

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

DISSERTAÇÃO DE MESTRADO

COMPARAÇÃO DOS ALGORITMOS C4.5 E MLP USADOS NA
AVALIAÇÃO DA SEGURANÇA DINÂMICA E NO AUXÍLIO AO CONTROLE
PREVENTIVO NO CONTEXTO DA ESTABILIDADE TRANSITÓRIA DE
SISTEMAS DE POTÊNCIA

WERBESTON DOUGLAS DE OLIVEIRA

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
Março/2013

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

COMPARAÇÃO DOS ALGORITMOS C4.5 E MLP USADOS NA
AVALIAÇÃO DA SEGURANÇA DINÂMICA E NO AUXÍLIO AO CONTROLE
PREVENTIVO NO CONTEXTO DA ESTABILIDADE TRANSITÓRIA DE
SISTEMAS DE POTÊNCIA

WERBESTON DOUGLAS DE OLIVEIRA

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
Março/2013

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

COMPARAÇÃO DOS ALGORITMOS C4.5 E MLP USADOS NA
AVALIAÇÃO DA SEGURANÇA DINÂMICA E NO AUXÍLIO AO CONTROLE
PREVENTIVO NO CONTEXTO DA ESTABILIDADE TRANSITÓRIA DE
SISTEMAS DE POTÊNCIA

WERBESTON DOUGLAS DE OLIVEIRA

Dissertação submetida à Banca
Examinadora do Programa de Pós-
Graduação em Engenharia Elétrica da
UFPA para a obtenção do Grau de Mestre
em Engenharia Elétrica.

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
Março/2013

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

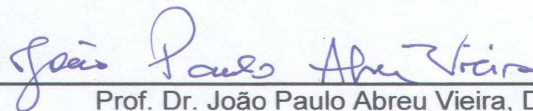
“COMPARAÇÃO DOS ALGORITMOS C4.5 E MLP USADOS NA AVALIAÇÃO DA SEGURANÇA DINÂMICA E NO AUXÍLIO AO CONTROLE PREVENTIVO NO CONTEXTO DA ESTABILIDADE TRANSITÓRIA DE SISTEMAS DE POTÊNCIA”

AUTOR: WERBESTON DOUGLAS DE OLIVEIRA

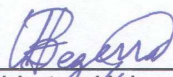
DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE SISTEMAS DE ENERGIA.

APROVADA EM: 06 / 03 / 2013

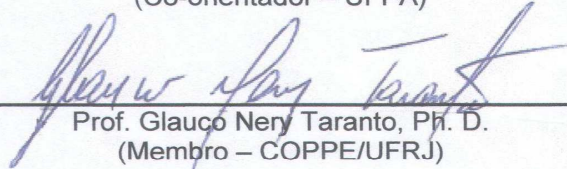
BANCA EXAMINADORA:



Prof. Dr. João Paulo Abreu Vieira, Dr. Eng.
(Orientador – UFPA)



Prof. Dr. Ubiratan Holanda Bezerra, Dr. Sc.
(Co-orientador – UFPA)



Prof. Glauco Nery Taranto, Ph. D.
(Membro – COPPE/UFRJ)



Prof. Roberto Célio Limão de Oliveira, Dr. Eng.
(Membro – UFPA)

VISTO:

Prof. Evaldo Gonçalves Pelaes, Dr. Eng.
(Coordenador do PPGE/ITEC/UFPA)

DEDICATÓRIA

*A todos os meus
familiares pelos ensinamentos,
paciência e incentivo por toda
a minha vida.*

AGRADECIMENTOS

Aos professores João Paulo Abreu Vieira e Ubiratan Holanda Bezerra, por todo apoio, confiança e amizade. Por permitir fazer parte de um grupo de pesquisadores que cresce a cada dia. Pela dedicação e ajuda na orientação deste trabalho, e pela influência e incentivo na busca por informação.

Aos meus familiares e amigos, que se listados aqui nomes e ações que direta ou indiretamente contribuíram para esse momento, as páginas de agradecimentos seriam infindáveis.

Ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará, com seus professores e funcionários pelo suporte dado nesses anos de curso.

A CAPES pelo suporte financeiro e a Universidade Federal do Pará pelo acesso aos meios essenciais para o desenvolvimento desta Dissertação.

Aos professores e amigos, companheiros de todas as horas, pela amizade e momentos felizes que me proporcionaram. Em especial aos amigos do Ceamazon pelo acolhimento nesses dois anos.

A todos os citados meus mais sinceros agradecimentos por fazerem parte da minha vida.

RESUMO

Esse trabalho compara os algoritmos C4.5 e MLP (do inglês “*Multilayer Perceptron*”) aplicados a avaliação de segurança dinâmica ou (DSA, do inglês “*Dynamic Security Assessment*”) e em projetos de controle preventivo, com foco na estabilidade transitória de sistemas elétricos de potência (SEPs). O C4.5 é um dos algoritmos da árvore de decisão ou (DT, do inglês “*Decision Tree*”) e a MLP é um dos membros da família das redes neurais artificiais (RNA). Ambos os algoritmos fornecem soluções para o problema da DSA em tempo real, identificando rapidamente quando um SEP está sujeito a uma perturbação crítica (curto-circuito, por exemplo) que pode levar para a instabilidade transitória. Além disso, o conhecimento obtido de ambas as técnicas, na forma de regras, pode ser utilizado em projetos de controle preventivo para restaurar a segurança do SEP contra perturbações críticas. Baseado na formação de base de dados com exaustivas simulações no domínio do tempo, algumas perturbações críticas específicas são tomadas como exemplo para comparar os algoritmos C4.5 e MLP empregadas a DSA e ao auxílio de ações preventivas. O estudo comparativo é testado no sistema elétrico “*New England*”. Nos estudos de caso, a base de dados é gerada por meio do programa PSTv3 (“*Power System Toolbox*”). As DTs e as RNAs são treinada e testadas usando o programa Rapidminer. Os resultados obtidos demonstram que os algoritmos C4.5 e MLP são promissores nas aplicações de DSA e em projetos de controle preventivo.

PALAVRAS-CHAVES: Estabilidade transitória, avaliação da segurança dinâmica, controle preventivo, sistemas elétrico de potência, árvore de decisão e redes neurais.

ABSTRACT

This work compares the C4.5 and multilayer perceptron (MLP) algorithms applied for dynamic security assessment (DSA) and power system stability transient preventive control design. C4.5 is an algorithm of the decision tree (DT) technique and the MLP is a member of artificial neural network (ANNs) family. The advent of DTs and ANNs provides solution to real-time DSA issues in order to identify quickly when a power system is subjected to a critical disturbance (short-circuit) that may lead to transient instability. In addition, the knowledge obtained by both techniques can be utilized in the preventive control design to restore the power system security against critical disturbances. Based on the data base generation with exhaustive time-domain simulations, some specific critical disturbances are taken as examples to compare the C4.5 and MLP algorithms employed to DSA and guideline to preventive actions. The comparative study is tested on the New England power system. In the case studies, the knowledge database is generated by using PSTv3 (Power System Toolbox) software. The DTs and ANNs are trained and tested by the Rapidminer software. The obtained results have demonstrated a promising application of the C4.5 and MLP algorithms used in power system DSA and preventive control design.

KEYWORDS: Transient stability, dynamic security assessment, preventive control, power system, decision tree and neural networks.

Sumário

Capítulo 1 - Introdução.....	1
1.1 Considerações Iniciais.....	1
1.2 – Objetivo Geral.....	2
1.3 - Principais Contribuições.....	2
1.4 - Revisão Bibliográfica.....	2
1.5 - Estrutura da Dissertação.....	4
Capítulo 2 – Segurança de Sistemas de Potência.....	5
2.1 Introdução.....	5
2.3 – Segurança Dinâmica de Sistemas Elétricos de Potência.....	7
2.3.1 – Estabilidade transitória.....	9
2.4 - Desafios para Operação Segura.....	12
2.5 - Avaliação da Segurança Dinâmica em Tempo Real.....	13
2.6 - Controle preventivo:.....	17
Capítulo 3 – Sistemas Inteligentes.....	20
3.1 – Introdução.....	20
3.2 – Aprendizado de Máquinas.....	21
3.3 - Árvore de Decisão (Paradigma Simbólico).....	24
3.3.1 – Algoritmo de Árvores de Decisão (C4.5).....	26
3.4.1 – MLP - Rede neural Perceptron de Múltiplas Camadas.....	32
3.4.2 – Algoritmo <i>Backpropagation</i> (Retropropagação).....	34
3.5 - Estimação de Erro do Classificador.....	37
3.5.1 - HoldOut.....	39
3.5.2 - Validação Cruzada (<i>Cross-Validation</i>).....	40
3.6 - Extração de Conhecimento em Redes Neurais.....	41
Capítulo 4 – Metodologia para DSA e Projeto de Controle Preventivo.....	43
4.1 - Introdução.....	43

4.2 - Ferramentas de Análise.....	44
4.2.1 - PSTv3.....	44
4.2.2 - Rapidminer.....	45
4.3 - Construção da Base de Dados.....	46
4.3.1 - Seleção das Condições de Operação.....	46
4.3.2 - Seleção das Contingências.....	47
4.3.3 - Detecção de Ilhamento.....	47
4.3.4 - Rotulação	47
4.3.5 - Atributos Selecionados (Base de Dados).....	48
Capítulo 5 - Resultados e Análise	51
5.1 Introdução.....	51
5.2 - Estudos de Caso	51
5.2.1 - DSA via Sistemas Inteligentes.....	53
5.2.1.1 - DSA utilizando DT	53
5.2.1.2 - DSA utilizando RNA	56
5.2.2 – Projeto de Controle Preventivo via Sistemas Inteligentes	57
5.2.2.1 – Projeto de Controle Preventivo Via RNA	59
5.2.2.2 – Projeto de Controle Preventivo via DT	63
5.2.2.4 – Discussão.....	73
Capítulo 6 - Conclusões.....	75
6.1 - Considerações Finais	75
6.2 – Sugestões para Trabalhos Futuros.....	76
Referências Bibliográficas	77
Anexo A - Conjunto de Pesos das RNAs.....	81
Anexo B - Regras das árvores de decisão.....	84

Lista de Figuras

Figura 2.1 -	Curvas PV.....	6
Figura 2.2 -	Classificação de Estabilidade de Sistemas de Potência.....	8
Figura 2.3 -	Resposta no tempo do ângulo do rotor frente a um distúrbio transitório.	10
Figura 2.4 -	Avaliação em tempo real de SEP.....	14
Figura 2.5 -	Subproblemas da análise on-line de segurança dinâmica.....	17
Figura 2.6 -	Nomograma de Segurança.....	18
Figura 3.1 -	Áreas Relacionadas com a Inteligência Artificial.....	21
Figura 3.2 -	Hierarquia de aprendizado.....	22
Figura 3.3 -	Árvore de Decisão com caminho de classificação.....	25
Figura 3.4 -	Modelo do neurônio artificial.....	29
Figura 3.5 -	Rede MLP com uma camada oculta.....	33
Figura 3.6 -	Método HoldOut.....	40
Figura 3.7 -	Exemplo de Cross-Validation.....	40
Figura 3.8 -	Identificação de entradas significantes em uma MLP.....	42
Figura 4.1 -	Características da Interface do RapidMiner 5.0.....	45
Figura 4.2 -	Fluxograma de passos da simulação.....	49
Figura 5.1 -	Diagrama Unifilar do Sistema Elétrico New England (10 Geradores 39 Barras) e divisão da base de dados.....	52
Figura 5.2 -	Árvore de Decisão para DSA para um curto-circuito aplicado no Trecho 8-9, considerando as contingência em operação N-1, 17#18 e 2#25.....	55
Figura 5.3 -	Árvore de Decisão para Controle Preventivo Trecho 8-9.....	64
Figura 5.4 -	Árvore de Decisão para Controle Preventivo Trecho 6-11.....	66
Figura 5.5 -	Árvore de Decisão para Controle Preventivo Trecho 9-39.....	67
Figura 5.6 -	Nomograma de segurança para um curto-circuito no trecho 8-9, para cenário com topologia completa do SEP em estudo.....	69
Figura 5.7 -	Nomograma de segurança para um curto-circuito aplicado no trecho 8-9, com contingência operação 16#21.....	70

Lista de Tabelas

Tabela 2.1 –	Características e potenciais impactos do estado atual da operação de SEP.....	12
Tabela 3.1 -	Exemplo de Matriz de Confusão.....	38
Tabela 5.1 –	Resultados de Classificação via Árvore de Decisão.....	56
Tabela 5.2 –	Resultados da Avaliação via RNA.....	57
Tabela 5.3 -	Desempenho da RNA para o curto-circuito aplicado no trecho 8-9, considerando a contingência em operação 26#29.....	59
Tabela 5.4 -	Pesos Sinápticos dos Neurônios da Camada de Saída da RNA para o curto-circuito aplicado no trecho 8-9, considerando a contingência em operação 26#29.....	60
Tabela 5.5 -	Valores dos Pesos que Conectam as Entradas ao Neurônio 1.....	61
Tabela 5.6 –	Desempenho da Rede Neural para o trecho 6-11.....	62
Tabela 5.7 –	Desempenho da Rede Neural para o trecho 9-39.....	63
Tabela 5.8 –	Resultados de Classificação via Árvore de Decisão 8-9.....	64
Tabela 5.9 –	Resultados de Classificação via Árvore de Decisão 6-11.....	65
Tabela 5.10 –	Resultados de Classificação via Árvore de Decisão 9-39.....	71
Tabela 5.11 –	Comparação entre os índices de desempenhos da árvore decisão e da rede neural para DSA.....	72
Tabela 5.12 -	Comparação entre índices de desempenhos da RNA e da DT para projeto de controle preventivo contra um curto-circuito aplicado no trecho 8-9, sem a linha 26#29.....	72
Tabela 5.13 -	Comparação entre os índices de desempenhos da RNA e da DT para controle preventivo contra um curto-circuito no trecho 6-11, com a linha 16#17 desligada.....	72
Tabela 5.14 -	Comparação entre os índices de desempenhos da RNA e da DT para controle preventivo contra um curto-circuito no trecho 9-39, com a linha 13#14 desligada.....	72

Capítulo 1 - Introdução

1.1 Considerações Iniciais

Atualmente os sistemas elétricos de potência (SEPs) tendem a operar próximos dos seus limites de segurança, devido ao crescente consumo de energia elétrica e a dificuldade de adição de novas instalações de transmissão e mesmo de geração, isso devido a questões financeiras e ambientais. Ao mesmo tempo, a desregulamentação do setor elétrico introduziu incertezas e complexidades crescentes nos sistemas elétricos, tornando-os mais vulneráveis a grandes perturbações que podem desencadear problemas de instabilidade transitória e, consequentes, desastrosos blecautes.

Um dos procedimentos efetivos para reduzir o risco de blecautes na operação de um sistema elétrico é a avaliação da segurança dinâmica (DSA, do inglês “*Dynamic Security Assessment*”) em tempo real, por meio da qual, os operadores podem tomar medidas preventivas, a fim de garantir a segurança do sistema (Genc et al., 2010). Contudo, o desenvolvimento de aplicativos para DSA, que possibilitem a operação em tempo real, com foco na estabilidade transitória, ainda é um grande desafio devido à alta não linearidade e a alta dimensionalidade do problema, tornando praticamente impossível verificar em um prazo de tempo razoável, todas as combinações de contingências em diversos cenários operativos. Tradicionalmente, ferramentas de DSA utilizam métodos convencionais de simulação no domínio de tempo, que levam em conta um conjunto de contingências, cujo processamento demanda um longo tempo, fato este que pode comprometer a tomada de decisão por intermédio de controles preventivos.

Atualmente a DSA, com foco na estabilidade transitória, já é uma realidade em diversos sistemas de grande porte por todo o mundo, como é o caso do Sistema Interligado Nacional (SIN) do Brasil, cuja avaliação é realizada pelo Operador Nacional do Sistema Elétrico Brasileiro (ONS). O ONS possui o software ORGANON. No entanto, esta ferramenta ainda não opera de maneira plena, devido a ao elevado esforço computacional exigido na avaliação de uma grande quantidade de contingências em diversos cenários operativos, o número de casos analisados ainda é limitado na operação em tempo real.

Com o avanço da inteligência artificial, o uso de técnicas de aprendizado de máquinas, como a árvore de decisão (DT, do inglês “*Decision Tree*”) e a rede neural artificial (RNA), são propostas de solução promissoras no desenvolvimento de

ferramentas para DSA em tempo real. Essas máquinas de aprendizado aproveitam os estudos *offline* para descobrir atributos críticos e inerentes ao SEP, relacionados a um objetivo (atributo alvo) (Diao, 2010). Assim, esse conhecimento obtido dessas máquinas de aprendizado pode ser diretamente utilizado para auxílio à tomada de ações preventivas, no sentido de garantir a melhoria da segurança, além de fornecer um modelo de predição rápido para aplicações em tempo real (Wehenkel, 1998).

1.2 – Objetivo Geral

O objetivo principal deste trabalho é comparar o uso de máquinas de aprendizado, mais especificamente, a DT do tipo C4.5 e a RNA do tipo (MLP, do inglês “*Multilayer Perceptron*”), aplicadas na DSA e no auxílio a tomada de ações de controle preventivo, com foco na estabilidade transitória. Esta comparação entre as técnicas utilizadas é verificada, a partir de seus percentuais de acerto, predição de classe, e sensibilidade; além da quantidade e do formato das entradas de dados, bem como a interpretabilidade e a flexibilidade do conhecimento extraído de cada modelo.

1.3 - Principais Contribuições

As principais contribuições deste trabalho são:

- Desenvolvimento de rotinas computacionais para geração automática da base dados rotulada, para fins de aplicação na DSA com foco na estabilidade transitória;
- Desenvolvimento de metodologia para DSA com foco na estabilidade transitória utilizando os algoritmos C4.5 e MLP;
- Desenvolvimento de metodologia para projeto de ações preventivas contra perturbações críticas que levam a instabilidade transitória.

1.4 - Revisão Bibliográfica

Várias aplicações envolvendo o desenvolvimento de metodologias para avaliar a segurança dinâmica de sistemas de potência utilizando árvores de decisão e redes neurais foram investigadas ao longo dos anos, a partir da evolução desses algoritmos.

Wehenkel, em 1994 treinou árvores de decisão para predição do tempo crítico de falta no sentido de avaliar a estabilidade transitória. Também em 1994, Rovnyak criou

um método para predição da estabilidade transitória em tempo real, que por meio da formação de uma base de dados que considerava os ângulos do rotor pós-contingência como atributos. Em 2000, Rovnyak usou DTs para identificar perdas de sincronismo entre máquinas, com base na resistência aparente medida próxima a interligação da central geradora.

Outros esforços mais recentes envolvendo o uso de árvores de decisão têm utilizado medidas provindas de unidades de medição fasorial ou (PMU, do inglês “*Phasor Measurement Units*”), para avaliar a segurança dinâmica no contexto da estabilidade transitória, Sun 2007 e Diao em 2010.

Em 2010, Genc utiliza DT (algoritmo de CART) para gerar restrições de potência nas máquinas que são inseridas em fluxo de potência ótimo, a fim de operar o sistema de forma segura, onde gerou também nomogramas com limites de segurança baseado em suas potências.

Em 2011 e 2012, a preocupação com a base de dados de treinamento de ferramentas baseadas em máquinas de aprendizado, foi evidenciada nos trabalhos de Krishnan.

O uso de redes neurais também está envolvido em diversos esforços para avaliação de segurança dinâmica. Em 1998 Srinivasan usa RNA do tipo MLP para encontrar o tempo crítico de eliminação de falta, onde compara os valores encontrados pela RNA com o método direto de Lyapunov. Outros inúmeros trabalhos utilizando outros tipos de arquiteturas de RNA podem ser citados.

Jensen em 2001 utiliza o discriminante linear de Fisher com o intuito de encontrar os melhores atributos para treinar redes neurais.

Em 2011, Gutierrez-Martinez, modela uma barra de limite de intercambio por uma rede MLP, em seguida extrai regras para um fluxo de potencia ótimo. Karami utiliza redes MLP para encontrar margens de estabilidade transitória de SEP.

Lin, em 2012 propõe uma metodologia de compensação serie para controle preventivo baseado em regras do tipo se-então extraídas de uma MLP.

Em 2013, Lin faz uma comparação entre os algoritmos CART e MLP pra o uso em controle preventivo, onde são citadas as vantagens e desvantagens dos algoritmos para esse fim.

1.5 - Estrutura da Dissertação

Os conceitos sobre segurança dinâmica, assim como conceitos de controle preventivo são apresentados no capítulo 2. Este capítulo também apresenta a forma tradicional na qual é avaliada a segurança do SEP, bem como os desafios que os modernos SEPs enfrentam para operar de forma segura.

O capítulo 3 aborda os conceitos sobre sistemas inteligentes, com ênfase nas técnicas de aprendizado de máquina baseadas em DTs do tipo C4.5 e em RNAs do tipo MLP.

As metodologias propostas para DSA em tempo real e para auxílio ao controle preventivo, com foco na estabilidade transitória, utilizando DTs e RNAs, respectivamente, são apresentadas no capítulo 4.

O capítulo 5 apresenta os estudos de casos comparativos das técnicas DTs e RNAs, ambas aplicadas a DSA e ao auxílio de ações preventivas, com foco na estabilidade transitória, mostrando uma discussão sobre as vantagens e desvantagens de cada técnica, além de apresentar nomogramas de segurança oriundos das regras extraídas da árvore de decisão.

Por fim, o capítulo 6 apresenta as conclusões gerais provindas dos estudos de casos e indica sugestões para trabalhos futuros.

Capítulo 2 – Segurança de Sistemas de Potência

2.1 Introdução

A segurança de um sistema elétrico de potência (SEP) refere-se ao grau de risco contido em sua habilidade para sobreviver a iminentes perturbações (contingências) sem a interrupção do serviço. Assim, reflete a robustez do SEP em relação a perturbações, sendo que o nível de sua robustez depende das condições operativas (patamares de carga, despacho e intercâmbio de geração, além da topologia da rede), assim como da probabilidade de ocorrência das perturbações (KUNDUR, 2004).

A avaliação da segurança geralmente tem como base critérios pré-definidos no que se refere às análises e métodos utilizados na determinação do estado de segurança (seguro ou inseguro), assim como quais contingências são avaliadas. Esta avaliação pode ser classificada como estática ou dinâmica, cuja classificação depende da modelagem e das técnicas utilizadas para se enquadrar em um dos tipos. A avaliação estática analisa as condições do sistema em regime permanente no que diz respeito aos níveis de tensão nas barras e aos limites térmicos dos circuitos de transmissão. Por outro lado, a segurança dinâmica pode englobar outros aspectos relativos à estabilidade do sistema, tais como estabilidade transitória, estabilidade a pequenas perturbações e estabilidade de tensão.

Em situações de casos identificados como instáveis em fase de estudos e planejamento, medidas de controle devem ser adotadas, essas medidas são conhecidas como controle preventivo, ou mesmo com o nome de controle de segurança. Essas ações de controle têm como objetivo levar o ponto de operação do SEP para uma condição segura.

Neste capítulo são estabelecidos alguns conceitos básicos relacionados à segurança estática e dinâmica de sistemas de potência, dando uma ênfase maior ao fenômeno da estabilidade transitória, assim como para algumas práticas comuns de controle preventivo.

2.2 – Segurança Estática de Sistemas de Elétricos de Potência

O SEP é seguro estaticamente se retorna a um ponto de operação seguro, próximo do anterior, caso sofra uma contingência. São comuns os estudos sobre contingências serem feitos com retiradas de linhas e/ou transformadores. A segurança estática de um sistema elétrico está ligada a um dado ponto de equilíbrio (ponto de

operação) da rede, para o qual são avaliados os níveis de tensão e carregamento nas linhas, assim com a retirada de algum componente da rede.

Normalmente, a ferramenta de análise da segurança estática se resume a algoritmos de fluxo de potência, também conhecido como fluxo de carga. Para um dado ponto de operação, se os níveis de tensão das barras e os fluxos nos circuitos de transmissão se encontram dentro dos respectivos limites, o sistema é dito estaticamente seguro.

O efeito das contingências deve ser levado em consideração de forma que os critérios de segurança de regime permanente sejam respeitados, mesmo quando houver perda de algum elemento do sistema. Um dos métodos mais utilizados para a determinação dos limites de carregamento em regime permanente é a determinação das curvas PxV mostrada na Figura 2.1 que estão relacionados com estabilidade de tensão.

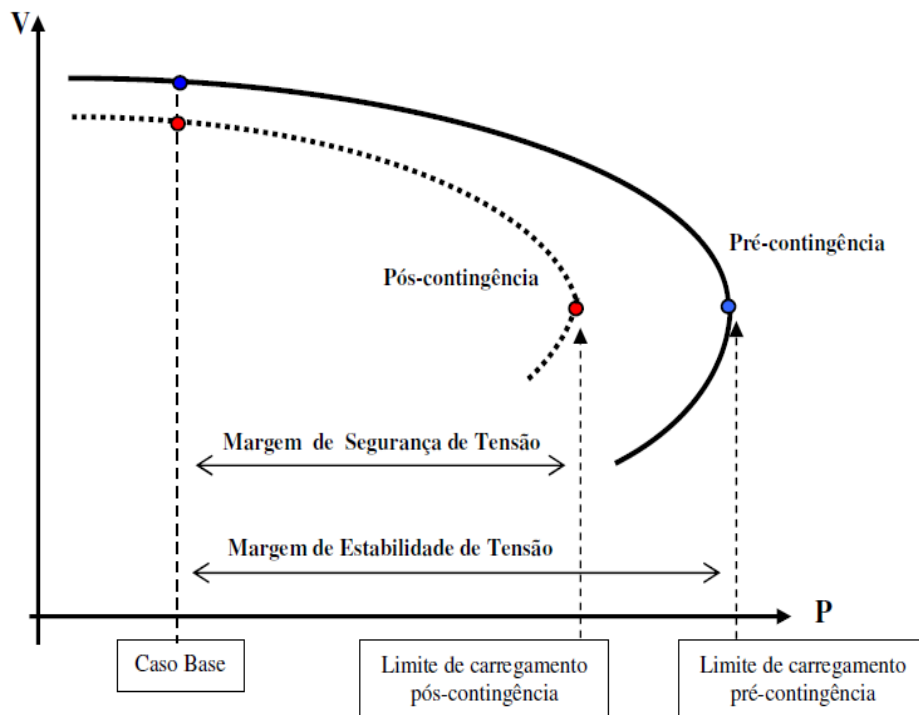


Figura 2.1 - Curvas PxV (Santa Rosa,2007)

A margem de carregamento pode ser calculada para se identificar a habilidade do sistema de se manter estável diante de uma previsão de elevação de carga. Entretanto, o cálculo da margem de carregamento do sistema após uma contingência, também chamada de margem de segurança de tensão como mostrado na (Figura 2.1) é o procedimento mais utilizado para identificar o quão seguro é o sistema elétrico.

2.3 – Segurança Dinâmica de Sistemas Elétricos de Potência

A segurança do sistema envolve um vasto conjunto de subproblemas, que podem ser enquadrados em dois níveis: a segurança do funcionamento em regime permanente, que está associada ao ponto de operação e ao comportamento da rede face à saída de serviço de alguns componentes e a segurança dinâmica do sistema, associada com problemas de estabilidade. Apesar de todas as precauções tomadas na etapa de planejamento e concepção de um SEP, a ocorrência de algumas contingências e posterior sequência de acontecimentos pode conduzir a situações de perda parcial ou total do sistema.

Entre os múltiplos eventos que podem provocar o colapso do sistema, destacam-se os seguintes: o aumento súbito de carregamento numa ou em várias linhas de interligação de uma rede elétrica à redes vizinhas; produção insuficiente de potência reativa, podendo provocar a degradação do nível de tensão; produção deficitária de potência ativa originada pela saída de serviço de uma unidade geradora do sistema por uma queda do valor da frequência; defeitos severos em algumas linhas ou barramentos da rede; e situações de instabilidade que conduzam à perda do sincronismo. Todos estes fenômenos, direta ou indiretamente, conduzem a problemas de estabilidade na rede elétrica (Ferreira, 2005).

A elevada dimensionalidade e a complexidade dos sistemas elétricos torna a simplificação e análise dos problemas específicos com grau de detalhe adequado uma tarefa essencial.

Os estudos de estabilidade transitória analisam o comportamento do sistema durante os primeiros segundos após a ocorrência de grandes perturbações. O sistema é dito transitoriamente estável caso mantenha o sincronismo entre as máquinas, retornando ao ponto de operação seguro estaticamente, muitas vezes próximo ao ponto de operação pré-falta.

O problema da análise da estabilidade de um SEP, atualmente, é dividido em três grandes classes (Kundur, 1994), figura 2.2. Esta classificação tem por base os seguintes fatores: o fenômeno que caracteriza o tipo de instabilidade e as causas físicas que conduzem à sua ocorrência.

A estabilidade angular é conceituada como a capacidade que as máquinas síncronas interconectadas a um SEP possuem de manter o sincronismo, quando o mesmo é submetido a uma perturbação.

A estabilidade a pequenas perturbações é a capacidade do sistema de potência de manter o sincronismo quando submetido a uma pequena perturbação. Atualmente, esse fenômeno é amplamente tratado como um problema de insuficiência de amortecimento de oscilações, nos quais os de maior interesse são:

- Modos Locais: Modos associados com a oscilação de unidades geradores de uma usina contra o resto do sistema de potência.
- Modos Inter-áreas: Modos associados com a oscilação de um grupo de geradores em uma área do sistema contra outro grupo de geradores em outra área.

A estabilidade transitória é a capacidade do SEP de manter o sincronismo quando submetido a uma grande perturbação, como um curto-circuito.

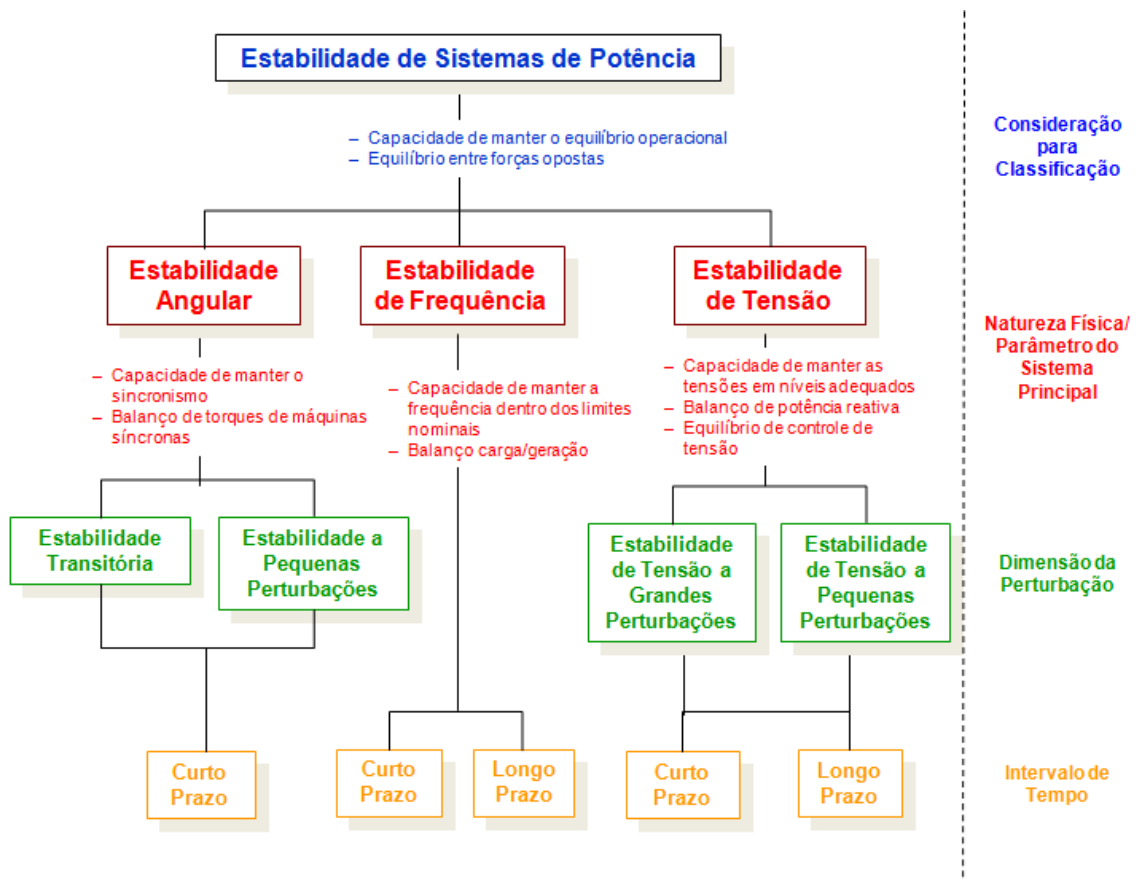


Figura 2.2 - Classificação de Estabilidade de Sistemas de Potência (Kundur, 1994)

A estabilidade de tensão é a capacidade de manter as tensões em níveis adequados em todas as barras do SEP após o mesmo ter sido submetido a uma perturbação em uma dada condição operacional inicial. A instabilidade de tensão ocorre quando uma perturbação, como um aumento de carga ou uma mudança da condição do

sistema ocasiona uma queda ou elevação progressiva e descontrolada de tensão em algumas barras. A instabilidade de tensão está ligada a incapacidade do SEP em manter um equilíbrio adequado de balanço de potência reativa e ações de controle de tensão. A instabilidade de tensão pode ou não está associada a um problema de estabilidade angular, quando os problemas estão associados, a queda de tensão ocorre de maneira rápida, na escala de segundos. Caso a queda de tensão ocorra mais lentamente, em uma escala de tempo de minutos, a estabilidade angular não é um problema (Kundur et al, 2004).

A estabilidade de frequência é a capacidade que o SEP possui de manter a frequência em níveis adequados, ou seja, dentro dos limites nominais após uma perturbação que resulta em um significativo desequilíbrio entre a geração e a carga. Esse fenômeno pode ocorrer na forma de oscilações sustentadas de frequência, levando ao desligamento de geradores e cargas, sendo determinado pela resposta global do sistema, tal como evidenciado pela sua frequência média, ao invés dos movimentos relativos dos rotores dos geradores (Kundur et al., 2004).

2.3.1 – Estabilidade transitória

Também denominada estabilidade de ângulo a grandes perturbações, a estabilidade transitória depende da capacidade que o SEP possui para manter e/ou restaurar o equilíbrio entre o torque eletromagnético e o torque mecânico de cada máquina síncrona interligada ao sistema.

A avaliação da estabilidade transitória, devido a sua importância, representa um grande esforço nos estudos sobre SEP, a qual é tradicionalmente realizada através de simulações no domínio do tempo. A simulação dinâmica no domínio do tempo, também conhecida como “simulação completa”, oferece a maior precisão com relação ao comportamento dinâmico do sistema.

A resposta do sistema a um curto-circuito envolve grandes excursões dos ângulos dos rotores dos geradores. Se a separação entre as máquinas interconectadas ao sistema cresce aperiodicamente, o sistema perde o sincronismo. A perda do sincronismo devido a instabilidade transitória, geralmente se evidencia na primeira oscilação, nos primeiros segundos após a perturbação. A figura 2.2 apresenta a variação do ângulo do rotor de um gerador síncrono no decorrer do tempo após uma perturbação, considerando três casos.

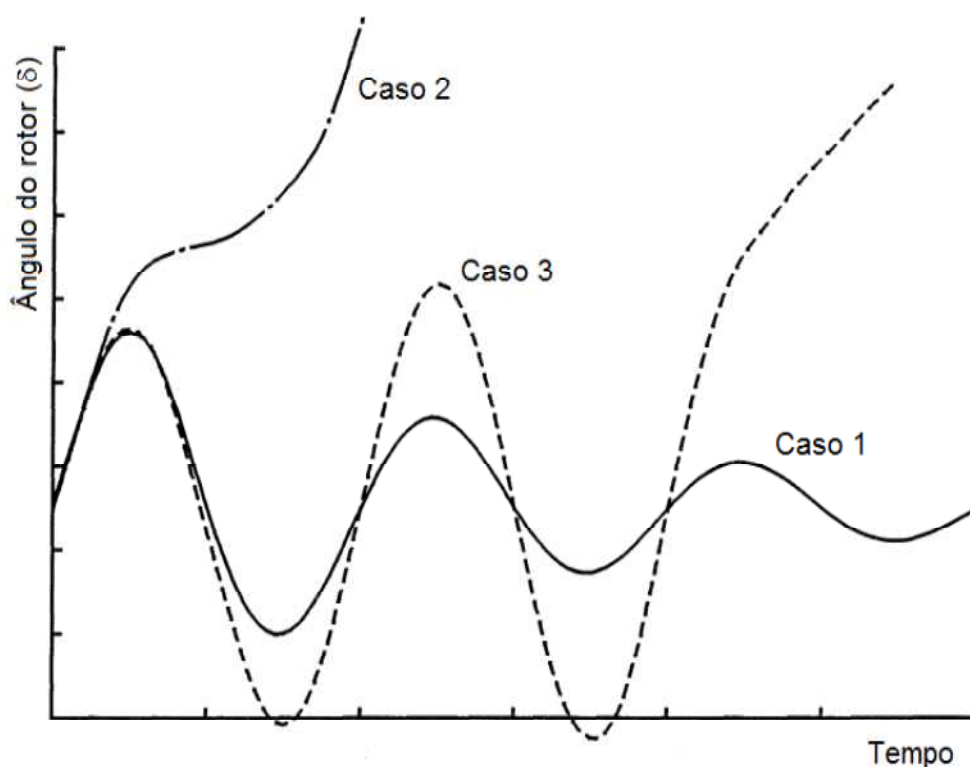


Figura 2.3 - Resposta no tempo do ângulo do rotor frente a um distúrbio transitório

No Caso 1 da figura 2.3 o sistema é dito estável (seguro), pois mesmo que o ângulo do rotor oscile após a perturbação, esse ângulo se estabiliza em um ponto de operação próximo do ponto de operação pré-falta, em regime permanente.

No Caso 2 o ângulo do rotor cresce aperiódicamente logo na primeira oscilação, caracterizando assim um caso de instabilidade transitória.

O Caso 3 apresenta uma situação de instabilidade angular devido à falta de torque de amortecimento. Neste caso, o ângulo do rotor não cresce aperiódicamente na primeira oscilação. Diferentemente do caso 1, essa oscilação começa a crescer indefinidamente levando o sistema a instabilidade.

Para a “simulação completa” o processo de cálculo consiste na solução de um conjunto de equações diferenciais e algébricas não-lineares que descreve o comportamento dinâmico dos elementos do sistema (equação 2.1).

$$\begin{cases} \dot{x} = f(x, r, u) \\ 0 = g(x, r, u) \end{cases} \quad (2.1)$$

Onde:

x é o vetor de variáveis de estado (velocidade da máquina, ângulo do rotor, etc.);

r é o vetor de variáveis algébricas (módulo, ângulos das tensões terminais, etc.);

u é o vetor de variáveis de entrada (controles);

f é o vetor de funções não-lineares das equações diferenciais dos modelos dos dispositivos dinâmicos do sistema (geradores, motores de indução, os diversos controles, etc.);

g é o vetor de funções não-lineares que descreve as equações algébricas da rede e de alguns modelos de dispositivos presentes no sistema.

A solução do sistema de equações descrito em (2.1) é realizada através de técnicas de integração numérica no domínio do tempo. Normalmente, utiliza-se entre 10 ou 20 segundos de simulação após o distúrbio, sendo que esse tempo é suficiente para indicar problemas de estabilidade transitória. Por esse motivo, o termo “estabilidade de curto-prazo” também é utilizado.

Uma vez detectada a possibilidade de ocorrência de instabilidade, medidas devem ser adotadas para garantir um ponto de operação seguro. Tais medidas podem incluir o redespacho de potência ativa ou até mudanças dos esquemas de proteção do sistema.

Em aplicações onde o tempo de simulação é crítico, como por exemplo, a operação em tempo-real, a solução numérica do sistema (Equação 2.1) pode ser inviável, devido ao tempo consumido pelo processo iterativo. Este fato tem fortalecido linhas de pesquisa que investigam os chamados métodos diretos baseados em funções de energia, sendo capazes de avaliar a estabilidade transitória sem a necessidade da integração numérica. Além disso, o uso de funções de energia tem como grande atrativo o fato de possibilitar a obtenção do grau de estabilidade do sistema com base na margem de energia.

Embora os métodos diretos apresentem grande vantagem na redução no tempo computacional, sua utilização requer limitações na modelagem dos elementos dinâmicos do sistema. Os chamados métodos híbridos, que combinam as vantagens da simulação no domínio do tempo e dos métodos diretos, também vêm sendo bastante explorados. Uma alternativa para a melhora o desempenho computacional, mantendo-se a modelagem detalhada dos elementos dinâmicos, é o uso de métodos de integração com passo e ordem variáveis. Outra maneira de acelerar um estudo de estabilidade transitória em tempo real é a execução concorrente de vários casos em sistemas multiprocessados (*clusters*).

2.4 - Desafios para Operação Segura

O aumento da demanda por energia elétrica e a necessidade das concessionárias manterem os padrões de qualidade de serviço, aliado há desregulamentação do setor elétrico certamente são fatores desfavoráveis à operação segura do sistema, e acabam acarretando em enormes desafios para a operação dos sistemas elétricos. Esse aumento progressivo do consumo acarreta no carregamento cada vez maior das linhas de transmissão existentes, e as unidades geradoras operam cada vez mais próximas dos seus limites máximos, tornando o sistema susceptível há uma condição de colapso frente a distúrbios. Algumas soluções para esse problema e seus respectivos impactos são mostrados na tabela 2.1 (Morinson et al, 2004):

Tabela 2.1 – Características e potenciais impactos do estado atual da operação de SEP

Características	Potenciais Impactos
➤ Envelhecimento das Linhas de Transmissão	<ul style="list-style-type: none"> • Aumento da probabilidade de falhas ou mau funcionamento de equipamentos, levando a perturbação no sistema.
➤ Dificuldade para instalações de novas Linhas de Transmissão seja por seu alto custo ou questões ambientais.	<ul style="list-style-type: none"> • Sobrecarga nas linhas das atuais instalações de transmissão que levam a atuação da proteção ou contribuindo para o fenômeno como o colapso de tensão. • Gargalos nos principais corredores de transmissão, levando a congestionamentos.
➤ Cortes na manutenção do sistema	<ul style="list-style-type: none"> • Falhas em componentes e distúrbios como flashovers.
➤ Maior dependência de controles e proteções especiais.	<ul style="list-style-type: none"> • Aumento da probabilidade de operação incorreta de proteções. • Aumento da imprevisibilidade de eventos em cascata.
➤ Grande número de pequenos geradores distribuídos.	<ul style="list-style-type: none"> • Maior dificuldade na concepção do sistema adequado devido à incerteza de planejamento. • Incerteza no despacho
➤ Intercambio de Potência	<ul style="list-style-type: none"> • Fluxos de potência menos previsíveis • Maiores sobrecargas nas principais linhas ou mesmo fraco comportamento dinâmico. • Problemas como estabilidade de tensão ou mesmo estabilidade a pequenas perturbações.
➤ Aumento da dependência de sistemas computadorizados e de comunicação.	<ul style="list-style-type: none"> • Falhas provindas de Software / Hardware podem deixar grande parte

	do sistema não observável aos operadores, levando à inadequada, ou falta de ações de controle durante distúrbios.
➤ Planejamento do sistema integrado limitado.	<ul style="list-style-type: none"> • Utilização de recursos inadequados ou mesmo a falta de recursos para geração de energia.
➤ Tendência de interconexão	<ul style="list-style-type: none"> • Exposição a distúrbios em cascata provocada por eventos em sistemas vizinhos.
➤ Novas tecnologias, como sistemas avançados de controle, geração eólica, geração fotovoltaica, etc.	<ul style="list-style-type: none"> • Falta de experiência do operador com tecnologias que podem ter características dinâmicas • Comportamentos imprevisíveis durante os distúrbios.

2.5 - Avaliação da Segurança Dinâmica em Tempo Real

A análise de segurança é classificada em tempo real ou *on-line* quando a determinação do estado e nível de segurança faz parte das tarefas rotineiras de monitoramento. Na análise *on-line*, as reais condições operativas do sistema podem ser levadas em conta, e os limites operativos e ações de controle podem ser atualizados periodicamente (CASTRO, 1995).

Nos estudos ditos *off-line*, opta-se por avaliar somente os piores casos, tendo em vista o tamanho da rede elétrica e o grande número de contingências a analisar. Contudo, esta atitude conservadora leva ao estabelecimento de limites operativos imprecisos e conservativos, válidos somente para situações genéricas de carga leve, média ou pesada, por exemplo. Porém, muitas vezes o sistema opera em uma condição diferente da antes avaliada. Portanto, as instruções operativas disponíveis para dados momentos podem não serem as mais adequadas para a execução pelo operador. Com a análise *on-line* a operação do sistema se dá de modo muito realista e confiável, pois se baseia nas informações mais recentes possíveis.

As ações básicas da análise de segurança *on-line* são: monitoramento, avaliação e melhoria da segurança, como ilustrado na figura 2.4:

a) Monitoramento da segurança: Usando medições do sistema em tempo real, é identificado se o sistema está em estado normal ou não. Se ele estiver em estado de emergência precisará de um controle de emergência. Se ocorrer perda de carga será necessário um controle restaurativo.

b) Avaliação da Segurança: Se o sistema está em estado normal, é determinado se o sistema está seguro ou inseguro com respeito a um conjunto de contingências, bem como são estabelecidos índices ou margens de segurança;

c) Melhoria da Segurança: Se inseguro, são determinadas ações corretivas que devem ser tomadas para que o sistema se torne seguro. Se seguro, com pouca margem, são determinadas as ações preventivas para evitar que o sistema se torne inseguro.

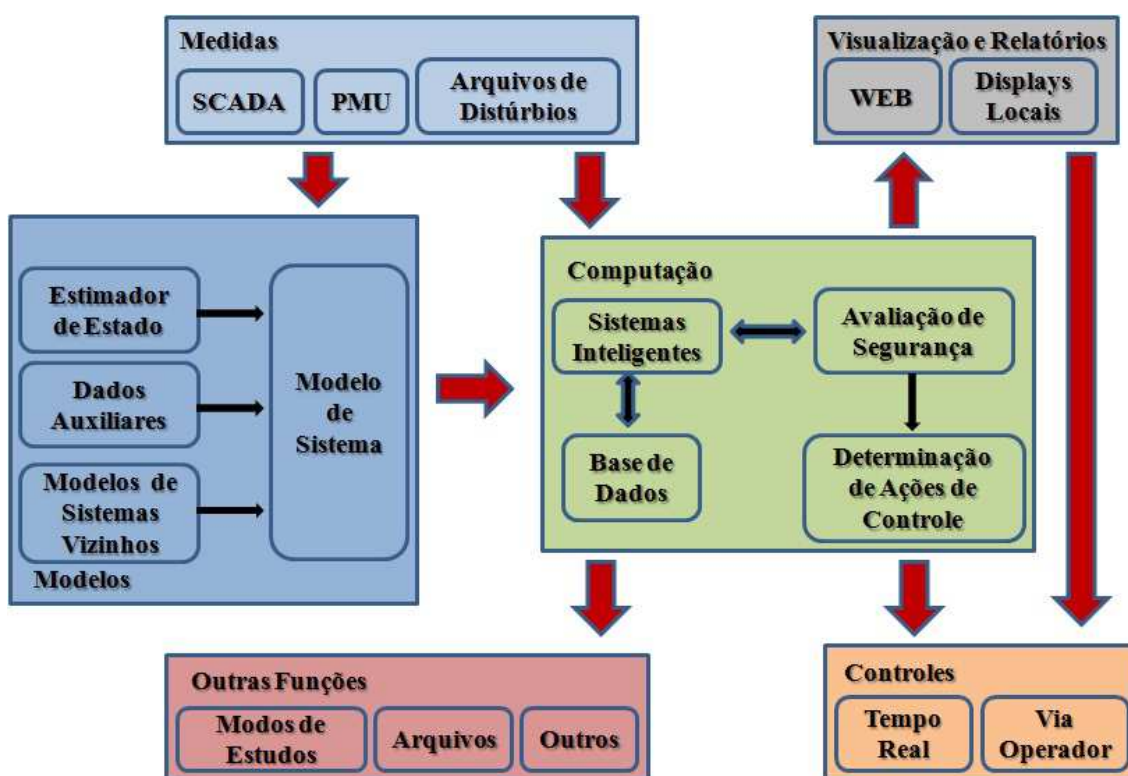


Figura 2.4 – Avaliação em tempo real de SEP (Morinson et al., 2004)

A busca por uma operação segura e econômica de um SEP é sempre prioritária. No decorrer dos anos os centros de controle das empresas de energia elétrica vêm evoluindo e incorporando os recursos computacionais que surgiram nesse ínterim, o que têm aberto o caminho para a implantação de ferramentas *on-line*, com as funções de monitoramento, análise e controle. Atualmente, vários centros de controle realizam análise de segurança estática (análise de contingências) em modo *on-line*. Já a implementação da avaliação segurança dinâmica *on-line* está em um estágio bem menos avançado devido a três motivos principais:

- A complexidade do problema dinâmico;
- Capacidade computacional insuficiente dos centros de controle para realizar a análise a tempo.

- Falta de métodos alternativos para análise de segurança que sejam eficientes e confiáveis;

Na análise *on-line*, o objetivo é determinar se um ponto de operação atual ou futuro será seguro ou não, partindo-se de medidas em tempo-real do sistema, detecção de configuração e estimação de estado que fornece o caso base presente. Em estudos *off-line* os casos base são previamente selecionados para representar os cenários de pior caso. Análise de segurança só faz sentido se o sistema estiver no estado normal, ou seja, toda a carga atendida e nenhuma restrição operativa violada.

A análise *on-line* de segurança dinâmica pode ser decomposta em vários subproblemas que serão descritos a seguir e interagem entre si na forma mostrada na figura 2.5.

- **Previsão de Carga:** A análise da segurança dinâmica *on-line* é uma análise prospectiva, isto é, visa determinar se um ponto de operação futuro será estável (sendo, portanto um estado operativo seguro), ou instável (levando o estado do sistema a ser classificado como inseguro). Sua finalidade é dar aos operadores informações atualizadas sobre limites operativos de transferência e ações preventivas de controle que ajude na tomada de decisões de forma antecipada. Deste modo, realizar previsão de carga e geração torna-se parte essencial na análise.
- **Preparação de Cenários:** Este módulo recebe informações de vários tipos e prepara convenientemente as bases de dados que serão utilizadas durante o cálculo da estabilidade transitória. Fornece o caso base futuro. Como funciona como uma espécie de gerente da análise de segurança pode ser adotada de certa dose de inteligência artificial.
- **Cálculo da Estabilidade Transitória:** Este subproblema corresponde ao que normalmente faz um programa de cálculo da estabilidade transitória. Embora qualquer método possa ser usado (direto, híbrido ou simulação) é mais confiável utilizar, como será feito neste trabalho para fins de geração da base de dados, a simulação no domínio do tempo por integração numérica. Desta maneira, evitam-se restrições na modelagem de geradores, seus controles e outros componentes. Por outro lado, a simulação passo-a-passo pode ser muito demorada, levando este módulo a ser excessivamente oneroso em termos computacionais.

- **Avaliação da Segurança:** os resultados das simulações são analisados e o sistema é classificado como seguro ou inseguro. O sistema (ponto de operação) é considerado inseguro se pelo menos uma contingência poder levar um ou mais geradores à instabilidade. Dentro deste módulo também se inclui a obtenção de margens ou índices de estabilidade ou instabilidade, dependendo da situação. A margem de estabilidade quantifica a distância que um sistema estável está da instabilidade, enquanto que a margem de instabilidade quantifica a distância que um sistema instável está da estabilidade.
- **Atualização de Limites Operativos:** Em geral, as redes elétricas têm limites operativos dinâmicos expressos através do máximo carregamento de um conjunto de linhas de transmissão ou máxima geração de um grupo de usinas (fluxo de interface). Se o sistema estiver dinamicamente seguro pode-se estudar o relaxamento dessas restrições e verificar até onde os limites podem ser relaxados. Frequentemente, isto é realizado através de sucessivos redespachos de geração incrementais em pontos localizados até que um estado inseguro seja alcançado.
- **Redespacho Corretivo de Geração:** Se for detectado que o sistema está dinamicamente inseguro pode-se estudar medidas preventivas ou corretivas para levá-lo ao estado seguro. Uma ação corretiva (atua após a ocorrência de uma contingência) que tem se mostrado efetiva consiste no corte de potência dos geradores que tendem à instabilidade (perda de sincronismo). Portanto, deve-se obter a mínima quantidade de geração a ser reduzida capaz de levar o sistema a um estado seguro. Isto corresponde a realizar redespachos incrementais de geração até se obter um esquema de corte de geração, que deverá ser implantado nos mecanismos automáticos de controle de geração.

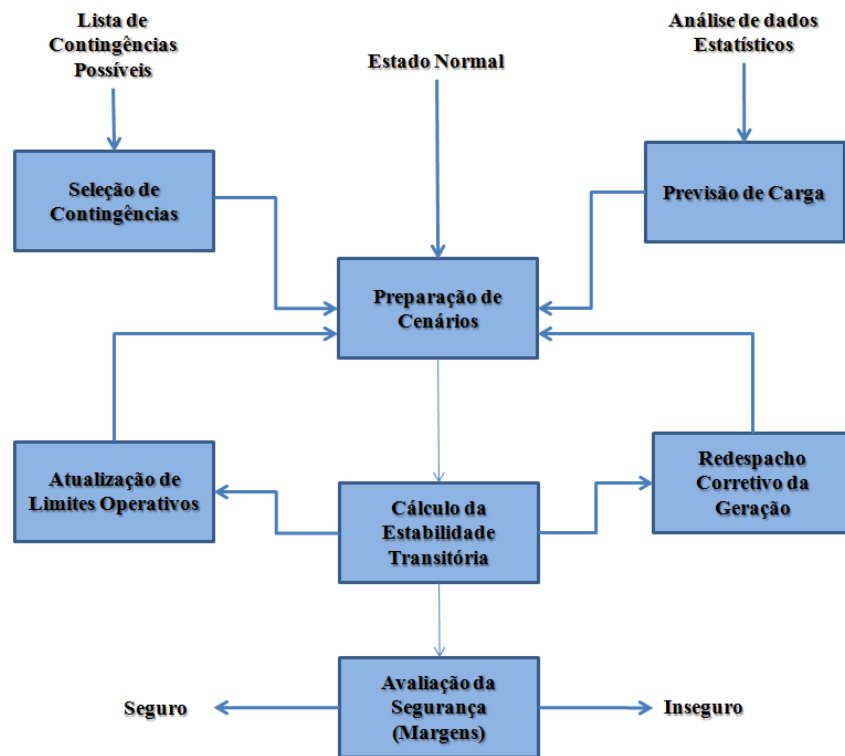


Figura 2.5 - Subproblemas da análise on-line de segurança dinâmica (CASTRO, 1995).

2.6 - Controle preventivo:

O comportamento não-linear dos SEP exige permanente análise de seus desempenhos frente à ocorrência de defeitos (contingências). Nos casos instáveis e/ou que ocorrem violação do limite de capacidade dos equipamentos, faz-se necessária a adoção de medidas que possibilitem conduzir o sistema ao estado seguro, sendo esta atividade definida como controle de segurança ou controle preventivo.

O objetivo do controle preventivo é o de aumentar a robustez do sistema no sentido de suportar a ocorrência de contingências severas, com ações que levem o ponto de operação antes em situação de insegurança, para dentro da região de segurança, a figura 2.5 ilustra uma região de segurança genérica.

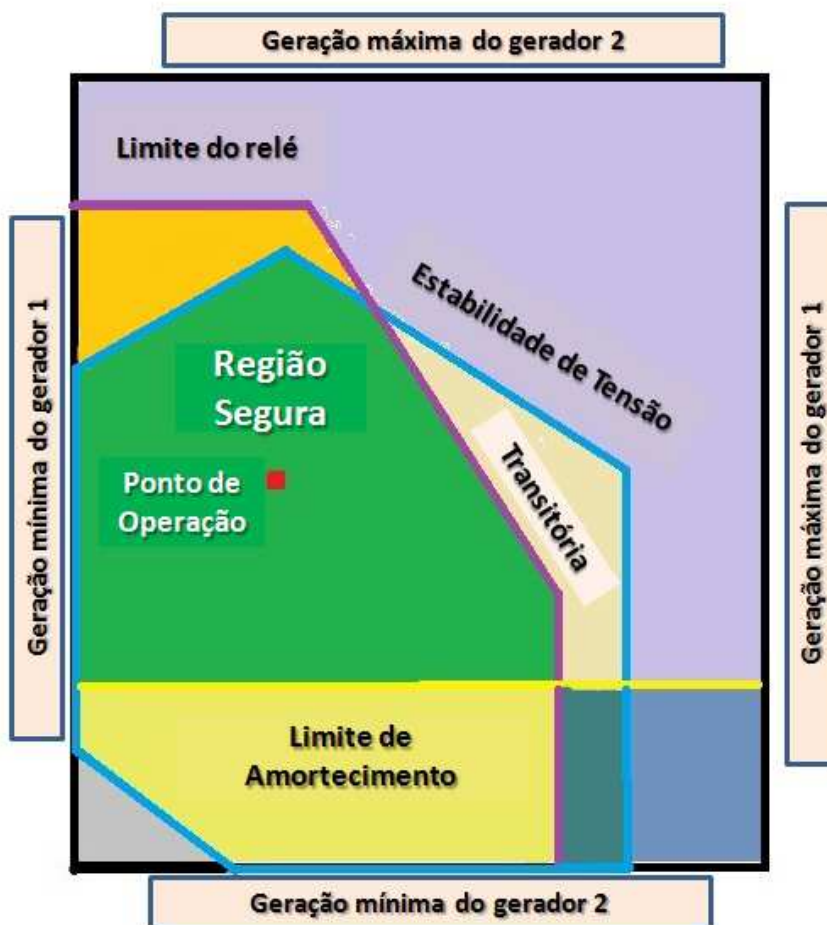


Figura 2.6 – Nomograma de segurança (Morinson et al., 2004)

A figura 2.6 ilustra os limites de segurança na forma de nomograma, mostrando uma zona segura (região em verde), na qual o sistema pode operar sem qualquer problema, ou seja, o operador pode manobrar o sistema sem que viole os critérios de funcionamento em segurança. Este é um nomograma bidimensional que representa o caso de dois grupos de geração. Os limites de um nomograma são determinados a partir de avaliação de limites térmicos, estabilidade transitória, estabilidade de tensão, estabilidade a pequenos sinais, limites de reativo e limites de margens dos relés.

Nos centros de operação um conjunto de possíveis contingências é permanentemente avaliado por meio de simulação, seja ela pelos modelos completos ou por métodos de função energia, quando alguma das contingências torna-se crítica. Quando é identificada alguma contingência crítica, ações de controle devem ser tomadas, porém o momento de atuação deste controle depende da decisão do operador em tempo real (tomar a ação agora ou postergá-la?).

As ações comuns para controle preventivo são:

- Redespacho de Geração;
- Chaveamento de dispositivos shunt;
- Mudança de TAP de transformadores;
- Mudanças de *offset* de reguladores;
- Corte preventivo de carga;

Contudo, devido à complexidade dos sistemas estudados, o número de contingências críticas selecionadas para estudos muitas vezes podem não ser suficientes para uma avaliação completa. Por isso, o uso de ferramentas inteligentes vem crescendo nos últimos anos. Essas ferramentas tem o intuito de auxiliar a equipe de operação a tomar decisões de forma rápida e confiável.

Capítulo 3 – Sistemas Inteligentes

3.1 – Introdução

Nos últimos anos, a Inteligência Artificial (IA) tornou-se um dos protagonistas no estado da arte de diversas áreas do conhecimento, devido à crescente necessidade do uso de ferramentas de apoio para tomadas de decisão por parte dos especialistas aliado aos avanços na ciência da computação. Contudo, os estudos em IA não é algo tão recente. As pesquisas foram iniciadas logo após o fim da Segunda Guerra Mundial e o seu nome foi definido no final da década de 50 (Russell & Norving, 2004).

A inteligência artificial é um ramo da Ciência da Computação cujo interesse é fazer com que os computadores “pensem” ou se comportem de forma inteligente. Os principais aspectos encontrados em sistemas inteligentes são:

- Aprender com a experiência.
- Aplicar o conhecimento adquirido da experiência.
- Tratar situações complexas.
- Resolver problemas quando faltam informações importantes.
- Determinar o que é mais importante.
- Capacidade para raciocinar e pensar.
- Reagir rápida e corretamente às novas situações.
- Utilizar heurísticas para resolução de problemas.

Uma forma para compreender a IA se dá por meio de uma tentativa de aproximar a capacidade de aprendizado do computador à do ser humano, mesmo que com enormes diferenças entre elas. Os algoritmos utilizados em IA são desenvolvidos baseados em modelos matemáticos que tentam emular alguma função específica, do cérebro, por exemplo, um comportamento racional para determinada situação ou mesmo um comportamento natural observado na biologia. A figura 3.1 apresenta algumas áreas do conhecimento que estão relacionadas a IA.

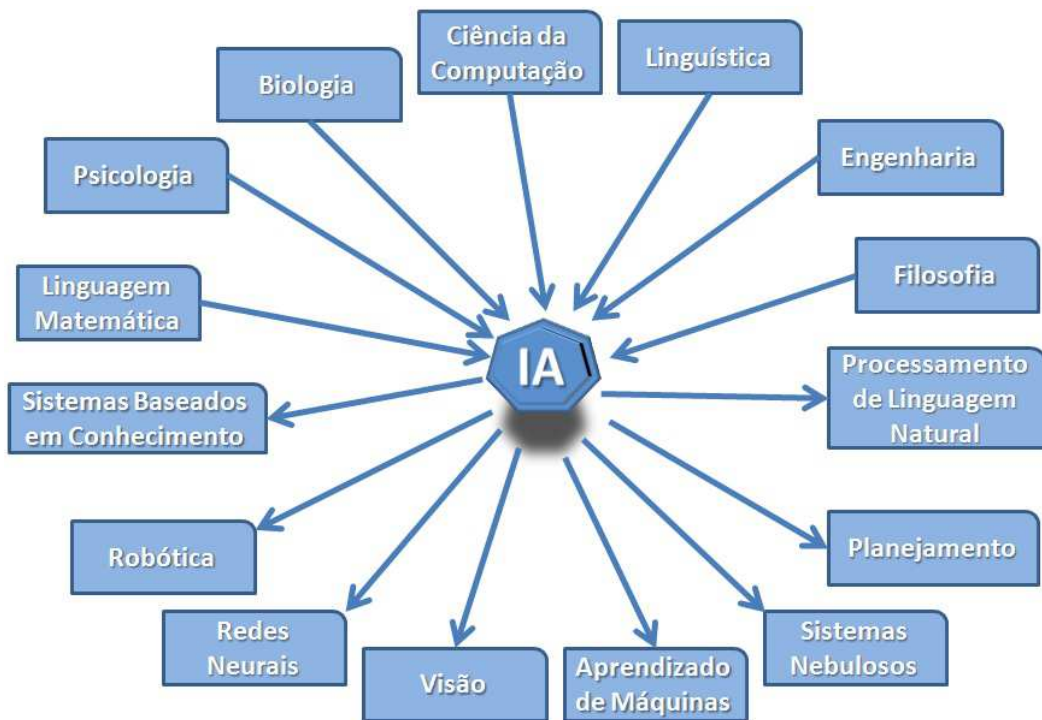


Figura 3.1 - Áreas Relacionadas com a Inteligência Artificial. (MONARD & BARANAUKAS, 2003)

O foco principal deste capítulo é apresentar os sistemas inteligentes baseados em formas de Aprendizado de Máquina (AM). O AM é uma peça fundamental para a evolução da robótica e automações de modo geral, assim como no apoio a tomada de decisão.

3.2 – Aprendizado de Máquinas

O aprendizado de máquinas lida com os problemas de aprendizado computacional, com o intuito de adquirir conhecimento de forma automática. Esses sistemas de aprendizado têm como função analisar e generalizar as informações, para extrair conhecimento dos mesmos (MORNARD & BARANAUSKAS, 2003). Assim, um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas através da solução bem sucedida de problemas anteriores.

Utilizando o princípio da indução, que é a inferência lógica, para obter a partir de um conjunto de exemplos conclusões genéricas. O aprendizado indutivo pode ser dividido em supervisionado e não-supervisionado. O aprendizado supervisionado é usado para resolver problemas preditivos, haja vista que essa tarefa analisa os exemplos em função de classes predefinidas com o intuito de compreender o relacionamento entre os atributos e a classe a fim de prever em que classe novos exemplos serão apontados.

Enquanto o aprendizado não supervisionado agrupa exemplos semelhantes (clusters), resolvendo assim problemas descritivos, podendo também ser usado para redução de dimensionalidade de dados, assim como encontrar causas ou fontes ocultas dos exemplos ou mesmo modelar a densidade dos exemplos. A figura 3.2 exemplifica essa divisão entre algoritmos supervisionados e não supervisionados.

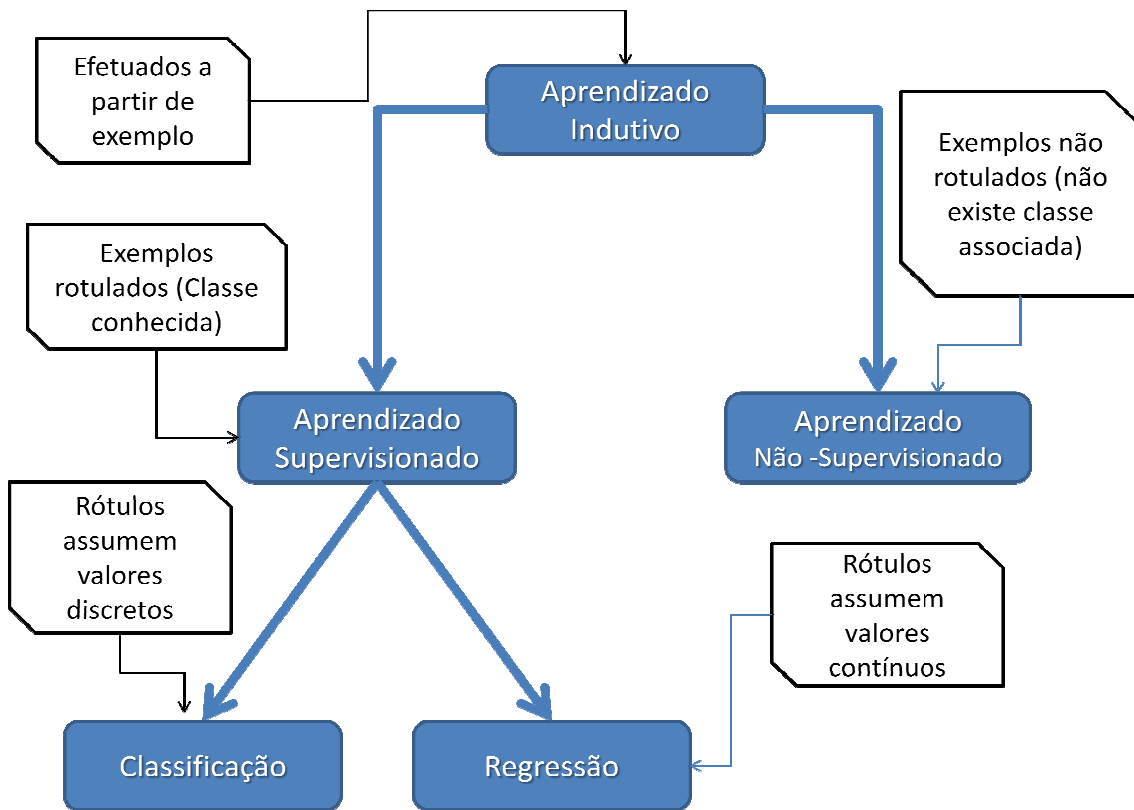


Figura 3.2 – Hierarquia de aprendizado (MORNARD, BARANAUSKAS, 2003) [Adaptado]

Outra subdivisão comum que pode ser feita para aprendizagem de máquina é quanto ao seu paradigma (MONARD & BARANAUKAS, 2003), sendo eles:

- Simbólico - Buscam aprender construindo representações simbólicas de um conceito através da análise de exemplos e contra-exemplos desse conceito. As representações simbólicas estão tipicamente na forma de alguma expressão lógica tais como árvores de decisão, regras ou redes semânticas.
- Protótipo ou Memorização (Instance Based) - Uma forma de classificar um exemplo é lembrar-se de outro similar cuja classe é conhecida e assumir que o novo exemplo terá a mesma classe. Essa filosofia exemplifica os sistemas baseados em exemplos, que classificam

exemplos nunca vistos através de exemplos similares conhecidos. Os algoritmos mais conhecidos neste paradigma são os de Vizinhos mais Próximos (Nearest Neighbours) e Raciocínio Baseado em Casos (Case Based Reasoning)

- Conexionista - A representação de uma Rede Neural envolve unidades altamente interconectadas e, por esse motivo, o nome conexionismo é utilizado para descrever a área de estudo.
- Genético - Este paradigma de aprendizado é derivado do modelo evolucionário de aprendizado. Um classificador genético consiste de uma população de elementos de classificação que competem para fazer a predição. Elementos que possuem um desempenho ruim são descartados, enquanto os elementos mais fortes proliferam, produzindo variações de si mesmos. Este paradigma possui uma analogia direta com a teoria de Darwin, na qual sobrevivem os mais bem adaptados ao ambiente.
- Estatístico - A ideia geral consiste em utilizar modelos estatísticos para encontrar uma boa aproximação do conceito induzido. Vários desses métodos são paramétricos, assumindo alguma forma de modelo, e então encontrando valores apropriados para os parâmetros do modelo a partir dos exemplos. Dentre os métodos estatísticos, destacam-se os de aprendizado Bayesiano, que utilizam um modelo probabilístico baseado no conhecimento prévio do problema, o qual é combinado com os exemplos de treinamento para determinar a probabilidade final de uma hipótese.

Para uma maior compreensão, além das divisões e subdivisões, existem definições importantes a serem citadas, tais como:

Exemplo:

Exemplo, caso ou registro (instance);

É um conjunto fixo de atributos;

Um exemplo descreve o objeto de interesse, tal como um paciente, exemplos médicos sobre uma determinada doença ou histórico de clientes de uma dada companhia;

Atributo:

Atributo ou campo (feature);

Uma única característica de um exemplo;

Classe:

Atributo especial que descreve o fenômeno de interesse (somente no Aprendizado Supervisionado)

3.3 - Árvore de Decisão (Paradigma Simbólico)

A árvore de decisão (DT, do inglês *Decision Tree*) é um algoritmo classificador, sendo um dos ramos na área de inteligência artificial. Mais especificamente, pertencente ao ramo da aprendizagem de máquina, pois possui a habilidade de aprender através de exemplos com o objetivo de classificar registros em uma base de dados. Além disso, a DT fornece uma maneira de visualizar graficamente as consequências de decisões atuais e futuras bem como os eventos aleatórios relacionados.

A DT pode ser usada como instrumento de apoio à tomada de decisão, sendo que uma das grandes vantagens de uma DT é a possibilidade de transformação/decomposição de um problema complexo em diversos subproblemas mais simples de uma forma recursiva, assim os subproblemas podem ser decompostos quantas vezes forem necessárias para uma melhor análise. Um dos pontos fortes das DTs é a sua eficiência em termos de processamento, além de fornecer resposta em representação simbólica simples e compreensível.

No final dos anos 50 surgiram os primeiros classificadores baseados em DT, baseando-se no trabalho de Hunt, no qual apresentava vários experimentos para indução de regras. Em seguida, Friedman desenvolveu o algoritmo CART, Quinlan desenvolveu o algoritmo ID3 e como sucessor deles surgiram os algoritmos C4.5 e C5.0 (Garcia, 2003).

A estrutura de uma DT é formada por nós que representam os atributos, por ramos (que são as ligações) provenientes dos nós e os nós folha, que representam as diferentes classes do conjunto de exemplos. Geralmente, o atributo mais efetivo dos dados é selecionado como o primeiro nó da árvore e seus valores limites são usados para criar os nós sucessores. Este procedimento é realizado em todos os nós criados, até que os nós folhas tenham somente uma classe. Uma árvore de decisão é essencialmente uma série de declarações if-then, que quando aplicados a um registro de uma base de dados, resultam na classificação daquele registro. A figura 3.3 ilustra a estrutura de uma DT, onde são apresentados os nós, os ramos e os nós folhas.

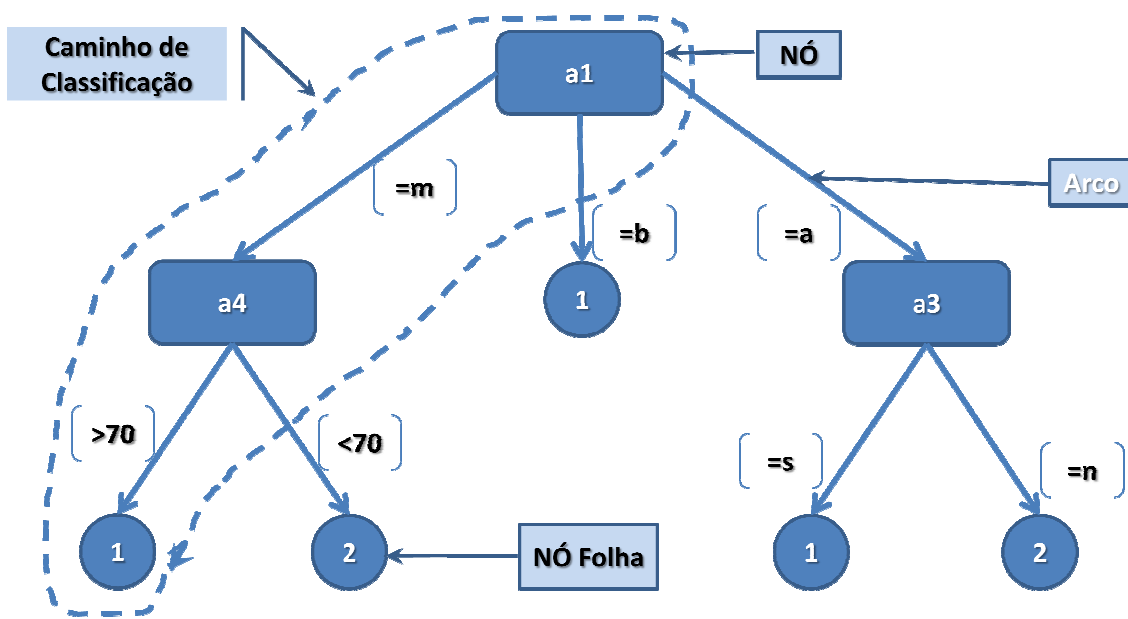


Figura 3.3 – Árvore de Decisão com caminho de classificação. Fonte: (Garcia, 2003)

A classificação de uma árvore de decisão consiste em seguir o caminho determinado pelos nós sucessivos ao longo da árvore, até chegar a um nó folha com a classe atribuída. Através do caminho descrito na figura 3.3 é possível derivar regras, que são geralmente utilizadas juntamente com a árvore. Algumas vezes as árvores são substituídas pelas regras, pois as DTs podem crescer muito. Um exemplo de regra pode ser extraído da figura 3.3:

“Se $a1=m$ e se $a4>70$ então 1”

Em todos os nós são aplicados testes definidos e um conjunto de exemplos, assim é decidido qual o caminho a percorrer na árvore durante o processo de classificação. Os testes dependem das características dos atributos, que podem ser quantitativos, categóricos ou com valores desconhecidos. Utiliza-se apenas um nó em cada teste, assim torna a árvore de fácil compreensão.

Os atributos quantitativos (numéricos) permitem uma grande variedade de testes, o que pode implicar em certa complexidade de cálculo. Em algoritmos para construção de DT, baseiam-se em testes do tipo “*atributo \leq ponto_de_quebra*” ou “*atributo $>$ ponto_de_quebra*”, nos quais devem ser considerados todos os valores pertencentes ao atributo, pois qualquer um deles pode ser o valor de quebra.

No entanto, os atributos categóricos (nominais) têm uma abordagem diferente em relação aos quantitativos. Pode-se criar um ramo para cada valor do atributo, o que torna a DT bastante detalhada, porém tende a criar uma DT com grande número de ramos, ou seja, com grande dimensão. Outra solução seria a criação de nós binários, atribuindo a um dos ramos um valor eleito, por exemplo, o que possui o maior número de casos, enquanto o restante ficaria no outro ramo, essa é uma solução simples e inteligível, porém é uma solução bastante limitada. Entretanto, outra possível abordagem é o agrupamento de valores característicos em dois conjuntos, que se baseia na criação de dois subconjuntos de valores associados ao ramo esquerdo e ao direito.

Já para valores desconhecidos, eles podem ser descartados, caso haja uma boa quantidade de exemplos sem falhas na base de treino. Também podem receber um valor que seja considerado provável ou mesmo dentro de uma densidade de probabilidade para acontecimentos.

Um modelo de pseudocódigo para indução de árvores de decisão pode ser:

```

1   Se CRITERIOPARADA(exemplos)
2       ESCOLHECLASSE(exemplos)
3   Senão
4       melhor = ESCOLHEATRIBUTO(subAtributos; exemplos)
5       arvore =nova arvore com nó raiz= melhor
6       particao = ESCOLHEPARTIÇÃO(melhor)
7       while particao
8           exp =elementos de exemplos com melhor= p
9           subAvr = INDUCAOCARTEC4.5(exp; subA □ melhor)
10      ADICIONARAMOARVORE(p; subAvr)
11      PODAARVORE(Arvore)
12  PODAARVORE(arvore)

```

3.3.1 – Algoritmo de Árvores de Decisão (C4.5)

Em 1993, Ross Quinlan publica um trabalho intitulado “C4.5: *Programs for Machine Learning*”, onde este trata-se de um aprimoramento de outro algoritmo seu, o ID3 (Rokach et al, 2008), tornando-o possível trabalhar com atributos numéricos e/ou nominais (simbólicos). Além de adotar o sistema de podas (Garcia, 2003).

Uma característica interessante do algoritmo C4.5 é que diferentemente do CART, não é obrigatório fazer uma divisão binária no seu particionamento, podendo fazê-lo de duas maneiras: Um ramo distinto para cada valor do atributo nominal (gerando pequenas sub-árvores). Formação de agrupamentos de valores em vários conjuntos.

Essa habilidade de criar árvores menores, tornando-as mais facilmente compreensíveis e conseqüentemente suscetíveis a ter maior precisão são desejáveis ao induzir árvores, o que faz o algoritmo buscar a menor árvore possível. Contudo, é inviável garantir a minimização da árvore. O método EscolheAtributo() do algoritmo de indução faz uma busca gulosa, selecionando a característica que maximiza a divisão dos dados por meio de entropia.

A entropia caracteriza a impureza dos dados, num conjunto de dados, caracterizando a falta de homogeneidade dos dados de entrada em relação a sua classificação. Por exemplo, a entropia é máxima (igual a 1) quando o conjunto de dados é heterogêneo (Mitchell, 1997). Isto é, dado um conjunto de entrada S que pode ter c classes distintas, a entropia de S será dada por:

$$E(S) = \sum_{j=1}^c -p_j \log_2 p_j \quad (3.1)$$

Onde:

c é o numero de classes;

p_j é a proporção de dados em S que pertence a classe j .

Para um atributo A pertencente a um conjunto S , o ganho de informação fornece a medida da diminuição da entropia esperada quando utilizado o atributo A na partição do conjunto de dados.

Seja $P(A)$ o conjunto de valores que o atributo A pode ter e x um elemento desse conjunto, e seja S_x um subconjunto de S formado pelos dados em que $A = x$, a entropia que se obtém ao particionar S em função do atributo A , é dada por:

$$E(A) = \sum_{x \in P(A)} \frac{|S_x|}{|S|} Entropia_{S_x} \quad (3.2)$$

Assim o ganho de informação será dado por:

$$ganho(S, A) = E(S) - E(A) \quad (3.3)$$

Onde:

$E(S)$ é a medida de não homogeneidade do conjunto S ;

$E(A)$ é a medida de não homogeneidade estimada para o conjunto S , caso A seja escolhido para fazer a partição.

No método de ESCOLHERPATIÇÃO, o algoritmo atribui um ramo para cada valor do atributo. Embora permita extrair todo o conteúdo informativo, este tipo de partição tem como desvantagem a criação de número de ramos muitas vezes desnecessários, por isso é executado um algoritmo de poda para resolver esse problema.

Para o método CRITERIOPARADA() o algoritmo só para de dividir se cada folha contém casos de uma única classe, ou até não ter como particionar mais, porque os dois casos têm os mesmos valores para cada atributo, mas pertencem a classes diferentes.

O método PODAARVORE() do algoritmo, usa a poda baseada no erro. Ele permite utilizar o próprio conjunto de treino para efetuar a poda da árvore e tem a vantagem de não obrigar a separação do conjunto de treino, em conjunto de treino e conjunto de teste.

3.4 - Redes Neurais (Paradigma Conexionista)

As redes neurais artificiais (RNAs) são modelos matemáticos desenvolvidos com o intuito de executar tarefas comuns ao cérebro. Uma RNA, portanto, deve ter a capacidade de modelar a maneira como o cérebro realiza uma determinada tarefa.

Diante sua característica de construção e resultados obtidos as redes neurais tem sido empregadas em diversos tipos de problema os quais não apresentam um modelo determinístico ou de difícil implementação, como é o caso de problemas não lineares.

O cérebro é formado por uma enorme quantidade de células denominadas neurônios (neurônios biológicos). Para o modelo de uma rede neural, tem-se uma unidade de processamento chamado de neurônio artificial, que foi desenvolvido baseado no conhecimento adquirido ao longo do tempo a respeito das células nervosas do ser humano, simulando desta forma seu comportamento e sua estrutura. O neurônio artificial, conhecido também por nó ou elemento processador, irá simular as características fundamentais do neurônio natural, sendo, portanto, uma unidade de processamento das informações (Haykin, 2008).

É possível destacar a partir da Figura 3.4 três elementos básicos do modelo do neurônio artificial: pesos (sinapses), somador e função de ativação (Haykin, 2008).

Pesos (sinapses) ou elos de conexão - Cada uma das sinapses é caracterizada por um peso ou força sináptica, a qual irá ser multiplicada pelo sinal de entrada da referida sinapse, podendo assumir tanto valores positivos como negativos, assim como ocorre na rede neural biológica. Na figura 3.4 os pesos são representados por w_1 , w_2 e w_n .

Somador ou combinador linear - É a parte do neurônio artificial responsável pela soma de suas entradas, ponderados pelos seus respectivos pesos sinápticos.

Função de ativação - Tem a função de implementar a relação de entrada e saída do neurônio. A função de ativação pode ser do tipo linear ou não linear, sendo esta última mais utilizada. Dentre as funções de ativação ou função restritiva, destacam-se: função de limiar, função linear por partes e a função sigmóide (Haykin, 2008):

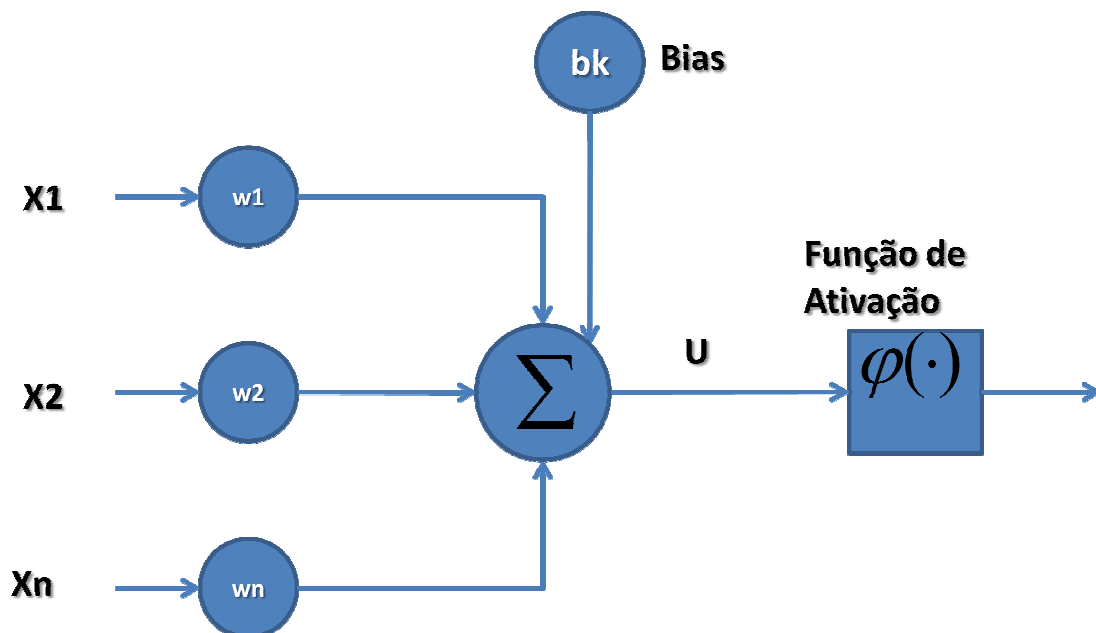


Figura 3.4 – Modelo do neurônio artificial (PEREIRA, 2009)

O modelo do neurônio artificial é composto por um vetor de entrada X , que na analogia ao neurônio biológico representa os dendritos e um conjunto ou vetor de pesos W correspondentes a cada entrada, emulando o comportamento das sinapses. A saída do modelo pode ser dividida em duas: uma primeira saída linear U , correspondendo à soma das entradas x_i do vetor X multiplicadas pelos seus respectivos pesos w_i do vetor W e uma segunda saída Y , representando a saída efetiva do modelo, a qual é obtida após a aplicação da função de ativação $\varphi(v_k)$ à saída v_k .

Este modelo de neurônio pode ser expresso pelas seguintes equações:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (3.3)$$

$$v_k = u_k + b_k \quad (3.4)$$

$$y_k = \varphi(v_k) \quad (3.5)$$

Ao conjunto de neurônios artificiais interagindo entre si através de suas conexões (sinapses) é dado o nome de Redes Neurais Artificiais (RNA), assim se forma uma rede de processamento de informações capaz de modelar a maneira como o cérebro humano realiza uma tarefa.

Uma rede neural é um processador paralelamente distribuído, constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torna-lo disponível para o uso. As principais características das redes são (Pereira, 2009):

- Capacidade para modelar fenômenos físicos não lineares;
- Mapeamento de entrada/saída, obtido através do aprendizado dos exemplos de treinamento;
- Necessidade de pouco conhecimento estatístico sobre o ambiente onde a rede está inserida;
- Adaptabilidade, em que os pesos sinápticos da rede são adaptados à medida que o meio ambiente sofre modificações;
- Informação contextual, onde cada neurônio é potencialmente afetado pela atividade dos outros neurônios da rede;
- Tolerância a falhas, devido à natureza distribuída da informação armazenada na rede neural;
- Armazenamento do conhecimento adquirido, através de forças de conexão entre neurônios, conhecidas como pesos sinápticos;
- Generalização, uma vez que a rede é capaz de responder adequadamente a entradas que não lhe foram apresentadas durante a fase de treinamento.

As redes neurais podem ser classificadas segundo critérios como:

- Tipo de arquitetura

Por número de camadas – Pode-se agrupar os neurônios em apenas uma camada, esse tipo de estrutura normalmente é usado em tarefas auto-associativas. Entretanto, os

neurônios podem ser agrupados em mais de uma camada entre a entrada e a saída, sendo conhecidas como camadas ocultas que capturam, por meio dos pesos sinápticos, as características dos padrões de entrada que lhe são apresentadas e geralmente realizam o mapeamento de classes ou regressão de dados.

Por tipo de conexões – As Redes alimentadas diretamente são as que a informação flui em uma única direção, ou seja, a informação se propaga da entrada para a saída (Barreto, 2002), como as perceptron multicamadas e a RBF. Enquanto podem existir redes que possuem pelo menos um laço de realimentação em sua estrutura. Como exemplos deste tipo de rede têm-se a rede de Hopfield, rede de Kohonen. Também existem as redes simétricas, cuja matriz de conectividade é simétrica, que é um caso particular de redes com ciclos (realimentação)

- Tipo de associação entre as informações de entrada e saída;

Auto-associativas - Durante a etapa de treinamento há o ajuste dos pesos sinápticos de modo a armazenar certos padrões de entrada. Com isso, a rede é capaz de responder com o padrão mais próximo dentre os armazenados a uma informação (entrada) incompleta ou ruidosa. Como exemplos destes tipos de rede têm-se a rede de Hopfield, a família de arquitetura ART, Mapas Auto-Organizáveis de Kohonen (Villanueva, 2011).

Hetero-associativas - Durante a etapa de treinamento há o ajuste dos pesos sinápticos de modo a armazenar certas associações de entrada/saída, fornecendo desse modo uma saída correspondente a uma determinada entrada. Inseridas nesse contexto pode-se citar a rede MLP (do inglês Multi-LayerPerceptron), a rede RBF (do inglês Radial Basis Function) (Villanueva, 2011).

- Tipo de mecanismo de aprendizagem;

A aprendizagem de uma RNA corresponde ao processo de ajustes dos pesos sinápticos, mediante aos estímulos recebidos pela rede, de tal forma a capturar o conhecimento do sistema (ambiente) no qual irá operar.

Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre (Haykin, 2008). O conjunto de procedimentos adotados para que haja esse aprendizado é chamado de algoritmo de aprendizado, sendo que podem existir centenas ou até mesmo milhares deles. Esses algoritmos podem ser supervisionados ou não.

- Tipo de procedimento de ajuste das conexões sinápticas.

Off-line - O ajuste dos pesos sinápticos da rede se dá em um momento anterior à sua colocação em operação. Portanto, depois de realizado o treinamento da rede, seus pesos se mantêm fixos.

On-line - A rede on-line não separa a fase de treinamento da fase de operação, uma vez que seus pesos são atualizados à medida que a rede neural recebe novos estímulos de entrada.

3.4.1 – MLP - Rede neural Perceptron de Múltiplas Camadas

As redes MLPs são redes neurais do tipo hetero-associativa, alimentadas diretamente e de múltiplas camadas. Uma de suas principais características é a capacidade de trabalhar com dados que não são linearmente separáveis. Em virtude disso, a rede MLP tem sido empregada com sucesso em diversas áreas do conhecimento, desempenhando tarefas como classificação de padrões, controle, aproximações de funções entre outras aplicações (Villanueva, 2011).

A estrutura típica de uma rede MLP contém uma camada de entrada que é responsável por apresentar os padrões de entrada à rede e não realiza processamento algum; uma ou mais camadas ocultas, funcionando como extratores de características e com a codificação interna dessas características traduzindo-se no ajuste dos pesos sinápticos; além de uma camada de saída, a qual tem a função de construir os padrões de resposta da rede.

A Figura 3.5 ilustra um tipo de arquitetura de uma rede MLP com duas camadas ocultas e uma camada de saída, sendo do tipo totalmente conectada, ou seja, um neurônio localizado em qualquer camada (oculta ou de saída) está ligado a todos os neurônios da camada anterior.

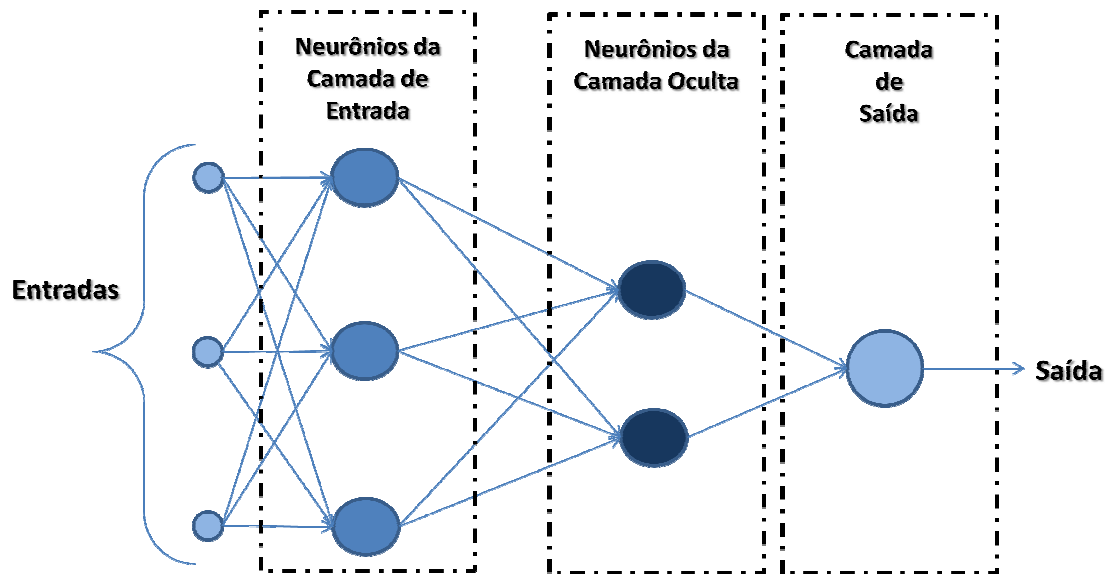


Figura 3.5 – Rede MLP com uma camada oculta (OLIVEIRA, 2005)

O fluxo do sinal de entrada é transferido para frente de camada em camada no sentido da direita para a esquerda, passando por todos os neurônios. Os sinais que chegam nestes neurônios são então multiplicados pelos seus respectivos pesos sinápticos e somados no corpo principal do neurônio. A saída final de cada neurônio é então obtida a partir da aplicação da função de ativação, sendo esta geralmente não linear, como as funções sigmóides.

De forma análoga ao modelo de neurônio apresentado na seção 3.2 deste capítulo, a saída de cada neurônio em uma rede MLP pode ser definida pelas seguintes equações:

$$y_j(n) = \varphi(u_j(n)) \quad (3.6)$$

$$u_j(n) = \sum_{i=1}^m w_{ji}(n) y_i(n) - x_0 \quad (3.7)$$

Em que,

W_{ji} – Peso que conecta a saída do neurônio i à entrada do neurônio j ;

y_i – Sinal de entrada do neurônio j .

Y_j – Sinal de saída do neurônio j .

X_0 - Limiar ou bias do neurônio j .

U_j – Potencial de ativação do neurônio j .

Pode-se dizer que o projeto de uma rede MLP leva em consideração três aspectos importantes: determinação do número de camadas ocultas, determinação do número de neurônios de cada camada oculta e ajuste dos pesos sinápticos da rede.

A respeito da quantidade de neurônios a ser utilizados na rede e o número de camadas dependem da dimensão dos dados de entrada e saída da RNA, onde é necessário um número maior de neurônios e/ou dependendo da complexidade é desejada (Neto, 2006).

No que diz respeito ao número de camadas e ao número de neurônios em cada uma destas camadas não há uma regra bem definida para tal especificação, sendo utilizadas técnicas de tentativa e erro ou algoritmos de otimização para obter uma melhor configuração desses parâmetros.

Com relação ao ajuste dos pesos, este é determinado mediante a utilização de um algoritmo de treinamento. Para redes MLP é utilizado um algoritmo baseado na regra de aprendizagem por correção de erro, denominado algoritmo de retropropagação, descrito na próxima seção.

3.4.2 – Algoritmo *Backpropagation* (Retropropagação)

O algoritmo *backpropagation* (retropropagação) é um algoritmo de treinamento supervisionado de redes de múltiplas camadas alimentadas adiante, baseado na correção de erro, cujo propósito é a obtenção dos ajustes dos pesos sinápticos da rede neural, sendo este algoritmo uma generalização do algoritmo conhecido como regra delta.

O algoritmo de retropropagação consiste em calcular o erro na saída da rede e propagar esse erro para as camadas anteriores, modificando os pesos para minimizar o erro na próxima saída. Esse algoritmo baseia-se no gradiente descendente, esse método tem como ideia de modificar os pesos proporcionalmente ao gradiente do erro. A direção do gradiente é onde o erro é minimizado. O método utiliza a 1ª derivada de todas as inclinações. Basicamente o algoritmo de retropropagação consiste nos seguintes passos:

1. Calcular o erro na saída da rede
2. Retropropagar o erro e modificar os parâmetros para minimizar o erro da próxima saída.

A aprendizagem por retropropagação do erro pode ser descrita como:

1. Um passo para a frente: propagação (o vetor de entrada é aplicado aos neurônios da rede e seu efeito se propaga através da rede, camada por

camada, até finalmente produzir o conjunto de saída. Durante este passo, os pesos da rede ficam fixos.

2. Um passo para trás: retropropagação (os pesos são ajustados de acordo com uma regra de correção de erro). A resposta real da rede é subtraída da resposta desejada para produzir um sinal de erro. Este sinal é propagado para trás através da rede, daí o nome retropropagação de erro. Os pesos são ajustados para aproximar a resposta da rede ao valor desejada, podendo ser ajustado passo a passo, por padrão ou por lote.

Na saída da rede como existe uma resposta desejável, é possível calcular o erro de resposta. Mas na camada oculta o erro não tem um sentido físico. Portanto, os neurônios de saída são as únicas unidades visíveis para as quais o sinal de erro pode ser diretamente calculado. Dessa forma, o algoritmo oferece um tratamento diferenciado aos neurônios da camada oculta e da camada de saída. O objetivo é minimizar o erro médio. A função custo é uma medida de desempenho. O somatório refere-se à soma dos erros de todos os neurônios da camada de saída.

Em sequência é apresentado o algoritmo de retropropagação (Haykin, 200).

O sinal de erro na saída do neurônio j , na iteração n , é definido através da equação (3.8), sendo o neurônio j um neurônio pertencente à camada de saída:

$$e_j(n) = d_j(n) - y_j(n) \quad (3.8)$$

Definindo o valor instantâneo da energia do erro para o neurônio j como $\frac{1}{2}e_j^2$, pode-se calcular o valor instantâneo da energia total do erro $E(n)$ e conseqüentemente a energia média do erro quadrado E_{med} , ou seja:

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (3.9)$$

$$E_{med} = \frac{1}{N} \sum_{n=1}^N E(n) \quad (3.10)$$

Em que,

C - Conjunto composto por todos os neurônios da camada de saída da rede.

N - Número total de padrões contidos no conjunto de treinamento.

Para a minimização de E_{med} os pesos são atualizados, podendo ser padrão a padrão, por época ou mesmo por lote. A convergência do backpropagation pode-se dá quando a diferença do valor de E_{med} entre duas épocas sucessivas atinge um valor satisfatório, ou seja:

$$|E_{med}(n+1) - E_{med}(n)| \leq \epsilon \quad (3.11)$$

A minimização do erro no algoritmo é determinada através do gradiente decrescente na superfície de erros do espaço de pesos, sendo que o vetor gradiente indica o sentido e direção de maior crescimento da função a partir do ponto em que foi calculado e o módulo do vetor indica a intensidade do crescimento desta função. O vetor gradiente para o algoritmo backpropagation pode ser definido através da equação (3.12) a seguir:

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{ji}(n)} \quad (3.12)$$

A correção Δw_{ji} a ser aplicada a w_{ji} é obtida pela equação (3.13):

$$\Delta w_{ji}(n) = w_{ji}(n+1) - w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \quad (3.13)$$

Onde η é o parâmetro da taxa de aprendizagem, sendo que o uso do sinal negativo na equação (3.13) indica que a atualização dos pesos é feita na direção da descida do gradiente no espaço de pesos.

A correção aplicada aos pesos sinápticos pode ainda ser definida através da equação (3.14):

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \quad (3.14)$$

Sendo,

$$\delta_j(n) = (d_j(n) - y_j(n)) \varphi'(v_j(n)) \quad (3.15)$$

Em que,

$\delta_j(n)$ - Gradiente local do neurônio j.

$\varphi'(v_j(n))$ - Derivada da função de ativação.

A equação (3.15) é válida somente para os neurônios pertencentes à camada de saída, uma vez que neste caso é possível utilizar a equação (3.8) para a determinação do erro $e(n)$. Tendo-se calculado $e(n)$, pode-se então obter diretamente o gradiente local $\delta_j(n)$ usando a equação (3.15).

No caso do neurônio j pertencer a uma camada oculta, não existirá uma saída desejada para este neurônio. Portanto, o sinal de erro é determinado recursivamente, levando em consideração os sinais de erro de todos os neurônios conectados diretamente ao neurônio j .

Sendo assim, o gradiente local do neurônio j é calculado utilizando a equação (3.16):

$$\delta_j(n) = \varphi'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (3.16)$$

O fator $\varphi'(v_j(n))$ envolvido no cálculo do gradiente local $\delta_j(n)$ depende unicamente da função de ativação associada ao neurônio oculto j . O fator restante envolvido no cálculo depende de dois termos:

- $\delta_k(n)$, requer conhecimento dos sinais de erro $e_k(n)$, para todos os neurônios que se encontram na camada imediatamente à direita do neurônio oculto j e que estão diretamente conectados ao neurônio j ;
- $w_{kj}(n)$, está relacionado aos pesos sinápticos associados com estas conexões.

3.5 - Estimação de Erro do Classificador

É de fundamental importância estimar a qualidade do classificador utilizado, na forma de percentual de erro do modelo gerado. Pode definir-se o erro de um classificador como sendo (4.5).

$$P_{e\%} = \frac{N_{erros}}{N_{casos}} \quad (3.17)$$

Onde:

$P_{e\%}$ é o percentual de erro;

N_{erros} é o número de casos classificados erroneamente;

N_{casos} é o número total de casos testados.

Uma importante ferramenta para a análise de desempenho de um classificador é a chamada de matriz de confusão, tabela 3.1.

Tabela 3.1 – Exemplo de Matriz de Confusão.

Matriz de Confusão		Classe Real	
		Seguro	Inseguro
Classe Predita	Seguro	PV	PF
	Inseguro	NF	NV

A matriz de confusão é uma matriz que contém os dados dos testes realizados no modelo induzido, tais como:

- PV – Positivos verdadeiros, ou seja, o número de exemplos classificados como “Seguro” que são verdadeiramente “Seguro”.
- PF – Positivos falsos, ou seja, o número de exemplos classificados como “Seguro” que são do tipo “Inseguro”.
- NV – Negativos Verdadeiros, ou seja, o número de exemplos classificados como “Inseguros” e são verdadeiramente “Inseguros”.
- NF – Negativos Falsos, ou seja, o número de exemplos classificados como “Inseguro” que são do tipo “Seguro”.

A partir da matriz de confusão é possível calcular alguns índices de desempenho para o classificador, estes são:

A taxa de acerto que é definida pela equação 3.18

$$P_{a\%} = \frac{PV + NV}{PV + PF + NV + NF} \quad (3.18)$$

A precisão de classe, que é a proporção de positivos verdadeiros do total de exemplos classificados como positivos dados pela equação 3.19.

$$P_{c\%} = \frac{PV}{PV + PF} \quad (3.19)$$

E a sensibilidade que é a proporção de exemplos positivos corretamente classificados, e equação 3.20 define a sensibilidade positiva.

$$P_{s\%} = \frac{PV}{PV + NV} \quad (3.20)$$

Para que seja possível avaliar o poder de classificação de um modelo é necessário que haja um conjunto de dados para testá-lo, verificando se as regras do classificador conseguem classificar de forma correta esse conjunto.

Acontece que muitas vezes não existem dois ou três bancos de dados separados, sendo, por exemplo, um para treino e outro pra teste e um terceiro para validação. Quando ocorre essa situação, lança-se mão de algoritmos que particionam o conjunto de dados que seria usado apenas para treino, permitindo assim que os modelos gerados possam ser testados, mesmo havendo apenas um conjunto de dados.

Serão apresentados dois algoritmos que permitem esse particionamento do conjunto de dados, o método Holdout que normalmente é usado para grandes bases de dados e o Cross-Validation (validação cruzada), normalmente utilizado para conjuntos de dados considerados pequenos.

3.5.1 - HoldOut

Nesse método os exemplos são divididos em uma percentagem fixa p de treinamento e $(1-p)$ para teste, normalmente se considera $(p > \frac{1}{2})$. Um valor muito utilizado para p é 0,7, ou seja, 70 % do total do conjunto. O tempo para calcular esse tipo de método de particionamento e validação não é considerado grande. A figura 3.6 mostra com é realizado esse processo.

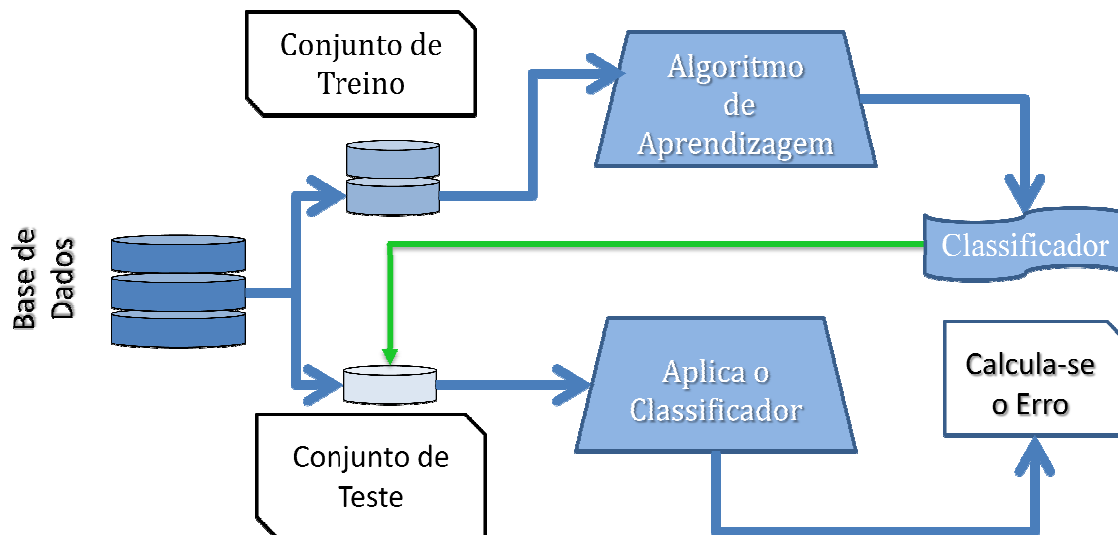


Figura 3.6 – Método HoldOut

Porém, dependendo de como é realizada a divisão do conjunto de treino e teste, essa avaliação pode ter uma alta variância. Para que os resultados não dependam da divisão dos dados (exemplos), pode-se calcular a média de vários resultados de *holdout*.

3.5.2 - Validação Cruzada (*Cross-Validation*)

É um dos algoritmos mais utilizados para conjuntos de dados que não possuem muitos exemplos. Ele consiste em usar os mesmo dados, de forma repetida, porém divididos de forma diferente, ou seja, o conjunto de dados é dividido aleatoriamente em k partições mutuamente exclusivas (*folds*) de tamanho aproximadamente igual (n/k).

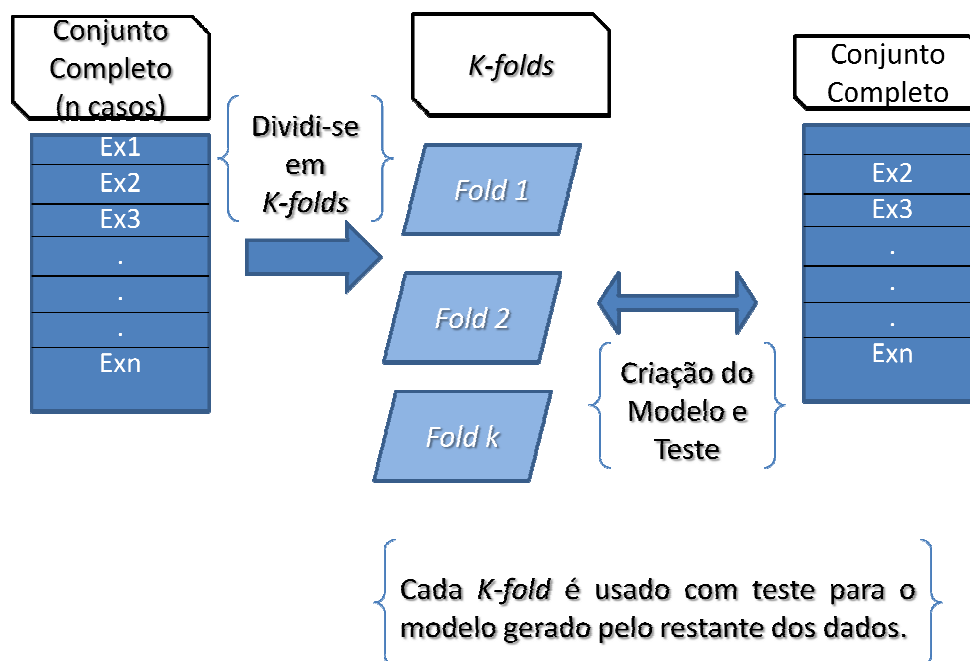


Figura 3.7 – Exemplo de Cross-Validation

Os exemplos de $(k-1)$ *folds* são independentemente usados no treinamento e os classificadores obtidos são testados com o *fold* remanescente. O processo é repetido r vezes, e a cada repetição um *fold* diferente é usado para teste de modelo.

A estimativa do erro verdadeiro é obtida como a média dos erros de cada experiência, equação 3.21:

$$E = \frac{1}{k} \sum_{i=1}^k E_i \quad (3.21)$$

3.6 - Extração de Conhecimento em Redes Neurais

Em redes neurais, todo conhecimento adquirido na etapa de treinamento está contido nos pesos sinápticos, isto é, está nos valores dos elementos da matriz de pesos que ligam as entradas as camadas ocultas e nos pesos que conectam as camadas ocultas entre si e a camada de saída.

Para se extrair regras do tipo SE-ENTÃO de uma RNA é necessário se fazer uma análise minuciosa desses conjuntos de pesos. O procedimento para essa tarefa são as seguintes: encontrar as entradas mais significantes, construir um relacionamento de dados de treinamento e teste, por fim a extração de uma regra geral (Lin, 2012).

O primeiro passo é selecionar o neurônio que contém a saída desejada, a partir disso, busca-se conhecer quais entradas são significantes para ativar a classe de saída. Essa ativação é dita significante quando o valor do peso que conecta o neurônio de saída escolhido a um neurônio da camada oculta tem valor positivo. Caso haja mais de uma camada oculta, busca-se nos pesos que ligam ao neurônio analisado na camada posterior valores positivos, repetindo esse processo até encontrar uma entrada.

Ao encontrar a entrada que ativa o neurônio de saída pode-se extrair a seguinte regra: Se o neurônio de entrada é significante então seleciona o neurônio de saída.

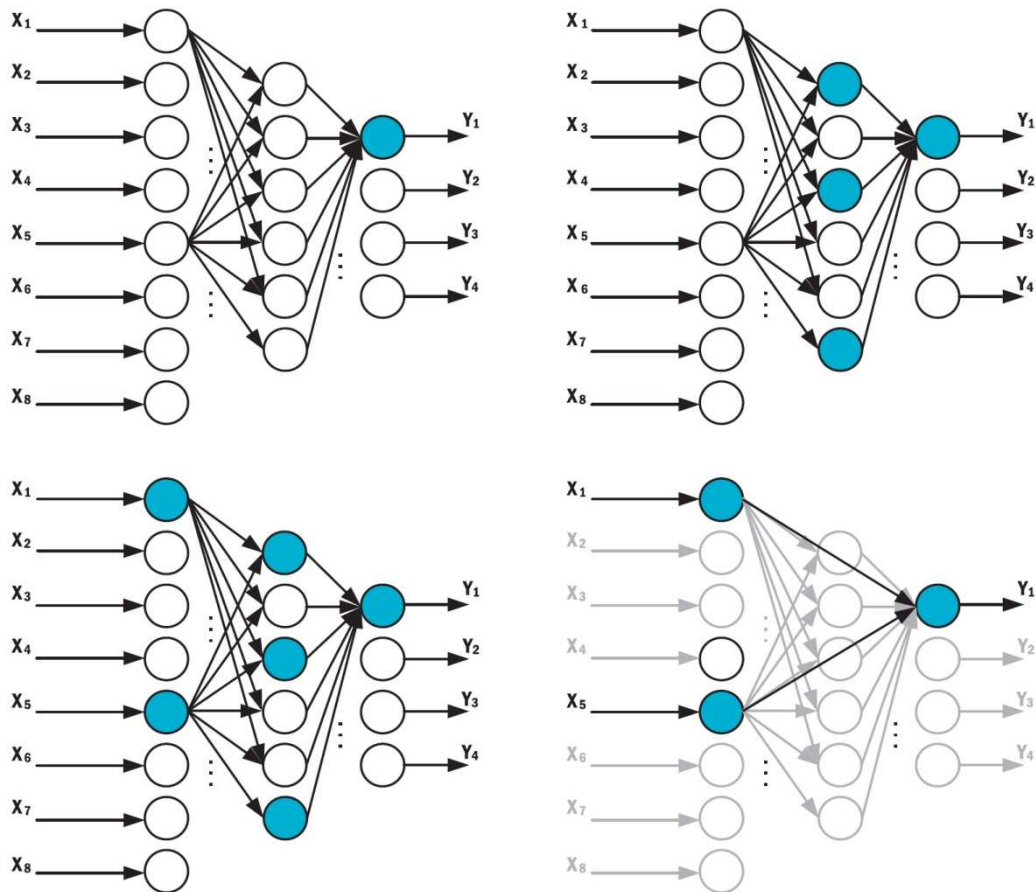


Figura 3.8 – Identificação de entradas significantes em uma MLP (Lin, 2012).

A figura 3.8 apresenta a identificação dos neurônios que mais ativam uma saída requerida. Contudo um problema que surge nessa análise é que mais de um neurônio pode ativar a saída desejada. Para solucionar isso, faz-se a escolha pelo peso com maior valor positivo, pois quanto maior o valor do peso, maior a influencia do mesmo na saída.

Entretanto, é importante salientar que a regra gerada pode contar com mais de uma entrada, haja vista que mais de um neurônio pode ter uma ativação significativa na saída. Por isso, a regra gerada pode ser dessa forma:

Se entrada X1

Ou entrada X5

Então saída Y1

Todavia, a regra gerada não indica valores de quanto alterar na variável escolhida a fim de ativar a saída escolhida, essa regra gerada indica as variáveis que mais influenciam na saída escolhida.

Capítulo 4 – Metodologia para DSA e Projeto de Controle Preventivo

4.1 - Introdução

Os modernos sistemas elétricos de potência possuem uma vasta gama de informações provindas de seus sistemas de medição, gerando assim enormes bancos de dados, no qual contém informações de diversas condições operacionais do sistema em geral, como bases de dados da atuação de alarmes e dispositivos de proteção, níveis de carregamento das linhas, assim como temperaturas de diversos equipamentos, além de medidas elétricas como tensão, corrente e potência. Em geral armazena-se uma enorme gama de variáveis que podem ser úteis para as diversas análises no sistema.

Essa grande variedade de informações pode ser tratada por diversos processos, mineração de dados, por exemplo, que podem descobrir conhecimentos ou associações que até então não era percebido pelos engenheiros e operadores das empresas. Assim, esses conhecimentos podem muito úteis para análises de planejamento, manutenção e outras atividades do setor. Todavia, é necessário estudar qual é a reação do sistema quando submetido a situações adversas, como saídas de grandes blocos de cargas, retirada de geração, retiradas de linhas, curto-circuito e outras ocorrências possíveis.

Contudo, é importante ratificar que quase na totalidade do tempo, os SEPs operam em condições consideradas normais, ou seja, sem grandes perturbações no modo de operação do mesmo. Logo, obter uma base de dados, suficientemente grande e eficiente, que leve em consideração todos os possíveis estados operacionais dentro de todas as possíveis contingências, previsíveis ou não, a que o sistema está sujeito, podem demandar anos e anos de coleta de dados nos históricos operacionais.

Para que seja possível a análise ou até mesmo o desenvolvimento de ferramentas para a avaliação de um sistema de potência, o uso de simulações é de fundamental importância, pois nelas são modelados os sistemas e são aplicadas as contingências desejadas, além do que, dificilmente, encontram-se disponíveis dados rotulados nas múltiplas bases de dados das empresas do setor elétrico, que podem ser utilizadas para tal finalidade.

A seguir serão apresentadas as ferramentas utilizadas, assim como as características de simulações para a criação da base de dados para treinamento da árvore

de decisão e da rede neural para avaliação de segurança dinâmica e projetos de controle preventivo, no contexto da estabilidade transitória.

4.2 - Ferramentas de Análise

Para a análise da estabilidade transitória e criação de modelos baseados em máquinas de aprendizado foram utilizados os programas PSTv3 e o RapidMiner 5, respectivamente.

4.2.1 - PSTv3

A ferramenta “*Power System Toolbox*” Versão3 (PSTv3) é um programa desenvolvido por Joe Chow e Graham Roger em MatLab (PST webpage), muito reconhecido no meio acadêmico. Tal ferramenta é um Toolbox que contém rotinas para o estudo da análise de sistemas elétricos tendo como foco o estudo do fluxo de carga, estabilidade de tensão, análise modal e simulação dinâmica no domínio do tempo.

Existem no mercado outras ferramentas que possibilitam o estudo dos fenômenos citados. Optou-se pelo PSTv3 por ele apresentar suas rotinas em código Matlab, possibilitando a geração de novas rotinas com maior facilidade, mantendo o seu código original inalterado, uma vez que o PSTv3 não é uma ferramenta Open Source. São utilizadas duas rotinas para geração da base de dados, *s_simu* (estudo de estabilidade transitória) e *loadflow* (estudo de fluxo de carga).

A principal rotina desenvolvida que foi acoplada ao programa PSTv3 consiste de uma rotina para execução em “*loop*” das várias simulações dinâmicas no domínio do tempo, no sentido de formar um extenso banco de dados automaticamente, considerando vários cenários de operação e aplicação de faltas trifásicas em todas as barras do sistema, além da inclusão do cálculo do índice de desempenho dinâmico, o qual fornece o rótulo ao atributo objetivo ao banco de dados.

Outras rotinas desenvolvidas foram:

- Modificação nas potências ativas e reativas geradas pelas máquinas de maneira, sendo de maneira aleatória dentro de um intervalo definido entre 0 e a capacidade nominal da máquina;
- Modificação dos valores de tensão nas barras, aleatoriamente, dentro do intervalo dos limites definidos;
- Aplicação de curto-circuito em todas as barras do sistema.

O desenvolvimento dessas rotinas foi necessário para a geração de uma extensa base de dados, a fim de treinar as máquinas de aprendizado via rede neural e árvore de decisão.

4.2.2 - Rapidminer

O *RapidMiner* é uma ferramenta que realiza inúmeras tarefas de mineração de dados, tais como: associações, agrupamento e classificação (rapidi.com), reconhecida no mercado de ferramentas para mineração. O RapidMiner é baseado nos pacotes de classe Weka, sendo que o próprio RapidMiner possui uma biblioteca para aplicações no padrão Weka. Contudo, a interface mais amigável e o fato de suportar uma gama maior de extensões de dados se comparado ao Weka, o torna mais prático e fácil de utilizar.

Neste trabalho o *RapidMiner 5.0* é utilizado para realizar a tarefa de classificação por meio do aprendizado automático de máquina, utilizando a árvore de decisão e a rede neural. O *RapidMiner* possui diversos algoritmos, que estão disponíveis em forma de operadores, por exemplo, os algoritmos de classificação, associação, agrupamentos, seleção de atributos, testes e validações, funções de tratamentos de dados e diversas outras ferramentas.



Figura 4.1 – Características da Interface do RapidMiner 5.0

A montagem dos modelos para a execução das tarefas no *Rapidminer* é feita conectando os blocos com as funções desejadas, a figura 4.1 ilustra a montagem modelo na ferramenta para a criação de uma árvore de decisão.

O lado esquerdo da figura 4.1 contém todos os blocos de operadores utilizados e suas respectivas conexões, e o centro é o local onde são inseridos e conectados os operadores. Enquanto que a região mais a direita é o local para alteração dos parâmetros dos operadores.

4.3 - Construção da Base de Dados

Para que um modelo baseado em técnicas de aprendizado de máquinas seja confiável e eficiente, é de fundamental importância que a base de dados de treinamento do algoritmo contemple de forma abrangente as possíveis entradas a qual o modelo estará sujeito após sua implementação. No caso da avaliação da estabilidade transitória de sistemas de potência, o ponto de operação é um fator determinante para se avaliar o comportamento do sistema após uma grande perturbação. Neste caso, é necessário avaliá-la a partir de dados como: níveis de geração de potência ativa e reativa, patamares de cargas ativa e reativa, níveis de tensão nas barras do sistema, níveis de carregamento nas linhas de transmissão, defasagens angulares entre barras do sistema, circuitos desconectados, entre outras possíveis variáveis.

Com o intuito de tornar a base de dados de treino mais eficiente, algumas medidas foram adotadas na construção das rotinas utilizadas no PSTv3.

4.3.1 - Seleção das Condições de Operação

A escolha das condições de operação do sistema visa introduzir na base de dados a maior quantidade de pontos de operação possíveis. Esses pontos diferem entre si em:

- Potências ativas e reativas geradas (barras PV) – As potências ativas nas barras de geração são geradas aleatoriamente dentro do intervalo de zero ao valor da potência nominal da máquina.
- Tensões terminais em barras com tensão controlada (barras PV) – São geradas para as tensões nas barras PV, aleatoriamente, entre os limites em p.u..
- Patamares de cargas nas barras PQ (barras de carga) - São gerados 96 pontos de cargas, que variam dentro de uma faixa de valores normalizados da potência máxima de carga instalada no sistema. A faixa de valores é entre 0,2 a 1,2, com acréscimo de 0,01042 para cada patamar

de carga. Esses valores abrangem as tradicionais curvas de cargas utilizadas para estudos de fluxo de carga, por exemplo.

4.3.2 - Seleção das Contingências

Para simular condições corriqueiras de operação e também um maior enriquecimento da base de dados para o estudo da estabilidade transitória é adotada duas situações topológicas: topologia completa e retirada de um elemento da rede elétrica (Contingência N-1). Para as duas situações topológicas, a cada simulação é aplicada uma falta trifásica em uma linha de transmissão, tanto no sentido da barra A para a barra B (a falta é aplicada próximo à barra A) quanto no sentido barra B para barra A (a falta é aplicada próximo a barra B). O tempo de eliminação da falta considerado foi de 100 ms para a abertura do disjuntor da extremidade da barra A e 140 ms para a abertura do disjuntor da extremidade da barra B, o mesmo tempo de eliminação foi considerado para o sentido barra B para barra A. Esse tempo de eliminação de falta foi adotado por ser um tempo médio para a atuação dos relés e disjuntores.

4.3.3 - Detecção de Ilhamento

Neste trabalho utilizou-se um método para identificação de casos de ilhamento baseado na fatoração triangular da “matriz incidência barra-ramo”, sendo capaz de identificar todas as possíveis ocorrências de ilhamento, para a perda de um equipamento ou linha (critério N-1) (Theodoro et al. , 2010). Uma rotina do método de identificação de ilhas foi desenvolvida neste trabalho, a fim de filtrar as ocorrências de ilhamento da base de dados rotulada. Os casos que são detectados como provocadores de ilhamento não são simulados.

4.3.4 - Rotulação

Em um sistema multimáquinas, a instabilidade transitória é definida como a perda de sincronismo entre as máquinas geradoras logo após o curto-circuito, ou seja, ainda na primeira oscilação. É comum utilizar apenas a diferença angular entre as máquinas considerando uma máquina como referência.

Neste trabalho utilizou-se o cálculo de um índice de desempenho dinâmico, como critério para avaliação da estabilidade transitória, por meio do ângulo do centro de inércia do sistema δ_{COI} (Amjady, 2004), dado por:

$$\delta_{COI} = \frac{1}{M_t} \sum_{i=1}^{NG} M_i \delta_i \quad (4.1)$$

Onde:

δ_i - É o ângulo interno (ângulo de carga) de cada gerador;

NG – É o número de geradores conectados no sistema.

M_i - É a constante de inércia de cada gerador e é dado por:

$$M_i = \sum_{i=1}^{NG} \frac{2H_i}{2\pi f} \quad (4.2)$$

H_i - É a constante de inércia de cada máquina, dado em segundos.

A partir do índice δ_{COI} calcula-se a diferença absoluta δ_{iCOI} entre o ângulo interno de cada máquina em relação ao centro de inércia do ângulo, e caso essa diferença ultrapasse o valor máximo de 120° no último passo de integração, o sistema é dito instável ou inseguro do ponto de vista da estabilidade transitória.

$$\delta_{iCOI} = |\delta_i - \delta_{COI}| \quad (4.3)$$

$$\delta_{max} < 120^\circ \quad (4.4)$$

4.3.5 - Atributos Selecionados (Base de Dados)

Uma vez que o principal objetivo da metodologia proposta é avaliar se uma perturbação pode causar um problema de segurança dinâmica em uma determinada condição operacional, os atributos devem conter informações suficientes que capturem precisamente em tempo real o comportamento do sistema diante de contingências.

Os atributos são variáveis que apresentam informações sobre a condição operacional do sistema, dessa forma, os atributos selecionados na ferramenta são:

- a) Potência Ativa Gerada (P_G) para as barras $V\theta$ e PV;
- b) Potência Reativa Gerada (Q_G) para as barras $V\theta$ e PV;
- c) Magnitudes das Tensões (V) para as barras $V\theta$ e PV;
- d) Ângulos de Fase da Tensão (θ) para as barras $V\theta$ e PV;

e) Potência Ativa da linha entre a barra “i” a barra “j”, P_{ij} ;

f) Potência Reativa da linha entre a barra “i” a barra “j”, Q_{ij} ;

Outro importante quesito que influenciou na escolha das variáveis é a facilidade de se obter essas informações em tempo real. As variáveis escolhidas podem ser diretamente oriundas de um sistema SCADA (Supervisory Control and Data Acquisition) ou de PMUs (Phasor Measurement Units).

Contudo, são utilizados mais dois atributos nominais (simbólicos), que são a configuração topológica de operação do sistema, que em situações práticas pode ser fornecido pelo configurador de redes e o local de aplicação de curto, informação útil para planejamento. A figura 4.2 ilustra de maneira sucinta o passo a passo da geração da base de dados, desde a escolha do caso base até a criação do banco de dados.

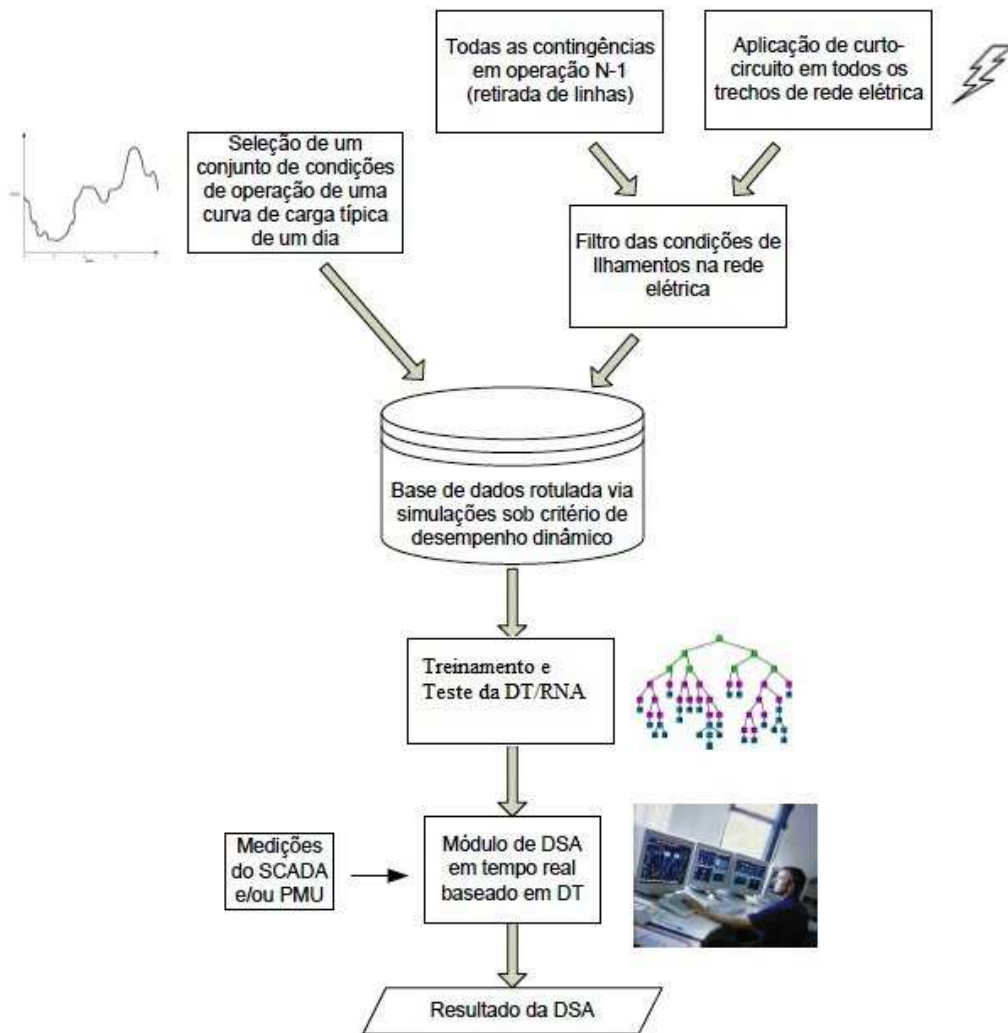


Figura 4.2 - Fluxograma de passos da simulação

Como pode ser observado na Figura 4.2, as técnicas de DT e RNA são utilizadas para solucionar o problema da DSA em tempo real, nesta presente dissertação, no sentido de identificar quando um SEP está sujeito a uma perturbação crítica, como um curto-circuito que pode levar para a instabilidade transitória. De posse do modelo obtido da máquina de aprendizado, as regras (conhecimento) embutidas no modelo são utilizadas em projetos de controle preventivo para restaurar a segurança do SEP contra perturbações críticas.

Capítulo 5 - Resultados e Análise

5.1 Introdução

Neste capítulo são apresentados os resultados da análise comparativa entre o uso de DTs e RNAs aplicadas para DSA em tempo real e para o auxílio a tomadas de ações preventivas para garantia da segurança no contexto da estabilidade transitória. Tal análise comparativa mostrará o desempenho de cada técnica com relação ao percentual de acerto, predição de classe e sensibilidade do modelo, além das vantagens e desvantagens com relação à interpretabilidade e flexibilidade do conhecimento obtido pelo modelo de cada técnica.

Além disso, são apresentados os nomogramas (regiões de segurança e seus contornos) gerados a partir das regras extraídas da árvore de decisão, com a finalidade de auxiliar no cálculo de margens de segurança dinâmica.

5.2 - Estudos de Caso

A análise comparativa proposta foi testada no sistema elétrico New England do IEEE, o qual representa um sistema de grande porte, no que tange a carga instalada, largamente utilizado no meio acadêmico para estudos de estabilidade transitória. Tal sistema elétrico consiste de 39 barras, 9 geradores síncronos e uma barra infinita (conectada na barra 39), de acordo com o diagrama unifilar mostrada na Figura 5.1, com suas respectivas áreas. Cada gerador síncrono é equipado de Regulador Automático de Tensão (RAT), e o trecho entre as barras 6 e 31 é composto por um transformador com mudança de tape sob carga ou (OLTC, do inglês *On Load Tap Changer*).

Inicialmente, a base de dados “rotulada” foi gerada a partir da execução de 62.322 simulações no domínio do tempo, sob uma gama de cenários operativos. A base de dados completa foi dividida inicialmente em quatro bancos de dados menores, sendo um para cada área do sistema New England, para fins de redução da dimensionalidade da base dados, e conseqüentemente do modelo de cada máquina de aprendizado em estudo. Em seguida, foram extraídas de cada base de dados já reduzida (correspondente a uma dada área), novas bases de dados associadas a cada perturbação (curto-circuito trifásico) aplicada em cada trecho da área em questão. Por fim, foi realizado o tratamento necessário dessas novas bases de dados, para o então fornecimento aos

algoritmos de classificação da DT C4.5 e da RNA MLP, respectivamente, a figura 5.1 ilustra a divisão da base de dados a partir do diagrama unifilar do sistema New England.

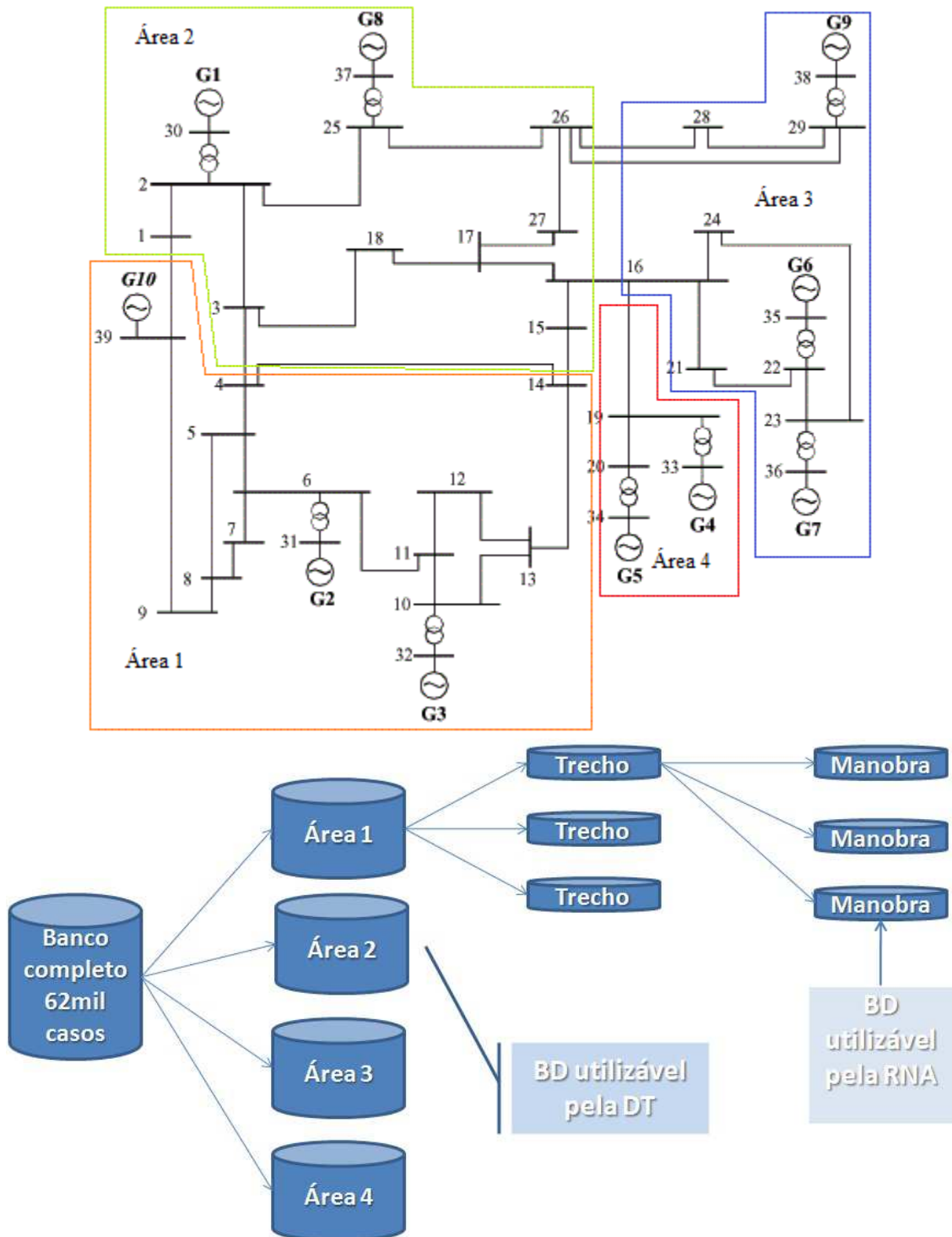


Figura 5.1 – Diagrama Unifilar do Sistema Elétrico New England (10 Geradores 39 Barras) e divisão da base de dados.

5.2.1 - DSA via Sistemas Inteligentes

Atualmente, a DSA com foco na estabilidade transitória tem por objetivo identificar as contingências críticas de um SEP, por meio de exaustivas simulações no domínio do tempo que demandam um elevado esforço computacional. No entanto, uma ferramenta computacional para análise em tempo real é necessária, tendo em vista as frequentes mudanças dos pontos de operação e da topologia de um SEP.

Por esse motivo, o uso de sistemas inteligentes para DSA em tempo real de SEPs torna-se a cada dia mais necessário, haja vista que a complexidade dos sistemas aumenta em grandes proporções, dificultando a cada dia a operação em tempo real.

Com relação à aplicação do uso de DTs e RNAs na avaliação da estabilidade transitória, o presente trabalho visa classificar o estado de operação do sistema em seguro ou inseguro, para um conjunto de perturbações (curto-circuito trifásico) que podem ocorrer no SEP em estudo.

Nas seções seguintes 5.2.1.1 e 5.2.1.2 são mostrados os resultados e análises do uso das árvores de decisão C4.5 e das redes neurais MLP, respectivamente, na avaliação da estabilidade transitória.

5.2.1.1 - DSA utilizando DT

Para aplicação da DT na DSA com foco na estabilidade transitória, utilizou-se como exemplo a base de dados correspondente a Área 1 do sistema New England. É importante destacar que a base de dados da área 1 consiste somente de atributos referentes aos componentes conectados a esta área, com exceção do atributo nominal referente as contingências ou manobras em operação, na condição N-1, que por sua vez correspondem as quatro áreas do sistema. Dessa forma os atributos para este estudo de caso da área 1 são dada por:

- Potências ativas e reativas nos geradores síncronos 03 e 02, e no barramento infinito (denominado neste trabalho de gerador 10);
- Tensões (módulo e ângulo de fase) nas barras 05, 06, 08, 09, 10, 11, 12, 13, 14, 31, 32 e 39;
- Fluxos de potência ativa e reativa nas linhas que estão dentro da Área 1.
- Todas as contingências ou manobras em operação N-1, além da topologia completa. Sendo que a contingência em operação N-1 pode ser proveniente de outra área.

- Atributo alvo denominado *Status*, assumindo dois possíveis valores nominais (simbólicos): seguro ou inseguro.

Uma grande vantagem da árvore de decisão, dependendo do algoritmo, é a capacidade de receber tantos atributos nominais (simbólicos) como numéricos, eliminando em muitos casos a necessidade de normalização ou discretização dos dados. Para aplicação em DSA no contexto da estabilidade transitória de SEPs, não há essa necessidade de normalização de dados, muito pelo contrario, é inclusive conveniente lançar mão dessa habilidade da DT de receber os dados de entrada diretamente de um sistema (SCADA, do inglês “Supervisory Control and Data Acquisition”) ou de unidades de medição fasorial ou (PMU, do inglês “Phasor Measurement Units”).

A figura 5.2 ilustra parte da DT criada referente à área 1, onde a estabilidade transitória é avaliada para um curto-circuito aplicado nas extremidades da linha conectada entre as barras 8 e 9 (trecho 8-9), somente para duas distintas configurações topológicas, na condição pré-falta N-1. Foram apresentados somente dois ramos da DT para análise desse estudo de caso, para uma maior compreensão dos resultados obtidos, haja vista que seu modelo completo contém um ramo para cada manobra efetuada no sistema. O método de particionamento da base de dados utilizado para treino e teste foi o método de *Cross Validation*.

Os dois ramos da DT de classificação ilustrados na figura 5.2 consistem de regras facilmente interpretáveis que descrevem a correlação dos atributos críticos com relação à segurança, para um curto-circuito aplicado no trecho 8-9. O primeiro nó da DT diz respeito à contingência ou manobra em operação N-1. O ramo à esquerda da DT consiste das regras críticas quando a linha 17#18 (trecho entre as barras 17 e 18) está fora de operação e o ramo a direita da DT apresenta as regras quando a linha 2#25 (trecho entre as barras 2 e 25) está desligada.

As regras encontradas pelo algoritmo que possuem maior influência com relação à segurança do sistema, quando a linha 17#18 está fora de operação, são por ordem hierárquica: 1) fluxo de potência ativa entre as barras 06 e 11 (P06-11) maior ou igual que 0,155 p.u.; e 2) módulo da tensão na barra 07 (V07) maior ou igual que 0,926 p.u.; e 3) módulo da tensão na barra 12 (V12) maior ou igual que 0.971 p.u.; e 4) fluxo de potência ativa entre as barras 10 e 11 (P10-11) menor ou igual que 4,467 p.u. Enquanto que a retirada da linha 2#25 apresenta como regras críticas, por ordem: 1) ângulo de tensão na barra 13 (Ang_b13) menor ou igual que $-5,101^\circ$; e 2) potência ativa do

gerador síncrono 02 (P02) maior ou igual que 1,402 p.u. e 3) potência ativa do barramento infinito (PG10) maior ou igual que 8,285 p.u.

Pode-se notar que a DT ilustrada na Figura 5.2 indica um caminho cujas regras devem ser respeitadas para garantir o sistema em um ponto de operação seguro do ponto de vista da estabilidade transitória. Pode-se observar também que o número de variáveis que são indicadas nos ramos da DT é bem menor que à quantidade de atributos contida na base de dados fornecida ao software minerador de dados para construção da DT. Isso se deve, a uma das características da DT, que é a redução de dimensionalidade devido ao índice que correlaciona os atributos críticos a segurança do sistema.

A tabela 5.1 apresenta o resultado de desempenho da DT referente à classificação da segurança. Para a situação avaliada, a DT tem um bom desempenho, próximo aos 92% de acerto para a classificação global (tanto para casos seguros, como para inseguros) com variância de +/- 2,62%. O percentual de precisão de classe segura tem percentual de 92,14%, isso significa que em 92,14% dos casos preditos como seguros eram realmente seguros.

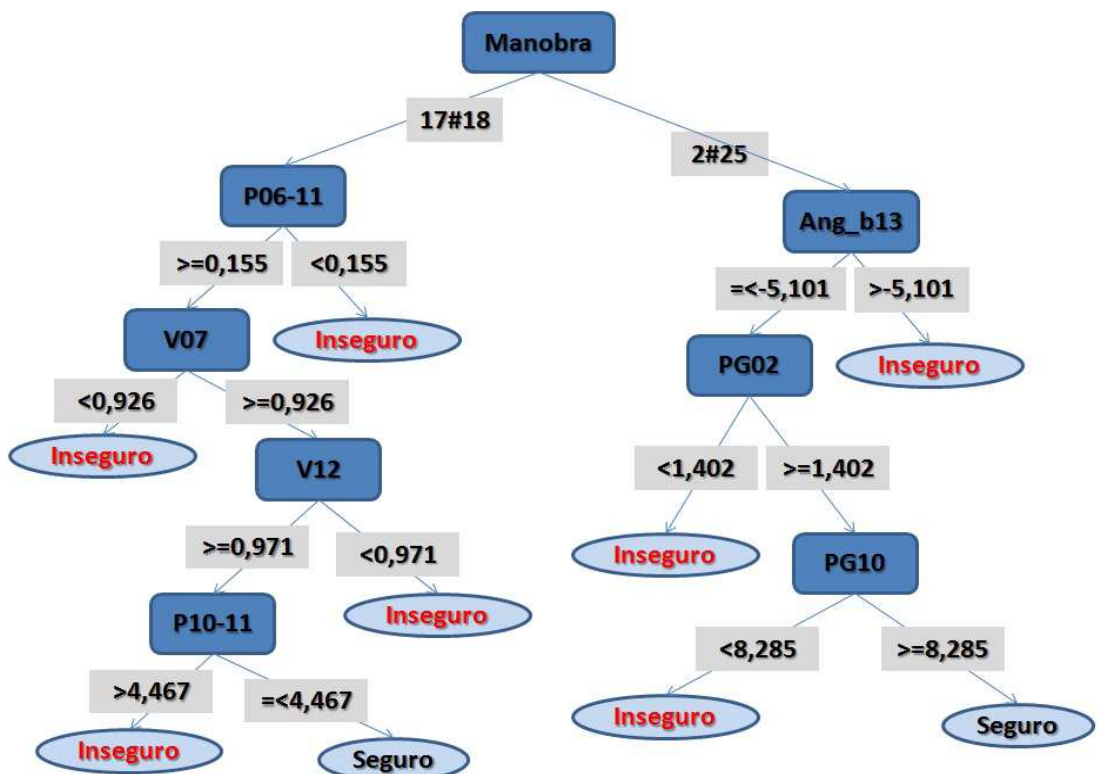


Figura 5.2 – Árvore de Decisão para DSA para um curto-circuito aplicado no Trecho 8-9, considerando as contingências em operação N-1, 17#18 e 2#25 .

Tabela 5.1 – Resultados de Classificação via Árvore de Decisão.

Acerto 92,03% +/- 2,62%	Inseguro Verdadeiro	Seguro Verdadeiro	Precisão de Classe
Predito Inseguro	346	32	91,53%
Predito Seguro	143	1676	92,14%
Sensibilidade	70,76%	98,12%	-----

Vale ressaltar que o conhecimento embutido nas regras da DT pode ser diretamente utilizado para tomada de ações preventivas, a fim de melhorar ou aumentar a margem da segurança dinâmica contra perturbações severas. Além disso, tais regras podem também auxiliar no planejamento da instalação de equipamentos de controle local de potências ativa e/ou reativa, como transformadores defasadores, banco de capacitores controlados, etc., e até mesmo em estratégias de controle centralizados utilizando sinais remotos.

5.2.1.2 - DSA utilizando RNA

Para aplicação da RNA MLP na DSA no contexto da estabilidade transitória, utilizou-se como exemplo a base de dados correspondente também a Área 1 do sistema New England. Os atributos para este estudo de caso da área 1 são dados por:

- Potências ativas e reativas nos geradores síncronos 03 e 02, e no barramento infinito (denominado neste trabalho de gerador 10);
- Tensões (módulo e ângulo de fase) nas barras 05, 06, 08, 09, 10, 11, 12, 13, 14, 31, 32 e 39;
- Fluxos de potência ativa e reativa nas linhas que estão dentro da Área 1.

Como a RNA trabalha, a priori, somente com dados numéricos, o atributo nominal que consta na base de dados usada para treinar a DT, referente às contingências em operação na condição N-1, foi eliminado da base de dados para treino da RNA. Esse atributo poderia ser inserido na base de dados na forma de alguma padronização numérica, o que resultaria no aumento da quantidade de dados de entrada, bem como do número de neurônios nas camadas intermediárias da RNA.

Por esse motivo, a base de dados fornecida ao software minerador de dados para treinamento da RNA MLP consiste apenas de atributos numéricos referentes à condição pré-falta do SEP em estudo com a contingência em operação 2#25 (linha desligada entre

as barras 2 e a 25). Com este procedimento, a base de dados utilizada para treino e teste da RNA foi reduzida para 118 casos. A RNA MLP gerada classifica o estado de operação com relação à estabilidade transitória, para o curto-circuito aplicado também nas extremidades da linha conectada entre as barras 8 e 9 (trecho 8-9).

Outro importante aspecto a ser considerado na entrada de dados para uma RNA é a normalização de dados entre 0 e 1, pois valores muito altos saturam rapidamente os neurônios, devido as suas funções de ativação serem do tipo sigmóide. Neste estudo de caso, a base de dados para treino da RNA foi tratada para transformar o atributo nominal (seguro ou inseguro) para um atributo numérico binominal (1 para seguro e 0 para inseguro).

A RNA treinada possui 46 entradas e 29 neurônios na camada oculta, assim como 2 neurônios na camada de saída. O resultado para a classificação da RNA MLP é apresentado na tabela 5.2.

Tabela 5.2 – Resultados da Avaliação via RNA.

Acerto 86,44% +/- 11,56%	Inseguro Verdadeiro	Seguro Verdadeiro	Precisão de Classe
Predito Inseguro	27	6	81,82%
Predito Seguro	10	74	88,10%
Sensibilidade	72,97%	92,50%	-----

A RNA MLP gerada para DSA, como foco na estabilidade transitória, apresenta um percentual de acerto baixo, no entanto, a taxa de variância de 11,56% acrescenta uma alta incerteza na classificação, isso é devido ao baixo número de casos na base de dados após o tratamento exigido para execução do treinamento da RNA.

5.2.2 – Projeto de Controle Preventivo via Sistemas Inteligentes

Quando um SEP está sujeito a uma perturbação crítica que pode levar a instabilidade transitória, ações de controle preventivo tornam-se indispensáveis no sentido de restaurar a segurança do SEP. Para melhorar a segurança dinâmica pela perspectiva da estabilidade transitória, ações de controle preventivo como o redespacho da geração ou chaveamentos de shunts podem ser implementadas, de modo que as consequências danosas das contingências críticas possam ser evitadas antes que ocorram.

Essas ações de controle preventivo normalmente dependem de um conjunto de medidas pré-estabelecidas e na experiência dos engenheiros e operadores. Entretanto,

essas análises tem um alto custo associado ao tempo, que pode ser crucial ao sistema nessas situações, levando-o a interrupção do fornecimento de energia.

Por esse motivo, o uso de ferramentas baseadas em sistemas inteligentes pode auxiliar nas tomadas de decisão em diversas situações, fornecendo regras (conjunto de passos a serem seguidos na operação) correlacionadas à segurança do SEP.

No que diz respeito ao projeto de controles preventivos, o uso de DTs e RNAs visa fornecer regras críticas que podem melhorar a segurança dinâmica do SEP, por meio da inclusão das restrições tratáveis, por exemplo, na execução de um fluxo de carga ótimo.

As seções 5.2.2.1 e 5.2.2.2 seguintes apresentam os modelos para avaliação e extração de regras críticas para projeto de controle preventivo baseadas em RNA do tipo MLP e DTs do tipo C4.5.

Para os estudos de casos de DSA, a base de dados consiste de atributos de todos os componentes do SEP em estudo. Para aplicação da RNA e DT no projeto de controles preventivos, a base de dados consiste de:

- Potências ativas de todos os geradores síncronos pertencentes ao SEP em estudo;
- Tensões (módulo) nas barras de geração pertencentes ao SEP em estudo.

Essas variáveis foram escolhidas por serem controláveis e com limites operacionais bem definidos.

As linhas de transmissão e transformadores possuem limites operacionais de carregamento baseados em seus limites térmicos. Esse limite térmico é bi-direcional, ou seja, uma linha conectada entre as barras A e B possui o mesmo limite de transferência em MVA, tanto no sentido de A para B quanto no de B para A. Contudo, as regras geradas pela máquina de aprendizado dependem dos dados de entrada para seu treinamento, que para essas situações podem não abranger de forma completa as possibilidades de direção de fluxo, fazendo com que as regras determinadas consistam somente de uma restrição de segurança para cada atributo.

Por exemplo, se a regra da DT indicar “fluxo de A para B menor que 80 MVA”, logo, é estabelecido um limite superior de 80 MVA com fluxo no sentido de A para B, porém quando o sentido do fluxo se modifica de B para A, não é conhecido o limite com relação a segurança dinâmica. Matematicamente, sabe-se que quando o fluxo de A para B é menor que 80 MVA, então o fluxo de A para B é maior que -80 MVA, ou seja, o fluxo de B para A também é menor que 80 MVA, porém essa restrição do fluxo de B

para A é válida somente para o limite térmico da linha, e não necessariamente válida do ponto de vista da estabilidade transitória.

5.2.2.1 – Projeto de Controle Preventivo Via RNA

Para aplicação da RNA MLP no projeto de controles preventivos contra instabilidades transitórias, foram treinadas 03 RNAs MLP para curto-circuito aplicados em 3 trechos do sistema New England. Os resultados de classificação do estado operacional com relação à estabilidade transitória apresentaram elevados percentuais de acerto.

Para estudo de caso em que foi aplicado um curto-circuito trifásico entre as barras 8 e 9 (trecho 8-9) na condição pré-falta com a linha 26#29 desligada, a RNA foi treinada e testada com uma base de dados consistindo de 68 casos com 20 atributos, formando uma RNA de 5 neurônios na camada oculta e 2 para camada de saída. A tabela 5.3 indica os valores em percentual das medidas de acerto da RNA, utilizando o particionamento *Cross Validation*.

Tabela 5.3 - Desempenho da RNA para o curto-circuito aplicado no trecho 8-9, considerando a contingência em operação 26#29

Acerto 100% +/- 0,00%	Inseguro Verdadeiro	Seguro Verdadeiro	Precisão de Classe
Predito Inseguro	32	0	100,00%
Predito Seguro	0	56	100,00%
Sensibilidade	100,00%	100,00%	-----

Os resultados da tabela 5.3 mostram que o desempenho da RNA foi excelente no que diz respeito a percentuais de acerto, precisão de classe e sensibilidade. Entretanto, esse percentual elevado de acerto se deu devido, provavelmente ao número reduzido de atributos na base de dados.

Contudo, o objetivo desta seção é explorar o potencial da RNA na utilização das conexões entre os neurônios para a extração de regras para o auxílio a ações de controle preventivo.

Tabela 5.4 - Pesos Sinápticos dos Neurônios da Camada de Saída da RNA para o curto-circuito aplicado no trecho 8-9, considerando a contingência em operação 26#29

Classe	'0' Inseguro		Classe	'1' Seguro
-----	-----		-----	-----
Peso1:	-5.046		Peso1:	5.039
Peso2:	-0.960		Peso2:	0.953
Peso3:	-2.600		Peso3:	2.606
Peso4:	5.937		Peso4:	-5.964
Peso5:	2.178		Peso5:	-2.142

Analisando o valor dos pesos que ligam a camada oculta aos neurônios da camada de saída da tabela 5.4 tem-se que o neurônio de saída que representa a classe 1 (estado seguro) possui 3 neurônios ativados positivamente, sendo que o peso no neurônio 1 tem a maior ativação se comparado aos demais pesos, isso o torna o neurônio mais ranqueado, ou seja, com maior relevância para com a saída. Com base nessa informação, buscam-se as variáveis que tem maior relevância na ativação do neurônio 1. A tabela 5.5 apresenta os valores dos pesos que conectam as entradas ao neurônio 1.

Com base nos dados da tabela 5.5 verifica-se que 7 entradas contribuem positivamente com a ativação no neurônio 1, sendo que as variáveis Pg07 (1,425), V34 (1,780) e V37 (2,038) possuem maiores contribuições. Essas variáveis podem ser interpretadas como variáveis de maior contribuição para ativar a saída de classe 1 (estado seguro), ou seja, alterando os valores dessas variáveis, pode se ativar a saída classe 1. Com base nessas informações é possível extrair regras que indicam ações de controle preventivo. Em forma de regra essa relação seria expressa como mostrado no quadro 5.1:

Tabela 5.5 - Valores dos Pesos que Conectam as Entradas ao Neurônio 1

Nó	1

Pg01:	-0.140
Pg02:	0.680
Pg03:	-1.591
Pg04:	-1.536
Pg05:	0.422
Pg06:	0.380
Pg07:	1.425
Pg08:	0.414
Pg09:	-1.203
Pg10:	-0.631
V30:	-1.053
V31:	-0.436
V32:	-1.029
V33:	-1.538
V34:	1.780
V35:	-0.233
V36:	-1.874
V37:	2.038
V38:	-1.186
V39:	-0.009

Se alterar a entrada PG07

 Ou alterar a entrada V34

 Ou alterar a entrada V37

Então Saída será classe 1

Quadro 5.1 – Conjunto de Regras extraídas da RNA para curto-circuito aplicado no trecho 8-9, considerando a contingência em operação 26#29.

Entretanto, como pode ser observado no quadro 5.1, as regras não fornecem indicações para valores, ou seja, não é conhecido o quão deve ser incrementado ou decrementado, por exemplo, o valor do PG07. Porém, isso pode ser realizado aumentando ou diminuindo esse valor por passos pequenos, manualmente ou de forma automática, e verificando a saída até que a mesma ative a classe desejada, tendo assim o valor normalizado para a entrada que foi alterada, e conseqüentemente extrair o valor na unidade de medida na qual a variável opera. Para o caso de PG07, essa unidade de medida pode ser fornecida em p.u. ou MW.

As regras críticas da RNA poderiam ainda contar com as variáveis de menor potencial de ativação, que nesse caso seria PG02, de acordo com a tabela 5.5, no entanto sua contribuição seria pequena. Como a regra de ajuste dada pela RNA baseia-se no operador lógico “ou”, a ação adotada pode afetar uma ou mais variável de entrada, dependendo da margem de regulação que a variável possua, ou seja, o quão próximo a variável está do seu limite operacional.

Para estudo de caso em que foi aplicado um curto-circuito trifásico entre as barras 6 e 11 (trecho 6-11) na condição pré-falta com a linha 16#17 desligada, a RNA foi treinada e testada com uma base de dados consistindo de 63 registros, com 20 atributos, formando uma RNA de 5 neurônios na camada oculta e 2 para camada de saída.

Os resultados da tabela 5.6 mostram percentuais de acerto, precisão de classe segura e sensibilidade em 95,24%, 97,83% e 93,75%, respectivamente. Com este excelente desempenho, as regras extraídas da RNA para projeto de controle preventivo contra a perturbação crítica no trecho 6-11, sob a contingência em operação 16#17, são mostradas no quadro 5.2 a seguir.

Tabela 5.6 – Desempenho da Rede Neural para o trecho 6-11.

Acerto 95,24% +/- 7,30%	Inseguro Verdadeiro	Seguro Verdadeiro	Precisão de Classe
Predito Inseguro	15	2	88,24%
Predito Seguro	1	45	97,83%
Sensibilidade	93,75%	95,74%	-----

Se alterar a entrada PG10

Ou alterar a entrada PG01

Ou alterar a entrada PG09

Então Saída será classe 1

Quadro 5.2 – Conjunto de Regras extraídas da RNA para o Trecho 6-11.

Para estudo de caso em que foi aplicado um curto-circuito trifásico entre as barras 9 e 39 (trecho 9-39) na condição pré-falta com a linha 13#14 desligada, a RNA foi treinada e testada com uma base de dados consistindo de 101 registros, com 20 atributos, formando uma RNA de 18 neurônios na camada oculta e 2 para camada de saída.

Pode-se notar na Tabela 5.7 que os resultados de desempenho da RNA treinada e testada são excelentes. Com este excelente desempenho, as regras extraídas da RNA para projeto de controle preventivo contra a perturbação crítica no trecho 9-39, sob a contingência em operação 13#14, são mostradas no quadro 5.3 a seguir.

Tabela 5.7 – Desempenho da Rede Neural para o trecho 9-39.

Acerto 99,00% +/- 3,00%	Inseguro Verdadeiro	Seguro Verdadeiro	Precisão de Classe
Predito Inseguro	16	1	94,12%
Predito Seguro	0	84	100,00%
Sensibilidade	84,21%	98,82%	-----

Se alterar a entrada PG10

 Ou alterar a entrada V32

 Ou alterar a entrada V35

Então Saída será classe 1

Quadro 5.3 – Conjunto de Regras extraídas da RNA para o Trecho 9-39.

No anexo A consta as conexões neurais das RNA treinadas que geraram as regras para auxílio à tomada de ações preventivas.

5.2.2.2 – Projeto de Controle Preventivo via DT

Para aplicação da DT no projeto de controle preventivo contra perturbações críticas que podem levar para uma instabilidade transitória, a base de dados utilizada pela DT para fins de DSA foi modificada. Além dos atributos referentes às potências ativas e os módulos das tensões nas barras de geração pertencentes ao todo SEP em estudo, também foi introduzido na base de dados à contingência em operação N-1.

Esse atributo nominal foi introduzido na base de dados pelo fato da DT C4.5 usar dados com atributos tanto numéricos como nominais (simbólicos), permitindo que seja realizada uma análise diferenciada quando comparada à realizada pela RNA.

Para aplicação da DT C4.5 no projeto de controles preventivos contra instabilidades transitórias, foram treinadas 03 DTs para perturbações (curto-circuito) aplicados em 3 trechos do sistema New England. Os estudos de casos do uso da DT C4.5 para projeto de controles preventivos foram realizadas, considerando aplicação de perturbações (curto-circuito) nos trechos 8-9, 6-11, 9-39, respectivamente.

Na tabela 5.8 são apresentados os resultados de desempenho satisfatórios da DT C4.5 para um curto-circuito aplicado no trecho 8-9 e na Figura 5.3 são ilustrados somente três ramos da DT construída, sucedidos pelo nó principal referente a manobra ou contingência em operação N-1.

Tabela 5.8 – Resultados de Classificação via Árvore de Decisão 8-9.

Acerto 95,31% +/- 2,24%	Inseguro Verdadeiro	Seguro Verdadeiro	Precisão de Classe
Predito Inseguro	405	19	95,52%
Predito Seguro	84	1689	95,26%
Sensibilidade	82,82%	98,89%	-----

Como podem ser observados na Figura 5.3, os três ramos de parte da DT criada consistem de regras facilmente interpretáveis, as quais descrevem a correlação entre os atributos críticos com relação à segurança, especificamente para um curto-circuito aplicado no trecho 8-9. O primeiro nó da DT diz respeito à contingência ou manobra em operação N-1. O ramo à esquerda da DT consiste das regras críticas quando a topologia do SEP em estudo está completa (0#0 indica que não há contingência em operação). No ramo central da DT é observado as regras quando a linha 16#21 (trecho entre as barras 16 e 21) está fora de operação. O ramo a direita da DT apresenta as regras quando a linha 26#29 (trecho entre as barras 26 e 29) está desligada. As setas em verde indicam o caminho da DT relativo ao estado seguro do ponto de vista da estabilidade transitória. Neste caso, as variáveis de controle devem ser ajustadas, por ordem hierárquica das regras, do 1º nó aos nós subsequentes até a folha (atributo alvo) da DT.

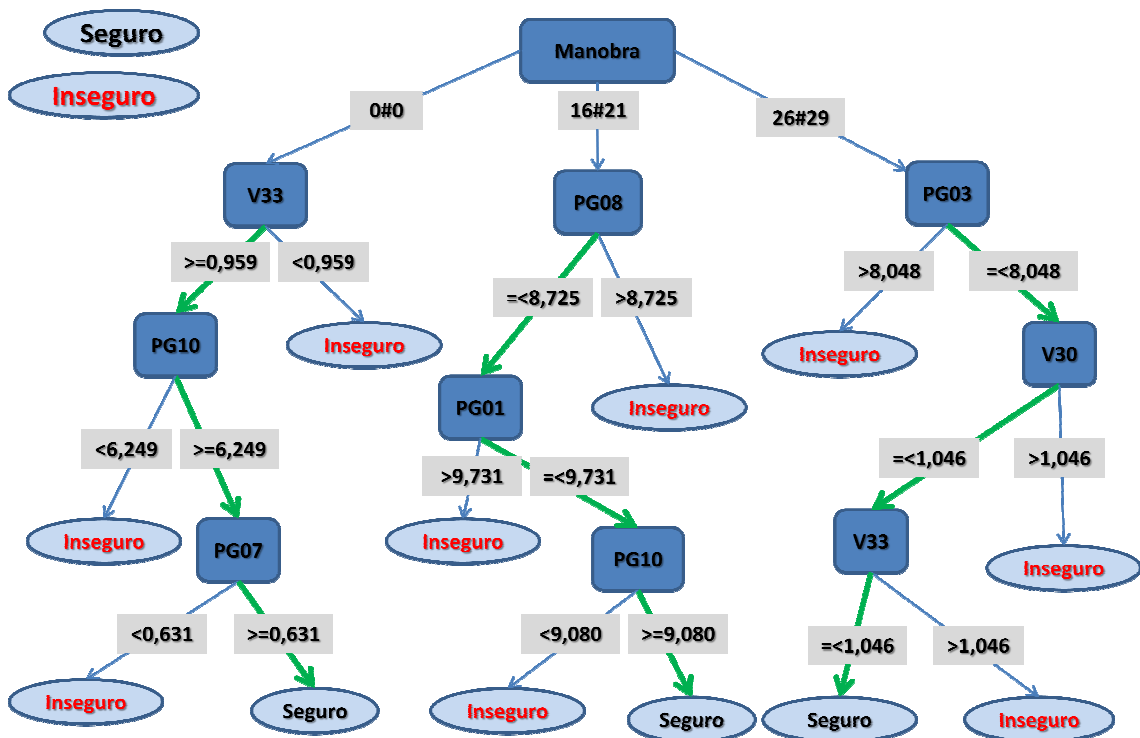


Figura 5.3 – Árvore de Decisão para Controle Preventivo Trecho 8-9.

O conjunto de regras gerado pela DT para estudo de caso é apresentado no quadro 5.4:

-
- Se (Manobra = 0#0) e (V33 > 0,959) e (PG10 > 6,249) e (PG07 > 0,361)
 Então Seguro
 Senão Inseguro
- Se (Manobra = 16#21) e (PG08 <= 8,725) e (PG10 <= 9,080) e (PG01 <= 9,731)
 Então Seguro
 Senão Inseguro
- Se (Manobra = 26#29) e (PG08 < 8,048) e (V30 <= 1,046) e (V33 <= 1,046)
 Então Seguro
 Senão Inseguro
-

Quadro 5.4 – Conjunto de Regras extraídas da Árvore de Decisão para o Trecho 8-9.

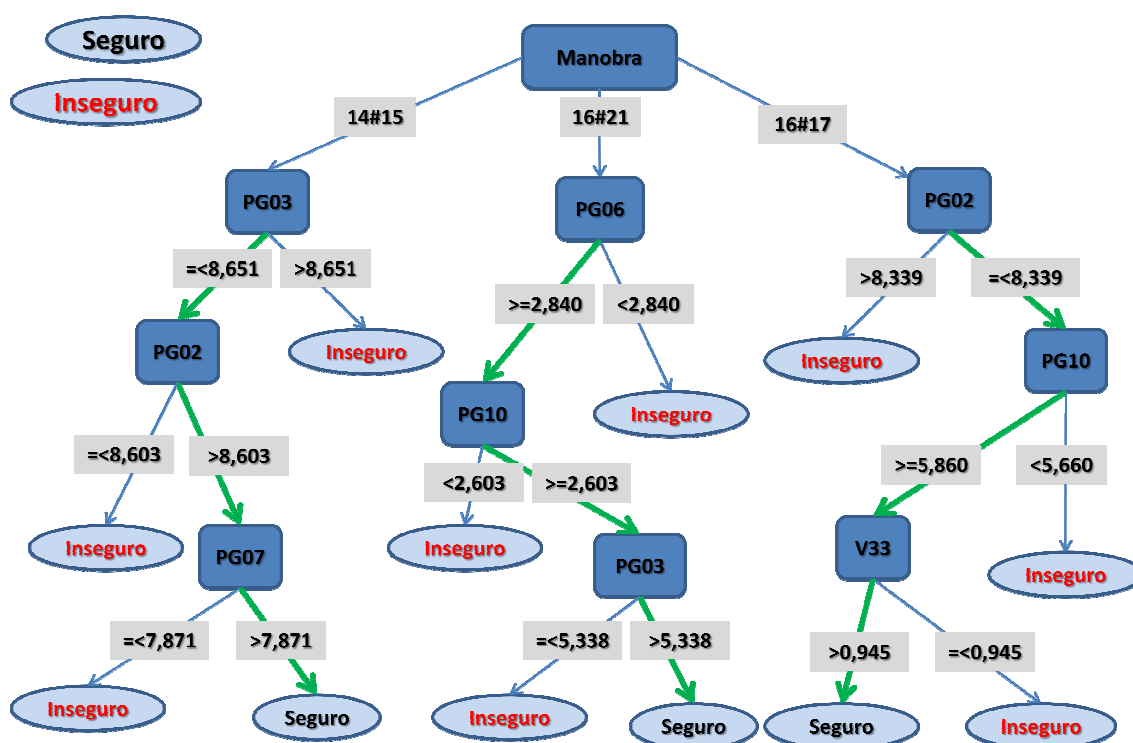
As regras geradas pela DT são do tipo Se-Então. Contudo, as condições obtidas são conectadas entre si pelo conector lógico “e”, significando que, para que o estado operacional seja predito como seguro, todas as condições devem ser verdadeiras, de acordo com a ordem hierárquica das regras, tornando a tomada de ações preventivas com pouca flexibilidade. Todavia, essa limitação é compensada pelo fato da restrição (ou regra) já possuir diretamente o valor numérico, por exemplo, $V33 > 0,959$, ou seja, a

tensão na barra 33 tem que ser maior que 0,959 p.u, como mostrado no 1º nó sucessor de atributo numérico do ramo a esquerda da DT ilustrada na Figura 5.3.

Na tabela 5.9 são apresentados os resultados satisfatórios de desempenho da DT C4.5 para um curto-circuito aplicado no trecho 6-11 e na Figura 5.4 são ilustrados somente três ramos da DT construída, sucedidos pelo nó principal referente a manobra ou contingência em operação N-1.

Tabela 5.9 – Resultados de Classificação via Árvore de Decisão 6-11.

Acerto 95,31% +/- 2,24%	Inseguro Verdadeiro	Seguro Verdadeiro	Precisão de Classe
Predito Inseguro	432	33	92,90%
Predito Seguro	70	1623	95,87%
Sensibilidade	86,06%	98,01%	-----



5.4 – Árvore de Decisão para Controle Preventivo Trecho 6-11.

A Figura 5.4 ilustra os três ramos escolhidos da DT completa criada, para um curto-circuito aplicado no trecho 6-11. Os ramos à esquerda, ao centro e a direita da DT consiste das regras críticas quando as linhas 14#15, 16#21 e 16#17 estão desligadas, respectivamente. As regras extraídas da DT C4.5 para este estudo de caso são dadas no quadro 5.4.

Se (Manobra = 14#15) e (PG03 =< 8,651) e (PG02 =< 8,603) e (PG03 =< 7,871)

Então Seguro

Senão Inseguro

Se (Manobra = 16#21) e (PG06 >= 2,840) e (PG10 >= 2,603) e (PG03 >5,338)

Então Seguro

Senão Inseguro

Se (Manobra = 16#17) e (PG02 =< 8,339) e (PG10 >= 5,860) e (V31 >= 0,945)

Então Seguro

Senão Inseguro

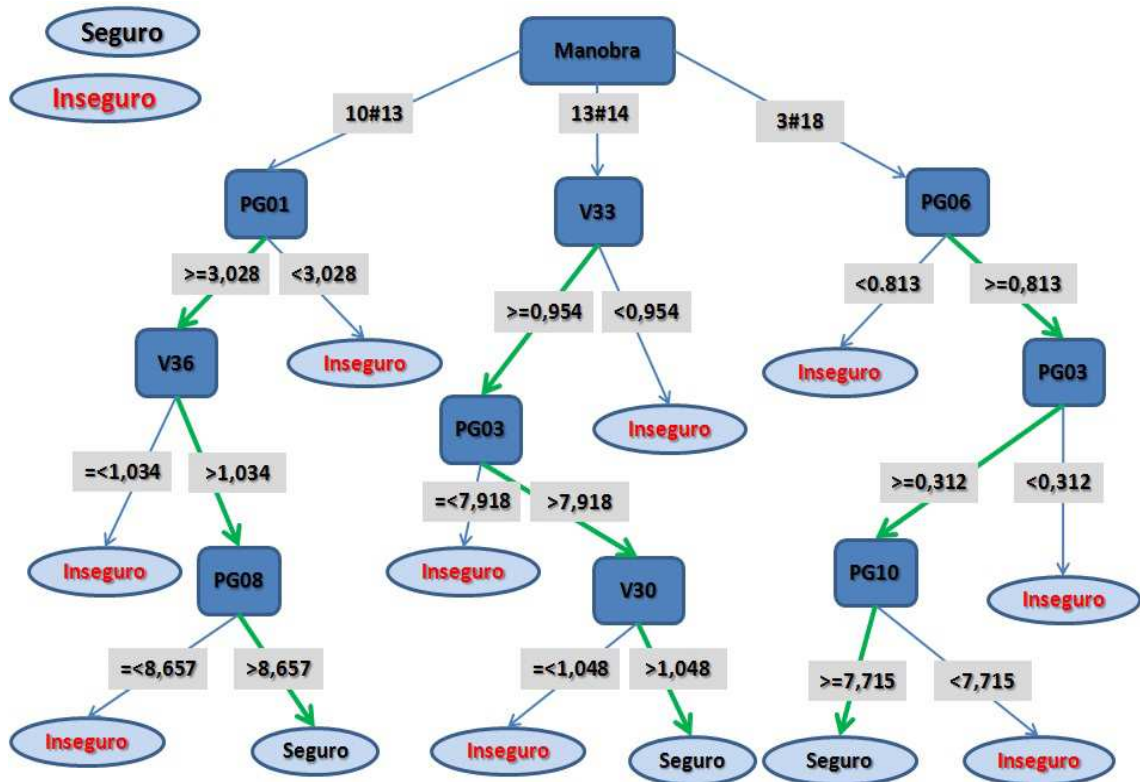
Quadro 5.5 – Conjunto de Regras extraídas da Árvore de Decisão para o Trecho 6-11.

Na tabela 5.10 são apresentados os resultados satisfatórios de desempenho da DT C4.5 para um curto-circuito aplicado no trecho 9-39 e na Figura 5.5 são ilustrados somente três ramos da DT construída, sucedidos pelo nó principal referente a manobra ou contingência em operação N-1.

Tabela 5.10 – Resultados de Classificação via Árvore de Decisão 9-39.

Acerto 96,14% +/- 1,30%	Inseguro Verdadeiro	Seguro Verdadeiro	Precisão de Classe
Predito Inseguro	315	17	94,88%
Predito Seguro	65	1726	96,37%
Sensibilidade	82,89%	99,02%	-----

A Figura 5.5 ilustra os três ramos escolhidos da DT completa criada, para um curto-circuito aplicado no trecho 9-39. Os ramos à esquerda, ao centro e a direita da DT consiste das regras críticas quando as linhas 10#13, 13#14 e 3#18 estão desligadas, respectivamente. As regras extraídas da DT C4.5 para este estudo de caso são dadas no quadro 5.5.



5.5 – Árvore de Decisão para Controle Preventivo Trecho 9-39.

Se (Manobra = 10#13) e (PG01 \geq 3,028) e (V36 \leq 1,034) e (PG08 \leq 8,657)

Então Seguro

Senão Inseguro

Se (Manobra = 13#14) e (V33 \geq 0,954) e (PG03 \leq 7,918) e (V30 \leq 1,048)

Então Seguro

Senão Inseguro

Se (Manobra = 3#18) e (PG06 \geq 0,813) e (PG03 \geq 0,312) e (PG10 \geq 7,715)

Então Seguro

Senão Inseguro

Quadro 5.6 – Conjunto de Regras extraídas da Árvore de Decisão para o Trecho 9-39.

É importante destacar que as regras geradas pelas DTs possuem na sua maioria, regras compostas por três variáveis, isso se deve ao fato da árvore de decisão já ser um redutor de atributos, ou seja, o algoritmo C4.5 seleciona os dados de acordo com o índice de seleção de atributos mostrado no capítulo 4.

Portanto, é possível montar nomogramas que permitem visualizar a região de segurança e seus contornos, a partir dos atributos críticos identificados pela DT. Um

nomograma pode indicar se o ponto de operação em tempo real está dentro de uma região segura ou não, no contexto da estabilidade transitória. Esse tipo de análise via nomogramas pode ser facilmente realizada quando as restrições das árvores possuem 2 ou 3 variáveis.

5.2.2.3 - Nomogramas determinados pelos Atributos Críticos

A figura 5.6 ilustra um nomograma de segurança criado a partir dos ramos da DT da figura 5.3, para um curto-circuito aplicado no trecho 8-9 com a contingência em operação #0#0. A DT calculou como variáveis críticas, por ordem: o módulo da tensão na barra 33, a potência ativa no gerador 07 e a potência ativa do barramento infinito (gerador 10).

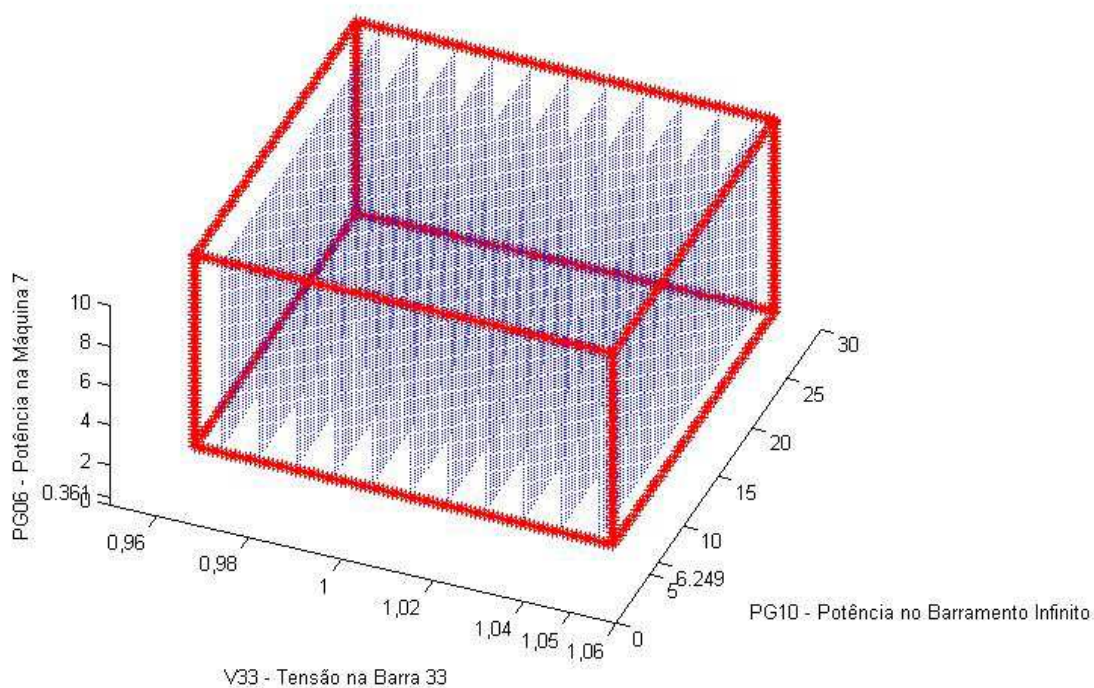


Figura 5.6 – Nomograma de segurança para um curto-circuito no trecho 8-9, para cenário com topologia completa do SEP em estudo.

Os pontos em azul no nomograma da figura 5.6 representam pontos de operação seguros baseados nas regras críticas da DT, enquanto que os pontos em vermelho delimitam a região de segurança. Dessa forma, é possível visualizar o ponto de operação e sua margem no que tange a segurança do ponto de vista da estabilidade transitória. A região de segurança na forma geométrica espacial de um paralelepípedo é formada tanto pelos limites estabelecidos pelas regras da DT, relacionadas à estabilidade transitória, como pelos limites de violação da segurança estática.

Por exemplo, a primeira regra indica que $V_{33} > 0,959$ p.u., isso implica em afirmar que essa restrição possui apenas o limite inferior com relação à estabilidade. No entanto, em tempo real, a tensão na barra 33 não pode violar seu limite superior (1,06 p.u.), tendo em vista que ações preventivas são realizadas quando o SEP ainda está em condição de operação normal do ponto de vista estático. Logo, o limite superior da V_{33} adotado para formação da região de segurança desse nomograma foi de 1,06. O mesmo critério foi adotado para definição dos limites superiores de PG10 e PG07, respectivamente. Ressalta-se que pelo fato da barra 39 (gerador 10) do sistema New England ser representada por uma barra infinita, PG10 pode, a priori, importar grandes blocos de potência ativa. Para este caso adotou-se um valor limite de PG10 em 30,0 p.u.

A figura 5.7 ilustra um nomograma de segurança criado a partir dos ramos da DT da figura 5.3, para um curto-circuito aplicado no trecho 8-9 com a contingência em operação 16#21. A DT calculou como variáveis críticas, por ordem: a potência ativa no gerador 08, a potência ativa da barra infinita (gerador 10) e a potência ativa do gerador 01.

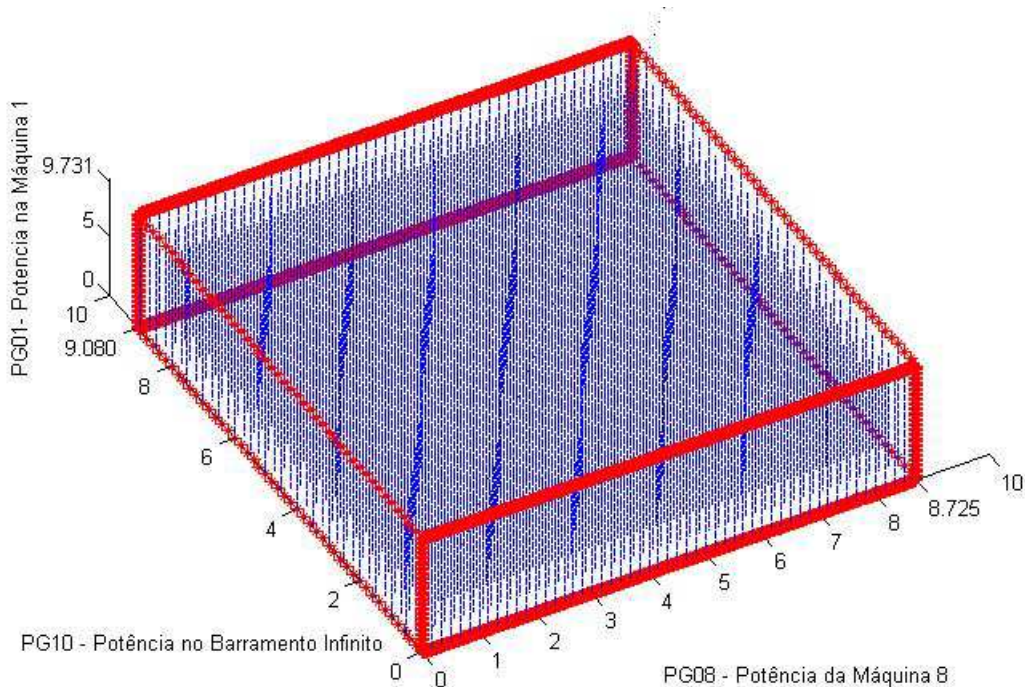


Figura 5.7 – Nomograma de segurança para um curto-circuito aplicado no trecho 8-9, com contingência operação 16#21.

Pode ser observado no nomograma da figura 5.7, que os geradores síncronos 08 e 01 podem operar bem próximo do seu limite de geração, enquanto que o barramento infinito apresenta limitação superior de 9,08 p.u. para transferência de potência ao SEP em estudo, isso é devido a contingência em operação 16#21, pois com a linha 16#21 desligada, o sistema torna-se menos robusto, reduzindo assim as margens de segurança.

5.2.2.3 - Comparativo de Desempenho das Técnicas (Precisão e Erro), Vantagens e Desvantagens.

Os resultados dos estudos de casos apresentados a DAS e ao projeto de controles preventivos via DT C4.5 e RNA MLP mostraram-se bastante promissores, devido aos seus altos desempenhos de classificação e de conhecimento embutido no modelo. Nas matrizes de confusão para cada exemplo, três índices de desempenho foram apresentados, são eles:

- A taxa de acerto;
- Precisão de Classe (seguro/inseguro);
- Sensibilidade (seguro/inseguro);

É importante ressaltar para fins de comparação de resultados, que neste caso, a bases de dados utilizadas pelas RNAs foram as mesmas utilizadas pelas DTs treinadas e testadas apresentados na tabelas.

A tabela 5.10 apresenta os resultados comparativos de classificação entre a RNA e DT, no que se refere à DSA para um curto-circuito aplicado no trecho 8-9, com a linha 2#25 fora de operação.

Tabela 5.10 – Comparação entre os índices de desempenhos da árvore decisão e da rede neural para DSA

Modelo	Taxa de Acerto	Precisão de Classe		Sensibilidade	
		Seguro	Inseguro	Seguro	Inseguro
Árvore de Decisão	92,03%	89,89%	100,00%	100%	72,72%
Rede Neural	86,44%	88,10%	81,82%	92,50%	72,97%

As taxas de acerto para o modelo baseado em DT têm percentuais de classificação com melhores desempenhos. Porém, isso não significa que a RNA possa ser descartada como um bom classificador para avaliação da segurança dinâmica, tendo

em vista que as máquinas de aprendizado dependem de eficientes bases dados de treino para obtenção de um bom desempenho.

A tabela 5.11 apresenta os resultados comparativos entre índices de desempenho da RNA e da DT para ações de controle preventivo contra uma instabilidade que possa ocorrer devido a um curto circuito no trecho 8-9, com a linha 26#29 fora de operação. Nota-se que os índices de desempenho dos classificadores têm valores iguais a 100%.

Tabela 5.11 - Comparação entre índices de desempenhos da RNA e da DT para projeto de controle preventivo contra um curto-circuito aplicado no trecho 8-9, sem a linha 26#29.

Modelo	Taxa de Acerto	Precisão de Classe		Sensibilidade	
		Seguro	Inseguro	Seguro	Inseguro
Árvore de Decisão	100,00%	100,00%	100,00%	100,00%	100,00%
Rede Neural	100,00%	100,00%	100,00%	100,00%	100,00%

As tabelas 5.12 e 5.13 apresentam os resultados comparativos entre índices de desempenho da RNA e da DT para ações de controle preventivo contra perturbações críticas que possam levar a uma instabilidade transitória, curto-circuito no trecho 6-11, com a linha 16#17 desligada e curto-circuito no trecho 9-39, com a linha 13#14 desligada, respectivamente. Pode-se observar nas tabelas que a diferença percentual entre as taxas de acertos dos classificadores é muito pequena, porém com uma pequena superioridade para a rede neural.

Tabela 5.12 - Comparação entre os índices de desempenhos da RNA e da DT para controle preventivo contra um curto-circuito no trecho 6-11, com a linha 16#17 desligada.

Modelo	Taxa de Acerto	Precisão de Classe		Sensibilidade	
		Seguro	Inseguro	Seguro	Inseguro
Árvore de Decisão	95,23%	94,00%	100,00%	100,00%	81,25%
Rede Neural	95,24%	97,83%	88,24%	95,74%	93,75%

Tabela 5.13 - Comparação entre os índices de desempenhos da RNA e da DT para controle preventivo contra um curto-circuito no trecho 9-39, com a linha 13#14 desligada.

Modelo	Taxa de Acerto	Precisão de Classe		Sensibilidade	
		Seguro	Inseguro	Seguro	Inseguro
Árvore de Decisão	98,02%	97,70%	100,00%	100,00%	87,50%
Rede Neural	99,00%	100,00%	94,12%	98,82%	84,21%

5.2.2.4 – Discussão

De acordo com resultados apresentados, ambas as técnicas apresentaram bons resultados de classificação, logo, seria praticamente irrelevante escolher um dos algoritmos baseado apenas nas suas taxas de acertos. Entretanto, outras características intrínsecas de cada classificador podem ser desejáveis em determinada aplicação.

Para o uso de um dos algoritmos na DSA e no auxílio ao controle preventivo, é preciso analisar algumas características. Primeiramente, qual a forma da entrada de dados para o modelo já treinado?

A RNA necessariamente trabalha com atributos numéricos, mais precisamente com valores dentro de uma faixa normalmente entre -1 e 1. Nessas aplicações específicas da presente dissertação, o intervalo está entre 0 e 1. Isso significa que os dados provindos do campo pelos sistemas de medição devem ser previamente tratados e normalizados para poderem alimentar o modelo.

Por outro lado, a DT pode receber tanto atributos numéricos como nominais (simbólicos), ou ainda ambos. No caso das aplicações em questão, há tanto atributos numéricos (potências, módulo das tensões, ângulos) e atributos simbólicos (trecho de curto circuito, linha fora de operação). A única exigência da árvore é manter a escala dos dados, por exemplo, se for treinada em p.u., o modelo gerado deve ser alimentado em p.u., caso seja treinada em outra unidade deve ser alimentada pela mesma, ou seja, não é obrigatória uma etapa de normalização dos dados de entrada em aplicação em tempo real.

Outra questão importante a se avaliar é o tamanho do modelo e a dimensionalidade dos dados, haja vista que em uma aplicação em tempo real o custo computacional do modelo pode ser relevante.

A rede neural trabalha com todos os dados apresentados a mesma na etapa de treino, ou seja, se a RNA foi treinada com 50 atributos em cada exemplo, a rede terá conexões para todos os 50 atributos. Contudo, algumas entradas ativam mais determinados neurônios do que outros. Baseado nesse princípio é que consiste a extração de regras para o controle preventivo via redes neurais. Para avaliação da segurança dinâmica, o uso de todas as variáveis tende a dar uma visão mais completa do sistema, isto é, uma mudança em qualquer variável é sentida pela RNA. Porém, essa característica pode tornar o modelo muito grande, muitas vezes até inviável para uma aplicação em tempo real.

As árvores de decisão são redutores naturais de dimensionalidade, pois o próprio algoritmo escolhe as variáveis que tem maior relação com o atributo alvo, devido ao índice utilizado para essa seleção. O número de atributos usados nos testes do modelo muitas vezes é bem reduzido quando comparado aos dados de entrada para treino. Essa característica de escolher apenas alguns atributos pode levar a não averiguação de variáveis importantes para o sistema. Entretanto, essa seleção de atributos críticos pode tornar a árvore de decisão uma importante ferramenta de planejamento, haja vista, que essas variáveis críticas encontradas por ela podem ser analisadas com maior cuidado.

Para o projeto de controle preventivo, a principal comparação que deve ser feita está no quesito da forma da regra associada. As redes neurais não geram regras claras e de fácil interpretação. Porém, é possível com uma análise detalhada dos pesos sinápticos e algumas considerações, extrair regras do tipo se-então, onde as variáveis das condições são testadas pelo conector lógico do tipo “ou”. Essa característica torna as regras extraídas da RNA flexíveis para operação, pois nem sempre todas as restrições impostas pelo modelo podem ser atendidas em determinadas condições operacionais do SEP. Contudo, a RNA não indica quanto deve ser o acréscimo ou o decréscimo em determinada variável, tornando a técnica relativamente complicada de se operar, exigindo um maior esforço computacional ou mesmo de um operador.

Na árvore de decisão, o controle preventivo torna-se mais fácil de ser executado, pois as regras são do tipo se-então, bem claras, diretas e interpretáveis. Contudo, as variáveis do teste condicional são conectadas pelo conector “e”, o que torna suas regras pouco flexíveis. Outra característica interessante da árvore de decisão é que suas regras já têm embutidos os limites para a tomada de decisão, ou seja, a própria regra indica se a variável tem que ser menor, maior ou igual a um determinado valor, fato que torna a árvore fácil de ser interpretada por qualquer pessoa.

Outra vantagem provinda das restrições da árvore de decisão é a criação de nomogramas, caso as regras de decisão sejam com duas ou três variáveis, de maneira simples é possível obter uma região de operação segura, a fim de visualizar o estado de segurança do ponto de operação.

Contudo, nada impede que ambas as ferramentas trabalhem em conjunto, de forma a explorarem melhor suas potencialidades.

Capítulo 6 - Conclusões

6.1 - Considerações Finais

Esta dissertação apresentou uma comparação entre as técnicas de aprendizado de máquina, árvores de decisão do tipo C4.5 e redes neurais MLP, para serem utilizadas como ferramentas de apoio a avaliação de segurança dinâmica e para auxílio ao controle preventivo, no contexto da estabilidade transitória. Com o objetivo de verificar a viabilidade de utilização de ambas as técnicas para solução desses problemas, modelos de DTs e RNAs foram treinados e testados, a partir da geração de base dados com diversos cenários de operação do sistema elétrico teste New England.

A partir dos testes realizados, algumas características importantes foram detectadas a respeito de cada uma das técnicas. A primeira característica importante diz respeito aos dados para treinamento das máquinas de aprendizado. A DT apresenta maior flexibilidade quanto ao tipo de dados, podendo utilizar tanto atributos numéricos como nominais (simbólicos), sendo que os numéricos não necessitam de normalização. Enquanto que a RNA utiliza apenas dados numéricos e normalizados entre 0 e 1, para realizar seu treinamento. Essa análise se estende ao modelo já induzido, onde as DTs podem receber dados provindos diretamente dos sistemas de medição, enquanto que para que sejam recebidos pelas RNAs, esse dados devem ser primeiramente normalizados.

Ainda com relação aos dados de treino, uma única DT poderia ter sido criada utilizando os 62 mil casos simulados, tendo em vista que o próprio algoritmo separa as contingências em operação N-1 em ramos da DT, como nós sucessores. Para as RNAs, uma possível forma de obtenção de uma única rede neural para treinar e avaliar os 62 mil casos seria realizar uma codificação binária nos dados, a fim de representar cada um dos atributos nominais (simbólicos). Isso tornaria muito mais complexo o modelo induzido, e conseqüentemente a extração de regras para auxílio ao controle preventivo.

Essas dificuldades que a RNA impõem com relação a seus dados de treino, acabaram induzindo a sucessivos tratamentos nas bases de dados. Esses tratamentos implicaram em uma redução no número de exemplos para treiná-las. Quanto ao desempenho de classificação, as medidas de precisão e erro, em ambas as técnicas apresentaram similaridades, não havendo grandes diferenças nos seus índices de desempenho.

No que tange a interpretabilidade dos resultados, as DTs são muito superiores as RNAs, já que o conhecimento adquirido é mostrado em um conjunto de regras do tipo se-então, ou seja, o conhecimento adquirido é repassado para o operador de forma simples e clara, onde os valores que a variável teste deve assumir são explicitados na própria regra. A forma como a rede neural guarda o conhecimento adquirido, exige um maior preparo do operador, pois suas regras são extraídas da matriz de pesos, a partir de alguma metodologia. Mesmo depois de extraída da RNA, essas regras só indicam quais variáveis podem ser alteradas, não indicando possíveis valores.

Contudo, as regras geradas pela RNA são mais flexíveis se comparadas às geradas pela DT, pois são baseadas no operador lógico “ou”, enquanto a regra da DT é baseada no operador lógico “e”, isso exige que todas as restrições impostas pela regra sejam atendidas seguindo uma ordem hierárquica.

Analisando os vários aspectos envolvidos na adoção de uma inteligente para fins de DSA e auxílio ao controle preventivo, do ponto de vista da estabilidade transitória, a DT leva vantagens em relação a RNA. Pois, de maneira geral, a simplicidade de suas regras e a forma intuitiva em que são apresentadas, aliadas ao bom desempenho de classificação e a flexibilidade quanto aos tipos e valores de dados, credenciam as DTs como uma boa opção de ferramenta para DSA em tempo real, planejamento e auxílio ao controle preventivo, no contexto da estabilidade transitória.

6.2 – Sugestões para Trabalhos Futuros

A partir dos resultados obtidos neste trabalho, é possível vislumbrar alguns possíveis avanços, tanto na metodologia, como no uso das máquinas de aprendizado para controle preventivo.

Tendo em vista a dificuldade de trabalhar com todas as possíveis contingências, do tipo N-1, N-2, N-3..., em um SEP real, a seleção de contingências severas pode ser encontrada a partir de uma metodologia baseada em análise de correlação, ou mesmo por meio de regras de associação de dados. A partir da seleção dessas contingências, máquinas de aprendizado podem ser treinadas para avaliar a segurança do SEP quando o mesmo operar nessas configurações topológicas. Outro trabalho interessante a ser realizado, é utilizar as regras geradas pela DT como restrições de segurança para alimentar um fluxo de carga ótimo.

Referências Bibliográficas

Kundur, 2004 “definition and classification of power system stability IEE/CIGRE joint task force on stability terms and definitions”

K. Morison, L. Wang and P. Kundur, “Power System Security Assessment”, IEEE Power & Energy Magazine, September/October, 2004.

Wehenkel, L. A., Automatic Learning Techniques in Power Systems. Norwell, MA: Kluwer, 1998.

Junior, Antonio carlos leal de castro, “Estudo de Controle Preventivo para Análise do Colapso de Tensão”, 2009.

Santa Rosa, Arthur da Silva, “influência da não representação de dispositivos de controle na análise de segurança de tensão”, 2007.

Assis, Tatiana Mariano Lessa de, “Cálculo da Capacidade de Transmissão Dinâmica em Sistemas de Potência através de Ferramentas Integradas e Sistemas Inteligentes”, 2007.

Mariotto, Lenois, “Avaliação da Segurança da Operação de Sistemas Elétricos de Potência Considerando os Limites de Estabilidade Angular e de Tensão”, 2008.

HERNANDEZ, EMÍLIO DEL MORAL, Inteligência Computacional e Redes Neurais em Engenharia Elétrica, USP, São Paulo, 2005.

MONARD, Maria Carolina; BARANAUKAS, José Augusto. Aplicações de Inteligência Artificial: Uma Visão Geral. São Carlos: Instituto de Ciências Matemáticas e de Computação de São Carlos, 2003.

Garcia, Simone Caboni, “O Uso de Árvores de Decisão na Descoberta de Conhecimento na Área da Saúde”, 2003

Freitas, A. Generic, Set-Oriented Primitives to Support Data-Parallel: Knowledge Discovery in Relational Database Systems. Thesis, UK: University of Essex, 1997

Power System Toolbox Webpage. Disponível em: <http://www.ecse.rpi.edu/pst/PST.html>. Acesso em: 30/01/2013

N. Amjady, "A framework of Reliability Assessment with consideration effect of transient and voltage stability," IEEE Trans. Power Systems, vol. 19, no. 2, pp. 1005–1014, 2004.

Theodoro, Edson Aparecido Rozas; Benedito, Raphael Augusto de Souza; Alberto; Luís Fernando Costa, “MÉTODOS RÁPIDOS PARA IDENTIFICAÇÃO DE

ILHAMENTO EM SISTEMAS ELÉTRICOS DE POTÊNCIA” - XVIII Congresso Brasileiro de Automática / 12 a 16-setembro-2010, Bonito-MS

Genc, R. Diao, V. Vittal, S. Kolluri, S. Mandal, "Decision Tree-Based Preventive and Corrective Control Applications for Dynamic Security Enhancement in Power Systems," IEEE Trans. Power Systems, vol. 25, no. 3, pp. 1611–1619, Aug. 2010.

L. A. Wehenkel, “Automatic Learning Techniques in Power Systems”. Norwell, MA: Kluwer, 1998.

Garcia, Simone Carboni, “O Uso de Árvores de Decisão na Descoberta de Conhecimento na Área da Saúde”, 2003

Rokach, Lior ; Maimon, Oded; “Data Mining with Decision Tree – Theory and Applications”, 2008

Mitchell, T. M., “Machine Learning”. McGraw Hill, 1997

HAYKIN, S., . “Redes Neurais princípios e prática”. Editora ARTMED, 2008.

S. Rovnyak, C. Taylor, and Y. Sheng, “Decision trees using apparent resistance to detect impending loss of synchronism,” IEEE Trans. Power Del., vol. 15, no. 4, pp. 1157–1162, Oct. 2000.

PEREIRA, A. L. S, 2009. “Utilização de Redes Neurais Artificiais no Ajuste de Controladores Suplementares e Dispositivo Facts STATCOM para a Melhoria da Estabilidade a Pequenas Perturbações do Sistema Elétrico de Potência”. Tese de Doutorado, Faculdade de Engenharia - UNESP, 2009.

VILLANUEVA, W. J. P, 2011. “Síntese Automática de Redes Neurais Artificiais com Conexões à Frente Arbitrárias”. Tese de Doutorado, Faculdade de Engenharia Elétrica e Computação, Campinas, São Paulo, 2011.

OLIVEIRA, A. R., 2005. “Redes Neurais Artificiais Aplicadas na Detecção, Classificação e Localização de Defeitos em Linhas de Transmissão”. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, UFJF, 2005.

FERRARI, A. C. K., 2011. “Controlador PID Sintonizado Por Redes Neurais Artificiais”. Trabalho de Conclusão de Curso, UFPR, 2011.

NETO, A. P., 2006. “Redes Neurais Artificiais Aplicadas às Avaliações Em Massa Estudo de Caso para a Cidade de Belo Horizonte/MG”. Dissertação de Mestrado, Programa de Pós-Graduação em Engenharia Elétrica, UFMG, 2006.

Barreto, J.M; Introdução às Redes Neurais, 2002.

Krishnan, Venkat; McCalley, James D., Progressive entropy based contingency grouping for deriving decision trees for multiple contingencies, 2012.

Krishnan, Venkat; McCalley ,James D.; Henry ,Sebastien; Issad, Samir Issad, “Efficient Database Generation for Decision Tree Based Power System Security Assessment”, 2011.

Gutierrez-Martinez, Victor J.; Cañizares , Claudio A.; Fuerte-Esquivel, Claudio R.; Pizano-Martinez, Alejandro; Gu, Xueping, “Neural-Network Security-Boundary Constrained Optimal Power Flow”, 2011.

Lin, Yu-Jen “Comparison of CART- and MLP-based power system transient stability preventive control”, 2013

Jensen, Craig A.; El-Sharkawi , Mohamed A., Marks, Robert J., “Power System Security Assessment Using Neural Networks: Feature Selection Using Fisher Discrimination”, 2001.

Karami, A. “Power system transient stability margin estimation using neural networks”, 2011.

Lin, Yu-Jen , “Prevention of transient instability employing rules based on back propagation based ANN for series compensation”, 2012

ANEXOS

ANEXO A

Tabelas de Pesos das RNAs

Pesos da RNA para controle trecho 6-11 manobra 16#17.

Hidden 1

Node	1
Pg01:	3.662
Pg02:	0.118
Pg03:	-1.804
Pg04:	1.119
Pg05:	-1.551
Pg06:	0.666
Pg07:	-0.567
Pg08:	-0.491
Pg09:	1.267
Pg10:	5.115
V30:	-2.203
V31:	-5.788
V32:	-0.043
V33:	0.996
V34:	0.595
V35:	0.596
V36:	1.896
V37:	-0.164
V38:	0.181
V39:	-0.943
Threshold:	2.445
Node	4
Pg01:	0.238
Pg02:	-0.224
Pg03:	-0.098
Pg04:	-0.289
Pg05:	0.070
Pg06:	-0.288
Pg07:	0.066
Pg08:	-0.028
Pg09:	-0.259
Pg10:	0.732
V30:	-0.663
V31:	-0.554
V32:	-0.542
V33:	-0.057

Node	2
Pg01:	0.826
Pg02:	-0.257
Pg03:	-0.374
Pg04:	-0.265
Pg05:	-0.138
Pg06:	-0.246
Pg07:	0.012
Pg08:	-0.193
Pg09:	-0.050
Pg10:	1.566
V30:	-1.055
V31:	-1.533
V32:	-0.699
V33:	0.070
V34:	0.141
V35:	0.396
V36:	0.945
V37:	0.011
V38:	0.272
V39:	-1.986
Threshold:	-0.358
Node	5
Pg01:	-3.652
Pg02:	-0.353
Pg03:	2.982
Pg04:	-1.166
Pg05:	1.153
Pg06:	-1.898
Pg07:	1.198
Pg08:	0.111
Pg09:	-1.505
Pg10:	-9.949
V30:	1.616
V31:	8.494
V32:	-0.974
V33:	-0.370

Node	3
Pg01:	0.941
Pg02:	-0.310
Pg03:	-0.406
Pg04:	-0.346
Pg05:	-0.117
Pg06:	-0.228
Pg07:	0.043
Pg08:	-0.179
Pg09:	-0.052
Pg10:	1.549
V30:	-1.097
V31:	-1.571
V32:	-0.751
V33:	0.037
V34:	0.201
V35:	0.453
V36:	1.108
V37:	0.076
V38:	0.259
V39:	-2.178
Threshold:	-0.433

V34:	0.068
V35:	0.324
V36:	0.622
V37:	0.063
V38:	0.239
V39:	-1.426
Threshold:	-0.557

V34:	0.120
V35:	-1.359
V36:	-1.851
V37:	0.091
V38:	-0.595
V39:	-1.509
Threshold:	-2.516

Output

Class	'1'	(Sigmoid)		Class	'0'	(Sigmoid)
Pesos	1:	6.726		Pesos	1:	-6.728
Pesos	2:	2.522		Pesos	2:	-2.527
Pesos	3:	2.759		Pesos	3:	-2.762
Pesos	4:	1.399		Pesos	4:	-1.388
Pesos	5:	-8.296		Pesos	5:	8.294

Threshold: 0.871

Threshold: -0.869

Pesos da RNA para controle trecho 8-9 manobra 26#19.

Hidden 1

Node	1		Node	2		Node	3
Pg01:	-0.140		Pg01:	0.070		Pg01:	-0.073
Pg02:	0.680		Pg02:	0.190		Pg02:	0.465
Pg03:	-1.591		Pg03:	-0.391		Pg03:	-0.913
Pg04:	-1.536		Pg04:	-0.312		Pg04:	-0.937
Pg05:	0.422		Pg05:	0.033		Pg05:	0.246
Pg06:	0.380		Pg06:	0.007		Pg06:	0.171
Pg07:	1.425		Pg07:	0.291		Pg07:	0.853
Pg08:	0.414		Pg08:	0.178		Pg08:	0.248
Pg09:	-1.203		Pg09:	-0.300		Pg09:	-0.737
Pg10:	-0.631		Pg10:	-0.058		Pg10:	-0.379
V30:	-1.053		V30:	-0.285		V30:	-0.651
V31:	-0.436		V31:	-0.148		V31:	-0.242
V32:	-1.029		V32:	-0.257		V32:	-0.671
V33:	-1.538		V33:	-0.318		V33:	-0.917
V34:	1.780		V34:	0.365		V34:	1.037
V35:	-0.233		V35:	-0.091		V35:	-0.123
V36:	-1.874		V36:	-0.469		V36:	-1.156
V37:	2.038		V37:	0.446		V37:	1.211
V38:	-1.186		V38:	-0.399		V38:	-0.716
V39:	-0.009		V39:	-0.070		V39:	-0.015
Threshold:	1.508		Threshold:	0.069		Threshold:	0.785
Node	4		Node	5			
Pg01:	0.148		Pg01:	0.078			

Pg02:	-0.800		Pg02:	-0.367
Pg03:	1.779		Pg03:	0.783
Pg04:	1.772		Pg04:	0.812
Pg05:	-0.496		Pg05:	-0.146
Pg06:	-0.498		Pg06:	-0.119
Pg07:	-1.704		Pg07:	-0.765
Pg08:	-0.543		Pg08:	-0.236
Pg09:	1.288		Pg09:	0.664
Pg10:	0.691		Pg10:	0.345
V30:	1.230		V30:	0.583
V31:	0.480		V31:	0.291
V32:	1.160		V32:	0.572
V33:	1.715		V33:	0.802
V34:	-1.965		V34:	-0.872
V35:	0.245		V35:	0.131
V36:	2.104		V36:	1.031
V37:	-2.261		V37:	-1.069
V38:	1.392		V38:	0.679
V39:	0.005		V39:	0.025
Threshold:	-1.779		Threshold:	-0.735

Classe	'0'	(Sigmoid)		Classe	'1'	(Sigmoid)
Peso	1:	-5.046		Peso	1:	5.039
Peso	2:	-0.960		Peso	2:	0.953
Peso	3:	-2.600		Peso	3:	2.606
Peso	4:	5.937		Peso	4:	-5.964
Peso	5:	2.178		Peso	5:	-2.142

ANEXOS B

Regras das árvores de decisão

Regras da Árvore Controle 06-11
Tree
Manobra = 0#0: seguro {inseguro=0, seguro=74}
Manobra = 1#2
Pg01 > 8.800: inseguro {inseguro=16, seguro=0}
Pg01 = 8.800
V31 > 0.912
V31 > 0.957: inseguro {inseguro=2, seguro=0}
V31 = 0.957: seguro {inseguro=0, seguro=27}
V31 = 0.912: seguro {inseguro=0, seguro=2}
Manobra = 1#39
V30 > 1.012: inseguro {inseguro=12, seguro=0}
V30 = 1.012
V30 > 0.852
Pg10 > 12.374
V31 > 0.942: inseguro {inseguro=5, seguro=0}
V31 = 0.942: seguro {inseguro=1, seguro=37}
Pg10 = 12.374: inseguro {inseguro=7, seguro=0}
V30 = 0.852: inseguro {inseguro=2, seguro=0}
Manobra = 10#11
Pg04 > 5.876: seguro {inseguro=0, seguro=26}
Pg04 = 5.876: inseguro {inseguro=10, seguro=0}
Manobra = 10#13
V34 > 0.973
V36 > 0.957: seguro {inseguro=6, seguro=50}
V36 = 0.957: inseguro {inseguro=2, seguro=0}
V34 = 0.973: inseguro {inseguro=17, seguro=1}
Manobra = 14#15
Pg03 > 8.651: inseguro {inseguro=14, seguro=0}
Pg03 = 8.651
Pg02 > 8.603: inseguro {inseguro=6, seguro=0}
Pg02 = 8.603
Pg03 > 7.871: inseguro {inseguro=2, seguro=0}


```

    | | | Pg03 = 7.871: seguro {inseguro=1,
seguro=34}
    Manobra = 15#16
    | V32 > 1.042: inseguro {inseguro=21, seguro=0}
    | V32 = 1.042
    | | Pg05 > 0.902
    | | | V31 > 0.937: inseguro {inseguro=8,
seguro=0}
    | | | V31 = 0.937: seguro {inseguro=2,
seguro=30}
    | | Pg05 = 0.902: seguro {inseguro=0, seguro=2}
    Manobra = 16#17
    | Pg02 > 8.339: inseguro {inseguro=5, seguro=0}
    | Pg02 = 8.339
    | | Pg10 > 5.860
    | | | V31 > 0.945: inseguro {inseguro=4,
seguro=0}
    | | | V31 = 0.945: seguro {inseguro=3,
seguro=47}
    | | Pg10 = 5.860: inseguro {inseguro=4,
seguro=0}
    Manobra = 16#21
    | Pg06 > 2.840
    | | Pg09 > 2.603
    | | | Pg03 > 5.338: inseguro {inseguro=6,
seguro=0}
    | | | Pg03 = 5.338: seguro {inseguro=5,
seguro=34}
    | | Pg09 = 2.603: seguro {inseguro=0, seguro=2}
    | Pg06 = 2.840: inseguro {inseguro=6, seguro=0}
    Manobra = 16#24
    | Pg01 > 9.145: inseguro {inseguro=11, seguro=1}
    | Pg01 = 9.145: seguro {inseguro=0, seguro=54}
    Manobra = 17#18
    | Pg07 > 2.692
    | | Pg02 > 1.090
    | | | V31 > 0.938
    | | | Pg01 > 5.214: seguro {inseguro=0,
seguro=2}
    | | | Pg01 = 5.214: inseguro {inseguro=14,
seguro=0}
    | | | V31 = 0.938: seguro {inseguro=4,
seguro=82}
    | | Pg02 = 1.090: seguro {inseguro=0, seguro=5}
    | Pg07 = 2.692: seguro {inseguro=0, seguro=4}

```

```

Manobra = 17#27
| V33 > 0.963
| | Pg09 > 1.204: seguro {inseguro=0, seguro=70}
| | Pg09 = 1.204: inseguro {inseguro=2,
seguro=0}
| V33 = 0.963: inseguro {inseguro=6, seguro=0}
Manobra = 2#25
| Pg02 > 2.244: seguro {inseguro=6, seguro=43}
| Pg02 = 2.244: inseguro {inseguro=17, seguro=0}
Manobra = 2#3: seguro {inseguro=0, seguro=23}
Manobra = 21#22
| Pg10 > 8.477
| | Pg03 > 0.290: seguro {inseguro=0, seguro=43}
| | Pg03 = 0.290: inseguro {inseguro=2,
seguro=0}
| Pg10 = 8.477: inseguro {inseguro=4, seguro=0}
Manobra = 22#23
| Pg02 > 6.693: inseguro {inseguro=15, seguro=0}
| Pg02 = 6.693
| | V31 > 0.944: inseguro {inseguro=9, seguro=0}
| | V31 = 0.944
| | | Pg07 > 8.609: inseguro {inseguro=2,
seguro=0}
| | | Pg07 = 8.609: seguro {inseguro=1,
seguro=63}
Manobra = 23#24
| Pg01 > 3.679
| | Pg01 > 8.501: inseguro {inseguro=7,
seguro=0}
| | Pg01 = 8.501: seguro {inseguro=0, seguro=44}
| Pg01 = 3.679: inseguro {inseguro=8, seguro=0}
Manobra = 25#26
| Pg07 > 1.700
| | Pg04 > 9.354: inseguro {inseguro=3,
seguro=0}
| | Pg04 = 9.354: seguro {inseguro=4, seguro=51}
| Pg07 = 1.700: inseguro {inseguro=6, seguro=0}
Manobra = 26#27
| V37 > 1.029: seguro {inseguro=0, seguro=30}
| V37 = 1.029
| | Pg02 > 2.668
| | | V32 > 0.975: inseguro {inseguro=39,
seguro=6}
| | | V32 = 0.975: seguro {inseguro=0,
seguro=2}

```

```

| | Pg02 = 2.668: seguro {inseguro=0, seguro=11}
Manobra = 26#28
| V37 > 0.960
| | Pg02 > 0.715
| | | Pg07 > 0.882
| | | | V34 > 0.977: seguro {inseguro=0,
seguro=78}
| | | | V34 = 0.977: inseguro {inseguro=9,
seguro=0}
| | | Pg07 = 0.882: inseguro {inseguro=6,
seguro=0}
| | Pg02 = 0.715: inseguro {inseguro=15,
seguro=0}
| V37 = 0.960: inseguro {inseguro=11, seguro=0}
Manobra = 26#29
| Pg07 > 8.477: inseguro {inseguro=5, seguro=0}
| Pg07 = 8.477
| | Pg01 > 2.240: seguro {inseguro=2, seguro=55}
| | Pg01 = 2.240: inseguro {inseguro=2,
seguro=0}
Manobra = 28#29
| Pg07 > 9.239: inseguro {inseguro=11, seguro=0}
| Pg07 = 9.239: seguro {inseguro=0, seguro=52}
Manobra = 3#18
| Pg10 > 6.439
| | V38 > 0.958
| | | Pg05 > 3.161: seguro {inseguro=0,
seguro=73}
| | | Pg05 = 3.161: inseguro {inseguro=5,
seguro=0}
| | V38 = 0.958: inseguro {inseguro=7, seguro=0}
| Pg10 = 6.439: seguro {inseguro=0, seguro=2}
Manobra = 3#4
| Pg02 > 8.098: inseguro {inseguro=15, seguro=0}
| Pg02 = 8.098: seguro {inseguro=0, seguro=42}
Manobra = 4#14
| Pg02 > 0.745
| | Pg01 > 2.262
| | | V31 > 0.918: seguro {inseguro=6,
seguro=45}
| | | V31 = 0.918: inseguro {inseguro=2,
seguro=0}
| | Pg01 = 2.262: inseguro {inseguro=2,
seguro=0}
| Pg02 = 0.745: inseguro {inseguro=4, seguro=0}

```

```

Manobra = 4#5
| V34 > 0.967
| | Pg04 > 1.005
| | | Pg08 > 1.747
| | | | Pg04 > 7.261: inseguro {inseguro=4,
seguro=0}
| | | | Pg04 = 7.261: seguro {inseguro=2,
seguro=63}
| | | | Pg08 = 1.747: seguro {inseguro=0,
seguro=2}
| | | Pg04 = 1.005: inseguro {inseguro=4,
seguro=0}
| V34 = 0.967: inseguro {inseguro=20, seguro=0}
Manobra = 5#8
| Pg07 > 9.353: inseguro {inseguro=5, seguro=0}
| Pg07 = 9.353: seguro {inseguro=0, seguro=78}
Manobra = 6#5
| Pg01 > 1.043
| | V31 > 0.961: inseguro {inseguro=5, seguro=0}
| | V31 = 0.961: seguro {inseguro=0, seguro=151}
| Pg01 = 1.043: seguro {inseguro=0, seguro=2}
Manobra = 6#7
| Pg01 > 1.812
| | V37 > 0.974: seguro {inseguro=0, seguro=50}
| | V37 = 0.974: inseguro {inseguro=7, seguro=0}
| Pg01 = 1.812: inseguro {inseguro=12, seguro=0}
Manobra = 7#8: seguro {inseguro=0, seguro=42}
Manobra = 8#9
| Pg01 > 1.742: seguro {inseguro=0, seguro=6}
| Pg01 = 1.742: inseguro {inseguro=4, seguro=0}
Manobra = 9#39
| V31 > 0.937: inseguro {inseguro=6, seguro=0}
| V31 = 0.937
| | Pg02 > 5.647: inseguro {inseguro=4,
seguro=0}
| | Pg02 = 5.647: seguro {inseguro=0, seguro=15}

```

Regras da Árvore Controle 08-09

dt cross 15

controle PG e V

Tree

Manobra = 0#0

| Pg08 > 0.299

| | Pg01 > 1.960

| | | V33 > 0.959

| | | | V35 > 0.953: seguro {inseguro=0, seguro=74}

| | | | V35 = 0.953: inseguro {inseguro=4, seguro=0}

| | | V33 = 0.959: inseguro {inseguro=7, seguro=0}

| | Pg01 = 1.960: seguro {inseguro=0, seguro=2}

| Pg08 = 0.299: inseguro {inseguro=5, seguro=0}

Manobra = 1#2

| Pg09 > 1.438: inseguro {inseguro=24, seguro=0}

| Pg09 = 1.438: seguro {inseguro=0, seguro=8}

Manobra = 1#39

| Pg05 > 7.667: seguro {inseguro=0, seguro=21}

| Pg05 = 7.667: inseguro {inseguro=38, seguro=0}

Manobra = 10#11

| Pg02 > 6.407: inseguro {inseguro=4, seguro=0}

| Pg02 = 6.407: seguro {inseguro=0, seguro=24}

Manobra = 10#13

| Pg04 > 2.056

| | Pg07 > 8.604: inseguro {inseguro=5, seguro=0}

| | Pg07 = 8.604: seguro {inseguro=0, seguro=87}

| Pg04 = 2.056: inseguro {inseguro=6, seguro=0}

Manobra = 13#14: seguro {inseguro=0, seguro=95}

Manobra = 14#15

| Pg10 > 8.264

| | Pg02 > 0.285

| | | Pg02 > 9.255: inseguro {inseguro=4, seguro=0}

| | | Pg02 = 9.255: seguro {inseguro=0, seguro=62}

| | Pg02 = 0.285: inseguro {inseguro=6, seguro=0}

| Pg10 = 8.264: inseguro {inseguro=7, seguro=0}

Manobra = 15#16

| Pg01 > 1.571: seguro {inseguro=1, seguro=27}

| Pg01 = 1.571: inseguro {inseguro=4, seguro=0}

Manobra = 16#17

| Pg01 > 2.984

| | Pg02 > 8.497: inseguro {inseguro=3, seguro=0}

| | Pg02 = 8.497: seguro {inseguro=0, seguro=37}

| Pg01 = 2.984

```

| | Pg01 > 1.438: inseguro {inseguro=17, seguro=0}
| | Pg01 = 1.438: seguro {inseguro=0, seguro=2}
Manobra = 16#21
| Pg01 > 0.491
| | Pg01 > 9.731: seguro {inseguro=0, seguro=2}
| | Pg01 = 9.731
| | | Pg10 > 9.080: seguro {inseguro=0, seguro=44}
| | | Pg10 = 9.080: inseguro {inseguro=4, seguro=0}
| Pg01 = 0.491: inseguro {inseguro=2, seguro=0}
Manobra = 16#24
| Pg02 > 7.580: inseguro {inseguro=21, seguro=0}
| Pg02 = 7.580: seguro {inseguro=2, seguro=70}
Manobra = 17#18
| Pg01 > 0.012
| | V35 > 0.952
| | | Pg06 > 8.665: inseguro {inseguro=2, seguro=0}
| | | Pg06 = 8.665: seguro {inseguro=2, seguro=39}
| | V35 = 0.952: inseguro {inseguro=4, seguro=0}
| Pg01 = 0.012: inseguro {inseguro=4, seguro=0}
Manobra = 17#27
| Pg02 > 0.850
| | Pg06 > 9.252: seguro {inseguro=0, seguro=2}
| | Pg06 = 9.252
| | | Pg09 > 1.858: seguro {inseguro=2, seguro=65}
| | | Pg09 = 1.858
| | | | V39 > 1.020
| | | | | Pg10 > 18.069
| | | | | | Pg10 > 20.484
| | | | | | | V31 > 0.936: inseguro {inseguro=2,
seguro=0}
| | | | | | | V31 = 0.936: seguro {inseguro=1,
seguro=2}
| | | | | | | Pg10 = 20.484: inseguro {inseguro=3,
seguro=0}
| | | | | | | Pg10 = 18.069: inseguro {inseguro=2,
seguro=0}
| | | | V39 = 1.020: inseguro {inseguro=4, seguro=0}
| Pg02 = 0.850: inseguro {inseguro=4, seguro=0}
Manobra = 2#25
| Pg01 > 9.484: inseguro {inseguro=4, seguro=0}
| Pg01 = 9.484
| | Pg10 > 8.285
| | | V37 > 1.045: inseguro {inseguro=3, seguro=0}
| | | V37 = 1.045: seguro {inseguro=13, seguro=80}
| | Pg10 = 8.285: inseguro {inseguro=17, seguro=0}

```

```

Manobra = 2#3
| V30 > 0.959: seguro {inseguro=0, seguro=17}
| V30 = 0.959: inseguro {inseguro=2, seguro=0}
Manobra = 21#22
| Pg02 > 4.648: inseguro {inseguro=10, seguro=0}
| Pg02 = 4.648
| | Pg06 > 2.069: seguro {inseguro=0, seguro=32}
| | Pg06 = 2.069: inseguro {inseguro=2, seguro=0}
Manobra = 22#23
| Pg02 > 8.438: inseguro {inseguro=14, seguro=0}
| Pg02 = 8.438
| | Pg10 > 5.415: seguro {inseguro=0, seguro=73}
| | Pg10 = 5.415: inseguro {inseguro=12, seguro=0}
Manobra = 23#24
| Pg01 > 3.823
| | Pg05 > 1.539: seguro {inseguro=2, seguro=34}
| | Pg05 = 1.539: inseguro {inseguro=2, seguro=0}
| Pg01 = 3.823: inseguro {inseguro=21, seguro=0}
Manobra = 25#26
| Pg03 > 0.670
| | V32 > 0.999
| | | Pg01 > 9.011: seguro {inseguro=0, seguro=4}
| | | Pg01 = 9.011
| | | | Pg09 > 2.712
| | | | | Pg07 > 3.749
| | | | | | Pg10 > 8.159: seguro {inseguro=6,
seguro=26}
| | | | | | | Pg10 = 8.159: inseguro {inseguro=10,
seguro=0}
| | | | | | Pg07 = 3.749: seguro {inseguro=0, seguro=5}
| | | | | Pg09 = 2.712: seguro {inseguro=0, seguro=2}
| | V32 = 0.999: seguro {inseguro=0, seguro=2}
| Pg03 = 0.670: seguro {inseguro=0, seguro=2}
Manobra = 26#27
| V37 > 0.973
| | Pg01 > 9.409: seguro {inseguro=0, seguro=2}
| | Pg01 = 9.409
| | | Pg08 > 7.438: inseguro {inseguro=15, seguro=0}
| | | Pg08 = 7.438: seguro {inseguro=0, seguro=66}
| V37 = 0.973: inseguro {inseguro=8, seguro=0}
Manobra = 26#28
| V30 > 0.954
| | V33 > 0.969
| | | Pg01 > 1.038

```

```

| | | | Pg01 > 1.717
| | | | | Pg02 > 9.082: seguro {inseguro=0, seguro=2}
| | | | | Pg02 = 9.082
| | | | | | | Pg03 > 7.578: seguro {inseguro=0,
seguro=2}
| | | | | | Pg03 = 7.578
| | | | | | | Pg01 > 9.512: seguro {inseguro=0,
seguro=4}
| | | | | | | Pg01 = 9.512
| | | | | | | | | V38 > 0.972: seguro {inseguro=0,
seguro=73}
| | | | | | | | | V38 = 0.972: inseguro {inseguro=6,
seguro=0}
| | | | Pg01 = 1.717: seguro {inseguro=0, seguro=2}
| | | Pg01 = 1.038: seguro {inseguro=0, seguro=2}
| | V33 = 0.969: inseguro {inseguro=2, seguro=0}
| V30 = 0.954: inseguro {inseguro=2, seguro=0}
Manobra = 26#29
| Pg03 > 8.048: inseguro {inseguro=21, seguro=0}
| Pg03 = 8.048
| | V30 > 1.046: inseguro {inseguro=9, seguro=0}
| | V30 = 1.046
| | | Pg10 > 21.826: inseguro {inseguro=2, seguro=0}
| | | Pg10 = 21.826: seguro {inseguro=0, seguro=56}
Manobra = 28#29
| V36 > 1.024
| | Pg01 > 7.078: inseguro {inseguro=11, seguro=0}
| | Pg01 = 7.078: seguro {inseguro=0, seguro=2}
| V36 = 1.024: seguro {inseguro=0, seguro=55}
Manobra = 3#18
| Pg10 > 3.780
| | Pg10 > 7.107: seguro {inseguro=4, seguro=103}
| | Pg10 = 7.107: inseguro {inseguro=7, seguro=0}
| Pg10 = 3.780: inseguro {inseguro=4, seguro=0}
Manobra = 3#4
| V35 > 0.979: seguro {inseguro=2, seguro=38}
| V35 = 0.979: inseguro {inseguro=20, seguro=0}
Manobra = 4#14
| Pg10 > 10.137
| | Pg01 > 8.898: inseguro {inseguro=4, seguro=0}
| | Pg01 = 8.898
| | | Pg09 > 8.465: inseguro {inseguro=2, seguro=0}
| | | Pg09 = 8.465: seguro {inseguro=0, seguro=44}
| Pg10 = 10.137
| | Pg02 > 4.510: seguro {inseguro=0, seguro=2}

```



```

| | Pg02 = 4.510: inseguro {inseguro=17, seguro=0}
Manobra = 4#5
| Pg01 > 0.878: seguro {inseguro=4, seguro=67}
| Pg01 = 0.878: inseguro {inseguro=9, seguro=0}
Manobra = 5#8
| Pg01 > 8.198: inseguro {inseguro=3, seguro=0}
| Pg01 = 8.198: seguro {inseguro=0, seguro=62}
Manobra = 6#11: seguro {inseguro=0, seguro=7}
Manobra = 6#5
| Pg02 > 0.951
| | V32 > 0.978: seguro {inseguro=6, seguro=101}
| | V32 = 0.978: inseguro {inseguro=2, seguro=0}
| Pg02 = 0.951: inseguro {inseguro=2, seguro=0}
Manobra = 6#7
| Pg02 > 0.895
| | Pg01 > 8.656: inseguro {inseguro=5, seguro=0}
| | Pg01 = 8.656
| | | Pg01 > 2.052: seguro {inseguro=0, seguro=49}
| | | Pg01 = 2.052: inseguro {inseguro=2, seguro=0}
| Pg02 = 0.895: inseguro {inseguro=6, seguro=0}
Manobra = 7#8
| V33 > 0.987: seguro {inseguro=0, seguro=29}
| V33 = 0.987: inseguro {inseguro=2, seguro=0}

```

Regras da Árvore Controle 09-39

```
Tree
Manobra = 0#0
| V34 > 0.965
| | Pg01 > 1.100
| | | Pg01 > 1.183
| | | | Pg08 > 0.922: seguro {seguro=87,
inseguro=2}
| | | | Pg08 = 0.922: inseguro {seguro=0,
inseguro=2}
| | | | Pg01 = 1.183: seguro {seguro=2,
inseguro=0}
| | | Pg01 = 1.100: inseguro {seguro=0,
inseguro=2}
| | V34 = 0.965: inseguro {seguro=0, inseguro=5}
Manobra = 1#2
| Pg05 > 7.784: seguro {seguro=32, inseguro=0}
| Pg05 = 7.784: inseguro {seguro=0, inseguro=28}
Manobra = 1#39
| Pg03 > 5.301: inseguro {seguro=0, inseguro=22}
| Pg03 = 5.301: seguro {seguro=10, inseguro=0}
Manobra = 10#11: seguro {seguro=14, inseguro=0}
Manobra = 10#13
| Pg01 > 3.028
| | V36 > 1.034: inseguro {seguro=0, inseguro=5}
| | V36 = 1.034
| | | Pg08 > 8.647: inseguro {seguro=0,
inseguro=2}
| | | Pg08 = 8.647: seguro {seguro=51,
inseguro=0}
| | Pg01 = 3.028: seguro {seguro=2, inseguro=0}
Manobra = 13#14
| V33 > 0.954
| | Pg03 > 7.918: inseguro {seguro=0,
inseguro=5}
| | Pg03 = 7.918
| | | V30 > 1.048: inseguro {seguro=0,
inseguro=2}
| | | V30 = 1.048: seguro {seguro=85,
inseguro=2}
| | V33 = 0.954: inseguro {seguro=0, inseguro=7}
Manobra = 14#15: seguro {seguro=87, inseguro=4}
Manobra = 15#16: seguro {seguro=59, inseguro=0}
Manobra = 16#17
| Pg06 > 7.608: inseguro {seguro=0, inseguro=10}
```

```

    | Pg06 = 7.608: seguro {seguro=20, inseguro=0}
  Manobra = 16#21
    | Pg02 > 8.299: inseguro {seguro=0, inseguro=4}
    | Pg02 = 8.299
    | | Pg02 > 0.245
    | | | Pg10 > 3.234: seguro {seguro=39,
inseguro=0}
    | | | Pg10 = 3.234: inseguro {seguro=0,
inseguro=2}
    | | Pg02 = 0.245: inseguro {seguro=0,
inseguro=2}
  Manobra = 16#24
    | V38 > 0.977
    | | Pg10 > 4.859
    | | | Pg02 > 2.128
    | | | | Pg06 > 1.218: seguro {seguro=34,
inseguro=0}
    | | | | Pg06 = 1.218: inseguro {seguro=0,
inseguro=2}
    | | | | Pg02 = 2.128: seguro {seguro=2,
inseguro=0}
    | | | Pg10 = 4.859: inseguro {seguro=0,
inseguro=4}
    | | V38 = 0.977
    | | | Pg01 > 3.192: inseguro {seguro=0,
inseguro=30}
    | | | | Pg01 = 3.192: seguro {seguro=2, inseguro=0}
  Manobra = 17#18
    | Pg07 > 7.654: inseguro {seguro=0, inseguro=10}
    | Pg07 = 7.654
    | | Pg10 > 4.522: seguro {seguro=51, inseguro=1}
    | | | Pg10 = 4.522: inseguro {seguro=0,
inseguro=6}
  Manobra = 17#27: seguro {seguro=65, inseguro=0}
  Manobra = 2#25
    | Pg01 > 0.447
    | | Pg02 > 8.657: inseguro {seguro=0,
inseguro=10}
    | | | Pg02 = 8.657: seguro {seguro=51, inseguro=6}
    | | Pg01 = 0.447: inseguro {seguro=0, inseguro=10}
  Manobra = 2#3
    | Pg02 > 6.984: seguro {seguro=4, inseguro=0}
    | Pg02 = 6.984
    | | Pg01 > 6.276: inseguro {seguro=0,
inseguro=2}

```

```

    | | Pg01 = 6.276
    | | | Pg01 > 2.643: inseguro {seguro=0,
inseguro=2}
    | | | Pg01 = 2.643
    | | | | Pg10 > 11.050: seguro {seguro=2,
inseguro=0}
    | | | | Pg10 = 11.050: inseguro {seguro=4,
inseguro=9}
    Manobra = 21#22
    | Pg01 > 4.153: seguro {seguro=37, inseguro=0}
    | Pg01 = 4.153: inseguro {seguro=0, inseguro=6}
    Manobra = 22#23: seguro {seguro=70, inseguro=0}
    Manobra = 23#24
    | Pg04 > 1.801
    | | V33 > 0.967: seguro {seguro=69, inseguro=0}
    | | V33 = 0.967: inseguro {seguro=0, inseguro=2}
    | Pg04 = 1.801: inseguro {seguro=0, inseguro=6}
    Manobra = 25#26: seguro {seguro=40, inseguro=0}
    Manobra = 26#27
    | Pg02 > 1.500
    | | Pg01 > 2.286
    | | | V33 > 0.977
    | | | | Pg10 > 7.484: seguro {seguro=76,
inseguro=0}
    | | | | Pg10 = 7.484: inseguro {seguro=0,
inseguro=4}
    | | | | V33 = 0.977: seguro {seguro=2,
inseguro=0}
    | | Pg01 = 2.286: seguro {seguro=2, inseguro=0}
    | Pg02 = 1.500: inseguro {seguro=0, inseguro=8}
    Manobra = 26#28
    | Pg08 > 9.667: inseguro {seguro=0, inseguro=8}
    | Pg08 = 9.667
    | | Pg01 > 8.963: inseguro {seguro=0,
inseguro=2}
    | | Pg01 = 8.963: seguro {seguro=84, inseguro=2}
    Manobra = 26#29
    | Pg01 > 9.205: inseguro {seguro=0, inseguro=6}
    | Pg01 = 9.205: seguro {seguro=71, inseguro=2}
    Manobra = 28#29
    | Pg07 > 9.732: inseguro {seguro=0, inseguro=19}
    | Pg07 = 9.732
    | | Pg02 > 7.417: inseguro {seguro=0,
inseguro=4}
    | | Pg02 = 7.417: seguro {seguro=69, inseguro=0}

```

```

    Manobra = 3#18
    | Pg06 > 0.813
    | | Pg03 > 0.312
    | | | Pg10 > 7.715: seguro {seguro=60,
inseguro=4}
    | | | Pg10 = 7.715: inseguro {seguro=0,
inseguro=4}
    | | Pg03 = 0.312: inseguro {seguro=0,
inseguro=2}
    | Pg06 = 0.813: inseguro {seguro=0, inseguro=4}
    Manobra = 3#4
    | Pg10 > 12.953: seguro {seguro=58, inseguro=0}
    | Pg10 = 12.953: inseguro {seguro=0, inseguro=4}
    Manobra = 4#14
    | V38 > 0.959
    | | Pg01 > 8.922: inseguro {seguro=0,
inseguro=6}
    | | Pg01 = 8.922: seguro {seguro=36, inseguro=0}
    | V38 = 0.959: inseguro {seguro=0, inseguro=10}
    Manobra = 4#5
    | Pg10 > 5.154
    | | V33 > 0.971: seguro {seguro=70, inseguro=10}
    | | V33 = 0.971: inseguro {seguro=0, inseguro=7}
    | Pg10 = 5.154: inseguro {seguro=0, inseguro=9}
    Manobra = 5#8
    | V30 > 1.040: inseguro {seguro=0, inseguro=10}
    | V30 = 1.040
    | | Pg06 > 1.890
    | | | V32 > 1.046: inseguro {seguro=0,
inseguro=6}
    | | | V32 = 1.046: seguro {seguro=67,
inseguro=0}
    | | Pg06 = 1.890: inseguro {seguro=0,
inseguro=4}
    Manobra = 6#11
    | Pg01 > 7.008: inseguro {seguro=0, inseguro=2}
    | Pg01 = 7.008: seguro {seguro=27, inseguro=0}
    Manobra = 6#5
    | Pg03 > 8.127: inseguro {seguro=0, inseguro=9}
    | Pg03 = 8.127
    | | Pg02 > 8.273: inseguro {seguro=1,
inseguro=16}
    | | Pg02 = 8.273
    | | | Pg04 > 0.623: seguro {seguro=112,
inseguro=0}

```

```
| | | Pg04 = 0.623: inseguro {seguro=0,
inseguro=2}
  Manobra = 6#7: seguro {seguro=47, inseguro=0}
  Manobra = 7#8
  | Pg06 > 9.939: inseguro {seguro=0, inseguro=2}
  | Pg06 = 9.939
  | | V36 > 0.956: seguro {seguro=42, inseguro=0}
  | | V36 = 0.956: inseguro {seguro=0, inseguro=2}
```