

Universidade Federal do Maranhão  
Centro de Ciências Exatas e Tecnologia  
Programa de Pós-Graduação em Engenharia de  
Eletricidade

---

*Análise de Ocorrências em Transformadores  
do SDEE usando Redes Neurais Artificiais  
MLP*

---

Diogo Luís Figueiredo Nina

São Luís  
2012

Universidade Federal do Maranhão  
Centro de Ciências Exatas e Tecnologia  
Programa de Pós-Graduação em Engenharia de  
Eletricidade

---

*Análise de Ocorrências em Transformadores  
do SDEE usando Redes Neurais Artificiais  
MLP*

---

**Diogo Luís Figueiredo Nina**

Dissertação apresentada ao Programa de Pós-Graduação  
em Engenharia de Eletricidade da UFMA como parte dos  
requisitos necessários para obtenção do grau de Mestre  
em Engenharia Elétrica.

**São Luís  
2012**

Nina, Diogo Luís Figueiredo

Análise de Ocorrências em Transformadores do SDEE usando Redes Neurais Artificiais MLP / Diogo Luís Figueiredo Nina. - São Luís, 2012.

80f.:il.

Dissertação (Mestrado em Engenharia de Eletricidade) - Centro de Ciências Exatas e Tecnologia, Universidade Federal do Maranhão, 2012.

1. Redes Neurais Artificiais 2.IED 3.SCADA 4. Tomada de Decisão  
5. Transformadores.

CDU 62-52

**REDES NEURAIAS ARTIFICIAIS MLP BASEADAS EM IED E  
SCADA PARA DETECÇÃO DE OCORRÊNCIAS EM  
TRANSFORMADORES DO SDEE**

**Diogo Luis Figueiredo Nina**

Dissertação aprovada em 03 de outubro de 2012.

  
Prof. João Viana da Fonseca Neto, Dr.  
(Orientador)

  
Prof. Roberto Célio Limão de Oliveira, Dr.  
(Membro da Banca Examinadora)

  
Prof. Aristófares Corrêa Silva, Dr.  
(Membro da Banca Examinadora)

“Confie no Senhor de todo o seu coração e não se apóie em seu próprio entendimento.”

*Provérbios 3:5*

*Dedico esta dissertação aqueles que acreditam na beleza da união  
entre a teoria e a prática da Engenharia*

# Agradecimentos

---

Agradecer é reconhecer que nossas conquistas não são unicamente nossas.

Por isso registro minha gratidão aos professores do mestrado: à Professora Alcionie Miranda que forneceu todo o suporte necessário como especialista nas ferramentas utilizadas nesta dissertação. Ao professor João Viana meu mais profundo agradecimento por toda paciência dispensada, pela completa disponibilidade e pela compreensão nos momentos em que falhamos; pela genialidade e criatividade sem fim, sempre enriquecendo o projeto.

Aos meus amigos do mestrado, em especial ao Ronnie Loureiro, amigo profissional e pessoal, com o qual abracei o mestrado e com quem pude aprender muito: um engenheiro completo, com sólida teoria e equivalente prática.

À minha amada família que sempre me apoiou e sem a qual eu não seria capaz de chegar até aqui. Em especial à minha mãe que me ensinou a ser inconformado, a perseverar para ser o melhor que posso, a ir mais longe.

À minha companheira de vida, Fernanda, que não apenas torce pelo meu sucesso, mas oferece todo o suporte necessário para juntos alcançarmos novas conquistas a cada dia.

A Deus, o sentido de tudo e Aquele que colocou cada uma dessas queridas pessoas no meu caminho.

## RESUMO

A operação e manutenção do sistema elétrico requerem atenção, diagnósticos precisos em caso de falhas e agilidade na recomposição do sistema. Por outro lado, sistemas elétricos têm um elevado risco, onde cada manobra precisa ser cuidadosamente planejada e executada, pois erros podem ser fatais. A boa operação e manutenção do sistema elétrico consistem em encontrar o ponto de equilíbrio entre esses dois extremos, atuando de forma cautelosa, porém ágil. Com esse intuito, propomos o desenvolvimento de um sistema inteligente dotado da capacidade de detectar padrões anormais no sinal elétrico, fornecendo apoio à decisão na operação em tempo real do SDEE, a partir da análise das correntes primárias e secundárias de transformadores de força de subestações de energia elétrica, incluindo aprendizado a cada nova informação integrada ao sistema. O desafio deste estudo é pesquisar e desenvolver um método baseado em RNA para classificação de padrões e apoio à decisão, visando a detecção e/ou recuperação de falhas. O método diferencia perturbações que culminarão em uma falta de perturbações geradas por transitórios na rede elétrica (por exemplo o afundamento de tensão gerado pela partida de uma máquina). Um sistema supervisor SCADA foi desenvolvido para hospedar o código de implementação da RNA, além de fornecer uma interface para o Operador, gerando alarmes visuais e sonoros e mensagens orientando a retomada do sistema. O método proposto foi avaliado utilizando-se dados reais coletados diretamente de relés digitais de proteção de transformadores de subestações do sistema da CEMAR, obtendo-se excelentes resultados. A RNA desenvolvida neste estudo apresentou desempenho satisfatório na classificação dos sinais a ela apresentados, detectando corretamente as faltas.

**Palavras-Chave:** Automação de Subestações, Faltas em Subestações, SDEE, Rede Neural Artificial, Sistema de apoio à decisão, Sistema SCADA.

## ABSTRACT

Power system operation and maintenance require attention, precise diagnostics on failure and agility on system recovery. On the other hand, power systems involve high risks, where each operation needs to be carefully planned and executed, once errors can be fatal. Power system satisfactory operation and maintenance consist on finding equilibrium between these extremes, acting on a cautious, but agile, way. For this purpose, we propose the development of an intelligent system with the ability of detecting abnormal patterns on the electrical signal, providing support for decisions on Power Distribution System real time operation, from the analysis of power substation transformers primary and secondary currents, including learning at each new information acquired by the system. The challenge of this study is to research and develop a method based on ANN for classifying patterns and providing support for decisions, aiming fault detection and/or fault recovery. The method differentiates disturbances that will lead to faults from disturbances generated by transients on power system (for example an undervoltage caused by powering on an engine). A SCADA supervisory system was developed to contain ANN implementation code and also to provide an interface for Operators, generating visual and sound alarms and messages guiding system recovery. The proposed method was evaluated using real data collected from transformers protection digital relays of CEMAR system substations, achieving excellent results. The ANN developed on this study presented satisfactory performance classifying signals and detecting faults properly.

**Keywords:** Substations Automation, Faults on Substations, SDEE, Artificial Neural Network, Decision Support System, SCADA System.

# Lista de Tabelas

2.1	Funções de Proteção . . . . .	24
3.1	Especificações da Rede Neural . . . . .	39
4.1	Erros de Época . . . . .	58
4.2	Resultados dos Testes da Rede Neural . . . . .	62

# Lista de Figuras

2.1	Esquema de Unidade McCullock - Pitts. Fonte: Carvalho (2010).	13
2.2	Vista Geral 1 da Subestação São Francisco (CEMAR).	18
2.3	Vista Geral 2 da Subestação São Francisco (CEMAR).	19
2.4	Visão Geral dos Transformadores da Subestação São Francisco (CEMAR).	21
2.5	Disjuntores da Subestação São Francisco (CEMAR).	22
2.6	Relé SEL 351-A em operação na subestação São Francisco (CEMAR).	25
2.7	Relé SEL 387-A para proteção do Transformador 1 da Subestação Renascença (CEMAR).	25
2.8	Esquema de proteção do IED de um Transformador de Força de Subestação. Fonte: Loureiro (2012).	26
2.9	Diagrama Hierárquico da Digitalização da Subestação.	28
3.1	Arquitetura do Sistema proposto.	34
3.2	Diagrama Unifilar da Subestação Renascença (CEMAR), com destaque para o transformador analisado neste estudo.	36
3.3	Oscilografia de um Evento de Falta.	38
4.1	Implementação da RNA, sendo possível realizar as Fases de Treinamento e Teste.	47
4.2	Resultado do Treinamento da Rede Neural.	47
4.3	Erros Quadráticos calculados durante o Processo de Treinamento da RNA.	48
4.4	Resultado do Teste da Rede Neural.	49
4.5	Oscilografias de Relés.	50
4.6	Função Radial Gaussiana.	50

4.7	Implementação do SAD no SCADA. . . . .	51
4.8	Implementação de Banco de Dados de Ocorrências no SCADA. . .	52
4.9	Etapa de Treinamento da Rede Neural. . . . .	55
4.10	Erro da 1ª Época obtido durante a Fase de Treinamento da Rede Neural. . . . .	56
4.11	Erro da 14ª Época obtido durante a Fase de Treinamento da Rede Neural. . . . .	56
4.12	Teste de Evento do Tipo NORMAL. . . . .	60
4.13	Teste de Evento do Tipo PSF. . . . .	61
4.14	Teste de Evento do Tipo PCF. . . . .	62



# Lista de Abreviaturas e Siglas

CEMAR	Companhia Energética do Maranhão
COS	Centro de Operação do Sistema
DNP	Distributed Network Protocol
DEC	Duração Equivalente da Falta por Consumidor
ER	<i>Event Report</i> (Relatório de Eventos)
FEC	Frequência Equivalente da Falta por Consumidor
GPRS	<i>General Packet Radio Services</i> (Serviços de Rádio de Pacote Geral)
$IA_{W1}$	Corrente da Fase A do Enrolamento Primário do Transformador de Força
$IB_{W1}$	Corrente da Fase B do Enrolamento Primário do Transformador de Força
$IC_{W1}$	Corrente da Fase C do Enrolamento Primário do Transformador de Força
$IA_{W2}$	Corrente da Fase A do Enrolamento Secundário do Transformador de Força
$IB_{W2}$	Corrente da Fase B do Enrolamento Secundário do Transformador de Força
$IC_{W2}$	Corrente da Fase C do Enrolamento Secundário do Transformador de Força
IED	<i>Intelligent Electronic Device</i> (Equipamento Eletrônico Inteligente)
IHM	Interface Homem-Máquina
PLC	<i>Programmable Logic Controller</i> (Controlador Lógico Programável)
RNA	Redes Neurais Artificiais
SAD	Sistema de Apoio à Decisão
SED	Sistema de Estados Discretos
SCADA	<i>Supervisory Control and Data Acquisition</i> (Controle Supervisório e Aquisição de Dados)
SDEE	Sistema de Distribuição de Energia Elétrica
SE	Sistema Especialista
SP	Sistema de Proteção
SVC	Sistema de Variáveis Contínuas
UTR	Unidade Terminal Remota
51N2T	<i>Winding 2 phase inverse-time overcurrent element timed out</i> Atuação do elemento sobrecorrente residual - Enrolamento 2 (curva tempo-inverso)

# Sumário

<b>Lista de Tabelas</b>	<b>1</b>
<b>Lista de Figuras</b>	<b>3</b>
<b>Lista de Abreviaturas e Siglas</b>	<b>5</b>
<b>1 Introdução</b>	<b>8</b>
1.1 Objetivos . . . . .	9
1.1.1 Objetivo Geral . . . . .	9
1.1.2 Objetivos Específicos . . . . .	9
1.2 Motivação . . . . .	9
1.3 Contribuições . . . . .	9
1.4 Organização . . . . .	10
<b>2 Fundamentação Teórica</b>	<b>12</b>
2.1 Teoria de Redes Neurais Artificiais . . . . .	12
2.1.1 Função de Ativação . . . . .	13
2.1.2 Algoritmo de Aprendizado . . . . .	14
2.1.3 Critério de Parada de Treinamento . . . . .	15
2.2 Treinamento da RNA . . . . .	16
2.3 Sistemas de Apoio à Decisão . . . . .	16
2.4 Subestações de Distribuição de Energia Elétrica . . . . .	17
2.4.1 Principais Equipamentos e Funções . . . . .	19
2.4.2 Transformador de Potência . . . . .	19
2.4.3 Equipamentos de Manobra . . . . .	21
2.4.4 Relés de Proteção . . . . .	22

---

2.4.5	Equipamentos Automatizados do SDEE . . . . .	26
2.5	RNA e Sistema de Apoio à Decisão . . . . .	28
2.6	Conclusões e Comentários . . . . .	31
<b>3</b>	<b>RNA para Diagnóstico de Faltas</b>	<b>32</b>
3.1	Formulação do Problema . . . . .	33
3.2	Base de Dados . . . . .	35
3.3	Projeto da RNA . . . . .	38
3.3.1	Configuração da RNA . . . . .	38
3.3.2	Estratégia de Treinamento da MLP . . . . .	40
3.3.3	Critério de Parada de Treinamento . . . . .	43
3.4	Conclusões e Comentários . . . . .	43
<b>4</b>	<b>Experimentos de Campo e Computacionais</b>	<b>45</b>
4.1	Desenvolvimento do Sistema SCADA . . . . .	46
4.2	Implementação da Rede MLP . . . . .	52
4.2.1	Quantidade de Neurônios da Camada de Saída . . . . .	53
4.2.2	Treinamento <i>Backpropagation</i> . . . . .	53
4.2.3	Funções de Ativação . . . . .	53
4.2.4	Validação do Treinamento . . . . .	54
4.3	Análise de Desempenho da MLP . . . . .	59
4.3.1	Etapa de Testes . . . . .	59
4.4	Conclusões e Comentários . . . . .	62
<b>5</b>	<b>Conclusões</b>	<b>64</b>
5.1	Trabalhos Futuros . . . . .	64
<b>A</b>	<b>Código desenvolvido no <i>software</i> Elipse E3 Power®</b>	<b>66</b>
	<b>Referências Bibliográficas</b>	<b>77</b>

## CAPÍTULO 1

# Introdução

---

O serviço de distribuição de energia elétrica no Brasil consiste em uma concessão de serviço público regulado por uma Agência Reguladora - ANEEL (Agência Nacional de Energia Elétrica). Esta, por sua vez, é responsável por regular a legislação do setor elétrico e fiscalizar, por meio de indicadores de qualidade, o desempenho das concessionárias de energia.

Com o intuito de melhorar a qualidade na distribuição de energia, as concessionárias têm evoluído no sentido de automatizarem suas plantas, monitorando as variáveis do seu sistema, evitando ou minimizando interrupções e diminuindo o tempo de restabelecimento após falta. A digitalização de uma subestação busca tornar o sistema mais confiável e flexível; permite também o estudo do sistema por meio das variáveis que vão sendo armazenadas diariamente em banco de dados. As falhas do sistema também são registradas, o que permite diagnosticar a causa de uma falta, evitando que o mesmo tipo de defeito ocorra novamente. Todas estas ações contribuem para a melhoria dos indicadores de qualidade, o que reflete, em última análise, a melhoria do sistema de distribuição de energia elétrica.

Nesta pesquisa desenvolve-se metodologias de implementação e treinamento de Redes Neurais Artificiais com o objetivo de detecção e classificação de padrões na análise de perturbações do sistema elétrico.

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

Desenvolver um sistema baseado em redes neurais artificiais integradas a um sistema SCADA para detectar e classificar as perturbações em um sistema de distribuição de energia elétrica.

### 1.1.2 Objetivos Específicos

- Desenvolver uma rede neural do tipo perceptron multicamada para detecção de anomalias em transformadores,
- Desenvolver um software para treinamento *off-line* e processamento *on-line* da rede neural,
- Integrar o sistema SCADA com a rede neural artificial,
- Minimizar o tempo de análise de ocorrências,
- Viabilizar para a operação e manutenção uma ferramenta de apoio à tomada de decisão.

## 1.2 Motivação

A motivação desta pesquisa está centrada no fato de viabilizar um sistema inteligente que ofereça apoio à tomada de decisão de operadores do setor elétrico, provocando impactos significativos nos indicadores operacionais DEC (Duração Equivalente da Falta por Consumidor) e FEC (Frequência Equivalente da Falta por Consumidor). Ter estes indicadores sob controle aumenta a confiabilidade do SDEE, gerando satisfação ao cliente consumidor deste serviço essencial e evitando para a concessionária penalidades por parte do órgão regulador.

## 1.3 Contribuições

- Aplicação prática da teoria de Redes Neurais a um problema de detecção e recuperação de faltas no SDEE, auxiliando a tomada de decisão.

- A aplicação desenvolvida reduz drasticamente o tempo para análise de ocorrências e restabelecimento do sistema após faltas. Sem a solução proposta nessa pesquisa, o tempo gasto nesta tarefa pode chegar a algumas horas.
- Intercâmbio de dados de forma transparente para diversas outras aplicações.
- Direcionamento da ação da Manutenção em caso de faltas, agilizando o restabelecimento do sistema.
- Geração de banco de dados histórico como fonte de conhecimento para construção de estatísticas do setor elétrico.
- Direcionamento dos investimentos da empresa para áreas de maior incidência de ocorrências, conforme estatísticas obtidas das classificações realizadas pela RNA.
- Aumento da confiabilidade do SDEE através da constante monitoração e análise realizada pela RNA integrada ao sistema SCADA.
- Aproveitamento da instrumentação já existente na planta da concessionária, minimizando assim os custos de implementação desta metodologia.
- Redução dos indicadores de DEC e FEC, uma vez que a solução proposta tem como objetivo evitar faltas ou agilizar a retomada do sistema após faltas.

## 1.4 Organização

A dissertação está organizada em três partes: a primeira parte é constituída dos conceitos teóricos e estado da arte; a segunda parte apresenta o modelo neuronal para detecção de faltas e na terceira parte apresenta-se o resultados de desempenho da rede.

No Capítulo 2 descreve-se a fundamentação teórica de redes neurais artificiais (RNA). Define-se com detalhes os elementos essenciais que constituem uma RNA. Também conceitua-se Sistemas de Apoio à Decisão (SAD), além de fornecer esclarecimentos específicos da área de automação de sistemas de distribuição de energia elétrica. Ainda neste capítulo apresenta-se uma seção sobre o estado da

arte e outros projetos semelhantes que vêm sendo realizados utilizando métodos baseados em RNA para detecção de falhas.

No Capítulo 3 procede-se à caracterização do problema e à descrição do projeto da RNA para diagnóstico de faltas. Indica-se também como foi gerada a base de dados utilizada neste projeto e alguns conceitos básicos sobre oscilografia de eventos são definidos.

Os resultados obtidos através dos experimentos de campo e computacionais são discutidos no Capítulo 4; descreve-se de forma detalhada como foram criados os vetores de eventos para as etapas de treinamento e de teste, além da escolha do critério de parada de treinamento. Discute-se ainda como a escolha dos parâmetros da RNA pode contribuir para o sucesso da etapa de aprendizagem. A aplicação SCADA desenvolvida e suas funcionalidades são ilustradas e por fim, analisa-se o desempenho da rede.

O Capítulo 5 traz as conclusões assimiladas através deste trabalho. Vislumbra-se também perspectivas de propostas futuras, em face dos resultados já conquistados nesta pesquisa.

# Fundamentação Teórica

---

Neste capítulo são apresentados os fundamentos teóricos de Redes Neurais Artificiais necessários para o desenvolvimento da dissertação. Aborda-se também a teoria de Sistemas de Apoio à Decisão (SAD). Entende-se ser necessário definir conceitos específicos da área de automação do setor elétrico para contextualizar a forma de aplicação desta dissertação. No final deste capítulo, cita-se na Seção RNA e Sistema de Apoio à Decisão pesquisas e projetos semelhantes que utilizam RNA para detecção de falhas.

## 2.1 Teoria de Redes Neurais Artificiais

A rede neural artificial (RNA) é um modelo matemático ou computacional baseado nas redes neurais biológicas. Consiste de um grupo interligado de neurônios artificiais que processa informação utilizando uma abordagem computacional (Horie *et al.* (2008)).

Basicamente, Redes Neurais Artificiais trabalham por meio de reconhecimento de padrões. A idéia principal da RNA é desenvolver um modelo constituído de várias unidades de processamento simples, chamadas **neurônios**, com múltiplas conexões entre eles, chamadas **sinapses** ou **pesos sinápticos**. Uma RNA típica é composta de 3 camadas: camada de entrada, camada intermediária (ou escondida) e camada de saída. Entre diversas topologias possíveis, nesta dissertação será utilizada uma ***RNA multicamadas feed-forward*** com três camadas, sendo uma **camada de entrada**, uma **camada de saída** e uma **camada intermediária**.

Para o desenvolvimento desta RNA, é necessário definir a quantidade de nós da camada de entrada, a quantidade de camadas intermediárias (e a quantidade de neurônios em cada uma dessas camadas) e a quantidade de neurônios na camada de saída. A quantidade de nós da camada de entrada corresponde à quantidade de variáveis utilizadas para alimentar a RNA. Ao desenvolver uma RNA multicamadas é necessário assegurar a capacidade de generalização: a habilidade de classificar padrões de um conjunto de teste não utilizado no treinamento da rede. Isso significa que, embora a RNA aprenda a partir de um conjunto específico (conjunto de treinamento), ela fornecerá resultados satisfatórios para um conjunto de teste contendo dados diferentes daqueles (Braga *et al.* (1998)).

Outros parâmetros da RNA também são relevantes: a função de ativação dos neurônios das camadas intermediária e de saída, o algoritmo de aprendizado, o método de padronização, o conjunto de treinamento e o conjunto de teste, o critério de parada do treinamento e a avaliação da performance da rede. Estes parâmetros são definidos a seguir.

### 2.1.1 Função de Ativação

A Figura 2.1 ilustra o modelo de uma unidade de processamento (neurônio) proposto por McCullock e Pitts em 1943 (Carvalho(2010)). Conforme o modelo, os sinais de entrada ponderados são somados e então a função de ativação é aplicada. O resultado é percebido portanto na saída do neurônio.

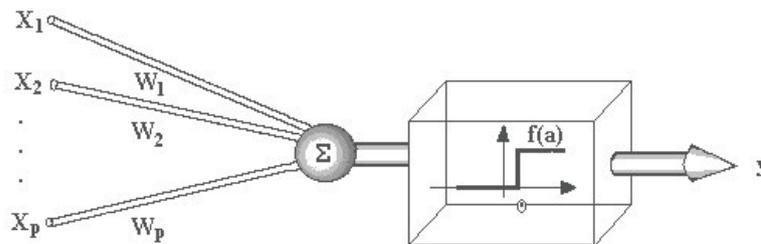


Figura 2.1: Esquema de Unidade McCullock - Pitts. Fonte: Carvalho (2010).

A função de ativação, também chamada de função de transferência, é uma função matemática que, aplicada à combinação linear entre as variáveis de entrada e pesos que chegam a determinado neurônio, retorna o seu valor de saída (Santos *et al.* (2005)). Existem diversas funções matemáticas que são utilizadas como

função de ativação; as mais comumente usadas são: função logística e função tangente hiperbólica, sendo que esta última foi escolhida para utilização nesta dissertação.

### 2.1.2 Algoritmo de Aprendizado

De acordo com Horie *et al.* (2008), na maioria dos casos, uma RNA é um sistema adaptativo que muda sua estrutura interna ou externa baseada na informação que flui através da rede, durante a fase da aprendizagem. Para o treinamento da RNA deve-se optar por um método para a correção de erros: o chamado modo normal (ou *online*) ou o modo batelada (ou *batch*). Enquanto no primeiro este erro é calculado a cada ciclo ou a cada amostra, no segundo o erro é calculado após a apresentação de todo o conjunto de treinamento à rede. Ou seja, no modo *online*, a correção de erro acontece a cada vez que um elemento do conjunto é apresentado à rede neural.

O algoritmo escolhido neste trabalho foi o *back-propagation*, um método de aprendizado por correção de erro, optando-se pelo modo *online*. A RNA opera em 2 etapas: primeiramente, a camada de entrada recebe um padrão, o qual é propagado, camada por camada, até que a resposta seja produzida na camada de saída. Na segunda etapa, a saída obtida é comparada à saída desejada e o erro obtido é propagado a partir da camada de saída em direção à camada de entrada. Os pesos das conexões das unidades da camada de saída e das unidades da camada intermediária são modificados à medida que a retropropagação do erro ocorre (Carvalho (2010)).

O erro da rede neural pode ser calculado pela diferença entre a saída gerada pela rede e a saída desejada, tal que:

$$e(t) = d(t) - y(t) \quad (2.1)$$

sendo  $e(t)$  o sinal de erro da resposta atual da rede,  $d(t)$  é a saída desejada no instante  $t$  e  $y(t)$  é a saída real da rede no instante  $t$ . Durante o aprendizado, portanto, os erros são calculados repetidamente.

Quando tem-se mais de um neurônio na camada de saída deve-se analisar o

erro quadrático  $e_{quad}(t)$  tal que:

$$e_{quad}(t) = \sum_{i=1}^n e_i^2(t) \quad (2.2)$$

sendo  $n$  a quantidade de neurônios na camada de saída da rede e  $e_i(t)$  o sinal de erro da resposta atual do  $i$ -ésimo neurônio desta camada.

A Equação do backpropagation ou Regra Delta Generalizada fornece a atualização dos pesos das conexões da RNA e é exibida na Equação 2.3:

$$w_i(t+1) = w_i(t) + \eta \frac{\delta J}{\delta w} \quad (2.3)$$

sendo  $w_i(t+1)$  o valor de ajuste a ser acrescentado ao peso sináptico  $w_i(t)$ ,  $\eta$  é a taxa de aprendizado e  $J$  é a função custo a ser minimizada.

De acordo com Pagliosa (2003), a taxa de aprendizado  $\eta$  controla a estabilidade e velocidade de convergência da rede. Esta taxa varia de 0 a 1, sendo que quanto mais próxima de 0, maior a estabilidade da RNA após o processo de treinamento; por outro lado, quanto mais próxima de 1, maior a velocidade de convergência da RNA e o período de treinamento se torna menor. Tanto uma escolha quanto outra implicam ganhos e perdas; cabe encontrar o ponto ótimo conforme os objetivos e requisitos do projeto.

### 2.1.3 Critério de Parada de Treinamento

Outro importante parâmetro necessário no projeto de uma RNA é o critério de parada de treinamento que define o ponto onde o resultado obtido atende determinada regra suficientemente satisfatória para que o treinamento seja interrompido. A continuação do treinamento de forma indefinida não só implica em maior comprometimento do desempenho do sistema em termos de performance como, principalmente, recai em um problema específico quando trabalha-se com redes neurais: o sobreajuste da rede. Conforme Calôba (1995), o critério de parada de treinamento exige considerar a capacidade de generalização da rede. O critério adotado neste projeto é definido na seção 3.3.3.

## 2.2 Treinamento da RNA

No projeto de uma RNA deve-se dividir o conjunto de dados disponíveis em um conjunto de treinamento e outro conjunto de teste. Nesta pesquisa, esta divisão foi realizada gerando um conjunto de treinamento com 60% dos dados e um conjunto de teste contendo os 40% restantes. Diversas proporções foram testadas, porém os melhores resultados foram obtidos nesta configuração. A formação desses conjuntos é descrita em detalhes na Seção 3.3.2.

## 2.3 Sistemas de Apoio à Decisão

Tomada de decisão é um processo de escolha entre diversas alternativas de ação para que determinado objetivo seja atingido. Ou seja, quando mais de uma solução é possível, o método de tomada de decisão visa garantir a melhor escolha sob determinados níveis de exigência.

De acordo com Teixeira Junior e Pereira (2006), os sistemas de apoio à decisão (SAD) são sistemas computacionais que visam sistematizar e apoiar os processos decisórios das empresas. Seus componentes básicos são: interface com o usuário, banco de dados e banco de modelos.

Ainda segundo Teixeira Junior e Pereira (2006), os SAD visam fornecer apoio em decisões semi-estruturadas e desestruturadas e estão sob controle do tomador de decisões, além de disponibilizar um conjunto de ferramentas para aumentar a efetividade das decisões.

No processo de tomada de decisão, o grau de eficiência aumenta à medida que a escolha da alternativa oferece melhores resultados. Conforme Cardoso *et al.* (2009), as alternativas factíveis de atender o objetivo da decisão e, portanto, selecionadas para avaliação, serão comparadas em função de critérios e sob a influência de atributos. Ainda conforme esta fonte, a subjetividade está sempre presente nos processos de decisão e o problema fundamental da decisão Multi-critério é associar a relação de preferências (subjetivas) entre os vários critérios no processo de decisão.

No desenvolvimento de um SAD, pode-se lançar mão de algumas ferramentas como, por exemplo, sistemas especialistas (SE). Conforme Giboshi *et al.* (2006), estes constituem uma ferramenta que pode ser integrada em um SAD, uma vez que

têm a capacidade de armazenar e processar uma vasta quantidade de informações. Para o desenvolvimento de um sistema especialista, é necessário implementar uma solução cuja fonte de conhecimento é o especialista da área em questão. Contudo, transferir para o sistema o conhecimento acumulado ao longo de anos de experiência por profissionais não é uma tarefa simples; é necessário compreender as regras ou exigências impostas no processo de tomada de decisão, considerando que algumas regras tem um peso maior que outras, conforme a subjetividade do tomador da decisão. Traduzir esses conceitos para o sistema e garantir a máxima efetividade nas decisões é um desafio.

É importante ter em vista que um SE dificilmente atinge a capacidade cognitiva do especialista humano, porém na sua ausência é uma ferramenta válida para a resolução de problemas (Oliveira *et al.* (2010)).

Por fim, as RNAs são uma alternativa aos SEs pois as mesmas aprendem a partir dos dados e assim não precisam de todo o conhecimento do especialista humano.

## 2.4 Subestações de Distribuição de Energia Elétrica

O fornecimento de energia elétrica aos consumidores finais ocorre em níveis de tensões compatíveis com a tensão de alimentação dos equipamentos de escritório, domésticos e industriais. No intuito de viabilizar o fornecimento de energia ao usuário final, o nível de tensão varia desde a geração até chegar às Subestações de Distribuição de Energia Elétrica.

O roteamento, controle do fluxo de energia elétrica, bem como a obtenção desses variados níveis de tensão ocorrem em razão do emprego de subestações, que se constituem os elos entre os sistemas.

Uma subestação de distribuição de energia elétrica é um conjunto de equipamentos de manobra e/ou transformação e ainda eventualmente de compensação de reativos usado para dirigir o fluxo de energia em sistema de potência e possibilitar a sua diversificação por meio de rotas alternativas, possuindo dispositivos de proteção capazes de detectar os diferentes tipos de falhas que ocorrem no sistema e de isolar os trechos de defeito do restante do SDEE.

A subestação da CEMAR São Francisco é classificada da seguinte forma se-

gundo os tópicos anteriores: quanto à sua função é caracterizada como Subestação Transformadora Abaixadora, e quanto ao modo de instalação dos equipamentos em relação ao meio ambiente, é uma subestação externa ou ao tempo. Nas Figuras 2.2 e 2.3 apresenta-se a Vista Geral da Subestação São Francisco; este pátio de subestação é constituído de transformadores, disjuntores, capacitores, chaves e equipamentos auxiliares.



Figura 2.2: Vista Geral 1 da Subestação São Francisco (CEMAR).



Figura 2.3: Vista Geral 2 da Subestação São Francisco (CEMAR).

### 2.4.1 Principais Equipamentos e Funções

Pode-se agrupar os equipamentos de uma subestação da seguinte forma: equipamentos de transformação, equipamentos de manobra, equipamentos para compensação de reativos, equipamentos de proteção e outros equipamentos (auxiliares). Os equipamentos de transformação são classificados em transformadores de potência que são responsáveis pela transformação do nível de tensão propriamente dito e transformadores de instrumentação que auxiliam os medidores e relés de proteção na subestação.

### 2.4.2 Transformador de Potência

Sem os transformadores de potência seria praticamente impossível o aproveitamento econômico da energia elétrica, pois a partir deles foi possível a transmissão em tensões cada vez mais altas, possibilitando grandes economias nas linhas de transmissão em trechos cada vez mais longos.

Os transformadores das subestações de alta-tensão podem ser classificados de

acordo com suas funções:

- Transformadores elevadores: elevam a tensão de geração para a tensão de transmissão;
- Transformadores de interligação: interligam partes do sistema de transmissão;
- Transformadores abaixadores: abaixam a tensão de transmissão para a tensão de subtransmissão ou de distribuição.

A escolha entre transformadores trifásicos e bancos de unidades monofásicas depende de estudos técnico-econômicos, que devem considerar os fatores:

- Custos dos investimentos;
- Custos da energia não fornecida;
- Confiabilidade - necessidade de unidades reserva;
- Limitadores de transporte (peso e altura máxima);
- Limitadores de capacidade de fabricação.

Os transformadores de potência são classificados segundo o seu meio isolante, podendo ser a óleo mineral, a líquidos isolantes sintéticos pouco inflamáveis (silicone) e secos. O óleo mineral (derivado do petróleo) e os líquidos isolantes sintéticos usados em transformadores possuem duas funções principais: isolar, evitando a formação de arco entre dois condutores que apresentem uma diferença de potencial, e resfriar, dissipando o calor originado da operação do equipamento. Os transformadores secos utilizam o ar circulante como meio isolante e refrigerante, possuindo isolamento classe B, classe F ou classe H.

Na Figura 2.4 apresenta-se uma Visão Geral dos transformadores da Subestação São Francisco.



Figura 2.4: Visão Geral dos Transformadores da Subestação São Francisco (CE-MAR).

### 2.4.3 Equipamentos de Manobra

Os equipamentos de manobra das subestações são os disjuntores e chaves exercendo diversas funções dependendo da sua aplicação na subestação. Os detalhes desses componentes são abordados nos próximos parágrafos.

Os disjuntores são os mais eficientes e mais complexos aparelhos de manobra em uso de redes elétricas, destinados à operação em carga, podendo sua operação ser manual ou automática. Os disjuntores são os principais equipamentos de segurança, bem como os mais eficientes dispositivos de manobra em uso nas redes elétricas. Possuem capacidade de fechamento e abertura que deve atender a todos os pré-requisitos de manobra sob condições normais e anormais de operação.

Além dos estados estacionários (fechado e aberto), define-se ambos os estados transitórios da manobra de fechamento (ligamento) e da manobra de abertura (desligamento).

Quanto à manobra de fechamento, o disjuntor deve, no caso de curto-circuito, atingir corretamente sua posição de fechado e conduzir a corrente de curto-circuito. No caso de abertura, o disjuntor deve dominar todos os casos de manobra possíveis na rede na qual está instalado.

Na Figura 2.5 apresenta-se os disjuntores da Subestação São Francisco.



Figura 2.5: Disjuntores da Subestação São Francisco (CEMAR).

#### 2.4.4 Relés de Proteção

Os relés têm por finalidade proteger o sistema contra faltas, permitindo, através da atuação sobre disjuntores, o isolamento dos trechos de localização das faltas.

Os relés de proteção são os dispositivos lógicos do sistema elétrico, pois observam e confrontam os parâmetros de corrente e tensão, fornecidos pelos transdutores, com seu pré-ajuste. Caso ocorra uma anomalia ou falta no sistema, de maneira que a variável sensível do relé ultrapasse o seu ajuste, o mesmo atua instantaneamente ou de forma temporizada, de acordo com a necessidade. O relé envia, por exemplo, um sinal para abertura do disjuntor (disparo) para isolar

uma determinada falta, conservando a parte remanescente do sistema em operação normal, sem que os efeitos dessa falta prejudiquem sua normalidade.

A função de proteção é a aglomeração de atributos exercidos por um SP, para fins previamente estabelecidos e definidos, dentro de uma determinada categoria ou modalidade de atuação.

A descrição da função exercida pelos elementos, equipamentos e dispositivos do SP segue a padronização numérica de apresentação C37.2-1996 IEEE/ANSI (*Institute of Electrical and Electronics Engineers/American National Standards Institute*), *Standard Electrical Power System Device Function Numbers and Contact Designations*, conforme descrito na Tabela 2.1:

Tabela 2.1: Funções de Proteção

Função de Proteção	Modalidade de atuação (Relé/Dispositivo Eletrônico)	Descrição da função de proteção
21	Distância	Atua para defeitos em linhas de transmissão de alta tensão, quando a impedância, admitância ou a reatância observada, diminui ou aumenta além dos limites.
26	Dispositivo térmico	Atua quando a temperatura em transformadores excede o limite normal.
27	Subtensão	Atua quando a tensão elétrica diminui além de um valor ajustado.
46	Desbalanceamento de fase	Atua quando há desbalanceamento de corrente de fase.
50	Sobrecorrente instantânea	Atua em poucos ciclos quando o valor da corrente excede certo limite.
51	Sobrecorrente temporizada	Opera quando o valor da corrente exceder o ajuste após um tempo pré-determinado.
59	Sobretensão	Produz efeito quando a tensão excede o limite ajustado.
63	Pressão	Entra em função de defeitos internos do transformador e para baixos níveis de pressão em equipamentos encapsulados.
81	Frequência	Opera quando a frequência atinge certos limites.
87	Diferencial	Opera por comparação de corrente (diferencial de transformador, gerador, linha de transmissão e barra).
98	Oscilografia	Possibilita o diagnóstico, pós-perturbação, de ocorrência de defeitos no sistema elétrico.

Os relés utilizados na subestação São Francisco para proteção de disjuntores são os SEL 351-A da Schweitzer Engineering Laboratories, apresentados na Figura 2.6.



Figura 2.6: Relé SEL 351-A em operação na subestação São Francisco (CEMAR).

Na Figura 2.7 apresenta-se o Relé SEL 387-A para proteção do Transformador 1 da Subestação Renascença; este foi o IED escolhido para a implementação desta dissertação.



Figura 2.7: Relé SEL 387-A para proteção do Transformador 1 da Subestação Renascença (CEMAR).

A Figura 2.8 apresenta o esquema de proteções do IED de um transformador

de força de subestação, onde as funções do IED são agrupadas em SVC (Sistema de Variáveis Contínuas) e SED (Sistema de Estados Discretos).

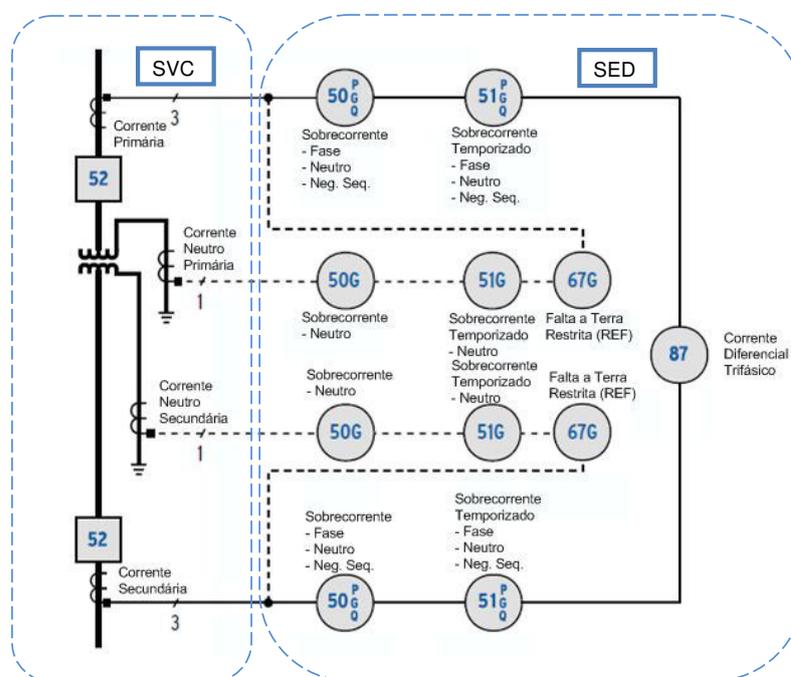


Figura 2.8: Esquema de proteção do IED de um Transformador de Força de Subestação. Fonte: Loureiro (2012).

### 2.4.5 Equipamentos Automatizados do SDEE

Neste item descreve-se de forma sistematizada a operação do processo desde os equipamentos básicos, UTRs, comunicação e supervisórios.

Os principais tópicos e tecnologias que estão relacionados com medições e monitoramento no desenvolvimento de um sistema de automação são:

1. Automação de subestações
  - IED - software e hardware de tempo real
  - UTR
  - Difusor Ótico
2. Tecnologia da Informação

- Redes Ethernet - GPRS

### 3. Falhas em Equipamentos/Dispositivos Elétricos

- Disjuntores
- Chaves
- Relés

Os equipamentos a serem automatizados em uma subestação são: disjuntores, religadores, transformadores, reguladores de tensão, bancos de capacitores etc. Para realizar a proteção desses equipamentos é necessária a instalação de IED's (Intelligent Electronic Devices - Equipamentos Eletrônicos Inteligentes). IED's são responsáveis por proteger o sistema desligando o trecho do circuito que se encontra sob condição de falta. Eles podem executar comandos sobre os equipamentos da subestação como, por exemplo, abertura de um disjuntor, acionamento da ventilação forçada do transformador etc. Também monitoram variáveis discretas, como estado do equipamento (disjuntor aberto ou fechado) e estado da ventilação forçada do transformador (ligada ou desligada). Além disso, IED's são capazes de monitorar variáveis analógicas - medições em geral (correntes dos alimentadores, tensões das barras, potências ativas, reativas e aparentes, etc).

Os níveis hierárquicos do sistema de automação são:

- Nível 1 - Relé digital (IED)
- Nível 2 - Unidade Terminal Remota (UTR)
- Nível 3 - *Front-end* de comunicação e SCADA

Portanto, todas as informações oriundas dos IED's das subestações são centralizadas em um equipamento de aquisição de dados similar ao PLC industrial. Na presente dissertação, denomina-se UTR e concentra toda a informação da subestação, reportando-a para o nível imediatamente superior. Na ponta do processo, temos o sistema SCADA, tecnologia que permite a um usuário coletar dados de um ou mais equipamentos remotos e enviar instruções de controle a estes mesmos equipamentos. Dessa forma, a UTR constitui o elo entre IED's e SCADA, processando comandos e instruções recebidas, bem como informando o estado de

todas as variáveis da subestação para o sistema supervisorio, conforme apresentando na Figura 2.9. Nela apresenta-se o processo de digitalização da subestação, salientando-se os principais dispositivos de *hardware/software* que compõem cada camada. No nível 1, tem-se o relés digitais (IEDs). No nível 2, a UTR local, sendo que o protocolo utilizado é o DNP 3.0. No Nível 3, a UTR do centro de operação do sistema (COS) recebe as informações da subestação e as envia para o supervisorio (SCADA) por meio do protocolo DNP 3.0, fazendo o papel de *front-end* de comunicação.

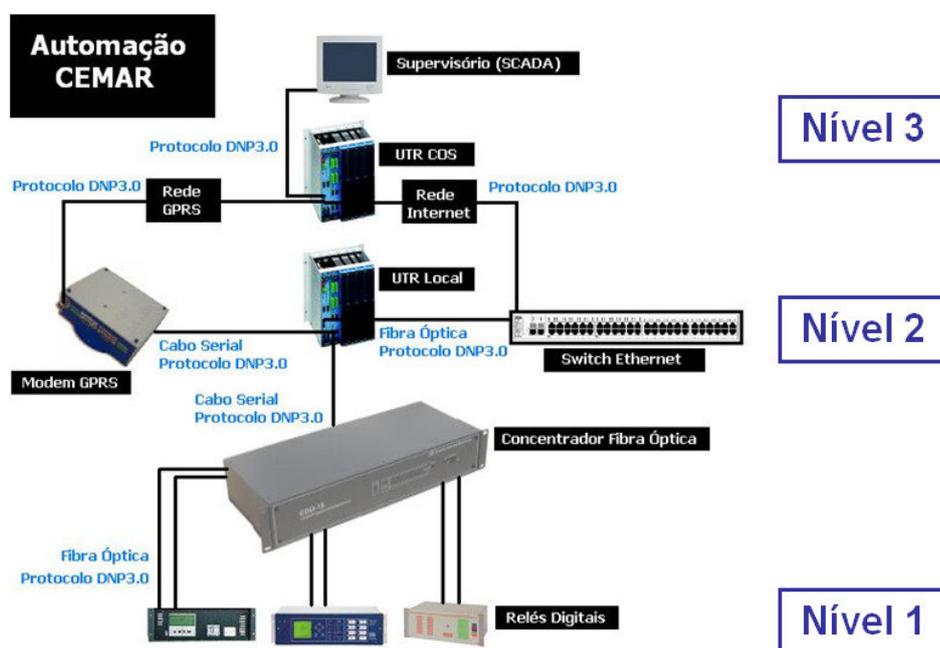


Figura 2.9: Diagrama Hierárquico da Digitalização da Subestação.

## 2.5 RNA e Sistema de Apoio à Decisão

Diversos trabalhos vêm sendo desenvolvidos utilizando-se métodos baseados em Redes Neurais para detecção de falhas e SAD. Trata-se de uma metodologia com as mais variadas aplicações, geralmente utilizada na detecção e classificação de padrões. Nesta seção descreve-se alguns dos muitos estudos desenvolvidos nesta área e os resultados alcançados até então.

Fernandes (2007) cita um sistema de detecção e isolamento de falhas desenvolvido utilizando RNA aplicado a um sistema de níveis ligado à uma rede industrial *Foundation Fieldbus*. O sistema foi dividido em duas partes: identificação neural e classificação/isolamento de falhas. As RNAs utilizadas são do tipo MLP (Perceptron de Múltiplas Camadas) e foram treinadas com o algoritmo *Backpropagation*. Também foi realizada a comparação entre as saídas geradas pela RNA e aquelas geradas por um modelo matemático, sendo que as primeiras apresentaram resultados mais satisfatórios.

Silva *et al.* (2007) apresentam um método de detecção e classificação de faltas em linhas de transmissão, a partir da análise de registros de oscilografia via RNA e transformada wavelet. Excelentes resultados foram obtidos e a principal contribuição do método utilizado foi a distinção de registros referentes à faltas de registros referentes a distúrbios relacionados à qualidade da energia elétrica tais como afundamentos de tensão e transitórios de chaveamento. O método ainda distingue manobras de manutenção e operação do sistema.

Tinós e Terra (2001) utilizam Redes Neurais para detecção e isolamento de faltas para robôs manipuladores. Duas redes são utilizadas: uma Perceptron Multicamadas para reproduzir o comportamento dinâmico do manipulador, gerando um vetor de resíduos que é classificado por uma RBF (Rede com Função de Base Radial). Dois métodos de treinamento foram empregados: um utilizando FS (*Forward Selection*) e outro utilizando o mapa auto-organizado de Kohonen. A RBF treinada pelo primeiro método apresentou menor erro para o conjunto de treinamento, já a treinada pelo segundo método mostrou-se mais flexível, apresentado os melhores resultados para o conjunto de teste de generalização.

Oleskovicz *et al.* (2003) adotam uma abordagem bastante próxima da que segue o presente estudo, aplicando RNA para detecção, classificação e localização de faltas em linhas de transmissão. São empregados os valores das tensões e correntes trifásicas de pré e pós-falta tanto para o processo de treinamento como para o de teste. Os resultados obtidos demonstraram elevada precisão e velocidade de resposta da RNA. O mesmo objetivo pode também ser percebido em Oliveira (2005).

Raybolt (2011) avalia uma técnica de localização de falta em linhas de transmissão em circuito duplo utilizando Redes Neurais Artificiais. A análise é feita

utilizando os dados de tensão e corrente oriundos dos dois terminais da linha de ambos os circuitos. Os testes realizados neste estudo mostram um sistema eficiente para classificação e localização de defeitos em linhas de transmissão.

Já Spatti (2011) utiliza sistemas especialistas híbridos com o objetivo de fornecer um sistema de detecção automática de curtos-circuitos. Os resultados demonstram grande potencialidade de aplicação em sistemas de distribuição e faz parte do objetivo deste estudo permitir o acompanhamento *online* de ocorrências de possíveis faltas e transitórios eletromagnéticos observados na rede primária de distribuição.

Em William *et al.* (2007), ilustra-se uma solução utilizando RNA para localização de faltas em sistemas de distribuição instalados a bordo de navios. O objetivo é aumentar a disponibilidade dos sistemas de distribuição elétrica a partir do isolamento das faltas utilizando para isso algoritmos de RNA implementados em relés de distância.

Em Coser *et al.* (2005), descreve-se um método baseado em redes neurais artificiais para localização de faltas em sistemas de distribuição de energia elétrica. Essa abordagem apresentou resultados promissores para valores de resistência de falta elevada, situação em que métodos analíticos geralmente falham.

Hang *et al.* (2008) apresentam uma nova abordagem baseada em redes neurais artificiais para otimização do cálculo de perda de energia na rede elétrica.

Em Stergiopoulos *et al.* (2003), três arquiteturas são propostas em um sistema de diagnóstico de faltas para redes de distribuição de energia elétrica: a primeira abordagem utiliza redes neurais artificiais; já a segunda usa lógica fuzzy, enquanto a terceira abordagem propõe um sistema híbrido combinando RNA e lógica fuzzy.

Em Salazar *et al.* (2006), um algoritmo baseado na teoria de redes neurais artificiais é proposto na abordagem do problema de reconfiguração de alimentadores em sistemas de distribuição para redução das perdas de energia. Essa metodologia apresentou bons resultados quando aplicada a dois sistemas elétricos.

Em Butler e Momoh (1993), utiliza-se uma abordagem de rede neural baseada em algoritmo de clusterização para detecção e classificação de faltas em linhas de um sistema de distribuição de energia. Como dados de entrada da rede neural, discute-se o uso de um conjunto reduzido gerado pela extração de características relevantes de formas de onda de corrente.

## 2.6 Conclusões e Comentários

Neste capítulo apresentou-se os principais fundamentos para o desenvolvimento do sistema para apoio a decisão que são redes neurais, teoria de SAD e automação de sistemas de distribuição.

Os conceitos de redes neurais artificiais enfatizaram o perceptron multicamadas, função de ativação e a regra delta para o processo de treinamento. Abordou-se os conceitos básicos de sistemas de apoio à decisão.

Apresentou-se a arquitetura de sistemas de automação da distribuição da empresa da região. Neste item salientou-se a importância dos IEDs como sistema de aquisição de dados do transformador e demais equipamentos da subestação.

Na Seção RNA e Sistema de Apoio à Decisão apresentou-se uma revisão bibliográfica que fornece subsídios para uma justificativa tecnológica quanto à relevância do desenvolvimento desta pesquisa. Observa-se que a rede neuronal já é amplamente aplicada no setor elétrico, contudo, a maior parte das aplicações estão orientadas para os sistemas de transmissão e industriais, enquanto este estudo enfoca a aplicação em sistemas de distribuição.

# RNA para Diagnóstico de Falhas

---

Este capítulo descreve a caracterização do problema observado. Em seguida, detalha-se ainda em linhas gerais o projeto da RNA para diagnóstico de falhas, indicando como foi gerada a base de dados utilizada. Alguns conceitos sobre registros de oscilografias também são definidos, uma vez que a interpretação do comportamento dos sinais observados é fundamental para compreender e alcançar a proposta deste estudo. Por fim, descreve-se de forma detalhada a criação dos vetores de eventos para as etapas de treinamento e de teste da rede e estipula-se o critério de parada de treinamento.

Conforme Fernandes (2007), para detectar e diagnosticar falhas eficientemente é importante conhecer antecipadamente os padrões de falhas e as tendências do sinal para que elas ocorram, possibilitando uma detecção e diagnóstico de falhas mais exatas e uma redução nos custos de manutenção do processo.

Portanto, a ferramenta proposta deve ser capaz de detectar padrões anormais, em uma janela de tempo substancial, de forma a analisar padrões de comportamento que possam futuramente culminar em uma falta. Esse é o grande salto de tecnologia quando se compara aos sistemas atualmente implementados. A detecção imediata de perturbações na rede elétrica já é realizada, de forma bastante satisfatória, por relés digitais de avançada tecnologia.

Compreender os padrões de pré-falta é um requisito imprescindível ao correto diagnóstico de falhas. Cada um destes padrões será utilizado para o treinamento da rede neural. Validando-se o treinamento, é importante integrar a RNA a um sistema online para testá-la. Na ponta deste processo deseja-se que, na ocorrência

de perturbações, uma saída alerte o operador do sistema sobre a probabilidade de ocorrência da falta (por exemplo, um alarme visual e/ou sonoro no sistema SCADA). Se, porventura, a falta for inevitável, uma mensagem de apoio à decisão deve ser fornecida pela aplicação, orientando como restabelecer o sistema rapidamente. A essa altura, é fundamental compreender que não se deseja apenas evitar faltas: minimizar o tempo de restabelecimento é um grande desafio para as distribuidoras de energia elétrica na atualidade.

### 3.1 Formulação do Problema

Nesta seção apresenta-se a descrição, caracterização e formulação do problema. Do ponto de vista da automação de subestações, ao se utilizar um sistema SCADA (Supervisory Control and Data Acquisition - Controle Supervisório e Aquisição de Dados), não é necessário que o operador permaneça fisicamente na subestação; pelo contrário, tal tecnologia lhe permite trabalhar em um Centro de Operação remoto e, mesmo distante, monitorar e controlar todos os equipamentos desta subestação.

Fica claro, portanto, a criticidade de se possuir um sistema de automação local na subestação que seja confiável e com alta disponibilidade. Chegar neste nível é atualmente um grande desafio para a maioria das concessionárias de energia. Neste estudo, trabalhamos na planta da Companhia Energética do Maranhão - CEMAR, uma concessionária de energia do estado do Maranhão, no Brasil.

A CEMAR possui aproximadamente 1.8 milhão de clientes no estado do Maranhão, em seus 217 municípios. Trata-se de um sistema de transmissão com linhas de 138kV, 69kV e 34,5kV e distribuição com alimentadores de 13,8kV. Essa distribuidora já possui um bom nível de automação em várias de suas subestações.

A limitação de dados operacionais disponíveis aos operadores dos sistemas de energia e a complexidade da análise de faltas no setor elétrico motivam o desenvolvimento de uma metodologia de suporte à decisão capaz de gerar o correto diagnóstico de faltas.

Em uma subestação de energia elétrica, o conceito de proteção está centrado em um equipamento de fundamental importância: o transformador de força. Vários estudos vêm sendo desenvolvidos com o objetivo de reduzir falhas em equi-

pamentos. Neste estudo, um algoritmo baseado na Teoria de Redes Neurais Artificiais foi desenvolvido com o intuito de detectar comportamentos característicos de condições de falta em transformadores de força de subestações de energia elétrica, bem como fornecer apoio à decisão dos operadores do sistema.

O problema é formulado como um problema de monitoramento de análise de anomalias em subestações de energia elétrica e propõe a integração de um sistema SCADA com uma rede neuronal.

A arquitetura do sistema proposto nesta pesquisa é apresentada na Figura 3.1.

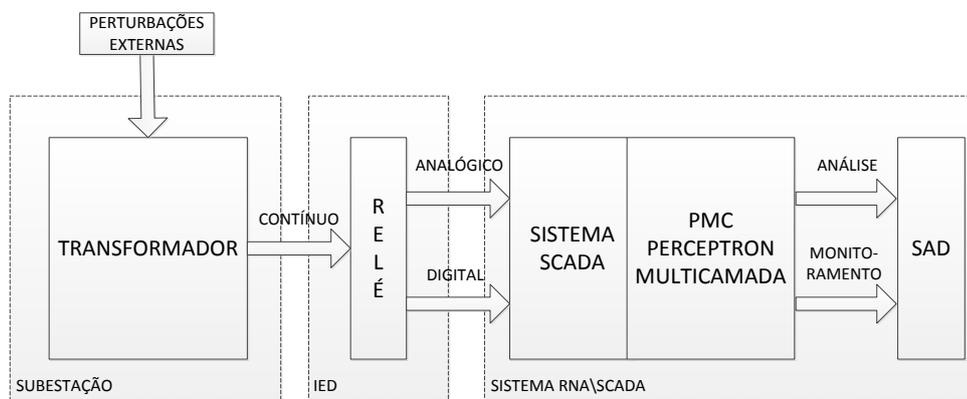


Figura 3.1: Arquitetura do Sistema proposto.

Como pode-se perceber, o equipamento presente no estágio inicial de nossa análise é o transformador de força, o qual é responsável pelo rebaixamento de tensão na subestação para nível adequado à distribuição. Este equipamento percebe as perturbações externas que são objeto desse estudo. As informações oriundas do transformador são monitoradas pelo relé digital de proteção, o qual repassa um conjunto de dados digitais e analógicos para o sistema SCADA, que serão traduzidos em informações e alarmes visuais e/ou sonoros na tela do operador do sistema. No sistema SCADA desenvolvido foi embarcada a Rede Neural do tipo Perceptron Multicamada para classificação do sinal observado. As informações analisadas são também encaminhadas para o sistema de apoio à decisão (SAD) que fornece as coordenadas necessárias à ação do operador. Essa figura representa o cerne dessa dissertação, concentrando os elementos fundamentais aqui analisados.

## 3.2 Base de Dados

A base de estudo considerada refere-se aos relatórios de eventos (*ER - Event Reports*) dos IED's. Um ER contém variáveis analógicas, como correntes primárias e secundárias (por fase) e também variáveis digitais, como alarmes. O IED escolhido neste estudo foi o SEL387-A (Laboratories (2004)), fabricado pela SEL (Schweitzer Engineering Laboratories). Os eventos de oscilografia foram coletados do IED do transformador de força 02T1 da Subestação Renascença da CEMAR, apresentado na Figura 2.7 da Seção 2.4.4. Um usuário pode configurar um IED para gerar ER's periodicamente (a cada uma hora, por exemplo) ou apenas na ocorrência de um determinado evento (por exemplo, na atuação de uma proteção do relé). No primeiro caso, o foco está em “fotografar” o sistema periodicamente para uma análise mais aprofundada. No segundo, interessa-se apenas em diagnosticar a causa da atuação da proteção, tratando a falta como um evento estanque em um específico intervalo de tempo. Nessa dissertação utiliza-se uma combinação das duas abordagens para gerar uma análise mais completa e confiável.

Os dados foram coletados do IED do transformador 02T1 da Subestação Renascença; na Figura 3.2 apresenta-se o diagrama unifilar desta subestação, destacando-se o transformador analisado.

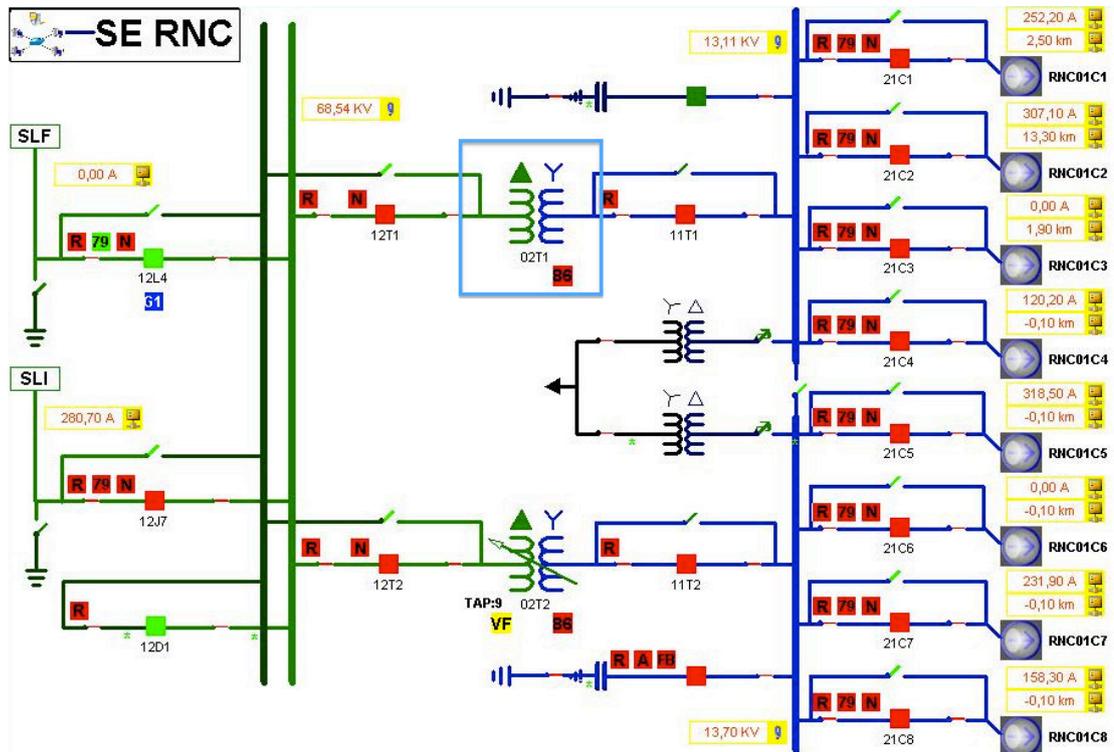


Figura 3.2: Diagrama Unifilar da Subestação Renascença (CEMAR), com destaque para o transformador analisado neste estudo.

Ajustando-se a configuração apropriada, é possível analisar tanto eventos *normais*, eventos de *perturbações transitórias* e eventos de *falta*. Eventos normais representam o comportamento esperado do sistema na ausência de perturbações; eventos de perturbações transitórias dizem respeito a comportamentos anormais que porém não culminarão em uma falta, enquanto eventos de falta retratam o sistema quando é submetido a perturbações que evoluem para uma condição não aceitável. Quando essa condição ocorre a proteção do IED atua, isolando o circuito onde o defeito está presente e cessando a condição que provocou a falta.

Este estudo está desenhado na linha do tempo da seguinte forma: inicialmente analisaremos os dados oriundos dos relés digitais da subestação, obtidos através dos ER. A detalhada análise desses relatórios de eventos constitui etapa imprescindível para um bom desempenho do estudo. O segundo passo será a criação de uma base de dados com diferentes eventos reais de períodos anteriores, bem caracterizados e classificados. O terceiro passo consiste em selecionar 60% de eventos

da base de dados para compor o conjunto de treinamento da RNA. O quarto passo consiste em validar o treinamento da RNA através de um conjunto de teste, contendo dados diferentes daqueles do treinamento e avaliando a performance da rede na classificação dos padrões a ela apresentados. Esse conjunto corresponde aos 40% restante da base de dados.

Como IHM utilizaremos uma aplicação SCADA desenvolvida utilizando-se o Elipse E3 Power®, uma ferramenta de desenvolvimento de sistemas fornecida pela Elipse. A função básica dessa aplicação é monitorar as variáveis de campo, alarmes e controles de forma amigável ao operador do sistema. Como função primordial, o SCADA permitirá a análise dos dados, oferecendo suporte à tomada de decisões do operador do sistema através da antecipação de possíveis faltas ou agilizando o processo de recuperação de falhas. A aplicação desenvolvida neste ambiente também realiza armazenamento histórico de estados e medidas. Dentre as diversas soluções disponíveis no mercado, optou-se pelo Elipse E3 Power® devido à alta flexibilidade de programação e pela excelente performance desta ferramenta.

Trata-se de uma solução baseada em observações do mundo real (medições dos estados de equipamentos em subestações), histórico de falhas e processamento da informação por meio da metodologia de Sistemas de Apoio à Decisão.

Portanto, neste estudo, para desenvolver um modelo neural para classificar padrões característicos de faltas, foi criada uma base de dados inicial, contendo registros de eventos normais, perturbações transitórias e eventos de falta. A base de dados foi construída através de relatórios de eventos (ER's) coletados diretamente de IED's. Portanto, a base de dados contém condições **normais** de operação, condições de **perturbações sem falta** e condições de **perturbações com falta**.

A Figura 3.3 ilustra uma oscilografia de um evento de falta.  $IA_{W1}$ ,  $IB_{W1}$  e  $IC_{W1}$  são correntes primárias e  $IA_{W2}$ ,  $IB_{W2}$  e  $IC_{W2}$  são correntes secundárias do transformador de força. Algumas variáveis digitais também estão disponíveis neste registro de oscilografia. 51N2T, por exemplo, representa a proteção chamada "*Winding 2 residual inverse-time overcurrent element timed out*" (Laboratories (2004)) - disparo do elemento sobrecorrente da corrente residual do enrolamento 2, obedecendo a uma curva do tipo tempo-inverso. Uma linha mais espessa na oscilografia demonstra a ativação da variável. O eixo-X demonstra a evolução

do tempo, em ciclos, onde 1 ciclo = 1/60 segundos. É possível verificar que por volta de 9,5 ciclos a proteção atua. Combinando todas essas informações de diferentes registros de oscilografia pode-se então alimentar a base de dados com características bastante relevantes de transformadores de força durante condições de operação normais, perturbações sem falta e perturbações com falta.

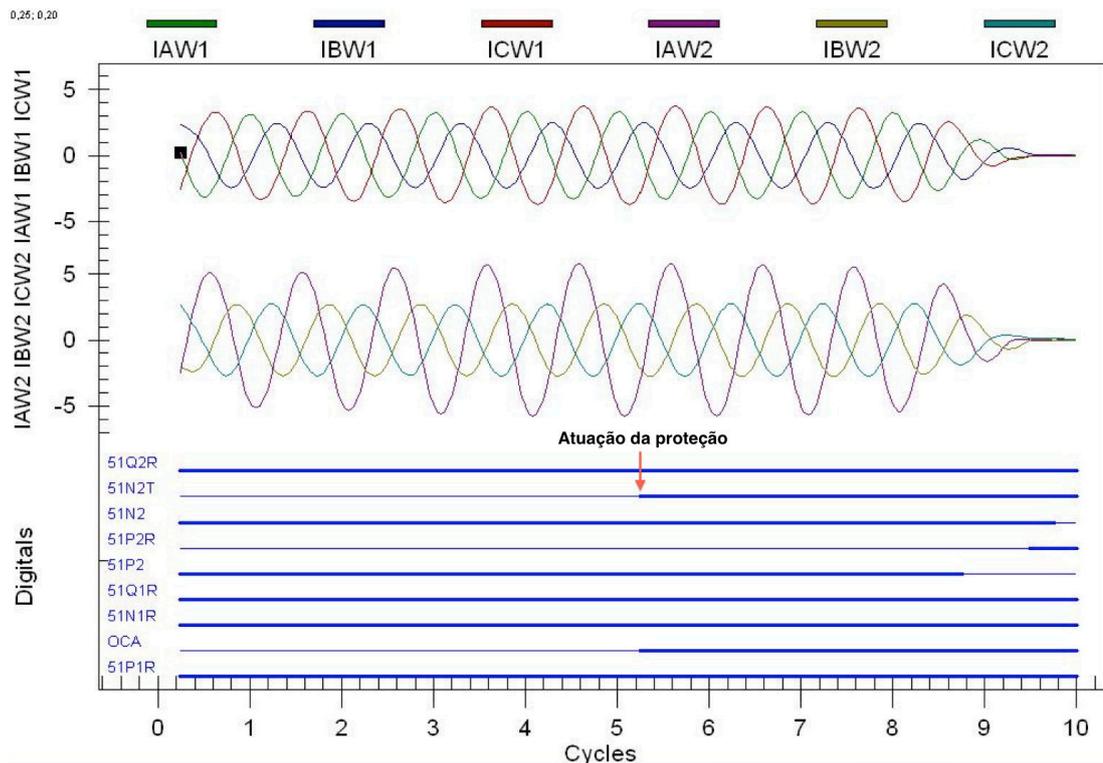


Figura 3.3: Oscilografia de um Evento de Falta.

### 3.3 Projeto da RNA

Nesta seção apresenta-se as etapas de configuração e treinamento da rede MLP para detecção de faltas. A etapa de validação é descrita na seção 4.2.4.

#### 3.3.1 Configuração da RNA

A RNA desenvolvida neste estudo conteve na camada de entrada as seguintes variáveis:

- $IA_{W1}$ : corrente da fase A do enrolamento primário do transformador.
- $IB_{W1}$ : corrente da fase B do enrolamento primário do transformador.
- $IC_{W1}$ : corrente da fase C do enrolamento primário do transformador.

e as correntes no lado de baixa do transformador:

- $IA_{W2}$ : corrente da fase A do enrolamento secundário do transformador.
- $IB_{W2}$ : corrente da fase B do enrolamento secundário do transformador.
- $IC_{W2}$ : corrente da fase C do enrolamento secundário do transformador.

Futuramente pode-se utilizar, além das variáveis analógicas supracitadas, variáveis digitais para auxiliar na caracterização de cada tipo de evento, checando quais os impactos dessa nova abordagem no aprendizado da rede. Uma boa escolha seria a ativação do elemento 51 ou ainda do elemento 50 (Laboratories (2004)).

O alvo da saída da RNA foi definido usando-se três classes de padrão: **normal**, **perturbação sem falta** e **perturbação com falta**. Para facilitar a análise, chamaremos estes padrões de **N**, **PSF** e **PCF**, respectivamente. Na camada de entrada da rede estão as correntes primárias ( $IA_{W1}$ ,  $IB_{W1}$  e  $IC_{W1}$ ) e as correntes secundárias do transformador ( $IA_{W2}$ ,  $IB_{W2}$  e  $IC_{W2}$ ). Na camada intermediária temos 3 neurônios; a função de ativação escolhida para esta camada será a função sigmóide, gerando portanto como saída de cada neurônio valores no intervalo entre 0 e 1. Já na camada de saída temos 2 neurônios cuja função de ativação será a própria soma ponderada das suas entradas. A Tabela 3.1 resume as características da RNA implementada:

Tabela 3.1: Especificações da Rede Neural

Camada	Quantidade de Neurônios	Função de Ativação	Observações
Entrada	6	-	$IA_{W1}, IB_{W1}, IC_{W1}$ $IA_{W2}, IB_{W2}, IC_{W2}$
Escondida	3	$\frac{1}{1+\exp -v}$	-
Saída	2	$y(v) = v$	-

sendo  $v$  a soma ponderada das entradas de um neurônio.

### 3.3.2 Estratégia de Treinamento da MLP

Como explicado anteriormente, a base de dados contém condições normais de operação, perturbações transitórias e faltas. O objetivo, portanto, é que tais eventos sejam adequadamente classificados pela rede conforme os padrões definidos N, PSF e PCF, respectivamente. Esta base de dados foi construída conforme os seguintes passos:

1. **Vetores originais:** para obtenção dos vetores originais foram coletados arquivos de oscilografias extraídos de um relé diferencial de transformador de força em diferentes datas. Considerando todos os valores de corrente em módulo, os seguintes vetores foram gerados:
  - OriginalNormal1 –  $VO_{N1}$  – vetor de 120 posições de evento NORMAL colhido na Oscilografia 1.
  - OriginalPSF1 –  $VO_{PSF1}$  – vetor de 120 posições de evento PSF colhido na Oscilografia 2.
  - OriginalPCF1 –  $VO_{PCF1}$  – vetor de 120 posições de evento PCF colhido na Oscilografia 3.
  - OriginalNormal2 –  $VO_{N2}$  – vetor de 120 posições de evento NORMAL colhido na Oscilografia 4.
  - OriginalPSF2 –  $VO_{PSF2}$  – vetor de 120 posições de evento PSF colhido na Oscilografia 5.
  - OriginalPCF2 –  $VO_{PCF2}$  – vetor de 120 posições de evento PCF colhido na Oscilografia 6.
2. Para garantir que a RNA não se especializasse no último grupo de padrões apresentado durante a fase de treinamento, os **vetores originais** foram reorganizados aleatoriamente resultando nos seguintes vetores:
  - EmbaralhadoNormal1 –  $VE_{N1}$  – corresponde ao vetor  $VO_{N1}$  embaralhado.

- EmbaralhadoPSF1 –  $VE_{PSF1}$  – corresponde ao vetor  $VO_{PSF1}$  embaralhado.
  - EmbaralhadoPCF1 –  $VE_{PCF1}$  – corresponde ao vetor  $VO_{PCF1}$  embaralhado.
  - EmbaralhadoNormal2 –  $VE_{N2}$  – corresponde ao vetor  $VO_{N2}$  embaralhado.
  - EmbaralhadoPSF2 –  $VE_{PSF2}$  – corresponde ao vetor  $VO_{PSF2}$  embaralhado.
  - EmbaralhadoPCF2 –  $VE_{PCF2}$  – corresponde ao vetor  $VO_{PCF2}$  embaralhado.
3. Diferentes divisões foram utilizadas para formação do conjunto de treino e de teste. Entretanto, considerando 60% de cada **vetor embaralhado** para gerar os **vetores de treinamento** obtivemos melhores resultados:

- TreinamentoNormal –  $VTR_N$  – vetor de 144 posições composto unindo-se 60% de  $VE_{N1}$  a 60% de  $VE_{N2}$  alternadamente, resultando na seguinte estrutura:

$$VTR_N = [VE_{N1}(1), VE_{N2}(1), VE_{N1}(2), VE_{N2}(2) \dots VE_{N1}(72), VE_{N2}(72)]$$

- TreinamentoPSF –  $VTR_{PSF}$  – vetor de 144 posições composto unindo-se 60% de  $VE_{PSF1}$  a 60% de  $VE_{PSF2}$  alternadamente, resultando na seguinte estrutura:

$$VTR_{PSF} = [VE_{PSF1}(1), VE_{PSF2}(1), VE_{PSF1}(2), VE_{PSF2}(2) \dots VE_{PSF1}(72), VE_{PSF2}(72)]$$

- TreinamentoPCF –  $VTR_{PCF}$  – vetor de 144 posições composto unindo-se 60% de  $VE_{PCF1}$  a 60% de  $VE_{PCF2}$  alternadamente, resultando na seguinte estrutura:

$$VTR_{PCF} = [VE_{PCF1}(1), VE_{PCF2}(1), VE_{PCF1}(2), VE_{PCF2}(2) \dots VE_{PCF1}(72), VE_{PCF2}(72)]$$

- **TreinamentoCompleto** –  $VTR$  – vetor de 432 posições composto unindo-se, alternadamente, os vetores  $VTR_N$ ,  $VTR_{PSF}$  e  $VTR_{PCF}$ , resultando na seguinte estrutura:

$$VTR = [VTR_N(1), VTR_{PSF}(1), VTR_{PCF}(1), VTR_N(2), VTR_{PSF}(2), \\ VTR_{PCF}(2) \dots VTR_N(144), VTR_{PSF}(144), VTR_{PCF}(144)]$$

Este é o vetor que será utilizado para o treinamento da rede; todas as manipulações anteriores buscam demonstrar a sua construção.

- **TreinamentoSaidaCompleto** –  $yTR$  - contém as saídas desejadas para cada posição do vetor  $VTR$ , constituindo o gabarito da rede durante a fase de treinamento. Este vetor foi construído adotando-se o seguinte critério:

- Evento Normal -  $yN(10) = -1$  e  $yN(11) = -1$
- Evento PSF -  $yN(10) = 0$  e  $yN(11) = 0$
- Evento PCF -  $yN(10) = +1$  e  $yN(11) = +1$

4. Os **vetores de teste** foram gerados a partir dos 40% restantes de cada **vetor embaralhado**:

- **TesteNormal** –  $VT_N$  – vetor de 96 posições composto unindo-se os 40% restantes de  $VE_{N1}$  aos 40% restantes de  $VE_{N2}$  alternadamente, resultando na seguinte estrutura:

$$VT_N = [VE_{N1}(73), VE_{N2}(73), VE_{N1}(74), VE_{N2}(74) \dots VE_{N1}(120), VE_{N2}(120)]$$

- **TestePSF** –  $VT_{PSF}$  – vetor de 96 posições composto unindo-se os 40% restantes de  $VE_{PSF1}$  aos 40% restantes de  $VE_{PSF2}$  alternadamente, resultando na seguinte estrutura:

$$VT_{PSF} = [VE_{PSF1}(73), VE_{PSF2}(73), VE_{PSF1}(74), VE_{PSF2}(74) \dots \\ VE_{PSF1}(120), VE_{PSF2}(120)]$$

- **TestePCF** –  $VT_{PCF}$  – vetor de 96 posições composto unindo-se os 40% restantes de  $VE_{PCF1}$  aos 40% restantes de  $VE_{PCF2}$  alternadamente,

resultando na seguinte estrutura:

$$VT_{PCF} = [VE_{PCF1}(73), VE_{PCF2}(73), VE_{PCF1}(74), \\ VE_{PCF2}(74) \dots VE_{PCF1}(120), VE_{PCF2}(120)]$$

Esses 3 vetores são utilizados na etapa de testes da RNA.

Portanto, a base de dados foi dividida em dois conjuntos: um conjunto de treinamento contendo 60% dos eventos, através do qual a RNA aprenderá como classificar os diferentes padrões; e um conjunto de teste, contendo os 40% restantes dos eventos, os quais serão apresentados à rede para fins de validação do treinamento e avaliação da performance da RNA.

### 3.3.3 Critério de Parada de Treinamento

Ao se definir o critério de parada de treinamento no projeto de uma RNA, deve-se lembrar que um treinamento prolongado demais pode levar ao sobreajuste da rede, especialmente no caso de dispormos de poucos pares de entrada e saída para o conjunto de treinamento, o que pode piorar o desempenho da rede quando o conjunto de teste lhe for apresentado. Embora nesta pesquisa disponha-se de um volume significativo de dados para treinamento, deve-se considerar um critério que não provoque a indesejada condição do sobreajuste.

Neste estudo, utilizaremos como critério de parada de treinamento a seguinte condição:

- Erro quadrático de época  $\leq 0,1$

Na seção 4.2.4 discute-se os resultados abordando este critério.

## 3.4 Conclusões e Comentários

Apresentou-se a caracterização do problema desde os IEDs até o supervisório, salientando-se a importância da oscilografia do transformador no contexto do sistema de distribuição e a viabilidade da proposta utilizando sistema SCADA.

Em relação à base de dados, os registros de oscilografias fornecem as informações para análise *off-line* do comportamento do sistema; estas informações

foram utilizadas como entrada da rede neural, enquanto que para o processamento *on-line* as informações são provenientes da leitura em tempo real dos IEDs.

O projeto da rede neuronal foi desenvolvido para a rede perceptron multicamada que tem como entrada as correntes primárias e secundárias do transformador. Utilizou-se uma função de ativação sigmoideal para a camada intermediária; já para a camada de saída utilizou-se simplesmente a soma ponderada das entradas dos neurônios.

Na Seção Estratégia de Treinamento da MLP apresentou-se um procedimento com quatro passos para treinar a rede. O primeiro passo consistiu em selecionar seis conjuntos de dados que representam as três situações operacionais: condição normal, condição PSF e condição PCF. O segundo passo visou minimizar os efeitos de especialização da rede. O terceiro passo consistiu na seleção do percentual dos vetores embaralhados para gerar vetores de treinamento e o quarto e último passo foi a geração dos vetores de teste com o percentual restante de dados.

# Experimentos de Campo e Computacionais

---

Neste capítulo ilustra-se a aplicação SCADA desenvolvida exibindo algumas de suas telas. Em seguida, apresenta-se e discute-se os resultados obtidos através dos experimentos de campo e computacionais. Posteriormente, demonstra-se a influência de alguns parâmetros na capacidade de aprendizagem da rede. Por fim, analisa-se a performance da rede conforme os resultados obtidos.

Para validar a abordagem proposta, uma interface foi desenvolvida utilizando o Elipse E3 Power®. A interface possui módulos para visualização e manipulação de oscilografias, módulos de treinamento e teste da RNA e módulos de integração com um sistema SCADA para classificação online de eventos. Neste último, a interface desenvolvida fornece informação visual e sonora sempre que um comportamento anormal do sistema é detectado pela RNA. O objetivo é fornecer suporte à tomada de decisão dos operadores do sistema SCADA, através de diagnósticos precisos e ações direcionadas. Essa interface permite também a entrada de dados pela área de Manutenção, funcionando como realimentação (*feedback*) de nosso sistema. Isso significa que, após uma detalhada análise de um determinado evento e a coleta de todas as informações disponíveis, o corpo técnico confirmará (ou invalidará) o diagnóstico fornecido pela aplicação. Esse *feedback* é de fundamental importância para o aprendizado contínuo da RNA.

Uma vez que a rede esteja devidamente treinada, o sistema funcionará como uma eficaz ferramenta de classificação de eventos e também de recuperação rápida

após faltas. É necessário compreender que, durante a ocorrência de uma falta, alguns erros humanos podem ocorrer. Isso se dá pois em algumas situações o controlador do sistema não dispõe de todas as informações necessárias para a tomada de uma decisão assertiva; além disso, existe algum nível de tensão envolvido e conseqüente possibilidade de erro humano durante a recuperação de um sistema após uma falta, uma vez que o fornecimento de energia elétrica está interrompido para diversos clientes ou até mesmo para cidades e estados. Essas são algumas das vantagens ao se implementar o sistema proposto neste estudo.

## 4.1 Desenvolvimento do Sistema SCADA

Nesta seção detalha-se as funcionalidade da aplicação desenvolvida. A Figura 4.1 exibe a tela principal da aplicação; nela ilustra-se a configuração da rede implementada. Através desta tela, pode-se escolher:

- o modo de funcionamento da rede: **Treinamento** ou **Teste**;
- o modo de visualização dos resultados: **Normal** ou **Passo-a-passo**;
- o vetor de entrada a ser utilizado;
- o modo de correção de erro: **Online** ou **Batelada**.

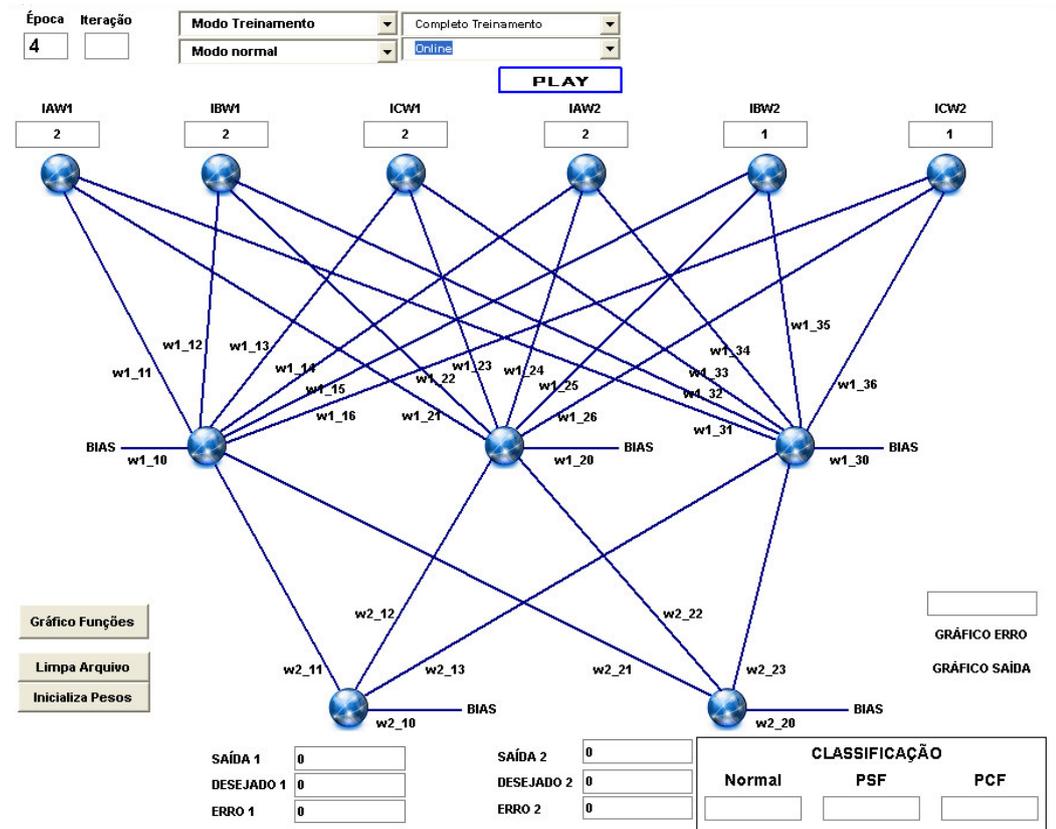


Figura 4.1: Implementação da RNA, sendo possível realizar as Fases de Treinamento e Teste.

Se escolhermos, por exemplo, o modo de **Treinamento** da rede e pressionando o botão *Play* o algoritmo de treinamento da RNA será executado tomando como parâmetros os valores escolhidos na tela inicial. Uma tela contendo o resultado do treinamento será exibida, conforme ilustrado na Figura 4.2. Como pode-se perceber, esta tela exibe as matrizes de pesos solução ( $W_1$  e  $W_2$ ).

**MATRIZES DE PESOS SOLUÇÃO**

<b>W1=</b>	0.282722454	1.056232126	0.857117230	0.849067119	1.014135743	1.818650887	0.759919951
	0.905714638	0.925993473	-3.281765588	-0.958351510	-0.434123614	0.369779386	4.214333288
	0.493033427	-0.942913307	0.314723631	-0.973293661	-0.145512093	-0.392956733	-0.551407497

<b>W2=</b>	0.306127526	0.169789365	-1.382537904	5.290643018
	0.990482370	0.163021408	-1.382494141	5.288821849

Figura 4.2: Resultado do Treinamento da Rede Neural.

Além das matrizes de pesos solução, pode-se analisar os erros calculados no processo de treinamento da RNA: erro quadrático absoluto e o erro quadrático de época. A Figura 4.3 ilustra essa funcionalidade.

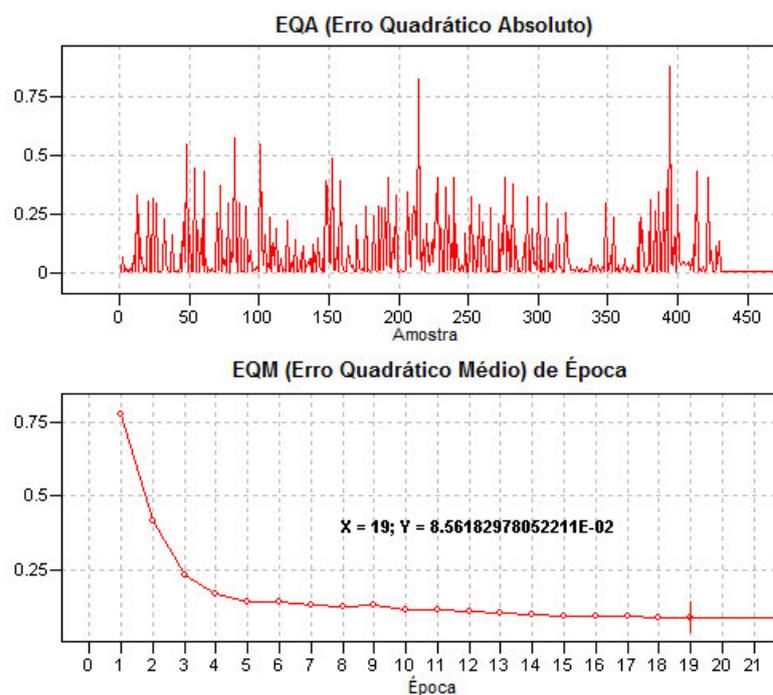


Figura 4.3: Erros Quadráticos calculados durante o Processo de Treinamento da RNA.

Já se escolhermos o modo de **Teste** da rede e pressionando o botão **Play** o algoritmo de treinamento da RNA será executado tomando como parâmetros os valores escolhidos na tela inicial, considerando apenas uma época, ou seja, executando apenas uma iteração do algoritmo da RNA. Na mesma tela será exibido o resultado da classificação realizada pela RNA, conforme ilustrado na Figura 4.4.

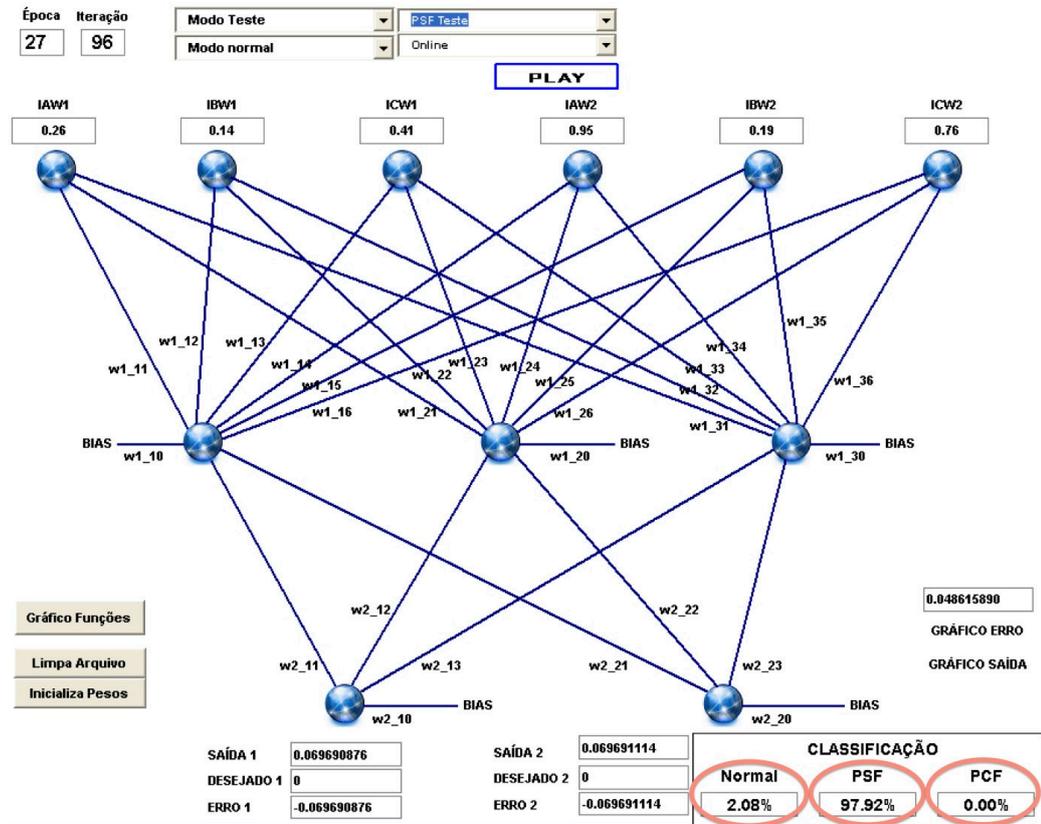


Figura 4.4: Resultado do Teste da Rede Neural.

A aplicação desenvolvida conta com um módulo de exibição de oscilografias de relés (Figura 4.5); esta função pode ser acessada através do botão **Gráficos** seguido da Opção **Oscilografias**. Na tela desta funcionalidade é possível escolher qualquer oscilografia que esteja na base de dados do programa e em seguida deve-se pressionar o botão **Carregar**. O gráfico superior exibe as correntes primárias do transformador ( $IA_{W1}$ ,  $IB_{W1}$  e  $IC_{W1}$ ) enquanto o gráfico inferior exibe as correntes secundárias ( $IA_{W2}$ ,  $IB_{W2}$  e  $IC_{W2}$ ).

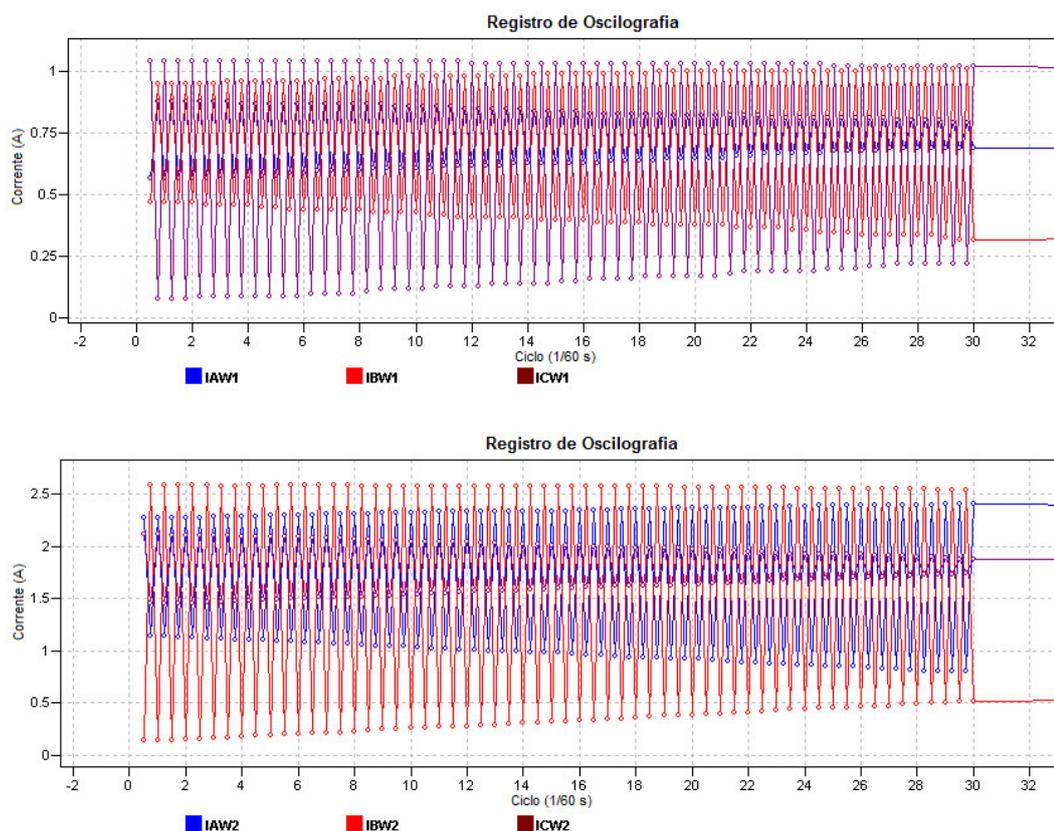


Figura 4.5: Oscilografias de Relés.

Utilizando-se o mesmo menu (botão **Gráficos**) seguido porém da Opção **Funções**, uma nova tela é exibida permitindo que sejam plotadas as funções que podem ser utilizadas como funções de ativação das camadas intermediárias e/ou de saída da RNA(Figura 4.6).

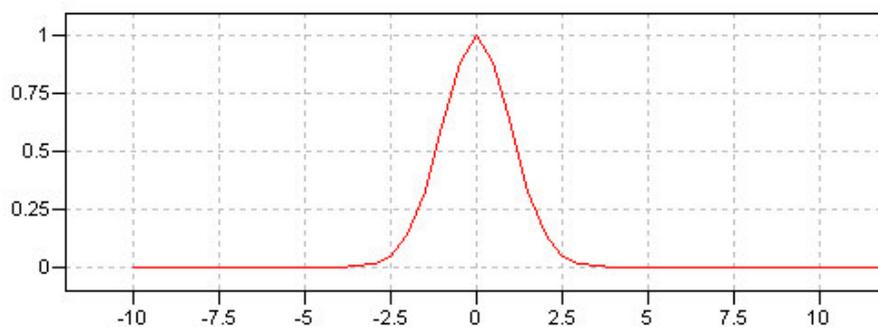


Figura 4.6: Função Radial Gaussiana.

Uma interface de apoio à decisão também foi desenvolvida no SCADA. Como percebe-se na Figura 4.7, uma breve estatística é exibida no canto direito da tela, contendo a classificação de eventos em Normal, PSF e PCF nas últimas vinte e quatro horas. Caso a quantidade de ocorrências ultrapasse o valor configurado como máximo aceitável, uma sinalização em cor vermelha destacará a variável violada e um roteiro de ações passo-a-passo auxiliará o operador na tomada de decisões para evitar uma falta ou ainda para restabelecer o sistema após uma falta.

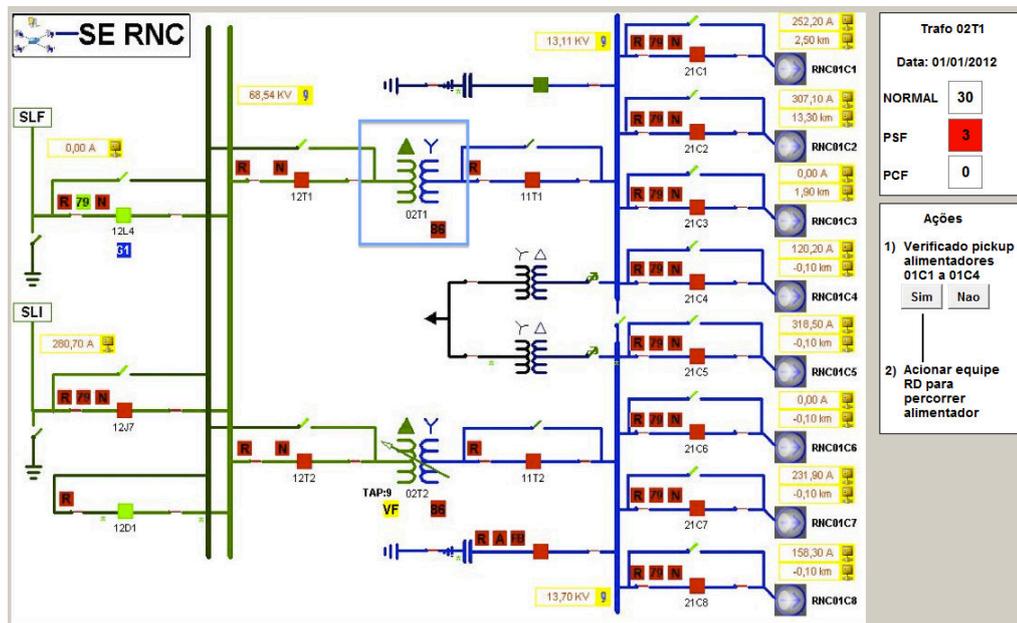


Figura 4.7: Implementação do SAD no SCADA.

Além disso, um banco de dados com o histórico de ocorrências da CEMAR foi desenvolvido e integrado ao SCADA. Os novos eventos adquiridos e classificados pela RNA também deverão ser armazenados nesse mesmo banco, guardando a informação do tipo de classificação fornecido pela rede. A Figura 4.8 exibe a tela do SCADA onde se pode visualizar este banco de dados, além de realizar pesquisas utilizando filtros específicos.

**FORMULÁRIO DE ENTRADA DE DADOS**

Classe Objeto ? Falha Funcional ? AF ? Expurgada

Modo Falha ?

Causa ?

Classificação RIA

Normal  
PSF  
PCF

RAP ? Hora ? DEC ? FEC ? Sistema ? Subsistema ? Cod Operação ? Equipamento ? Categoria Evento ?

CADASTRAR LIMPAR

RAP	ClasseObjeto	FalhaFuncional	ModoFalha	Causa	AF	EXP	Hora	DEC_
2011/051	T.Potência	Não isola completamente as bobinas	Condução de eletricidade entre bobinas o...	Comprometimento do isolante	N	N	4/1/2011 10:33:08 AM	0.089
2011/050	B.Capacitor	Avaria do Banco de Capacitores			S	N	4/1/2011 10:04:17 AM	0.060
2011/049	T.Potência	Não isola completamente as bob	Condução de eletricidade entre	Comprometimento do isolante	N	N	4/1/2011 10:03:27 AM	0.040
2011/048	T.Potência	Não isola completamente as bob	Condução de eletricidade entre	Comprometimento do isolante	N	N	4/1/2011 10:01:32 AM	4.000
2011/047	Religador	Não opera continuamente	Aquecimento das conexões	Conexão com problema	S	N	3/31/2011 5:58:26 PM	5.000
2011/046	T.Força	Tensão de saída fora do padrão	Bobinas danificadas	Deposição de material	N	S	3/31/2011 5:56:27 PM	2.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000
2011/045	Religador	Abriu sem motivo	Proteção envia comando de aber	Relé recebe informação errada	S	N	3/31/2011 5:53:25 PM	3.000

Registro: 1 de 27

Figura 4.8: Implementação de Banco de Dados de Ocorrências no SCADA.

A área de Estudos da Operação do Sistema e a área de Manutenção poderão utilizar este banco de dados para consultar a classificação apresentada pela RNA para cada evento, com o objetivo de confrontá-la com a classificação feita pelo grupo de Engenharia após análise dos fatos e dados. Em caso de erro no diagnóstico fornecido pela RNA, a área de Manutenção deverá atualizar o vetor de treinamento da rede, acrescentando as informações do evento em que houve a divergência; isso servirá como *feedback* do sistema. Um novo treinamento deverá então ser realizado.

Trechos do código do desenvolvimento da aplicação SCADA podem ser verificados no Apêndice A.

## 4.2 Implementação da Rede MLP

Ao longo do desenvolvimento da dissertação, ajustes nas escolhas de alguns parâmetros foram realizados para garantir o sucesso do treinamento da rede. A seguir descreve-se os seguintes tópicos: a quantidade de neurônios da camada de saída, o treinamento *backpropagation* e as funções de ativação. Estes item viabilizam a preparação da rede para operar na detecção e classificação de eventos

do sistema.

### 4.2.1 Quantidade de Neurônios da Camada de Saída

Inicialmente definiu-se apenas 1 neurônio na camada de saída. Adotando-se o critério de  $n - 1$  (da quantidade de padrões) neurônios e uma vez definidos os 3 padrões - Normal, PSF e PCF - a rede foi reprojeta para possuir 2 neurônios na camada de saída. Percebeu-se que esta escolha apresentou bons resultados, embora não seja possível definir ganhos reais se comparada à rede com apenas 1 neurônio na camada de saída.

### 4.2.2 Treinamento *Backpropagation*

O resultado final do treinamento da rede neural gerou os pesos sinápticos da camada intermediária dados por

$$W_1 = \begin{bmatrix} 0.742 & 0.443 & -0.506 & 0.612 & 0.377 & 0.334 & -2.149 \\ 0.775 & 1.188 & 0.958 & 0.126 & 0.715 & 2.169 & 0.912 \\ 0.304 & 0.648 & 1.684 & -0.309 & 1.683 & -1.896 & 2.449 \end{bmatrix} \quad (4.1)$$

e da camada de saída dados por

$$W_2 = \begin{bmatrix} 0.798 & 1.180 & 1.594 & -2.769 \\ 0.328 & 1.180 & 1.600 & -2.771 \end{bmatrix} \quad (4.2)$$

### 4.2.3 Funções de Ativação

A tangente hiperbólica foi a função de ativação escolhida para as camadas intermediária e de saída:

$$y(v) = \frac{1 - e^{-v}}{1 + e^{-v}} \quad (4.3)$$

No entanto, durante a fase de treinamento percebeu-se que o treinamento da rede não obtinha sucesso e o erro médio quadrático não reduzia de forma constante conforme esperado, oscilando indefinidamente. Ao modificar-se a escolha

da função de ativação da camada intermediária para função sigmóide e da camada de saída para função soma ponderada das entradas, os resultados esperados foram obtidos. De fato, essa modificação foi determinante para o sucesso desta dissertação pois até então nenhum resultado conclusivo havia sido alcançado.

#### 4.2.4 Validação do Treinamento

O vetor de treinamento completo,  $VTR$ , contém informações dos diferentes possíveis padrões de eventos combinados em único vetor. Inicialmente esse vetor foi montado pela concatenação sequencial dos vetores  $VO_{N1}$ ,  $VO_{PSF1}$  e  $VO_{PCF1}$ . No entanto, pôde-se perceber claramente a influência do último grupo apresentado à rede na fase de treinamento. Em outras palavras, a rede se especializava apenas no treinamento da última classe. Modificou-se, portanto, a forma de criação de  $VTR$ , de tal forma que tanto os vetores originais fossem embaralhados quanto o vetor de treinamento fosse composto pela composição de forma alternada destes novos vetores - a construção deste vetor foi detalhada anteriormente na seção 3.3.3 do Capítulo 3. Essa alteração produziu os resultados desejados para o treinamento com os três tipos de padrões, ao contrário da proposta inicial em que a rede só conseguia êxito para o último padrão apresentado.

De posse das definições anteriormente descritas, utilizou-se a aplicação desenvolvida para realizar o treinamento da rede. Desta forma, escolheu-se como vetor de entrada da rede  $VTR$ , uma vez que este vetor continha as informações dos três tipos de eventos definidos neste estudo (Normal, PSF e PCF). Na Figura 4.9 ilustra-se a realização da etapa de treinamento da rede neural, assumindo todas as considerações já descritas anteriormente. Neste exemplo é realizado um treinamento utilizando o vetor  $VTR$  para 27 épocas com correção de erro em modo *Online*.

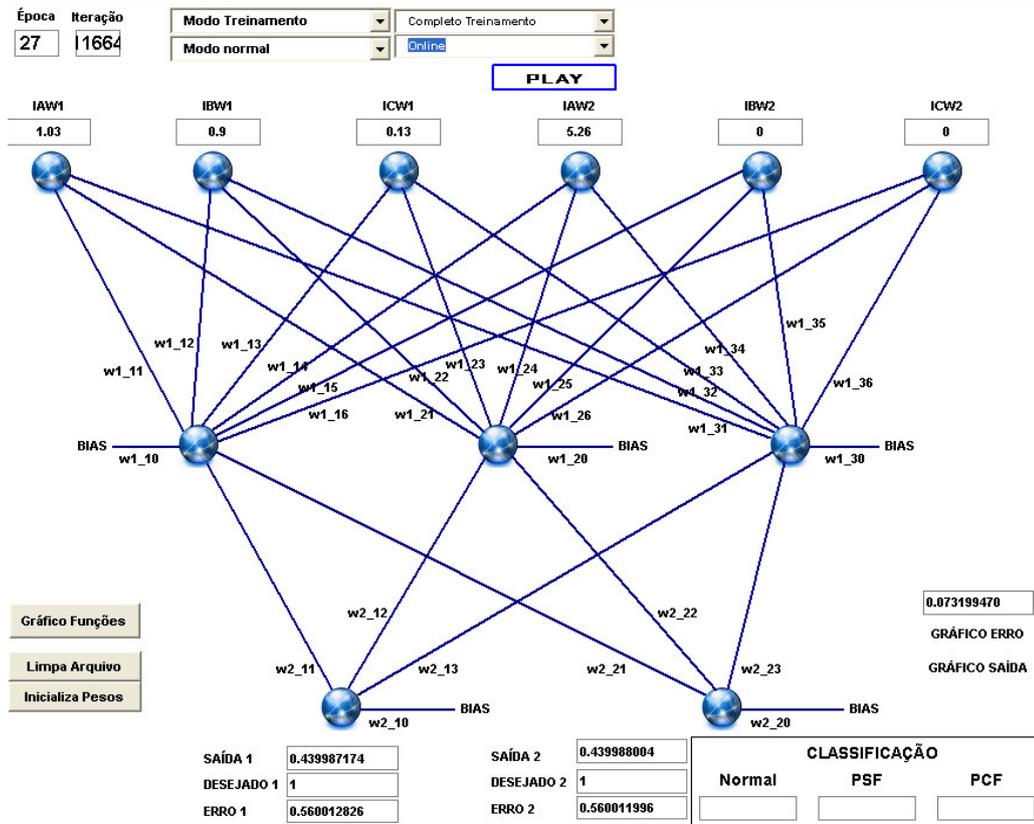


Figura 4.9: Etapa de Treinamento da Rede Neural.

Como forma de validar o treinamento da rede, dois pontos devem ser analisados:

1. Erro Quadrático Médio: Esta variável nos fornece uma idéia de como o erro se comporta ao longo das várias iterações de uma época. Como optamos por trabalhar com correção de erros no Modo *Online*, a cada iteração a retropropagação do erro ocorre e os pesos da rede são reajustados. Como estamos utilizando *VTR* como vetor de treinamento, 432 padrões serão apresentados à rede durante a fase de treinamento. Para cada padrão apresentado, a saída é calculada utilizando-se os pesos atuais da rede, comparada a saída desejada e o erro absoluto é calculado. A partir do segundo erro absoluto, já pode-se calcular o erro quadrático médio. Ao final das 432 iterações teremos o erro quadrático da época.
2. Erro de Época: fornece o erro quadrático médio de cada época, demons-

trando como o número de épocas influencia nos resultados. De fato, este parâmetro será utilizado como critério de parada de treinamento conforme descrito na seção 3.3.3 do Capítulo 3.

A Figura 4.10 ilustra o erro de época obtido na etapa de treinamento:

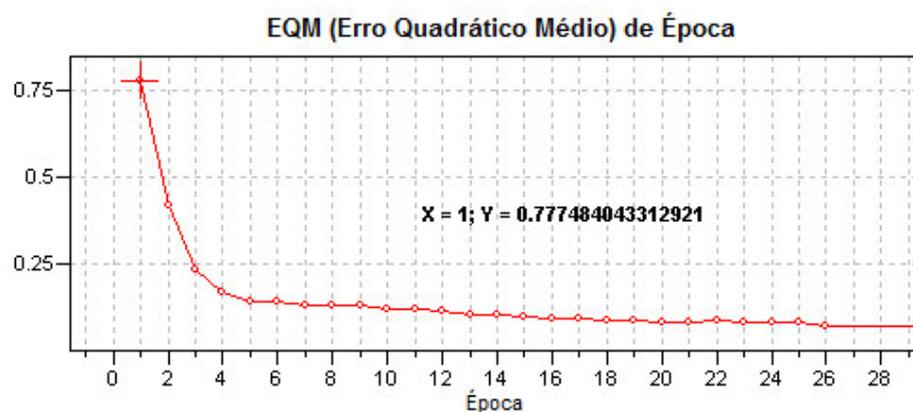


Figura 4.10: Erro da 1ª Época obtido durante a Fase de Treinamento da Rede Neural.

Percebe-se que após a 1ª época, o valor do erro é de aproximadamente 0,78. Já ao término da 14ª época o valor é inferior a 0,1, como verifica-se através da Figura 4.11. Adotando-se o critério de parada de treinamento, a partir da 14ª época não seria mais necessário continuar o treinamento da rede.

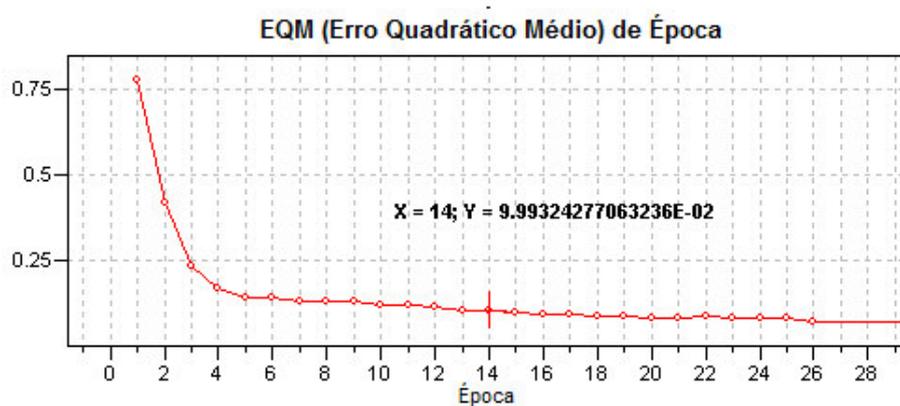


Figura 4.11: Erro da 14ª Época obtido durante a Fase de Treinamento da Rede Neural.

No entanto, manteve-se o treinamento para que fosse possível perceber a influência que um maior número de épocas poderia trazer. Em termos percentuais, é possível resumir na Tabela 4.1 a evolução do erro de época para este estudo de caso. Percebe-se que a partir da 16<sup>a</sup> época a redução do erro é pouco significativa.

Tabela 4.1: Erros de Época

Época	Erro	Variação a cada época	Variação com relação à 1ª época
1	0,7774	-	-
2	0,4175	-46,30%	-46,30%
3	0,23	-44,91%	-70,41%
4	0,1656	-28,00%	-78,70%
5	0,1386	-16,30%	-82,17%
6	0,1404	1,30%	-81,94%
7	0,1305	-7,05%	-83,21%
8	0,1267	-2,91%	-83,70%
9	0,1285	1,42%	-83,47%
10	0,115	-10,51%	-85,21%
11	0,1164	1,22%	-85,03%
12	0,111	-4,64%	-85,72%
13	0,1037	-6,58%	-86,66%
14	0,0999	-3,66%	-87,15%
15	0,0941	-5,81%	-87,90%
16	0,09208	-2,15%	-88,16%
17	0,0907	-1,50%	-88,33%
18	0,0858	-5,40%	-88,96%
19	0,0856	-0,23%	-88,99%
20	0,0794	-7,24%	-89,79%
21	0,0789	-0,63%	-89,85%
22	0,085	7,73%	-89,07%
23	0,0816	-4,00%	-89,50%
24	0,0799	-2,08%	-89,72%
25	0,0782	-2,13%	-89,94%
26	0,0713	-8,82%	-90,83%
27	0,0731	2,52%	-90,60%

Percebe-se que durante as três primeiras épocas a queda percentual do erro de

época é expressiva. A cada nova época o percentual de redução do erro torna-se menor e o que determinará que o treinamento deve ser encerrado é o critério de parada de treinamento escolhido. Na seção 3.3.3 do Capítulo 3, definiu-se que para um erro de época  $\leq 0,1$  o treinamento deveria ser interrompido, condição essa que é atingida na 14ª época do exemplo analisado. Vale ressaltar que um critério de parada de treinamento mais exigente que este (erro de época  $\leq 0,075$ , por exemplo) só seria alcançado na 26ª época, ou seja, quase o dobro de iterações seriam necessárias. Naturalmente que essa segunda condição resultaria em maior processamento computacional da aplicação. De forma geral, a definição do critério de parada de treinamento precisa equilibrar a qualidade do treinamento realizado (conforme as premissas e requisitos de projeto) e a boa performance computacional do conjunto hardware/*software* utilizado.

### 4.3 Análise de Desempenho da MLP

A análise de desempenho da MLP da RNA é realizada com os dados reservados para a fase de testes. Estes dados não são os mesmos utilizados na etapa de treinamento, mas também contêm os três tipos de eventos definidos neste projeto. Nesta etapa não existe nenhuma forma de correção da matriz de pesos; esta já foi calculada na fase de treinamento e a solução encontrada será agora utilizada para qualquer entrada que seja apresentada a RNA. O algoritmo para processar a rede permanece o mesmo, porém sem a etapa de retropropagação dos erros e correção dos pesos.

#### 4.3.1 Etapa de Testes

Nesta etapa utilizou-se o critério abaixo, respeitando a estratégia de treinamento descrita na seção 3.3.2 do Capítulo 3:

- Evento Normal -  $yN(10) < -0.5$  e  $yN(11) < -0.5$
- Evento PSF -  $-0.5 < yN(10) < 0.3$  e  $-0.5 < yN(11) < 0.3$
- Evento PCF -  $yN(10) > 0.3$  e  $yN(11) > 0.3$

Os testes, portanto, podem ser divididos em três etapas:

- Teste da rede apresentando um vetor de eventos do tipo NORMAL ( $VT_N$ ): cada posição do vetor  $VT_N$  será apresentada à rede e, naturalmente, espera-se que a rede classifique todos esses dados como característicos de evento NORMAL. A Figura 4.12 exibe o resultado desse teste; percebe-se no canto inferior direito da tela a classificação gerada pela RNA. Neste exemplo, a rede obteve sucesso ao classificar 100% dos dados como evento NORMAL.

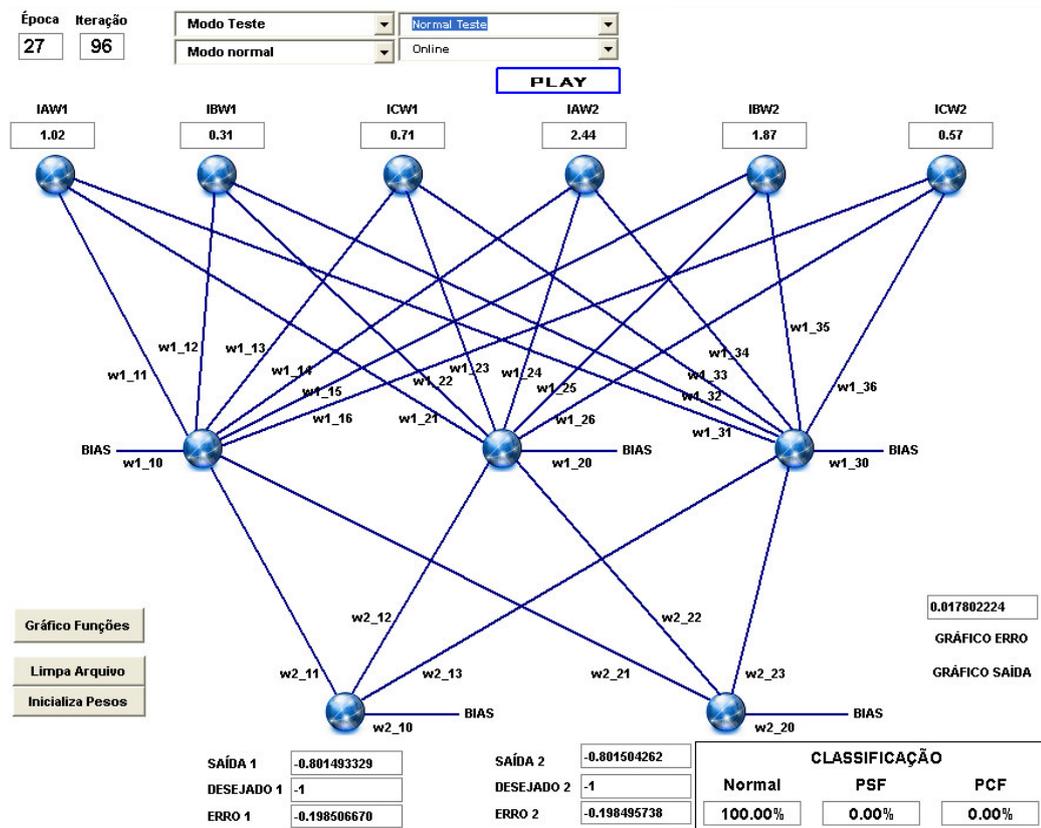


Figura 4.12: Teste de Evento do Tipo NORMAL.

- Teste da rede apresentando um vetor de eventos do tipo PSF ( $VT_{PSF}$ ): cada posição do vetor  $VT_{PSF}$  será apresentada à rede e, naturalmente, espera-se que a rede classifique todos esses dados como característicos de evento PSF. A Figura 4.13 exibe o resultado desse teste. Neste exemplo, a rede obteve sucesso ao classificar 97,92% dos dados como evento PSF.

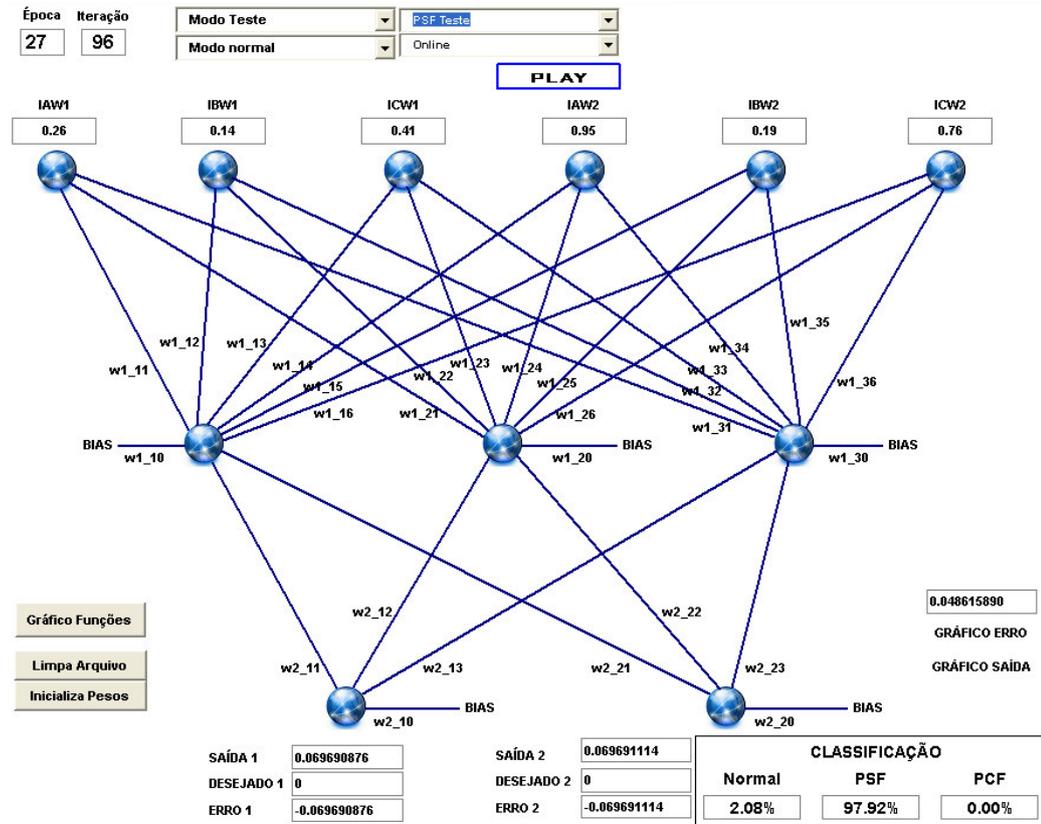


Figura 4.13: Teste de Evento do Tipo PSF.

- Teste da rede apresentando um vetor de eventos do tipo PCF ( $VT_{PCF}$ ): cada posição do vetor  $VT_{PCF}$  será apresentada à rede e, naturalmente, espera-se que a rede classifique todos esses dados como característicos de evento PCF. A Figura 4.14 exibe o resultado desse teste. Neste exemplo, a rede obteve sucesso ao classificar 100% dos dados como evento PCF. O resumo dos testes acima é ilustrado na Tabela 4.2.

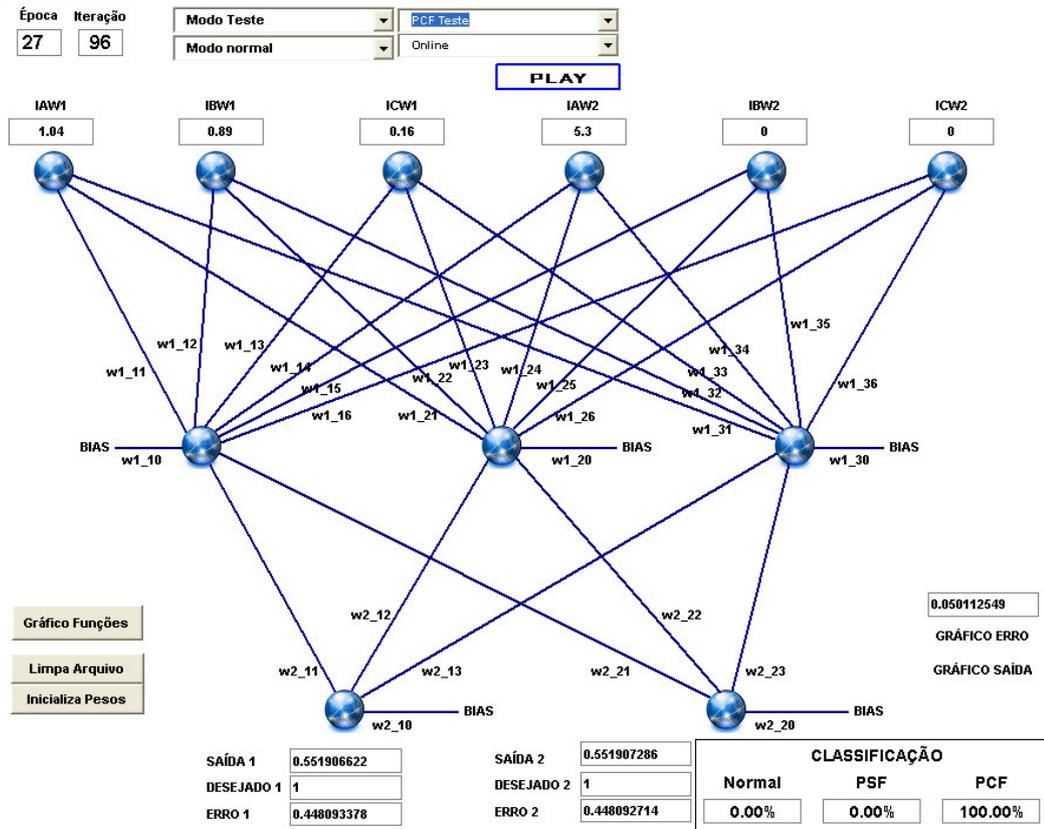


Figura 4.14: Teste de Evento do Tipo PCF.

Tabela 4.2: Resultados dos Testes da Rede Neural

Tipo de Evento	Classificação gerada pela RNA		
	NORMAL	PSF	PCF
NORMAL	100%	0%	0%
PSF	2,08%	97,92%	0%
PCF	0%	0%	100%

## 4.4 Conclusões e Comentários

Neste capítulo apresentou-se o desenvolvimento de um sistema para diagnóstico de falhas baseado em redes neurais integradas a um SCADA. Na Seção 4.3 apresentou-

se a análise de Desempenho da MLP, verificando-se que o critério de parada de treinamento escolhido neste projeto trouxe excelentes resultados, não sendo necessário estabelecer um segundo critério mais restritivo. De forma generalizada, resume-se o conteúdo do capítulo em experimentos computacionais que consistem no treinamento da rede, enquanto os experimentos de campo consistem no processamento *on-line* da rede MLP.

Ainda, apresentou-se o desenvolvimento do sistema SCADA na Seção 4.1 para embarcar a rede neuronal no sistema supervisório do SDEE. O software desenvolvido possui as funcionalidades de treinamento e processamento *on-line* da rede para detecção de padrões de falhas analisando dados aquisitados do relé do transformador. Este desenvolvimento foi apresentado por meio de telas de saída que mostram a topologia e configuração da rede e janelas que exibem os menus de opção do *software*.

A implementação da rede abordou desde a quantidade de neurônios da camada de saída, a função de ativação escolhida na configuração da rede MLP, até os aspectos do treinamento.

Os pesos sinápticos para as camadas intermediária e de saída foram apresentados como resultado do procedimento apresentado no Capítulo 3 para formação do conjunto de treinamento.

As funções de ativação tangente hiperbólica e sigmoideal foram avaliadas. Ao utilizar-se a tangente hiperbólica, os testes empíricos mostraram que a convergência do processo de treinamento, avaliada através do erro quadrático médio de época, não ocorria. Melhores resultados, porém, foram obtidos utilizando a função sigmoideal na camada intermediária e a função soma ponderada na camada de saída, conduzindo a saída da rede aos valores desejados.

O treinamento foi avaliado em termos de critérios de erros quadrático e de época, concluiu-se que existe um conflito quanto a escolha do critério, a qualidade está relacionada com o erro quadrático médio de época e o desempenho computacional com o número de épocas.

A análise de desempenho da rede neuronal proposta foi realizada através de testes executados utilizando eventos reais coletados de IEDs; os eventos selecionados correspondem aos tipos normal, PSF e PCF e foram corretamente classificados pela RNA.

## CAPÍTULO 5

# Conclusões

---

Nesta pesquisa demonstrou-se que a RNA é uma eficiente ferramenta para reconhecimento de padrões em sistemas de distribuição de energia elétrica. Uma aplicação foi desenvolvida comprovando a contribuição dessa metodologia na detecção de anomalias e no auxílio à tomada de decisão. Neste estudo, desenvolveu-se um sistema que integra uma RNA a um SCADA, auxiliando operadores do SDEE seja através de alarmes indicando comportamentos anormais da rede que podem evoluir para faltas ou ainda fornecendo suporte à tomada de decisão em caso de faltas inevitáveis. Nesse caso, a aplicação apresenta ações direcionadas à área de Manutenção para promover o rápido restabelecimento do sistema após a ocorrência. Reduzir o tempo de restabelecimento pós-falta é um grande desafio do setor elétrico e essa pesquisa representa um significativo salto de evolução tanto para distribuidoras de energia elétrica quanto para clientes. Além disso, o sistema desenvolvido permite o constante aprendizado, recebendo como *feedback* a constante avaliação da área de Estudos da concessionária quanto à classificação das anomalias feita pela RNA. Em caso de uma classificação estar incorreta, a amostra de dados é novamente apresentada à rede, em modo treinamento, para que as características do evento sejam então aprendidas.

### 5.1 Trabalhos Futuros

Como trabalhos a serem desenvolvidos a partir desta dissertação, propõe-se a integração da informação gerada pela RNA a um sistema especialista. Dessa

forma, poderia-se aproveitar a classificação de dados gerada pela rede como um banco de dados altamente confiável para o sistema especialista. Este, por sua vez, seria responsável pela análise dos dados do sistema em uma janela de tempo bem maior que a da RNA e pela tomada de decisão no que refere a ações a longo prazo e menos imediatistas, como por exemplo, a escolha de que áreas da manutenção a empresa deveria se dedicar mais por estarem gerando maior impacto nos indicadores de DEC e FEC. Outra forma de dar continuidade aos resultados desta dissertação seria realizar uma abordagem semelhante, porém utilizando uma RNA que convergisse informações dos vários equipamentos de uma subestação no momento da ocorrência do evento. Seria necessário um esforço no sentido de garantir o sincronismo de tempo entre os diversos equipamentos para que fosse possível assim analisar o mesmo evento a partir de diferentes IEDs. Os resultados obtidos certamente seriam de significativo valor acadêmico e empresarial.

## APÊNDICE A

# Código desenvolvido no *software* Eclipse E3 Power®

---

```
'Início de código de treinamento/teste da RNA

saidacompleta = Application.GetObject("Dados.Treinamento.SaidaCompleto").Value
saidacompleta2 = Application.GetObject("Dados.saidaFaltaCompleto").Value
saidaportaAND = Application.GetObject("Dados.vetorsaidaportaAND").Value

If Screen.Item("ComboBox1").Value = "Modo Treinamento" Then
Treinamento = true
ElseIf Screen.Item("ComboBox1").Value = "Modo Teste" Then
Treinamento = false
End If
ModoErro = Screen.Item("ComboBox5").Value

If Screen.Item("ComboBox3").Value = "Normal 1" Then
vetoreentrada = Application.GetObject("Dados.ArrayCEVNormal1").Value
ElseIf Screen.Item("ComboBox3").Value = "Pert. sem falta 1" Then
vetoreentrada = Application.GetObject("Dados.ArrayCEVPSF1").Value
ElseIf Screen.Item("ComboBox3").Value = "Pert. com falta 1" Then
vetoreentrada = Application.GetObject("Dados.ArrayCEVPCF1").Value
ElseIf Screen.Item("ComboBox3").Value = "Completo" Then
```

```
vetoreentrada = Application.GetObject("Dados.ArrayCEVCompleto").Value
ElseIf Screen.Item("ComboBox3").Value = "Porta AND" Then
vetoreentrada = Application.GetObject("Dados.vetorportaAND").Value
ElseIf Screen.Item("ComboBox3").Value = "Falta A" Then
vetoreentrada = Application.GetObject("Dados.vetorFaltaA").Value
ElseIf Screen.Item("ComboBox3").Value = "Falta B" Then
vetoreentrada = Application.GetObject("Dados.vetorFaltaB").Value
ElseIf Screen.Item("ComboBox3").Value = "Falta AB" Then
vetoreentrada = Application.GetObject("Dados.vetorFaltaAB").Value
ElseIf Screen.Item("ComboBox3").Value = "Falta Completo" Then
vetoreentrada = Application.GetObject("Dados.vetorFaltaCompleto").Value
ElseIf Screen.Item("ComboBox3").Value = "Normal 2" Then
vetoreentrada = Application.GetObject("Dados.ArrayCEVNormal2").Value
ElseIf Screen.Item("ComboBox3").Value = "Pert. sem falta 2" Then
vetoreentrada = Application.GetObject("Dados.ArrayCEVPSF2").Value
ElseIf Screen.Item("ComboBox3").Value = "Pert. com falta 2" Then
vetoreentrada = Application.GetObject("Dados.ArrayCEVPCF2").Value
ElseIf Screen.Item("ComboBox3").Value = "Completo Treinamento" Then
vetoreentrada = Application.GetObject("Dados.Treinamento.Completo").Value
ElseIf Screen.Item("ComboBox3").Value = "Normal Teste" Then
vetoreentrada = Application.GetObject("Dados.Teste.Normal").Value
ElseIf Screen.Item("ComboBox3").Value = "PSF Teste" Then
vetoreentrada = Application.GetObject("Dados.Teste.PSF").Value
ElseIf Screen.Item("ComboBox3").Value = "PCF Teste" Then
vetoreentrada = Application.GetObject("Dados.Teste.PCF").Value
End If
```

```
ydes=Application.GetObject("Dados.Treinamento.SaidaCompleto").Value
```

```
VetorErro1 = Array(0)
```

```
VetorErro2 = Array(0)
```

```
VetorErroQuadInst =Array(0)
```

```
VetorErroQuadMedio =Array(0)
```

```
VetorSaida1 = Array(0)
VetorSaida2 = Array(0)

for pos =1 to ubound(vetoreentrada)

If Screen.Item("ComboBox3").Value = "Normal 1" Then
ydes (pos)(0)= -1
ydes (pos)(1)= -1
ElseIf Screen.Item("ComboBox3").Value = "Pert. sem falta 1" Then
ydes (pos)(0) = 0
ydes (pos)(1)= 0
ElseIf Screen.Item("ComboBox3").Value = "Pert. com falta 1" Then
ydes (pos)(0) = 1
ydes (pos)(1)= 1

ElseIf Screen.Item("ComboBox3").Value = "Completo" Then
ydes (pos) = saidacompleta (pos)

ElseIf Screen.Item("ComboBox3").Value = "Porta AND" Then
ydes (pos) = saidaportaAND (pos)

ElseIf Screen.Item("ComboBox3").Value = "Falta A" Then
ydes (pos)(0)= 1
ydes (pos)(1)= 0

ElseIf Screen.Item("ComboBox3").Value = "Falta B" Then
ydes (pos)(0)= 0
ydes (pos)(1)= 1

ElseIf Screen.Item("ComboBox3").Value = "Falta AB" Then
ydes (pos)(0)= 1
ydes (pos)(1)= 1
```

```
ElseIf Screen.Item("ComboBox3").Value = "Falta Completo" Then
ydes (pos) = saidacompleta2 (pos)

ElseIf Screen.Item("ComboBox3").Value = "Normal 2" Then
ydes (pos)(0)= -1
ydes (pos)(1)= -1

ElseIf Screen.Item("ComboBox3").Value = "Pert. sem falta 2" Then
ydes (pos)(0) = 0
ydes (pos)(1)= 0

ElseIf Screen.Item("ComboBox3").Value = "Pert. com falta 2" Then
ydes (pos)(0) = 1
ydes (pos)(1)= 1

ElseIf Screen.Item("ComboBox3").Value = "Completo Treinamento" Then
ydes (pos) = saidacompleta (pos)

ElseIf Screen.Item("ComboBox3").Value = "Normal Teste" Then
ydes (pos)(0)= -1
ydes (pos)(1)= -1
ElseIf Screen.Item("ComboBox3").Value = "PSF Teste" Then
ydes (pos)(0)= 0
ydes (pos)(1)= 0
ElseIf Screen.Item("ComboBox3").Value = "PCF Teste" Then
ydes (pos)(0)= 1
ydes (pos)(1)= 1

End If

Redim Preserve VetorErro(pos)
Redim Preserve VetorErro1(pos)
```

```
Redim Preserve VetorErro2(pos)
Redim Preserve VetorErroQuadInst(pos)

VetorErro1 (pos) = 0
VetorErro2 (pos) = 0
VetorErroQuadInst (pos) = 0

Redim Preserve VetorSaida1(pos)
Redim Preserve VetorSaida2(pos)
VetorSaida1 (pos) = 0
VetorSaida2 (pos) = 0
next

bias = 0.01
somaN=Array (0,0,0,0,0,0,0,0,0,0,0,0,0)
yN=Array (0,0,0,0,0,0,0,0,0,0,0,0,0)
totnormal = 0
totpsf = 0
totpcf = 0
a = 1.71
b=2/3

W1 = Application.GetObject("Dados.MatrizPesosW1").Value
W2 = Application.GetObject("Dados.MatrizPesosW2").Value

n = 0.1

iteracoes = 0

tamanhoamostra = ubound (vetoreentrada)

If Treinamento = True Then
epocas= Screen.Item("Texto66").Value
```

```

Else
epocas= 1
End If

aux = Array(0)
For t=1 to epocas

aux(0) = "Epoca"&t
for r=1 to tamanhoamostra
Redim Preserve aux (r)
aux (r) = Array(r,r)
Next
Redim Preserve VetorErroQuadMedio(t)
VetorErroQuadMedio (t) = aux

For i=1 to tamanhoamostra

Screen.Item("TextoN1").Value = vetoreentrada(i)(0)
Screen.Item("TextoN2").Value = vetoreentrada(i)(1)
Screen.Item("TextoN3").Value = vetoreentrada(i)(2)
Screen.Item("TextoN4").Value = vetoreentrada(i)(3)
Screen.Item("TextoN5").Value = vetoreentrada(i)(4)
Screen.Item("TextoN6").Value = vetoreentrada(i)(5)
Screen.Item("Texto3").Value = iteracoes+1

,,,,,,,,,,,,,,,,,,,,,,,,,CAMADA INTERMEDIARIA,,,,,,,,,,,,,,,,,,,,,,,,,

for k=7 to 9
'soman7 = bias * w10 + 1* w11 + 2*w12 + 3 *w13 + 4 *w14 + 5*w15 +6*w16
somaN(k) = bias*W1(k-7)(0)
for m=0 to 5
somaN(k)=somaN(k)+vetoreentrada(i)(m)*W1(k-7)(m+1)
next
yN(k) = 1/(1+Exp(-somaN(k)))  '' funcao sigmoidal

```

```

next

,,,,,,,,,,,,,,,,,,,,,,,,,CAMADA SAÍDA,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

for k=10 to 11
somaN(k) = bias*W2(k-10)(0)
for m=0 to 2
somaN(k)=somaN(k)+yN(7+m)*W2(k-10)(m+1)
next
yN(k)=somaN(k)
next

Erro1 = ydes(i)(0) - yN(10)
Erro2 = ydes(i)(1) - yN(11)
ErroQuad = 1/2*((Erro1)^2+(Erro2)^2)
VetorErroQuadInst (i) = Array(i,ErroQuad)
ErroQuadMedio=0
for q=1 to i
ErroQuadMedio = ErroQuadMedio + VetorErroQuadInst(q)(1)
Next
ErroQuadMedio = ErroQuadMedio/i

VetorErroQuadMedio(t)(i) = Array(i,ErroQuadMedio)

If yN(10)<-0.5 and yN(11)<-0.5 Then
totnormal = totnormal + 1
ElseIf yN(10)>-0.5 and yN(10)<0.3 and yN(11)>-0.5 and yN(11)<0.3 Then
totpsf = totpsf + 1
ElseIf yN(10)>0.3 and yN(11)>0.3 Then
totpcf = totpcf + 1

End If

```

```
Screen.Item("Texto60").Value = yN(10)
Screen.Item("Texto73").Value = yN(11)
Screen.Item("Texto64").Value = ydes(i)(0)
Screen.Item("Texto70").Value = ydes(i)(1)
Screen.Item("Texto62").Value = Erro1
Screen.Item("Texto67").Value = Erro2
Screen.Item("Texto11").Value = ErroQuadMedio
VetorErro1 (i) = Array(i,Erro1)
VetorErro2 (i) = Array(i,Erro2)
VetorSaida1(i) = Array(i,yN(10))
VetorSaida2(i) = Array(i,yN(11))

If Screen.Item("ComboBox2").Value = "Modo passo-a-passo" Then
MsgBox("Continuar")
End If

If (Abs(ErroQuadMedio) > 0 and Treinamento=True and ModoErro="Online") Then
ReajustaPesos W1,W2,n,Erro1,Erro2,bias,yN,vetorentrada,i
End If

iteracoes = iteracoes + 1

Next ' fim da variação de i

if (Abs(ErroQuadMedio) > 0 and Treinamento=True and ModoErro="Batelada") Then
ReajustaPesos W1,W2,n,Erro1,Erro2,bias,yN,vetorentrada,i-1
End If
Next

Application.GetObject("Dados.ArrayErrosQuad").Value = VetorErroQuadInst
Application.GetObject("Dados.ArrayErrosQuadMedio").Value = VetorErroQuadMedio
Application.GetObject("Dados.ArraySaida1").Value = VetorSaida1
```

```
Application.GetObject("Dados.ArraySaida2").Value = VetorSaida2

for i=0 to ubound(W1)
str4 = str4 & "Array("
for j= 0 to ubound(W1(i))
str4 = str4 & W1(i)(j) & ", "
Next
str4 = Mid (str4,1,Len(str4)-1)
str4 = str4 & "), "
Next
str4 = Mid (str4,1,Len(str4)-1)
str4 = "Array("&str4 & ")"

for i=0 to ubound(W2)
str5 = str5 & "Array("
for j= 0 to ubound(W2(i))
str5 = str5 & W2(i)(j) & ", "
Next
str5 = Mid (str5,1,Len(str5)-1)
str5 = str5 & "), "
Next
str5 = Mid (str5,1,Len(str5)-1)
str5 = "Array("&str5 & ")"

set fonte1 =
Application.GetObject("Dados.MatrizPesosW1").Links.CreateLink("Value", str4, 0)
set fonte2 =
Application.GetObject("Dados.MatrizPesosW2").Links.CreateLink("Value", str5, 0)

If Treinamento = True Then
Arg = Array(W1,W2)
Application.GetFrame("abcdN").OpenScreen("Solucao"), Arg
```



```

for m=1 to 3
W1(m-1)(0)=
W1(m-1)(0)+n*yN(6+m)*(1-yN(6+m))*(Erro1*W2(0)(m)+Erro2*W2(1)(m))*bias
for p=1 to 6
W1(m-1)(p)=
W1(m-1)(p)+n*yN(6+m)*(1-yN(6+m))*(Erro1*W2(0)(m)
+Erro2*W2(1)(m))*vetoreentrada(i)(p-1)
Next
next

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,REAJUSTE DOS PESOS DA MATRIZ W2,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
'delta_saida = n * FAT' * Erro * y_j
'peso_atual = peso_antigo + delta_saida

for m=1 to 2
if m=1 Then
Erro = Erro1
ElseIf m=2 Then
Erro=Erro2
End If
W2(m-1)(0)=W2(m-1)(0)+n * 1 * Erro * bias
for p=1 to 3
W2(m-1)(p)=W2(m-1)(p)+n * 1 * Erro * yN(6+p)
Next
next

End Sub
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
Sub SalvaEmArquivo (vetorArq)
'inicio de salvar novo teste no arquivo
Const ForReading = 1, ForWriting = 2, ForAppending = 8

```

```
NomeArquivo = "C:\pesosRNA.txt"
Set fso = CreateObject("Scripting.FileSystemObject")

Set fr = fso.OpenTextFile(NomeArquivo, ForReading, True)

Do until fr.AtEndOfStream
linha = fr.ReadLine
Loop

indice = Mid (linha,1,1)
indice = indice + 1
indicestr = CStr (indice)
totnormalpercentstr = CStr(vetorArq(4))
totpsfpercentstr = CStr(vetorArq(5))
totpcfpercentstr = CStr(vetorArq(6))
Set fw = fso.OpenTextFile(NomeArquivo,8)
fw.WriteLine indicestr + vbTab + vetorArq(0) + vbTab + vetorArq(1)
+ vbTab + vetorArq(2) + vbTab + vetorArq(3) + vbTab + totnormalpercentstr
+ vbTab + totpsfpercentstr + vbTab + totpcfpercentstr
fw.Close

'fim de salvar novo teste no arquivo

'Fim de código de treinamento/teste da RNA
```

## Referências Bibliográficas

- Braga, A.P., A.P.L.F. Carvalho and T.B. Ludemir (1998). Fundamentos de redes neurais artificiais. *11ª Escola de Computação*.
- Butler, K.L. and J.A. Momoh (1993). Detection and classification of line faults on power distribution systems using neural networks. In: *Circuits and Systems, 1993., Proceedings of the 36th Midwest Symposium on*. pp. 368–371 vol.1.
- Calôba, L.P. (1995). Introdução à computação neuronal. *Anais do 9o Congresso Brasileiro de Automação; Vitória- ES* pp. 25–38.
- Cardoso, R.S., L.H. Xavier, C.F.S. Gomes and P.J. Adissi (2009). Uso de sad no apoio à decisão na destinação de resíduos plásticos e gestão de materiais. *Pesqui. Oper. [online]*. **29**(1), 67–95.
- Carvalho, A.P.L.F. (2010). Perceptron multi-camadas. *Departamento de Ciência da Computação da USP-SP*. Disponível em <http://www.icmc.usp.br/andre/-research/neural/MLP.htm>. Acessado em outubro de 2010.
- Coser, J., D.T. do Vale and J.G. Rolim (2005). Artificial neural network based method for fault location in distribution systems. In: *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference on*. p. 6 pp.
- Fernandes, R.G. (2007). Detecção e isolamento de falhas em sistemas dinâmicos baseados em redes neurais. *Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte*.

- Giboshi, M.L., L.H.A. Rodrigues and F.L. Neto (2006). Sistema de suporte à decisão para recomendação de uso e manejo da terra. *Revista Brasileira de Engenharia Agrícola e Ambiental* **10**(4), 861–866.
- Hang, X., L. Huiming and L. Jie (2008). Research on application of artificial neural network in the network loss calculation of power system. In: *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*. pp. 318 –320.
- Horie, A.K., H. Brizard and R.H.G. Guiraldelli (2008). Redes neurais multicamadas. *Seminário de Inteligência Artificial, Escola Politécnica da Universidade de São Paulo*.
- Laboratories, Schweitzer Engineering (2004). Sel-387a current differential relay - instruction manual.
- Oleskovicz, M., D.V. Coury and R.K. Aggarwal (2003). O emprego de redes neurais artificiais na detecção, classificação e localização de faltas em linhas de transmissão. *Sba Controle & Automação*.
- Oliveira, A.R. de (2005). Redes neurais artificiais aplicadas na detecção, classificação e localização de defeitos em linhas de transmissão. *Dissertação de Mestrado. Universidade Federal de Juiz de Fora*.
- Oliveira, M.O., D.S. Gazzana and A.S. Bretas (2010). Sistema especialista para detecção e diagnóstico de faltas em plantas de geração termoeletrica. *XVIII Congresso Brasileiro de Automática. Bonito-MS*.
- Pagliosa, A.L. (2003). Obtenção das funções de pertinência de um sistema neuro-fuzzy modificado pela rede de kohonen. *Dissertação de Mestrado. Universidade do Estado de Santa Catarina - UDESC*.
- Raybolt, A.C.S. (2011). Localização de faltas em linhas de transmissão em circuito duplo utilizando redes neurais artificiais. *Dissertação de Mestrado. Universidade Federal do Rio de Janeiro*.
- Salazar, H., R. Gallego and R. Romero (2006). Artificial neural networks and clustering techniques applied in the reconfiguration of distribution systems. *Power Delivery, IEEE Transactions on* **21**(3), 1735 –1742.

- Santos, A.M., J.M. Seixas, B.B. Pereira and R.A. Medronho (2005). Usando redes neurais artificiais e regressão logística na predição da hepatite a. *Rev Bras Epidemiol* **8**(2), 117–26.
- Silva, K.M., B.A. Souza, N.S.D. Brito, K.M.C. Dantas, F.B. Costa and S.S.B. Silva (2007). Detecção e classificação de faltas a partir da análise de registros oscilográficos via redes neurais artificiais e transformada wavelet. *Sba Controle & Automação*.
- Spatti, D.H. (2011). Automatização de processos de detecção de faltas em linhas de distribuição utilizando sistemas especialistas híbridos. *Tese de Doutorado. Universidade de São Paulo*.
- Stergiopoulos, K., A.G. Pipe and H. Nouri (2003). Intelligent control architectures for fault diagnosis in electrical power distribution networks. In: *Intelligent Control. 2003 IEEE International Symposium on*. pp. 569–573.
- Teixeira Junior, R.F. and Fernandes, F.C.F. and N.A. Pereira (2006). Sistema de apoio à decisão para programação da produção em fundições de mercado. *Gest. Prod. [online]* **13**(2), 205–221.
- Tinós, R. and M.H. Terra (2001). Fault detection and isolation in robotic manipulators using a multilayer perceptron and a rbf network trained by the kohonen’s self-organizing map. *Revista SBA* **12**(1), 11–18.
- William, E.J., M. Manic and B.K. Johnson (2007). Ann relays used to determine fault locations on shipboard electrical distribution systems. In: *Power Symposium, 2007. NAPS '07. 39th North American*. pp. 143–147.