Aplicação da mineração de dados à análise das condições de operação de transformadores

Luciana Gomes Castanheira (UEMG)
lugcastanheira@yahoo.com.br

RESUMO
O processo de descoberta de conhecimento em bases de dados (Knowledge Discovery in Databases – KDD) vem sendo amplamente utilizado como ferramenta para auxiliar a tomada de decisão. Neste trabalho, esse processo é estudado tendo como objetivo avaliar a utilização de métodos de mineração de dados aplicados em áreas da Engenharia Elétrica, sendo a sua abordagem sobre uma base de dados oriunda de testes de cromatografia de transformadores de potência. A mineração de dados é aplicada para obter a classificação de tipos de defeitos dos transformadores. As técnicas abordadas são redes neurais e árvores de decisão. As estruturas de algoritmos escolhidas nessas técnicas foram, respectivamente, a rede MLP com treinamento através do algoritmo de retropropagação resiliente e a árvore gerada pelo algoritmo J4.8, simulada no aplicativo weka. O melhor resultado foi com a utilização da árvore de decisão, em que foram conseguidos resultados com acerto entre 75 e 90%. Através dos resultados, viu-se que o processo de mineração de dados pode ser aplicado em problemas na área da Engenharia Elétrica. Entretanto, devem ser feitos estudos sobre o domínio de cada base de dados a ser tratada.
Palavras-chave: *Redes neurais, árvores de decisão, mineração de dados, tomada de decisão e transformadores.*

Application of data mining to the analysis of the conditions of operation of transformers

ABSTRACT
The Knowledge Discovery in Databases process (KDD), have been widely used as a tool to assist in decision-making. In this work this process is studied with the objective of estimate the data mining methods use is applied in areas of electrical engineering, and the approach made on the data bases from chromatography's tests of power transformers. The data mining is applied for a classification of the types of transformers's defects. The techniques broached are neural networks and decision trees. The algorithms chosen in these techniques were, respectively, MLP's network with resilient backpropagation algorithm training, and the tree generated by the J4.8 algorithm, simulated in weka. The best result was using the decision tree in wich results were reached with accuracy between 75 and 90%. With the results it is seen that the data mining can be applied to problems in the electrical engineering area, however studies should be made in each database area to be treated.
Keywords*: Neural networks, decision tree, data mining, decision making and transformer .*

1. Introdução

Durante os últimos anos, tem-se verificado crescimento substancial da quantidade de dados armazenados em meios magnéticos. Segundo Fayyad et al. (1997), esses dados, produzidos e armazenados em larga escala, são inviáveis de serem lidos ou analisados por especialistas através de métodos, como planilhas de dados e relatórios informativos operacionais, em que o especialista testa sua hipótese contra a base de dados. Ou seja, as informações contidas nos dados não estão caracterizadas explicitamente, uma vez que, sendo dados operacionais, não interessam quando estudados individualmente. Logo, não bastava armazená-los; era preciso transformá-los em informações.

Essas informações se tornaram essenciais para as empresas, já que as bases de dados deixaram de ser apenas repositórios de informações, passando a ser tratadas como patrimônio destas.

Segundo Cova e Cruz (2007), o dado é um elemento puro, quantificável sobre determinado evento. Já a informação é o dado analisado e contextualizado e envolve a interpretação de um conjunto de dados, ou seja, a informação é constituída por padrões, associações ou relações que todos aqueles dados acumulados podem proporcionar.

Diante das diversas aplicações da mineração de dados, para o desenvolvimento deste trabalho foi proposta uma forma de utilizá-la para auxiliar em áreas da Engenharia Elétrica. O objetivo é utilizar ferramentas de mineração de dados, eficientes para extração do conhecimento implícito, em auxílio à tomada de decisões em áreas da Engenharia Elétrica, mais especificamente para diagnóstico de falhas em transformadores de potência. Para isso, foram compreendidas, analisadas e comparadas as técnicas de redes neurais e árvores de decisão, aplicadas a problemas de mineração de dados oriundos de testes de cromatografia de transformadores de potência.

Segundo Costa e Brandão (2001), durante muitos anos os programas de manutenção preventiva em transformadores consistiram em inspeções, testes e ações periódicas em intervalos de tempo normalmente sugeridos pelo fabricante ou determinados através da experiência prática. Incluem-se nesses programas os testes de rotina e a execução de serviços como medição de perdas dielétricas; de resistência de isolação e dos enrolamentos; análise físico-química e cromatográfica do óleo; monitoramento manual ou automático da temperatura e do carregamento; tratamento, troca ou a regeneração do óleo isolante; limpeza dos terminais; e outros.

Com a demanda crescente por energia e sobrecarga dos sistemas de potência, a eficiência na distribuição da energia torna-se ponto crucial para as empresas do setor. Com os resultados de análises como as propostas neste trabalho em mãos, as empresas poderiam partir para uma manutenção preventiva, vistoriando os transformadores de forma mais tendenciosa, diminuindo, assim, o custo com manutenção corretiva, aumentando a confiabilidade dos sistemas e equipamentos elétricos, reduzindo o número de paradas programadas e eventuais e otimizando o fornecimento e uso das instalações elétricas.

A escolha do uso de mineração de dados para auxiliar a tomada de decisão, através da tarefa de classificação e do uso das técnicas que envolvem redes neurais e árvores de decisão, se deve a algumas vantagens que a mineração de dados proporciona, como o fato de serem de fácil compreensão e de as variáveis envolvidas poderem ser usadas na forma original, como aparecem nas bases de dados, não necessitando, pois, de normalização. O fato de serem de fácil compreensão possibilita às pessoas sem conhecimento estatístico interpretarem os modelos.

A aplicação do trabalho aos transformadores de potência se justifica pelo fato de este ser um dos maiores aparelhos em sistemas de potência, tornando-se vital para a operação dos sistemas. Logo, as técnicas para diagnóstico e detecção de suas falhas são valiosas. A análise de gás dissolvido no óleo do transformador é ferramenta poderosa. Neste trabalho foi utilizada essa análise, baseada na pesquisa do Duval (2002), em que é proposto um método para identificação da falha considerando-se os teores de formação dos gases etileno (C_2H_4), metano (CH_4), acetileno (C_2H_2), hidrogênio (H_2) e etano (C_2H_6).

A aplicação dos métodos para elaboração de classificadores de falhas baseadas em concentrações de gases no óleo dos transformadores foi escolhida pelo fato de o problema não possuir função matemática que descreva o comportamento da taxa de evolução das concentrações em função das falhas. Assim, é justificado o uso de dados históricos aplicados em métodos heurísticos como redes neurais e árvores de decisão.

O trabalho tem algumas limitações inerentes à situação. As mais claras são as atividades de pré-processamento que exigem a participação de especialistas do domínio de aplicação das bases de dados.

Essas atividades foram escolhidas, então, de forma a não precisar desse requisito, ou seja, foram realizados os pré-processamentos que não dependiam do domínio de aplicação das bases de dados. Além disso, o processo de KDD apresenta melhor resultado quando submetido a análises de grandes bases de dados. No caso do trabalho proposto, as bases de dados não são muito extensas, devido à dificuldade de obtenção de dados de cromatografia confiáveis.

A tomada de decisão realizada com o auxílio da mineração de dados vem sendo usada para diversas aplicações. São encontrados na bibliografia trabalhos dos mais variados assuntos, por exemplo: auxílio em diagnósticos médicos, analisando-se o histórico dos pacientes; avaliação de riscos de inadimplência em empresas de grande porte; ajuste de variáveis em processos de siderurgia; e precificação de opções no mercado de ações.

2. Métodos utilizados

2.1. Processo de descoberta de conhecimento

O processo capaz de descobrir conhecimento em bancos de dados é chamado de *Knowledge Discovery Database* – KDD. Segundo Fayyad et al. (1997), esse processo foi proposto em 1989 para se referir às etapas que produzem conhecimentos a partir dos dados. Dentro desse processo, a etapa de mineração de dados é a fase que transforma dados em informação. Seu objetivo principal é extrair conhecimento a partir de grandes bases de dados. Para isso, ele envolve diversos conceitos, como: estatística, matemática, inteligência artificial e reconhecimento de padrões, além de bancos de dados e técnicas de visualização dos dados.

Para iniciar um processo de KDD, é preciso ter o entendimento do domínio da aplicação e dos objetivos finais a serem atingidos.

Segundo Fayyad et al. (1997), o processo de KDD é composto basicamente por cinco etapas, relacionadas na Figura 1.

A primeira etapa é um agrupamento de forma organizada dos dados (seleção). A etapa da limpeza dos dados vem a seguir, através de um pré-processamento dos dados, visando adequá-los aos algoritmos que serão utilizados. Para facilitar o uso das técnicas de mineração de dados, os dados ainda podem passar por uma transformação que os armazena adequadamente em arquivos para serem lidos pelos algoritmos. É a partir desse momento que se chega à fase de mineração de dados especificamente, que começa com a escolha das ferramentas (algoritmos) a serem utilizadas. Essa escolha depende fundamentalmente do objetivo do processo de KDD: classificação, agrupamento, regras associativas ou desvio. De acordo com o algoritmo utilizado será gerado um arquivo de descobertas (que pode ser um relatório ou um gráfico, por exemplo). Esse arquivo deve ser interpretado, gerando-se as conclusões que fornecem o conhecimento da base de dados estudada.

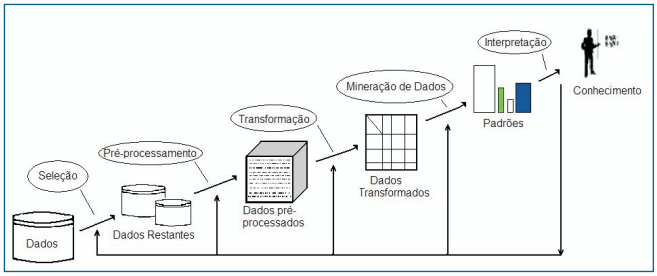

Figura 1- Fases do KDD.
Fonte: FAYYAD et al., 1997.

A mineração de dados é a etapa mais importante do processo de KDD. Segundo Possa et al. (1998), o cérebro humano, comprovadamente, consegue fazer até oito comparações ao mesmo tempo. A função da mineração de dados é justamente ampliar essa comparação para "infinito" e tornar isso visível ao olho humano.

Pode-se concluir que a mineração de dados se caracteriza pela existência de um algoritmo que, diante da tarefa proposta, será eficiente em extrair conhecimento implícito e útil de um banco de dados. Pode-se dizer que mineração de dados é a fase que transforma dados puros em informações úteis.

A tarefa que será aplicada no estudo é a de classificação, dentro da etapa de mineração de dados. A classificação pode ser considerada como uma função de aprendizado que mapeia dados de entrada, ou conjuntos de dados de entrada, em um número finito de classes. Nela, cada exemplo pertence a uma classe, entre um conjunto pré-definido de classes. O objetivo de um algoritmo de classificação é encontrar alguma correlação entre os atributos e uma classe, de modo que o processo de classificação possa usá-la para predizer a classe de um exemplo novo e desconhecido. Para efetuar essa classificação serão utilizadas duas técnicas: redes neurais e árvores de decisão.

2.2. Redes neurais

Rede neural artificial (RNA) é uma técnica que constrói um modelo matemático, de um sistema neural biológico simplificado, com capacidade de aprendizado, generalização, associação e abstração. Assim como no cérebro humano, as redes neurais apresentam estrutura altamente paralelizada, composta por processadores simples (neurônios artificiais) conectados entre si.

De acordo com Haykin (2001), uma propriedade importante das redes neurais é a sua habilidade para aprender a partir do ambiente na qual estão inseridas, ou ambiente de aprendizado, e melhorar seu desempenho através da aprendizagem. As RNAs tentam aprender por experiência, ou seja, diretamente dos dados, através de um processo de repetidas apresentações dos dados à rede.

Uma rede neural artificial é composta por várias unidades de processamento, que geralmente são conectadas por canais de comunicação que estão associados a determinados pesos. Os pesos nada mais são do que um modelo para simular os dendritos. São os pesos que, alterando os seus valores representativos durante os estímulos, influenciam o resultado do sinal de saída, segundo Tafner (1998).

As entradas, simulando uma área de captação de estímulos, podem ser conectadas em muitos neurônios, resultando em uma série de saídas, em que cada neurônio representa uma saída. Essas conexões, em comparação com o sistema biológico, representam o contato dos dendritos com outros neurônios, formando, assim, as sinapses. A função da conexão em si é tornar o sinal de saída de um neurônio em um sinal de entrada de outro ou, ainda, orientar o sinal de saída para o mundo externo (mundo real). Ainda segundo Tafner (1998), as diferentes possibilidades de conexões entre as camadas de neurônios podem ter, em geral, *n* números de estruturas diferentes. Usualmente, trabalha-se com três camadas, que são classificadas em:

- Camada de entrada: onde os padrões são apresentados à rede.
- Camadas intermediárias ou ocultas: onde é feita a maior parte do processamento, através das conexões ponderadas. Estas podem ser consideradas como extratoras de características.
- Camada de saída: onde o resultado final é concluído e apresentado.

O primeiro trabalho a ter ligação direta com o aprendizado de redes artificiais foi apresentado por Donald Hebb, em 1949. Hebb mostrou como a plasticidade da aprendizagem de redes neurais é conseguida através da variação dos pesos de entrada dos neurônios. Ele propôs uma teoria para explicar o aprendizado em neurônios biológicos baseada no reforço das ligações sinápticas entre neurônios excitados. Mais tarde, Widrow e Hoff (1960) sugeriram uma regra de aprendizado, conhecida como regra de delta. Esta, por sua vez, é baseada no método do gradiente descendente para minimização do erro na saída de um neurônio com resposta linear.

O método do gradiente é uma técnica numérica para a minimização de funções como uma função $f(x(n))$ contínua, em dada iteração n, através de suas derivadas. A direção de pesquisa em busca do mínimo da função será a direção negativa do gradiente. Ou seja:

$$x(n + 1) = x(n) - ç \nabla f(x(n)) \quad (4.5)$$

em que ç é uma constante que determina a amplitude do passo na direção de descida da função, e ∇ é o operador matemático que representa o gradiente de uma função escalar multivariável. A convergência será acelerada se for utilizado um valor de ç grande, porém isso dificultará o encontro do mínimo apropriado. No entanto, ocorre lentidão considerável na convergência quando o valor de ç for muito pequeno. O ideal é que para cada iteração se conheça o ç ótimo.

Em 1958, Rosenblatt (1958) demonstrou com o *perceptron* que, se fossem acrescidas de sinapses ajustáveis, as redes com neurônios MCP poderiam ser treinadas para classificar certos tipos de padrões. Rosenblatt descreveu uma topologia de rede com estruturas de ligação entre os neurônios e propôs um algoritmo para treinar a rede para executar determinados tipos de funções.

Em 1986, Rumelhart et al. publicaram um trabalho em que foi desenvolvido o algoritmo de retropropagação para treinamento de redes MLP *(multi layer perceptron*), que são redes *perceptron* multicamadas.

2.3. Redes perceptron multicamadas

As redes *perceptron* multicamadas têm como unidade básica o *perceptron* descrito por Mcculloch e Pitts (1943). Segundo Passos (2006), essas unidades são distribuídas em camadas onde cada uma está conectada a todas as unidades da camada anterior. Nesse modelo, é calculado o produto interno das entradas aplicadas, x_i,

com os pesos, $w_{ji,}$ e também é incorporada uma polarização, x_0, aplicada externamente. Ainda de acordo com Passos (2006), a soma resultante, considerada como nível de atividade interna ou potencial de ativação, é aplicada, então, a uma função de ativação, ö(.), que pode ser a saída final da rede, ou a entrada de outros *perceptrons* da camada seguinte. A Figura 2 apresenta a configuração do *perceptron*.

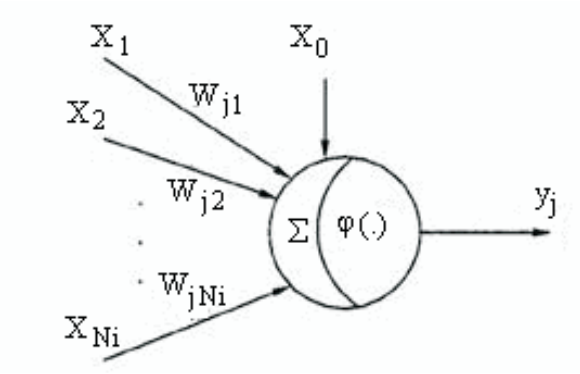

Figura 2 - Modelo do p*erceptron* utilizado nas redes MLPs.

Fonte: PASSOS, 2006.

O princípio do algoritmo de retropropagação é, utilizando-se o método do gradiente descendente, minimizar o erro das camadas intermediárias por meio de uma estimativa do efeito que estas causam ao erro da camada de saída. Assim, o erro de saída da rede é calculado e retroalimentado para as camadas intermediárias, possibilitando o ajuste dos pesos proporcionalmente aos valores das conexões entre camadas. A utilização do gradiente descendente requer o uso de função de ativação contínua e diferenciável.

Esse algoritmo, contudo, apresenta convergência lenta, causada pelo tamanho das derivadas parciais nos pesos. Jacobs (1987) identificou causas fundamentais para esse fato. Segundo ele, quando a superfície de erro (*E*) apresentar variação pequena em relação a dado peso, sua derivada terá magnitude pequena e, consequentemente, o ajuste será pequeno, requerendo muitas iterações para a convergência. Se a variação for elevada, o gradiente e o ajuste também serão elevados acarretando uma passagem pelo mínimo da superfície de erro.

Logo, para uma boa convergência no modelo de retropropagação, deve-se ter uma boa escolha da taxa de aprendizado ç. Uma técnica aplicada para essa escolha é o uso do algoritmo de retropropagação resiliente (RPROP), utilizado neste trabalho, e que foi proposto por Riedmiller e Braun (1993).

A ideia básica do algoritmo RPROP é eliminar a influência do valor das derivadas parciais na atualização dos pesos. Como consequência, só é considerada a indicação do sinal da derivada parcial. A atualização dos pesos é determinada, de acordo com Riedmiller e Braun (1993), exclusivamente por um valor de atualização $Ä_{ji}(n)$, conforme (1).

$$\Delta w_{ji}(n) = \begin{cases} -\Delta_{ji}^{(n)}, & se \quad \dfrac{\partial E^{(n)}}{\partial w_{ji}} > 0 \\ +\Delta_{ji}^{(n)}, & se \quad \dfrac{\partial E^{(n)}}{\partial w_{ji}} < 0 \\ 0, & demais \quad casos \end{cases} \tag{1}$$

em que $Ä_{ji}(n)$ é aumentado ou diminuído segundo o procedimento dado em (2).

$$\Delta ji(n) = \begin{cases} \eta^{+}\Delta_{ji}^{(n-1)}, & se \quad \dfrac{\partial E^{(n)}}{\partial w_{ji}} \dfrac{\partial E^{(n-1)}}{\partial w_{ji}} > 0 \\ \eta^{-}\Delta_{ji}^{(n-1)}, & se \quad \dfrac{\partial E^{(n)}}{\partial w_{ji}} \dfrac{\partial E^{(n-1)}}{\partial w_{ji}} < 0 \\ \Delta_{ji}^{(n-1)}, & demais \quad casos \end{cases} \tag{2}$$

em (1) e (2), $E^{(n)}$ é a função erro quadrática, $ç^{+} = 1{,}2$ e $ç^{-} = 0{,}5$ são constantes escolhidas empiricamente.

Segundo Riedmiller e Braun (1993), a regra de adaptação dos pesos trabalha do seguinte modo: cada vez que a derivada parcial do erro correspondente muda de sinal, ela indica que a última atualização foi muito grande (o algoritmo saltou o mínimo local). Assim, o valor de adaptação é diminuído pelo fator $ç^{-}$. Se o sinal da derivada permanece o mesmo, isso indica que o valor de adaptação deve ser aumentado, acelerando a convergência mesmo em regiões suaves da superfície de erro.

Uma vez que os valores de atualização para cada peso são adaptados, a atualização dos pesos segue uma regra muito simples:

• Se a derivada trocar de sinal (erro de incremento), o peso é diminuído.
• Se a derivada mantiver o sinal, o peso é aumentado.

Um problema que a rede neural pode apresentar é denominado *overfitting*. Nesse caso ocorre generalização pobre da rede, ou seja, ela aprende os dados de treinamento (apresentando erro pequeno no treinamento), mas apresenta erro elevado quando apresentados os dados de validação.

A generalização da rede pode ser melhorada quando a base de dados utilizada for grande o suficiente para garantir ajuste adequado. Quanto mais dados forem apresentados à rede, mais complexas são as funções que a ela pode criar. Logo, encontrar o número de parâmetros ideal para a rede é um dos objetivos do treinamento, mas estimar esse número normalmente não é tarefa fácil e requer conhecimento sobre a complexidade do problema, que geralmente não se tem, pois muitas vezes é esse conhecimento que se deseja obter por meio do processo de modelagem (BRAGA et al., 2003).

Para evitar o *overfitting*, tornando a rede capaz de generalizar, podem-se usar os métodos de *early stopping* (parada antecipada) ou da regularização.

A técnica de parada antecipada pára o treinamento quando as diferenças entre erro de treinamento e erro de validação começam a crescer. Ela consiste em treinar a rede neural com determinada amostra (que no caso seria o conjunto de treinamento) e em validar seu desempenho periodicamente, empregando outra amostra (conjunto de validação). Se os dados obtidos com a validação atingirem nível satisfatório, o treinamento é interrompido, independentemente do número de iterações realizado. Haveria a necessidade da criação de um terceiro grupo (denominado conjunto de teste), em que a técnica seria aplicada para confirmar a eficiência.

Já a regularização (também conhecida como redução de pesos) tem o objetivo de limitar a complexidade da rede. A regularização envolve a modificação da função-objetivo, que é normalmente escolhida para ser a média dos erros quadrados da rede no conjunto de treinamento. Uma regularização muito utilizada é a regularização *bayesiana*.

Pode-se dizer que, na fase de treinamento, o erro da rede na n-ésima iteração (i.e., na apresentação do n-ésimo exemplo de treinamento) é calculado tomando-se a diferença entre o valor desejado $d_k(n)$ (*i.e., valor de saída conhecido para o k-ésimo neurônio*) e o valor de saída da rede $z_k(n)$ *(i.e., valor de saída da rede para o k-ésimo neurônio)*, conforme a equação 3.

$$e_k(n) = d_k(n) - z_k(n) \tag{3}$$

O valor instantâneo da energia do erro para a k-ésima saída é definido como $e_k^2(n)/2$. Para avaliar a energia instantânea total do erro, somam-se as contribuições de todas as saídas, conforme mostrado na equação a seguir:

$$E(n) = \frac{1}{2}\sum_{k=1}^{Ns} e_k^2(n) \tag{4}$$

A média dos erros quadrados de todo o conjunto de treinamento Z é utilizada para uma análise geral do treinamento. Ela é avaliada conforme a equação 5.

$$E_{med} = \frac{1}{Z}\sum_{n=1}^{Z} E(n) \tag{5}$$

O treinamento é todo realizado com o objetivo de ajustar os pesos da rede, tal que a média dos erros quadrados seja minimizada.

De acordo com Demuth e Beale (2002), no algoritmo de regularização bayesiana a função-objetivo assume a forma descrita na equação 6.

Assumindo F como a função-objetivo:

$$F = á.SSE + â.SSW \tag{6}$$

em que:
SSE = somatório dos erros quadrados;
SSW = somatórios do quadrado dos pesos e bias; e
á e â = parâmetros da função objetivo.

Segundo Hagan e Foresse (1997), os parâmetros de regularização são obtidos com a estrutura de *Bayesian*, que estima esses parâmetros usando-se técnicas estatísticas. Para aplicar a regularização, o algoritmo de treinamento utilizado deve ser o Levemberg-Marquardt, já que a técnica requer o cálculo da matriz de *Hessian*.

O algoritmo de Levemberg-Marquardt tem a característica de fornecer estimativa de quantos parâmetros da rede (pesos e bias) estiverem efetivamente sendo usados por ela. Esse número efetivo de parâmetros permanece aproximadamente constante, não importando quão grande é o número total de parâmetros da rede. Para aplicação desse algoritmo, deve-se tomar o cuidado de ter uma rede com dimensões suficientes para representar adequadamente a função real.

2.4. Árvore de decisão

As árvores de decisão são representações simples do conhecimento e um meio eficiente de construir classificadores que predizem classes baseadas nos valores de atributos de um conjunto de dados.

Uma árvore de decisão tem a função de particionar recursivamente um conjunto de treinamento até que cada subconjunto obtido contenha casos de uma única classe. Elas são construídas baseadas no modelo *Top-Down*, ou seja, utilizam a técnica de dividir para conquistar, baseando-se na sucessiva divisão do problema em vários subproblemas de menores dimensões até que uma solução para cada um dos problemas mais simples seja encontrada. Para atingir essa meta, o algoritmo escolhido para a árvore de decisão examina e compara a distribuição de classes durante a construção da árvore. Segundo Quinlan (1993), os resultados obtidos após a construção de uma árvore de decisão são dados organizados de maneira compacta, com a árvore podendo ser utilizada para classificar novos casos.

Uma questão-chave para a construção de uma árvore de decisão consiste na estratégia para a escolha dos atributos que estarão mais próximos da raiz da árvore (ou seja, os atributos que são inicialmente avaliados para determinar a classe a qual uma observação pertence).

O algoritmo J4.8, escolhido para geração da árvore de classificação com a técnica de árvores de decisão, utiliza os conceitos de entropia e ganho de informação para a implementação de sua árvore.

O conceito de entropia é uma medida de informação calculada pelas probabilidades de ocorrência de eventos individuais ou combinados. Pode-se dizer que a entropia é dada como medida da impureza em um conjunto arbitrário de amostras de treinamento. Pode ser considerada a medida da quantidade de desordem de um conjunto de amostras.

Dado um atributo classe A, de um conjunto de amostras S, em que A pode assumir v_i valores de classes diferentes, então a entropia de A relativa a essa classificação é definida na equação 7.

$$Entropia(A) = -\sum_{i=1}^{m} p_i \log_2 p_i \qquad (7)$$

em que m é o número total de classes e $p_i = p(A = v_i)$ é a probabilidade de o atributo classe A ser igual à classe cujo índice é *i* (i.e., é a proporção do número de amostras com valor v_i em relação ao número total de amostras de S).

Já o ganho de informação é definido como uma soma das entropias individuais menos a entropia conjunta, sendo uma medida de correlação entre duas variáveis. É uma propriedade estatística que mede como determinado atributo separa as amostras de treinamento de acordo com sua classificação. Ele mede a eficácia de um atributo em classificar os dados de treinamento.

Um dos objetivos da construção de árvores de decisão é diminuir o valor da entropia. A medida do ganho de informação representa a redução esperada na entropia de um atributo preditivo, considerando que um atributo classe já tenha sido determinado. Ou seja, o valor do ganho de informação fornece redução esperada na entropia causada pela partição das amostras de acordo com esse atributo-classe conhecido previamente. No processo de construção da árvore de decisão, o atributo preditivo que possuir o maior ganho de informação deve ser colocado como raiz da árvore, pois é esse atributo que fornecerá a maior redução na entropia, classificando os dados de forma mais rápida.

Para conhecer o valor do ganho de informação, devem ser feitos dois cálculos:

- A entropia conjunta, ou seja, para todo o conjunto de dados – nesse caso, levando-se em consideração os subconjuntos referentes às classificações existentes.
- A entropia individual de cada atributo do conjunto de dados.

Considere um conjunto de amostras, contendo um atributo-classe definido como A e um dos atributos preditivos definido como B. O ganho de informação (GI) do atributo preditivo B é definido como a diferença entre a entropia do atributo classe A (*Entropia (A)*) menos a entropia condicional do atributo preditivo B, tendo sido definido o valor do atributo classe A (*Entropia (B|A)*). Matematicamente, o ganho de informação é dado pela equação 8.

$$GI(B, A) = Entropia(A) - Entropia(B \mid A) \qquad (8)$$

A entropia condicional, definida como a entropia de um atributo preditivo B, sendo conhecido o atributo classe A, é dada por (9):

$$Entropia(B \mid A) = \sum_{i=1}^{m} p_i . Entropia(B \mid A = v_i) \qquad (9)$$

em que *m* é o número total de classes do conjunto de amostras, B é o atributo preditivo que está sendo considerado. A é o atributo-classe assumindo o valor v_i.

Além disso, p_i é como definido antes, i.e., $p_i = p(A = v_i)$, é a proporção dada pela razão entre o número de amostras com valor v_i e o número total de amostras de S.

O termo *Entropia* $(B \mid A = v_i)$ é a entropia do atributo preditivo B, sendo dado o valor do atributo classe $A = v_i$, como definido na equação 10.

$$Entropia(B \mid A = v_i) = -\sum_{i=1}^{m} p(B \mid A = v_i) \log_2 p(B \mid A = v_i) \quad (10)$$

em que *m* é o número de classes que o atributo classe A pode assumir, $p(B \mid A = v_i)$ é a probabilidade condicional do atributo B, i.e., é a proporção dada pela razão entre o número de exemplos de B com $A = v_i$ e o número total de amostras na classe $A = v_i$.

O algoritmo J4.8 utiliza a razão do ganho para escolha do atributo que será o nó-raiz. O atributo que apresentar o maior valor dessa razão será escolhido como nó-raiz, já que é esse atributo que faz a classificação dos outros atributos de forma mais direta. A partir daí o algoritmo repete os mesmos cálculos, mas agora apenas com os filhos desse nó-pai. Esses passos são realizados de forma recursiva até que não existam mais possibilidades ou exista um dos nós que apresente clara maioria. A razão do ganho é a razão entre o ganho de informação (GI) e a informação dividida. Os cálculos desses valores são realizados de acordo com as equações 11, 8 e 12.

$$Razão\_Ganho = \frac{GI}{Informação\_Dividida} \quad (11)$$

$$Informação\,Dividida = -\sum_{j=1}^{n} p_j \cdot \log_2 p_j \quad (12)$$

em que *m* é o número de classes que o atributo classe A pode assumir, $p_i = p(A = v_i)$ é a probabilidade de o atributo classe A ser igual à classe cujo índice é i, é a probabilidade condicional do atributo B, i.e., é a proporção dada pela razão entre o número de exemplos de B com $A = v_i$ e o número total de amostras na classe $A = v_i$.

2.5. Descrição das bases de dados

A classificação dos dados utilizados neste trabalho foi feita baseada em um método proposto por Duval (2002). Ele propôs o método para identificação da falha baseado nos cinco gases citados, criando o chamado triangulo de Duval. O método proposto leva em consideração apenas a concentração percentual relativa dos gases acetileno, etano e metano. Em um triângulo, como na Figura 3, é representada a evolução de gases gerados para algumas falhas. É feita uma relação percentual de cada gás em relação ao total dos gases gerados para definir as coordenadas. Dessa forma, podem ser identificadas três falhas de origem elétrica e três falhas de origem térmica, utilizando-se os códigos apresentados na Figura 3, cuja legenda vem a seguir.

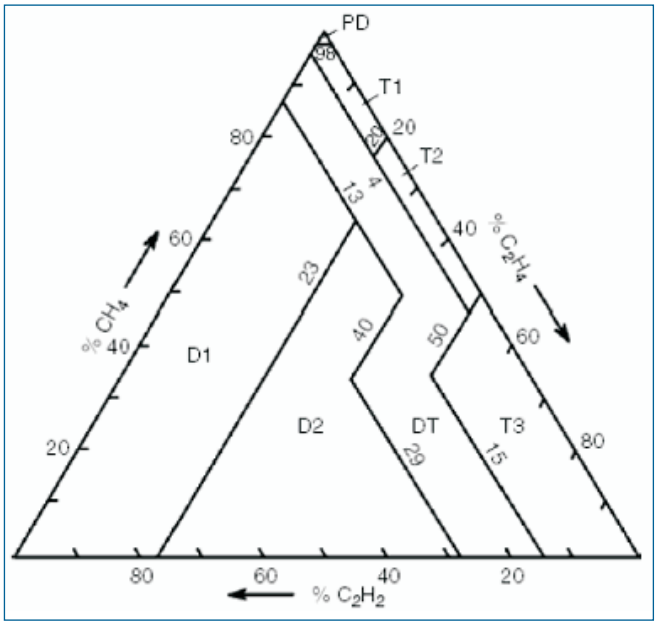

Figura 3 - Triângulo de Duval.
Fonte: DUVAL, 2002.

em que:

PD = descargas parciais;
T1 = falha térmica com temperatura T < 300 ºC;
T2 = falha térmica com temperatura: 300 ºC < T < 700 ºC;
T3 = falha térmica com temperatura T > 700 ºC;
D1 = descargas de baixa energia;
D2 = descargas de alta energia; e
DT = mistura de falhas.

Para aplicação dos algoritmos estudados, foram utilizadas três bases de dados contendo os diagnósticos de defeitos e as concentrações de gases diluídos no óleo isolante dos transformadores. As bases de dados são compostas, então, por concentrações de cinco dos gases mais importantes encontrados no óleo dos transformadores de potência e utilizados no triângulo de Duval. São eles: hidrogênio (H_2), metano (CH_4), etileno

(C_2H_4), etano (C_2H_6) e acetileno (C_2H_2). A produção de alguns desses gases se dá por:
- Hidrogênio: grandes quantidades associadas com condições de descarga parcial.
- Hidrogênio, etano, metano e etileno: resultados da decomposição térmica do óleo, ou seja, contato do óleo isolante com partes quentes.

- Acetileno: associado com arco elétrico no óleo.

A primeira base de dados descrita foi denominada IEC. Ela contempla parte da base de dados da norma IEC TC 10 (DUVAL; PABLO, 2001). É uma base composta por 53 amostras com diagnósticos determinados através de medições específicas e inspeções visuais feitas por especialistas, com 16 amostras com diagnóstico de normalidade, 22 apresentando falha elétrica e 14, falha térmica.

A segunda base de dados foi denominada Base 1 e contempla dados fornecidos pelo centro de pesquisas do setor elétrico (CEPEL), composta por amostras com diagnósticos determinados através de medições específicas e inspeções visuais, feitas por especialistas, considerando-se transformadores de vários níveis de tensão. A base de dados totaliza 224 amostras, divididas em 83 amostras com diagnóstico de normalidade, 61 com falha elétrica e 80 com falha térmica.

Por fim, a terceira base de dados descrita foi denominada Base 2. Essa base contempla dados históricos de transformadores (MORAIS, 2004), composta por amostras com diagnósticos determinados através de medições específicas e feitas por especialistas, num total de 212 amostras, divididas em 180 delas com diagnóstico de normalidade, 10 com falha elétrica e 22 com falha térmica.

A partir dessas três bases de dados, foram constituídos dois grupos para simulações:

• Na primeira análise, os dados foram treinados com 70% dos dados da base IEC e validados com os 30% dos dados restantes. Isso tanto para a rede neural quanto para a árvore de decisão. A rede neural e a árvore de decisão geradas foram utilizadas para classificar as bases de dados Base 1 e Base 2. Realizou-se o mesmo procedimento considerando as bases de dados de geração e utilização das técnicas balanceadas.

• Na segunda análise, os dados das bases IEC e Base 1 foram agrupados, sendo o treinamento feito com 70% deles e a validação, com os 30% dos dados restantes (também das duas bases); isso tanto para rede neural quanto para árvore de decisão. A rede neural e a árvore de decisão geradas foram utilizadas para classificar os dados da Base 2. Realizou-se o mesmo procedimento considerando as bases de dados de geração e utilização das técnicas balanceadas. Dessa forma está sendo aplicado o método de avaliação cruzada, realizando-se a construção do classificador com uma base de dados e usando-o em outra base de dados.

Os diagnósticos de normalidade, falha elétrica e falha térmica foram agrupados em três falhas e codificados da seguinte forma:

Classificação A: transformador com diagnóstico de normalidade.

Classificação B: transformador com diagnóstico de falha elétrica (que agrupou as falhas com legenda PD, D1 e D2 na Figura X).

Classificação C: transformador com diagnóstico de falha térmica (que agrupou as falhas com legenda T1, T2 e T3 na Figura X).

Esses dados foram simulados com a rede neural, utilizando-se o software MatLab®, e a árvore de decisão o foi com o uso do algoritmo J4.8, do software weka®.

Nas redes neurais, as simulações foram realizadas com variações da quantidade de neurônios e da função de ativação, que têm o papel de mapear a camada de saída de acordo com as entradas da rede. Foram realizadas simulações com as três funções de ativação mais utilizadas, sendo elas a tangente hiperbólica, a sigmoide e a linear. A função de ativação linear foi utilizada apenas para a saída.

Para cada uma dessas funções de ativação as bases de dados foram submetidas aos diferentes parâmetros:

• Quantidade de iterações (ou ciclos): em cada conjunto de teste, o conjunto utilizado para treinamento da rede foi submetido às seguintes quantidades de iterações: 1.000, 4.000 e 8.000.

• Quantidade de neurônios intermediários (ou escondidos) da rede: a rede foi treinada variando-se o número dos neurônios da camada escondida. Foram realizados testes com 4, 6, 8 e 10 neurônios.

Na técnica de árvore de decisão, as simulações foram feitas variando-se os parâmetros de poda ou não da árvore e do fator de confiança (CF). O fator de confiança é uma forma simples de avaliar a precisão das regras obtidas nos dados de treinamento. Esse fator é calculado pela razão X/Y, em que X é o número de registros que satisfazem o antecedente e o consequente da regra e Y é o número total de registros que satisfazem o antecedente da regra.

2.6. Simulações e resultados

As bases de dados foram aplicadas da forma como descrita anteriormente (as três bases de dados agrupadas em dois conjuntos – primeira e segunda análises). As simulações foram feitas com os conjuntos da forma original e, posteriormente, com os conjuntos balanceados. Para o balanceamento, foi utilizada a técnica de replicação dos dados em menor quantidade. Também foram realizadas simulações das bases de dados, considerando-se a concentração de cada tipo de gás dividida pelo TGC. Para construção dessa nova base de dados, cada concentração de determinado gás foi dividida pela soma de todas as concentrações para esse mesmo gás. Nesse caso, as simulações foram realizadas apenas para as bases de dados balanceadas, já que os resultados delas são melhores que quando utilizadas as bases desbalanceadas.

Os resultados dessas simulações estão apresentados nas tabelas seguintes, em que:

- Na Tabela 1 estão representados os índices de concordância percentual dos dados das bases desbalanceadas, separados por diagnósticos (normalidade, defeito elétrico e defeito térmico).
- Na Tabela 2 estão representados os mesmos dados, mas das bases balanceadas.
- Na Tabela 3 estão representados os índices de concordância percentual dos dados utilizando o TGC, com a base balanceada.

Tabela 1- Índice de concordância percentual discriminado por tipo de defeito para as bases desbalanceadas

	Rede neural				Árvore de decisão			
Primeira análise diagnóstico	**Índice de concordância (%)**				**Índice de concordância (%)**			
	Geração da Rede (IEC)		Diagnóstico		Montagem da Árvore (IEC)		Diagnóstico	
	Trein.	Valid.	Base 1	Base 2	Trein.	Valid.	Base 1	Base 2
Normal	98,8	98,6	42,8	51,1	100,0	100,0	59,1	46,9
Def Elétrico	90,5	46,8	49,9	71,5	93,3	42,9	67,2	80,0
Def Térmico	99,0	98,0	65,6	61,1	100,0	100,0	77,5	72,7
Segunda Análise Diagnóstico	Geração da Rede (IEC + Base 1)		Diagnóstico		Montagem da Árvore (IEC + Base 1)		Diagnóstico	
	Trein.	Valid.	Base 2		Trein.	Valid.	Base 2	
Normal	93,5	86,2	78,1		91,2	90,3	83,3	
Def Elétrico	96,2	87,3	48,3		93,1	92,0	53,9	
Def Térmico	84,3	41,1	3,8		75,8	64,3	11,0	

Tabela 2 - Índice de concordância percentual discriminado por tipo de defeito para a base balanceada

	Rede neural				Árvore de decisão			
Primeira análise diagnóstico	**Índice de concordância (%)**				**Índice de concordância (%)**			
	Geração da Rede (IEC)		Diagnóstico		Montagem da Árvore (IEC)		Diagnóstico	
	Trein.	Valid.	Base 1	Base 2	Trein.	Valid.	Base 1	Base 2
Normal	100,0	100,0	45,8	68,2	100,0	100,0	65,9	88,8
Def Elétrico	86,7	57,2	63,9	52,6	93,4	57,2	86,1	60,7
Def Térmico	100,0	85,8	69,9	60,9	100,0	100,0	77,0	63,3
Segunda análise diagnóstico	Geração da Rede (IEC + Base 1)		Diagnóstico		Montagem da Árvore (IEC + Base 1)		Diagnóstico	
	Trein.	Valid.	Base 2		Trein.	Valid.	Base 2	
Normal	91,2	68,0	89,0		92,7	90,3	92,2	
Def Elétrico	97,1	83,6	62,6		97,1	93,6	79,9	
Def Térmico	96,6	90,3	83,3		96,9	96,8	82,1	

Tabela 3 - Índice de concordância percentual discriminado por tipo de defeito para a base balanceada, considerando o TGC

	Rede neural				Árvore de decisão			
Primeira análise diagnóstico	**Índice de concordância (%)**				**Índice de concordância (%)**			
	Geração da Rede (IEC)		Diagnóstico		Montagem da Árvore (IEC)		Diagnóstico	
	Trein.	Valid.	Base 1	Base 2	Trein.	Valid.	Base 1	Base 2
Normal	100,0	100,0	63,8	79,5	100,0	100,0	67,4	78,3
Def Elétrico	92,2	57,4	66,2	54,9	94,0	57,3	86,7	57,3
Def Térmico	100,0	85,7	71,2	70,7	100,0	100,0	75,9	74,8
Segunda análise diagnóstico	Geração da Rede (IEC + Base 1)		Diagnóstico		Montagem da Árvore (IEC + Base 1)		Diagnóstico	
	Trein.	Valid.	Base 2		Trein.	Valid.	Base 2	
Normal	94,1	80,6	89,9		94,1	90,6	92,8	
Def Elétrico	98,5	83,8	70,5		97,8	93,8	82,8	
Def Térmico	97,1	93,5	83,3		96,9	96,8	84,0	

3. Discussões e conclusões

Em todas as análises com redes neurais foram feitas 36 configurações para simulações, variando-se os parâmetros de acordo com o relatado anteriormente. Foram, então, realizadas 12 simulações com a função de ativação, para a camada de saída, sendo a logsig, 12 sendo a tansig e 12 com a purelin. Nessas 12 simulações foram variados os números de neurônios e de iterações. Os resultados apresentados nas tabelas anteriormente citadas são correspondentes aos melhores resultados de cada conjunto dessas configurações de simulações.

Os melhores resultados de cada análise foram obtidos com a base de dados balanceada (Tabela 2) ou utilizando o fator TGC (que correlaciona os próprios dados das bases). O fato de a base de dados ser balanceada evita alguns dos problemas como *overfitting*, que é causado quando a rede neural ou a árvore de decisão tem bons resultados para o treinamento, mas apresentam generalização pobre, tendo resultados ruins para a validação. Isso significa que a rede piorou seu desempenho em vez de melhorar, a partir de certo ponto de treinamento.

O resultado mais eficiente encontrado foi na segunda análise com o algoritmo J4.8 (Tabela 3). Também foi nessa análise que a rede neural obteve os melhores resultados. A base de dados utilizada para gerar o classificador, ou seja, a rede neural ou a árvore de decisão possuía variação maior nos dados, já que foi constituída pelo agrupamento da base IEC com a Base 1. Esse resultado era esperado, já que o processo de KDD é mais eficiente para grandes bases de dados.

Quando a base de dados foi considerada utilizando-se o cálculo com o TGC, a técnica de redes neurais melhorou um pouco os resultados e, na árvore de decisão, não foi significante a modificação.

Os resultados apresentados nas três tabelas são considerados satisfatórios, com acerto entre 75 e 90%. Esse resultado ainda pode ser melhorado se o pré-processamento realizado nas bases de dados for realizado com especialistas no conhecimento do domínio de aplicação. Outra sugestão para um trabalho futuro é utilizar a técnica de *eraly stop* como critério de parada do algoritmo de rede neural.

Um fato pertinente a se discutir é a dificuldade de obtenção de dados cromatográficos organizados e com diagnósticos confirmados por medições específicas. Não se devem levar em consideração apenas os teores de concentrações instantâneos; o mais confiável seria um estudo da taxa de variação desses teores, sendo essa taxa essencial para a decisão de diagnóstico ou não de determinado transformador.

Com a disponibilidade de um banco de dados adequado para treinamento, também é possível aumentar o número de saídas da rede neural, por exemplo dividindo os casos de falha elétrica em alta energia e baixa energia.

Outros fatores também precisam ser considerados, por exemplo a migração de gases entre a celulose e o óleo do transformador de acordo com a temperatura do meio. Esse fato proporciona, para o mesmo transformador, valores diferentes de teor de concentrações dos gases, de acordo com a temperatura ambiente.

As diferenças entre os transformadores, como: volume do óleo isolante, aspectos construtivos, classes de tensões e fatores ambientais envolvidos, aliados à incerteza nos processos de cromatografia dos transformadores, impossibilitam a obtenção de um

classificador com 100% de diagnósticos corretos. Mas a combinação dos resultados com os métodos apresentados e a experiência dos especialistas aumentam a confiabilidade dos diagnósticos.

Referências

BRAGA, A. P.; CARVALHO, A. C. P. L. F.; LUDERMIR, T. B. Redes neurais artificiais. In: REZENDE, Solange Oliveira (Org.). **Sistemas inteligentes**. 1. ed. Barueri, SP: Manole, 2003. v. 1, p. 141-168.

COVA, C. J. G.; CRUZ, E. A. Teoria das decisões: um estudo do método lexicográfico. **Revista Pensamento Contemporâneo em Administração**, v. 1, p. 3-4, 2007.

DUVAL, M.; de PABLO, A. Interpretation of gas-in-oil analysis using IEC publication 60599 and IEC TC 10 databases. **IEEE Electrical Insulation Magazine**, v. 17, n. 2, mar./abr. 2001.

DEMUTH, H.; BEALE, M. **Neural network toolbox user´s guide for use with MATLAB®**. Versão 4, 2002.

DUVAL, M. A Review of faults detectable by gas-in-oil analysis in transformers. **IEEE Electrical Insulation Magazine,** v. 18, n.3, p. 8-17, maio/jun. 2002.

FAYYAD, U.; SHAPIRO, G.P.; SMYTH, P. From data mining to knowledge discovery in databases. In: SCIENTIFIC AND STATISTICAL DATABASE MANAGEMENT CONFERENCE, 9., 1997. **Proceedings...** [S.l. : s.n.], 1997. p. 2-11.

HAGAN, M. T.; FORESSE, F. D. Gauss-Newton Approximation to Bayesian Learning. In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 1997. **Proceedings**... [S.l. : s.n.], 1997. v. 3, p. 1930-1935.

HAYKIN, S. **Redes Neurais** – Princípios e prática. 1. ed. Bookman, 2001. 898 p.

IEC 60599. **Mineral oil-impregnated electrical equipment in service** – Guide to the interpretation of dissolved and free gases analysis. [S.l.]: International Electrotechnical Commission, 1999.

JACOBS, R. A. **Increased rates of convergence through learning rate adaptation**. Massachusetts: University of , 1987. p. 295-307. (Technical Reprt number 1).

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

MORAIS, D. R. **Ferramenta inteligente para detecção de falhas incipientes em transformadores baseada na análise de gases dissolvidos no óleo isolante**. Florianópolis: UFSC, 2004.

PASSOS, M. G. **Modelos de dispositivos de microondas e ópticos através de redes neurais artificiais de alimentação direta**. Natal: UFRGN, 2006.

POSSA, B. A. V.; CARVALHO, M. L. B. de; REZENDE, R. S. F.; MEIRA JR., W. **Data mining**: técnicas para exploração de dados. Belo Horizonte: UFMG, 1998.

QUINLAN, J. C. **C4.5**: programs for machine learning. San Mateo: Morgan Kaufmann, 1993. 302 p.

RIEDMILLER, M.; BRAUN, H. A direct adaptive mMethod for faster backpropagation learning: the RPROP algorithm. In: IEEE INTERNATIONAL CONFERENCE ON NEURAL NETWORKS, 1993. **Proceedings**... [S.l. : s.n.], 1993. v. 1, p. 586-591.

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

TAFNER, M. A. Redes neurais artificiais: aprendizado e plasticidade. **Revista Cérebro e Mente**, Campinas, UNICAMP, mar./maio 1998.

WIDROW, B.; HOFF, M. E. **Adaptative switching circuit**. New York: IRE WESCON Convention Record, 1960. p. 96-104.

Recebido em 05/02/2009

Publicado em 02/10/2009