4.10 – Resultados para o conjunto <i>Solar classification</i> com <i>DE-MLP</i> .....	72
Tabela 4.11 – Resultados para o conjunto <i>Solar classification</i> com <i>DE-LM</i> .....	72
Tabela 4.12 – Resultados para o conjunto <i>Solar classification</i> com <i>GA-MLP</i> .....	73
Tabela 4.13 – Resultados para o conjunto <i>Solar classification</i> com <i>GA-LM</i> .....	73
Tabela 4.14 – Resultados para o conjunto <i>Solar classification</i> com <i>PSO-MLP</i> .....	73
Tabela 4.15 – Resultados para o conjunto <i>Solar classification</i> com <i>PSO-LM</i> .....	73
Tabela 4.16 – Acurácia dos modelos testados .....	74
Tabela 4.17 – Exemplo de base de dados desbalanceada .....	76

## LISTA DE ABREVIATURAS E SIGLAS

<i>ABC</i>	<i>Artificial Bee Colony</i>
<i>ACO</i>	<i>Ant Colony Optimization</i>
<i>AIS</i>	<i>Artificial Immune System</i>
<i>ANN</i>	<i>Artificial Neural Networks</i>
<i>ARIMA</i>	<i>Autoregressive Integrated Moving Average</i>
<i>BFO</i>	<i>Bacterial Foraging Optimization</i>
<i>BNN</i>	<i>Bayesian Neural Network</i>
<i>CA</i>	<i>Cultural Algorithm</i>
<i>CE</i>	<i>Computação Evolutiva</i>
<i>CPU</i>	<i>Central Processing Unit</i>
<i>CSO</i>	<i>Cat Swarm Optimization</i>
<i>DCA</i>	<i>Dendritic Cell Algorithm</i>
<i>DE</i>	<i>Differential Evolution</i>
<i>DT</i>	<i>Decision Trees</i>
<i>ES</i>	<i>Evolutionary Strategy</i>
<i>FA</i>	<i>Firefly Algorithm</i>
<i>FCM</i>	<i>Fuzzy c-means</i>
<i>FSA</i>	<i>Fish Swarm Algorithm</i>
<i>GA</i>	<i>Genetic Algorithm</i>
<i>GMDH</i>	<i>Group Method of Data Handling</i>
<i>GP</i>	<i>Genetic Programming</i>
<i>GSA</i>	<i>Gravitational Search Algorithm</i>
<i>HSA</i>	<i>Harmony Search Algorithm</i>
<i>IA</i>	<i>Inteligência Artificial</i>
<i>IC</i>	<i>Inteligência Computacional</i>
<i>IWO</i>	<i>Invasive Weed Optimization</i>
<i>KHA</i>	<i>Krill Herd Algorithm</i>
<i>LDA</i>	<i>Linear Discriminant Analysis</i>
<i>MA</i>	<i>Memetic Algorithm</i>
<i>MH</i>	<i>Meta-heurística</i>

<i>MLP</i>	<i>Multilayer Perceptron</i>
<i>MSE</i>	<i>Mean Squared Error</i>
<i>NB</i>	<i>Naïve Bayes</i>
<i>NAILS</i>	<i>Network Analysis Interface for Literature Studies</i>
<i>PCA</i>	<i>Principal Component Analysis</i>
<i>PNN</i>	<i>Probabilistic Neural Network</i>
<i>PO</i>	<i>Pesquisa Operacional</i>
<i>PSO</i>	<i>Particle Swarm Optimization</i>
<i>RBF</i>	<i>Radial Basis Function</i>
<i>RF</i>	<i>Random Forest</i>
<i>RFD</i>	<i>River Formation Dynamics</i>
<i>RNA</i>	<i>Rede Neural Artificial</i>
<i>SA</i>	<i>Simulated Annealing</i>
<i>SFLA</i>	<i>Shuffled Frog-Leaping Algorithm</i>
<i>SI</i>	<i>Swarm Intelligence</i>
<i>SVM</i>	<i>Support Vector Machine</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>14</b>
1.1 PROBLEMATIZAÇÃO.....	16
1.2 OBJETIVOS.....	17
<b>1.2.1 Objetivo geral</b> .....	<b>17</b>
<b>1.2.2 Objetivos específicos</b> .....	<b>17</b>
1.3 JUSTIFICATIVA.....	18
1.4 CONTRIBUIÇÕES DO TRABALHO.....	18
1.5 LIMITAÇÕES DA PESQUISA.....	19
1.6 ORGANIZAÇÃO DO TRABALHO.....	19
<b>2 REFERENCIAL TEÓRICO</b> .....	<b>20</b>
2.1 DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS.....	20
2.2 INTELIGÊNCIA COMPUTACIONAL.....	24
2.3 META-HEURÍSTICAS.....	25
<b>2.3.1 <i>Differential Evolution</i></b> .....	<b>27</b>
<b>2.3.2 <i>Genetic Algorithms</i></b> .....	<b>29</b>
<b>2.3.3 <i>Particle Swarm Optimization</i></b> .....	<b>31</b>
2.4 FUZZY C-MEANS.....	32
2.5 REDES NEURAS ARTIFICIAIS.....	34
<b>2.5.1 Histórico das Redes Neurais Artificiais</b> .....	<b>35</b>
<b>2.5.2 O <i>Perceptron</i> de múltiplas camadas</b> .....	<b>40</b>
<b>2.5.3 Treinamento por Levenberg-Marquardt</b> .....	<b>42</b>
<b>2.5.4 Teoria da Regularização Bayesiana</b> .....	<b>43</b>
2.6 TRABALHOS CORRELATOS.....	44
<b>2.6.1 Análise bibliométrica</b> .....	<b>44</b>
<b>2.6.2 Estado da arte</b> .....	<b>49</b>
2.6.2.1 Redes Neurais Artificiais.....	50
2.6.2.2 Clusterização.....	52
<b>3 METODOLOGIA</b> .....	<b>55</b>
3.1 FERRAMENTAS E MÉTODOS.....	56
<b>3.1.1 O problema da energia solar</b> .....	<b>56</b>

<b>3.1.2 Seleção e pré-processamento de dados .....</b>	<b>59</b>
3.1.2.1 Preenchimento dos dados faltantes.....	61
<b>3.1.3 Otimização dos pesos iniciais das RNA .....</b>	<b>64</b>
<b>4 RESULTADOS .....</b>	<b>66</b>
4.1 ALGORITMO <i>HFCM</i> .....	66
4.1.1 Clusterização aplicada ao problema de Energia Solar.....	66
4.1.2 Instâncias da literatura.....	68
4.2 RNA OTIMIZADA.....	71
4.2.1 Classificação aplicada ao problema da energia solar .....	71
4.2.2 Instâncias da literatura.....	74
4.2.3 Base de dados desbalanceada.....	75
<b>5 CONCLUSÕES.....</b>	<b>80</b>
<b>REFERÊNCIAS.....</b>	<b>85</b>

## 1 INTRODUÇÃO

As primeiras atividades formais da Pesquisa Operacional (PO) se iniciaram na Inglaterra durante a Segunda Guerra Mundial, quando um time de cientistas britânicos decidiu avaliar a melhor forma de alocação de materiais bélicos e suprimentos com base em princípios científicos. Depois da guerra, as ideias desenvolvidas nas operações militares foram adaptadas à melhoria da eficiência e produtividade no setor civil, em um processo de otimização produtiva (Gass, 2011).

A otimização de processos decisórios e operacionais tem potencial para aperfeiçoar a correta alocação dos recursos naturais, de natureza finita e de valor crescente (Gunasekaran e Subramanian, 2018; Jaehn, 2016). Inovações de cunho tecnológico, tanto de produtos como de processos, podem contribuir para essa melhor utilização, seja pela redução do consumo e melhoria da eficiência como pelo maior aproveitamento de recursos subutilizados (Diaz-Rainey e Ashton, 2015; Yabar, Hara e Uwasu, 2012).

Com a difusão do uso de bancos de dados e o crescimento exponencial do tamanho dos mesmos, a ciência da PO se vê em meio ao desafio de como detectar dados que sejam relevantes e de como encontrar significado em tais dados. Entre as metodologias (ou processos) disponíveis para enfrentar este problema está a Mineração de Dados (*Data Mining, DM*), cujas técnicas “garimpam” grandes conjuntos de dados em busca de padrões que possam levar a decisões proveitosas (Meisel e Mattfeld, 2010; Olafsson, Li e Wu, 2008).

Devido à grande variedade de técnicas de *DM* e aos inúmeros tipos e fontes de informação disponíveis é fundamental se definir os limites de aplicação e a relevância de certos métodos de acordo com os objetivos a serem atingidos pelo pesquisador. Também faz-se necessário entender o modo pelo qual o problema deva ser resolvido, escolhendo entre domínios de aplicação como classificação, agrupamento (clusterização) e associação, entre outros (Vadim, 2018).

Existem, basicamente, três categorias de problemas em *DM*: aprendizado supervisionado (com treinamento guiado por dados rotulados), aprendizado não-supervisionado (que não utiliza rótulos de classes) e aprendizado semi-supervisionado, um meio-termo entre os dois tipos anteriores. Estes termos provêm do Aprendizado de Máquina (*Machine Learning, ML*), conceito que combina todo o conjunto de tecnologias do *DM* (Sammut, Webb e Webb, 2017).



Na primeira categoria, aprendizado supervisionado, um modelo de análise de dados (ou classificador) é construído e treinado até atingir uma qualidade de resposta (acurácia) satisfatória. Há casos em que tal nível de qualidade pode ser atingido, quer seja pela inaptidão do algoritmo (procedimento sistemático de resolução) escolhido, quer seja por os dados utilizados não possuírem uma estrutura que possa ser identificada. Exemplos deste tipo de aprendizado são os problemas de classificação, nos quais existem determinados conjuntos de entidades (instâncias) distribuídas de uma maneira específica em certas classes. Para um conjunto limitado dessas entidades (conjunto de treinamento) é conhecida a maneira em que estão distribuídas e rotuladas e, baseado nesse conjunto, é feita a adaptação do modelo (treinamento ou aprendizado) de modo que se consiga extrapolar o conhecimento adquirido para o restante do conjunto (Witten *et al.*, 2017).

No aprendizado não-supervisionado, o modelo é construído sem a ajuda de rótulos pré-definidos. As instâncias são agrupadas com base em medidas de similaridades entre suas variáveis. Cabe, então, ao pesquisador, definir os significados dos grupos formados (Roiger, 2017).

Já no aprendizado semi-supervisionado existem tanto dados rotulados quanto dados sem rótulo que serão utilizados no treinamento do algoritmo. É bastante útil quando apenas um pequeno número de instâncias rotuladas encontra-se disponível. Este tipo de aprendizado é utilizado tanto em tarefas de classificação quanto de clusterização (Chapelle, Schölkopf e Zien, 2006).

Entre as técnicas mais utilizadas e que mais se destacam em problemas de classificação estão as Redes Neurais Artificiais (RNA) (Seo e Shin, 2019; Ting, Tan e Sim, 2019), modelos “caixa-preta” que funcionam como aproximadores universais de funções que podem ajustar qualquer tipo de função para relacionar variáveis explanatórias com variáveis de resposta, mesmo que tal relacionamento seja não-linear e complexo (Gibert *et al.*, 2018). Essa característica é levada além com o Teorema da Aproximação Universal (Hornik, Stinchcombe e White, 1989), que demonstra que uma RNA do tipo *Multilayer Perceptron (MLP)* com apenas uma camada oculta pode se adequar precisamente a qualquer função contínua. Entretanto, para que a aproximação seja satisfatória, é necessário definir parâmetros para os quais não existem regras pré-estabelecidas. Entre estes parâmetros, podemos citar os pesos sinápticos iniciais e vieses que, geralmente, são inicializados aleatoriamente (Villarrubia *et al.*, 2018).

Este trabalho abordará este problema específico das RNAs, de inicialização dos pesos iniciais e vieses da rede. Será conduzido um estudo de caso aplicado a um problema de classificação na área de energia solar. De forma secundária, será abordado o problema de clusterização e, também, sua otimização na fase de pré-processamento dos dados.

## 1.1 PROBLEMATIZAÇÃO

RNAs são consideradas um sistema não-linear e seu processo de treinamento pode ser visto como um problema de otimização não linear (Luo *et al.*, 2016; Timotheou, 2009). Uma vez que os pesos sinápticos (estrutura onde o aprendizado é armazenado) ótimos são difíceis de serem encontrados por métodos analíticos, faz-se necessário o uso de métodos iterativos de otimização local ou global (Erdogmus *et al.*, 2005). Os algoritmos de treinamento baseados no gradiente são amplamente utilizados devido a sua eficácia (Erdogmus *et al.*, 2005) porém, são de lenta convergência e, muitas vezes, não conseguem fugir de mínimos locais (Nasr e Chtourou, 2011).

Tem sido demonstrado que o desempenho da RNA (velocidade e acurácia) treinada com algoritmos baseados no gradiente depende de fatores que incluem o algoritmo de treinamento, as funções de transferência (também chamadas funções de ativação), as condições iniciais (pesos iniciais), os dados utilizados e seu pré-processamento e a arquitetura da rede (número de camadas e números de neurônios) (Man *et al.*, 2011; Ros *et al.*, 2007; Yam e Chow, 1995).

Historicamente, os pesos sinápticos eram inicializados com valores iguais, o que levava a uma convergência coletiva dos mesmos, o que era indesejado (Sodhi e Chandra, 2014). Para quebrar esta simetria, o método da inicialização aleatória em um intervalo definido foi proposto por Rumelhart, Hinton e Williams (1986), embora a aleatoriedade esteja presente nas RNAs desde o modelo do *Perceptron*, que assumia conexões aleatórias entre suas unidade funcionais básicas, os neurônios (Rosenblatt, 1958).

A apropriada inicialização dos pesos sinápticos da rede pode diminuir o tempo de treinamento assim como evitar os tão indesejados mínimos locais (Adam *et al.*, 2014; Napolitano, Serinaldi e See, 2011; Qing Song, 2011; Talaška *et al.*, 2015) e muitos métodos têm sido desenvolvidos visando superar estas dificuldades, como os

que envolvem mínimos quadrados e análise intervalar, que foram efetivos em diminuir o erro inicial, embora estes tenham se mostrado instáveis e, muitas vezes, incapazes de superar mínimos locais (Qiao, Li e Li, 2016).

## 1.2 OBJETIVOS

Os objetivos estão aqui divididos em geral e específicos.

### 1.2.1 Objetivo geral

Propor uma nova técnica de otimização para a obtenção dos pesos iniciais das RNAs *MLP* por meio das meta-heurísticas (MHs) Evolução Diferencial (*Differential Evolution – DE*), Algoritmos Genéticos (*Genetic Algorithms – GA*) e Otimização por Enxame de Partículas (*Particle Swarm Optimization – PSO*), abordagem denominada *MH-MLP (Metaheuristic-Multilayer Peceptron)*. Posteriormente será utilizado o algoritmo *Levenberg-Marquardt (LM)* para refinamento da solução, o que será chamado de *MH-LM (Metaheuristic-Levenberg-Marquardt)*. Na etapa de pré-processamento dos dados, no problema de preenchimento de dados faltantes, é proposta a otimização do algoritmo *Fuzzy C-means (FCM)* através da pré-seleção dos centroides iniciais por meio, também, das MHs *DE*, *GA* e *PSO*, técnica que será aqui denominada *HFCM (Hybrid Fuzzy C-means)*.

### 1.2.2 Objetivos específicos

- Adaptar as MHs *DE*, *GA* e *PSO* aos problemas de otimização e treinamento de RNAs, além do problema de clusterização;
- Programar, em MATLAB, o algoritmo *MH-LM* e *MH-MLP*, além do algoritmo *HFCM*, a partir das MHs *DE*, *GA* e *PSO* e dos algoritmos *LM* e *FCM*;
- Selecionar, a partir de bases de dados públicas, os dados relativos ao problema das instalações de energia solar e outros problemas clássicos da literatura;
- Pré-processar os dados selecionados tornando-os adequados à metodologia proposta;

- Testar os algoritmos híbridos propostos com um problema real e inédito (classificação de áreas abandonadas no território continental Estadunidense visando à instalação de usinas de energia solar fotovoltaica);
- Comparar os algoritmos propostos em outros conjuntos de dados disponíveis na literatura.

### 1.3 JUSTIFICATIVA

Uma vez que existem diversas técnicas de extração de conhecimento de bases de dados (número que cresce a cada dia), faz-se necessária a seleção do modelo mais adequado aos dados do problema, minimizando o erro entre os valores-alvo e as respostas do modelo e/ou reduzindo o tempo de obtenção da solução do problema (Mehdiyev *et al.*, 2016).

A proposta principal do trabalho, *MH-LM*, visa melhorar o desempenho de classificação das RNAs em um problema envolvendo energia renovável (fotovoltaica) e áreas abandonadas com potencial de reutilização. Também foi tratado o problema de inicialização dos centroides do algoritmo *FCM*, geralmente inicializados aleatoriamente a partir de uma amostra dos dados de entrada.

### 1.4 CONTRIBUIÇÕES DO TRABALHO

A relevância desta pesquisa consiste em tratar o problema da classificação de áreas abandonadas para instalação de usinas fotovoltaicas utilizando RNAs híbridas, com inicialização dos pesos sinápticos, da qual depende fortemente a qualidade do processo de aprendizagem (Cao *et al.*, 2018; Qiao, Li e Li, 2016; Sodhi e Chandra, 2014), por meio das MHs *DE*, *GA* e *PSO* (*MH-LM*). Também será testado o uso destas MHs de forma isolada no treinamento da RNA (*MH-MLP*), em substituição ao algoritmo *LM*, o qual é amplamente utilizado no treinamento de RNAs por acelerar a velocidade de convergência do treinamento e assegurar maior acurácia em relação a outros algoritmos que utilizam derivadas de primeira ordem (Daniel, Chaturvedi e Kolhe, 2018; Mammadli, 2017; Vakili *et al.*, 2016; Ye e Kim, 2018).

Também foi utilizado ineditamente o algoritmo *HFCM* (Franco e Steiner, 2018) na etapa de pré-processamento dos dados para agrupamento e preenchimento dos dados faltantes visando obter maior realidade no resultado final. O algoritmo *HFCM*

foi fruto do desenvolvimento desta tese, já tendo sido publicado no periódico *Journal of Cleaner Production* (Qualis A1).

## 1.5 LIMITAÇÕES DA PESQUISA

Uma vez que o número de MHs disponíveis é demasiado grande e vem crescendo continuamente, neste trabalho o modelo proposto foi limitado a um pequeno nicho que contemplasse tanto abordagens evolutivas (*DE* e *GA*) quanto de enxames (*PSO*), além do modelo neuronal (*RNA*) e de agrupamento difuso (*FCM*) em problemas de classificação e, secundariamente, clusterização.

A pesquisa se limitou, também, durante o posterior treinamento da *RNA*, ao uso do algoritmo *LM*.

Entretanto, este modelo híbrido proposto é flexível, do ponto de vista de sua adaptação a outras MHs e a outros métodos de otimização, além de poder ser aplicada a outros problemas como, por exemplo, previsão de séries temporais.

## 1.6 ORGANIZAÇÃO DO TRABALHO

Após esta seção introdutória (capítulo 1), o trabalho está dividido da seguinte maneira: no capítulo 2 tem-se o referencial teórico, com uma visão geral do processo de Descoberta de Conhecimento em Bases de Dados (*Knowledge Discovery in Databases, KDD*) e da Inteligência Computacional (*IC*) e suas principais contribuições para a *PO*. Também serão descritas neste capítulo as MHs de otimização empregadas: *DE*, *GA* e *PSO*; o algoritmo *FCM*, de agrupamento difuso, e as *RNAs*, seguido dos trabalhos correlatos ao reconhecimento de padrões e à análise de agrupamentos empregando as técnicas mais atuais.

O capítulo 3 traz a descrição do estudo de caso conduzido na área das energias renováveis e as metodologias, primeiramente da análise de agrupamento aplicada ao problema do preenchimento dos dados faltantes seguida da otimização de *RNAs* por MHs. No capítulo 4 são apresentados e discutidos os resultados referentes aos problemas de clusterização e otimização de *RNAs* e, finalmente, no capítulo 5 são expostas as conclusões do trabalho.

## 2 REFERENCIAL TEÓRICO

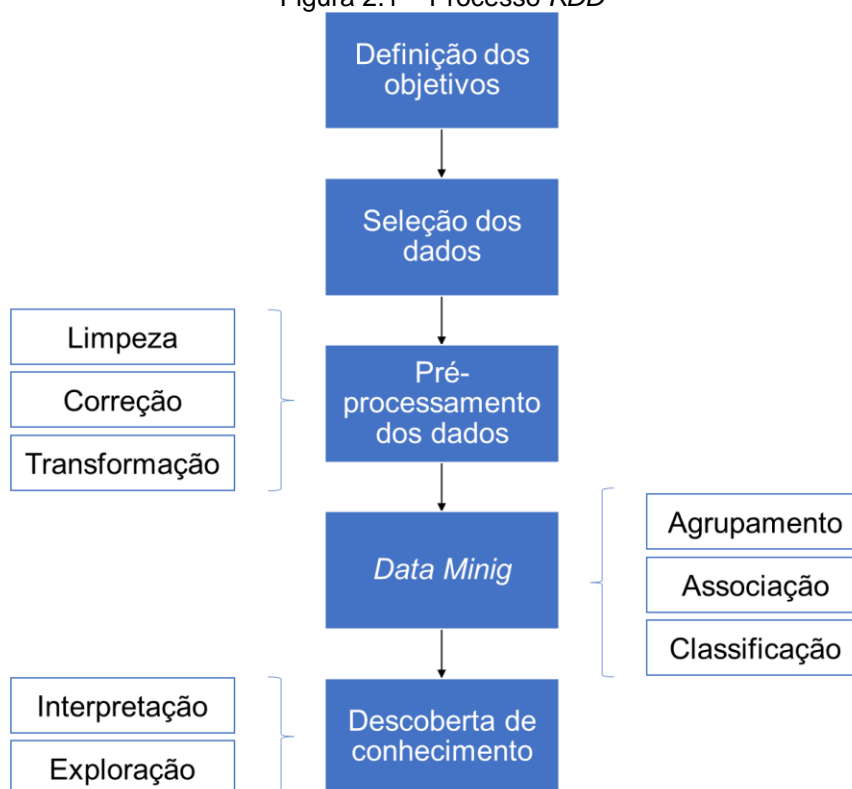
A fundamentação teórica do trabalho foi realizada a partir de uma pesquisa bibliográfica sistemática nas principais bases de publicações científicas: *Science Direct*, *Scopus*, *Springer*, *Taylor & Francis*, *Web of Science* e *Wiley*. Os trabalhos aqui analisados foram escolhidos em função do número de citações à época da coleta e o período de publicação foi limitado ao intervalo 2010-2018.

Na sequência é analisado brevemente o processo de Descoberta de Conhecimento em Bases de Dados, o assunto de IC, ramo da ciência da computação que trata da habilidade dos computadores em aprender a partir de dados coletados do ambiente, seguida da descrição das técnicas que serão utilizadas para atingir o objetivo proposto: *DE*, *GA*, *PSO*, *FCM* e *RNA* e, finalmente, de uma revisão sistemática de trabalhos científicos relacionados ao problema estudado.

### 2.1 DESCOBERTA DE CONHECIMENTO EM BASES DE DADOS

*KDD* é uma área do conhecimento dedicada à identificação e extração de padrões significativos de informação a partir de bases de dados (Fayyad, Piatetsky-Shapiro e Smyth, 1996). Pode-se definir o *KDD* como um campo multidisciplinar que combina análise de dados com métodos de processamento de dados e conhecimento de domínio, transformando dados em conhecimento inteligível e aplicável, relevante para um processo decisório bem informado (Gibert *et al.*, 2018).

A aplicação do *KDD* se dá em múltiplos estágios, começando com a seleção, pré-processamento e transformação dos dados, que pode incluir remoção de valores atípicos (*outliers*), substituição de dados faltantes, normalização, análise de componentes principais (*Principal Component Analysis, PCA*), dentre outras técnicas, dependendo do algoritmo a ser utilizado na etapa subsequente, de mineração (ou aprendizado) de dados, terminando com a interpretação dos resultados e a geração do conhecimento (Fayyad, Piatetsky-Shapiro e Smyth, 1996; Gamarra, Guerrero e Montero, 2016; Orriols-Puig *et al.*, 2013). O processo *KDD* é sumarizado na Figura 2.1.

Figura 2.1 – Processo *KDD*

Fonte: adaptado de: Gamarra, Guerrero e Montero, 2016

A etapa de mineração de dados (*Data Mining*) pode utilizar um ou vários algoritmos em busca de padrões, tendências e estruturas na base de dados, as quais podem assumir variadas formas, como equações, redes, grafos, conjuntos de regras, entre outros (Roiger, 2017; Witten *et al.*, 2017). Nessa etapa de aprendizado pode-se adotar duas abordagens distintas: na primeira abordagem, de aprendizado supervisionado, consideram-se, *a priori*, estruturas e padrões pré-definidos; na segunda abordagem, de aprendizado não-supervisionado, não se consideram essas possíveis estruturas e padrões, deixando ao algoritmo a tarefa de identificar tais relações entre as variáveis (Orriols-Puig *et al.*, 2013).

Na primeira abordagem tem-se, por exemplo, os problemas de classificação, que têm sido amplamente utilizados em diversas tarefas, como reconhecimento de texto (Hadi, Al-Radaideh e Alhawari, 2018; Mironczuk e Protasiewicz, 2018) e imagens (Yuan *et al.*, 2018), classificação de fraturas ósseas (Agrawal, 2018; Soni *et al.*, 2018), endometriose (Andres, Borrelli e Abrão, 2018), batimentos cardíacos e arritmia (Cappe *et al.*, 2018; Sannino e Pietro, De, 2018), qualidade de minerais (Tutmez, 2018) e identificação de plantas medicinais (Ren, Guo e Pang, 2018), só para citar alguns exemplos.

Entre as muitas técnicas disponíveis para tratar problemas de classificação pode-se citar *Naïve Bayes* (Hadi, Al-Radaideh e Alhawari, 2018; Kim *et al.*, 2018), *Decision Trees* (Beucher, Møller e Greve, 2017; Chao e Junzheng, 2018; Wu *et al.*, 2018), *Support Vector Machines* (Mancuhan e Clifton, 2018; Wu *et al.*, 2018), *Gaussian Process Classification* (Andrade, Tamura e Tsuchida, 2018; Rinta-Koski *et al.*, 2018; Sovizi *et al.*, 2017), *k-Nearest Neighbors* (Gallego *et al.*, 2018; Guo *et al.*, 2018; Müller *et al.*, 2019; Zhang *et al.*, 2017), *Ensemble methods* (Ankit e Saleena, 2018; Feng *et al.*, 2018; Jackowski, 2018) e *Artificial Neural Networks* (Gotsopoulos *et al.*, 2018; Li *et al.*, 2018; Sá Junior, Backes e Bruno, 2018; Sharma, Jain e Mishra, 2018; Simoncini *et al.*, 2018).

Na segunda abordagem, de aprendizado não-supervisionado, pode-se citar o problema de agrupamento, no qual as instâncias são agrupadas com base em alguma regra inerente à sua estrutura, como, por exemplo, a distância entre elas (Bramer, 2016; Roiger, 2017). Simplificadamente, pode-se dizer que, após a definição do número de *clusters* (grupos) e seus respectivos centros (chamados centroides), há uma primeira etapa de designação das instâncias para o centroide mais próximo, seguida pela otimização da localização do mesmo, minimizando o custo (Aggarwal, 2015).

Há um variado número e tipos de algoritmos de agrupamento (Fahad *et al.*, 2014; Halkidi, Batistakis e Vazirgiannis, 2001; Nanda e Panda, 2014; Xu e Wunsch II, 2005), dentre os quais destacam-se alguns na Tabela 2.1, com suas respectivas literaturas de origem.



Tabela 2.1 – Algoritmos de agrupamento

<b>Tipo</b>	<b>Algoritmo</b>	<b>Referência</b>
Particional	<i>K-Means</i>	(MacQueen, 1967)
	<i>PAM (Partitioning Around Medoids)</i>	
	<i>CLARA (Clustering Large Applications)</i>	(Ng e Han, 1994)
	<i>CLARANS (Clustering Large Applications based on Randomized Search)</i>	
	<i>K-prototypes, K-mode</i>	(Huang, 1998)
Hierárquico	<i>BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)</i>	(Zhang, Ramakrishnan e Livny, 1996)
	<i>CURE (Clustering Using Representatives)</i>	(Guha, Rastogi e Shim, 1998)
	<i>ROCK (Robust Clustering using linKs)</i>	(Guha, Rastogi e Shim, 2000)
Density-based	<i>DBSCAN (Density Based Spatial Clustering of Applications with Noise)</i>	(Ester <i>et al.</i> , 1996)
	<i>DENCLUE (Density-based Clustering)</i>	(Hinneburg e Keim, 1998, 2003)
Grid-based	<i>STING (Statistical Information Grid-based method)</i>	(Wang, Yang e Muntz, 1997)
	<i>WaveCluster</i>	(Sheikholeslami, Chatterjee e Zhang, 1998)
Fuzzy (soft)	<i>FCM</i>	(Bezdek, Ehrlich e Full, 1984)
	<i>EM (Expectation Maximization)</i>	(Dempster, Laird e Rubin, 1977)
Neural Network-based	<i>GLVQ (Generalized Learning Vector Quantization), SOFM (Self-Organizing Feature Maps)</i>	(Pal, Bezdek e Tsao, 1993)
Kernel-based	<i>SVC (Support Vector Clustering)</i>	(Ben-Hur <i>et al.</i> , 2001)

Fonte: o autor, 2018

Os resultados do agrupamento devem ser validados por alguma métrica, que pode ser externa, como rótulo de classes; interna, quando a avaliação se dá por características inerentes aos dados como, por exemplo, a variância e a separação dos pontos em diferentes *clusters*; ou relativa, onde as métricas visam comparar o uso de diferentes parâmetros durante a execução do algoritmo. A Tabela 2.2 sumariza algumas dessas métricas (Amigó *et al.*, 2009; Arbelaitz *et al.*, 2013; Halkidi, Batistakis e Vazirgiannis, 2001; Meilă, 2007; Zaki e Meira Jr., 2014).

Tabela 2.2 – Métricas de validação

Externa	Interna	Relativa
<i>Purity</i>	<i>BetaCV Measure</i>	<i>Calinski-Harabasz Criterion</i>
<i>Maximum Matching</i>	<i>Normalized Cut Measure</i>	<i>Gap Statistic</i>
<i>F-Measure</i>	<i>Modularity</i>	
<i>Conditional Entropy</i>	<i>Dunn Index</i>	
<i>Normalized Mutual Information</i>	<i>Davies-Bouldin Criterion</i>	
<i>Variation of Information</i>	<i>Silhouette Coefficient</i>	
<i>Jaccard Coefficient</i>	<i>Hubert Statistic</i>	
<i>Rand Statistic</i>		
<i>Fowlkes-Mallows Measure</i>		
<i>Discretized Hubert Statistic</i>		
<i>Normalized Discretized Hubert Statistic</i>		

Fonte: o autor, 2018

## 2.2 INTELIGÊNCIA COMPUTACIONAL

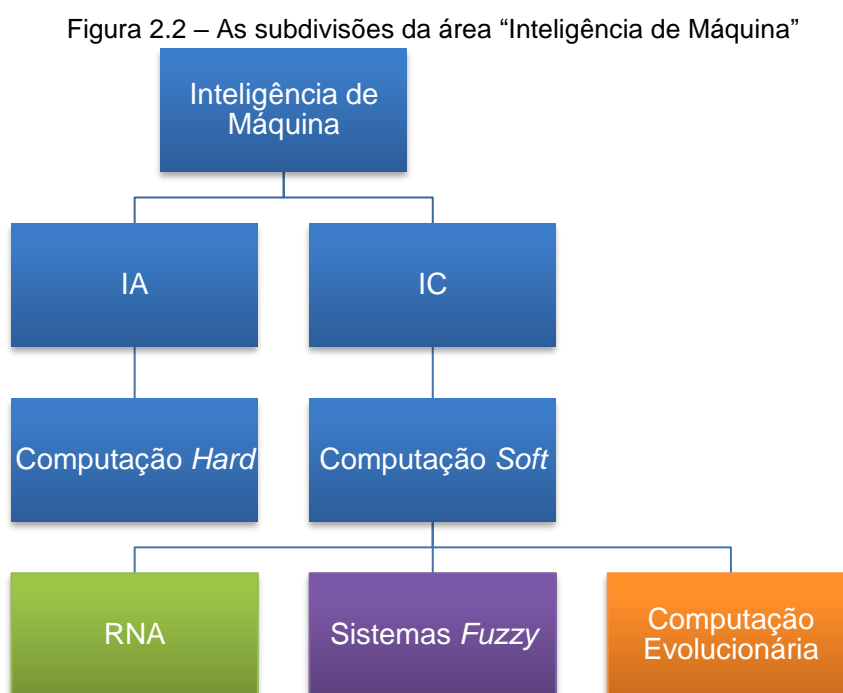
A Inteligência Artificial (IA) tradicional sempre objetivou simular comportamentos inteligentes a partir de representações exatas e completas do conhecimento, porém muitos problemas do mundo real não podem ser descritos de forma exata ou não se tem o conhecimento apropriado de seu funcionamento (são “caixas-pretas”). Desta forma, a Inteligência Computacional (IC) surgiu como uma solução a estas dificuldades, sem necessitar de muito conhecimento *a priori* do problema e produzindo soluções robustas e adaptáveis (flexíveis) a diversos cenários (Siddique e Adeli, 2013).

A área de IC envolve conceitos e paradigmas da Ciência da Computação e da PO objetivando a implementação de sistemas que apresentem comportamento inteligente (que pode ser definido como a habilidade de aprender e aplicar este aprendizado em novos cenários) em processos decisórios complexos. Entre estes paradigmas predominam os inspirados na natureza, como RNA, Sistemas *Fuzzy* e Computação Evolutiva (CE), além de sistemas híbridos, que apresentam vantagens como tolerância a falhas e incompletude ou imprecisão dos dados utilizados como entrada nos algoritmos (Kacprzyk e Pedrycz, 2015).

A estratégia geralmente adotada em IC é o uso de técnicas aproximativas que encontrem soluções parciais ou mesmo incompletas em um espaço de tempo viável

e a um custo computacional aceitável uma vez que, geralmente, envolvem problemas de alta dimensionalidade e com muitas instâncias (Kruse *et al.*, 2016).

A Figura 2.2 traz uma esquematização geral, embora não consensual, do que seria a “inteligência de máquina” e suas subdivisões, tendo como principal diferenciador entre IA e IC, a computação *Hard*, onde imprecisão e incerteza são indesejados, e a computação *Soft*, onde estes fatores estão frequentemente presentes (Siddique e Adeli, 2013).



Fonte: adaptado de: Siddique e Adeli, 2013

### 2.3 META-HEURÍSTICAS

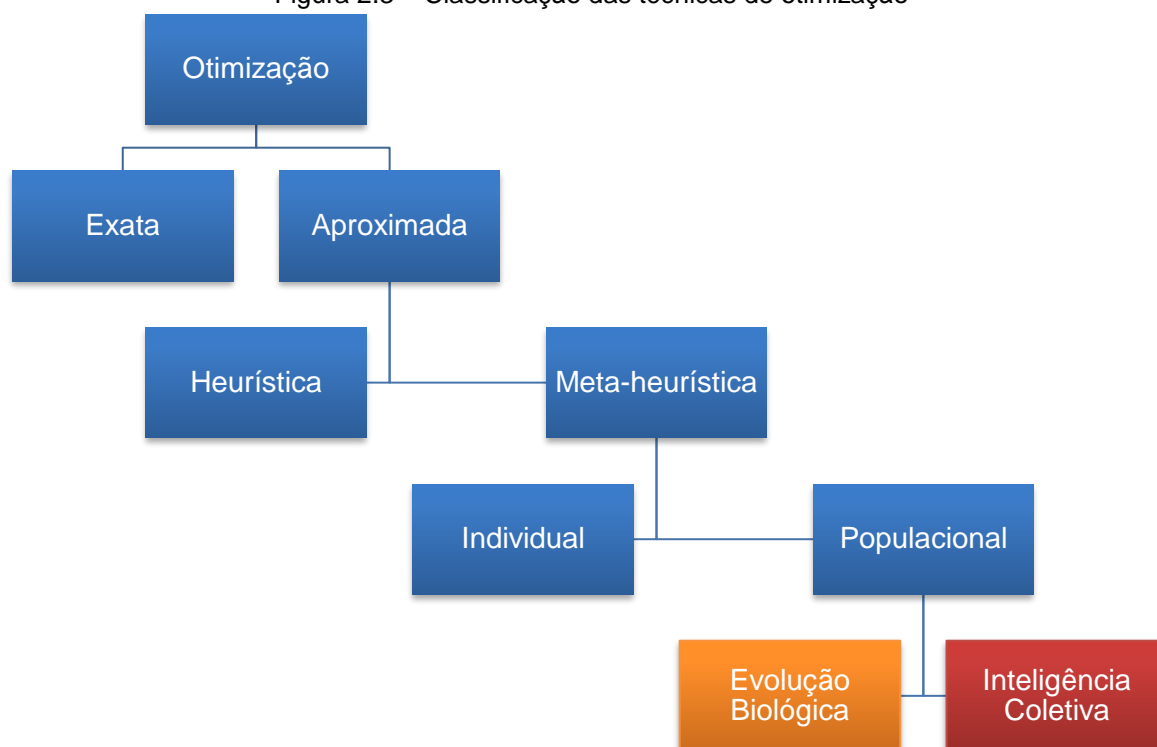
Nas últimas décadas métodos exatos em PO, como modelagem matemática de Programação Linear, têm desempenhado um papel central na tomada de decisão, buscando soluções ótimas para problemas do mundo real. Porém, em termos de complexidade computacional, uma grande parcela destes problemas (classificados como NP-difícil, ou *NP-hard*, *Non-deterministic Polynomial-time hard*) parecem ser implausíveis de se achar a solução ótima em um tempo polinomial em função do número de instâncias do problema (Ng *et al.*, 2018).

Em virtude disso, as pesquisas têm migrado para abordagens meta-heurísticas (MH), que são independentes das características do problema e oferecem soluções

próximas ao ótimo, para problemas reais, em um tempo computacional satisfatório (Bianchi *et al.*, 2009; Glover, 1986). Tais MHs fazem uso dos conceitos de *exploitation* e *exploration* (ambas palavras traduzidas como exploração, sendo que a 1ª. no sentido de profundidade e a 2ª. no sentido de largura), que se referem, respectivamente, à capacidade de varrer as vizinhanças de uma solução candidata promissora em busca da solução ideal (ótimo global) e à habilidade em escapar à armadilha de soluções parciais (ótimos locais) (Trelea, 2003).

As MHs aqui selecionadas representam duas grandes categorias: Algoritmos Evolucionários (*Evolutionary Algorithms, EA*) e Inteligência Coletiva (*Swarm Intelligence, SI*). *EA* lida com procedimentos contínuos de troca de informação entre soluções candidatas, aumentando a qualidade da resposta ao longo das iterações, processo conhecido como busca cega (inspirada na teoria da seleção natural), que raramente explora o conhecimento do domínio, e usa operadores evolucionários iterativamente a partir de soluções conhecidas (Whitley, 2001). Por outro lado, *SI* incorpora o comportamento coletivo de seres vivos, balanceando entre o conhecimento local do agente e o conhecimento adquirido pela população inteira (Ng *et al.*, 2018). A Figura 2.3 esquematiza a classificação das técnicas de otimização.

Figura 2.3 – Classificação das técnicas de otimização



Fonte: adaptado de: Ng *et al.*, 2018

A seguir são apresentadas as MHs selecionadas para a fase de otimização dos algoritmos propostos.

*DE*, que tem provado ser uma das estratégias de maior eficiência em otimização global, além de ser de simples implementação, ser robusta e de rápida convergência (Cai *et al.*, 2018). Tem sido aplicada em problemas de otimização de grande escala, multi-objetivo, com ou sem restrições, sob incerteza ou, ainda, em sistemas dinâmicos, em vários campos da ciência e engenharia, como reconhecimento de padrões, processamento de sinais, comunicações por satélite, redes de sensores *wireless*, entre outras (Das e Suganthan, 2011).

*GA*, estratégia baseada no princípio da seleção natural “sobrevivência do mais apto” e que explora inteligentemente uma busca aleatorizada da solução ótima, utilizando informações das gerações anteriores para guiar sua convergência pelo espaço de busca. É capaz de solucionar problemas com múltiplas soluções e, por ser uma técnica independente da superfície de erro, também pode solucionar problemas multidimensionais, não-diferenciáveis, descontínuos e não-paramétricos (Gupta, Vasudev e Bhattacharyya, 2018).

*PSO*, uma das técnicas baseadas em enxames de partículas mais populares e bem conhecidas. Entre suas características destacam-se a robustez e eficiência de busca, evitando a convergência prematura do algoritmo (Zhang *et al.*, 2018). Por ser de simples implementação, uma vez que utiliza apenas duas equações (de posição e velocidade) que guiam as partículas no espaço de busca, o algoritmo *PSO* tem sido utilizado em muitos campos do conhecimento, como roteirização de veículos, otimização multiobjetivo e sistemas de controle, entre outros (Chen *et al.*, 2018).

### **2.3.1 Differential Evolution**

*DE* é um algoritmo evolutivo de busca estocástica e otimização global que mantém, em cada iteração  $G$ , uma população de  $N$  candidatos (chamados vetores-alvo)  $x_{i,G} = [x_{j,i}]$ , onde  $j = 1, 2, \dots, D$  é o número de parâmetros do modelo (a dimensão do problema,  $\mathbb{R}^D$ ) e  $i = 1, 2, \dots, N$ , que são submetidos a sucessivas etapas de seleção, mutação e recombinação. A mutação expande o espaço de busca, criando uma nova solução baseada na diferença ponderada entre dois membros da população prévia somada a um terceiro membro, como mostrado na equação 2.1, onde  $v_{i,G+1}$  é chamado vetor doador (Brownlee, 2011; Storn e Price, 1997).

$$v_{i,G+1} = x_{r_1,G} + F * (x_{r_2,G} - x_{r_3,G}) \quad (2.1)$$

Na iteração  $G$  os índices  $r_1, r_2, r_3 \in \{1, 2, \dots, N\}$  são selecionados aleatoriamente e são mutuamente excludentes, além de diferentes do índice  $i$ , o que impõe uma população  $N \geq 4$ .  $F \in [0, 2]$  é a constante real de ponderação que controla a amplitude da variação diferencial. Visando aumentar a diversidade das soluções geradas na etapa de mutação e reutilizar indivíduos aptos, é executada a recombinação de acordo com as equações 2.2 e 2.3. O vetor  $u_{i,G+1}$  é chamado vetor de teste.

$$u_{i,G+1} = (u_{1,i,G+1}, u_{2,i,G+1}, \dots, u_{D,i,G+1}) \quad (2.2)$$

onde

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & \text{se } rand_{j,i} \leq CR \vee j = rand_i \\ x_{j,i,G} & \text{se } rand_{j,i} > CR \wedge j \neq rand_i \end{cases} \quad (2.3)$$

$$i = 1, 2, \dots, N; j = 1, 2, \dots, D$$

Na equação 2.3,  $rand_{j,i}$  é um número aleatório com distribuição uniforme no intervalo  $[0, 1]$ ,  $CR$  é a constante de recombinação pertencente ao intervalo  $[0, 1]$ , e  $rand_i$  é um número inteiro selecionado aleatoriamente no intervalo  $[1, 2, \dots, D]$ , que visa garantir um  $v_{i,G+1} \neq x_{i,G}$ .

O vetor-alvo  $x_{i,G}$  é comparado com o vetor de teste  $u_{i,G+1}$  e o de menor valor é selecionado para a próxima geração (sobrevivência do mais apto), como resumido na equação 2.4.

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{se } f(u_{i,G+1}) \leq f(x_{i,G}) \\ x_{i,G} & \text{caso contrário} \end{cases} \quad (2.4)$$

$$i = 1, 2, \dots, N$$

É apresentado na Figura 2.4, a seguir, o pseudocódigo do algoritmo *DE* (Brownlee, 2011; Chakraborty, 2008; Price, Storn e Lampinen, 2005; Storn e Price, 1997; Xing e Gao, 2014a).

Figura 2.4 – Formalização do algoritmo DE

---

**Entradas:**  $N, D, F, CR$

**Saída:**  $S_{melhor}$

---

1.  $População(x_{i,G}) \leftarrow Aleatório(N, D)$
2. **enquanto**  $\neg$  Critério de Parada
3.  $x_{r1}, x_{r2}, x_{r3} \leftarrow Aleatório(População(x_{i,G})); r_1 \neq r_2$   
 $\neq r_3 \neq i$
4.  $v_{i,G+1} \leftarrow Mutação(x_r, F)$
5.  $u_{i,G+1} \leftarrow Recombinação(x_{i,G}, v_{i,G+1}, CR)$
6. **se**  $f(u_{i,G+1}) \leq f(x_{i,G})$
7.  $S_{melhor} \leftarrow u_{i,G+1}$
8. **caso contrário**
9.  $S_{melhor} \leftarrow x_{i,G}$
10. **fim**
11. **fim**
12. **Retorna**  $S_{melhor}$

---

Fonte: o autor, 2018

Além do esquema apresentado anteriormente, existem inúmeros outros que são úteis. A classificação das variantes se dá pela notação  $DE/x/y/z$ , onde  $x$  especifica o modo de mutação, que pode ser *rand* (aleatório) ou *best* (o elemento de menor custo da população atual),  $y$  é o número de elementos de diferença e  $z$  é o modo de recombinação (*bin*).

Por essa notação, o método clássico é o  $DE/rand/1/bin$  (Storn e Price, 1997), mas vale citar que uma variante interessante (em termos de diversificação da população, se a mesma é suficientemente grande) é  $DE/best/2/bin$ , cuja mutação se dá de acordo com a equação 2.5 (Storn, 1996).

$$v_{i,G+1} = x_{best,G} + F * (x_{r1,G} + x_{r2,G} - x_{r3,G} - x_{r4,G}) \quad (2.5)$$

### 2.3.2 Genetic Algorithms

GA é uma técnica adaptativa de otimização global que pertence ao grupo dos algoritmos evolutivos, ramo da computação evolutiva. Foi inspirado pela genética e evolução populacionais e faz uso de suas estruturas (como cromossomos, genes e alelos) e seus mecanismos (como recombinação e mutação), mimetizando a

sobrevivência do mais apto, conceito que é extraído do modelo de seleção natural de Darwin (Eberhart e Shi, 2007; Karray e Silva, 2004; Kruse *et al.*, 2016; Steiner *et al.*, 2015; Zgurovsky e Zaychenko, 2017).

A uma população de  $N$  elementos, inicializada aleatoriamente, são aplicados operadores de seleção, *crossover* e mutação que, ao longo de um número pré-definido de iterações (gerações), criam novas soluções a partir da população prévia e preservam os elementos mais aptos (em termos de máximo ou mínimo), de acordo com o valor da função objetivo, também chamada *fitness* (Steiner *et al.*, 2015).

A Figura 2.5 apresenta o pseudocódigo do algoritmo GA, onde  $D$  representa o número de parâmetros do modelo, ou dimensão do problema ( $\mathbb{R}^D$ ), e  $MR$  e  $CR$  são as probabilidades de mutação e *crossover*, respectivamente. Os critérios de parada dizem respeito ao número máximo de iterações ou valor alvo a ser atingido para o *fitness* (Brownlee, 2011; Engelbrecht, 2007; Kruse *et al.*, 2016).

Figura 2.5 – Formalização do algoritmo GA

---

**Entradas:**  $N, D, MR, CR$

**Saída:**  $S_{melhor}$

---

1.  $População \leftarrow Aleatório(N, D)$
2.  $S_{melhor} \leftarrow Avalia(População)$
3. **enquanto**  $\neg$  Critério de Parada
4.      $Pais \leftarrow Seleção(População, N)$
5.     **para** Pais
6.          $Criança \leftarrow Cruzamento(Pais, CR)$
7.          $Criança \leftarrow Mutação(Criança, MR)$
8.     **fim**
9.      $S_{melhor} \leftarrow Avalia(Criança)$
10.     $População \leftarrow S_{melhor}$
11. **fim**
12. **Retorna**  $S_{melhor}$

---

Fonte: o autor, 2018

Entre os métodos de seleção dos pais tem-se o da amostragem universal estocástica, também chamado método da roleta, onde cada indivíduo recebe uma probabilidade de ser selecionado proporcional à sua aptidão (*fitness*), ou seja, quanto mais adequado à solução do problema, maior sua probabilidade de ser selecionado.

Na etapa de recombinação, ou cruzamento, os pais são particionados em determinados pontos de corte, determinados por  $CR$ , e têm suas coordenadas (para o



problema de agrupamento) permutadas, dando origem a filhos que, a seguir, sofrem mutação nos pontos determinados a partir de  $MR$ , visando aumentar a diversidade de soluções disponíveis.

### 2.3.3 Particle Swarm Optimization

PSO é uma técnica de otimização que se caracteriza por varrer o espaço de busca se utilizando de um enxame de partículas. Pertence aos campos da inteligência de enxames e inteligência coletiva, ramos da inteligência computacional. Foi originalmente inspirada pelo comportamento social de alguns animais, como bandos de pássaros e cardumes de peixes (Kruse *et al.*, 2016).

A busca que o algoritmo realiza consiste em  $N$  indivíduos explorando as vizinhanças do enxame até certa distância e retornando informações aos outros membros sobre descobertas realizadas. Pode ser visto como um método que combina busca baseada no gradiente com busca baseada em população, o que requer que a função a ser otimizada seja do tipo  $f : \mathbb{R}^D \rightarrow \mathbb{R}$  (Brownlee, 2011; Engelbrecht, 2007; Kruse *et al.*, 2016). A Figura 2.6 exibe o pseudocódigo do algoritmo PSO.

No pseudocódigo o parâmetro  $w$  corresponde ao fator de inércia, normalmente decrescido ao longo das iterações,  $C_1$  e  $C_2$  são duas constantes definidas pelo pesquisador, que determinarão o peso dado à informação proveniente do indivíduo e do enxame, respectivamente, e  $\phi_1$  e  $\phi_2$  são valores aleatórios compreendidos no intervalo  $[0,1]$ , responsáveis por adicionar diversidade à solução.  $P_{p\text{melhor}}$  e  $P_{g\text{melhor}}$  são a memória da melhor solução alcançada, na iteração anterior, de cada partícula e pelo enxame como um todo, respectivamente.

Figura 2.6 – Formalização do algoritmo PSO

---

	<b>Entradas:</b> $N, D, w, C_1, C_2$
	<b>Saída:</b> $P_{gmelhor}$

---

1.  $P_{gmelhor} \leftarrow \infty$
2. **para**  $i = 1:N$
3.      $P_{velocidade} \leftarrow 0$
4.      $P_{posição} \leftarrow \text{Aleatório}(N, D)$
5.      $P_{pmelhor} \leftarrow P_{posição}$
6.     **se**  $f(P_{posição}) \leq f(P_{gmelhor})$
7.          $P_{gmelhor} \leftarrow P_{posição}$
8.     **fim**
9. **fim**
10. **enquanto**  $\neg$  Critério de Parada
11.     **para**  $i = 1:N$
12.         **se**  $f(P_{posição}) \leq f(P_{pmelhor})$
13.              $P_{pmelhor} \leftarrow P_{posição}$
14.         **fim**
15.         **se**  $f(P_{posição}) \leq f(P_{gmelhor})$
16.              $P_{gmelhor} \leftarrow P_{posição}$
17.         **fim**
18.          $P_{velocidade} \leftarrow w * P_{velocidade} + \phi_1$   
   $* C_1(P_{pmelhor} - P_{posição}) + \phi_2$   
   $* C_2(P_{gmelhor} - P_{posição})$
19.          $P_{posição} \leftarrow P_{posição} + P_{velocidade}$
20.     **fim**
21. **fim**
22. **Retorna**  $P_{gmelhor}$

---

Fonte: o autor, 2018

Essas três MHs serão utilizadas na otimização dos centroides iniciais do algoritmo *FCM*, apresentado a seguir, e dos pesos iniciais das RNAs, apresentado na seção 2.5.

## 2.4 FUZZY C-MEANS

O algoritmo *FCM* foi escolhido por ser amplamente utilizado em diversas aplicações, estendendo a capacidade do algoritmo *K-means* para conjuntos de dados que possuem como característica a incerteza, a ambiguidade ou a dificuldade em gerar agrupamentos consistentes (Stetco, Zeng e Keane, 2015). Entretanto, o

algoritmo *FCM* sofre com o alto custo computacional, pois para cada iteração a complexidade computacional do algoritmo é quadrática ao número de *clusters*, ou seja,  $O(NC^{2P})$ , onde  $N$  é o número de instâncias,  $C$  é o número de agrupamentos e  $P$  é a dimensionalidade dos dados (Stetco, Zeng e Keane, 2015).

No agrupamento *fuzzy*, ou *soft*, cada instância possui graus de participação, no intervalo  $[0,1]$ , em cada um dos *clusters*, de modo que a soma total das participações, para cada instância, é igual a “1”. Os graus de participação compõem a matriz de partição *fuzzy*, cujos valores serão determinados a partir da minimização da função objetivo do problema.

Tal função objetivo é denominada *FCM functional*, e é a base para uma ampla família de algoritmos de agrupamento *fuzzy* (Babuška, 1998; Bezdek, 1981; Dunn, 1974; Lu, Shang e Li, 2017), baseados na equação 2.6. O valor desta função pode ser entendido como uma medida da variância total entre cada instância e o centro de seu respectivo *cluster* — o centroide (Babuška, 1998). A minimização desta função representa um problema de otimização não-linear, que pode ser solucionado de inúmeras maneiras, entre elas: algoritmos genéticos (Babu e Murty, 1994), *simulated annealing* (Desarbo, 1982) e *grouped coordinate minimization* (Bezdek *et al.*, 1987; Hathaway e Bezdek, 1991). A mais popular, entretanto, é a *FCM* (Babuška, 1998; Zhou, Yang e Shao, 2017).

$$Z = \sum_{i=1}^N \sum_{j=1}^C \mu_{ij}^m \|x_i - c_j\|^2 \quad (2.6)$$

onde  $\mu_{ij} \in [0,1]$  é a matriz de partição *fuzzy* ( $\sum_{j=1}^C \mu_{ij} = 1 \forall_i$ ), cujo parâmetro  $m$  determina a sobreposição (*fuzziness*) dos *clusters* resultantes, podendo variar entre “1” (nenhuma sobreposição, com o algoritmo deixando de ser *fuzzy*) e infinito (total sobreposição e, conseqüentemente, indefinição de a qual *cluster* pertence cada ponto), e  $\|x_i - c_j\|^2$  é a distância euclidiana quadrada entre as instâncias  $x_i$  e os centroides  $c_j$ .

O consenso entre os autores é um valor de  $m$  compreendido no intervalo  $[1.25, 4]$ , com  $m = 2$  sendo o valor geral (Ozkan e Turksen, 2007; Wu, 2012; Zhou, Fu e Yang, 2014; Zhou, Yang e Shao, 2017).

O algoritmo *FCM* pode ser formalizado como se segue na Figura 2.7 (Bezdek,

1981; Bezdek *et al.*, 1987; Bezdek, Ehrlich e Full, 1984; Dunn, 1974; Zhou, Yang e Shao, 2017). O número de *clusters*  $C$  está compreendido no intervalo  $[2, N]$ , onde  $N$  é o número de instâncias do problema, cabendo ao pesquisador determinar (com ou sem o auxílio de métricas) o número ideal de *clusters* para o problema. O erro  $\varepsilon$  também é determinado pelo pesquisador, de acordo com a aplicação desejada.

Figura 2.7 – Formalização do algoritmo FCM

---

**Entradas:**  $Clusters, m, \varepsilon, Max_{iterações}$

**Saída:**  $\mu_{melhor}$

---

1.  $\mu_0 \leftarrow Aleatório(Clusters, m)$
2. **enquanto**  $\neg$  Critério de Parada
3. 
$$c_j = \frac{\sum_{i=1}^N \mu_{ij}^m x_i}{\sum_{i=1}^N \mu_{ij}^m}$$
4. **se**  $|Custo(Z_t) - Custo(Z_{t-1})| < \varepsilon \vee Max_{iterações}$
5. **Pare**
6. **caso contrário**
7. 
$$\mu_{ij} = 1 / \sum_{k=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{2/m-1}$$
8. **fim**
9.  $\mu_{melhor} \leftarrow \mu_{ij}$
10. **fim**
11. **Retorna**  $\mu_{melhor}$

---

Fonte: o autor, 2018

## 2.5 REDES NEURAIS ARTIFICIAIS

Uma RNA consiste em um modelo computacional que simula o comportamento do neurônio biológico, aprendendo com a alteração de seu estado interno a partir de estímulos do ambiente externo. Segundo Haykin (2009):

Uma rede neural é uma máquina projetada para modelar o modo como o cérebro desempenha determinada tarefa ou função de interesse.

A mimetização do cérebro se dá em dois aspectos:

1. O conhecimento é adquirido pela rede do ambiente através de um processo de aprendizado.

2. A força das conexões entre neurônios, conhecida como peso sináptico, é utilizada para armazenar o conhecimento adquirido.

Para atingir o desempenho esperado, as RNAs empregam interconexões de células computacionais simples, referidas como neurônios ou unidades de processamento.

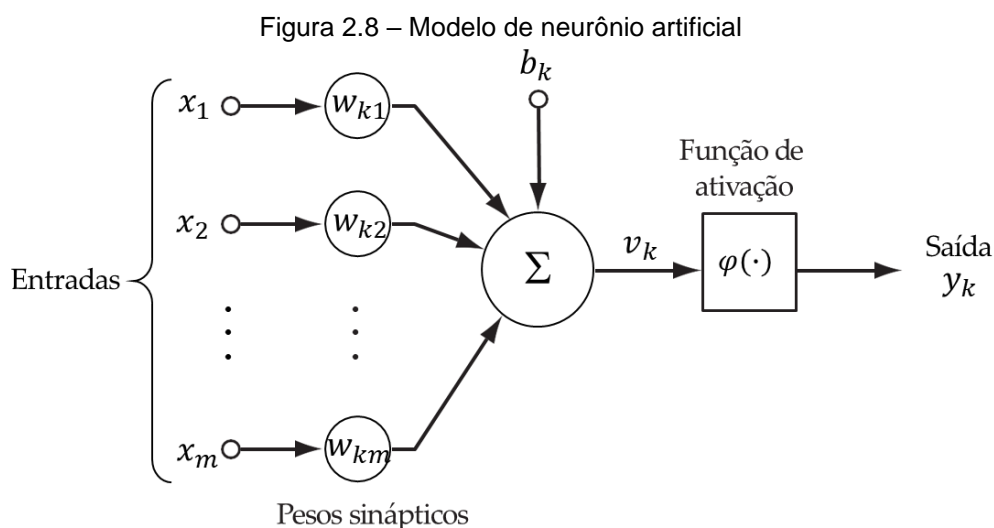
Pode-se entender uma RNA como uma máquina de aprendizado adaptativa, feita de uma rede de processadores distribuídos em paralelo e que é propensa a armazenar conhecimento experimental.

### 2.5.1 Histórico das Redes Neurais Artificiais

O primeiro modelo de um neurônio artificial foi proposto por McCulloch e Pitts em 1943 no trabalho intitulado *A Logical Calculus of the Ideas Immanent in Nervous Activity*. E o primeiro estudo sobre o aprendizado de redes neurais foi proposto por Donald Hebb (1949), que ficou conhecido como a regra de Hebb. Posteriormente foi proposta por Widrow e Hoff (1960) uma nova regra de aprendizado, a regra delta, que ainda é utilizada.

O *Perceptron*, em sua forma simples, apareceria em 1958, proposto por Frank Rosenblatt (1958). Em 1986 seria aplicado o algoritmo *backpropagation* (algoritmo por retropropagação do erro) no contexto das redes neurais por Rumelhart, Hinton e Williams (1986).

Um neurônio artificial pode ser representado como na Figura 2.8.



Em um neurônio artificial pode-se identificar três elementos:

- a) as sinapses que recebem os sinais de entrada;
- b) o somatório para adicionar os sinais de entrada com seus respectivos pesos sinápticos;
- c) a função de ativação, que restringe a amplitude da saída do neurônio.

Além desses elementos, há também a polarização (viés), que atua aumentando ou diminuindo a entrada da função de ativação.

Matematicamente, o neurônio artificial  $k$  da Figura 2.8 pode ser descrito pelas equações (2.7) e (2.8):

$$v_k = \sum_{j=1}^m w_{kj}x_j + b_k \quad (2.7)$$

$$y_k = \varphi(v_k) \quad (2.8)$$

onde  $x_1, x_2, \dots, x_m$  são os sinais de entrada;  $w_{k1}, w_{k2}, \dots, w_{km}$  são os pesos sinápticos;  $v_k$  é o campo local induzido;  $b_k$  é o viés;  $\varphi(\cdot)$  é a função de ativação; e  $y_k$  é o sinal de saída.

A função de ativação  $\varphi(\cdot)$ , que restringe a saída do neurônio, pode ser do tipo limiar (2.9), linear (2.10), sigmoide (2.11) ou tangente hiperbólica, entre outras.

$$\varphi(x) = \begin{cases} 1, & \text{se } x \geq 0 \\ 0, & \text{se } x < 0 \end{cases} \quad (2.9)$$

$$\varphi(x) = x \quad (2.10)$$

$$\varphi(x) = \frac{1}{1 + e^{-ax}} \quad (2.11)$$

onde o parâmetro  $a$  relaciona-se à declividade da curva, sendo diretamente proporcional ao aumento de sua inclinação.

Neurônios artificiais, isoladamente, possuem baixa capacidade computacional, mas conectados em uma rede são capazes de resolver problemas de grande

complexidade (Braga, Carvalho e Ludermir, 2007). Basicamente, podem apresentar uma única camada ou múltiplas camadas, e serem alimentadas para frente (*feedforward*) ou serem recorrentes (*feedback*), quando o sinal de saída volta para a entrada da rede ou para a entrada de uma camada intermediária.

Em uma RNA, o aprendizado ocorre à medida que os pesos sinápticos são ajustados com base em alguma regra pré-estabelecida, como a regra delta (ou regra de Widrow-Hoff). Nas palavras de Braga, Carvalho e Ludermir (2007):

aprendizado é o processo pelo qual os parâmetros livres de uma rede (pesos e vieses) são ajustados por meio de uma forma continuada de estímulo pelo ambiente externo, sendo o tipo específico de aprendizado definido pela maneira particular como ocorrem os ajustes dos parâmetros livres.

Matematicamente, pode-se definir a atualização dos pesos da rede na iteração  $n$  pela equação (2.12):

$$w_{kj}(n + 1) = w_{kj}(n) + \Delta w_{kj}(n) \quad (2.12)$$

onde  $w_{kj}$  representa os pesos do neurônio  $k$  e  $\Delta w_{kj}$  os ajustes dos pesos, todos com  $0 < j < m$  ( $j = 0$  representa o viés).

Os algoritmos de aprendizado diferem na forma como  $\Delta w_{kj}$  é calculado. De acordo com a regra delta, o ajuste  $\Delta w_{kj}$  aplicado ao peso sináptico  $w_{kj}$  é definido pela equação (2.13) (Haykin, 2009):

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n) \quad (2.13)$$

onde  $\eta$  é a taxa de aprendizado,  $e_k$  é o sinal de erro, ou seja, a diferença entre o valor esperado  $d_j$  e o sinal efetivo de saída da rede  $y_k$ , dado pela equação (2.14), e  $x_j$  é o sinal de entrada.

$$e_k(n) = d_j(n) - y_k(n) \quad (2.14)$$

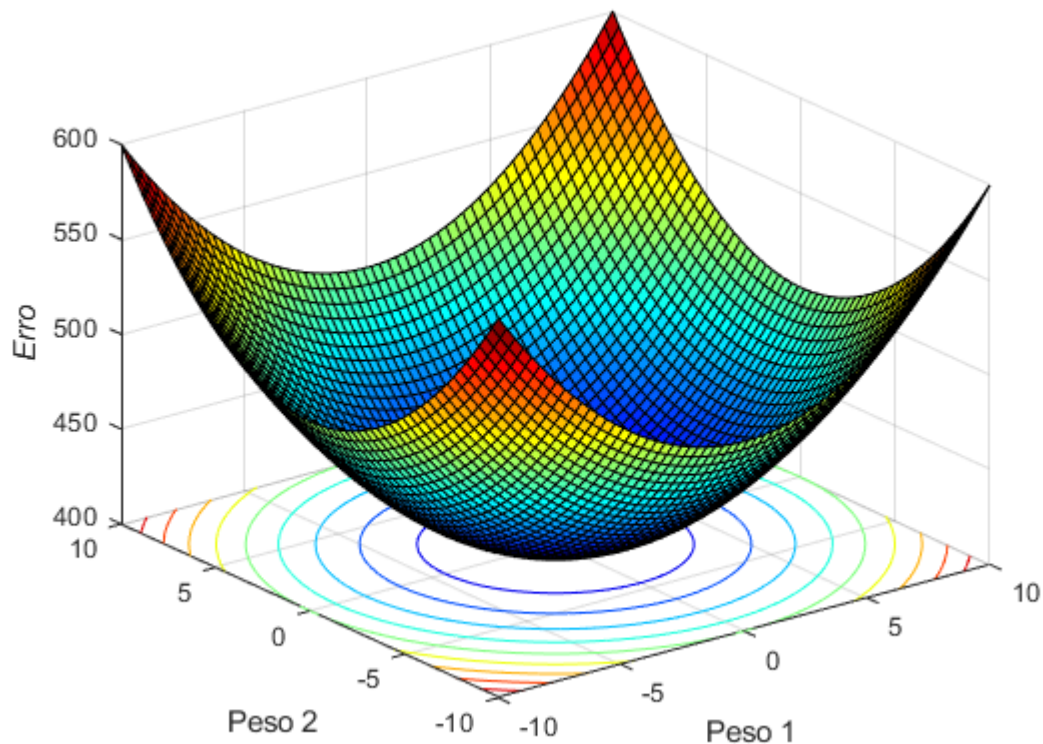
Segundo Haykin (2009), “os ajustes corretivos dos pesos são projetados para aproximar passo a passo o sinal de saída  $y_k$  da resposta desejada  $d_j$ ”. Para tanto, usa-se minimizar a função do erro quadrático médio (2.15).

$$MSE = \frac{1}{N} \sum_{k=1}^N e_k^2(n) \quad (2.15)$$

para os  $N$  exemplos de treinamento.

A partir do valor do  $MSE$ , o conjunto de dados formado pelos pares de entrada e saída  $[x_j; d_j]$  define a superfície de erro. No caso da rede formada por unidades de processamento lineares, a superfície de erro é dada por uma função quadrática dos pesos da rede (Figura 2.9). Para uma rede formada por unidades de processamento não-lineares, a superfície de erro poderá ter uma forma irregular e vários mínimos locais, além do mínimo global (Figura 2.10).

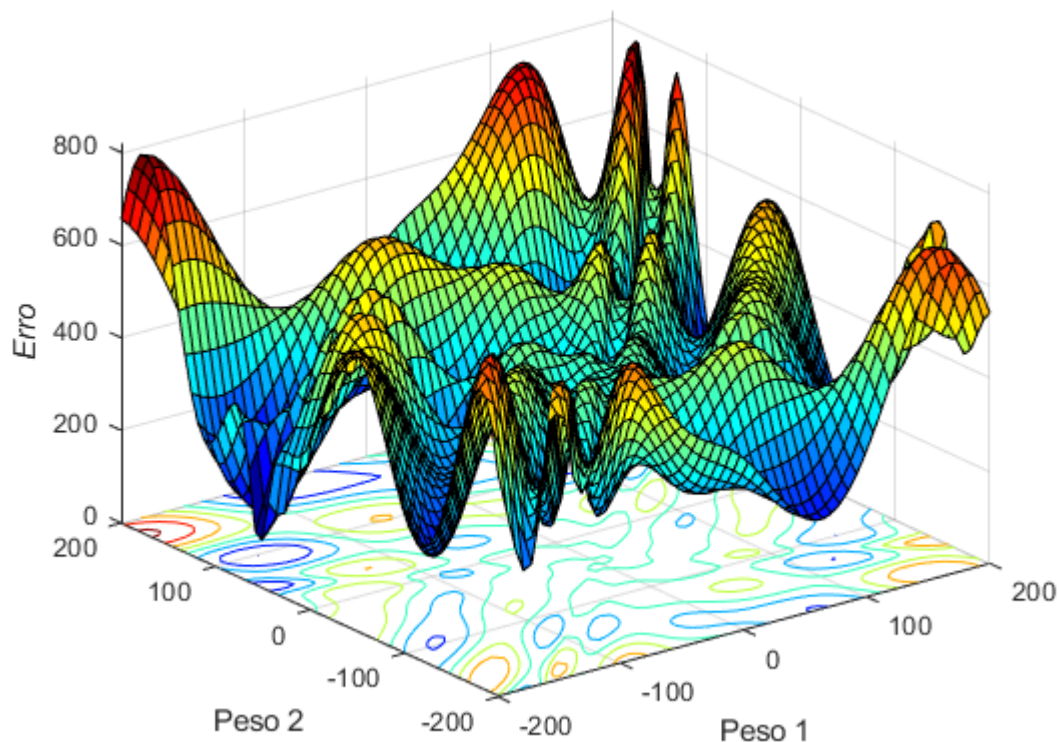
Figura 2.9 – Superfície de erro para neurônios lineares



Fonte: adaptado de: BRAGA; CARVALHO e LUDERMIR, 2007



Figura 2.10 – Superfície de erro para neurônios não-lineares



Fonte: adaptado de: BRAGA; CARVALHO e LUDERMIR, 2007

Segundo Haykin (2009), “o *perceptron* de camada única é a forma mais simples de uma rede neural usada para a classificação de padrões linearmente separáveis”. Outro ponto a respeito do *Perceptron* de camada única é destacado por Braga, Carvalho e Ludermir (2007): “sabe-se que, independentemente do valor  $\eta$ , haverá convergência em um tempo finito, caso as classes sejam linearmente separáveis”.

De um modo resumido tem-se, a seguir, o algoritmo do *Perceptron* de camada única, com  $k$  neurônios (Braga, Carvalho e Ludermir, 2007; Haykin, 2009):

#### Passo 1. Inicialização

- a) faça o vetor de pesos sinápticos  $w_k(n) = 0$  e os vieses  $b_k(n) = 0$ .

#### Passo 2. Ativação

- a) execute o somatório  $v_k(n) = \sum_{j=1}^m w_{kj}(n)x_j(n) + b_k$ , onde  $m$  é a quantidade de pesos sinápticos para o neurônio  $k$ ;
- b) execute a função de ativação  $y_k = \varphi(v_k(n))$ , onde  $\varphi(\cdot)$  é uma função de limiar, linear ou sigmoide, entre outras;

c) calcule o sinal de erro  $e_k(n) = d_j(n) - y_k(n)$ , onde  $d_j$  é o valor esperado.

### Passo 3. *Atualização dos pesos*

a) calcule o novo peso sináptico  $w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n)$ , onde  $\Delta w_{kj}(n) = \eta e_k(n)x_j(n)$ , com  $0 < \eta < 1$ ,  $x_j(n)$  e  $0 < j < m$  ( $w_{k0} = b_k$  e  $x_0 = 1$ ) correspondendo aos sinais de entrada.

### Passo 4. *Análise da situação atual*

a) após todos os exemplos de treinamento terem passado pela rede uma única vez finda-se a 1ª iteração e calcula-se o erro global  $\varepsilon(n) = \frac{1}{N} \sum_{k=1}^N e_k^2(n)$ , para os  $N$  exemplos de treinamento.

### Passo 5. *Continuação*

a) incremente a iteração  $n$  em uma unidade e volte ao passo 2.

O procedimento deverá continuar até que um erro  $\varepsilon$  suficientemente pequeno seja encontrado.

## 2.5.2 O *Perceptron* de múltiplas camadas

Um *Perceptron* de múltiplas camadas (*Multilayer Perceptron; MLP*), nada mais é do que uma generalização do *Perceptron* de camada única (HAYKIN, 2009). Pode-se descrevê-lo como uma rede constituída pela camada de entrada, uma ou mais camadas ocultas e uma camada de saída; rede essa que apresenta um alto grau de conectividade (Braga, Carvalho e Ludermir, 2007; Haykin, 2009). Seu treinamento se dá em três etapas: a alimentação para frente (*feedforward*) da rede com os padrões de entrada, o cálculo e retropropagação (*backpropagation*) do erro, e o ajuste dos pesos.

Para uma rede *MLP* tem-se o algoritmo de treinamento como o representado na sequência (Haykin, 2009).

#### Propagação *forward*:

a) inicialize os pesos sinápticos  $w_{ji}$  e vieses  $b_j$  (da camada oculta), e pesos sinápticos  $w_{kj}$  e vieses  $b_k$  (da camada de saída) com valores aleatórios;

- b) execute o somatório  $v_j(n) = \sum_{i=1}^m w_{ji}x_i(n) + b_j(n)$  para cada neurônio  $j$  da camada oculta, onde  $x_i$  são os dados de entrada da rede;
- c) calcule a função de transferência  $y_j(n) = \varphi_j(v_j(n))$  para cada neurônio  $j$  da camada oculta, onde  $\varphi_j(\cdot)$  é, geralmente, uma função sigmoide;
- d) na camada de saída, execute o somatório  $v_k(n) = \sum_{j=1}^m w_{kj}y_j(n) + b_k(n)$  para cada neurônio  $k$  da camada de saída, caso haja mais de um;
- e) então calcule a função de transferência  $y_k(n) = \varphi_k(v_k(n))$  para cada neurônio  $k$  da camada de saída, onde  $\varphi_k(\cdot)$  é, geralmente, uma função linear;

Propagação *backward*:

- a) calcule o erro da camada de saída utilizando a equação  $\delta_k(n) = (d_k(n) - y_k(n))\varphi'_k(v_k(n))$ , onde  $d_k$  é o valor esperado e  $\varphi'_k(\cdot)$  é a derivada da função de transferência do neurônio de saída;
- b) ajuste os pesos sinápticos da rede na camada de saída de acordo com a regra delta generalizada:

$$w_{kj}(n+1) = w_{kj}(n) + \alpha (w_{kj}(n-1)) + \eta \delta_k(n) y_j(n)$$

onde  $\eta$  é a taxa de aprendizagem e  $\alpha$  é a constante de *momentum*.

- c) calcule o erro da camada oculta fazendo  $\delta_j(n) = \varphi'_j(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$ , onde  $\varphi'_j(\cdot)$  é a derivada da função de transferência do neurônio oculto;
- d) ajuste os pesos sinápticos da rede na camada oculta de acordo com a regra delta generalizada:

$$w_{ji}(n+1) = w_{ji}(n) + \alpha (w_{ji}(n-1)) + \eta \delta_j(n) y_i(n)$$

Após todos os exemplos de treinamento terem passado pela rede finda-se a 1ª iteração e calcula-se o erro global  $\varepsilon(n) = \frac{1}{N} \sum_{k=1}^N e_k^2(n)$ , para os  $N$  exemplos de treinamento, onde  $e_k(n) = d_k(n) - y_k(n)$ .

As iterações devem prosseguir até que um valor  $\varepsilon$  suficientemente pequeno seja encontrado, sendo assim, o processo de aprendizagem da rede neural pode ser visto como um problema de minimização com função objetivo  $\varepsilon$  no espaço de  $w$ .

### 2.5.3 Treinamento por Levenberg-Marquardt

O algoritmo de *Levenberg-Marquardt* foi projetado para se aproximar, em velocidade, aos métodos de treinamento de segunda ordem sem ter que computar a matriz Hessiana (Levenberg, 1944; Marquardt, 1963). Quando a função de desempenho tem a forma de uma soma de quadrados (como é típico do treinamento de redes neurais *feedforward*), então a matriz Hessiana pode ser aproximada pela equação (2.16).

$$H \approx J^T J \quad (2.16)$$

e o gradiente pode ser computado pela equação (2.17), tal que

$$g = J^T e \quad (2.17)$$

onde  $J$  é a matriz Jacobiana que contém as derivadas primeiras dos erros da rede neural em relação aos pesos e vieses, e  $e$  é o vetor dos erros da rede. A matriz Jacobiana pode ser calculada através do algoritmo *backpropagation* padrão, que é muito menos complexo do que computar a matriz Hessiana.

A atualização dos pesos no algoritmo de *Levenberg-Marquardt* é dada pela equação (2.18).

$$w_{k+1} = w_k - [J^T J + \lambda I]^{-1} J^T e \quad (2.18)$$

onde  $\lambda$  é um escalar denominado parâmetro de *damping* e  $I$  é a matriz identidade.

Quando  $\lambda$  é igual a zero, este se torna o método de Newton, usando a aproximação da matriz Hessiana. Quando  $\lambda$  aumenta, este tende ao método do gradiente descendente com tamanho de passo pequeno. O método de Newton é mais rápido e mais preciso nas proximidades do erro mínimo, então o objetivo é se deslocar para o método de Newton o mais rápido possível. Então  $\lambda$  é decrementado após cada iteração com sucesso (onde o erro diminui), ou incrementado caso contrário.

### 2.5.4 Teoria da Regularização Bayesiana

Outro método para aperfeiçoar a generalização da rede neural é a chamada regularização. Esta técnica envolve modificar a função desempenho, que normalmente é uma soma quadrática dos erros da rede (Bühlmann e van de Geer, 2011).

Geralmente a função desempenho usada para treinar uma rede *feedforward* é o erro quadrático médio entre os valores esperados e os valores encontrados pela rede (2.15).

É possível melhorar a generalização ao se modificar a função desempenho pela adição de um termo  $m_{sw}$  que consiste da média da soma quadrática dos pesos e vieses da rede, como mostrado pela equação (2.19).

$$m_{sereg} = \beta * mse + \alpha * m_{sw} \quad (2.19)$$

onde  $\beta$  e  $\alpha$  são parâmetros da função objetivo e  $m_{sw}$  é dado pela equação (2.20).

$$m_{sw} = \frac{1}{N} \sum_{j=1}^N (w_j + b)^2 \quad (2.20)$$

Esta função desempenho minimiza a soma quadrática dos pesos e vieses da rede, e isto força a resposta ser mais suave e menos suscetível ao sobreajuste.

O problema com a regularização Bayesiana é a determinação dos parâmetros  $\beta$  e  $\alpha$ , pois se  $\alpha \ll \beta$  o algoritmo de treinamento produzirá erros menores, porém com tendência ao sobreajuste, e se  $\alpha \gg \beta$  o treinamento enfatizará a redução do quadrado da soma dos pesos e vieses da rede, às custas do erro, produzindo uma resposta mais “suave” e menos suscetível ao sobreajuste.

Uma solução para este problema é a chamada regularização Bayesiana, que consiste na aplicação do teorema de Bayes à otimização dos parâmetros  $\beta$  e  $\alpha$  (Foresee e Hagan, 1997; MacKay, 1992).

## 2.6 TRABALHOS CORRELATOS

A seguir é apresentada uma análise bibliométrica relacionada ao tema aqui proposto e, na sequência, são apresentados os artigos mais relevantes relacionados ao assunto de problemas de classificação e aplicação de MHs à otimização de RNA.

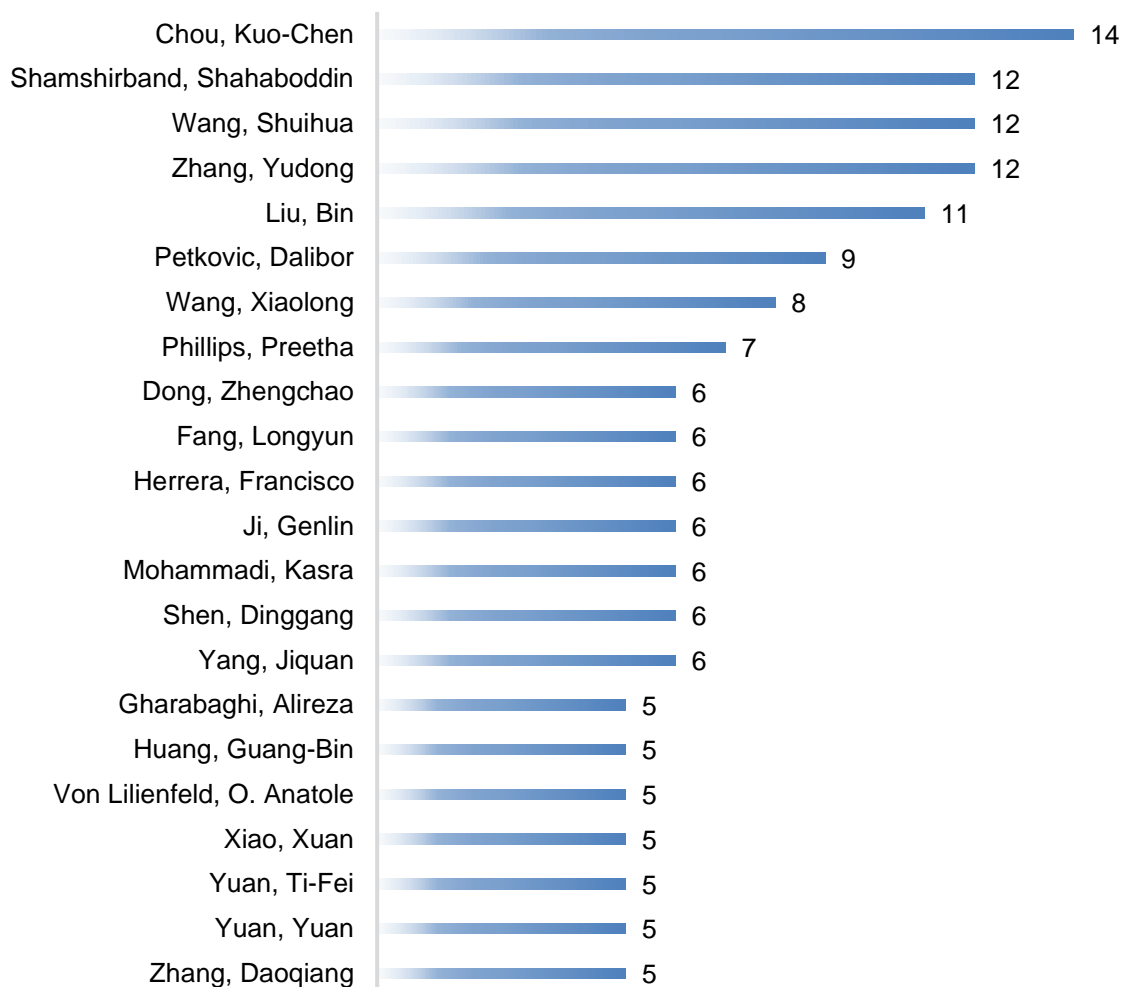
### 2.6.1 Análise bibliométrica

Visando obter um panorama geral da área de KDD, foi realizada a coleta de artigos científicos nas bases de dados *Scopus*, *Science Direct* e *Web of Science*, onde foram pesquisados os termos *computational intelligence*, *soft computing*, *natural computing* e *machine learning*. Foram selecionados os 1.000 artigos mais citados, publicados no período 2010-2018. Deve ser ressaltado que o número de publicações foi contabilizado de acordo com o número de autores por artigo.

Para o processamento da base de dados coletada utilizou-se a plataforma de análise estatística e social *NAILS – Network Analysis Interface for Literature Studies* (Knutas *et al.*, 2015). Uma vez que o número de artigos científicos publicados anualmente tem crescido exponencialmente (Bornmann e Mutz, 2015), faz-se necessária uma análise automatizada das tendências de pesquisa em nível global, com o objetivo de identificar áreas com maior potencial para exploração e descoberta.

A seguir, na Figura 2.11, é mostrada a lista de autores mais produtivos, com relação ao número de artigos publicados, no período analisado.

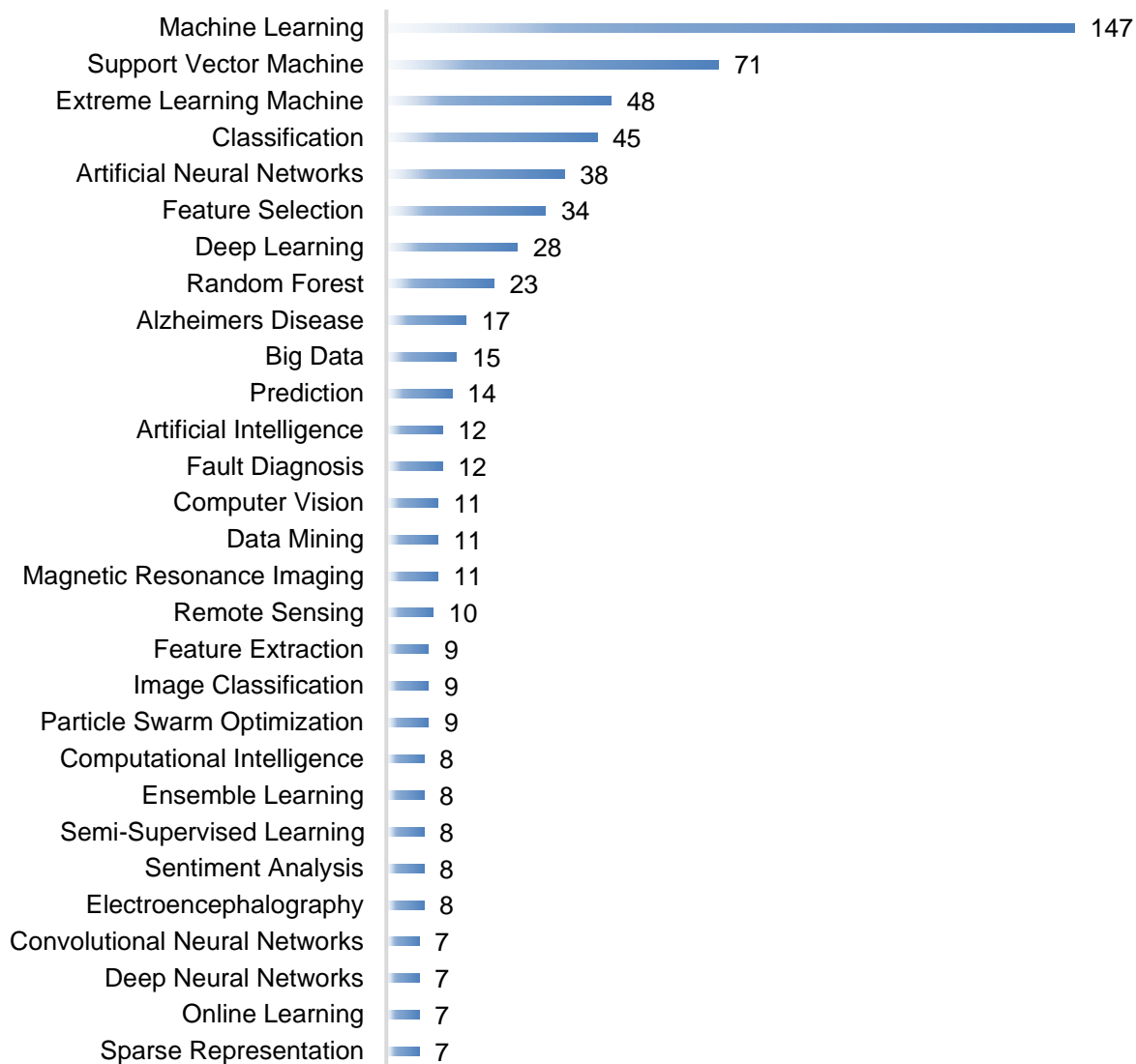
Figura 2.11 – Autores mais produtivos (2010-2018)



Fonte: o autor, 2018

Para os artigos analisados, foi extraída a lista de palavras-chave citadas pelos autores. A Figura 2.12 lista os termos mais utilizados no período. Embora tenham sido encontradas mais de 4.000 palavras-chave diferentes, as aqui apresentadas são as que possuem pelo menos 7 ocorrências (cerca de 0,17% do total).

Figura 2.12 – Palavras-chave mais citadas (2010-2018)



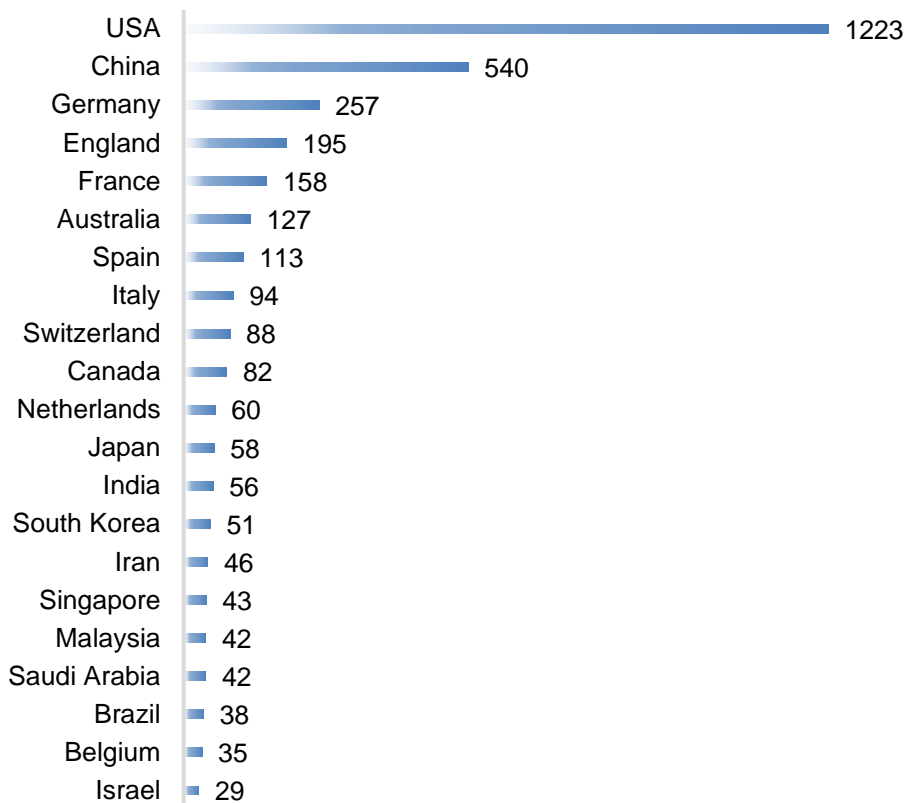
Fonte: o autor, 2018

Quanto aos países-sede dos autores das publicações, tem-se uma discrepância significativa, com Estados Unidos (33% do total, ou 1.223 publicações) e China (14% do total, ou 540 publicações) sendo os responsáveis por quase metade das publicações mundiais (47% do total). As Figuras 2.13 e 2.14 apresentam os países responsáveis por 90% do total de publicações. O Brasil responde por cerca de 1% do total do período (38 publicações).

Na Tabela 2.3 foram listados alguns algoritmos de otimização inspirados na natureza, pertencentes à área de IC, assim como as suas respectivas inspirações e referências primárias. Esta não é uma apresentação exaustiva, uma vez que o número de técnicas inspiradas na natureza cresce diariamente, e faz-se necessária constante atualização.

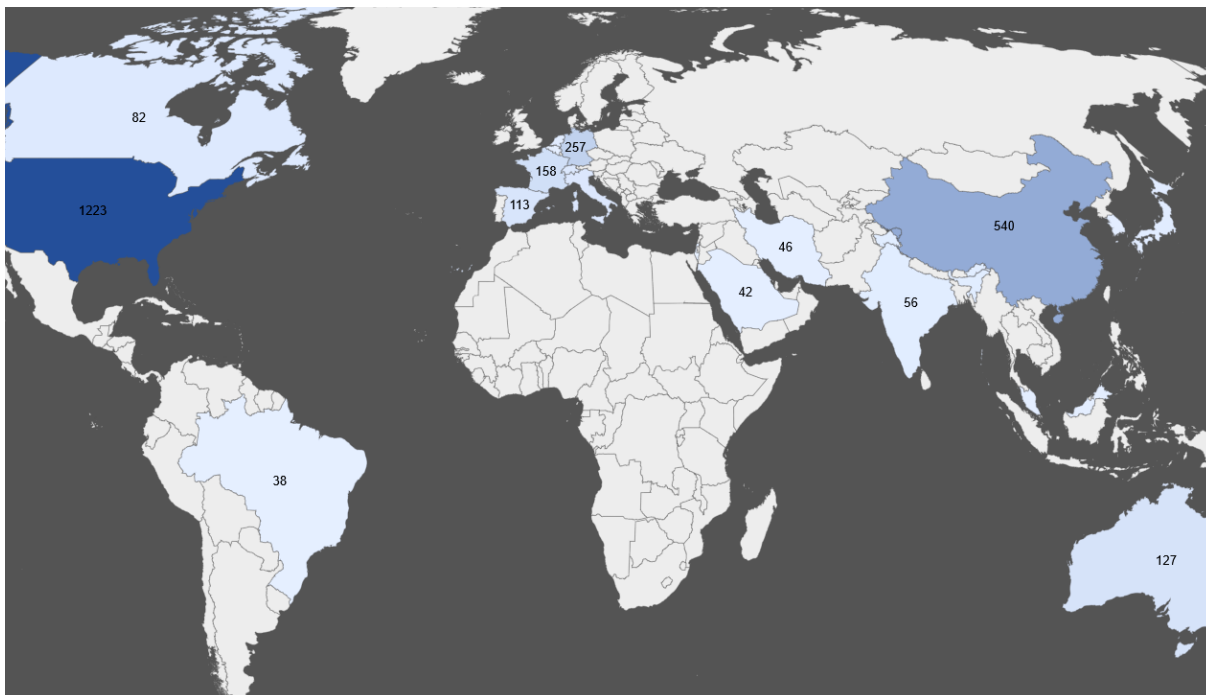


Figura 2.13 – Publicações por países (2010-2018)



Fonte: o autor, 2018

Figura 2.14 – Mapa das publicações (2010-2018)



Fonte: o autor, 2018

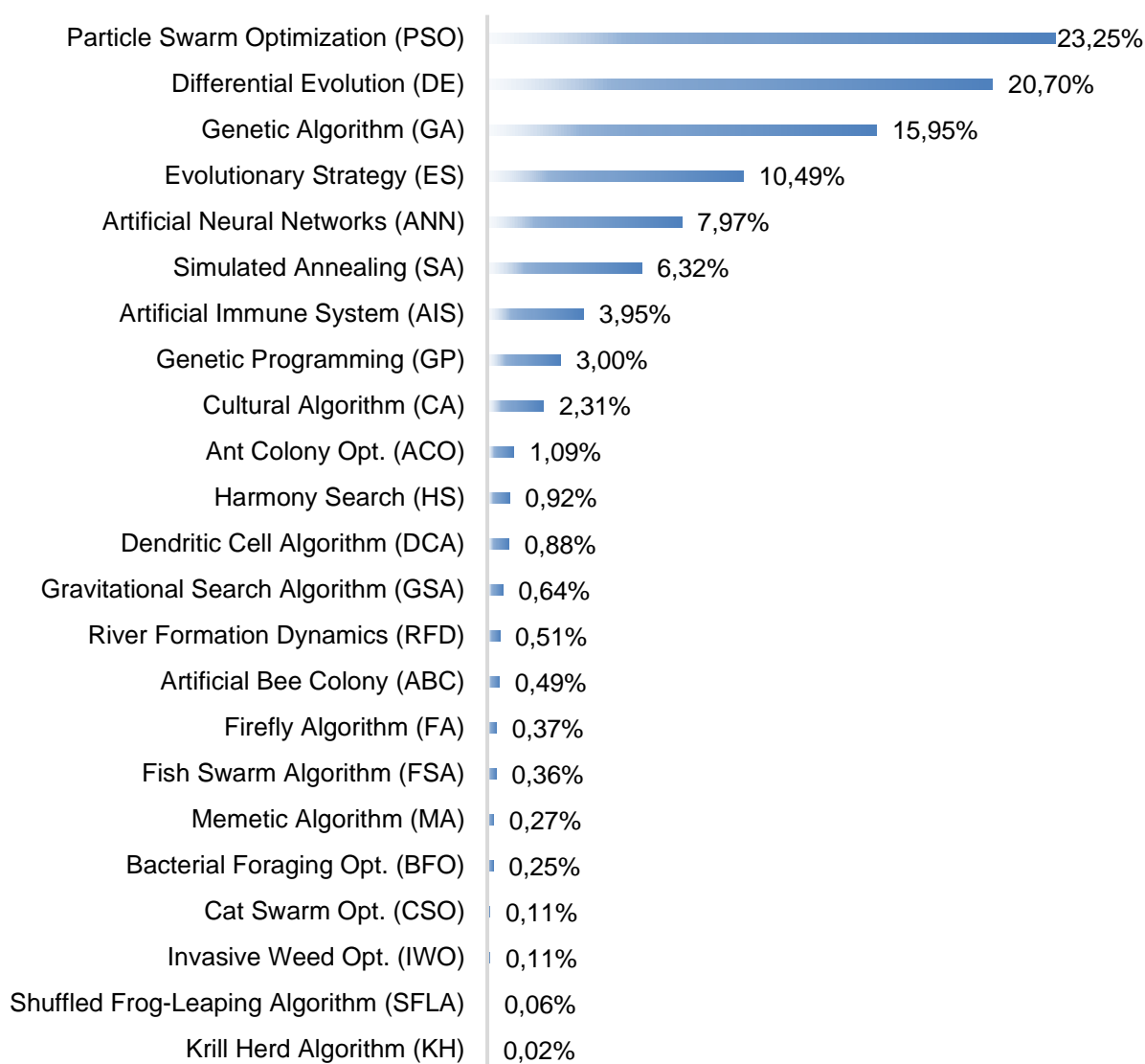
Tabela 2.3 – Algoritmos de otimização inspirados na natureza

Inspiração	Algoritmo	Referência primária
Biologia	Artificial Immune System (AIS)	(Dasgupta, 1999)
	Artificial Neural Networks (ANN)	(Rosenblatt, 1958)
	Bacterial Foraging Opt. (BFO)	(Passino, 2002)
	Dendritic Cell Algorithm (DCA)	(Greensmith, Aickelin e Twycross, 2006)
Enxames ou Populações	Ant Colony Opt. (ACO)	(Dorigo, Maniezzo e Colorni, 1996)
	Artificial Bee Colony (ABC)	(Karaboga e Basturk, 2007)
	Cat Swarm Opt. (CSO)	(Xing e Gao, 2014b)
	Firefly Algorithm (FA)	(Yang, 2009)
	Fish Swarm Algorithm (FSA)	(Li, Shao e Qian, 2002)
	Invasive Weed Opt. (IWO)	(Mehrabian e Lucas, 2006)
	Krill Herd Algorithm (KH)	(Gandomi e Alavi, 2012)
Particle Swarm Opt. (PSO)	(Kennedy e Eberhart, 1995)	
Evolução	Differential Evolution (DE)	(Storn e Price, 1997)
	Evolutionary Strategy (ES)	(Beyer e Schwefel, 2002)
	Genetic Algorithm (GA)	(Goldberg, 1989; Holland, 1992)
	Genetic Programming (GP)	(Koza, 1994)
Física	Cultural Algorithm (CA)	(Reynolds, 1994)
	Gravitational Search Algorithm (GSA)	(Rashedi, Nezamabadi-pour e Saryazdi, 2009)
	Harmony Search (HS)	(Geem, Kim e Loganathan, 2001)
	Memetic Algorithm (MA)	(Moscato e Cotta, 2010)
	River Formation Dynamics (RFD)	(Rabanal, Rodríguez e Rubio, 2007)
	Simulated Annealing (SA)	(Kirkpatrick, Gelatt e Vecchi, 1983)
Shuffled Frog-Leaping Algorithm (SFLA)	(Eusuff e Lansey, 2003)	

Fonte: o autor, 2018

A partir dos artigos analisados foram levantadas as MHs mais citadas no período, o que pode ser visto na Figura 2.15 como percentual de artigos onde a MH é citada. A partir deste levantamento, as três MHs mais citadas foram selecionadas para comporem a metodologia de otimização dos pesos iniciais das RNAs e dos centroides iniciais do algoritmo *FCM*.

Figura 2.15 – Meta-heurísticas mais citadas entre 2010-2018



Fonte: o autor, 2018

A seguir são apresentados os artigos que, dentre os levantados na revisão sistemática, possuíam o maior número de citações. Estes artigos compõem o estado da arte do que há de mais relevante dentro do tema pesquisado neste trabalho.

### 2.6.2 Estado da arte

Com o avanço rápido da área de IC nos anos recentes, muitas abordagens têm sido implementadas no aperfeiçoamento das RNAs. A seguir serão apresentados alguns exemplos recentes do estado da arte de tais técnicas.

### 2.6.2.1 Redes Neurais Artificiais

Explorando a área de mecânica estrutural, Stoffel *et al.* (2018) empregaram RNAs na predição de deformações complexas em *shock-wave loaded plates*, sujeitos a não-linearidades geométricas e físicas de amplo alcance. Além da RNA, para predição de todo o sistema, também utilizaram um modelo híbrido, unindo RNA e equações de deformação viscoplástica a um sistema de elementos finitos. O modelo RNA conseguiu bons resultados apenas no domínio dos dados de treinamento sem, no entanto, conseguir extrapolar para outras amplitudes. O modelo híbrido, RNA e elementos finitos, atingiu melhores resultados e com menor esforço computacional do que o modelo de elementos finitos clássico.

Fontanella *et al.* (2018) implementaram RNAs na calibração do viés de temperatura de giroscópios micro eletromecânicos de sistema de *GPS (Global Positioning System)*. Usualmente esta calibração se dá por interpolação polinomial, porém esta técnica não se adequa bem a modelos de mudança abrupta de tendências em pequenos intervalos (histerese do sensor). Foi então proposto o uso de RNAs como um aperfeiçoamento da predição que, comparativamente ao modelo polinomial, obtiveram um desempenho 20% superior.

Na área de bioinformática, Sudha *et al.* (2018) utilizaram uma RNA nas predições de *Protein Fold Recognition (PFR)* e *Structural Class (SCP)*. Tal rede calcula seus pesos e vieses por meio de técnicas de agrupamento (*k-nearest neighbors, KNN*), com base na distância euclidiana entre as entradas e os alvos. O algoritmo proposto superou as técnicas *Support Vector Machine (SVM)* e as redes *MLP* nos quatro conjuntos de dados testados: *DD* (695 sequências de aminoácidos), *EDD* (3418 sequências), *TG* (1612 sequências) e *RDD* (691 sequências).

Abordando o problema de modelagem da poluição atmosférica urbana, Alimissis *et al.* (2018) empregaram redes MLP e *Multiple Linear Regression (MLR)* em dados reais da região metropolitana de Atenas, Grécia. O objetivo foi estimar a concentração de cinco poluentes (dióxido de nitrogênio, monóxido de nitrogênio, ozônio, monóxido de carbono e dióxido de enxofre) em locais não monitorados pela interpolação dos dados de regiões monitoradas. Na maioria dos testes, a RNA se mostrou significativamente superior (com erros até 63% menores que os do modelo *MLR*), principalmente em condições onde a rede de monitoramento era menos densa, levando ao decréscimo da correlação espacial entre os sítios monitorados.

Khoshroo *et al.* (2018) utilizaram uma RNA treinada por *Gradient Descent with Momentum (GDM)* na predição da produção de uvas (kg), utilizando como variáveis de entrada o número de horas de mão de obra e maquinário, o uso de produtos químicos (inseticidas, fungicidas e herbicidas, em kg), o uso de estrume de gado (kg), a quantidade de óleo diesel consumido (em litros), a eletricidade consumida (kWh) e a água para irrigação (m<sup>3</sup>). Na sequência foi conduzida uma análise de sensibilidade (*Partial Rank Correlation Coefficient, PRCC*) em 100 amostras geradas pelo método *Latin Hypercube Sampling (LHS)* e alimentadas na rede, indicando o tipo de maquinário que gera o maior impacto na produção, seguido por diesel e mão de obra.

Em *Network Intrusion Detection Systems (NIDS)*, Shenfield *et al.* (2018) conseguiram uma acurácia média de 98% na distinção entre tráfico benigno e maligno em redes de computadores. A arquitetura ótima da RNA foi obtida por busca exaustiva do número de camadas ocultas e seus respectivos neurônios com *10-fold cross-validation*. O algoritmo de treinamento adotado foi o *Resilient Backpropagation* com pesos inicializados pela heurística de *Glorot* (Glorot e Bengio, 2010).

Fazendo uma análise do uso de RNA em sistemas de energia eólica, Marugán *et al.* (2018) descobriram que os problemas que podem ser eficazmente resolvidos pelas mesmas são: previsão da velocidade do vento e potência (com horizontes de tempo variando entre segundos e semanas), detecção de falhas de caixa de engrenagens e rolamentos, gerador, eletrônica, rotor, hidráulica e lâminas, e otimização e controle em monitoramento de potência e velocidade. Além disso, utilizando o *MLP*, os autores conseguiram superar as técnicas *SVM*, *RFIA (Random Forest Algorithm)* e *GBRT (Gradient Boosted Regression Trees)* em problemas de previsão da direção do vento.

Na área de Engenharia Civil, Leśniak e Juszczak (2018), empregaram a *MLP* treinada pelos algoritmos *LM*, *Conjugate Gradient (CG)* e *Broyden-Fletcher-Goldfarb-Shanno (BFGS)* na predição, em estágio prematuro de implantação da obra, dos custos indiretos em canteiros de obra visando uma técnica que fosse, ao mesmo tempo, rápida e confiável. Os testes foram conduzidos em um conjunto de 143 projetos de construção concluídos e conseguiram uma precisão de 84,13%, superando o modelo de regressão linear múltipla, com precisão de 54,14%.

Em uma aplicação de RNA para previsão no campo das energias renováveis, especificamente de chaminé solar (armazenamento de energia solar em forma térmica), Fadaei *et al.* (2018) conseguiram um erro relativo médio de 1,83% e máximo

de 2,15% na predição da temperatura do ar na saída da chaminé utilizando como entrada do modelo, a temperatura da superfície de absorção da radiação solar, um modelo não-linear *input-output*, uma vez que as séries empregadas são fortemente correlacionadas. Foram utilizadas 1445 amostras colhidas em um dia (intervalo de 1 minuto entre cada amostra, aproximadamente), das quais 924 foram utilizadas para treinamento (64%), 231 para validação (16%), responsável por evitar o sobreajuste da RNA e 290 para teste (20%). O algoritmo de treinamento utilizado foi o *LM* e a RNA possuía duas camadas ocultas com 4 e 5 neurônios.

No contexto do desenvolvimento sustentável, Hao *et al.* (2018) utilizaram um modelo híbrido combinando *Grey Model (GM)*, *Triple Exponential Smoothing (TES)* e RNA otimizada por enxame de partículas para predição do número de veículos em final de vida útil a serem reciclados na cidade de Xangai, China. Foram utilizadas 11 séries históricas compreendidas no período 2005-2016, todas referentes à cidade de Xangai: volume e preço da reciclagem de veículos, número de empregados, investimento em ativos fixos e *networking* na indústria de reciclagem, preços dos materiais recicláveis, número de proprietários de veículos, PIB (Produto Interno Bruto), população, malha rodoviária e renda *per capita*. O modelo proposto obteve um Erro Absoluto Médio Percentual (*Mean Absolute Percentage Error, MAPE*) igual a 0,096, contra 0,63 para o GM e 0,22 para o TES.

#### 2.6.2.2 Clusterização

Por outro lado, com relação a problemas de agrupamento, é feito a seguir um breve levantamento da evolução das técnicas empregadas em diversas áreas do conhecimento ao longo dos últimos anos.

Em 2010, Bailador e Triviño empregaram o conceito *fuzzy automata* (Doostfatemeheh e Kremer, 2005), com a inclusão da dimensão temporal, no reconhecimento de gestos da mão humana sem a necessidade de pré-processamento dos dados. O modelo é temporal uma vez que permite a inclusão de restrições temporais que modelem a duração de diferentes estados *fuzzy*, o que torna possível lidar com ambiguidade, uma vez que o autômato pode estar em vários estados ao mesmo tempo.

Também utilizando uma abordagem *fuzzy*, desta vez em reconhecimento não-supervisionado de danos estruturais em edificações, Chen e Zang (2011) criaram uma

estratégia híbrida que envolve *Artificial Immune System (AIS)* e *FCM*. Aplicado a uma base de dados coletados por sensores o algoritmo conseguiu superar as técnicas *Naive Bayes* e *Support Vector Machine (SVM)*.

Comparando os métodos estatísticos (*SVM* e *k-Nearest Neighbor*) e estruturais (*Graph Classification*) de reconhecimento de padrões, Bunke e Riesen (2012) avaliaram que o segundo conseguiu superar ou, pelo menos, se igualar ao primeiro, uma vez que permite representar objetos em função de suas partes. Os autores ainda concluem que há oportunidade de integração destes dois métodos, conduzindo a possíveis melhores resultados.

No ano seguinte foi proposta uma nova medida de similaridade para algoritmos *fuzzy* de classificação (Yen, Fan e Chao, 2013). A proposta satisfaz aos axiomas de medidas de similaridade propostos por Mitchell (2003) e foi capaz de resolver o problema até então não solucionado de Hung e Yang (2004).

Em um cenário de detecção de gases contaminantes no ar, utilizando sensores eletrônicos de baixo custo, Zhang *et al.* (2014) propuseram um híbrido entre *Kernel-based Principal Component Analysis (KPCA)* e *Linear Discriminant Analysis (LDA)*. A técnica foi testada com seis amostras de gases poluentes, combinados dois a dois (estratégia *OAO*, *one-against-one*), totalizaram 15 classificadores binários. Os resultados mostraram ser possível o uso de sensores de baixo custo neste tipo de aplicação, com taxas de acerto que permitem aplicações em situações reais. Os autores concluem que, para se determinar o nível de contaminação, é preciso utilizar modelos de regressão, como redes neurais (Zhang e Tian, 2014).

Sever (2015), identificando a relação entre a teoria da regularização para problemas inversos, como as de reconhecimento de padrões, propuseram um algoritmo de aprendizado baseado na regularização (*RBA – Regularization-based Algorithm*) de Tikhonov e na Transformada de Riesz, que superou a Máquina de Vetores de Suporte (*SVM*) para os conjuntos de dados descontínuos utilizados.

Adereçando o problema da indisponibilidade de suficientes conjuntos de treinamento para reconhecimento de padrões em cartas de controle, Gutierrez e Pham (2016) propuseram uma nova metodologia (*PGS – Pattern Generation Scheme*) para geração destes conjuntos de modo que sejam diversificados e comparáveis, quando aplicados a problemas da indústria. O foco do trabalho foi treinar os algoritmos *SVM* e *Probabilistic Neural Network (PNN)* de modo a identificar alterações nos parâmetros do processo que possuíssem uma causa definida. Esta nova abordagem conseguiu

atingir resultados significativamente melhores quando comparados ao previamente existente na literatura (Barghash e Santarisi, 2004).

Finalmente, em 2017, Ali *et al.* compararam métodos estatísticos de reconhecimento de padrões, entre os quais *Naive Bayes*, *LDA*, *Kernel Classifier*, *Linear Least Mean Squares*, *Quadratic Least Mean Squares* e Regressão Logística Multinomial, e propuseram um método híbrido meta-heurístico-estatístico (*PSO-LDA*) que, testado em 12 conjuntos da literatura, obteve melhor desempenho em 11 deles.

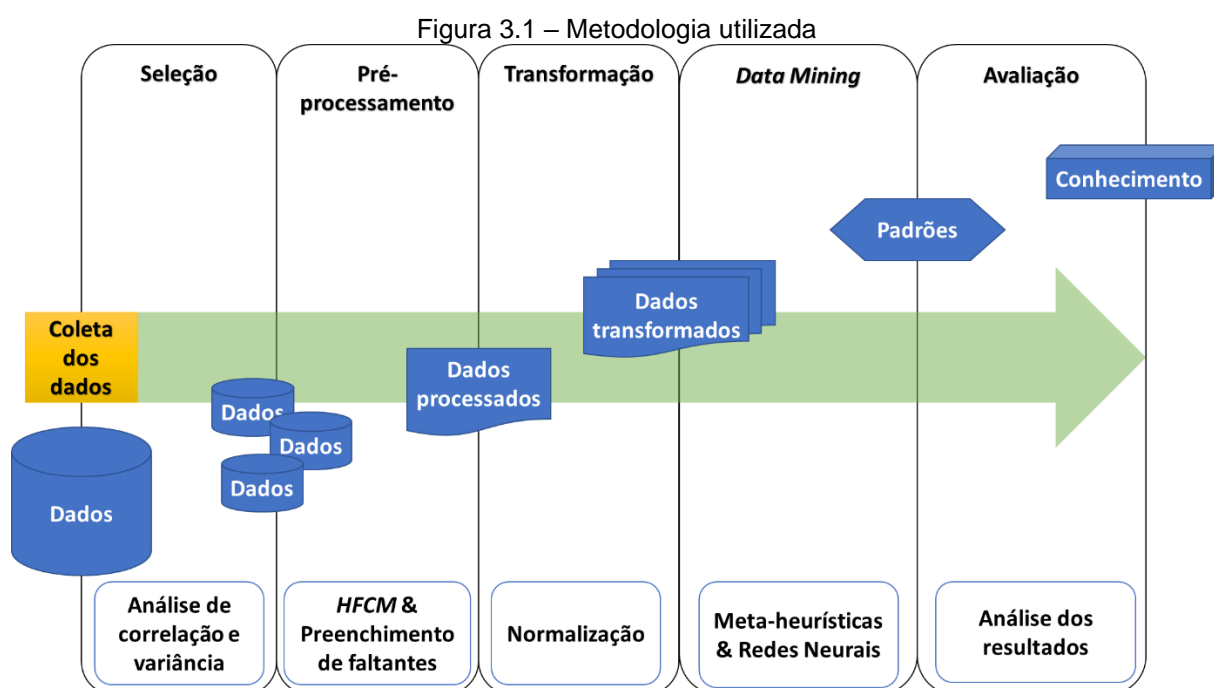
A partir deste levantamento foi identificado o que há de mais atual e eficiente em se tratando das RNAs e do problema de clusterização, permitindo a correta definição da metodologia que melhor se adapte ao objetivo do trabalho. Esta metodologia será apresentada no capítulo seguinte, que abordará o problema da energia solar, da seleção e pré-processamento dos dados que serão utilizados no modelo, o que inclui o uso inédito do algoritmo *HFCM* no problema de preenchimento dos dados faltantes, e a nova proposta de otimização dos pesos iniciais das RNAs.



### 3 METODOLOGIA

Neste capítulo são descritas as ferramentas e métodos envolvidos no desenvolvimento da técnica de otimização da RNA e, também, a elaboração do algoritmo *HFCM*, empregado na etapa de pré-processamento, no problema de preenchimento dos dados faltantes da base de dados de energia solar.

A Figura 3.1, concebida a partir da Figura 2.1, esquematiza o fluxo de informações do método empregado.



Fonte: o autor, 2018

O processo inicia-se com o acesso aos dados a serem investigados, que podem estar dispersos em vários bancos de dados. É errônea a pressuposição de que, para um modelo efetivo, necessitem-se milhões de instâncias de dados (Bühlmann e Geer, van de, 2011). De fato, o que se necessita é que os dados sejam representativos do problema tratado e que não contenham erros de mensuração, por exemplo (não sejam enviesados). Uma vez que o problema esteja definido, a primeira etapa é selecionar as variáveis que irão compor o modelo e construir um subconjunto relevante para a análise. Nesta fase foram removidas as variáveis correlacionadas do modelo.

Na fase seguinte, de pré-processamento, os dados foram corrigidos com relação a incompletudes e ruídos e, na sequência, padronizados, uma vez que cada

variável possuía intervalos com diferentes ordens de magnitude, abordagem obrigatória para RNAs.

Para a fase de *DM*, as MHs *DE*, *GA* e *PSO* foram adaptadas ao problema de classificação. A otimização promovida por estas técnicas visa definir o melhor conjunto de pesos iniciais para a RNA. Na sequência, esse conjunto de pesos sinápticos será novamente otimizado, desta vez pelo algoritmo *LM*.

Serão conduzidos experimentos com outras seis bases de dados para validação desta metodologia e os resultados serão analisados com relação aos objetivos iniciais da pesquisa.

### 3.1 FERRAMENTAS E MÉTODOS

São descritos a seguir o problema de classificação referente à energia solar, principal aplicação desta tese, assim como as ferramentas e métodos utilizados durante o desenvolvimento do trabalho.

#### 3.1.1 O problema da energia solar

Com o crescimento da população mundial e da demanda por recursos naturais, principalmente energéticos, os problemas de destruição do meio ambiente estão se tornando frequentes e tais recursos estão se tornando raros (Miao *et al.*, 2017; Tõnurist, 2015; Vivanco, Kemp e Voet, van der, 2015). Esse cenário está levando muitos países a buscarem alternativas que evitem ou tentem mitigar ao máximo os efeitos indesejados de seus sistemas produtivos, como poluição e degradação da fauna e flora, sem com isso afetar o crescimento econômico ou até mesmo impulsionando a economia de maneira sustentável (Lee, Park e Kim, 2015; Lindman e Söderholm, 2016).

Áreas abandonadas, que possuem substâncias potencialmente perigosas ao meio ambiente ou à saúde humana, estão se tornando o centro de uma das inúmeras preocupações mundiais (Bergius e Öberg, 2007; Greenberg e Lewis, 2000; Hartmann *et al.*, 2014; Li *et al.*, 2017; Straalen, van, 2002). Entre estas áreas de risco, no contexto norte-americano, tem-se: 1) minas abandonadas, que incluem *spoil banks* (pilhas de rejeitos) e plantas de processamento de metais, geralmente contaminadas por metais pesados (Kovacs e Szemmelveisz, 2017); 2) *brownfields* (campos marrons), que podem ser definidos como instalações industriais ou comerciais que

apresentam dificuldades para reutilização, devido à presença de substâncias perigosas, poluentes ou contaminantes (U.S. Government Publishing Office, 2002); 3) locais que se enquadram no *Superfund*, programa do governo federal norte-americano voltado para a localização e limpeza de áreas contaminadas com substâncias perigosas ou poluentes (U.S. Government Publishing Office, 2015); 4) aterros sanitários, que nos países desenvolvidos incluem, basicamente, restos de alimentos e embalagens (Rong *et al.*, 2017); e 5) áreas abrangidas pela legislação do *Resource Conservation and Recovery Act (RCRA)*, de destinação de rejeitos sólidos (U.S. Government Publishing Office, 2011). A Figura 3.2 ilustra cada uma dessas áreas, com as suas numerações correspondentes.

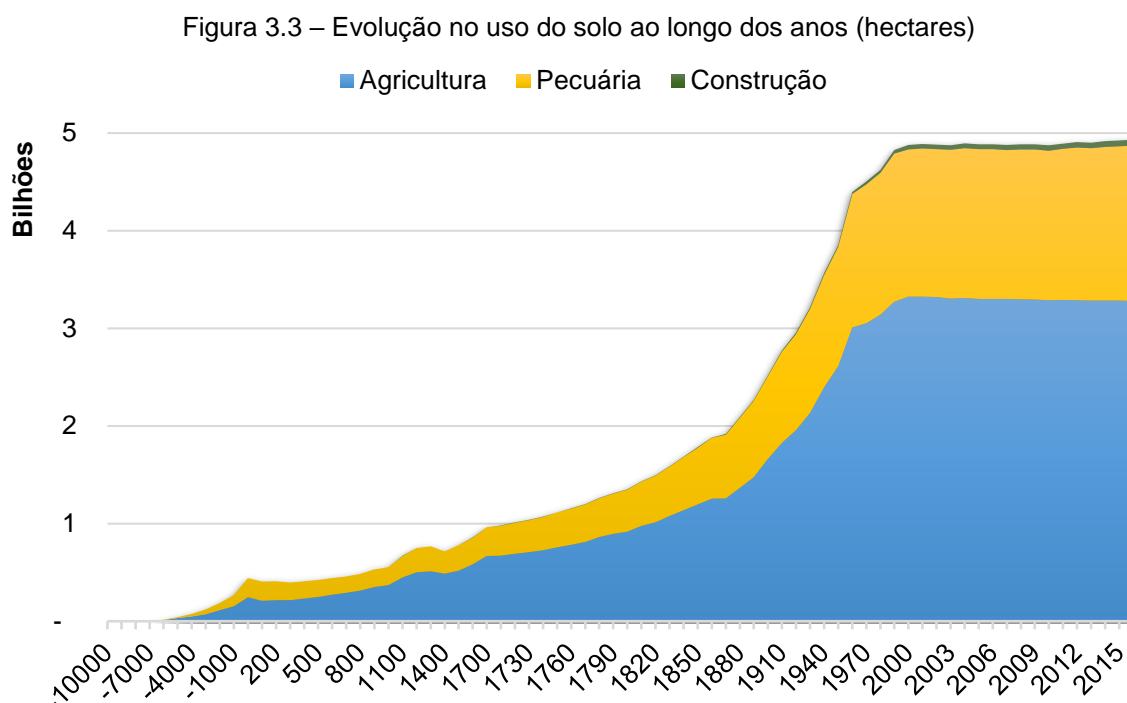
Figura 3.2 – Exemplo de áreas abandonadas



Fonte: o autor, 2018

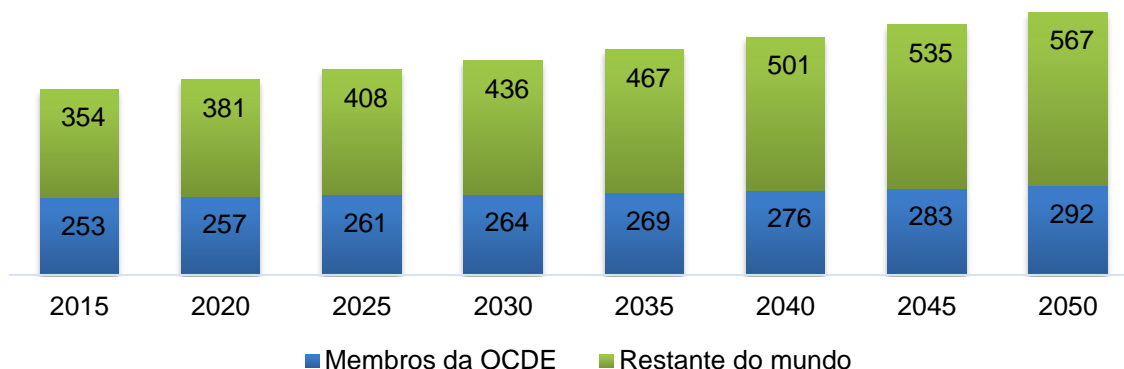
Por outro lado, a crescente ocupação do espaço urbano e rural tem se tornado um problema no mundo moderno (Lambin e Meyfroidt, 2011; Morio, Schädler e Finkel, 2013), o que tem demandado maior eficiência na ocupação territorial (veja a Figura 3.3), principalmente a reutilização de áreas abandonadas, que tem se tornado um dos maiores desafios (Morio, Schädler e Finkel, 2013; Nuissl e Schroeter-Schlaack, 2009; U.S. Government Publishing Office, 2002). Esta situação se agrava ainda mais se tais áreas possuírem considerável tamanho e estiverem contaminadas, gerando, além dos riscos ambiental e de saúde, riscos econômicos (Apostolidis e Hutton, 2006; Cao e

Guan, 2007; Kaufman, Rogers e Murray, 2005; Morio, Schädler e Finkel, 2013; Sousa, de, 2003).



Fonte: adaptado de: Goldewijk *et al.*, 2017

Uma vez que o aumento do consumo de energia vem se acentuando ao longo das últimas décadas (veja a Figura 3.4, que compara os países membros da Organização para a Cooperação e Desenvolvimento Econômico, OCDE, com o restante do mundo) e entra em choque com as implicações do consumo de combustíveis fósseis e a consequente emissão de gases tóxicos e de efeito estufa, fazem-se necessários investimentos em pesquisa e desenvolvimento de novas fontes limpas e renováveis de energia (Almeida *et al.*, 2017; Baños *et al.*, 2011; Cadez e Czerny, 2016; Manan *et al.*, 2017; Manzano-Agugliaro *et al.*, 2012; Perea-Moreno *et al.*, 2017). Desse modo, as energias renováveis, como a solar, por exemplo, têm-se colocado como fortes candidatas na nova corrida por produtividade e bem-estar ambiental e social (González, Gonçalves e Vasconcelos, 2017; Lima, Ferreira e Vieira, 2013), além de estarem em voga nos discursos políticos, empresariais e sociais de uma forma geral (Onat, Kucukvar e Tatari, 2014; Simas e Pacca, 2013). Neste cenário, a energia solar é considerada uma fonte abundante, gratuita e limpa (Fernández-García *et al.*, 2015).

Figura 3.4 – Consumo de energia no mundo ( $10^{18}$  Joules)

Fonte: adaptado de: U.S. Energy Information Administration, 2017

### 3.1.2 Seleção e pré-processamento de dados

O conjunto de dados utilizado no problema (que será aqui chamado de *Solar classification*) foi obtido no site da Agência de Proteção Ambiental dos Estados Unidos (EPA, *United States Environmental Protection Agency*), responsável pelo programa *RE-Powering America's Land*, que identifica áreas abandonadas com potencial para recuperação e implantação de projetos de energia renovável.

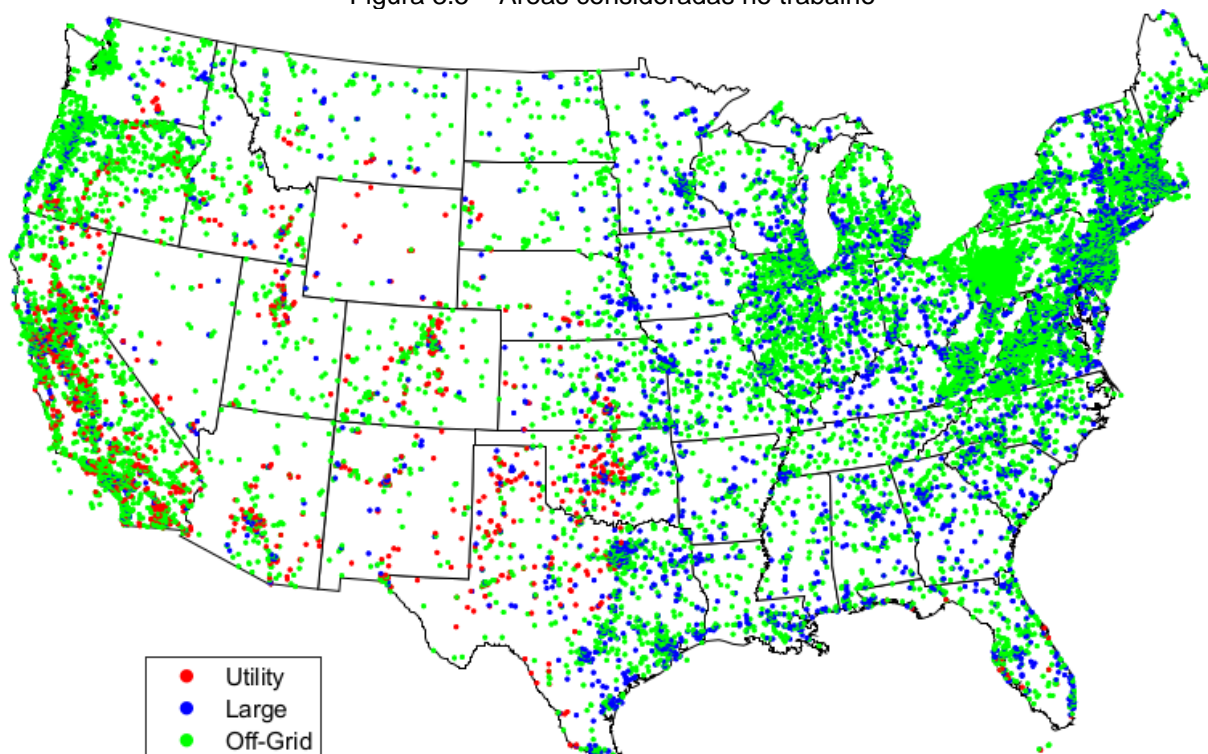
Os dados totalizam 81.533 (Figura 3.5) instâncias, cada qual com 13 variáveis independentes, listadas a seguir, além de 3 variáveis dependentes.

1. Latitude;
2. Longitude;
3. Área, em  $m^2$ ;
4. Incidência solar (DNI, Direct Normal Irradiance), em  $kWh/m^2/dia$ ;
5. Estado da subestação mais próxima (Projeto ou Em serviço);
6. Tensão da subestação mais próxima, em  $kV$ ;
7. Distância até a subestação mais próxima, em milhas;
8. Estado da linha de transmissão mais próxima (Projeto ou Em serviço);
9. Tensão da linha de transmissão mais próxima, em  $kV$ ;
10. Distância até a linha de transmissão mais próxima, em milhas;
11. Distância até a rodovia mais próxima, em milhas;
12. População da área urbana mais próxima;
13. Distância até a área urbana mais próxima, em milhas.

As variáveis dependentes se referem ao potencial do local para uma usina fotovoltaica, que podem ser classificadas em 3 tipos conforme listados a seguir, das quais, 62.092 são do tipo *off-grid*, 17.329 são de grande escala e 2.112 possuem escala de utilidade.

1. *Off-grid*, unidades que normalmente não exportam a energia gerada para o sistema elétrico e cuja incidência solar seja de, pelo menos,  $2,5 \text{ kWh/m}^2/\text{dia}$ ;
2. Grande escala (*Large*), com potência de, pelo menos, 300 quilowatts (kW) em áreas com, no mínimo, 8 mil  $\text{m}^2$ , distando não mais que 1,6 km de linhas de transmissão, e incidência solar mínima de  $3,5 \text{ kWh/m}^2/\text{dia}$ ;
3. Escala de utilidade (*Utility*), que operam na escala de megawatts (MW) em áreas maiores que 160 mil  $\text{m}^2$  onde a disponibilidade de radiação solar seja maior ou igual a  $5 \text{ kWh/m}^2/\text{dia}$  e cuja distância até linhas de transmissão seja de até 16 km.

Figura 3.5 – Áreas consideradas no trabalho



Fonte: o autor, 2018

Na sequência foi realizada análise de correlação, através do coeficiente de correlação de Pearson, entre 11 variáveis, de 3 a 13 (já que as variáveis 1 e 2 que

representam a latitude e a longitude, respectivamente, não foram consideradas, uma vez que não serão utilizadas na análise de classificação). O resultado da análise é apresentado na Tabela 3.1, onde pode-se observar uma correlação de 91% entre as variáveis 10 (“Distância até a linha de transmissão mais próxima”) e 11 (“Distância até a rodovia mais próxima”), destacado em negrito. Para evitar que ocorra multicolinearidade a variável 11 foi, então, eliminada do modelo de classificação.

Tabela 3.1 – Análise de correlação

Variáveis	3	4	5	6	7	8	9	10	11	12	13
3	100%	7%	-3%	0%	2%	-1%	1%	1%	1%	-2%	5%
4	7%	100%	-6%	-15%	-1%	2%	-15%	-3%	-1%	-12%	11%
5	-3%	-6%	100%	-6%	-7%	8%	-5%	-3%	-1%	2%	-5%
6	0%	-15%	-6%	100%	3%	-7%	29%	0%	0%	23%	-8%
7	2%	-1%	-7%	3%	100%	-3%	9%	38%	30%	-17%	55%
8	-1%	2%	8%	-7%	-3%	100%	-23%	-1%	1%	-6%	-1%
9	1%	-15%	-5%	29%	9%	-23%	100%	0%	-1%	21%	-4%
10	1%	-3%	-3%	0%	38%	-1%	0%	100%	91%	-5%	47%
11	1%	-1%	-1%	0%	30%	1%	-1%	91%	100%	-5%	46%
12	-2%	-12%	2%	23%	-17%	-6%	21%	-5%	-5%	100%	-19%
13	5%	11%	-5%	-8%	55%	-1%	-4%	47%	46%	-19%	100%

Fonte: o autor, 2018

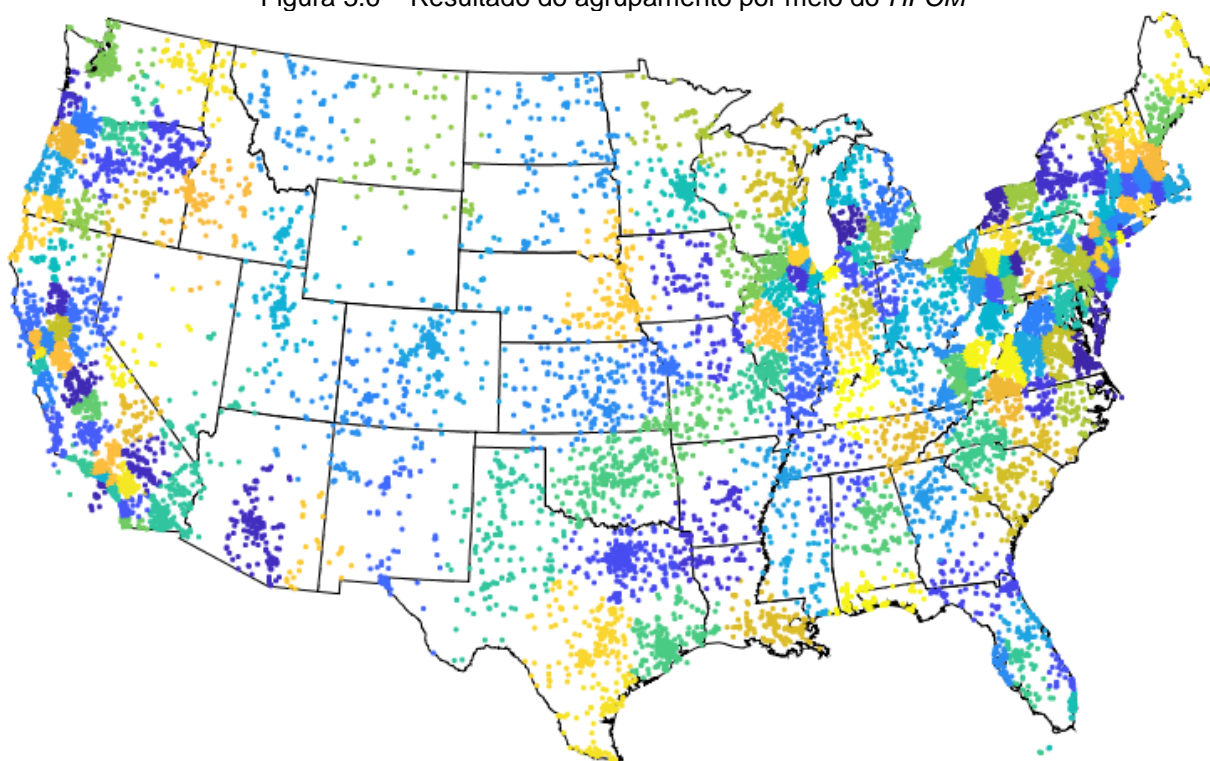
### 3.1.2.1 Preenchimento dos dados faltantes

Dos 81.533 pontos analisados, 32.429 possuíam um ou mais dados faltantes, que precisavam ser preenchidos. Para sanar esta deficiência, poderia ser utilizada a média de cada variável, que poderia gerar discrepâncias, uma vez que a amplitude do conjunto se mostrou significativa. Pensando nisto, foi proposto o agrupamento das instâncias em conjuntos menores (levando em consideração suas coordenadas geográficas: latitude e longitude), utilizando o algoritmo *Hybrid Fuzzy C-Means* (HFCM) (Franco e Steiner, 2018), visando diminuir a amplitude de cada variável para, em seguida, realizar a interpolação linear, suprindo os dados faltantes.

Era necessário que nenhuma variável, em determinada instância de qualquer um dos *clusters*, ficasse vazia e, ao mesmo tempo, que os agrupamentos fossem

pequenos o suficiente para minimizar distorções. O número de agrupamentos, então, foi definido experimentalmente em 200, sendo que o número de instâncias por agrupamento variou entre 42 e 1.863, com uma média de 407. A Figura 3.6 apresenta o resultado do agrupamento.

Figura 3.6 – Resultado do agrupamento por meio do *HFCM*



Fonte: o autor, 2018

Uma vez que o algoritmo *FCM* clássico (Seção 2.4) pressupõe a inicialização aleatória da matriz de partição *fuzzy*, existe a necessidade de um maior número de iterações para alcançar a solução final. Pensando nessa limitação, é aqui proposta a inicialização da matriz de partição difusa por meio de três MHs distintas (detalhadas na Seção 2.3): *DE* (Storn, 1996; Storn e Price, 1997) e *GA* (Goldberg, 1989), enquadradas nas estratégias evolutivas, e *PSO* (Kennedy e Eberhart, 1995; Kennedy, Eberhart e Shi, 2001; Poli, 2008), das estratégias de enxames.

O pseudocódigo do algoritmo resultante é apresentado na Figura 3.7 a seguir. Na linha 1, o termo *Meta – heurística* refere-se à inicialização da matriz de partição *fuzzy* ( $\mu_0$ ) por uma das três MHs: *DE*, *GA* ou *PSO*.



Figura 3.7 – Algoritmo HFCM

---

**Entradas:** *Clusters, m, ε, Maxiterrações*

---

**Saída:**  $\mu_{melhor}$

---

1.  $\mu_0 \leftarrow Meta - heurística(Clusters, m)$
2. **enquanto**  $\neg$  Critério de Parada
3. 
$$c_j = \sum_{i=1}^N \mu_{ij}^m x_i / \sum_{i=1}^N \mu_{ij}^m$$
4. **se**  $|Custo(Z_t) - Custo(Z_{t-1})| < \epsilon \vee Maxiterrações$
5.     **Pare**
6.     **caso contrário**
7. 
$$\mu_{ij} = 1 / \sum_{k=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{2/m-1}$$
8.     **fim**
9.      $\mu_{melhor} \leftarrow \mu_{ij}$
10. **fim**
11. **Retorna**  $\mu_{melhor}$

---

Fonte: o autor, 2018

Na fase de inicialização foi realizada apenas uma iteração e utilizou-se uma população de 4 indivíduos (mínimo de indivíduos para o algoritmo *DE*). Tais valores mínimos visaram acelerar a convergência do algoritmo sem, no entanto, comprometer a qualidade da solução final.

Para a apresentação visual dos resultados, adotou-se o arredondamento simples da matriz de partição difusa, que armazena os graus de participação para cada instância em cada *cluster*, de modo que os valores maiores ou iguais a “0,5” foram arredondados para “1” e os valores menores que “0,5” foram arredondados para “0”. Poder-se-ia adotar que os valores intermediários de graus de participação (por exemplo, entre 0,4 e 0,6) seriam representantes de um terceiro *cluster*, intermediário.

O número de *clusters* foi validado por três métricas consideradas, pela literatura, as mais utilizadas (Arbelaitz *et al.*, 2013): *Calinski-Harabasz Criterion*, *Davies-Bouldin Criterion* e *Silhouette Coefficient*.

O critério de *Calinski-Harabasz*, também chamado de *Variance Ratio Criterion* (*VRC*), pode ser definido como a razão entre a variância média *between-cluster* e a variância média *within-cluster* (equação 3.1). Quanto maior o seu valor, melhor o particionamento dos dados.

$$CH_k = \frac{\sum_{i=1}^k n_i \|c_i - \bar{x}\|^2}{\sum_{i=1}^k \sum_{x \in k_i} \|x - c_i\|^2} * \frac{(N - k)}{(k - 1)} \quad (3.1)$$

onde  $n_i$  é o número de elementos no *cluster*  $i$ ;  $c_i$  é o centroide do *cluster*  $i$ ;  $\bar{x}$  é a média do conjunto de dados;  $N$  é o número de elementos do conjunto de dados;  $k$  é o número de *clusters* e  $x$  é um elemento do conjunto de dados pertencente ao *cluster*  $k_i$ .  $\|\cdot\|$  é a distância euclidiana.

O critério de *Davies-Bouldin* é baseado na razão entre as distâncias *within-cluster* e *between-cluster*, conforme a equação 3.2. Quanto menor seu valor, melhor o resultado.

$$DB_k = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \left\{ \frac{\bar{d}_i + \bar{d}_j}{d_{i,j}} \right\} \quad (3.2)$$

onde  $k$  é o número de *clusters*;  $\bar{d}_i$  é a distância média entre cada ponto do  $i^{th}$  *cluster* e o centroide do respectivo *cluster* (analogamente para  $\bar{d}_j$ ) e  $d_{i,j}$  é a distância euclidiana entre os centroides dos *clusters*  $i$  e  $j$ .

O coeficiente *Silhouette*, para cada ponto do conjunto de dados, é a medida do quão similar aquele ponto é em relação aos outros pontos do mesmo *cluster*, quando comparado aos pontos de outro *cluster*. Seu valor,  $S_i \in [-1, 1]$ , para o  $i^{th}$  ponto é definido conforme a equação 3.3.

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)} \quad (3.3)$$

onde  $a_i$  é a distância média entre o ponto  $i$  e os demais pontos do *cluster* e  $b_i$  é a distância média mínima entre o ponto  $i$  e os pontos dos demais *clusters*.

Se a maioria dos pontos possuírem um valor alto de coeficiente *Silhouette*, então a solução está apropriada.

### 3.1.3 Otimização dos pesos iniciais das RNA

O processo de treinamento proposto consiste em minimizar, utilizando as MHS *DE*, *GA* e *PSO*, a função de erro quadrático médio (*MSE*) entre os valores-alvo da

RNA e os valores preditos durante o processo de aprendizado. O conjunto de pesos e vieses,  $wb$ , corresponderá à posição dos elementos (exame ou população) a ser otimizada pelas MHs.

No pseudocódigo apresentado na Figura 3.8,  $H$  é o número de neurônios da camada oculta,  $N$  é o tamanho da população para a MH em questão e  $D$  é a dimensão do problema (função do número de variáveis dos dados de entrada e alvo, e do número de neurônios da camada oculta).

Figura 3.8 – Algoritmo *MH-MLP*

---

**Entradas:** *entrada, alvo, H, iter, N*

**Saídas:**  $wb_{melhor}$ , *saída*

---

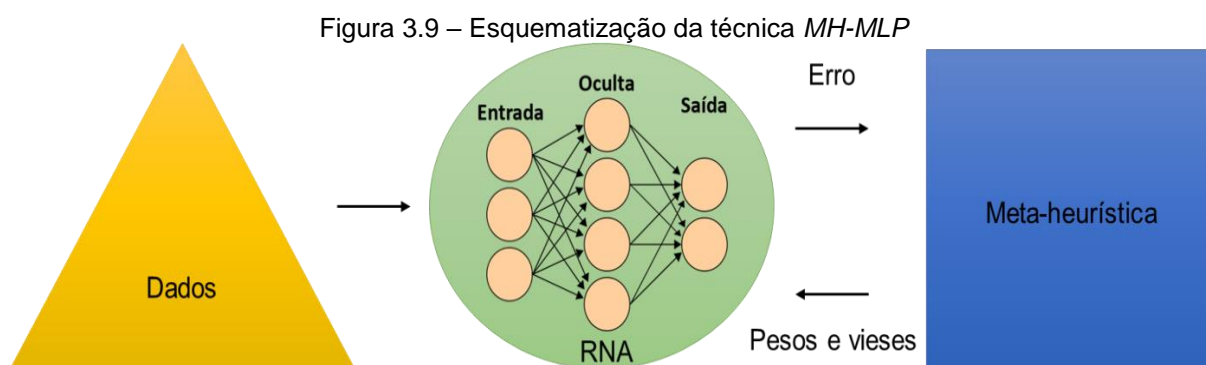
1.  $i \leftarrow tamanho(entrada)$
2.  $j \leftarrow tamanho(alvo)$
3.  $D \leftarrow (i + 1) * H + (H + 1) * j$
4.  $wb \leftarrow aleatório(1, D)$
5.  $f \leftarrow MSE(wb, entrada, alvo)$
6. **enquanto**  $\neg$  Critério de Parada
7.      $[saída, wb] \leftarrow Meta - heurística(f, D, N, iter)$
8. **fim**
9.  $wb_{melhor} \leftarrow w$
10. **Retorna**  $wb_{melhor}$ , *saída*

---

Fonte: o autor, 2018

Como saída tem-se os valores previstos pela rede (*saída*), que serão comparados com os valores desejados (*alvo*) para mensuração da acurácia percentual, e os pesos e vieses otimizados ( $wb_{melhor}$ ), que também serão utilizados para inicialização do algoritmo *LM*, abordagem denominada *MH-LM*.

A Figura 3.9 esquematiza o treinamento proposto da RNA por meio de MHs.



Fonte: o autor, 2018

## 4 RESULTADOS

Neste capítulo são apresentados os resultados dos estudos de caso com a aplicação do algoritmo *HFCM* ao problema de agrupamento, atestando sua eficácia e justificando seu uso posterior na etapa de pré-processamento do problema principal da tese, de classificação, e dos algoritmos *MH-MLP* e *MH-LM* à classificação de instalações de energia solar e a outros seis conjuntos de dados visando validar as técnicas propostas.

O computador utilizado foi um *Intel i7-2600* (3,40GHz), com 16GB de memória *ram*, *SSD* 128GB e *Windows* 10. A versão *MATLAB* foi a R2018b (64 bits).

### 4.1 ALGORITMO *HFCM*

Nesta primeira etapa, o algoritmo *HFCM* foi testado com sete conjuntos de dados:

1. *Solar clustering* (Franco e Steiner, 2018);
2. *Aggregation* (Gionis, Mannila e Tsaparas, 2007);
3. *Compound* (Zahn, 1971);
4. *D31* (Veenman, Reinders e Backer, 2002);
5. *T4.8k* (Karypis, Eui-Hong Han e Kumar, 1999);
6. *Credit card* (Yeh e Lien, 2009);
7. *Wine quality* (Cortez *et al.*, 2009).

O primeiro conjunto (*Solar clustering*) é um problema de clusterização de áreas abandonadas também para instalação de usinas fotovoltaicas, e os demais seis conjuntos são instâncias da literatura utilizadas para validar a metodologia.

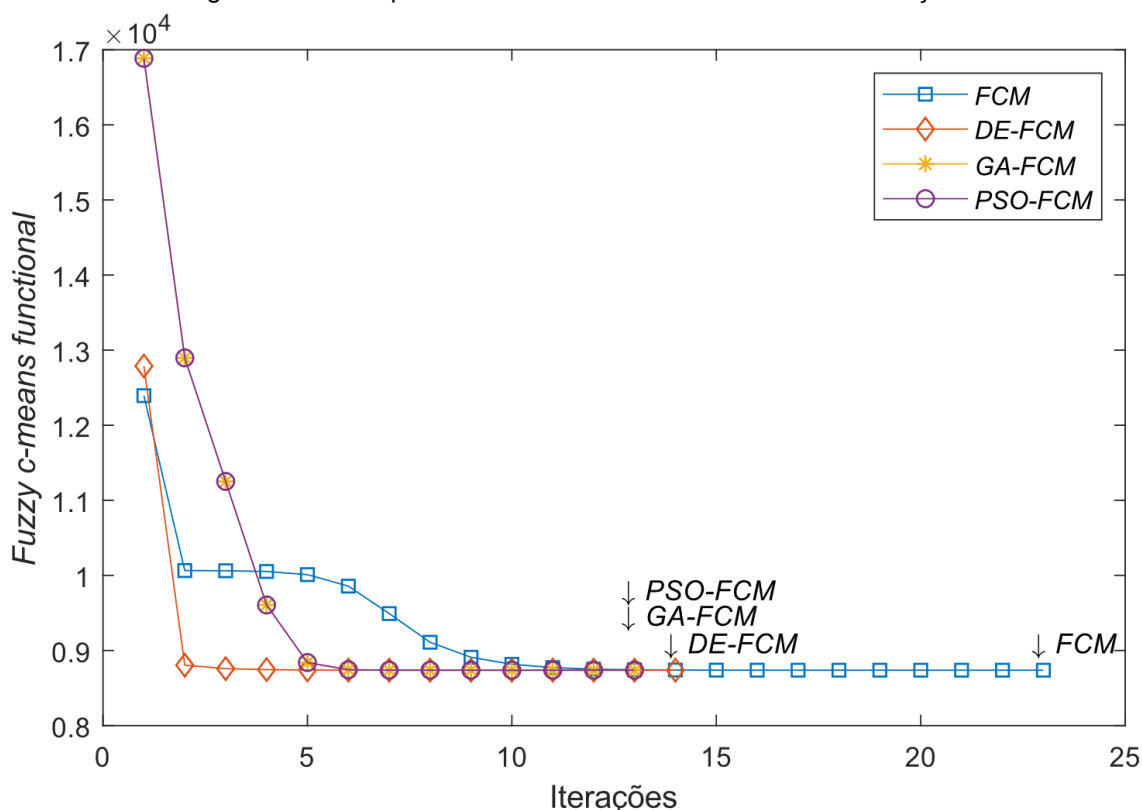
#### 4.1.1 Clusterização aplicada ao problema de Energia Solar

Para o conjunto de dados *Solar clustering*, composto por 5.063 instâncias e quatro variáveis (“tamanho da área”, em  $m^2$ ; “distâncias até as linhas de transmissão”, em  $m$ ; “radiação solar direta em escala de utilidade” e “radiação solar direta *off-grid*”, ambas medidas em  $kWh/m^2/dia$ ), o algoritmo *HFCM* reduziu o número de iterações e o tempo total de execução. A versão clássica do algoritmo *FCM* demandou 23

iterações, enquanto que a inicializada por *GA* e *PSO* demandou 13 iterações e a versão inicializada pela *DE* precisou de 14 iterações. Todas alcançaram o mesmo valor final para a função objetivo ( $8.737,761$ ), conforme mostrado na Figura 4.1.

Embora a inicialização pela MH *DE* tenha consumido mais iterações, o tempo total de execução foi menor, devido à maior simplicidade do algoritmo (Tabela 4.1). As inicializações por *GA* e *PSO* também tiveram tempos de execução diferentes.

Figura 4.1 – Comparativo: Meta-heurísticas x Número de iterações



Fonte: o autor, 2018

Percebe-se que, quando o algoritmo é inicializado pelas MHs, a localização dos centroides é otimizada antes da execução do algoritmo *FCM*, o que garante a rápida convergência do mesmo. A Tabela 4.1 compara a média e o desvio-padrão para o tempo de execução do algoritmo, em segundos, e para o valor da função objetivo, em 100 repetições realizadas. A diferença percentual se refere à comparação com o valor do algoritmo *FCM* clássico.

Tabela 4.1 – Resultados para o conjunto *Solar clustering*

Parâmetros: $N = 5.063$ ; $D = 4$ ; $C = 2$						
	Tempo (s)			Custo da solução		
	Média	Diferença %	Desvio-Padrão	Média	Diferença %	Desvio-Padrão
<i>FCM</i>	0,01532	-	0,0014	8737,76155	-	0,0001
<i>DE-FCM</i>	0,01175	-23,30	0,0013	8737,76152	0,00	0,0001
<i>GA-FCM</i>	0,01374	-10,31	0,0068	8737,76152	0,00	0,0001
<i>PSO-FCM</i>	0,01284	-16,19	0,0042	8737,76153	0,00	0,0001

Fonte: o autor, 2018

#### 4.1.2 Instâncias da literatura

Para validar o potencial da técnica *HFCM*, foram realizadas 100 repetições (100 inicializações diferentes) com outros seis conjuntos de dados, com diferentes números de instâncias  $N$  (amostras), dimensões  $D$  (número de variáveis) e número de agrupamentos  $C$  (predefinidos em suas respectivas referências), cujos resultados, em termos de valores mínimos (Min.), médios, máximos (Max.) e desvio-padrão (DP.), com o tempo medido em segundos (tempo total de *CPU* – *Central Processing Unit* – utilizado pela aplicação), são apresentados nas Tabelas 4.2 a 4.7, com destaque para os menores valores (mínimos, médios e máximos).

Tabela 4.2 – Resultados com o conjunto *Aggregation*

Parâmetros: $N = 788$ ; $D = 2$ ; $C = 7$								
	Tempo (s)				Custo da solução			
	Min.	Média	Max.	DP.	Min.	Média	Max.	DP.
<i>FCM</i>	0,0138	0,0184	0,0267	0,002	6889,2553	6889,2556	6889,2558	0,000
<i>DE-FCM</i>	0,0108	<b>0,0164</b>	<b>0,0225</b>	0,002	<b>6889,2548</b>	<b>6889,2555</b>	6889,2558	0,000
<i>GA-FCM</i>	0,0108	0,0169	0,0276	0,003	6889,2553	6889,2556	6889,2558	0,000
<i>PSO-FCM</i>	<b>0,0096</b>	<b>0,0164</b>	0,0263	0,002	6889,2553	6889,2556	<b>6889,2557</b>	0,000

Fonte: o autor, 2018

Tabela 4.3 – Resultados com o conjunto *Compound*

Parâmetros: $N = 399$ ; $D = 2$ ; $C = 6$								
	Tempo (s)				Custo da solução			
	Min.	Média	Max.	DP.	Min.	Média	Max.	DP.
<i>FCM</i>	0,0062	0,0120	<b>0,0226</b>	0,003	2272,2842	<b>2272,2842</b>	<b>2272,2843</b>	0,000
<i>DE-FCM</i>	0,0053	0,0118	0,0237	0,004	2272,2842	2272,2843	<b>2272,2843</b>	0,000
<i>GA-FCM</i>	<b>0,0050</b>	<b>0,0114</b>	0,0345	0,004	2272,2842	2272,2843	2272,2844	0,000
<i>PSO-FCM</i>	0,0052	0,0117	0,0251	0,004	<b>2272,2841</b>	2272,2843	2272,2845	0,000

Fonte: o autor, 2018

Tabela 4.4 – Resultados com o conjunto *D31*

Parâmetros: $N = 3100$ ; $D = 2$ ; $C = 31$								
	Tempo (s)				Custo da solução			
	Min.	Média	Max.	DP.	Min.	Média	Max.	DP.
<i>FCM</i>	0,1416	0,2917	<b>0,4336</b>	0,082	2018,2199	2102,5906	<b>2143,6115</b>	45,056
<i>DE-FCM</i>	<b>0,0724</b>	0,2430	0,4468	0,103	1927,2967	2157,4849	2251,2808	69,025
<i>GA-FCM</i>	0,0951	<b>0,2389</b>	0,4383	0,090	<b>1927,2964</b>	2102,8785	2177,2867	59,583
<i>PSO-FCM</i>	0,1086	0,2526	0,4370	0,098	1927,2965	<b>2093,8631</b>	2149,7790	48,705

Fonte: o autor, 2018

Tabela 4.5 – Resultados com o conjunto *T4.8k*

Parâmetros: $N = 8000$ ; $D = 2$ ; $C = 6$								
	Tempo (s)				Custo da solução			
	Min.	Média	Max.	DP.	Min.	Média	Max.	DP.
<i>FCM</i>	0,1149	0,1564	0,2210	0,021	1,7621E+07	1,762E+07	1,7621E+07	0,000
<i>DE-FCM</i>	0,0856	<b>0,1316</b>	0,2291	0,023	1,7621E+07	1,762E+07	1,7621E+07	0,000
<i>GA-FCM</i>	<b>0,0850</b>	0,1386	0,2306	0,029	1,7621E+07	1,762E+07	1,7621E+07	0,000
<i>PSO-FCM</i>	0,0905	0,1378	<b>0,2152</b>	0,026	1,7621E+07	1,762E+07	1,7621E+07	0,000

Fonte: o autor, 2018

Tabela 4.6 – Resultados com o conjunto *Credit Card*

Parâmetros: $N = 30000$ ; $D = 23$ ; $C = 2$								
	Tempo (s)				Custo da solução			
	Min.	Média	Max.	DP.	Min.	Média	Max.	DP.
<i>FCM</i>	0,4846	0,6766	0,7738	0,047	5,9365E+14	5,936E+14	5,9365E+14	1,321
<i>DE-FCM</i>	<b>0,0802</b>	<b>0,4947</b>	0,7694	0,251	5,9365E+14	5,936E+14	5,9365E+14	1,332
<i>GA-FCM</i>	0,0804	0,5231	<b>0,7257</b>	0,197	5,9365E+14	5,936E+14	5,9365E+14	1,314
<i>PSO-FCM</i>	0,0821	0,5999	0,7726	0,146	5,9365E+14	5,936E+14	5,9365E+14	1,338

Fonte: o autor, 2018

Tabela 4.7 – Resultados com o conjunto *Wine Quality*

Parâmetros: $N = 4898$ ; $D = 11$ ; $C = 7$								
	Tempo (s)				Custo da solução			
	Min.	Média	Max.	DP.	Min.	Média	Max.	DP.
<i>FCM</i>	0,1136	0,1693	0,1952	0,016	6,5131E+05	6,513E+05	6,5131E+05	0,003
<i>DE-FCM</i>	0,0602	0,1524	0,1905	0,028	6,5131E+05	6,513E+05	6,5131E+05	1,452
<i>GA-FCM</i>	0,0740	<b>0,1410</b>	<b>0,1903</b>	0,032	6,5131E+05	6,513E+05	6,5131E+05	0,812
<i>PSO-FCM</i>	<b>0,0501</b>	0,1474	0,1925	0,033	6,5131E+05	6,513E+05	6,5131E+05	0,991

Fonte: o autor, 2018

Para todos os conjuntos de dados testados, o algoritmo *HFCM* obteve uma redução no tempo de processamento. Este é um ponto forte para aplicações que demandem rapidez no processamento, como sistemas de processamento *on-line*. A Tabela 4.8 apresenta a redução de tempo alcançada pelo algoritmo *HFCM* e sua proporcionalidade ao número de instâncias  $N$  em cada base de dados ( $D$  refere-se à dimensão do problema e  $C$  ao número de *clusters*).

Tabela 4.8 – Redução de tempo alcançada por *HFCM*

Conjunto	N	D	C	Redução	Meta-heurística
<i>Credit card</i>	30.000	23	2	26,88%	<i>DE</i>
<i>Solar</i>	5.063	4	2	23,30%	<i>DE</i>
<i>D31</i>	3.100	2	31	18,10%	<i>GA</i>
<i>Wine quality</i>	4.898	11	7	16,72%	<i>GA</i>
<i>T4.8k</i>	8.000	22	6	15,86%	<i>DE</i>
<i>Aggregation</i>	788	2	7	10,87%	<i>DE e PSO</i>
<i>Compound</i>	399	2	6	5,00%	<i>GA</i>

Fonte: o autor, 2018



## 4.2 RNA OTIMIZADA

As MHs, quando aplicadas isoladamente, para o treinamento completo da RNA, serão chamadas genericamente de *MH-MLP* (ou *DE-MLP*, *GA-MLP* e *PSO-MLP*, quando referidas especificamente). Quando estas MHs são utilizadas apenas na fase de inicialização dos pesos neurais, com o algoritmo *LM* aplicado na sequência, serão chamadas, em conjunto, de *MH-LM* (ou *DE-LM*, *GA-LM* e *PSO-LM*, quando referidas individualmente).

Nesta segunda etapa, os algoritmos *MH-MLP* e *MH-LM* foram testados também com sete conjuntos de dados:

1. *Solar classification*, descrito na seção 3.1.2;
2. *Breast cancer*: 9 atributos e 699 instâncias, classificadas em benigno ou maligno;
3. *Crab gender*: 6 atributos e 200 instâncias, classificadas em macho ou fêmea;
4. *Ovarian cancer*: 100 atributos e 216 instâncias, classificadas em paciente com câncer ou paciente sem câncer;
5. *Thyroid function*: 21 atributos e 7.200 instâncias, classificadas em função normal, hipertireoidismo e hipotireoidismo;
6. *Parkinson disease*: 22 atributos e 195 instâncias, classificadas em Parkinson's disease ou saudável;
7. *Ionosphere*: 34 atributos e 351 instâncias, classificadas em *good radar returns* e *bad radar returns*.

Os conjuntos 2 a 7 estão disponíveis no *UCI Machine Learning Repository*.

### 4.2.1 Classificação aplicada ao problema da energia solar

Utilizando a estratégia *hold-out*, considerada uma das mais confiáveis para estimar a acurácia de um modelo preditivo (Lee, Liong e Jemain, 2018), os dados foram divididos em dois conjuntos, treinamento e teste, com 50% dos dados em cada um, visando maior segurança na avaliação da qualidade da classificação e maior simplicidade computacional em relação ao *k-fold cross-validation* (Arlot e Celisse,

2010; Massart, 2007; Yadav e Shukla, 2016). Também foi utilizada a Regularização Bayesiana na RNA, o que dispensa o uso do conjunto de validação.

O número de neurônios testado foi  $N = [10, 30, 50]$  e o número de elementos na população foi  $P = [50, 100]$ . As Tabelas 4.9 a 4.15 apresentam os resultados em termos de *MSE* e tempo total de *CPU* utilizado pela aplicação, em segundos, enquanto que a Tabela 4.16 apresentam a acurácia percentual alcançada pela topologia com o menor *MSE* no conjunto de teste, para o conjunto de áreas abandonadas e outros conjuntos clássicos encontrados na literatura.

Tabela 4.9 – Resultados para o conjunto *Solar classification* com *LM*

		MSE Train		Time (training)	MSE Test
Neurons	10	0,1139		818,14	0,1143
	30	0,1283		4450,58	0,1285
	50	0,1047		11206,09	<b>0,1054</b>

Fonte: o autor, 2018

Tabela 4.10 – Resultados para o conjunto *Solar classification* com *DE-MLP*

		MSE Train		Time (training)		MSE Test	
DE	Population	Population		Population		Population	
		50	100	50	100	50	100
Neurons	10	0,1293	0,1358	2426,06	4837,85	<b>0,1289</b>	0,1349
	30	0,3990	0,4160	3121,10	6240,68	0,3964	0,4114
	50	0,7790	0,6821	3730,23	8049,35	0,7804	0,6801

Fonte: o autor, 2018

Tabela 4.11 – Resultados para o conjunto *Solar classification* com *DE-LM*

		MSE Train		Time (training)		MSE Test	
DE	Population	Population		Population		Population	
		50	100	50	100	50	100
Neurons	10	0,0948	0,0902	1048,53	1031,57	0,0959	0,0913
	30	0,1021	0,0867	5124,31	5089,43	0,1029	<b>0,0873</b>
	50	0,09152	0,1051	12994,40	10323,53	0,0923	0,1060

Fonte: o autor, 2018

Tabela 4.12 – Resultados para o conjunto *Solar classification* com *GA-MLP*

		MSE Train		Time (training)		MSE Test	
GA		Population		Population		Population	
		50	100	50	100	50	100
Neurons	10	0,1247	0,1233	1857,78	3687,45	0,1244	<b>0,1232</b>
	30	0,1347	0,1265	2372,45	4685,17	0,1345	0,1264
	50	0,1421	0,1236	3040,21	5679,78	0,1412	0,1235

Fonte: o autor, 2018

Tabela 4.13 – Resultados para o conjunto *Solar classification* com *GA-LM*

		MSE Train		Time (training)		MSE Test	
GA		Population		Population		Population	
		50	100	50	100	50	100
Neurons	10	0,0802	0,0833	884,45	1000,87	<b>0,0813</b>	0,0840
	30	0,1067	0,1037	3377,32	3657,95	0,1072	0,1046
	50	0,1103	0,1073	10918,07	9889,54	0,1106	0,1078

Fonte: o autor, 2018

Tabela 4.14 – Resultados para o conjunto *Solar classification* com *PSO-MLP*

		MSE Train		Time (training)		MSE Test	
PSO		Population		Population		Population	
		50	100	50	100	50	100
Neurons	10	0,1222	0,1219	2010,58	4000,58	0,1220	0,1217
	30	0,1215	0,1221	2475,05	5046,16	0,1212	0,1219
	50	0,1210	0,1213	3193,44	6297,69	<b>0,1208</b>	0,1212

Fonte: o autor, 2018

Tabela 4.15 – Resultados para o conjunto *Solar classification* com *PSO-LM*

		MSE Train		Time (training)		MSE Test	
PSO		Population		Population		Population	
		50	100	50	100	50	100
Neurons	10	0,0902	0,0834	837,42	744,78	0,0909	<b>0,0847</b>
	30	0,0858	0,1016	3637,27	3675,50	0,0864	0,1025
	50	0,1062	0,1004	10216,52	9613,09	0,1068	0,1014

Fonte: o autor, 2018

Em termos de *MSE*, a técnica híbrida *MH-LM* obteve uma redução média de aproximadamente 20% em relação ao algoritmo *LM*: *GA-LM* obteve uma redução de 22,9%, *PSO-LM* alcançou uma redução de 19,6% e *DE-LM* reduziu o *MSE* em 17,2%.

#### 4.2.2 Instâncias da literatura

A divisão dos dados foi a mesma apresentada na Seção 4.2.1: 50% para treinamento e 50% para teste, assim como o número de neurônios e partículas, ou seja,  $N = [10,30,50]$  e  $P = [50,100]$ , respectivamente. A Tabela 4.16 apresenta a acurácia para todos os conjuntos de dados testados, com destaque para os melhores resultados com relação ao conjunto de teste.

Tabela 4.16 – Acurácia dos modelos testados

Conjunto		<i>Solar</i>	<i>Breast Cancer</i>	<i>Crab Gender</i>	<i>Ovarian Cancer</i>	<i>Thyroid Function</i>	<i>Parkinson Disease</i>	<i>Ionosphere</i>
<i>LM</i>	Treina.	75,70%	100%	100%	100%	99,40%	100%	100%
	Teste	75,50%	95,10%	94,00%	93,50%	97,80%	89,70%	89,70%
<i>DE- MLP</i>	Treina.	76,10%	93,40%	81,00%	72,20%	91,10%	84,70%	81,80%
	Teste	76,10%	91,10%	82,00%	77,80%	91,10%	78,40%	77,10%
<i>DE- LM</i>	Treina.	79,30%	100%	100%	100%	99,60%	100%	100%
	Teste	79,30%	<b>95,40%</b>	96,00%	93,50%	<b>98,00%</b>	<b>90,70%</b>	90,30%
<i>GA- MLP</i>	Treina.	76,20%	98,90%	99,00%	98,10%	92,60%	99,00%	99,40%
	Teste	76,20%	94,60%	92,00%	93,50%	92,60%	81,40%	<b>91,40%</b>
<i>GA- LM</i>	Treina.	81,60%	100%	100%	100%	99,70%	100%	100%
	Teste	<b>81,20%</b>	<b>95,40%</b>	95,00%	<b>94,40%</b>	97,70%	89,70%	90,30%
<i>PSO- MLP</i>	Treina.	76,10%	100%	100%	100%	94,30%	100%	100%
	Teste	76,10%	94,60%	<b>97,00%</b>	89,80%	94,10%	89,70%	85,70%
<i>PSO- LM</i>	Treina.	80,90%	100%	100%	100%	99,40%	100%	100%
	Teste	81,00%	<b>95,40%</b>	96,00%	<b>94,40%</b>	<b>98,00%</b>	87,60%	90,30%
<i>LDA</i>	Treina.	70,00%	90,10%	91,10%	90,10%	91,60%	90,90%	95,80%
	Teste	69,70%	84,80%	88,50%	72,90%	90,70%	80,40%	85,10%
<i>NB</i>	Treina.	58,80%	93,40%	62,00%	91,60%	92,10%	88,70%	94,30%
	Teste	54,30%	90,70%	53,00%	90,70%	91,90%	76,20%	86,70%
<i>DT</i>	Treina.	75,70%	92,40%	96,00%	90,20%	92,60%	84,90%	90,20%
	Teste	71,20%	91,70%	87,00%	85,10%	91,40%	79,30%	83,90%
<i>SVM</i>	Treina.	76,90%	94,10%	95,00%	96,30%	92,80%	92,80%	91,30%
	Teste	74,30%	92,20%	91,00%	92,50%	92,10%	85,40%	87,80%
<i>RF</i>	Treina.	77,80%	94,00%	90,80%	97,80%	93,00%	88,90%	90,10%
	Teste	75,50%	90,80%	86,00%	92,50%	92,70%	80,40%	88,10%

Fonte: o autor, 2018

Também foram comparados os resultados utilizando as técnicas: *Linear Discriminant Analysis (LDA)*, *Naive Bayes (NB)*, *Decision Trees (DT)*, *Support Vector Machines (SVM)* e *Random Forest (RF)*, uma estratégia de aprendizado de conjunto (*ensemble learning*).

Todos os melhores resultados foram obtidos com os algoritmos *MH-LM (DE-LM* em 3 casos, *GA-LM* em 3 casos e *PSO-LM* em 3 casos, considerando empates), satisfazendo o objetivo do trabalho de propor uma nova estratégia de inicialização dos pesos e vieses para as RNAs com a máxima acurácia. A estratégia *MH-MLP* obteve sucesso contra *MH-LM* em dois casos (*PSO-MLP*, em *Crab gender*, e *GA-MLP*, em *Ionosphere*), embora tenha superado *LM* em cinco casos e ficado empatado em dois casos. Em todos os cenários as estratégias propostas, baseadas em RNAs, superaram as outras técnicas comparativas, conforme apresentado na Tabela 4.16.

Visando assegurar a veracidade dos resultados, foi aplicado o teste de McNemar ao melhor resultado obtido para o conjunto de dados Solar. Este teste compara os valores previstos (por dois algoritmos) com os valores verdadeiros e verifica se a diferença entre as taxas de erro é estatisticamente significativa.

Primeiramente foi testado se os resultados obtidos por *LM* e *GA-LM* eram estatisticamente iguais. Tanto para o conjunto de treinamento ( $p\text{-valor} = 1,2e-169$ ) quanto para o conjunto de teste ( $p\text{-valor} = 1,7e-146$ ), com um nível de significância igual a 5%, rejeitou-se a hipótese nula de que, prevendo os valores reais, *GA-LM* e *LM* possuem acurácias iguais.

Em segundo lugar testou-se se a acurácia de *LM* era maior ou menor do que a acurácia de *GA-LM*. Também com um nível de significância de 5% os resultados mostraram que, tanto para o conjunto de treinamento ( $p\text{-valor} = 6,2e-170$ ) quanto para o conjunto de teste ( $p\text{-valor} = 8,6e-147$ ), *GA-LM* é estatisticamente mais acurado do que *LM*.

### 4.2.3 Base de dados desbalanceada

Um dos problemas enfrentado com o conjunto *Solar classification* foi o desbalanceamento dos dados. Neste conjunto 76,2% das instâncias pertenciam à classe 3, *off-grid*, induzindo a RNA a se sobreajustar a este grupo (92,4% de acerto) em detrimento dos outros dois grupos. A Tabela 4.17 exemplifica o problema utilizando o melhor resultado obtido.

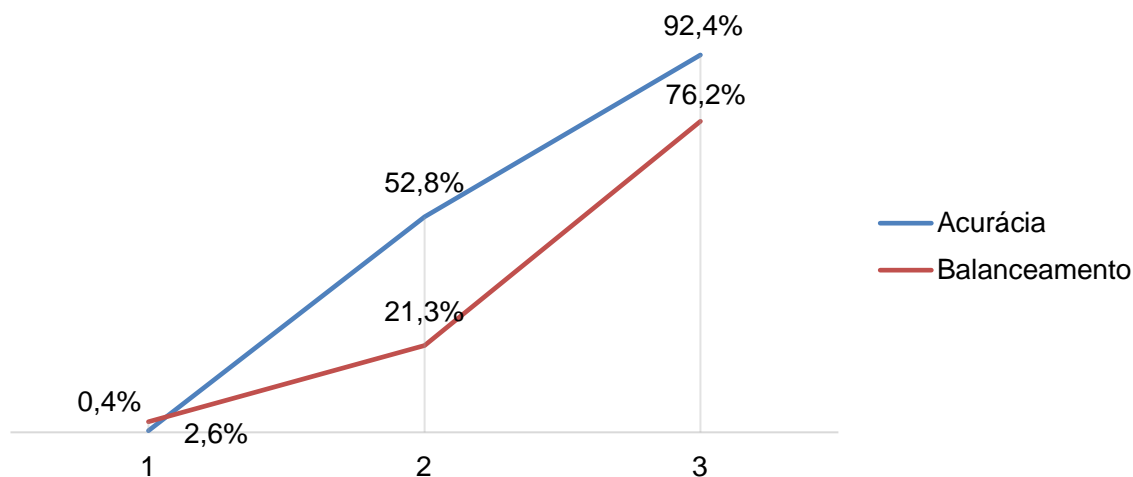
Tabela 4.17 – Exemplo de base de dados desbalanceada  
GA-LM (Treinamento) N = 10 P = 50

Previsão	1	4 0,0%	2 0,0%	2 0,0%	50,0% 50,0%
	2	146 0,4%	4572 11,2%	2354 5,8%	64,6% 35,4%
	3	906 2,2%	4091 10,0%	28690 70,4%	85,2% 14,8%
		0,4% 99,6%	52,8% 47,2%	92,4% 92,4%	81,6% 18,4%
		1	2	3	
		Alvo			

Fonte: o autor, 2018

Para as classes 2, *Large scale* (21,2% das amostras), e 1, *Utility scale* (2,6% das amostras), a acurácia ficou em 52,8% e 0,4%, respectivamente. A Figura 4.2 faz um comparativo entre a acurácia e o balanceamento das classes.

Figura 4.2 – Comparativo entre a acurácia e o balanceamento



Fonte: o autor, 2018

Analisando os resultados podemos notar o seguinte:

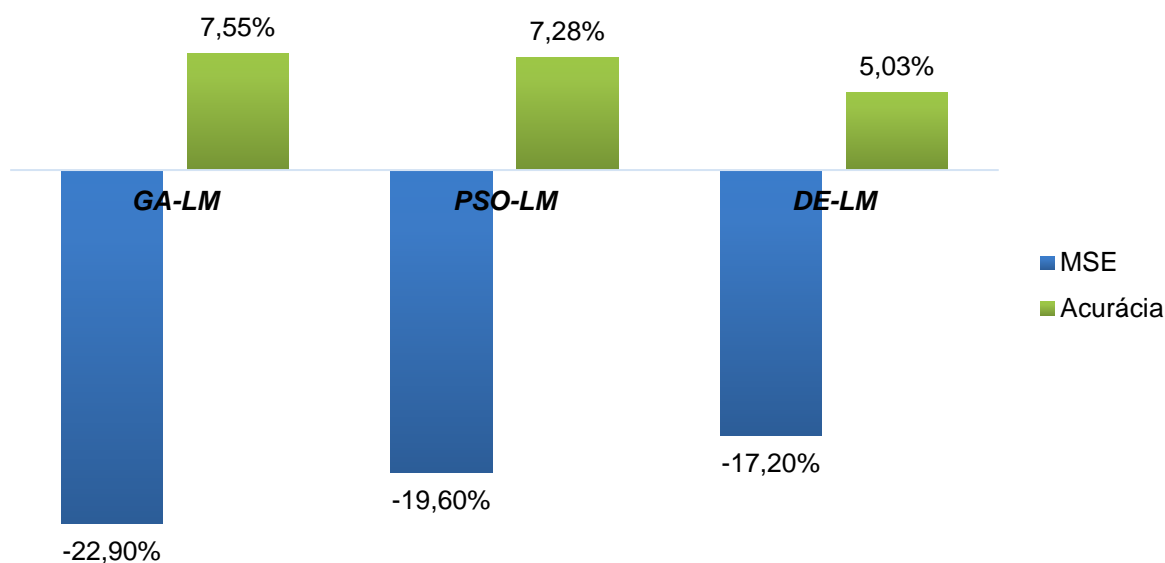
1. Para a base de dados *Solar classification* (que possui o maior número de instâncias) o maior incremento na acurácia, para o conjunto de testes, foi obtido pelo algoritmo *GA-LM* (7,5% em relação à técnica *LM*), seguido

por *PSO-LM* (incremento de 7,3% em relação a *LM*) e *DE-LM* (5,0% de incremento em relação a *LM*);

2. Em média, as estratégias *MH-MLP* mostraram desempenho inferior ao algoritmo *LM* “puro”, tanto em relação ao conjunto de testes quanto em relação ao conjunto de treinamento embora, em dois casos, tenha superado todos os outros métodos;
3. Para o conjunto de teste a estratégia *GA-LM* foi a que se saiu melhor, considerando todas as bases de dados testadas, com um incremento médio de 1,5% na acurácia, seguida por *DE-LM* (incremento médio de 1,4%) e *PSO-LM* (incremento médio de 1,3%);
4. Para o conjunto de treinamento *GA-LM* também foi a melhor colocada, considerando os sete conjuntos de dados testados, com incremento médio de 1,2%, seguida por *PSO-LM* (incremento médio de 1,0%) e *DE-LM* (incremento médio de 0,7%);
5. Deve-se salientar que, para o conjunto de treinamento, em 5 das 7 bases de dados testadas o algoritmo *LM*, assim como as *MH-LM*, obtiveram 100% de acerto; nas outras duas bases (*Solar classification* e *Thyroid Function*) o algoritmo *MH-LM* obteve os melhores resultados.

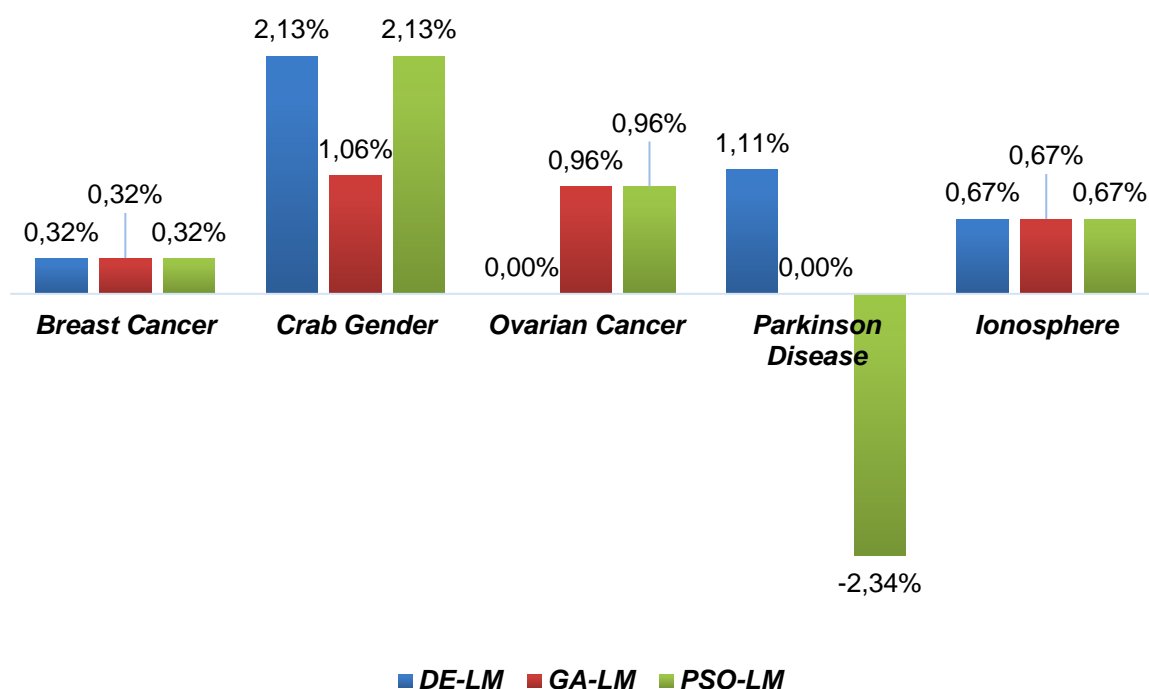
Estes resultados atestam a capacidade da abordagem híbrida (inicialização por *MH* seguida de otimização por *LM*) em obter um resultado superior, tanto para o conjunto de treinamento quanto para o conjunto de teste, este último mais importante por representar a capacidade de generalização. A Figura 4.3 apresenta graficamente a melhoria de desempenho para a base de dados solar, enquanto que a Figura 4.4 resume a capacidade de generalização (incremento na acurácia do conjunto de teste) para as cinco bases de dados onde o algoritmo *LM* já havia alcançado uma acurácia de 100% no conjunto de treinamento.

Figura 4.3 – Melhorias atingidas pela técnica proposta (MH-LM)



Fonte: o autor, 2018

Figura 4.4 – Capacidade de generalização da técnica proposta (MH-LM)



Fonte: o autor, 2018

Ao trabalhar, ineditamente, com o conjunto *Solar classification*, o objetivo foi demonstrar a capacidade das técnicas propostas em um problema real que ainda é pouco explorado em nosso país: o reaproveitamento de áreas improdutivas, como aterros sanitários e terrenos abandonados, para implantação de projetos voltados à geração de energia renovável. Uma vez que no Brasil não existem bases de dados



suficientes (com todas as variáveis de interesse aqui tratadas) referentes a este problema, que poderiam ser utilizadas para o treinamento de um classificador, a abordagem proposta permite transferir o conhecimento adquirido em outra região (neste caso específico, os Estados Unidos) para a resolução de um problema local.

Na prática isso se daria pela inserção, no modelo treinado, das variáveis medidas em cada área disponível para a implantação de um projeto de geração de energia solar. Como resultado a RNA classificaria o novo terreno em um dos três tipos possíveis de usina solar: *off-grid*, *large* ou *utility*. Esse modelo poderá ser expandido para outros tipos de energia renovável, uma vez que os dados necessários estão disponíveis em abundância (nos Estados Unidos). Basta, para isso, que uma RNA *MH-LM* (que se saiu melhor nos testes) seja treinada com os dados referentes a cada tipo de energia renovável.

## 5 CONCLUSÕES

O objetivo central deste trabalho foi desenvolver uma nova técnica de otimização para a obtenção dos pesos iniciais das RNAs visando aumentar sua acurácia e capacidade de generalização da técnica em problemas de classificação, especialmente no problema de classificação de áreas abandonadas para implantação de usinas de geração de energia solar fotovoltaica. Também foi proposta uma nova forma de inicialização dos centroides do algoritmo *FCM*, de clusterização. Este algoritmo híbrido foi utilizado, de forma inédita, no problema de preenchimento de dados faltantes.

Os resultados, validados em diferentes conjuntos de dados, mostraram que a estratégia proposta *MH-LM* é a mais eficiente no aumento da acurácia das RNAs, sendo capaz de superar o algoritmo *LM* (puro) e outras técnicas consagradas de classificação em todas as bases de dados testadas, aliando a capacidade de otimização e fuga de mínimos locais das MHs *DE*, *GA* e *PSO* à versatilidade das RNAs em problemas de classificação. Esta era a proposta central do trabalho e foi atingida com sucesso.

Dentre as MHs utilizadas, a que apresentou melhor acurácia média no conjunto de testes foi *GA-LM*, seguida por *DE-LM* e *PSO-LM*. Isso abre um “leque” de possibilidades para testes futuros com outros algoritmos evolucionários.

De forma isolada, o treinamento apenas pelas MHs não se mostrou suficiente no aumento do desempenho da rede, fazendo-se necessária posterior otimização.

Na etapa de pré-processamento dos dados, aplicado ao problema de preenchimento de dados faltantes, o algoritmo *HFCM* proposto também cumpriu o objetivo de ser mais rápido do que o algoritmo *FCM* sem comprometer o custo da solução. Esta técnica também foi testada em seis problemas de clusterização encontrados na literatura, mostrando-se superior em todos eles. Vale ressaltar que este incremento na velocidade de treinamento será especialmente útil em problemas que exigem resposta rápida, como aprendizado *on-line*, isto é, onde os dados estão sendo processados em tempo real (ou próximo ao real).

Uma vez que foram utilizados centroides iniciais aleatórios no algoritmo *HFCM*, os valores de inicialização foram diversos (confira a Figura 4.1). Uma sugestão é que futuramente sejam utilizados os mesmos centroides, comparativamente.

O processo de implementação dos algoritmos permitirá que novas MHs, como as citadas na Tabela 2.3, assim como outras técnicas de otimização global, sejam testadas futuramente na otimização dos pesos iniciais das RNAs. Também será possível testar outros métodos de otimização não-linear irrestrita, na fase de treinamento, em substituição ao algoritmo *LM*. Entre eles podem ser citados Gradiente conjugado, Fletcher-Reeves, Polak-Ribière, Gradiente conjugado escalonado de Moller, Métodos de Newton e Gauss-Newton, Davidon-Fletcher-Powell, Broyden-Fletcher-Goldfarb-Shanno, Battiti's One-step Secant, Fahlman's Quickprop, Resilient Backpropagation, etc. Além disso, o tamanho da população utilizada por cada meta-heurística poderá ser explorado de forma mais exaustiva, com outros valores além dos aqui propostos.

Sugere-se, também, que futuramente sejam aplicadas estratégias de balanceamento dos conjuntos de dados, pois as classes menos representadas sofreram impacto negativo em suas respectivas acurácias em detrimento das classes com maior número de instâncias, embora o impacto no resultado global tenha sido pequeno, percentualmente. Por exemplo, no problema *Solar classification*, na fase de treinamento, mais de 5 mil instâncias das duas classes sub-representadas foram classificadas erroneamente (veja a Figura 4.2). Isso representa cerca de 12% do total dos dados utilizados para treinamento da RNA, o que teve um impacto significativo na acurácia final daquele modelo. Além disso, as variáveis 3 (área) e 4 (incidência solar) poderão ser combinadas formando uma única variável, reduzindo a dimensionalidade do problema.

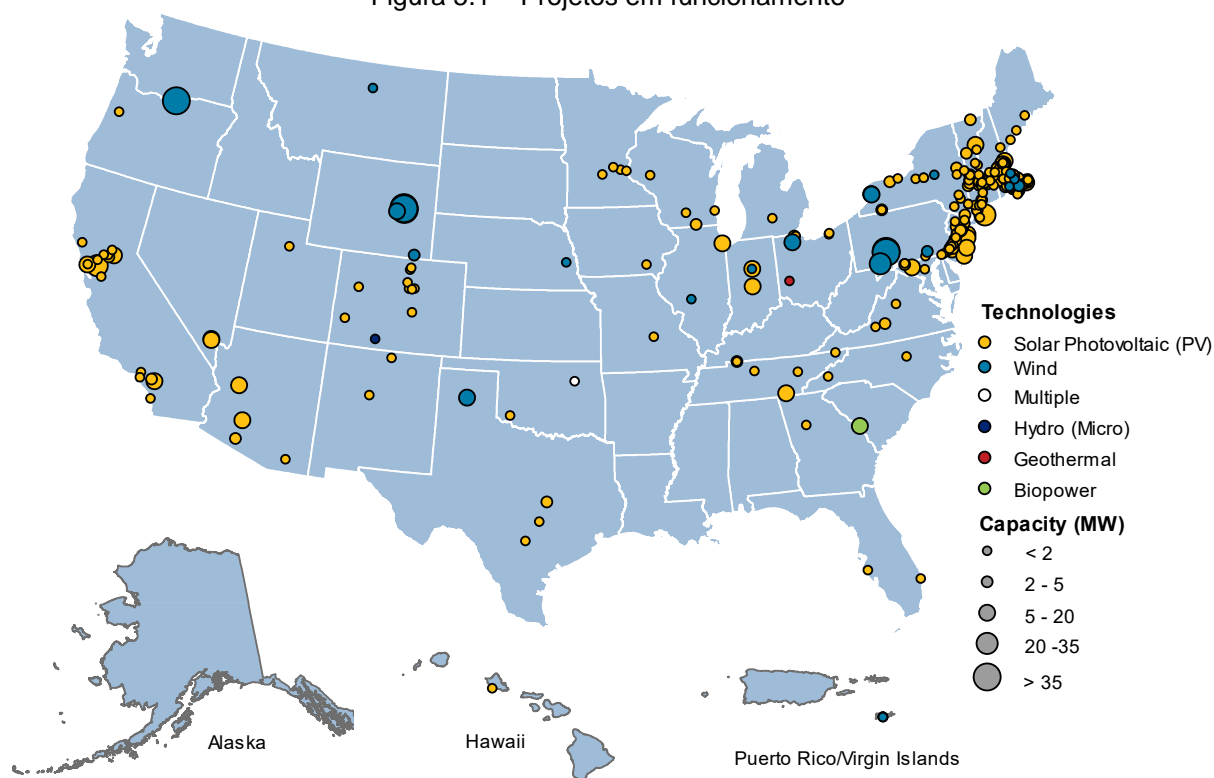
Uma vez que o uso crescente de energia elétrica em todo o mundo tem pressionado a demanda por recursos naturais, buscam-se fontes alternativas e sustentáveis que tenham menor impacto ambiental. Neste cenário, o Brasil se destaca, uma vez que já possui uma matriz energética extremamente limpa, quando comparada aos demais países, principalmente pelo uso que faz de hidroelétricas e combustíveis renováveis. Isso favorece a implantação de novos modelos de exploração energética que integrem recursos que atualmente são desperdiçados, como a energia solar, abundante em todo o nosso país, e áreas abandonadas onde se possam implantar novas usinas geradoras.

Para o problema de classificação de usinas de energia solar em terrenos abandonados pode-se extrapolar o conhecimento adquirido pela RNA com o conjunto *Solar classification* para outras regiões que não disponham de uma base de dados

completa, permitindo a identificação de terrenos em localidades que ainda não possuem ferramentas adequadas de classificação. A técnica *MH-LM* também poderá ser empregada em outros problemas de classificação, inclusive em outras áreas de energias renováveis, como eólica, geotérmica e de biomassa.

A Figura 5.1 (U.S. Environmental Protection Agency, 2018) apresenta áreas dos Estados Unidos onde já estão em funcionamento 274 projetos de geração de energia renovável das mais diversas fontes, com capacidade instalada de 1,4 *Gigawatt (GW)*. Hoje, no mundo, há uma potência instalada de cerca de 2.180 *GW*. A Figura 5.2 detalha a geração de energia renovável ao nível de continentes.

Figura 5.1 – Projetos em funcionamento



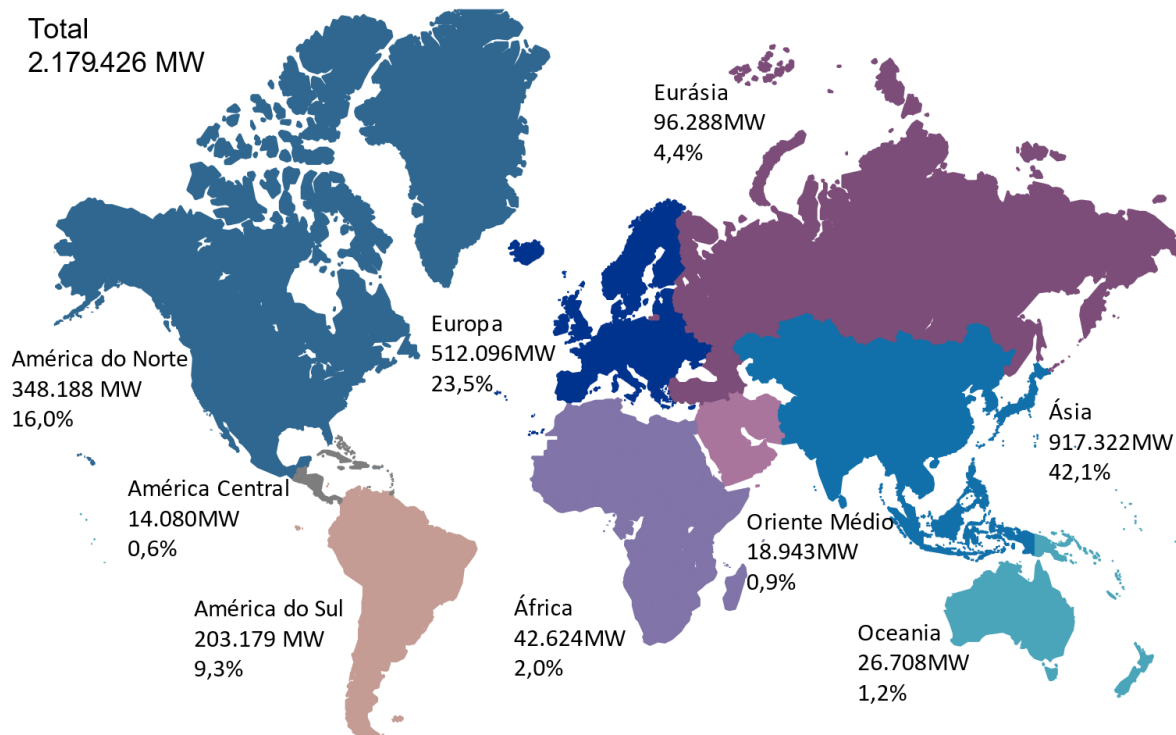
Fonte: U.S. Environmental Protection Agency, 2018

Todos os 81.533 pontos analisados neste trabalho possuem, juntos, um potencial estimado em mais de 6.775 *GW*, aproximadamente 3 vezes o que é gerado em todo o mundo. Este potencial equivale a mais de 44 milhões de toneladas de dióxido de carbono ( $CO_2$ ) que poderiam deixar de ser lançados na atmosfera apenas pelos Estados Unidos (calculado utilizando o *EPA's Avoided Emissions and Generation Tool, AVERT*).

Além do potencial de geração de energia, também deve-se levar em consideração os ganhos econômicos e ambientais decorrentes da restauração e

reutilização de áreas abandonadas ou fora de uso que, além de depreciarem a região onde estão inseridas, podem gerar riscos associados à saúde da população local e do meio ambiente. Com a correta exploração dessas áreas seria possível melhorar o equilíbrio entre o crescimento das nações e a sustentação da vida no planeta.

Figura 5.2 – Geração de energia renovável no mundo

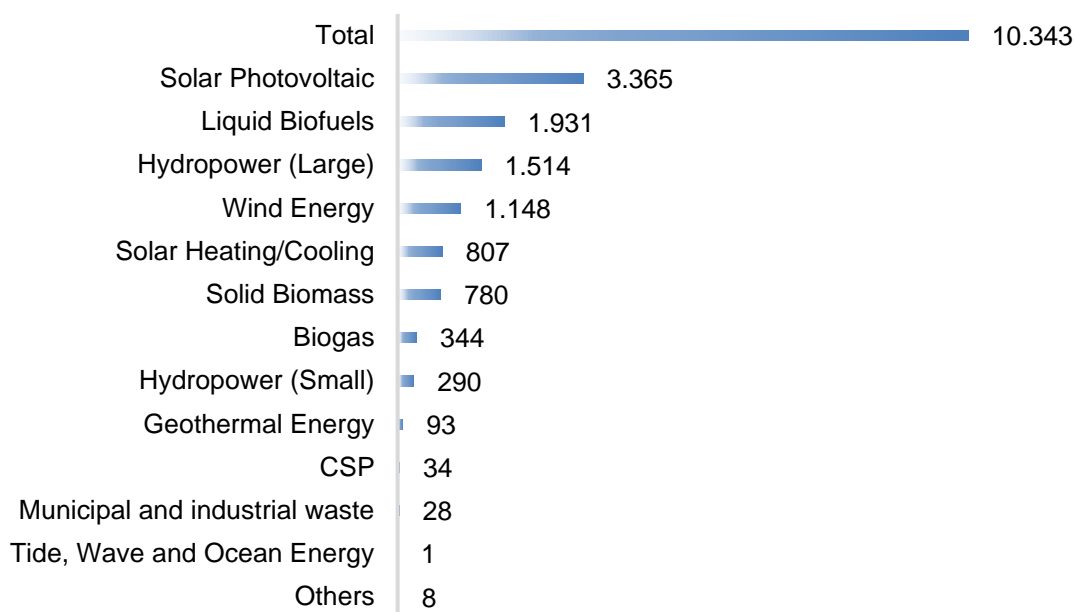


Fonte: International Renewable Energy Agency, 2018a

Desde a mais recente crise financeira o mercado de trabalho tem se deteriorado em todo o mundo. Estima-se que mais de 200 milhões de pessoas estavam desempregadas em 2013, um crescimento de mais de 30 milhões de desempregados desde 2007 (Borsi, 2018). Taxas crescentes de desemprego podem gerar efeitos negativos a longo prazo, como agitações sociais e políticas e crescimento da desigualdade econômica.

Neste sentido outra contribuição a se destacar com a expansão e implantação de novas unidades geradoras de energia renovável é a geração de empregos. Segundo a *International Renewable Energy Agency* (2018b), em seu relatório anual referente ao ano de 2017, existem mais de 10 milhões de empregados no setor das energias renováveis (veja a Figura 5.3). Com a criação de projetos renováveis em áreas atualmente inutilizadas seria possível multiplicar este número, gerando impactos positivos na economia e no desenvolvimento social.

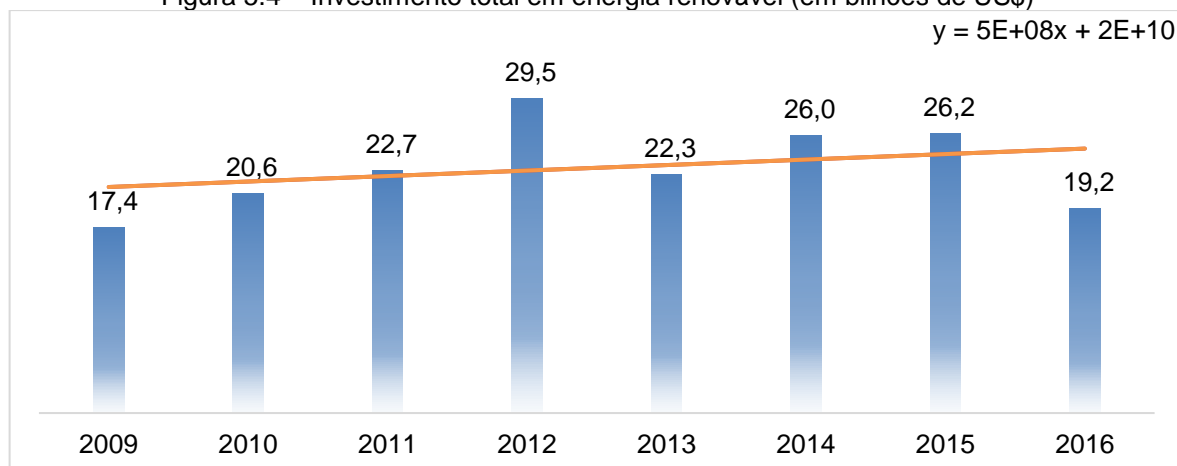
Figura 5.3 – Geração de empregos em energia renovável no mundo



Fonte: International Renewable Energy Agency, 2018b

A Figura 5.4 apresenta o nível mundial de investimentos anuais no setor de energias renováveis, com tendência de crescimento para os próximos anos (International Renewable Energy Agency, 2018a). Com técnicas como a aqui proposta, de classificação *a priori* de áreas potenciais para criação de infraestrutura energética renovável, pode-se dar maior segurança aos investidores pela garantia de melhores retornos sobre o capital investido, estimulando ainda mais o crescimento do setor.

Figura 5.4 – Investimento total em energia renovável (em bilhões de US\$)



Fonte: International Renewable Energy Agency, 2018a

## REFERÊNCIAS

- ADAM, S. P.; KARRAS, D. A.; MAGOULAS, G. D.; VRAHATIS, M. N. Solving the linear interval tolerance problem for weight initialization of neural networks. **Neural Networks**, v. 54, p. 17–37, jun. 2014.
- AGGARWAL, C. C. **Data Mining**. Cham: Springer, 2015.
- AGRAWAL, A. Unified classification of open fractures: Based on Gustilo and OTA classification schemes. **Injury**, v. 49, n. 8, p. 1526–1531, ago. 2018.
- ALI, Z.; SHAHZAD, S. K.; SHAHZAD, W. Performance Analysis of Statistical Pattern Recognition Methods in KEEL. **Procedia Computer Science**, v. 112, p. 2022–2030, 2017.
- ALIMISSIS, A.; PHILIPPOPOULOS, K.; TZANIS, C. G.; DELIGIORGI, D. Spatial estimation of urban air pollution with the use of artificial neural network models. **Atmospheric Environment**, v. 191, n. July, p. 205–213, out. 2018.
- ALMEIDA, C. M. V. B.; AGOSTINHO, F.; HUISINGH, D.; GIANNETTI, B. F. Cleaner Production towards a sustainable transition. **Journal of Cleaner Production**, v. 142, p. 1–7, 2017.
- AMIGÓ, E.; GONZALO, J.; ARTILES, J.; VERDEJO, F. A comparison of extrinsic clustering evaluation metrics based on formal constraints. **Information Retrieval**, v. 12, n. 4, p. 461–486, 2009.
- ANDRADE, D.; TAMURA, A.; TSUCHIDA, M. Exploiting covariate embeddings for classification using Gaussian processes. **Pattern Recognition Letters**, v. 104, p. 8–14, 2018.
- ANDRES, M. P.; BORRELLI, G. M.; ABRÃO, M. S. Endometriosis classification according to pain symptoms: can the ASRM classification be improved? **Best Practice & Research Clinical Obstetrics & Gynaecology**, p. 1–8, jun. 2018.
- ANKIT; SALEENA, N. An Ensemble Classification System for Twitter Sentiment Analysis. **Procedia Computer Science**, v. 132, p. 937–946, 2018.
- APOSTOLIDIS, N.; HUTTON, N. Integrated water management in brownfield sites: More opportunities than you think. **Desalination**, v. 188, n. 1–3, p. 169–175, 2006.
- ARBELAITZ, O.; GURRUTXAGA, I.; MUGUERZA, J.; PÉREZ, J. M.; PERONA, I. An extensive comparative study of *cluster* validity indices. **Pattern Recognition**, v. 46, n. 1, p. 243–256, jan. 2013.
- ARLOT, S.; CELISSE, A. A survey of cross-validation procedures for model selection. **Statistics Surveys**, v. 4, p. 40–79, 2010.
- BABU, G. P.; MURTY, M. N. Clustering with evolution strategies. **Pattern Recognition**, v. 27, n. 2, p. 321–329, 1994.
- BABUŠKA, R. **Fuzzy Modeling for Control**. New York: Springer, 1998.

BAILADOR, G.; TRIVIÑO, G. Pattern recognition using temporal fuzzy automata. **Fuzzy Sets and Systems**, v. 161, n. 1, p. 37–55, jan. 2010.

BAÑOS, R.; MANZANO-AGUGLIARO, F.; MONTOYA, F. G.; GIL, C.; ALCAYDE, A.; GÓMEZ, J. Optimization methods applied to renewable and sustainable energy: A review. **Renewable and Sustainable Energy Reviews**, v. 15, n. 4, p. 1753–1766, 2011.

BARGHASH, M. A.; SANTARISI, N. S. Pattern recognition of control charts using artificial neural networks — analyzing the effect of the training parameters. **Journal of Intelligent Manufacturing**, v. 15, n. 5, p. 635–644, out. 2004.

BEN-HUR, A.; HORN, D.; SIEGELMANN, H. T.; VAPNIK, V. Support Vector Clustering. **Journal of Machine Learning Research**, v. 2, p. 125–137, 2001.

BERGIUS, K.; ÖBERG, T. Initial screening of contaminated land: A comparison of US and Swedish methods. **Environmental Management**, v. 39, n. 2, p. 226–234, 2007.

BEUCHER, A.; MØLLER, A. B.; GREVE, M. H. Artificial neural networks and decision tree classification for predicting soil drainage classes in Denmark. **Geoderma**, v. In Press, nov. 2017.

BEYER, H.-G.; SCHWEFEL, H.-P. Evolution strategies - A comprehensive introduction. **Natural Computing**, v. 1, n. 1, p. 3–52, 2002.

BEZDEK, J. C. **Pattern Recognition with Fuzzy Objective Function Algorithms**. New York: Springer, 1981.

BEZDEK, J. C.; EHRLICH, R.; FULL, W. FCM: The fuzzy c-means clustering algorithm. **Computers & Geosciences**, v. 10, n. 2–3, p. 191–203, 1984.

BEZDEK, J. C.; HATHAWAY, R. J.; HOWARD, R. E.; WILSON, C. A.; WINDHAM, M. P. Local convergence analysis of a grouped variable version of coordinate descent. **Journal of Optimization Theory and Applications**, v. 54, n. 3, p. 471–477, 1987.

BIANCHI, L.; DORIGO, M.; GAMBARDELLA, L. M.; GUTJAHR, W. J. A survey on metaheuristics for stochastic combinatorial optimization. **Natural Computing**, v. 8, n. 2, p. 239–287, 19 jun. 2009.

BORNMANN, L.; MUTZ, R. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. **Journal of the Association for Information Science and Technology**, v. 66, n. 11, p. 2215–2222, 2015.

BORSI, M. T. Credit contractions and unemployment. **International Review of Economics & Finance**, v. 58, n. April, p. 573–593, nov. 2018.

BRAGA, A. P.; CARVALHO, A. P. L. F.; LUDERMIR, T. B. **Redes Neurais Artificiais: Teoria e Aplicações**. 2. ed. São Paulo: LTC, 2007.

BRAMER, M. **Principles of Data Mining**. 3. ed. London: Springer, 2016.

BROWNLEE, J. **Clever Algorithms**. [s.l.] LuLu, 2011.

BÜHLMANN, P.; GEER, S. VAN DE. **Statistics for High-Dimensional Data**. Berlin, Heidelberg: Springer, 2011.



- BUNKE, H.; RIESEN, K. Towards the unification of structural and statistical pattern recognition. **Pattern Recognition Letters**, v. 33, n. 7, p. 811–825, maio 2012.
- CADEZ, S.; CZERNY, A. Climate change mitigation strategies in carbon-intensive firms. **Journal of Cleaner Production**, v. 112, p. 4132–4143, 2016.
- CAI, Y.; LIAO, J.; WANG, T.; CHEN, Y.; TIAN, H. Social learning differential evolution. **Information Sciences**, v. 433–434, p. 464–509, abr. 2018.
- CAO, K.; GUAN, H. Brownfield redevelopment toward sustainable urban land use in China. **Chinese Geographical Science**, v. 17, n. 2, p. 127–134, 2007.
- CAO, W.; WANG, X.; MING, Z.; GAO, J. A review on neural networks with random weights. **Neurocomputing**, v. 275, p. 278–287, jan. 2018.
- CAPPE, M.; DERUELLE, P.; DEPRET, S.; HOUFFLIN-DEBARGE, V.; GHESQUIÈRE, L.; GARABEDIAN, C. Fetal heart rate classification in routine use: Do you prefer a 3-tier or a 5-tier classification? **Journal of Gynecology Obstetrics and Human Reproduction**, ago. 2018.
- CHAKRABORTY, U. K. (ED.). **Advances in Differential Evolution**. Berlin, Heidelberg: Springer, 2008. v. 143
- CHAO, W.; JUNZHENG, W. Cloud-service decision tree classification for education platform. **Cognitive Systems Research**, v. 52, p. 234–239, dez. 2018.
- CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. **Semi-Supervised Learning**. London: MIT Press, 2006.
- CHEN, B.; ZANG, C. A hybrid immune model for unsupervised structural damage pattern recognition. **Expert Systems with Applications**, v. 38, n. 3, p. 1650–1658, mar. 2011.
- CHEN, Y.; LI, L.; PENG, H.; XIAO, J.; WU, Q. Dynamic multi-swarm differential learning particle swarm optimizer. **Swarm and Evolutionary Computation**, v. 39, p. 209–221, abr. 2018.
- CORTEZ, P.; CERDEIRA, A.; ALMEIDA, F.; MATOS, T.; REIS, J. Modeling wine preferences by data mining from physicochemical properties. **Decision Support Systems**, v. 47, n. 4, p. 547–553, nov. 2009.
- DANIEL, L.; CHATURVEDI, K. T.; KOLHE, M. L. Dynamic Economic Load Dispatch using Levenberg Marquardt Algorithm. **Energy Procedia**, v. 144, p. 95–103, jul. 2018.
- DAS, S.; SUGANTHAN, P. N. Differential Evolution: A Survey of the State-of-the-Art. **IEEE Transactions on Evolutionary Computation**, v. 15, n. 1, p. 4–31, fev. 2011.
- DASGUPTA, D. (ED.). **Artificial Immune Systems and Their Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.
- DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. **Journal of the Royal Statistical Society. Series B - Methodological**, v. 39, n. 1, p. 1–38, 1977.
- DESARBO, W. S. Gennclus: New models for general nonhierarchical clustering

analysis. **Psychometrika**, v. 47, n. 4, p. 449–475, 1982.

DIAZ-RAINEY, I.; ASHTON, J. K. Investment inefficiency and the adoption of eco-innovations: The case of household energy efficiency technologies. **Energy Policy**, v. 82, n. 1, p. 105–117, jul. 2015.

DOOSTFATEMEH, M.; KREMER, S. C. New directions in fuzzy automata. **International Journal of Approximate Reasoning**, v. 38, n. 2, p. 175–214, fev. 2005.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)**, v. 26, n. 1, p. 29–41, 1996.

DUNN, J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated *Clusters*. **Journal of Cybernetics**, v. 3, n. 3, p. 32–57, 1974.

EBERHART, R. C.; SHI, Y. **Computational Intelligence: Concepts to Implementations**. Burlington: Morgan Kaufmann, 2007.

ENGELBRECHT, A. P. **Computational intelligence: An introduction**. 2. ed. [s.l.] Wiley, 2007.

ERDOGMUS, D.; FONTENLA-ROMERO, O.; PRINCIPE, J. C.; ALONSO-BETANZOS, A.; CASTILLO, E. Linear-Least-Squares Initialization of Multilayer Perceptrons Through Backpropagation of the Desired Response. **IEEE Transactions on Neural Networks**, v. 16, n. 2, p. 325–337, mar. 2005.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. **A Density-Based Algorithm for Discovering *Clusters* in Large Spatial Databases with Noise**. Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining - KDD '96. Portland: AAAI Press, 1996.

EUSUFF, M. M.; LANSEY, K. E. Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm. **Journal of Water Resources Planning and Management**, v. 129, n. 3, p. 210–225, 2003.

FADAEI, N.; YAN, W.-M.; MAHDI TAFARROJ, M.; KASAEIAN, A. The application of artificial neural networks to predict the performance of solar chimney filled with phase change materials. **Energy Conversion and Management**, v. 171, n. June, p. 1255–1262, set. 2018.

FAHAD, A.; ALSHATRI, N.; TARI, Z.; ALAMRI, A.; KHALIL, I.; ZOMAYA, A. Y.; FOUFOU, S.; BOURAS, A. A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. **IEEE Transactions on Emerging Topics in Computing**, v. 2, n. 3, p. 267–279, set. 2014.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, p. 37–54, 1996.

FENG, X.; XIAO, Z.; ZHONG, B.; QIU, J.; DONG, Y. Dynamic ensemble classification for credit scoring using soft probability. **Applied Soft Computing**, v. 65, p. 139–151, abr. 2018.

FERNÁNDEZ-GARCÍA, A.; ROJAS, E.; PÉREZ, M.; SILVA, R.; HERNÁNDEZ-

ESCOBEDO, Q.; MANZANO-AGUGLIARO, F. A parabolic-trough collector for cleaner industrial process heat. **Journal of Cleaner Production**, v. 89, p. 272–285, 2015.

FONTANELLA, R.; ACCARDO, D.; MORIELLO, R. S. LO; ANGRISANI, L.; SIMONE, D. DE. MEMS gyros temperature calibration through artificial neural networks. **Sensors and Actuators A: Physical**, v. 279, p. 553–565, ago. 2018.

FORESEE, F. D.; HAGAN, M. T. **Gauss-Newton approximation to Bayesian learning**. Proceedings of International Conference on Neural Networks (ICNN'97). IEEE, 1997.

FRANCO, D. G. DE B.; STEINER, M. T. A. Clustering of solar energy facilities using a hybrid fuzzy c-means algorithm initialized by metaheuristics. **Journal of Cleaner Production**, v. 191, p. 445–457, ago. 2018.

GALLEGO, A.-J.; CALVO-ZARAGOZA, J.; VALERO-MAS, J. J.; RICO-JUAN, J. R. Clustering-based k -nearest neighbor classification for large-scale data with neural codes representation. **Pattern Recognition**, v. 74, p. 531–543, fev. 2018.

GAMARRA, C.; GUERRERO, J. M.; MONTERO, E. A knowledge discovery in databases approach for industrial microgrid planning. **Renewable and Sustainable Energy Reviews**, v. 60, p. 615–630, jul. 2016.

GANDOMI, A. H.; ALAVI, A. H. Krill herd: A new bio-inspired optimization algorithm. **Communications in Nonlinear Science and Numerical Simulation**, v. 17, n. 12, p. 4831–4845, 2012.

GASS, S. I. Model World: On the Evolution of Operations Research. **Interfaces**, v. 41, n. 4, p. 389–393, ago. 2011.

GEEM, Z. W.; KIM, J. H.; LOGANATHAN, G. V. A New Heuristic Optimization Algorithm: Harmony Search. **SIMULATION**, v. 76, n. 2, p. 60–68, 2001.

GIBERT, K.; IZQUIERDO, J.; SÀNCHEZ-MARRÈ, M.; HAMILTON, S. H.; RODRÍGUEZ-RODA, I.; HOLMES, G. Which method to use? An assessment of data mining methods in Environmental Data Science. **Environmental Modelling & Software**, v. 110, n. September, p. 3–27, dez. 2018.

GIONIS, A.; MANNILA, H.; TSAPARAS, P. Clustering aggregation. **ACM Transactions on Knowledge Discovery from Data**, v. 1, n. 1, p. 1–30, 2007.

GLOROT, X.; BENGIO, Y. **Understanding the difficulty of training deep feedforward neural networks**. (Y. W. Teh & M. Titterton, Eds.) 13th International Conference on Artificial Intelligence and Statistics (AISTATS). Sardinia: PMLR, 2010. Disponível em: <[http://machinelearning.wustl.edu/mlpapers/paper\\_files/AISTATS2010\\_GlorotB10.pdf](http://machinelearning.wustl.edu/mlpapers/paper_files/AISTATS2010_GlorotB10.pdf)>

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & Operations Research**, v. 13, n. 5, p. 533–549, jan. 1986.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization & Machine Learning**. Boston: Addison-Wesley, 1989.

GOLDEWIJK, K. K.; BEUSEN, A.; DOELMAN, J.; STEHFEST, E. Anthropogenic land use estimates for the Holocene – HYDE 3.2. **Earth System Science Data**, v. 9, n. 2, p. 927–953, 1 dez. 2017.

GONZÁLEZ, M. O. A.; GONÇALVES, J. S.; VASCONCELOS, R. M. Sustainable development: Case study in the implementation of renewable energy in Brazil. **Journal of Cleaner Production**, v. 142, p. 461–475, 2017.

GOTSOPOULOS, A.; SAARIMÄKI, H.; GLERAN, E.; JÄÄSKELÄINEN, I. P.; SAMS, M.; NUMMENMAA, L.; LAMPINEN, J. Reproducibility of importance extraction methods in neural network based fMRI classification. **NeuroImage**, v. 181, p. 44–54, nov. 2018.

GREENBERG, M.; LEWIS, M. J. Brownfields Redevelopment, Preferences and Public Involvement: A Case Study of an Ethically Mixed Neighbourhood. **Urban Studies**, v. 37, n. 13, p. 2501–2514, 2000.

GREENSMITH, J.; AICKELIN, U.; TWYCROSS, J. **Articulation and Clarification of the Dendritic Cell Algorithm**. (H. Bersini & J. Carneiro, Eds.) 5th International Conference on Artificial Immune Systems. ICARIS 2006. : Lecture Notes in Computer Science. Oeiras, Portugal: Springer, 2006.

GUHA, S.; RASTOGI, R.; SHIM, K. **CURE: An Efficient Clustering Algorithm for Large Databases**. Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data - SIGMOD '98. New York: ACM, 1998.

\_\_\_\_\_. Rock: a robust clustering algorithm for categorical attributes. **Information Systems**, v. 25, n. 5, p. 345–366, 2000.

GUNASEKARAN, A.; SUBRAMANIAN, N. Sustainable operations modeling and data analytics. **Computers & Operations Research**, v. 89, p. 163–167, jan. 2018.

GUO, Y.; HAN, S.; LI, Y.; ZHANG, C.; BAI, Y. K-Nearest Neighbor combined with guided filter for hyperspectral image classification. **Procedia Computer Science**, v. 129, p. 159–165, 2018.

GUPTA, D. K.; VASUDEV, K. L.; BHATTACHARYYA, S. K. Genetic algorithm optimization based nonlinear ship maneuvering control. **Applied Ocean Research**, v. 74, p. 142–153, maio 2018.

GUTIERREZ, H. D. LA T.; PHAM, D. T. Estimation and generation of training patterns for control chart pattern recognition. **Computers & Industrial Engineering**, v. 95, p. 72–82, maio 2016.

HADI, W.; AL-RADAIDEH, Q. A.; ALHAWARI, S. Integrating associative rule-based classification with Naïve Bayes for text classification. **Applied Soft Computing**, v. 69, p. 344–356, ago. 2018.

HALKIDI, M.; BATISTAKIS, Y.; VAZIRGIANNIS, M. On clustering validation techniques. **Journal of Intelligent Information Systems**, v. 17, n. 2–3, p. 107–145, 2001.

HAO, H.; ZHANG, Q.; WANG, Z.; ZHANG, J. Forecasting the number of end-of-life vehicles using a hybrid model based on grey model and artificial neural network.

**Journal of Cleaner Production**, v. 202, p. 684–696, 2018.

HARTMANN, B.; TÖRÖK, S.; BÖRCSÖK, E.; OLÁHNÉ GROMA, V. Multi-objective method for energy purpose redevelopment of brownfield sites. **Journal of Cleaner Production**, v. 82, p. 202–212, nov. 2014.

HATHAWAY, R. J.; BEZDEK, J. C. Grouped coordinate minimization using Newton's method for inexact minimization in one vector coordinate. **Journal of Optimization Theory and Applications**, v. 71, n. 3, p. 503–516, 1991.

HAYKIN, S. **Neural networks and learning machines**. 3. ed. Upper Saddle River, New Jersey: Pearson, 2009.

HEBB, D. O. **The Organization of Behavior: A Neuropsychological Theory**. New York: Wiley, 1949.

HINNEBURG, A.; KEIM, D. A. **An efficient approach to clustering in large multimedia databases with noise**. Proceedings of 4th International Conference on Knowledge Discovery and Data Mining - KDD '98. New York: AAAI Press, 1998.

HINNEBURG, A.; KEIM, D. A. A general approach to clustering in large databases with noise. **Knowledge and Information Systems**, v. 5, n. 4, p. 387–415, 2003.

HOLLAND, J. H. **Adaptation in Natural and Artificial Systems**. [s.l.] The MIT Press, 1992.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural Networks**, v. 2, n. 5, p. 359–366, jan. 1989.

HUANG, Z. Extensions to the k-means algorithm for clustering large data sets with categorical values. **Data Mining and Knowledge Discovery**, v. 2, n. 3, p. 283–304, 1998.

HUNG, W.-L.; YANG, M.-S. Similarity measures of intuitionistic fuzzy sets based on Hausdorff distance. **Pattern Recognition Letters**, v. 25, n. 14, p. 1603–1611, out. 2004.

INTERNATIONAL RENEWABLE ENERGY AGENCY. **Renewable Energy Statistics 2018**. Abu Dhabi: [s.n.]. Disponível em: <<https://bit.ly/2MmLqPk>>. Acesso em: 11 set. 2018a.

\_\_\_\_\_. **Renewable Energy and Jobs Annual Review 2018**. Abu Dhabi: [s.n.]. Disponível em: <[www.irena.org](http://www.irena.org)>. Acesso em: 5 jun. 2018b.

JACKOWSKI, K. New diversity measure for data stream classification ensembles. **Engineering Applications of Artificial Intelligence**, v. 74, p. 23–34, set. 2018.

JAEHN, F. Sustainable Operations. **European Journal of Operational Research**, v. 253, n. 2, p. 243–264, set. 2016.

KACPRZYK, J.; PEDRYCZ, W. (EDS.). **Springer Handbook of Computational Intelligence**. Berlin: Springer, 2015.

KARABOGA, D.; BASTURK, B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. **Journal of Global**

**Optimization**, v. 39, n. 3, p. 459–471, 9 out. 2007.

KARRAY, F. O.; SILVA, C. DE. **Soft Computing and Intelligent Systems Design: Theory, Tools, and Applications**. Harlow: Addison-Wesley, 2004.

KARYPIS, G.; EUI-HONG HAN; KUMAR, V. Chameleon: hierarchical clustering using dynamic modeling. **Computer**, v. 32, n. 8, p. 68–75, 1999.

KAUFMAN, M. M.; ROGERS, D. T.; MURRAY, K. S. An empirical model for estimating remediation costs at contaminated sites. **Water, Air, and Soil Pollution**, v. 167, n. 1–4, p. 365–386, 2005.

KENNEDY, J.; EBERHART, R. **Particle swarm optimization**. Proceedings of ICNN'95 - International Conference on Neural Networks. Perth, Australia: IEEE, 1995.

KENNEDY, J.; EBERHART, R. C.; SHI, Y. **Swarm intelligence**. San Francisco: Morgan Kaufmann, 2001.

KHOSHROO, A.; EMROUZNEJAD, A.; GHAFFARIZADEH, A.; KASRAEI, M.; OMID, M. Sensitivity analysis of energy inputs in crop production using artificial neural networks. **Journal of Cleaner Production**, v. 197, p. 992–998, out. 2018.

KIM, H.; KIM, J.; KIM, J.; LIM, P. Towards perfect text classification with Wikipedia-based semantic Naïve Bayes learning. **Neurocomputing**, v. 315, p. 128–134, nov. 2018.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. **Science**, v. 220, n. 4598, p. 671–680, 1983.

KNUTAS, A.; HAJIKHANI, A.; SALMINEN, J.; IKONEN, J.; PORRAS, J. **Cloud-based bibliometric analysis service for systematic mapping studies**. (B. Rachev & A. Smrikarov, Eds.) Proceedings of the 16th International Conference on Computer Systems and Technologies - CompSysTech '15. Dublin, Ireland: ACM Press, 2015.

KOVACS, H.; SZEMMELVEISZ, K. Disposal options for polluted plants grown on heavy metal contaminated brownfield lands: A review. **Chemosphere**, v. 166, p. 8–20, 2017.

KOZA, J. Genetic programming as a means for programming computers by natural selection. **Statistics and Computing**, v. 4, n. 2, p. 87–112, 1994.

KRUSE, R.; BORGELT, C.; BRAUNE, C.; MOSTAGHIM, S.; STEINBRECHER, M. **Computational Intelligence: A Methodological Introduction**. 2. ed. London: Springer, 2016.

LAMBIN, E. F.; MEYFROIDT, P. Global land use change, economic globalization, and the looming land scarcity. **Proceedings of the National Academy of Sciences of the United States of America**, v. 108, n. 9, p. 3465–3472, 2011.

LEE, L. C.; LIONG, C.-Y.; JEMAIN, A. A. Validity of the best practice in splitting data for hold-out validation strategy as performed on the ink strokes in the context of forensic science. **Microchemical Journal**, v. 139, p. 125–133, jun. 2018.

LEE, S.-H.; PARK, S.; KIM, T. Review on investment direction of green technology

R&D in Korea. **Renewable and Sustainable Energy Reviews**, v. 50, p. 186–193, out. 2015.

LEŚNIAK, A.; JUSZCZYK, M. Prediction of site overhead costs with the use of artificial neural network based model. **Archives of Civil and Mechanical Engineering**, v. 18, n. 3, p. 973–982, jul. 2018.

LEVENBERG, K. A Method for the Solution of Certain Non-Linear Problems in Least Squares. **Quarterly of Applied Mathematics**, v. 2, n. 2, p. 164–168, 1944.

LI, H.; LI, G.; JI, X.; SHI, L. Deep representation via convolutional neural network for classification of spatiotemporal event streams. **Neurocomputing**, v. 299, p. 1–9, jul. 2018.

LI, X.; JIAO, W.; XIAO, R.; CHEN, W.; LIU, W. Contaminated sites in China: Countermeasures of provincial governments. **Journal of Cleaner Production**, v. 147, p. 485–496, 2017.

LI, X.; SHAO, Z.; QIAN, J. An Optimizing Method Based on Autonomous Animats: Fish-swarm Algorithm. **Systems Engineering - Theory & Practice**, v. 22, p. 32–38, 2002.

LIMA, F.; FERREIRA, P.; VIEIRA, F. Strategic impact management of wind power projects. **Renewable and Sustainable Energy Reviews**, v. 25, p. 277–290, 2013.

LINDMAN, Å.; SÖDERHOLM, P. Wind energy and green economy in Europe: Measuring policy-induced innovation using patent data. **Applied Energy**, v. 179, p. 1351–1359, out. 2016.

LU, S.; SHANG, Y.; LI, Y. A research on the application of fuzzy iteration clustering in the water conservancy project. **Journal of Cleaner Production**, v. 151, p. 356–360, maio 2017.

LUO, Y.; WANG, Z.; WEI, G.; ALSAADI, F. E.; HAYAT, T. State estimation for a class of artificial neural networks with stochastically corrupted measurements under Round-Robin protocol. **Neural Networks**, v. 77, p. 70–79, maio 2016.

MACKAY, D. J. C. Bayesian Interpolation. **Neural Computation**, v. 4, n. 3, p. 415–447, maio 1992.

MACQUEEN, J. **Some methods for classification and analysis of multivariate observations**. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Berkeley: University of California Press, 1967.

MAMMADLI, S. Financial time series prediction using artificial neural network based on Levenberg-Marquardt algorithm. **Procedia Computer Science**, v. 120, p. 602–607, 2017.

MAN, Z.; LEE, K.; WANG, D.; CAO, Z.; MIAO, C. A new robust training algorithm for a class of single-hidden layer feedforward neural networks. **Neurocomputing**, v. 74, n. 16, p. 2491–2501, set. 2011.

MANAN, Z. A.; MOHD NAWI, W. N. R.; WAN ALWI, S. R.; KLEMEŠ, J. J. Advances in Process Integration research for CO<sub>2</sub> emission reduction – A review. **Journal of**

**Cleaner Production**, v. 167, p. 1–13, nov. 2017.

MANCUHAN, K.; CLIFTON, C. Support vector classification with  $\ell$ -diversity. **Computers & Security**, v. 77, p. 653–665, ago. 2018.

MANZANO-AGUGLIARO, F.; SANCHEZ-MUROS, M. J.; BARROSO, F. G.; MARTÍNEZ-SÁNCHEZ, A.; ROJO, S.; PÉREZ-BAÑÓN, C. Insects for biodiesel production. **Renewable and Sustainable Energy Reviews**, v. 16, n. 6, p. 3744–3753, 2012.

MARQUARDT, D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. **Journal of the Society for Industrial and Applied Mathematics**, v. 11, n. 2, p. 431–441, 1963.

MARUGÁN, A. P.; MÁRQUEZ, F. P. G.; PEREZ, J. M. P.; RUIZ-HERNÁNDEZ, D. A survey of artificial neural network in wind energy systems. **Applied Energy**, v. 228, p. 1822–1836, out. 2018.

MASSART, P. **Concentration Inequalities and Model Selection**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. v. 1896

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115–133, dez. 1943.

MEHDIYEV, N.; ENKE, D.; FETTKE, P.; LOOS, P. Evaluating Forecasting Methods by Considering Different Accuracy Measures. **Procedia Computer Science**, v. 95, p. 264–271, 2016.

MEHRABIAN, A. R.; LUCAS, C. A novel numerical optimization algorithm inspired from weed colonization. **Ecological Informatics**, v. 1, n. 4, p. 355–366, 2006.

MEILĀ, M. Comparing clusterings: an information based distance. **Journal of Multivariate Analysis**, v. 98, n. 5, p. 873–895, 2007.

MEISEL, S.; MATTFELD, D. Synergies of Operations Research and Data Mining. **European Journal of Operational Research**, v. 206, n. 1, p. 1–10, out. 2010.

MIAO, C.; FANG, D.; SUN, L.; LUO, Q. Natural resources utilization efficiency under the influence of green technological innovation. **Resources, Conservation and Recycling**, v. 126, n. March, p. 153–161, nov. 2017.

MIROŃCZUK, M. M.; PROTASIEWICZ, J. A recent overview of the state-of-the-art elements of text classification. **Expert Systems with Applications**, v. 106, p. 36–54, set. 2018.

MITCHELL, H. . On the Dengfeng–Chuntian similarity measure and its application to pattern recognition. **Pattern Recognition Letters**, v. 24, n. 16, p. 3101–3104, dez. 2003.

MORIO, M.; SCHÄDLER, S.; FINKEL, M. Applying a multi-criteria genetic algorithm framework for brownfield reuse optimization: Improving redevelopment options based on stakeholder preferences. **Journal of Environmental Management**, v. 130, p. 331–346, 2013.



MOSCATO, P.; COTTA, C. A Modern Introduction to Memetic Algorithms. *In*: GENDREAU, M.; POTVIN, J.-Y. (Eds.). . **Handbook of Metaheuristics**. 2. ed. Boston: Springer, 2010. v. 146p. 141–183.

MÜLLER, P. *et al.* Scent classification by K nearest neighbors using ion-mobility spectrometry measurements. **Expert Systems with Applications**, v. 115, p. 593–606, jan. 2019.

NANDA, S. J.; PANDA, G. A survey on nature inspired metaheuristic algorithms for partitional clustering. **Swarm and Evolutionary Computation**, v. 16, p. 1–18, jun. 2014.

NAPOLITANO, G.; SERINALDI, F.; SEE, L. Impact of EMD decomposition and random initialisation of weights in ANN hindcasting of daily stream flow series: An empirical examination. **Journal of Hydrology**, v. 406, n. 3–4, p. 199–214, set. 2011.

NASR, M. BEN; CHTOUROU, M. A self-organizing map-based initialization for hybrid training of feedforward neural networks. **Applied Soft Computing**, v. 11, n. 8, p. 4458–4464, dez. 2011.

NG, K. K. H.; LEE, C. K. M.; CHAN, F. T. S.; LV, Y. Review on meta-heuristics approaches for airside operation research. **Applied Soft Computing**, v. 66, p. 104–133, maio 2018.

NG, R. T.; HAN, J. **Efficient and effective clustering methods for spatial data mining**. Proceedings of the 20th International Conference on Very Large Data Bases - VLDB '94. San Francisco: Morgan Kaufmann, 1994.

NUISSL, H.; SCHROETER-SCHLAACK, C. On the economic approach to the containment of land consumption. **Environmental Science and Policy**, v. 12, n. 3, p. 270–280, 2009.

OLAFSSON, S.; LI, X.; WU, S. Operations research and data mining. **European Journal of Operational Research**, v. 187, n. 3, p. 1429–1448, jun. 2008.

ONAT, N. C.; KUCUKVAR, M.; TATARI, O. Integrating triple bottom line input-output analysis into life cycle sustainability assessment framework: The case for US buildings. **International Journal of Life Cycle Assessment**, v. 19, n. 8, p. 1488–1505, 2014.

ORRIOLS-PUIG, A.; MARTÍNEZ-LÓPEZ, F. J.; CASILLAS, J.; LEE, N. Unsupervised KDD to creatively support managers' decision making with fuzzy association rules: A distribution channel application. **Industrial Marketing Management**, v. 42, n. 4, p. 532–543, 2013.

OZKAN, I.; TURKSEN, I. B. Upper and lower values for the level of fuzziness in FCM. **Information Sciences**, v. 177, n. 23, p. 5143–5152, dez. 2007.

PAL, N. R.; BEZDEK, J. C.; TSAO, E. C.-K. Generalized Clustering Networks and Kohonen's Self-Organizing Scheme. **IEEE Transactions on Neural Networks**, v. 4, n. 4, p. 549–557, 1993.

PASSINO, K. M. Biomimicry of bacterial foraging for distributed optimization and control. **IEEE Control Systems Magazine**, v. 22, n. 3, p. 52–67, jun. 2002.

PEREA-MORENO, A.-J.; GARCÍA-CRUZ, A.; NOVAS, N.; MANZANO-AGUGLIARO, F. Rooftop analysis for solar flat plate collector assessment to achieving sustainability energy. **Journal of Cleaner Production**, v. 148, p. 545–554, 2017.

POLI, R. Analysis of the Publications on the Applications of Particle Swarm Optimisation. **Journal of Artificial Evolution and Applications**, v. 2008, p. 1–10, 2008.

PRICE, K. V.; STORN, R. M.; LAMPINEN, J. A. **Differential Evolution**. Berlin: Springer, 2005.

QIAO, J.; LI, S.; LI, W. Mutual information based weight initialization method for sigmoidal feedforward neural networks. **Neurocomputing**, v. 207, p. 676–683, set. 2016.

QING SONG. Robust Initialization of a Jordan Network With Recurrent Constrained Learning. **IEEE Transactions on Neural Networks**, v. 22, n. 12, p. 2460–2473, dez. 2011.

RABANAL, P.; RODRÍGUEZ, I.; RUBIO, F. Using River Formation Dynamics to Design Heuristic Algorithms. *In*: AKL, S. G.; CALUDE, C. S.; DINNEEN, M. J.; ROZENBERG, G.; WAREHAM, H. T. (Eds.). . **Unconventional Computation**. Berlin: Springer, 2007. p. 163–177.

RASHEDI, E.; NEZAMABADI-POUR, H.; SARYAZDI, S. GSA: A Gravitational Search Algorithm. **Information Sciences**, v. 179, n. 13, p. 2232–2248, jun. 2009.

REN, L.; GUO, M.; PANG, X. Identification and classification of medicinal plants in Epimedium. **Chinese Herbal Medicines**, v. 10, n. 3, p. 249–254, jul. 2018.

REYNOLDS, R. G. **An introduction to cultural algorithms**. (A. V. Sebald & L. J. Fogel, Eds.) Proceedings of the Third Annual Conference on Evolutionary Programming. San Diego, USA: World Scientific, 1994.

RINTA-KOSKI, O.-P.; SÄRKKÄ, S.; HOLLMÉN, J.; LESKINEN, M.; ANDERSSON, S. Gaussian process classification for prediction of in-hospital mortality among preterm infants. **Neurocomputing**, v. 298, p. 134–141, jul. 2018.

ROIGER, R. J. **Data Mining: A Tutorial-Based Primer**. 2. ed. Boca Raton: CRC Press, 2017.

RONG, L.; ZHANG, C.; JIN, D.; DAI, Z. Assessment of the potential utilization of municipal solid waste from a closed irregular landfill. **Journal of Cleaner Production**, v. 142, p. 413–419, 2017.

ROS, F.; PINTORE, M.; DEMAN, A.; CHRÉTIEN, J. R. Automatical initialization of RBF neural networks. **Chemometrics and Intelligent Laboratory Systems**, v. 87, n. 1, p. 26–32, maio 2007.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, n. 6088, p. 533–536, 9 out. 1986.

SÁ JUNIOR, J. J. DE M.; BACKES, A. R.; BRUNO, O. M. Randomized neural network based descriptors for shape classification. **Neurocomputing**, v. 312, p. 201–209, out. 2018.

SAMMUT, C.; WEBB, G. I.; WEBB, G. I. **Encyclopedia of Machine Learning and Data Mining**. Boston, MA: Springer US, 2017.

SANNINO, G.; PIETRO, G. DE. A deep learning approach for ECG-based heartbeat classification for arrhythmia detection. **Future Generation Computer Systems**, v. 86, p. 446–455, set. 2018.

SEO, Y.; SHIN, K. Hierarchical convolutional neural networks for fashion image classification. **Expert Systems with Applications**, v. 116, p. 328–339, fev. 2019.

SEVER, A. An inverse problem approach to pattern recognition in industry. **Applied Computing and Informatics**, v. 11, n. 1, p. 1–12, jan. 2015.

SHARMA, N.; JAIN, V.; MISHRA, A. An Analysis Of Convolutional Neural Networks For Image Classification. **Procedia Computer Science**, v. 132, p. 377–384, 2018.

SHEIKHOESLAMI, G.; CHATTERJEE, S.; ZHANG, A. **Wavecluster: A multi-Resolution Clustering Approach for Very Large Spatial Databases**. Proceedings of 24rd International Conference on Very Large Data Bases - VLDB '98. San Francisco: Morgan Kaufmann, 1998.

SHENFIELD, A.; DAY, D.; AYESH, A. Intelligent intrusion detection systems using artificial neural networks. **ICT Express**, v. 4, n. 2, p. 95–99, jun. 2018.

SIDDIQUE, N.; ADELI, H. **Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing**. Oxford: Wiley, 2013.

SIMAS, M.; PACCA, S. Energia eólica, geração de empregos e desenvolvimento sustentável. **Estudos Avançados**, v. 27, n. 77, p. 99–116, 2013.

SIMONCINI, M.; TACCARI, L.; SAMBO, F.; BRAVI, L.; SALTI, S.; LORI, A. Vehicle classification from low-frequency GPS data with recurrent neural networks. **Transportation Research Part C: Emerging Technologies**, v. 91, p. 176–191, jun. 2018.

SODHI, S. S.; CHANDRA, P. Interval based Weight Initialization Method for Sigmoidal Feedforward Artificial Neural Networks. **AASRI Procedia**, v. 6, p. 19–25, 2014.

SONI, A.; GUPTA, R.; GUPTA, S.; KANSAY, R.; KAPOOR, L. Mechanism of injury based classification of proximal tibia fractures. **Journal of Clinical Orthopaedics and Trauma**, p. 10–13, ago. 2018.

SOUSA, C. A. DE. Turning brownfields into green space in the City of Toronto. **Landscape and Urban Planning**, v. 62, n. 4, p. 181–198, 2003.

SOVIZI, J.; MATHIEU, K. B.; THROWER, S. L.; STEFAN, W.; HAZLE, J. D.; FUENTES, D. Gaussian process classification of superparamagnetic relaxometry data: Phantom study. **Artificial Intelligence in Medicine**, v. 82, p. 47–59, out. 2017.

STEINER, M. T. A.; DATTA, D.; STEINER NETO, P. J.; SCARPIN, C. T.; RUI

FIGUEIRA, J. Multi-objective optimization in partitioning the healthcare system of Parana State in Brazil. **Omega**, v. 52, p. 53–64, abr. 2015.

STETCO, A.; ZENG, X.-J.; KEANE, J. Fuzzy C-means++: Fuzzy C-means with effective seeding initialization. **Expert Systems with Applications**, v. 42, n. 21, p. 7541–7548, nov. 2015.

STOFFEL, M.; BAMER, F.; MARKERT, B. Artificial neural networks and intelligent finite elements in non-linear structural mechanics. **Thin-Walled Structures**, v. 131, n. July, p. 102–106, out. 2018.

STORN, R. M. **On the usage of differential evolution for function optimization**. Proceedings of North American Fuzzy Information Processing. IEEE, 1996.

STORN, R. M.; PRICE, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341–359, 1997.

STRAALEN, N. M. VAN. Assessment of soil contamination: a functional perspective. **Biodegradation**, v. 13, n. 1, p. 41–52, 2002.

SUDHA, P.; RAMYACHITRA, D.; MANIKANDAN, P. Enhanced Artificial Neural Network for Protein Fold Recognition and Structural Class Prediction. **Gene Reports**, v. 12, n. July, p. 261–275, set. 2018.

TALAŚKA, T.; KOLASA, M.; DŁUGOSZ, R.; FARINE, P.-A. An efficient initialization mechanism of neurons for Winner Takes All Neural Network implemented in the CMOS technology. **Applied Mathematics and Computation**, v. 267, p. 119–138, set. 2015.

TIMOTHEOU, S. A novel weight initialization method for the random neural network. **Neurocomputing**, v. 73, n. 1–3, p. 160–168, dez. 2009.

TING, F. F.; TAN, Y. J.; SIM, K. S. Convolutional neural network improvement for breast cancer classification. **Expert Systems with Applications**, v. 120, p. 103–115, abr. 2019.

TÕNURIST, P. Framework for analysing the role of state owned enterprises in innovation policy management: The case of energy technologies and Eesti Energia. **Technovation**, v. 38, p. 1–14, abr. 2015.

TRELEA, I. C. The particle swarm optimization algorithm: convergence analysis and parameter selection. **Information Processing Letters**, v. 85, n. 6, p. 317–325, mar. 2003.

TUTMEZ, B. Bauxite quality classification by shrinkage methods. **Journal of Geochemical Exploration**, v. 191, p. 22–27, ago. 2018.

U.S. ENERGY INFORMATION ADMINISTRATION. **International Energy Outlook 2017**. [s.l.: s.n.]. Disponível em: <<https://goo.gl/hTG93V>>. Acesso em: 8 nov. 2018.

U.S. ENVIRONMENTAL PROTECTION AGENCY. **RE-Powering America's Land Initiative: Project Tracking Matrix**. Washington, DC: [s.n.]. Disponível em: <<https://bit.ly/2N5ZmCz>>. Acesso em: 11 set. 2018.

U.S. GOVERNMENT PUBLISHING OFFICE. **Public Law 107-118 - Small Business Liability Relief and Brownfields Revitalization Act**. Disponível em: <<https://goo.gl/UK19n2>>. Acesso em: 5 jun. 2018.

\_\_\_\_\_. **40 C.F.R. 239-282 - Solid Wastes**. Disponível em: <<https://goo.gl/UBCLDF>>. Acesso em: 5 jun. 2018.

\_\_\_\_\_. **42 U.S.C. 9601-9628 - Hazardous Substances Releases, Liability, Compensation**. Disponível em: <<https://goo.gl/y0ki6N>>. Acesso em: 5 jun. 2018.

VADIM, K. Overview of different approaches to solving problems of Data Mining. **Procedia Computer Science**, v. 123, p. 234–239, 2018.

VAKILI, M.; KARAMI, M.; DELFANI, S.; KHOSROJERDI, S. Experimental investigation and modeling of thermal radiative properties of f-CNTs nanofluid by artificial neural network with Levenberg–Marquardt algorithm. **International Communications in Heat and Mass Transfer**, v. 78, p. 224–230, nov. 2016.

VEENMAN, C. J.; REINDERS, M. J. T.; BACKER, E. A maximum variance *cluster* algorithm. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 24, n. 9, p. 1273–1280, set. 2002.

VILLARRUBIA, G.; PAZ, J. F. DE; CHAMOSO, P.; PRIETA, F. DE LA. Artificial neural networks used in optimization problems. **Neurocomputing**, v. 272, p. 10–16, jan. 2018.

VIVANCO, D. F.; KEMP, R.; VOET, E. VAN DER. The relativity of eco-innovation: environmental rebound effects from past transport innovations in Europe. **Journal of Cleaner Production**, v. 101, p. 71–85, ago. 2015.

WANG, W.; YANG, J.; MUNTZ, R. R. **STING: A Statistical Information Grid Approach to Spatial Data Mining**. Proceedings of 23rd International Conference on Very Large Data Bases - VLDB '97. San Francisco: Morgan Kaufmann, 1997.

WHITLEY, D. An overview of evolutionary algorithms: practical issues and common pitfalls. **Information and Software Technology**, v. 43, n. 14, p. 817–831, dez. 2001.

WIDROW, B.; HOFF, M. E. **Adaptive Switching Circuits**. Stanford: [s.n.].

WITTEN, I. H.; FRANK, E.; HALL, M. A.; PAL, C. J. **Data Mining**. 4. ed. Cambridge: Elsevier, 2017.

WU, K.-L. Analysis of parameter selections for fuzzy c-means. **Pattern Recognition**, v. 45, n. 1, p. 407–415, jan. 2012.

WU, W.; LI, A.-D.; HE, X.-H.; MA, R.; LIU, H.-B.; LV, J.-K. A comparison of support vector machines, artificial neural network and classification tree for identifying soil texture classes in southwest China. **Computers and Electronics in Agriculture**, v. 144, p. 86–93, jan. 2018.

XING, B.; GAO, W.-J. **Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms**. Cham: Springer, 2014.a. v. 62

\_\_\_\_\_. Cat Swarm Optimization Algorithm. *In*: YANG, Q.; WEBB, G. (Eds.). . **9th Pacific**

**Rim International Conference on Artificial Intelligence. PRICAI 2006.** Lecture Notes in Artificial Intelligence. Guilin, China: Springer, 2014. b. v. 4099p. 93–104.

XU, R.; WUNSCH II, D. Survey of Clustering Algorithms. **IEEE Transactions on Neural Networks**, v. 16, n. 3, p. 645–678, 2005.

YABAR, H.; HARA, K.; UWASU, M. Comparative assessment of the co-evolution of environmental indicator systems in Japan and China. **Resources, Conservation and Recycling**, v. 61, p. 43–51, abr. 2012.

YADAV, S.; SHUKLA, S. **Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification.** 6th International Conference on Advanced Computing (IACC 2016). Bhimavaram: IEEE, fev. 2016. Disponível em: <<http://ieeexplore.ieee.org/document/7544814/>>

YAM, Y. F.; CHOW, T. W. S. Determining initial weights of feedforward neural networks based on least squares method. **Neural Processing Letters**, v. 2, n. 2, p. 13–17, mar. 1995.

YANG, X.-S. Firefly Algorithms for Multimodal Optimization. *In*: WATANABE, O.; ZEUGMANN, T. (Eds.). **Proceedings of the 5th International Conference on Stochastic Algorithms: Foundations and Applications.** Sapporo: Springer, 2009. p. 169–178.

YE, Z.; KIM, M. K. Predicting electricity consumption in a building using an optimized back-propagation and Levenberg–Marquardt back-propagation neural network: Case study of a shopping mall in China. **Sustainable Cities and Society**, v. 42, n. March, p. 176–183, out. 2018.

YEH, I.-C.; LIEN, C. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. **Expert Systems with Applications**, v. 36, n. 2, p. 2473–2480, mar. 2009.

YEN, P. C. P.; FAN, K.; CHAO, H. C. J. A new method for similarity measures for pattern recognition. **Applied Mathematical Modelling**, v. 37, n. 7, p. 5335–5342, abr. 2013.

YUAN, H.; LI, J.; LAI, L. L.; TANG, Y. Y. Graph-based multiple rank regression for image classification. **Neurocomputing**, v. 315, p. 394–404, nov. 2018.

ZAHN, C. T. Graph-Theoretical Methods for Detecting and Describing Gestalt *Clusters*. **IEEE Transactions on Computers**, v. C-20, n. 1, p. 68–86, jan. 1971.

ZAKI, M. J.; MEIRA JR., W. **Data Mining and Analysis: Fundamental Concepts and Algorithms.** [s.l.] Cambridge University Press, 2014.

ZGUROVSKY, M. Z.; ZAYCHENKO, Y. P. **The Fundamentals of Computational Intelligence: System Approach.** Cham: Springer, 2017. v. 652

ZHANG, H.; YUAN, M.; LIANG, Y.; LIAO, Q. A novel particle swarm optimization based on prey–predator relationship. **Applied Soft Computing**, v. 68, p. 202–218, jul. 2018.

ZHANG, L.; TIAN, F. Performance Study of Multilayer Perceptrons in a Low-Cost Electronic Nose. **IEEE Transactions on Instrumentation and Measurement**, v. 63,

n. 7, p. 1670–1679, jul. 2014.

ZHANG, L.; TIAN, F.; PEI, G. A novel sensor selection using pattern recognition in electronic nose. **Measurement**, v. 54, p. 31–39, ago. 2014.

ZHANG, T.; RAMAKRISHNAN, R.; LIVNY, M. **BIRCH: An Efficient Data Clustering Databases Method for Very Large**. Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data - SIGMOD '96. New York: ACM, 1996.

ZHANG, X.; LI, Y.; KOTAGIRI, R.; WU, L.; TARI, Z.; CHERIET, M. KRNN: k Rare-class Nearest Neighbour classification. **Pattern Recognition**, v. 62, p. 33–44, fev. 2017.

ZHOU, K.; FU, C.; YANG, S. Fuzziness parameter selection in fuzzy c-means: The perspective of *cluster* validation. **Science China Information Sciences**, v. 57, n. 11, p. 1–8, 4 nov. 2014.

ZHOU, K.; YANG, S.; SHAO, Z. Household monthly electricity consumption pattern mining: A fuzzy clustering-based model and a case study. **Journal of Cleaner Production**, v. 141, p. 900–908, jan. 2017.