

Um Comitê de Classificadores SVM para Diagnóstico de Falhas em Motobombas baseado em Técnicas de Soft-Computing

Jeferson de Oliveira Batista

Universidade Federal do Espírito Santo
Av. Fernando Ferrari s/n, 29075-910 Vitória, ES, Brasil
jeferson.de.oliveira.batista@gmail.com

Rodrigo Biancardi Rodrigues

Universidade Federal do Espírito Santo
Av. Fernando Ferrari s/n, 29075-910 Vitória, ES, Brasil
rodrigorbgr@gmail.com

Flávio Miguel Varejão

Universidade Federal do Espírito Santo
Av. Fernando Ferrari s/n, 29075-910 Vitória, ES, Brasil
fmvarejao@gmail.com

RESUMO

Neste trabalho investigamos o uso de técnicas de *soft-computing* em um processo de formação de um comitê de classificadores SVM, que integraria um sistema de diagnóstico de falhas em complexas máquinas industriais. O processo de formação consiste em três estágios. No primeiro estágio, utilizamos um algoritmo genético para investigar o espaço global de características, a fim de encontrar um conjunto de subconjuntos das mesmas, para formar um comitê de classificadores acurados e divergentes. No segundo estágio, utilizamos o algoritmo PSO para otimizar os hiperparâmetros utilizados para cada subconjunto de características. E, por fim, no terceiro estágio, utilizamos os algoritmos CLONALG e SA para a seleção dos classificadores que irão compor o ensemble final. Em nossos experimentos, foram utilizados dados extraídos de sinais de sensores em bombas de plataformas petrolíferas. Os resultados mostram que nosso método é mais eficiente que outros bem estabelecidos na resolução de problemas semelhantes.

PALAVRAS CHAVE. Computação Evolutiva, Comitês de Classificadores, Soft-Computing, Seleção de Características

Pesquisa Operacional na Área de Petróleo e Gás, Metaheurísticas

ABSTRACT

In this work we investigate the use of soft-computing techniques in a process to build a SVM classifiers ensemble, that integrates a complex industrial machine fault diagnosis system. The process to build have three stages. In the first, we use a genetic algorithm to investigate the feature global space, to find a set of feature subsets, with the goal to build an ensemble with acurated and diversified classifiers. In the second stage, we use the PSO algorithm to tuning the hyperparameters applied to each feature subset. And, at the end, in the third stage, we use the CLONALG and SA algorithms to select the classifiers that will compose the final ensemble. In our experiments, we use extracted data of sensor signals in oil rigs motor pumps. The results show that our method is more efficient that others well-established methods to solve similar problems.

KEYWORDS. Evolutionary Computation. Classifier Ensembles. Soft-Computing. Feature Selection

Operational Research in Oil and Gas Area, Metaheuristics

1. Introdução

Detectar e diagnosticar falhas em máquinas industriais complexas é vantajoso por razões de segurança e econômicas [Dounias et al., 2001]. A detecção preventiva de uma falha permite reparar componentes danificados durante a manutenção planejada, o que minimiza perdas no maquinário enquanto provê operações mais seguras. O recente progresso na tecnologia de sensores e inteligência computacional permite a construção de sistemas de diagnóstico mais poderosos, o que pode ajudar relativamente operadores inexperientes na tomada de decisões confiáveis sobre a condição da máquina tanto quanto prover informações valiosas para experts na tomada de decisões sobre intrincadas ocorrências de falhas. [Wandekoken et al., 2011]

Nosso trabalho se apoia em uma base de dados com dados reais, com aquisições de sinais de vibração extraídos de motobombas operacionais. Há duas aproximações principais ao problema de diagnóstico de falhas: técnicas baseadas em modelos e técnicas livres de modelo. A linha de pesquisa baseada em modelos se baseia num modelo analítico do processo estudado, envolvendo equações diferenciais dependentes do tempo. Usualmente o processo experimental é instalado em um laboratório controlado e é incorporado em uma malha de controle em que insumos, variáveis controladas e saídas de sensores são modelados. Porém, em processos do mundo real a disponibilidade de modelos analíticos frequentemente é irreal ou tais modelos são imprecisos devido à complexidade do processo. Nesse caso, técnicas livres de modelo são aproximações alternativas, que se baseiam em reconhecimento de padrões para o aprendizado automático de regras que descrevem as falhas apresentadas nos dados. [Wandekoken et al., 2011]

Para podermos utilizar uma técnica livre de modelo para resolver o problema de diagnóstico de falhas em motobombas, é preciso primeiro extrair um vetor global de características, que são dados numéricos, para descrever aspectos relevantes das condições das motobombas. A estratégia geral é obter o máximo de informações possível na extração de características, filtrar as características mais relevantes e então construir um comitê de classificadores treinados com as características selecionadas. [Wandekoken et al., 2011]

Os dados utilizados no presente trabalho foram obtidos através de sinais de vibração coletados por acelerômetros instalados estrategicamente em pontos próximos aos rolamentos e motores de bombas petrolíferas, permitindo a medição do deslocamento, da velocidade e da aceleração da máquina, ao longo do tempo, gerando um sinal discreto do nível de vibração. Os sinais de vibração no domínio do tempo foram coletados através de um analisador de vibração em conjunto com os sensores e então submetidos a conhecidas técnicas de processamento de sinais como Transformada de Fourier e análise de envelope baseada na transformação de Hilbert.

A primeira base de dados que utilizamos se chama desalinhamento e é composta por 81 atributos (características) numéricos e 1586 instâncias; a segunda, desbalanceamento, também é composta por 81 atributos numéricos e 1305 instâncias e; a terceira, rolamento, é composta por 28 atributos numéricos e 1697 instâncias. Essas bases compõem uma quantidade estatisticamente significativa de dados reais onde pudemos então investigar a efetividade de nossos métodos.

Sistemas de diagnóstico podem ser construídos com a utilização de comitês de classificadores que têm sido utilizados para aumentar a acurácia em decisões de classificação, tais como diagnóstico automático, detecção de fraude, verificação de moeda e recuperação de documentos [Wandekoken et al., 2011]. Cada classificador realiza predições com base no reconhecimento de padrões nos dados que lhe são fornecidos para treinamento. O uso de comitês de classificadores é mais eficiente que o uso de um único classificador [Kuncheva, 2004], desde que os classificadores que compõem o comitê errem em diferentes locais do conjunto de dados [Opitz, 1999]. Vários trabalhos têm empregado diferentes técnicas para construir classificadores que são acurados e divergentes em suas predições o máximo possível [De Oliveira et al., 2010].

Entre as técnicas de formação de comitês de classificadores, temos Bagging e Boosting (particularmente AdaBoost). Bagging é um método de formação de comitês que treina cada classificador com uma partição diferente do conjunto de treino. As partições são geradas através da

seleção aleatória de instâncias, com reposição. Assim como Bagging, a técnica AdaBoost é baseada em classificadores que são treinados em diferentes partições do conjunto de treino, no entanto, essa técnica recalcula a probabilidade de uma instância ser escolhida, com base no número de vezes que tal instância foi classificada corretamente pelo comitê atual. [Schapire, 1999]

Um algoritmo genético que apresentou melhores resultados que a técnica de AdaBoost foi o GEFS (sigla em inglês para *Genetic Ensemble Feature Selection*) [Opitz, 1999]. Tal algoritmo busca obter um conjunto de subconjuntos de características que forme um comitê de classificadores acurados e divergentes, treinando cada classificador em um subconjunto diferente. Para isso, ele utiliza uma função de aptidão (fitness) que avalia, para cada classificador, tanto a acurácia como a divergência que há entre suas predições e as predições do comitê como um todo. A importância dada à diversidade é ajustada de acordo com medições feitas sobre o comitê atual [Opitz, 1999].

Em [Wandekoken et al., 2011], são utilizados classificadores SVM (Support Vector Machine), que é considerado um dos mais poderosos métodos de aprendizado de máquina para resolver problemas de classificação com duas classes [Widodo e Yang, 2007]. Considerando que o SVM é um classificador estável, no sentido de que uma pequena variação no seu conjunto de treino causa apenas uma pequena variação na função de decisão do SVM, uma aproximação mais conveniente e poderosa para gerar diversidade num comitê de SVMs poderia aproveitar a alta sensibilidade de classificadores SVM à variação no conjunto de características com o qual é treinado e à variação dos valores dos hiperparâmetros [Wandekoken et al., 2011].

Dessa forma, em [Wandekoken et al., 2011] é proposto processo com três estágios, sendo que no primeiro estágio, ocorre a seleção de subconjuntos de características para diferentes combinações de valores dos hiperparâmetros. É importante que cada subconjunto possua características relevantes para a classificação e seja suficientemente diferente dos demais subconjuntos para que, com eles, seja construído um comitê com classificadores acurados e os mais divergentes possível. O resultado esperado são diferentes comitês com classificadores acurados e divergentes. No segundo estágio, cada classificador de cada comitê formado é tunado individualmente otimizando-se os valores dos hiperparâmetros para o conjunto de características em que o classificador é treinado, e o resultado é um conjunto com todos os classificadores tunados. E finalmente, no terceiro estágio, é construído um comitê final com um subconjunto otimizado dos classificadores do conjunto do estágio anterior.

No trabalho descrito em [Wandekoken et al., 2011], é utilizado o algoritmo SBS (Sequential Backward Selection) para a seleção de características, a técnica *grid-search* para a otimização dos hiperparâmetros e o algoritmo SFS (Sequential Forward Selection) para a seleção dos classificadores, em um método chamado BSFS (Best Selected Feature Subsets). Em [Wandekoken et al., 2011], o algoritmo SBS foi substituído pela utilização do algoritmo genético GEFS no primeiro estágio, dando origem ao MultiGEFS.

Nós propomos um método que utiliza apenas técnicas de soft-computing em todos os estágios da formação do comitê de classificadores. No primeiro estágio, utilizamos os algoritmos GEFS [Opitz, 1999] para a seleção de características; no segundo estágio, utilizamos o algoritmo PSO [Ren e Bai, 2010] para otimizar os hiperparâmetros e; no terceiro estágio, utilizamos os algoritmos SA e CLONALG para a seleção de classificadores.

Os experimentos realizados mostraram que a nossa proposta obteve melhores resultados que os do método proposto em [Wandekoken et al., 2011] na maioria das situações. Os resultados foram obtidos em uma base de treino e em uma base de teste e estão relatados e comentados na seção 4 - Experimentos e Resultados.

Esse artigo é subdividido nas seguintes seções: na seção 2, Comitês de classificadores baseados em características, apresentamos alguns exemplos de utilização desses comitês. Na seção 3, Propostas, descrevemos detalhadamente nosso método Soft-MultiGEFS. Na seção 4, Experimentos e Resultados, descrevemos os experimentos realizados e apresentamos os resultados obtidos. E, finalmente, na seção 5, Conclusões e Trabalhos Futuros, apresentamos conclusões e traçamos

possíveis continuações para os nossos trabalhos.

2. Comitês de classificadores baseados em características

Pesquisas anteriores têm mostrado que combinando os resultados de diversos classificadores podemos obter predições melhores do que as de um único classificador [Kuncheva, 2004]. Isso ocorre caso os classificadores combinados errem em diferentes locais do conjunto de dados [Opitz, 1999]. Um conjunto de classificadores que têm seus resultados combinados é chamado de comitê de classificadores, mas também é conhecido como sistema de múltiplos classificadores [Chowriappa et al., 2013].

Muitas técnicas têm sido utilizadas para a formação de comitês de classificadores e, como explicado anteriormente, tais técnicas devem prover um conjunto de classificadores que sejam acurados e diversificados em suas predições.

Um modo de construir modelos com representação homogênea, que provou ser efetivo, é o uso de diferentes subconjuntos de características para cada modelo [Tsymbal et al., 2005]. Por isso, o método descrito no presente artigo, se baseia na seleção de características onde cada classificador é treinado. Através da formação apropriada de subconjuntos de características e fazendo com que cada classificador observe apenas um desses subconjuntos, espera-se obter um conjunto de classificadores que apresentem um bom resultado individualmente e errem em diferentes locais do conjunto de testes.

Algo que geralmente é observado em algoritmos de classificação é que o uso de um grande número de características pode fazer com que características irrelevantes exerçam uma influência indesejada em decisões de classificação devido ao tamanho finito da amostra de treino [Ozcift, 2012]. O algoritmo GEFS é um método desenvolvido para a busca de um conjunto de subconjuntos de características apropriado para um comitê de classificadores e foi introduzido em [Opitz, 1999]. GEFS, como um algoritmo de seleção de características, emprega técnicas para selecionar as características mais discriminativas do conjunto de dados para obter um subconjunto menor de características [Ozcift, 2012]. GEFS é um algoritmo genético que, a cada passo, produz novos classificadores candidatos pelo uso das operações genéticas de cruzamento e mutação. Cada indivíduo é definido por um vetor de inteiros, em que cada inteiro indexa uma característica particular no conjunto global de características. Cada um desses vetores pode ter um comprimento que varia entre um e o dobro do número original de características. Nesse sentido, pode ser produzido subconjuntos de características que são menores ou maiores que o original. Além disso, pode haver um subconjunto de características maior que o original, mas que não tem todas as suas diferentes características. Outros aspectos dessa técnica são muito similares a maioria dos algoritmos genéticos [De Oliveira et al., 2010].

O algoritmo começa criando uma população inicial aleatória de indivíduos que são apenas vetores de inteiros, com comprimentos variados, que indexam características no conjunto original de características. Depois, novos indivíduos são produzidos pela aplicação de operações genéticas sobre a população. Então, os classificadores são treinados usando-se apenas as características que estão presentes em seu conjunto de treino, e os resultados são obtidos pela classificação em um conjunto de dados de teste ou empregando-se uma validação cruzada sobre o conjunto de dados.

É razoável incluir na função objetivo, explicitamente ou implicitamente, tanto a acurácia como a diversidade [Tsymbal et al., 2005]. Por isso, para cada classificador, nós obtemos essas duas medidas. A diversidade é a média da diferença entre a predição de cada classificador e a predição do comitê como um todo. A predição do comitê é obtida calculando-se a média da pontuação dada por todos os classificadores. Essas duas medidas compõem a fórmula, apresentada em [Opitz, 1999], para o fitness (aptidão) de cada indivíduo, dada por:

$$fitness = accuracy + \lambda \cdot diversity$$

Onde λ é uma constante experimental cujo propósito é regular a influência da diversidade. Os outros três valores que o algoritmo usa como métricas são o erro do comitê (EE - Ensemble

Error), a média do erro da população (APE - Average Population Error) e a diversidade média (AD - Average Diversity). EE é o erro médio comparando-se as previsões dadas pelo comitê e a classe real. APE é calculado obtendo-se a taxa de erro de cada classificador, e então calculando-se a média. AD é o resultado da média de todos os valores de diversidade que são usados para determinar o fitness de cada conjunto de características. Quando EE aumenta, o algoritmo analisa a causa. Se APE não aumenta e AD diminui, então aumentamos λ para dar mais importância à diversidade na próxima geração. Se AD não diminui e APE aumenta, diminuímos λ para dar mais importância à acurácia [De Oliveira et al., 2010].

Depois de o fitness de cada classificador ser determinado, nós selecionamos os classificadores que são mais aptos, fazendo com que a população volte ao tamanho original. Esse processo é repetido até que um critério de parada seja atingido.

Os resultados obtidos em [Opitz, 1999] mostram que o GEFS pode produzir boas populações iniciais, e o algoritmo que produz esses indivíduos é ao mesmo tempo simples e rápido. O fato de a população inicial ser competitiva com comitês formados tanto por Bagging quanto por AdaBoost é surpreendente. Isso mostra que em muitos casos mais diversidade é criada entre os classificadores pela variação do conjunto de características. Os resultados também mostram que GEFS obtém melhores performances ao longo do tempo que o algoritmo é rodado.

O método de formação de comitês de classificadores BSFS, usado em [Wandekoken et al., 2011], é baseado no conceito de seleção de características e consiste de um processo com três etapas que utiliza basicamente dois algoritmos gulosos, sendo o SFS (Sequential Forward Selection) e o SBS (Sequential Backward Selection). O algoritmo SBS é usado na primeira etapa e funciona da seguinte forma: inicia-se com um classificador que é treinado com todas as características. Então, a cada passo, uma característica é removida do conjunto, avaliando-se qual das características, quando removida do conjunto atual, faz com que o classificador tenha o melhor resultado. Dessa forma, sendo F o número total de características, obtemos um conjunto com F subconjuntos de características, sendo que o tamanho desses subconjuntos varia de $F - 1$, e cada subconjunto é resultado de um passo do algoritmo SBS. Podemos, então, formar um comitê, onde cada classificador é treinado com um dos subconjuntos descritos anteriormente. Na segunda etapa, cada um dos classificadores tem seus hiperparâmetros C e γ tunados com a técnica de grid-search. E, por fim, na terceira etapa é usado o algoritmo SFS, também conhecido como Forward Selection Strategy [Chowriappa et al., 2013], com o objetivo de se obter um conjunto reduzido e otimizado de classificadores. O algoritmo SFS funciona da seguinte maneira: inicia-se com um conjunto vazio de classificadores, sendo que, a cada passo, um classificador do comitê descrito anteriormente é adicionado, avaliando-se qual dos classificadores apresenta o melhor resultado quando adicionado ao comitê atual. O processo é repetido, até que um número predeterminado de classificadores seja atingido, obtendo-se o comitê final [Wandekoken et al., 2011].

No trabalho descrito em [Wandekoken et al., 2011], é proposto um método semelhante ao BSFS, que utiliza o algoritmo GEFS na seleção de características. Tal método foi denominado MultiGEFS.

3. Um método de formação de comitês em 3 estágios baseado em técnicas de soft computing

Nós propomos um método semelhante ao utilizado MultiGEFS para a construção de um comitê de classificadores que pode ser integrado a um sistema de diagnóstico de falhas em máquinas industriais. Porém, nos três estágios do processo, utilizamos técnicas de *soft-computing*, como descrito nas subseções seguintes, pois as mesmas apresentam alto desempenho com baixa utilização de recursos computacionais.

3.1. Seleção de características (GEFS)

Segue abaixo uma versão simplificada do algoritmo GEFS, descrito na subseção 2.1. A entrada do algoritmo apresenta os seguintes parâmetros:

- n é o tamanho da população inicial

- t é o número máximo de indivíduos produzidos
- G é o conjunto global de características da base de dados analisada

Algorithm 3.1: GEFS(n, t, G)

```

for  $i \leftarrow 1$  to  $n$ 
  do  $\left\{ \begin{array}{l} S(i) \leftarrow \text{RandomSelection}(G) \\ \text{Train}(S(i)) \end{array} \right.$ 
for  $i \leftarrow 1$  to  $n$ 
  do  $\text{Fitness}(S(i))$ 
 $\text{individuals} \leftarrow n$ 
repeat
   $\left\{ \begin{array}{l} \text{NEW} \leftarrow \text{Cross}(S) \\ \text{for each } c \in \text{NEW} \\ \text{do } \left\{ \begin{array}{l} \text{Mutation}(c) \\ \text{Train}(c) \end{array} \right. \\ \text{for each } c \in \text{NEW} \\ \text{do } \text{Fitness}(c) \\ S \leftarrow S + \text{NEW} \\ \text{individuals} \leftarrow \text{individuals} + \text{tam}(\text{NEW}) \\ \text{Prune}(S) \end{array} \right.$ 
until  $\text{individuals} \geq t$ 
return ( $S$ )

```

A variável S representa o conjunto atual de indivíduos do algoritmo, ou seja, sua população. Logo no início do algoritmo, cada indivíduo é formado através da seleção aleatória das características presentes em G , como descrito na subseção 2.1. Após todos os indivíduos serem treinados, eles têm sua aptidão (fitness) calculada de acordo com a fórmula apresentada na subseção 2.1. A variável individuals apenas armazena quantos indivíduos foram produzidos. O seguinte processo é repetido até que o número de indivíduos produzidos iguale ou supere o parâmetro t : a população NEW é formada cruzando-se os indivíduos de S segundo parâmetros que não são explicitados aqui; depois, os indivíduos de NEW sofrem mutação, são treinados e têm sua aptidão calculada; então, os indivíduos de NEW são integrados à população S , que é “podada” retornando ao seu tamanho original, apenas com seus indivíduos mais aptos.

3.2. Tunagem dos hiperparâmetros (PSO)

Os classificadores SVM apresentam parâmetros livres (hiperparâmetros e parâmetros de núcleo) que precisam ser definidos pelo usuário [Ren e Bai, 2010]. Para realizar esse ajuste, utilizamos o algoritmo PSO (Particle Swarm Optimization, ou Otimização por Enxame de Partículas, em português), que é um algoritmo estocástico, de otimização, baseado em uma população.

No início do algoritmo, é criada uma população em que cada indivíduo consiste de um vetor, no qual cada componente é um parâmetro a ser otimizado. A inicialização da população é feita aleatoriamente, dentro de um espaço de busca. O algoritmo segue, a cada iteração, atualizando a velocidade e a posição de cada partícula, sempre armazenando a melhor posição já obtida por cada partícula e por toda a população, até que um número máximo de iterações seja atingido.

As equações de atualização da velocidade e da posição de cada partícula são as seguintes:

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot r_1 \cdot (p_i - z_i^k) + c_2 \cdot r_2 \cdot (p_g - z_i^k)$$

$$z_i^{k+1} = z_i^k + v_i^{k+1}$$

onde v_i^k é a velocidade da partícula i na iteração k , w é coeficiente de inércia, c_1 e c_2 são coeficientes, r_1 e r_2 são números aleatórios no intervalo $(0, 1)$, p_i é a melhor posição já alcançada pela

partícula i , p_g é a melhor posição já alcançada por toda a população, e z_i^k é a posição da partícula i na iteração k .

O algoritmo PSO pode ser descrito pelo pseudocódigo abaixo:

Algorithm 3.2: PSO(*ProblemSize*, *PopulationSize*)

```

Population ← Initialize(PopulationSize)
repeat
  for  $i \leftarrow 1$  to PopulationSize
    do
       $v_i \leftarrow \text{updateVelocity}(v_i, z_i, p_i, p_g)$ 
       $z_i \leftarrow \text{updatePosition}(z_i, v_i)$ 
      if  $\text{ObjFunction}(z_i) > \text{ObjFunction}(p_i)$ 
        then
           $p_i \leftarrow z_i$ 
          if  $\text{ObjFunction}(p_i) > \text{ObjFunction}(p_g)$ 
            then  $p_g \leftarrow p_i$ 
until a maximum number of iterations be reached
return ( $p_g$ )
  
```

em que *ObjFunction* representa a função objetivo do problema a ser resolvido pelo algoritmo. No nosso caso, *ObjFunction*(z) representa a qualidade da classificação utilizando-se os hiperparâmetros indicados pela posição z e *ProblemSize* é o número de hiperparâmetros a ser ajustado.

3.3. Seleção dos classificadores (CLONALG e SA)

As áreas de aplicação de Sistemas Imunológicos Artificiais (AIS, em inglês) podem ser resumidas como segue [Dudek, 2012], [Hart e Timmis, 2008]:

- 1) aprendizado de máquina (clusterização, classificação, reconhecimento, aplicações em controle e robótica),
- 2) detecção de anomalias (detecção de falhas, aplicações em segurança de computadores e redes),
- 3) otimização (contínua e combinatorial).

Os tipos mais importantes de AIS são baseados nos conceitos de seleção negativa, seleção de clones, e redes imunológicas. No presente trabalho, utilizamos um algoritmo desenvolvido por de Castro e Von Zuben, denominado CLONALG que se baseia na seleção de clones e mostrou ser capaz de resolver tarefas complexas de aprendizado de máquina, como otimização multi-modal e reconhecimento de padrões [De Castro e Von Zuben, 2000].

O algoritmo CLONALG é baseado em uma população e é semelhante ao algoritmo GEFS em muitos aspectos. Nos experimentos que realizamos, cada indivíduo representa um subconjunto dos classificadores tunados no segundo estágio. A diferença em relação ao GEFS é que, a cada geração, novos indivíduos são formados por clonagem, e não por cruzamento, ou seja, são feitas cópias dos indivíduos mais aptos. Tais cópias (ou clones) sofrem mutação, de tal forma que, quanto maior for a aptidão de um clone, tanto menor será a alteração que ele irá sofrer durante a mutação. Outra diferença é que, a população do CLONALG aumenta ao longo do tempo com a inclusão dos clones mutados mais aptos. Assim que o número máximo de gerações é atingido, o algoritmo retorna o melhor indivíduo obtido. Entende-se por aptidão aqui, a qualidade da classificação do comitê formado pelo subconjunto de classificadores representado por um indivíduo. A métrica utilizada para se avaliar essa qualidade será descrita na seção 4.

Outra técnica de soft-computing que utilizamos para selecionar os classificadores que compõem o comitê final é o algoritmo SA (Simulated Annealing) [Kirkpatrick, 1984]. Esse algoritmo resolve problemas de otimização combinatorial por uma analogia com a mecânica estatística. O objetivo é minimizar a entropia total produzida durante o processo enquanto se encontra o estado de menor energia, em um dado tempo.

O algoritmo SA tem sido usado em um grande número de problemas, incluindo problemas combinatorias da classe NP-difícil. Em problemas de otimização combinatorial o objetivo é

desenvolver técnicas eficientes para se encontrar o mínimo (ou máximo) valor de uma função com muitos graus de liberdade e vários mínimos locais.

Os estados termodinâmicos correspondem a soluções no problema de otimização combinatorial. Energia em termodinâmica é a função objetivo em SA. O estado de menor energia, mudança de estado, e a temperatura em termodinâmica são traduzidas por solução ótima, uma solução vizinha, e o parâmetro de controle em SA, respectivamente. Portanto, o sistema abstrato pode ser descrito como se fosse sistema físico térmico para o qual o objetivo é localizar o estado de menor energia a medida que a temperatura diminui.

4. Experimentos e resultados

Como dito anteriormente, utilizamos em nossos experimentos dados obtidos a partir de sinais de sensores instalados em máquinas reais utilizadas em plataformas petrolíferas. Três bases de dados são utilizadas, uma para cada tipo de falha a ser diagnosticada. A base de desalinhamento possui 81 atributos numéricos e um atributo nominal indicando se o componente representado apresenta ou não desalinhamento. A base de desbalanceamento também possui 81 atributos numéricos e um atributo nominal indicando se houve tal falha. E, a base de rolamento, possui 28 atributos numéricos e um atributo nominal indicando se há algum defeito de rolamento.

Nos experimentos realizados com cada uma das bases descritas acima, um décimo do conjunto de dados foi separado do restante, para servir de base de teste, com o fim de se avaliar a qualidade do método proposto de forma mais realista. O restante, ou seja, os outros nove décimos do conjunto de dados, são utilizados para treinar os classificadores ao longo do processo, sendo denominado base de treino. Denominamos nosso método que combina as técnicas GEFS, PSO e CLONALG, nessa ordem, de Soft-MultiGEFS 1; e nosso método que combina as técnicas GEFS, PSO e SA, nessa ordem, de Soft-MultiGEFS 2. Utilizamos a métrica conhecida em inglês por Area Under ROC-Curve (AUC) em uma validação cruzada com 5 pastas para avaliar a qualidade da classificação.

Na base de treino, obtivemos os resultados apresentados na tabela abaixo:

Tabela 1: AUC estimado na base de treino por validação cruzada de 5 pastas

Tipo de falha	BSFS	MultiGEFS	Soft-MultiGEFS 1	Soft-MultiGEFS 2
desalinhamento	0,9061415886	0,9112106704	0,9102340592	0,9232422024
desbalanceamento	0,9462653559	0,9483262799	0,9400293158	0,9503982249
rolamento	0,9597985582	0,9666655557	0,9549862136	0,9668024161

Na base de teste, obtivemos os seguintes resultados:

Tabela 2: AUC estimado na base de teste por validação cruzada de 5 pastas

Tipo de falha	BSFS	MultiGEFS	Soft-MultiGEFS 1	Soft-MultiGEFS 2
desalinhamento	0,7897469176	0,7903958469	0,808904331	0,7720635951
desbalanceamento	0,912202381	0,9044642857	0,93125	0,9160714286
rolamento	0,9316125204	0,9425901202	0,935483871	0,942886812

5. Conclusões e trabalhos futuros

Através da observação da Tabela 1, concluímos que o método Soft-MultiGEFS 2 foi o que obteve o melhor resultado na base de treino de todos os tipos de falha. Porém, comparando-se a Tabela 1 com a Tabela 2, percebe-se que nem sempre o melhor desempenho na base de treino corresponde ao melhor desempenho na base de teste. Isso ocorre porque um comitê de classificadores pode ficar super-ajustado à base de treino, reduzindo sua acurácia na base de teste.

Na Tabela 2, podemos ver que o método Soft-MultiGEFS 1 apresentou um resultado consideravelmente melhor que todos os outros métodos na base de teste das falhas de desalinhamento e desbalanceamento. Na base de teste de desbalanceamento e de rolamento, o método Soft-MultiGEFS 2 se apresentou superior aos métodos BSFS e MultiGEFS.

Dessa forma, concluímos que nossa proposta de construir um comitê de classificadores utilizando técnicas de soft-computing se mostra promissora, visto que obtivemos bons resultados com os dois métodos propostos aqui. Futuramente, pretendemos investigar novas possibilidades de combinação de técnicas de soft-computing e dos parâmetros utilizados nas mesmas na resolução do mesmo problema de classificação.

Referências

- Chowriappa, P., Dua, S., Acharya, U. R., e Krishnan, M. M. R. (2013). Ensemble selection for feature-based classification of diabetic maculopathy images. *Computers in biology and medicine*, 43(12):2156–2162.
- De Castro, L. N. e Von Zuben, F. J. (2000). The clonal selection algorithm with engineering applications. In *Proceedings of GECCO*, volume 2000, p. 36–39.
- De Oliveira, M. V., Wandekokem, E. D., Mendel, E., Fabris, F., Varejão, F. M., Rauber, T. W., e Batista, R. J. (2010). Constructing feature-based ensemble classifiers for real-world machines fault diagnosis. In *IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society*, p. 1099–1104. IEEE.
- Dounias, G., Tselentis, G., e Moustakis, V. (2001). Machine learning based feature extraction for quality control in a production line. *Integrated Computer-Aided Engineering*, 8(4):325–336.
- Dudek, G. (2012). An artificial immune system for classification with local feature selection. *Evolutionary Computation, IEEE Transactions on*, 16(6):847–860.
- Hart, E. e Timmis, J. (2008). Application areas of ais: The past, the present and the future. *Applied soft computing*, 8(1):191–201.
- Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of statistical physics*, 34(5-6):975–986.
- Kuncheva, L. I. (2004). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.
- Opitz, D. W. (1999). Feature selection for ensembles. In *AAAI/IAAI*, p. 379–384.
- Ozcift, A. (2012). Svm feature selection based rotation forest ensemble classifiers to improve computer-aided diagnosis of parkinson disease. *Journal of medical systems*, 36(4):2141–2147.
- Ren, Y. e Bai, G. (2010). Determination of optimal svm parameters by using ga/pso. *Journal of Computers*, 5(8):1160–1168.
- Schapire, R. E. (1999). A brief introduction to boosting. In *Ijcai*, volume 99, p. 1401–1406.
- Tsymbal, A., Pechenizkiy, M., e Cunningham, P. (2005). Diversity in search strategies for ensemble feature selection. *Information fusion*, 6(1):83–98.
- Wandekokem, E., Mendel, E., Fabris, F., Valentim, M., Batista, R. J., Varejão, F. M., Rauber, T. W., et al. (2011). Diagnosing multiple faults in oil rig motor pumps using support vector machine classifier ensembles. *Integrated Computer-Aided Engineering*, 18(1):61–74.

Wandekoken, E. D., Varejao, F. M., Batista, R., e Rauber, T. W. (2011). Support vector machine ensemble based on feature and hyperparameter variation for real-world machine fault diagnosis. In *Soft Computing in Industrial Applications*, p. 271–282. Springer.

Widodo, A. e Yang, B.-S. (2007). Support vector machine in machine condition monitoring and fault diagnosis. *Mechanical Systems and Signal Processing*, 21(6):2560–2574.