

# SISTEMA INTELIGENTE PARA DETECÇÃO E DIAGNÓSTICO DE FALHAS EM GERADORES DE ENERGIA ELÉTRICA DE USINAS TERMOELÉTRICAS

*Luiz Biondi Neto<sup>1</sup>*

*Fernando Hideo Fukuda<sup>2</sup>*

*Pierre Teixeira Rodrigues<sup>3</sup>*

*Marcio Américo<sup>4</sup>*

*Moacir Pereira Lima Junior<sup>5</sup>*

**Resumo:** Admitindo-se que todos os dispositivos de proteção dos Geradores Termoelétricos de uma usina operem normal e precisamente e que os alarmes enviados pelos dispositivos de sensoriamento sejam recebidos corretamente, uma ocorrência de falha será configurada por um padrão particular dos estados desse conjunto de alarmes acionados. Assim, um alarme específico ou um conjunto desses alarmes acionados simultaneamente podem indicar uma falha de um gerador. Técnicas inteligentes se mostram adequadas para tratar o processamento desses alarmes. A pesquisa envolveu 4 geradores de características praticamente idênticas na qual foram mapeadas 27 falhas por Gerador. Foram modelados uma Rede Neural Artificial e um Sistema Especialista com a base de conhecimento armazenada em um sistema de gerenciamento de banco de dados relacional. O sistema final foi construído, rigorosamente, conforme as necessidades apresentadas pela Usina Termoelétrica Norte Fluminense (UTENF) e os resultados, segundo os especialistas atendem aos seus requisitos na detecção e diagnóstico das falhas dos geradores.

**Palavras-chave:** Diagnóstico de Falhas em Geradores Termoelétricos; Processamento de Alarmes; Redes Neurais Artificiais, Sistema Especialista.

**Abstract:** Admitting that all Thermoelectric Generators' protection devices of a plant operate normal and precisely and that the alarms sent by sensory devices be received correctly, a fault occurrence will be configured by a particular standard of the state of this set of activated alarms. Therefore, a specific alarm or a set of these alarms activated simultaneously can indicate a fault of a generator. Intelligent techniques are adequate to process these alarms. This research involved 4 generators of practically identical characteristics in which 27 faults were mapped for each Generator. They were modelled by an Artificial Neural Network and an Expert System with the knowledge base stored in a relational database management system. The system was built, rigorously, conformably to the needs presented by the Usina Termoelétrica Norte Fluminense Plant (UTENF) and the results, according to the experts, attend to their requirements in detecting and diagnosing the generators' failures.

**Keywords:** Faults Detection and Diagnosis in Thermoelectric Generators; Alarms processing; Artificial Neural Networks, Expert System.

<sup>1</sup> Departamento de Engenharia Eletrônica, DETEL/FEN/UERJ, E-mail: luizbiondi@terra.com.br

<sup>2</sup> Centro de Estudos da Faculdade de Engenharia, CEFEN/FEN/UERJ, E-mail: fernando@fukuda.com.br

<sup>3</sup> Jordão Engenharia, JORDÃO, E-mail: pierre.rodrigues@jordaoengenharia.com.br

<sup>4</sup> Jordão Engenharia, JORDÃO, E-mail: marcio.americo@jordaoengenharia.com.br

<sup>5</sup> Usina Termoelétrica Norte Fluminense, UTENF, E-mail: mpereira@nortefluminense.com.br

## 1. INTRODUÇÃO

Provenientes dos diversos dispositivos de sensoriamento distribuídos pelo sistema termoeletrico, um conjunto específico de alarmes mapeia uma falha do sistema KHAN (2007) e MERCEDE (1993). Esses alarmes são sinais binários que revelam, basicamente, o estado “1” / “0” dos sensores, relés de proteção e disjuntores, embora sinais referentes à situação da tensão, corrente e frequência também possam ser recebidos do sistema de aquisição de dados.

Se todos os dispositivos de proteção operar normal e precisamente e, além disso, os dados enviados pelos dispositivos de sensoriamento forem recebidos corretamente, uma ocorrência de falha será configurada por um padrão particular dos estados desse conjunto de alarmes acionados DAVIS (2006) e HEWITSON (2004). É possível que apenas um alarme acionado defina uma situação de operação anormal e conseqüentemente uma falha no gerador, situação semelhante a que ocorre no caso investigado.

Assim o problema da detecção de falhas revela-se como um problema típico de classificação e ocorrências de falhas e podem ser detectadas por operadores. No entanto, fatores como a fadiga, a inexperiência e a dificuldade do operador em lidar com situações não usuais ou de missões críticas, levam, eventualmente, a diagnósticos incorretos, com conseqüente prejuízo da segurança e eficiência do sistema elétrico.

Os sinais de alarmes usados na modelagem de nosso problema foram: Distância de fase; Variação de Volts / Hz; Subtensão (módulo); Subtensão harmônica; Direcional de potência; Perda de campo; Sobrecorrente seqüência negativa; Sobrecorrente instantânea; Falha no disjuntor; Tempo definido de sobrecorrente; Sobrecorrente instantânea - neutro; Energização inadvertida; Sobrecorrente de neutro (curva tempo inverso); Sobrecorrente (curva tempo inverso) com restrição de tensão; Sobretensão (módulo); Sobretensão (módulo) para neutro; Detecção de abertura de fusível; Proteção de escalonamento de carga; Frequência; Taxa de mudança de frequência; Diferencial de fase; Diferencial de terra; Sobrecorrente no terminal do estator; Proteção de terra do campo / Detecção de desgaste das escovas; Falha no disjuntor do gerador; Proveniente do BOP, CT's e ST. - Falha no sistema de excitação da máquina;

Trip por alta temperatura do gás frio do gerador. Esses alarmes estão relacionados com os seguintes equipamentos de proteção: 21; 24; 27; 27TN; 32; 40; 46; 50; 50BF; 50DT; 50N 50/27; 51N; 51V; 59; 59N; 60FL; 78; 81; 81R; 87; 87GD; 51T; 64F/64B; 86BF; 86-1, sendo que cada alarme mapeia uma falha correspondente, segundo mapeamento fornecido pela usina.

Técnicas computacionais inteligentes como as Redes Neurais artificiais (RNAs) BIONDI e COELHO (2004), BIONDI e LINS (2004), ZURADA (1992) e BRAGA (2000), com características adaptativas de generalização e de aprendizado por experiência, e os Sistemas Especialistas (SE), que permitem o processamento simbólico do conhecimento organizado do problema, e podem ainda, tratar adequadamente as incertezas que envolvem o processamento dos alarmes, têm sido pesquisadas para auxiliar o profissional de operação de sistemas BIONDI (2003).

Essas técnicas, isoladas ou combinadas em um sistema híbrido, visam deixar o operador mais concentrado na ação corretiva final a ser tomada, na qual sua participação é muito importante.

Assim o objetivo principal dessa pesquisa é investigar um Sistema Inteligente para Detecção e Diagnóstico de Falhas em Geradores de Energia Elétrica de Usinas Termoeletricas.

## 2. FUNDAMENTAÇÃO TEÓRICA

### 2.1 REDES NEURAIS ARTIFICIAIS

As RNAs são estruturas paralelas compostas de neurônios artificiais, densamente conectadas que apresentam duas fases distintas de processamento: a aprendizagem (treinamento) e a execução ZURADA (1992) e HAYKIN (1994).

Na aprendizagem são apresentados estímulos à entrada denominados padrões de treinamento um mapeamento envolvendo Alarmes e Falhas definidos pelos especialistas em nosso caso o mapa de falhas fornecido pela usina. Na fase de execução, a RNA recebe como entrada um conjunto de padrões de testes (alarmes) que, normalmente, não fizeram parte do treinamento e apresenta o resultado, que no caso investigado representa a Falha detectada.

A arquitetura utilizada foi baseada no Perceptron de Múltiplas Camadas (PMC). O algo-

ritmo usado no treinamento da rede PMC, denominado de retro-propagação do erro, pode ser dividido em duas etapas: “direta” e “reversa”.

A primeira consiste na propagação dos estímulos apresentados à entrada para a saída. Esses estímulos fluem por toda a rede, recebendo a computação neural, camada por camada, até gerarem a saída. A partir do resultado desejado, calcula-se um erro  $E$  na camada de saída.

A segunda etapa ocorre em sentido contrário. O erro calculado é então retro propagado através das camadas antecessoras, atualizando os pesos  $W$  das conexões.

O algoritmo de retro-propagação de erro é baseado no método do gradiente descendente, que computa as derivadas parciais de uma função de erro, com relação ao vetor peso  $W$  de certo vetor de entrada HAYKIN (1994).

A regra de Widrow-Hoff ZURADA (1992) é a base fundamental de diversos métodos de treinamento das RNAs, inclusive PMCs, HAYKIN (1994). Esta regra avalia o erro  $E$  a cada iteração através da derivada parcial desse erro  $E$  em relação ao peso  $W$ . A variação do peso é dada por (1):

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} \quad (1)$$

O algoritmo de retro-propagação de erro se baseia no modelo do aprendizado supervisionado, retro propagando o erro  $E$  da camada de saída na atualização dos pesos de todas as camadas da RNA.

A camada de entrada, que não apresenta atividade neural, tem apenas a finalidade de transferir os estímulos apresentados à RNA para as camadas seguintes.

A camada de saída contém tantos neurônios artificiais quantos forem necessários para modelar o problema.

A estrutura pode ter uma ou várias camadas intermediárias de neurônios artificiais.

O número de camadas e o número de neurônios por camada intermediária dependem da complexidade do problema que se deseja resolver influenciando, finalmente, na capacidade de generalização da RNA.

Esses parâmetros são ajustados durante o treinamento e dependem da experiência profissional de quem está treinando a rede BIONDI (2005).

A arquitetura do PMCs é apresentada na Figura 1, onde  $N_i$ ,  $N_h$  e  $N_o$  correspondem, respectivamente, ao número de elementos da camada de entrada, intermediária e saída.

Os vetores  $\mathbf{X}$ ,  $\mathbf{O}$  e Target  $\mathbf{T}$  representam o estímulo apresentado a entrada, a resposta da rede àquele estímulo e o resultado desejado fornecido pelo “supervisor”.

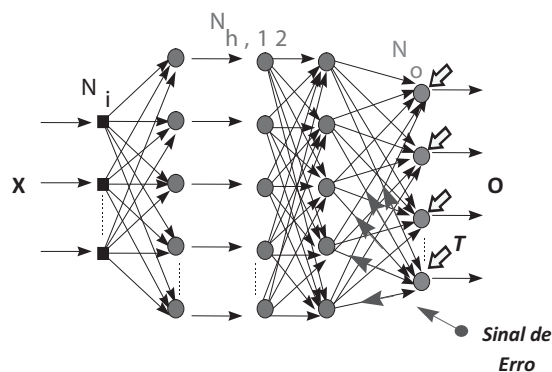


Figura 1. Perceptron de Múltiplas Camadas.

A seguir o algoritmo de retro-propagação de erro é desenvolvido passo a passo.

**Passo 1** – São apresentados à rede os pares estímulos  $\mathbf{X}$  e resultados desejados (alvos)  $\mathbf{T}_j$ , correspondentes, disponíveis no mapeamento feito, previamente, por especialistas.

Chama-se de  $NET_j$  a soma ponderada dos estímulos pelos respectivos pesos.

**Passo 2** - São calculadas, para todas as camadas, as respostas  $O_j$  de cada neurônio artificial que compõe a rede.

**Passo 3** - Na camada de saída calcula-se o erro entre a resposta desejada  $T$  e a resposta atual da rede  $O_j$ , usando-se a função de custo  $E$  que representa o erro quadrático médio (2).

Se  $E$  for menor que um valor mínimo pré-estabelecido, o treinamento é encerrado; caso contrário calcula-se o sinal de erro, atualizando-se os pesos por retro-propagação.

$$E = \frac{1}{2} \sum_{i=1}^N (T_j - O_j)^2 \quad (2)$$

$$O_j = F(NE_{T_j}) = F\left(\sum_{i=1}^N X_i W_{ij}\right) \quad (3)$$

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial(NET_j)} \frac{\partial(NET_j)}{\partial W_{ij}} \quad (4)$$

$$\text{Se } NET_j = \sum_{i=1}^N X_i W_{ij} \quad (5)$$

$$\frac{\partial(NET_j)}{\partial W_{ij}} = X_i \quad (6)$$

$$\frac{\partial E}{\partial W_{ij}} = \frac{\partial E}{\partial(NET_j)} X_i \quad (7)$$

O sinal de erro é definido por (8):

$$\delta_j \equiv - \frac{\partial E}{\partial(NET_j)} \quad (8)$$

Os pesos são atualizados por (9):

$$\Delta W_{ij} = - \eta \frac{\partial E}{\partial W_{ij}} = \eta \delta \quad (9)$$

**Passo 4** - Definição e cálculo do sinal de erro para cada camada da rede.

**Passo 5** - Para os neurônios artificiais da camada de saída o valor do sinal de erro  $\delta_{j0}$  e da atualização dos pesos é dado por (13):

$$\delta_{j0} = - \frac{\partial E}{\partial(NET_j)} = - \frac{\partial E}{\partial(O_j)} \frac{\partial(O_j)}{\partial(NET_j)} \quad (10)$$

$$O_j = F(NET_j) \quad (11)$$

$$\frac{\partial(O_j)}{\partial(NET_j)} = F'(NET_j) \quad (12)$$

$$\delta_{j0} = - \frac{\partial E}{\partial(NET_j)} = -F'(NET_j) \frac{\partial E}{\partial(O_j)} \quad (13)$$

Como o erro quadrático médio é dado por (2) vem que:

$$\frac{\partial E}{\partial(O_j)} = -(T_j - O_j) \quad (14)$$

$$\delta_{j0} = F'(NET_j)(T_j - O_j) \quad (15)$$

Então: a variação do peso é dada por (16):

$$\Delta W_{ij} = \eta \delta_{j0} X_i = \eta F'(NET_j)(T_j - O_j) X_i \quad (16)$$

**Passo 6** - Para os neurônios das camadas intermediárias o valor do sinal de erro  $\delta_{jh}$  e da atualização dos pesos é calculado da seguinte forma:

Suponha inicialmente que  $k$  pertença à camada de saída (17).

Assim:

$$E = \frac{1}{2} \sum_{k=1}^{N_0} (T_k - O_k)^2 \quad (17)$$

Reescrevendo o erro  $E$  em função de  $NET$ , é possível verificar a contribuição de cada neurônio artificial de cada camada intermediária.

Assim se:

$$E = E(NET_1, NET_2, NET_3, \dots, NET_k, \dots, NET_{N_0}) \quad (18)$$

Então:

$$\delta_{jh} = - \frac{\partial E}{\partial(NET_j)} = -F'(NET_j) \frac{\partial E}{\partial(O_j)} \quad (19)$$

Para cada neurônio pertencente a uma intermediária tem-se (20):

$$\begin{aligned} \frac{\partial E}{\partial(O_j)} &= \sum_{k=1}^{N_0} \frac{\partial E}{\partial(NET_k)} \frac{\partial(NET_k)}{\partial(O_j)} = \\ &= \sum_{k=1}^{N_0} \frac{\partial E}{\partial(NET_k)} \frac{\partial \sum_{l=1}^{N_h} O_l W_{lk}}{\partial(O_j)} \end{aligned} \quad (20)$$

Na qual  $l$  pertence a camada escondida. O único termo diferente de zero é o de  $l=j$  e vale  $W_{jk}$ .

$$\frac{\partial E}{\partial(O_j)} = \sum_{k=1}^{N_0} \frac{\partial E}{\partial(NET_k)} W_{jk} \quad (21)$$

Como:

$$\delta_k = - \frac{\partial E}{\partial(NET_k)} \quad (22)$$

$$\frac{\partial E}{\partial(O_j)} = - \sum_{k=1}^{N_0} \delta_k W_{jk} \quad (23)$$

Então:

$$\delta_{jh} = -F'(NET_j) \frac{\partial E}{\partial(O_j)} = F'(NET_j) \sum_{k=1}^{N_0} \delta_k W_{jk} \quad (24)$$

Finalmente, atualizam-se os pesos na camada escondida por (25):

$$\Delta W_{ij} = \eta \delta_{jh} X_i = \eta F'(NET_j) \sum_{k=1}^{N_0} \delta_k W_{jk} X_i \quad (25)$$

O algoritmo de retro-propagação, embora aplicado com êxito em várias aplicações, apresenta algumas dificuldades: tempo de convergência longo; paralisia da rede; e mínimo local CIHAN (2007).

Se a taxa de aprendizado e a arquitetura da rede não forem escolhidas adequadamente ou se o ajuste dos pesos for infinitesimal, o tempo de treinamento cresce infinitamente.

Por outro lado, a rede pode operar em uma região onde a derivada da função de ativação seja nula ou infinitamente pequena, provocando a paralisia da rede.

A equação (13) mostra que se a derivada da função de ativação for nula, o reajuste dos pesos também será nulo e o treinamento fica paralisado.

Outro ponto que merece destaque é a possibilidade da existência de superfícies de erro que apresentem mínimos locais.

Como se sabe, o algoritmo emprega o método do gradiente decrescente, que flui na superfície de erro, ajustando os pesos na direção de um mínimo. Assim, durante o treinamento a rede pode ficar presa em um desses mínimos locais, não conseguindo chegar ao mínimo global HAYKIN (1994).

A forma de minimizar esses problemas é encontrar o valor adequado para taxa de aprendizado  $\eta$ . Se  $\eta$  for muito grande o aprendizado é rápido, porém corre-se o risco de a rede entrar em oscilação sem alcançar o mínimo desejado.

Se  $\eta$  for muito baixo, aumentam as chances de a rede ficar presa em um mínimo local ou, na melhor das hipóteses, o treinamento será muito lento.

A maneira de se oferecer mais estabilidade ao algoritmo de retro-propagação é através da adição de um termo conhecido por momento ( $\alpha$ ).

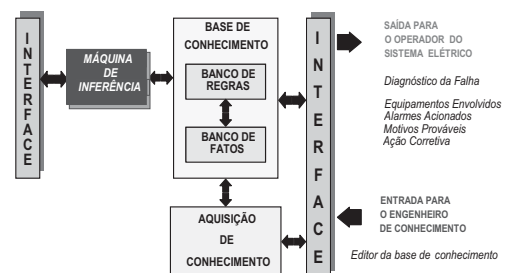
O momento permite que o ajuste dos pesos, a cada iteração, seja proporcional à magnitude do último ajuste.

O termo  $\alpha$  é geralmente escolhido menor que a unidade. A equação (26) mostra a inserção do termo momento no ajuste dos pesos.

$$\Delta W_{ij}(n+1) = \eta \delta_j O_j + \alpha \Delta W_{ij}(n) \quad (26)$$

## 2.2 SISTEMA ESPECIALISTA

Um Sistema Especialista (SE) é um programa inteligente de computador que usa “raciocínio inferencial” para solucionar problemas de um campo específico do conhecimento humano. Sua característica principal é a capacidade de explicação, semelhante a encontrada nos seres humanos, dotados de alto grau de especialização GIARRATANO (2005). A Figura 2 representa o diagrama em blocos de um sistema especialista, composto de dois módulos principais: máquina de inferência; e base de conhecimento.



**Figura 2. Sistema Especialista.**

Além dos blocos principais existe uma interface homem-máquina (HMI – *Human-Machine Interface*), que o usuário ou o especialista se comunique com o sistema sem a assistência de um programador.

A máquina de inferência ou interpretador de regras guia o processo de raciocínio através da base de conhecimento, comparando fatos contidos na base de fatos com o conjunto de regras.

Por muito tempo foram usados programas que continham heurísticas gerais de como solucionar problemas. Entretanto se o conhecimento estiver representado por lógica matemática a máquina de inferência é o método de Resolução. Neste caso, descreve-se o problema através de lógica de 1ª ordem ou de predicados, deduzindo-se as soluções por um procedimento geral de provas GIARRATANO (2005), RUSSELL (2003).

Se o conhecimento estiver representado por regras de produção a máquina de inferência é o encadeamento para frente ou o encadeamento para trás. No encadeamento para frente o objetivo é descobrir consequência de novos fatos e no encadeamento para trás a prova da hipótese é procurada na base de conhecimento (regras e fatos).



Assim, a máquina de inferência é a parte do sistema inteligente responsável pela geração de novos conhecimentos obtidos através da manipulação da base de conhecimentos existente e depende da maneira como o conhecimento está representado.

A base de fatos é responsável pelo armazenamento do contexto do sistema em um dado instante de tempo, deixando esses dados disponíveis para serem usados pelo sistema a qualquer momento. Assim, antes de uma regra ser aplicada, seus elementos condicionais devem estar presentes na base de fatos (contexto).

A base de regras encerra o conjunto de todas as regras de produção, necessárias para representar o conhecimento estruturado do problema. Uma regra de produção é representada por um par ordenado (*condição, ação*), onde a condição é uma expressão lógica de fatos e a ação a geração de novos fatos. Se a condição for verdadeira uma regra está apta a ser disparada RUSSELL (2003). O disparo da regra cria novos fatos que são atualizados, na base de fatos, para que outras regras que possuam esses fatos em sua condição possam ser satisfeitas.

Finalmente, é importante ressaltar o processo de aquisição do conhecimento, realizado através da interação com especialistas humanos. Nessa etapa o engenheiro do conhecimento, administrador da base de conhecimento, através de inúmeras entrevistas com diversos especialistas da área do conhecimento do problema, procura construir e aperfeiçoar a base de conhecimento. Muitas vezes as informações passadas pelos diversos especialistas, embora corretas e confiáveis, são eventualmente incompletas ou conflitantes, tornando esta fase a mais crítica no desenvolvimento de qualquer sistema especialista.

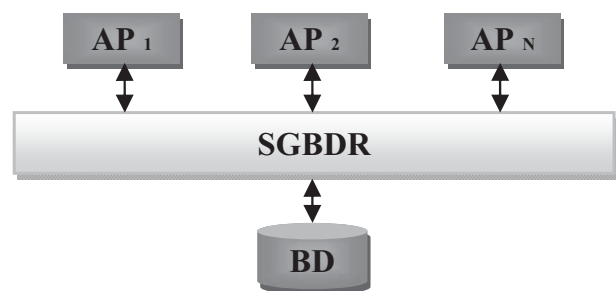
Atualmente sistemas que permitem adquirir, processar e armazenar o conhecimento são baseados em sistemas de gerenciamento de banco de dados relacionais, facilitando sobremaneira a ação do engenheiro do conhecimento no tocante ao aproveitamento das facilidades inerentes nestes softwares, eliminando a necessidade de implementá-las.

### 2.3 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS RELACIONAL

Os dados são fatos que possuem um significado implícito. Um banco de dados é uma

coleção de dados relacionados com algum significado inerente SILBERSCHATZ(1997). Um banco de dados representa alguns aspectos de um mundo real particular conhecido como universo de discurso ELMASRI (2005). Um sistema de gerenciamento de banco de dados relacional (SGBDR) é uma coleção de softwares de propósito geral que facilita o processo de: definir, construir e manipular um banco de dados (BD); situada entre uma ou mais aplicações (APs) e um ou mais BDs ELMASRI (2005). Portanto, um SGBDR é uma camada intermediária, conhecida também como *middleware*, que provê a interface entre as aplicações e os BDs, conforme mostrado na Figura 3.

Esta arquitetura permite que as aplicações compartilhem as funções especializadas existentes na coleção de programas do SGBDR e desta forma evita-se que estas funções sejam implementadas nas aplicações. As principais vantagens dos SGBDRs são: separação entre programas e dados, armazenamento persistente dos dados, padronização dos dados, compartilhamento dos dados, controle de redundância dos dados, restrições de integridade, restrição de acesso, eliminação de códigos redundantes nas aplicações para gerenciamento de dados, flexibilidade, disponibilidade e escalabilidade SILBERSCHATZ (1997) e ELMASRI (2005). O conjunto formado pelo SGBDR e os BDs é conhecido como sistema de banco de dados ELMASRI (2005).



**Figura 3. Arquitetura em camadas de um sistema de banco de dados.**

A modelagem de um BD de um SGBDR é realizada através do modelo entidade-relaciona-

mento (MER) que especifica: entidades de dados, relacionamentos entre as entidades de dados, atributos das entidades de dados e regras de um BD ELMASRI (2005).

Uma entidade é um objeto do mundo real que pode ser identificada de forma unívoca em relação a todos os outros objetos. Uma entidade pode ser concreta, abstrata ou um conceito. Uma entidade é representada por um conjunto de atributos.

Os atributos são propriedades descritivas de cada membro de um conjunto de entidades. Cada entidade tem atributos com seus próprios valores. Para cada atributo existe um conjunto de valores possíveis, chamado domínio ou conjunto de valores SILBERSCHATZ (1997). Uma super-chave é um conjunto de um ou mais atributos que permitem identificar univocamente uma entidade. Chaves candidatas são super-chaves para as quais nenhum subconjunto possa ser uma super-chave. A chave primária é uma chave candidata escolhida pelo projetista do BD como de significado principal para a identificação de entidades.

Um relacionamento é uma associação entre uma ou várias entidades SILBERSCHATZ (1997).

As regras podem ser definidas como predicados, asserções ou verdades estabelecidas sobre entidades, relacionamentos e atributos, que complementam a semântica de um modelo de dados BARBIERE (1994). As regras podem ser classificadas em: regras de entidades, regras de relacionamentos, regras de atributos e regras de negócio.

O projeto geral de um BD é chamado esquema do BD, que é definido durante o projeto do BD SILBERSCHATZ (1997). O conjunto de informações contidas em determinado BD, em um dado momento, é chamado instância do BD SILBERSCHATZ (1997).

Os dados e os relacionamentos de um BD de um SGBDR são representados por tabelas.

A linguagem padrão suportada pela maioria dos SGBDR é a *Structured Query Language* – SQL originalmente desenvolvida pela IBM e atualmente mantida pelo ANSI e pela ISO SILBERSCHATZ (1997). A SQL atualmente permite definir, construir e manipular um BD ELMASRI (2005).

### 3. TREINAMENTO E RESULTADOS DA RNA

#### 3.1 TREINAMENTO

A RNA foi treinada usando-se o algoritmo de retro-propagação de erro. Foram testadas as metodologias usando-se o algoritmo de Levenberg-Marquardt RANGANATHAN (2004) e ROWEIS (2003) e o algoritmo Resiliente de retro-propagação de erro RIEDMILLER (1993) que se mostrou o mais eficiente.

O bom resultado atingido deve-se ao fato do algoritmo usar o sinal do gradiente na atualização dos pesos ao invés do módulo do gradiente normalmente usado nos demais. Dessa forma o algoritmo converge mais rapidamente no final do treinamento fornecendo resultados plenamente satisfatórios.

A ferramenta computacional utilizada para desenvolver as RNAs foi o ToolBox do MatLab Vers: 2008a. Com esse aplicativo é possível treinar e testar RNAs com diversas arquiteturas, variar os parâmetros de treinamento, escolher o algoritmo de aprendizado de melhor desempenho e produzir curvas de ótima qualidade técnica.

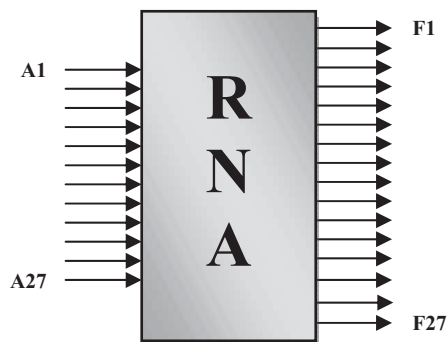
A máquina usada nos testes foi um computador com processador Intel Centrino Dual Core, com frequência de clock de 2,5 GHz, memória RAM de 4 GB e disco rígido de 200 GB.

Foram realizados diversos testes usando-se diferentes arquiteturas, sendo adotada a arquitetura com uma camada escondida com 35 neurônios artificiais.

A função de ativação escolhida foi a “tangente hiperbólica” e usada tanto na camada escondida quanto na de saída. A tolerância de erro de treinamento adotada foi de  $10^{-6}$ .

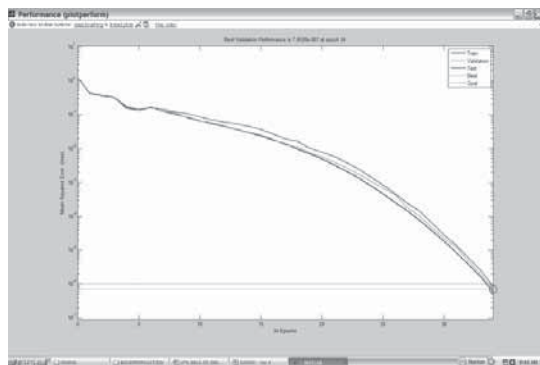
Aproveitamos também para, durante o treinamento, realizarmos os procedimentos referentes a validação e teste da rede. Para isso dos padrões de treinamento foram reservados 20 % para validação e 20 % para teste. Dessa forma ao final do treinamento tem-se boa idéia da capacidade de generalização da mesma. Diversos testes foram realizados visando escolher a configuração mais adequada.

O Modelo da RNA referente ao Gerador #1 é mostrado na Figura 4 e dispõem de 27 alarques na entrada para 27 neurônios na saída, um para cada falha, correspondentes ao mapeamento fornecido pela usina.



**Figura 4. Modelo da RNA do Gerador #1.**

As curvas de treinamento (Train), validação (Validation) e teste (Test) citadas anteriormente e referentes ao Gerador #1 estão mostradas na Figura 5.



**Figura 5. Curvas de treinamento, validação e teste do Gerador #1.**

A curva de treinamento reflete a variação do erro quadrático médio em função da quantidade de épocas (epochs) isto é, o número de vezes que um conjunto completo de padrões foi apresentado a RNA.

Os valores desejados especificados nos padrões de treinamento foram +1 para situação de Falha e -1 para situação de não ocorrência de falha.

Esse aumento do poder de discriminação da RNA só foi possível, pois usamos como a função de ativação a tangente hiperbólica que varia entre -1 e +1 limitando a saída dos neurônios artificiais dentro dessa faixa.

### 3.2 RESULTADOS

Depois de devidamente treinada a RNA generalizou adequadamente, conforme mostra a

Tabela 1 que representa a saída dos 27 neurônios artificiais (linhas) da RNA e que por questões de espaço e legibilidade são apresentados apenas 5 casos de Falhas detectadas (colunas).

**Tabela 1 - Detecção Neural da Falhas**

FLH 1	FLH 2	FLH 3	FLH 4	FLH 5	FLH 6
0.9945	-0.9965	-0.9999	-0.9998	-0.9978	-0.9995
-1.0000	0.9970	-1.0000	-1.0000	-0.9999	-1.0000
-1.0000	-0.9999	0.9987	-1.0000	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	0.9940	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	0.9999	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	0.9984
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-0.9982	-0.9999	-1.0000	-1.0000	-1.0000
-1.0000	-0.9999	-0.9988	-1.0000	-1.0000	-0.9999
-1.0000	-1.0000	-1.0000	-0.9998	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-0.9999	-1.0000	-1.0000	-0.9985	-0.9994
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-0.9999	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-0.9999	-0.9981	-1.0000	-0.9997	-0.9998
-1.0000	-0.9996	-1.0000	-1.0000	-0.9998	-1.0000
-1.0000	-0.9971	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-0.9999	-1.0000
-1.0000	-1.0000	-1.0000	-1.0000	-0.9999	-1.0000
-0.9999	-1.0000	-1.0000	-0.9998	-0.9997	-1.0000

O resultado do teste final, que envolveu mais de uma centena de cenários, apresentou erro percentual máximo de 0.3782 % o que significa que a rede, de fato, aprendeu os padrões.

Considerando-se que as RNAs são técnicas adequadas à classificação e portando boa opção para a detecção da falhas, foram usadas inicialmente apresentando bons resultados, conforme mostram a Figura 5 e a Tabela 1.



No entanto seria necessário que além da RNA fosse usado um SE, constituindo dois módulos distintos e apresentando maior dificuldade na manutenção do sistema completo.

Assim, buscou-se com o SGBDR uma melhor opção de desenvolvimento, mais integrada e eficiente, envolvendo a detecção e o diagnóstico da falha em um mesmo módulo, usando-se as RNAs para validação da detecção.

#### 4. MODELAGEM E RESULTADOS DO SISTEMA ESPECIALISTA COMPLETO

##### 4.1 MODELAGEM

Uma regra de produção tem, em geral, a seguinte estrutura básica BARBIERE (1994):

**SE** antecedente  
**ENTÃO** conseqüente

onde antecedente pode ser uma condição e o conseqüente pode ser um fato. Neste caso, pode-se considerar que o motor de inferência do SE busca na sua base de conhecimento um fato para uma dada condição.

Assim, foi desenvolvido um mecanismo equivalente de busca do fato para a condição em particular usando-se tabelas que armazenam o conhecimento. Assim, a regra de produção pode ser traduzida para uma consulta em SQL com a seguinte declaração:

**SELECT** conseqüente **FROM** tabela  
**WHERE** antecedente

onde a tabela ou o conjunto de tabelas armazenam o conhecimento do SE.

Como, por exemplo, considerando-se que quando o alarme de identificação 2 do gerador de identificação 1 ocorra, o SE tenha de inferir que o código do equipamento envolvido é 24, teríamos a seguinte regra de produção simplificada:

**SE** gerador\_id = 1 **E** alarme\_id = 2  
**ENTÃO** equipamento\_cod = 24

onde gerador\_id, alarme\_id e equipamento\_cod são, respectivamente, os códigos do gerador, do alarme e do equipamento.

A modelagem da base de conhecimento do SE usando MER pode ser realizada, por exemplo, com as seguintes tabelas:

- GERADORES  
(@GERADOR\_ID,GERADOR\_DESC)
- EQUIPAMENTOS  
( @ E Q U I P A M E N T O \_ ID,EQUIPAMENTO\_COD, EQUIPAMENTO\_DESC)
- ALARMES  
(@GERADOR\_ID,@ALARME\_ID,EQUIPAMENTO\_ID, ...) , onde @GERADOR\_ID REFERENCIA GERADOR\_ID DA TABELA GERADORES; ALARME\_ID REFERENCIA ALARME\_ID DA TABELA ALARMES E EQUIPAMENTO\_ID REFERENCIA EQUIPAMENTO\_ID DA TABELA EQUIPAMENTOS.
- OCORRENCIAS  
(@OCORRENCIA\_DATA, @OCORRENCIA\_HORARIO,@GERADOR\_ID,@ALARME\_ID, ...) , onde @GERADOR\_ID REFERENCIA GERADOR\_ID DA TABELA GERADORES; ALARME\_ID REFERENCIA ALARME\_ID DA TABELA ALARMES

As chaves primárias das respectivas tabelas são indicadas pelo prefixo @.

Considerando-se que a instância das Tabelas 2, 3 e 4 sejam:

**Tabela 2 – Tabela equipamentos**

equipamento id	equipamento cod	equipamento desc
1	24	Variação de Volts/Hz

**Tabela 3 – Tabela alarmes**

gerador id	alarme id	equipamento id	...
1	2	1	...

**Tabela 4 – Tabela ocorrências**

...	gerador id	alarme id	...
...	1	2	...

O código do equipamento envolvido na ocorrência do alarme pode ser inferido das tabelas pela seguinte declaração SQL:

```
SELECT DISTINCT equipamento_cod
FROM ocorrencias, alarmes, equipamentos
WHERE alarmes.equipamento_id = equipamentos.equipamento_id AND alarmes.gerador_id = ocorrencias.gerador_id AND alarmes.alarme_id = ocorrencias.alarme_id AND ocorrencias.gerador_id = 1 AND ocorrencias.alarme_id = 2
```

O SE do Sistema Inteligente de Detecção e Diagnóstico de Falhas (SIDDF) foi desenvolvido em linguagem C++. A linguagem C++ é orientada a objetos e compilada e foi escolhida porque gera códigos executáveis de alto desempenho e possibilita a portabilidade da aplicação. Desta forma, o SE pode ser portado para plataformas Unix de 64/32 bits, Linux de Tempo Real de 64/32 bits, Linux de 64/32 bits, MS Windows Server, MS Windows Vista, MS Windows XP Professional e Mac OS X.

A conexão do SIDDF em C++ com o SGBDR é feita através do ODBC (*Open Data Base Connectivity*), um conjunto padrão de interfaces, que permitem ter acesso a uma vasta gama de banco de dados sem a necessidade de codificar métodos especializados. Assim, o SIDDF suporta os principais SGBDR do mercado como, por exemplo, Oracle, IBM DB 2, MS SQL Server, Firebird, MySQL, PostgreSQL, Ingres, Sybase, Pervasive SQL, Informix, Interbase, XBase Sequiter, Virtuoso.

As interfaces homem-máquina (HMI – *Human-Machine Interfaces*) é gráfica e projetada considerando-se os conceitos de usabilidade e comunicabilidade adequados à área da aplicação, que simplificam e facilitam as operações dos usuários, maximizam a produtividade e a segurança, e minimizam os riscos, danos e prejuízos.

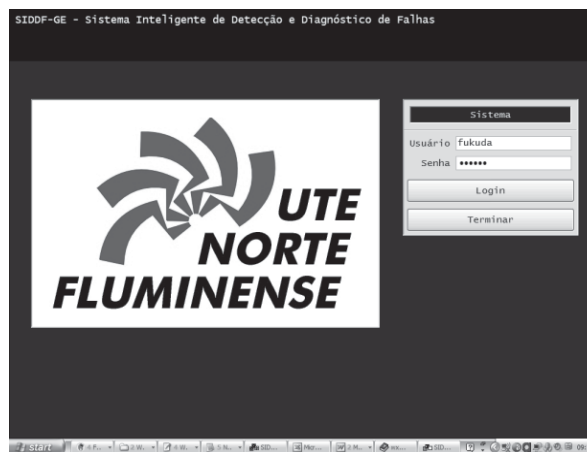
Os principais recursos do SIDDF são:

- Manter Cadastro de Usuários
- Manter Cadastro de Geradores
- Manter Cadastro de Equipamentos
- Manter Cadastro de Alarmes
- Manter Cadastro de Falhas

- Manter Cadastro de Motivos Prováveis
- Manter Cadastro de Ações Corretivas
- Manter Registro de Ocorrências de Alarmes
- Pesquisar Ocorrências de Alarmes em um período
- Detectar e Diagnosticar Falhas, que inclui Exibir Motivos Prováveis da Falha e Exibir Ações Corretivas que podem tomadas pelo operador, apoiando sua decisão.

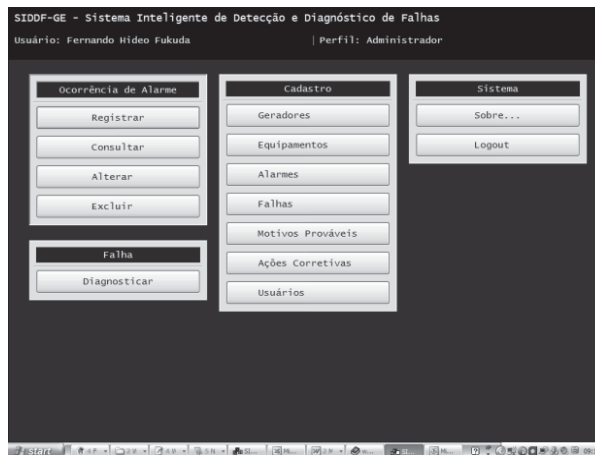
#### 4.2 RESULTADOS

A Figura 6 mostra a tela inicial do sistema completo. Realiza-se então o “login” do usuário habilitando-o à utilizar o sistema.

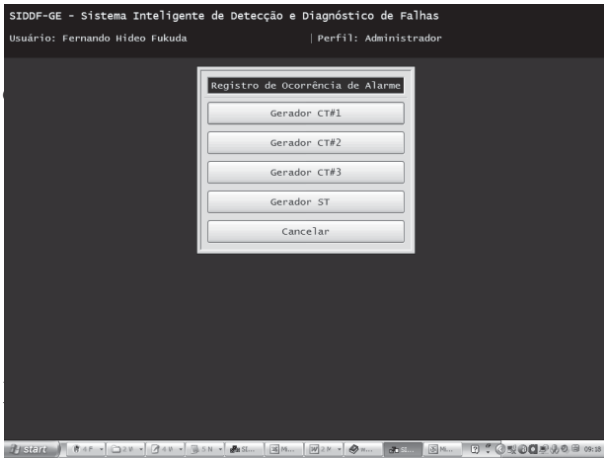


**Figura 6. Tela inicial de “login”**

Prosseguindo o usuário é capaz de registrar, consultar, alterar e excluir a ocorrência do alarme. É possível cadastrar geradores equipamentos, alarmes, falhas, motivos prováveis, ações corretivas e também usuários, bastando escolher a opção adequada conforme mostra a Figura 7.



**Figura 7. Menu Principal**



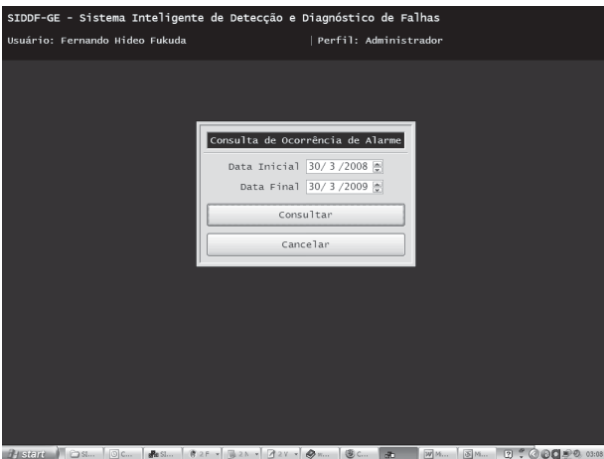
**Figura 8. Seleção do Gerador**

A Figura 9 mostra como registrar adequadamente a ocorrência de alarme.



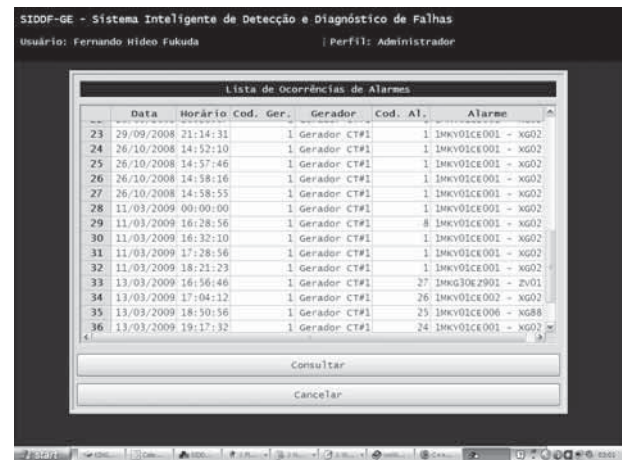
**Figura 9. Registro de Ocorrência**

A Figura 10 mostra como proceder a consulta de ocorrência de alarmes, bastando para isso digitar os limites de datas solicitados. Nesse caso o sistema proposto apresenta a listagem mostrada na Figura 11.



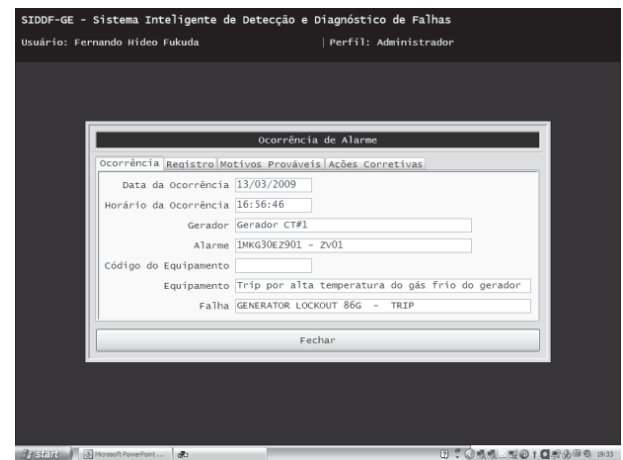
**Figura 10. Consulta de Ocorrência**

Como resultado da pesquisa é mostrado, sob forma de lista, Figura 11, qualquer uma dessas ocorrências pode ser marcada e conforme as Figuras 12 a 15 o resultado do Diagnóstico da Falha é obtido.



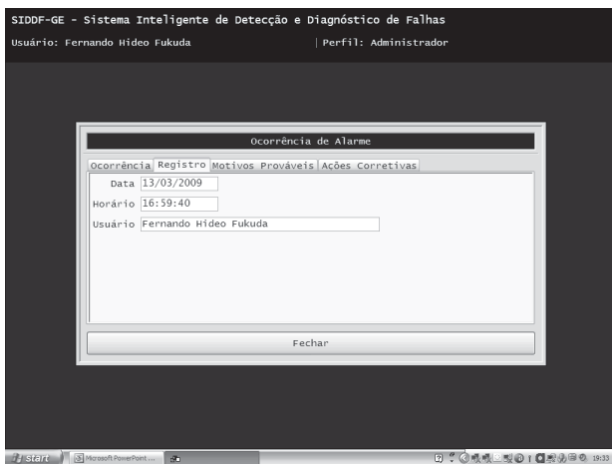
**Figura 11. Lista de Ocorrências**

Na Figura 12 a ocorrência é bem especificada e Falha é detectada segundo informações passadas pelos especialistas da usina.



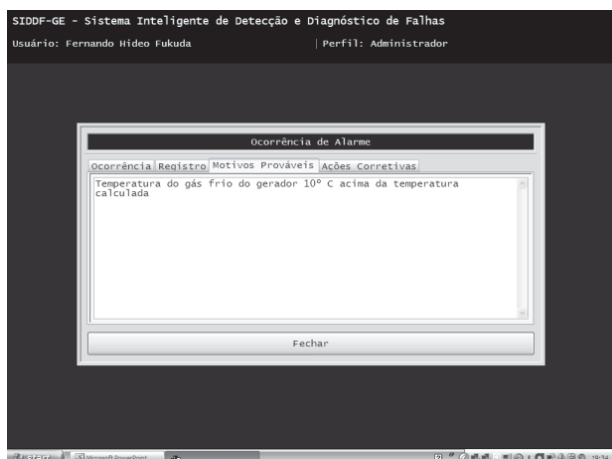
**Figura 12. Detecção da Falha**

Na Figura 13 é mostrada a tela que atende mais uma especificação dos especialistas da usina, isto é, a data, a hora e o nome do usuário responsável pelo registro.



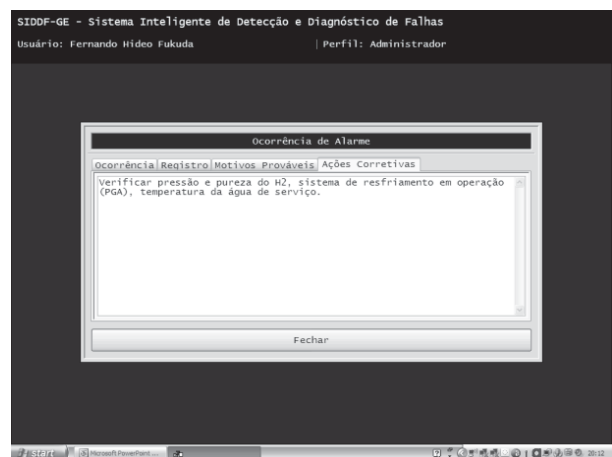
**Figura 13. Data, horário e responsável**

Na Figura 14 é apresentado o resultado referente aos motivos prováveis da ocorrência da Falha.



**Figura 14. Motivos Prováveis**

Finalmente, na Figura 15 é apresentado, visando apoiar a decisão do operador, um conjunto de ações corretivas que objetivam o retorno da operação normal do gerador.



**Figura 15. Ações Corretivas**

## 5. CONCLUSÕES

Como produto final desta investigação, foi disponibilizado para usina um Sistema Inteligente que realiza Detecção e o Diagnóstico das 27 Falhas por Gerador, as quais os quatro geradores da UTE Norte Fluminense podem ser afetados. Os resultados alcançados mostram termos atingido o objetivo principal da pesquisa.

O sistema proposto exibe uma lista da ocorrência de Alarmes, explicando características relativas a data e hora da ocorrência do alarme, descrição do alarme, código do equipamento, descrição do equipamento, descrição da falha e ações corretivas a serem tomadas visando o retorno a operação normal de funcionamento e em última análise apoiando a decisão do operador.

O sistema completo foi concebido usando-se, como base, um gerenciador de banco de dados relacional, baseado na linguagem padrão SQL desenvolvida pela IBM.

O sistema especialista completo, que envolve a detecção e o diagnóstico das falhas, desenvolvido nesse projeto de pesquisa, foi implementado em linguagem C++.

As principais vantagens em se implementar aplicações na linguagem C++ é o alto desempenho do código executável e a portabilidade.

Assim, caso haja necessidade da usina migrar para outra plataforma suportada, as mudanças no código da aplicação serão mínimas ou inexistentes.

Como o mapeamento das falhas, fornecido pelos especialistas da usina obedece a uma relação de um padrão de estados de alarmes para cada falha detectada e, além disso, baseado na característica do próprio SE, que infere a falha a partir deste padrão, não há erro na detecção das falhas.

Não havendo erro de detecção, o SE completo só depende da qualidade das explicações inseridas no sistema. Como pode ser constatado pela Figura 7 é possível registrar, consultar, alterar e excluir a ocorrência tornando as ações corretivas cada vez mais refinadas e seguras.

É importante salientar que foram realizados muitos testes usando-se RNAs e os bons resultados apresentados servem como base de comparação com os obtidos pelo SE. Caso fossem incluídas no produto final, seria necessário que além da RNA fosse incorporado um SE, constituindo assim dois módulos distintos e apre-

sentando maior dificuldade na manutenção do sistema completo.

Buscou-se então, com o SE especificado ao longo da pesquisa, uma melhor opção de desenvolvimento, mais integrada e eficiente, isto é otimizada, envolvendo a detecção e o diagnóstico da falha em um único SE.

Finalmente é preciso ressaltar o apoio e a total participação dos profissionais da UTENF no sentido de disponibilizar os dados necessários e participar das alterações realizadas ao longo da pesquisa e desenvolvimento com vista a converter esse trabalho de pesquisa e desenvolvimento em uma ferramenta útil ao dia a dia dos engenheiros e operadores da usina Termoelétrica.

## 6. AGRADECIMENTOS

Os autores agradecem as contribuições de Mauricio José Ferrari Rey, da UERJ, Geraldo Braga e Jaime Oliveira, da UTE Norte Fluminense recebidas durante o desenvolvimento do projeto e na elaboração deste artigo.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARBIERE, C. Modelagem de Dados, Livraria e Editora Infobook S. A., 1994.

BIONDI, L. N., COELHO P. H., GOMES E. G., MELLO, J. C., RIBEIRO, A. M. Identificação de Falhas em Subestação: Um Sistema de Apoio à Decisão Baseado em Redes Neurais, Congresso Nacional de Tecnologia da Informação e Comunicação, 2003, Bahia. SUCESU-NACIONAL. Pará: IDETI Eventos em Tecnologia da Informação, 2003.

BIONDI, L. N., COELHO, P. H. G., CHIGANER, L., ANGULO, M. L., MONTEIRO, L. A. P. Previsão da Vazão Afluente Mensal do Rio São Francisco usando-se Redes de Elman, Engenista (UFF), Niteroi, v. 07, n. 01, p. 63-72, 2005, ISSN: 14157314.

BIONDI, L. N., COELHO, P. H., RODRIGUES, M. A., ALVES, M. H. Sistema Inteligente

na Identificação e Diagnóstico de Defeitos em Subestação, Eletricidade Moderna, Rio de Janeiro, V. EM-367, N° OUT-2004, pp. 194-205, 2004, ISSN/ISBN: 01002104.

BIONDI, L. N., LINS, M. E., MOREIRA A. C. Utilização de Redes Neurais para Análise Condicionada de Demanda de Emergência Elétrica, 11° CONGRESSO DA APDIO, 2004, Porto - Portugal. IO - 2004 - 11° Congresso da APDIO. Porto: Faculdade de Engenharia da Universidade do Porto, 2004. V. 1, p. 72-72.

BRAGA, A. P., CARVALHO, A. C. P. L. F., LUDERMIR, T. B. Redes Neurais Artificiais: teoria e aplicações, Livros Técnicos e Científicos - LTC, 2000.

CIHAN, H. (EDT) DAGLI & ANNA L. (EDT) BUCZAK & DAVID L. (EDT) ENKE & MARK (EDT) EMBRECHTS & OKAN (EDT) ER-SOY Intelligent Engineering Systems Through Artificial Neural Networks - Amer Society of Mechanical, 2007

DAVIS, T. Protection of Industrial Power Systems, Newnes Elsevier Ltda, 2<sup>nd</sup> Edition, 2006.

ELMASRI, R., NAVATHE, S. B., Sistemas de Banco de Dados, Pearson, 2005.

GIARRATANO, J. C., RILEY, G. D. Expert Systems: Principals and Programming, 4<sup>th</sup> Edition, Thomson Course Technology, 2005.

HAYKIN, S. Neural Networks a Comprehensive Foundation, Macmillan College Publishing Company, 1994.

HEWITSON, L., BROWN, M., BALAKRISHNAN, R. Power System Protection, Newnes Elsevier Ltda, 2004

KHAN, S., SHEBA, K. AHAMED, G. Industrial Power systems, CRC press, 2007.

MERCEDE, F. Fault Calculations of Industrial / Commercial Power Systems, Institute of Electrical and Electronic Engineers Publisher, 1993.



RANGANATHAN, A. (2004), The Levenberg-Marquardt Algorithm, [Online] Disponível: <http://www.cc.gatech.edu/people/home/ananth/lmtut.pdf>.

RIEDMILLER, M. and BRAUN, H. A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP algorithm, Proceedings of the IEEE International Conference on Neural Networks, San Francisco, 1993, pp. 234-241.

ROWEIS, S. (2003), Levenberg Marquardt Optimization, [Online] Disponível: <http://www.cs.toronto.edu/~roweis/notes/lm.pdf>.

RUSSELL, S., NORVIG, P. Artificial Intelligence: A Modern Approach, Prentice Hall, 2003

SILBERSCHATZ, A., KORTH, H. F., SUDARSHAN, S. Sistema de Banco de Dados, Makron Books, 1997.

ZURADA, J. M. Introduction to Artificial Neural Systems, West Publishing Company, 1992.