Ministério da
Ciência e Tecnologia

sid.inpe.br/mtc-m19/2011/04.15.19.12-TDI

UMA ESTRATÉGIA BASEADA EM ALGORITMOS DE MINERAÇÃO DE DADOS PARA VALIDAR PLANO DE OPERAÇÃO DE VOO A PARTIR DE PREDIÇÕES DE ESTADOS DOS SATÉLITES DO INPE

Primavera Botelho de Souza

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. José Demisio Simões da Silva (in memoriam), e Mauricio Gonçalves Vieira Ferreira, aprovada em 16 de maio de 2011.

URL do documento original:
<http://urlib.net/8JMKD3MGP7W/39GL532>

INPE
São José dos Campos
2011

PUBLICADO POR:

Instituto Nacional de Pesquisas Espaciais - INPE
Gabinete do Diretor (GB)
Serviço de Informação e Documentação (SID)
Caixa Postal 515 - CEP 12.245-970
São José dos Campos - SP - Brasil
Tel.:(012) 3208-6923/6921
Fax: (012) 3208-6919
E-mail: pubtc@sid.inpe.br

Ministério da
Ciência e Tecnologia

sid.inpe.br/mtc-m19/2011/04.15.19.12-TDI

UMA ESTRATÉGIA BASEADA EM ALGORITMOS DE MINERAÇÃO DE DADOS PARA VALIDAR PLANO DE OPERAÇÃO DE VOO A PARTIR DE PREDIÇÕES DE ESTADOS DOS SATÉLITES DO INPE

Primavera Botelho de Souza

Tese de Doutorado do Curso de Pós-Graduação em Computação Aplicada, orientada pelos Drs. José Demisio Simões da Silva (in memoriam), e Mauricio Gonçalves Vieira Ferreira, aprovada em 16 de maio de 2011.

URL do documento original:
<http://urlib.net/8JMKD3MGP7W/39GL532>

INPE
São José dos Campos
2011

Dados Internacionais de Catalogação na Publicação (CIP)

So86e

Souza, Primavera Botelho de.
Uma estratégia baseada em algoritmos de mineração de dados para validar plano de operação de voo a partir de predições de estados dos satélites do INPE / Primavera Botelho de Souza. – São José dos Campos : INPE, 2011.
xxii+149 p. ; (sid.inpe.br/mtc-m19/2011/04.15.19.12-TDI)

Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2011.
Orientadores : Drs. José Demisio Simões da Silva (in memoriam), e Mauricio Gonçalves Vieira Ferreira.

1. Inteligência artificial. 2. Mineração de dados. 3. Satélite artificial. I.Título.

CDU 004.82:829.783

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de Doutor(a) em

Computação Aplicada

Dr. Nandamudi Lankalapalli Vijaykumar

Presidente / INPE / SJCampos - SP

Dr. Mauricio Gonçalves Vieira Ferreira

Orientador(a) / INPE / SJCampos - SP

Dr. Edson Luiz França Senne

Membro da Banca / UNESP/GUARA / Guaratinguetá - SP

Dr. Edgar Toshiro Yano

Convidado(a) / ITA / São José dos Campos - SP

Dr. João Bosco Schumann Cunha

Convidado(a) / UNIFEI / Itajubá - MG

Este trabalho foi aprovado por:

(X) maioria simples

( ) unanimidade

Aluno (a): Primavera Botelho de Souza

São José dos Campos, 16 de maio de 2011

Dedico este trabalho
a meu pai Luiz Carlos Gil Botelho,
cujos ensinamentos fundamentam as minhas escolhas.

AGRADECIMENTOS

Ao orientador Dr. Mauricio Gonçalves Vieira Ferreira pela incansável dedicação.

Ao orientador Dr. José Demisio Simões da Silva pela significativa colaboração durante a elaboração do trabalho.

Ao Centro de Rastreio e Controle - CRC, em especial aos colegas de trabalho, Jun Tominaga e Odair Aparecido de Oliveira, que não mediram esforços em passar a experiência e o conhecimento adquiridos nas operações dos satélites do INPE.

Ao Dr. Hisao Takahashi e Dr. Delano Gobbi pelo apoio e pela confiança depositada que permitiram o desenvolvimento deste trabalho.

Ao meu companheiro Rubens, minhas filhas Luiza e Helena e minha mãe, Neyde, pelo apoio e incentivo, assim como pela compreensão e paciência durante os momentos em que não pude me dedicar a eles.

Aos amigos das Linhas de Pesquisa LUME e FISAT pelo imenso companheirismo.

Aos amigos de curso da Computação Aplicada sempre prestativos.

Aos colegas do Serviço de Pós-graduação sempre disponíveis e prestativos.

Ao Instituto Nacional de Pesquisas Espaciais pela oportunidade oferecida.

A todos os amigos que, direta ou indiretamente contribuíram para o enriquecimento deste trabalho.

RESUMO

A perspectiva de múltiplos lançamentos e o aumento da demanda de satélites em operação de acordo com o programa espacial do INPE, poderia tornar inviável uma análise criteriosa dos planos de voo que controlam satélites antes da execução real. Em função deste panorama, propõe-se uma solução para avançar na melhoria da segurança no planejamento das operações de rotina que controlam os satélites em órbita. Trata-se de uma estratégia para validar plano de operação de voo de satélites, tendo como parte relevante uma ferramenta de software concebida com o propósito de gerar diagnóstico e realizar predições de estados operacionais do satélite em função das ações contidas no plano. O objetivo da estratégia consiste em empregar os conceitos de mineração de dados na análise dos dados para prever estados do satélite, auxiliando os especialistas na validação dos planos de operação de voo, que podem ser gerados manualmente ou de forma automática por um planejador. E, a partir desta melhoria na segurança no planejamento das operações também garantir a integridade dos satélites em órbita. Assim, com base em um modelamento matemático contribuir na direção de oferecer ao INPE uma alternativa aos onerosos simuladores indicados pela literatura produzida pela comunidade espacial para a tarefa de realizarem predições de estados operacionais de satélites.

A STRATEGY BASED ON DATA MINING ALGORITHMS TO VALIDATE FLIGHT OPERATIONAL PLAN FROM PREDICTIONS FOR THE OPERATIONAL STATES OF INPE SATELLITES

ABSTRACT

The prospect of multiple launches and increased on demand from orbiting satellites in operation according to the INPE's satellite program, could become unviable to make a critical analysis of plans that control these satellites before real world implementation. Therefore, we propose a solution to improving safety in the planning of routine operations that control the satellites in orbit. This is a strategy for the validation of flight operation plan for satellites. As relevant part of the strategy to validate these plans, a software tool designed to generate diagnostic and predictions of satellite states, resulting of actions contained in the plan. The aim of the strategy is to employ the data mining concepts to analyze the data and predict the satellite operational states, assisting experts in evaluating the performance of the plan, being these plans manually or automatically generated by a planner. This solution to improving safety in the planning of operations also ensures the integrity of satellites in orbit. Hence, based on a mathematical model to contribute towards the INPE with an alternative to expensive simulators indicated in the literature produced by the space community to predict of satellite operational states.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

ALC - Alcântara
AXLOG Ingénierie - *Société d'ingénierie experte en informatique industrielle*
CBA - Cuiabá
CBERS - *China Brazil Earth Resource Satellites*
CCS - Centro de Controle de Satélites
SCDAv - Satélites de Coleta de Dados Avançados
CRC - Centro de Rastreio e Controle
EADS ASTRIUM - *European Aeronautic Defence and Space Company*
ESA - *European Space Agency*
ESOC - *European Space Operations Centre*
ESTEC - *European Space Research and Technology Centre*
GPM-Br - Satélite para Medida de Precipitação Global
INPE - Instituto Nacional de Pesquisas Espaciais
ISO - *International Organization for Standardization*
KDD - Descoberta de Conhecimento em Banco de Dados
LAAS-CNRS - *Laboratoire d'Architecture et d'Analyse des Systèmes*
LVQ - Aprendizagem por Quantização Vetorial
MAPSAR - Satélite de Multiaplicações
MECB - Missão Espacial Completa Brasileira
PlanIPOV - Planejamento Inteligente de Planos de Operação de Vôo
PMM - Plataforma Multimissão
POV - Plano de Operação de Vôo
PVP - Plano de Previsão de Passagem
RIPPER - Poda Incremental Repetida para Produzir Redução de Erro
RNA - Rede Neural Artificial
SCD - Satélite de Coleta de Dados
SICS - Sistema de Controle de Satélites
SPACEOPS - *International Conference on Space Operations*
SPAAS - *Software Product Assurance for Autonomy on-board Spacecraft technique et scientifique*
SMO - Otimização Sequencial Mínima
SVM - Máquina de Vetores de Suporte

LISTA DE SÍMBOLOS

P(Y) - Probabilidade anterior ou prévia de Y
P(X) - Evidência
P(Y|X) - Probabilidade posterior de Y
P(X|Y) - Probabilidade condicional de classe Y
X - Conjunto de atributos
Y - Variável de classe

SUMÁRIO

Pág.

1 INTRODUÇÃO

Atualmente uma das maiores preocupações da comunidade envolvida com atividades espaciais relaciona-se ao alto custo associado aos recursos para manutenção das operações das missões espaciais. Cada vez mais as atenções estão voltadas a encontrar soluções, que possibilitem reduzir o custo total das missões com maior confiabilidade das operações.

Existe por parte do programa espacial do INPE, um grande interesse relacionado à tarefa de controlar múltiplos satélites com soluções que possam aliar redução de custos e confiabilidade nas atividades de controle dos satélites em órbita. Esta realidade motivou o desenvolvimento de sistemas para automação das atividades de controle de satélites no INPE, com o propósito de atender a demanda crescente de lançamentos e o interesse de reduzir custos.

No entanto, dependendo da demanda de satélites em órbita, se tornaria impossível realizar antes da execução real, uma análise crítica dos planos de operação de voo gerados para controlar cada satélite. A solução apontada pela literatura produzida pela comunidade espacial para melhoria na segurança no planejamento das operações de satélites, faz menção ao uso de simuladores de satélite (BLANQUART et al., 2004; WERTZ e LARSON, 1999). Porém, o alto custo associado à aquisição ou desenvolvimento de simuladores de satélite no INPE, estimulou a busca por uma solução que permitisse unir eficiência e baixo custo.

Para alcançar esse avanço em segurança a custo reduzido, uma ferramenta de software baseada em análise matemática foi projetada para realizar predições de estados operacionais de satélite. A ferramenta constitui parte relevante de uma estratégia para validar plano de operação de voo.

Concebida como uma ferramenta de apoio à tomada de decisão no auxílio aos especialistas na avaliação das ações de um plano contendo as operações de voo visa garantir a saúde e consequentemente, a integridade do satélite. A ferramenta emprega técnicas de mineração de dados para realizar a predição de estados do satélite, representados por níveis de segurança em função do comportamento de um subsistema crítico de suporte do satélite: o subsistema de suprimento de energia, diretamente afetado pelas ações contidas em cada plano de operação de voo.

1.1 Motivação do Trabalho de Pesquisa

O Instituto Nacional de Pesquisas Espaciais (INPE) tem como um de seus principais objetivos o domínio de tecnologia espacial através do desenvolvimento e operação de satélites artificiais. Dada a extensão do território brasileiro, os satélites constituem uma importante ferramenta para várias aplicações de grande interesse para a população do país, como por exemplo: o monitoramento de vegetação, plantações, rios, clima; o levantamento de recursos naturais; e telecomunicações.

No momento, o INPE tem dois satélites em operação: os Satélites de Coleta de Dados SCD1 e SCD2, que foram integralmente desenvolvidos pelo INPE, como parte da chamada Missão Espacial Completa Brasileira (MECB).

Em linhas gerais, o Plano de Missões do INPE engloba as futuras missões da série de satélites de sensoriamento remoto desenvolvido em conjunto com a China dentro do programa *China Brazil Earth Resource Satellites* (CBERS): CBERS-3 e CBERS-4. Também inclui os satélites baseados na plataforma Multimissão (PMM): Amazônia-1 e 2; satélites científicos: Lattes1 e 2; satélite de Multiaplicações: *Multi-Application Purpose SAR* (MAPSAR) e satélite para medida de precipitação global: *Global Precipitation Measurement* (GPM-Br). Ainda estão previstos os lançamentos de mais dois satélites de coleta de dados

Avançados: SCDAv-1 e 2. A Figura 1.1 ilustra o plano de missões de satélites do INPE com previsão de futuros lançamentos até 2020.

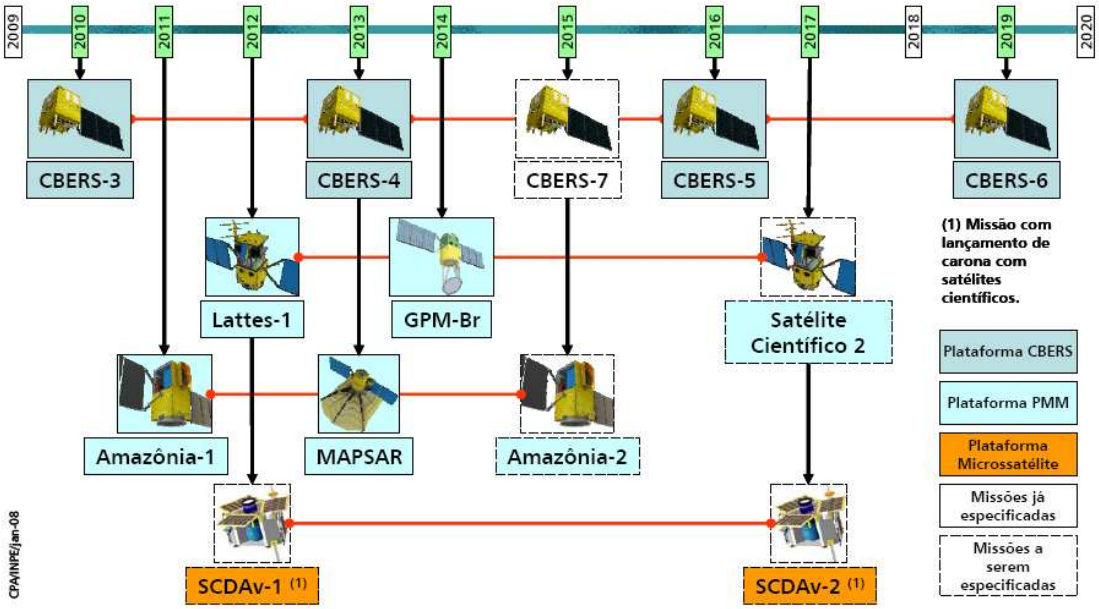

Figura 1.1 – Plano de missões 2008-2020

Fonte: INPE (2008).

Na estrutura organizacional formal do INPE, a entidade interna responsável pelas atividades de rastreio e controle em órbita de satélites é o Centro de Rastreio e Controle de Satélites (CRC), que é composto pelo Centro de Controle de Satélites (CCS), em São José dos Campos, SP e pelas estações de rastreio de Cuiabá (CBA), MT e de Alcântara (ALC), MA. Estes três locais estão interligados por uma rede privada de comunicação de dados, que permite a troca de informações entre estas unidades.

O contato do sistema de controle de solo com o satélite é estabelecido pelas estações de rastreio, quando este passa sobre a região de visibilidade de suas antenas. Durante períodos de visibilidade o sinal transmitido pelo satélite é captado pela antena da estação, ficando estabelecido um enlace descendente de comunicação. O sinal recebido do satélite contém as informações de telemetria que revelam seu estado atual de funcionamento. Após o estabelecimento do enlace descendente, a estação estabelece também um

enlace ascendente, que é utilizado para envio de telecomandos e execução de medidas de rastreio (medidas de distância e de velocidade).
Cada passagem de um dos satélites sobre uma estação de rastreio, corresponde em média a cada período orbital, o que significa, aproximadamente, uma passagem a cada 100 minutos. As previsões de passagens do satélite sobre as estações terrenas (KUGA e KONDAPALLI, 1993), também são utilizadas para a geração da programação de cada ação de controle a ser executada em cada passagem de satélite sobre estações de rastreio. Uma programação individual é gerada para cada satélite, sob a forma de um roteiro de operação de rotina denominado Plano de Operação de Voo, que abrange o período de uma semana. Os Planos de Operação de Voo são gerados por programas desenvolvidos dentro do próprio CCS, por especialistas em controle de satélites.

As operações de controle dos satélites são realizadas por equipes de operação bem treinadas, responsáveis pelas atividades em cada unidade da infra-estrutura de solo do INPE. Essas equipes executam fielmente as ações listadas no correspondente plano de operação em voo para a operação de cada um dos satélites controlados. Essas ações são executadas manualmente pelos operadores, a partir dos consoles de operação da sala de controle do CCS. A maior parte do trabalho desses operadores concentra-se na fase de operação de rotina dos satélites.

Esta previsão do aumento do número de satélites a serem controlados (Figura 1.1) e diminuição dos recursos financeiros destinados a operação de satélites no INPE, levou ao desenvolvimento de sistemas para automação das atividades de controle de satélites capaz de gerar os planos de operação de voo de forma automatizada por planejadores (CARDOSO, 2006) e assim, reduzir custos relacionados às equipes de operação.

No entanto, dependendo deste aumento da demanda de satélites em órbita a serem controlados, tornaria humanamente impossível realizar antes da

execução real, uma análise criteriosa dos planos de operação de voo para controlar cada satélite, sendo os planos gerados manualmente ou de forma automática.

Portanto, na busca de um caminho para aumentar o grau de confiabilidade, previsibilidade e segurança no planejamento das operações dos satélites, visando à manutenção da integridade dos satélites em órbita, surgiu à ideia de se obter diagnóstico e predições de estados do satélite.

Sem desconsiderar a realidade do INPE, em termos de redução de recursos financeiros para a operação de satélites, o sistema para realizar as predições de estados do satélite deveria então, contemplar uma solução de custo bem inferior aos simuladores de satélites atuais (BARRETO, 2010). Assim, o caminho da análise matemática dos dados obtidos durante a própria operação do satélite se apresentou como uma solução razoável.

1.2 Objetivos do Trabalho de Pesquisa

Esta pesquisa visa buscar um caminho para a melhoria em segurança no planejamento das operações de satélite do INPE. Propõe-se a criação de um sistema que permita prever os estados do satélite para fase operacional de rotina dos satélites controlados. Esta fase representa o período mais longo e importante para a missão a que se destina e que se confunde com a própria vida útil. Portanto, soluções para avançar na melhoria da segurança no planejamento das atividades operacionais de rotina desta fase são determinantes para o sucesso da missão.

O objetivo deste trabalho de pesquisa consiste em empregar os conceitos de mineração de dados na análise dos dados para prever estados do satélite, auxiliando os especialistas na validação dos planos de operação de voo, que podem ser gerados manualmente ou de forma automática por um planejador. E, a partir desta melhoria na segurança no planejamento das operações

também garantir a integridade dos satélites em órbita. Assim, com base em um modelamento matemático contribuir na direção de oferecer uma alternativa aos onerosos simuladores na tarefa de realizar predições de estados operacionais de satélites.

1.3 Metodologia de Desenvolvimento do Trabalho de Pesquisa

Na primeira etapa deste trabalho, a pesquisa se concentrou no levantamento da bibliografia existente sobre as soluções de segurança no planejamento das operações de missões espaciais que foram ou estão sendo adotadas pela comunidade espacial por aplicativos endereçados a área espacial, bem como o estado da arte na construção de modelos preditivos a partir de técnicas de mineração de dados, e também um levantamento da situação atual das atividades de operação de satélites no INPE.

Foram realizados também estudos relacionados a um aplicativo contendo o ambiente para análise do conhecimento (*Waikato Environment for Knowledge Analysis* - WEKA), dedicado a uma subárea da inteligência artificial, voltada ao estudo da aprendizagem de máquina. O aplicativo consiste em um conjunto de algoritmos implementados em linguagem Java, que representam diversas técnicas de mineração de dados, utilizadas no processo de construção de modelos preditivos a partir de uma base de dados supervisionada.

Em seguida foram desenvolvidas as etapas do processo de mineração de dados, em que a partir de uma base de dados classificados em níveis de segurança, definidos para um estudo de caso de um satélite virtual, cada algoritmo de classificação construiu um modelo de classificação.

A última etapa deste trabalho de pesquisa teve como foco a análise estatística dos modelos construídos, que faz uso de um conjunto de funções estatísticas para avaliação de classificadores e assim, determinar o modelo de classificação mais adequado para predição dos estados de um satélite.

Finalmente, o trabalho de pesquisa realizado e apresentado nesta tese, tem a sua estrutura detalhada a seguir.

1.4 Esboço Geral

Além deste capítulo introdutório, este trabalho contém outros seis capítulos, descritos a seguir:

CAPÍTULO 2 – PLANEJAMENTO DAS MISSÕES ESPACIAIS: neste capítulo são abordados conceitos de operação de uma missão espacial com ênfase no sistema de solo, as atividades para controlar satélites, o desenvolvimento de aplicações voltadas à automatização no planejamento das operações de sistemas espaciais de satélites do INPE, bem como caminhos apontados pela comunidade espacial para melhoria em segurança no planejamento das operações de satélites.

CAPÍTULO 3 – MINERAÇÃO DE DADOS: este capítulo apresenta os conceitos de mineração de dados, as fases ou passos de transformação dos dados, que compõem o processo de exploração e análise dos dados, as técnicas para detectar e descrever padrões estruturais de dados, as funções estatísticas para avaliação de classificadores e por fim, trabalhos correlatos na área espacial.

CAPÍTULO 4 – ESTRATÉGIA PARA VALIDAR PLANO DE OPERAÇÃO DE VOO: este capítulo apresenta uma visão geral da estratégia, incluindo a arquitetura concebida para validação de um plano de operação de voo. São também apresentados os parâmetros, telemetrias e limites operacionais, que compõem os estados operacionais do satélite em função da execução do plano. Por fim, a solução proposta: a ferramenta de validação, que se baseia em análise matemática para realizar a tarefa de predição dos estados do satélite a partir de estados simulados ou a partir de telemetrias recebidas pela estação terrena.

CAPÍTULO 5 – CONSTRUÇÃO DA BASE DE TREINAMENTO E EXECUÇÃO DOS ALGORITMOS CLASSIFICADORES: este capítulo apresenta a

construção da base de classificação supervisionada a partir da implementação do aplicativo desenvolvido para gerar o conjunto de dados de treinamento utilizado como estudo de caso. Apresenta ainda, a fase de mineração mostrando a abordagem para validação dos modelos utilizada pelos algoritmos classificadores, bem como apresenta os parâmetros de configuração para a execução dos algoritmos classificadores selecionados para a construção dos modelos de classificação preditiva

CAPÍTULO 6 – APRESENTAÇÃO E ANÁLISE DOS RESULTADOS OBTIDOS: neste capítulo são apresentados os modelos de classificação gerados pelos algoritmos classificadores dos estados do satélite XSAT, utilizado como estudo de caso e a análise estatística dos modelos. Por fim, como resultado da análise é indicado o modelo de classificação mais adequado para o desenvolvimento da ferramenta gerador de diagnóstico.

CAPÍTULO 7 – CONCLUSÃO: neste capítulo são apresentadas as conclusões e os trabalhos futuros a serem explorados.

2 PLANEJAMENTO DAS MISSÕES ESPACIAIS

Neste capítulo é apresentado o planejamento das missões espaciais, a definição de um sistema espacial, com ênfase no sistema de solo e as atividades para controlar satélites. São abordados conceitos de operação de uma missão espacial, o desenvolvimento de aplicações voltadas à automatização no planejamento das operações de sistemas espaciais de satélites do INPE, bem como caminhos apontados pela comunidade espacial, para melhoria em segurança no planejamento das operações de satélites.

2.1 Sistema Espacial

Uma missão espacial consiste no esforço para desenvolver e operar um sistema espacial com um objetivo específico (WERTZ e LARSON, 1999). A ênfase deste objetivo pode ser em telecomunicações, sensoriamento remoto, coleta de dados ou em experimentos científicos e tecnológicos (ECSS-E-70 A, 2000).

Um sistema espacial é desenvolvido para atender os requisitos de uma missão espacial, sendo dividido em dois segmentos: o segmento espacial e o segmento solo. O segmento espacial consiste na espaçonave em si, como por exemplo: satélite. A espaçonave, por sua vez é composta por plataforma e carga útil.

A plataforma de um satélite possui equipamentos que fornecem suporte à operação das cargas úteis, tais como: estrutura, energia, computador de bordo e comunicação de dados. Sendo a plataforma também responsável pelas operações de bordo relativas ao plano de operação de voo, ou seja, responsáveis pelas atividades a serem desempenhadas pelo computador de bordo central. Essas tarefas envolvem duas frentes: as operações de rotina e de contingência relacionadas ao monitoramento, operação e manutenção do

satélite em órbita, e pela comunicação com as estações terrenas de rastreio e com a carga útil.

A carga útil é a parte dedicada à aplicação, ou seja, ao motivo pelo qual o satélite foi lançado. É ela que gera os dados da missão. Em um satélite de sensoriamento remoto, por exemplo, a carga útil é composta por câmeras imageadoras e outros sensores. Em um satélite científico, a carga útil é formada por experimentos de natureza científica e tecnológica. A Figura 2.1 ilustra as atividades do segmento espacial e segmento solo, os usuários da missão e suas interações.

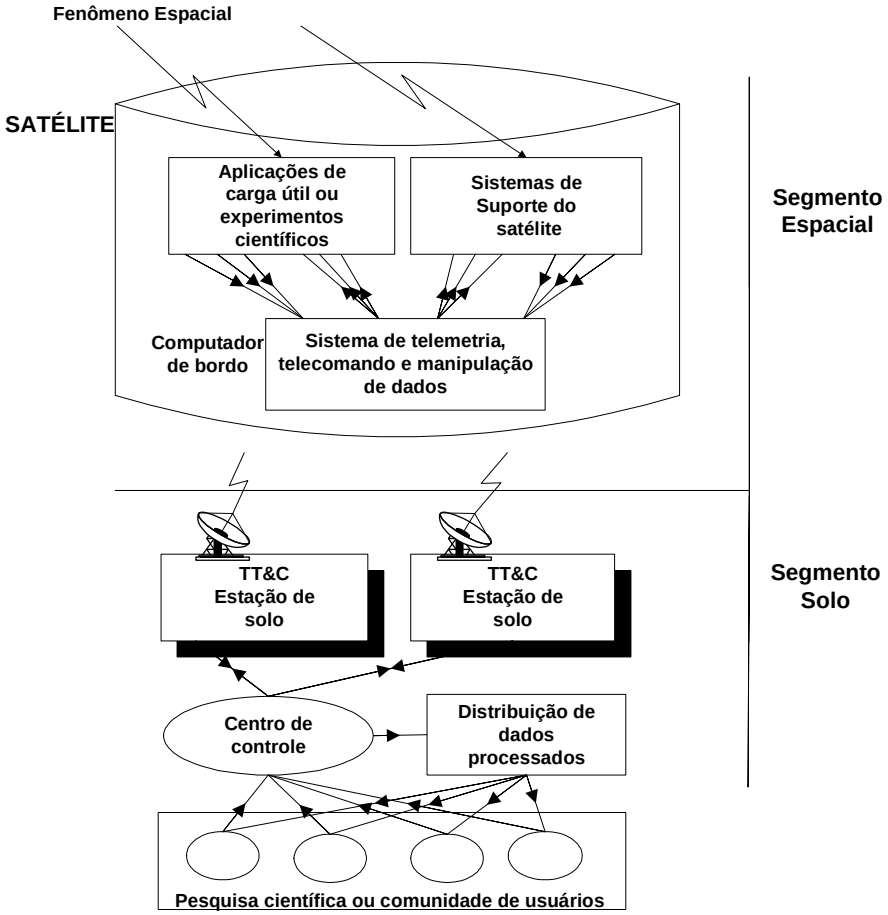

Figura 2.1 - Comunicação entre segmento solo e segmento espacial

Fonte: Souza (2002).

O monitoramento, o controle e a comunicação com o satélite são feitos a partir de sistemas de telemetria e de telecomandos. Um sistema de telemetria tem como propósito transportar, de forma transparente e confiável, informações de medição de uma origem de dados remotamente localizada para usuários distantes desta origem (WERTZ e LARSON, 1999). Um sistema de telecomandos deve transportar informações de controle de um ponto de origem como, por exemplo, um equipamento operado por humano para um equipamento remotamente localizado (WERTZ e LARSON, 1999). As unidades de telemetria e telecomandos são chamadas de pacotes, ou mensagens.

Os pacotes de telemetria são enviados do satélite para a estação terrena. Eles podem conter dados sobre o estado do satélite, como temperatura, níveis de tensão elétrica, situação de seus subsistemas (se estão ligados ou não, seus modos de operação e parâmetros de funcionamento atuais), ou dados de carga útil, como dados brutos de imagens, valores lidos por sensores de experimentos, etc. A telemetria carrega ainda relatos sobre o sucesso ou falha na execução de comandos, mensagens de erro e alertas sobre anomalias detectadas.

Uma vez em órbita, o satélite só pode receber comandos e transmitir dados durante o período em que existe visibilidade entre sua antena e a da estação terrena, ou seja, no período em que é possível a comunicação entre as antenas de solo e de bordo. Isto ocorre quando ele se encontra sobre a estação terrena. A este período de visibilidade e comunicação entre o satélite e a estação terrena é dado o nome de passagem. O tempo de passagem varia de uma órbita para outra, podendo mesmo não ocorrer em alguns casos.

O segmento solo é constituído pelos elementos de sistemas de solo, que são as estações terrenas e toda a infra-estrutura de hardware e de software, tais como: procedimentos envolvidos na preparação e execução das operações de uma missão, recursos humanos envolvidos nas operações de monitoramento e manutenção do satélite, bem como a disponibilização dos produtos do satélite

(imagens, dados científicos, etc.) aos seus usuários, conforme apresentado na Figura 2.2:

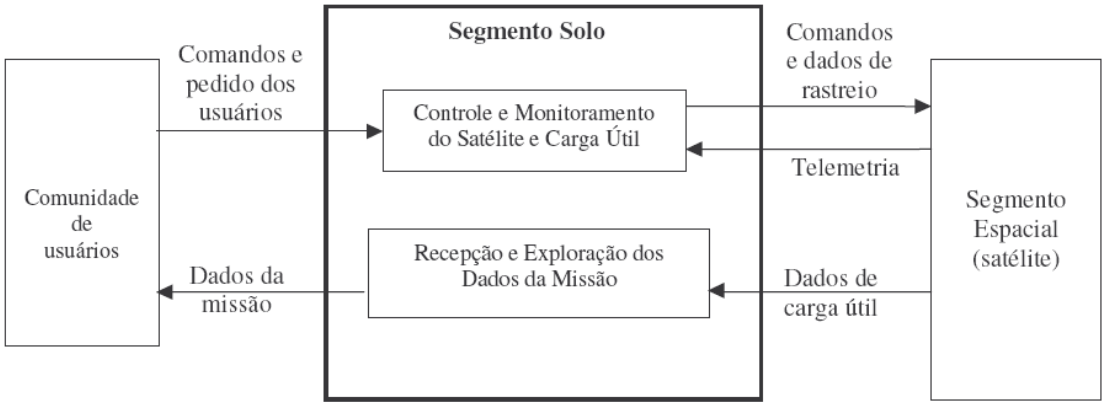

Figura 2.2 - Relacionamento entre o satélite, segmento solo e usuários
Fonte: Adaptada de Wertz e Larson (1999).

Os principais elementos do sistema de solo são:

Sistema de Controle de Missão: decomposto logicamente nos seguintes elementos funcionais: Sistema de Controle de Operações, Sistema de Controle de Carga Útil e Sistema de Exploração de Missão. Os dois primeiros são responsáveis por planejar, monitorar e controlar a plataforma e as cargas úteis, respectivamente. O último sistema é responsável por distribuir aos usuários os produtos da missão e dados extras necessários para a utilização dos dados de missão;

Sistema de Estação Terrena: constitui a interface direta com o satélite. Esse sistema realiza a conexão com o satélite durante os períodos em que este passa sobre a estação. Durante esses períodos são transmitidos comandos ao satélite (telecomandos) e recebidos dados da plataforma e das cargas úteis (telemetrias). O sistema de Estação Terrena é utilizado para o controle do satélite, provendo serviços de telemetria, telecomando e de rastreio tanto para a plataforma como para a carga útil. O sistema de Estação Terrena fornece ainda suporte a exploração da missão, provendo serviços de recepção dos

dados de carga útil tais como o recebimento de sinais de telecomunicações, de dados científicos e de imagens da terra;

Rede de comunicação de dados: responsável pela conexão entre todos os sistemas de solo permitindo a comunicação entre eles e a comunicação com os usuários da missão.

As atividades referentes ao segmento espacial e ao segmento solo das aplicações espaciais estão baseadas no planejamento de missão espacial, que consiste no desenvolvimento de um plano baseado na linha de tempo, para todas as atividades a serem executadas nas operações de bordo e de solo, que de forma conjunta compõe a operação de satélite (RABENAU et al., 2002).

2.2 Operação de Missões Espaciais

Para que uma missão espacial seja bem sucedida, torna-se necessário além da preparação física dos segmentos espacial e solo, o planejamento da operação da missão. Este planejamento deve definir como a operação deve ser realizada nas suas diferentes fases, para permitir que os objetivos da missão sejam alcançados da maneira mais eficiente possível.

A operação de uma missão espacial consiste em uma atividade sob responsabilidade do segmento solo, que tem como objetivo (ISO, 2004):

- Garantir a provisão da missão e dos serviços e produtos científicos;
- Executar as tarefas de operações de rotina;
- Recuperar as contingências de bordo;
- Gerenciar os recursos de bordo de forma a maximizar a vida útil da missão e as provisões de serviços e produtos.

A operação de uma missão pode ser classificada em dois grupos: operações de bordo e operações de solo. O primeiro grupo corresponde a todas as atividades relacionadas ao planejamento, execução e avaliação do controle do segmento espacial quando em órbita. Já as operações de solo correspondem a todas as atividades relacionadas ao planejamento, à execução e à avaliação do controle das facilidades de solo de suporte, tais como as estações terrenas e a rede de comunicação de dados de solo (ECSS-E-70 A, 2000).

O planejamento das operações de controle de missões espaciais envolve dois tipos de planos principais: o Plano de Operação de Voo e o Plano de Operação de Solo.

O plano de operação de voo tem como finalidade manter o satélite em órbita trabalhando de forma a atingir os objetivos da missão. Para isso, o plano contém todas as informações necessárias para o controle em órbita do satélite, tais como: procedimentos de controle de voo, procedimentos de recuperação de contingências, regras, planos e cronogramas.

Todas as atividades do plano de voo têm como ponto de partida a passagem do satélite sobre a estação terrena. A quantidade de tempo que um satélite permanece visível a uma determinada estação terrena é chamado de período de visada. E, define quais operações de voo devem ser realizadas durante uma passagem. As atividades típicas de uma passagem, apresentadas na Figura 2.3, se dividem em três períodos: pré-passagem, passagem e pós-passagem.

Os comandos de solo consistem nos telecomandos executados em tempo real durante a passagem de um satélite sobre uma estação terrena. Os comandos de bordo são executados fora de passagem e podem conter, por exemplo, comandos para atualizar softwares em bordo ou para realizar manobras. Existem arquivos de comandos de bordo, contendo o plano para carga útil. Neste caso, cada plano é formado por comandos voltados para a carga útil do satélite, ou seja, para a obtenção dos dados que constituem o próprio objetivo

da missão espacial. A elaboração de um plano para carga útil de satélites de sensoriamento remoto, por exemplo, obedece aos requisitos de captura de imagens dos vários usuários de uma missão espacial (ELDER e PAYNE, 2004).

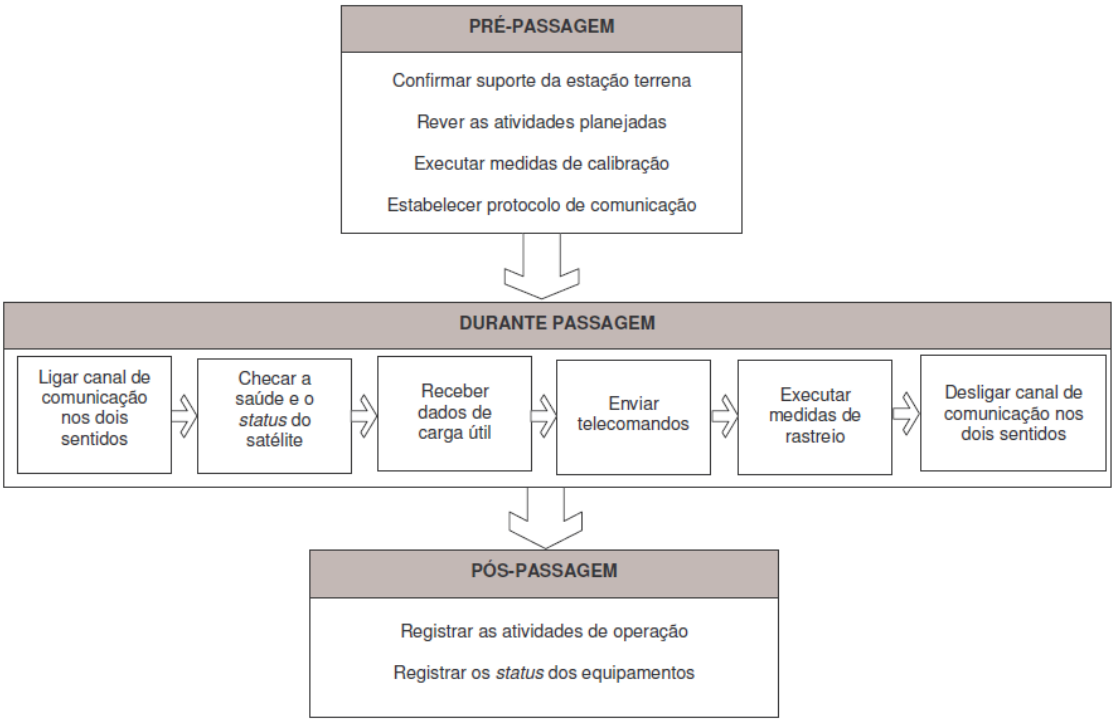

Figura 2.3 - Atividades típicas de uma passagem

Fonte: Cardoso (2006).

O plano de operação de solo, em contraste com o plano de operação de voo, é um plano que orienta a operação das facilidades de solo para oferecer suporte ao controle de satélites em órbita. Pode ainda, ser único para vários satélites que utilizam os mesmos recursos de solo, ou seja, que compartilham uma mesma estação terrena e a rede de comunicação de dados (ECSS-E-70 A, 2000). O plano de solo é constituído por planos que escalonam os satélites a serem rastreados por uma estação terrena e que configuram a estação terrena em relação a cada satélite. Essa configuração consiste em ações pré-passagem, realizadas antes do início de um rastreio, e pós-passagem, realizadas ao término do rastreio.

Existem duas equipes para controlar uma missão espacial e gerenciar as facilidades de solo: a equipe de operação e a equipe de suporte. A equipe de operação é constituída pelo grupo de controle de voo, grupo de dinâmica de voo, grupo de operações de solo e o grupo de exploração de missão. Enquanto a equipe de suporte é formada pelo grupo de suporte de solo e o grupo de suporte de projeto (ECSS-E-70 A, 2000). Estas equipes constituem recursos humanos responsáveis pela preparação e execução das várias tarefas de operações de uma missão. No ambiente de controle de missões espaciais, essas equipes devem ser suportadas ainda por uma infra-estrutura de hardware e de software a fim de contornar a complexidade do ambiente e também reduzir os custos associados à sua operação.

A operação da missão é dividida em duas fases operacionais principais. A fase considerada crítica, que cobre todas as operações necessárias durante a fase de lançamento e aquisição do satélite, o comissionamento em órbita e validação do segmento espacial. Após esta fase, se inicia a fase chamada de rotina ou nominal quando o segmento espacial está pronto para iniciar a exploração da missão.

Durante a fase operacional de rotina, um processo interativo composto das tarefas de preparação, execução e avaliação conduz a execução das operações contidas no plano de voo e no plano de solo. A tarefa de preparação envolve todas as atividades relativas ao planejamento das operações futuras e das atividades de manutenção, a produção dos cronogramas detalhados de operação baseados nos pedidos dos usuários e nos resultados da avaliação de desempenho dos segmentos espacial e solo. A tarefa de execução cobre a implementação das operações contidas nos planos e a execução da monitoração do estado dos elementos dos segmentos espacial e solo, incluindo a execução de ações corretivas necessárias no caso de anomalias. A tarefa de avaliação compreende a avaliação dos segmentos espacial e solo incluindo as análises de tendências que suportam a preparação de operações futuras e atividades de manutenção preventiva.

Portanto, em função do contingente humano envolvido no controle de missões espaciais, principalmente durante a fase operacional de rotina, existe um grande interesse por parte da comunidade da área espacial em reduzir custos relacionados às equipes de operação, a partir de sistemas que automatizem o processo de planejamento e de controle para várias missões.

2.3 Operação de Missões Espaciais no INPE

No INPE, a diminuição dos recursos financeiros destinados a operação de satélites e também a previsão do aumento do número de satélites a serem controlados (vide seção 1.1), motivou o desenvolvimento de sistemas para automação das atividades de controle de satélites voltados à fase operacional de rotina.

No Centro de Rastreio e Controle de Satélites (CRC) do INPE. A fase operacional de rotina de um satélite inclui a geração do plano de previsão de passagem (PVP) e do plano de operação de voo (POV) para cada satélite. Os planos indicam todas as ações de operação que o operador do satélite deve executar antes, durante e após cada passagem do satélite.

Os operadores controlam e monitoram o satélite executando as atividades contidas no plano de operação de voo. Para executar estas atividades, os operadores utilizam o Sistema de Controle de Satélites (SICS), que é o software aplicativo de tempo real do Centro de Controle de Satélites (CCS), composto dos subsistemas de telemetria, telecomandos, medidas de distância e medidas de velocidades.

Durante a passagem de um satélite na região de visibilidade de uma estação terrena, o SICS comunica-se em tempo real com o satélite, utilizando a rede privada de comunicação de dados que liga o CCS com as estações, enviando telecomandos, recebendo telemetria e realizando medidas de distância e

velocidade. As funções do SICS, correspondentes às atividades contidas no plano de operação de voo, são acionadas manualmente pelos operadores nos respectivos instantes previstos para as mesmas. Após a execução das atividades, os operadores registram no plano, os resultados obtidos.

O cenário de operação típica dos satélites controlados pelo INPE durante uma passagem sobre uma estação terrena, inclui as seguintes ações:

- Execução de medidas de calibração do equipamento de medida de distância antes do horário previsto para o início da passagem;
- Recepção de telemetria que é uma ação passiva, ou seja, o sistema de telemetria fica disponível durante toda passagem para receber automaticamente as telemetrias enviadas pelo satélite;
- Envio de telecomando. Esta ação está relacionada ao satélite sendo rastreado.
- Cada satélite possui um conjunto diferente de telecomandos. Os telecomandos podem ser classificados de acordo com o seu envio da seguinte forma:
    - Todas as passagens: telecomandos que devem ser enviados em todas as passagens úteis;
    - Diários: devem ser enviados todos os dias;
    - Periódicos: aqueles que são enviados de acordo com a rotina de operação estabelecida para o satélite;
- Execução de medida de distância entre o satélite e a estação. Esta ação tem uma duração e deve ser executada quantas vezes for possível durante a passagem, mas não pode ser concorrente com a

ação de envio de telecomando. As medidas coletadas são usadas na geração do próximo plano de previsão de passagem;

- Execução de medida de velocidade do satélite em relação à estação. Esta ação também deve ser executada durante toda a passagem. As medidas coletadas também são entradas para a geração do próximo plano de previsão de passagem.

Para cada um dos Satélites de Coleta de Dados-1 (SCD1) e Satélites de Coleta de Dados-2 (SCD2) do INPE, foram rastreadas 57 passagens em média durante o período de uma semana por estarem estes satélites em órbitas de baixa inclinação. Já para *o China Brazil Earth Resource Satellite* (CBERS2), que tem uma órbita polar heliosíncrona, foram rastreadas 27 passagens em média durante o período de uma semana. As passagens de alguns satélites podem ter prioridade sobre as passagens de outros como ocorria com o CBERS2, cuja passagens concorrentes tinham prioridade sobre as passagens dos SCDs, sendo então descartadas. Como o ambiente de rastreio é alterado a cada passagem, um plano de operação de voo para cada passagem do satélite é gerado para cada estação utilizada em seu rastreio.

No INPE, o rastreio e controle da operação de vários satélites por uma mesma estação terrena de rastreio tem se tornado uma preocupação com a perspectiva de aumento no número de lançamentos. Faz-se necessário incluir maior possibilidade de reconfiguração e confiabilidade às tarefas do ambiente de controle de missões espaciais para várias missões.

Longas distâncias, limites nos canais de comunicação e altos custos operacionais por causa da necessidade de mão-de-obra especializada são limitantes decorrentes das operações tradicionais. Alguns satélites não podem ser tele-operados devido a limites impostos por aspectos operacionais (BRAT et al., 2006). Uma tendência dos sistemas espaciais, para minimizar tais restrições é o aumento da autonomia operacional, uma vez que as aplicações

espaciais podem ser beneficiadas com a implantação de sistemas autônomos. O alto custo para manter uma equipe de operação para as missões espaciais tem sido uma preocupação de toda a comunidade da área espacial. O Centro de Operação Espacial Europeu (ESOC), durante os últimos anos, tem como objetivo explorar caminhos possíveis para aumentar a automação das atividades de operações de satélites (ERCOLANI et al, 2004).

Em conformidade com esta tendência atual, estudos recentes em planejamento baseado em conceitos de inteligência artificial estão sendo realizados no INPE, visando o desenvolvimento de ferramentas capazes de automatizar as tarefas de controle das operações de solo de seus satélites (BIANCHO et al., 2006), (CARDOSO et al., 2006). Um dos projetos concebidos até o momento consiste na automatização de operações espaciais do segmento solo, como uma primeira abordagem de automatização dentro do contexto das missões espaciais brasileiras, a caminho da automatização e autonomia de ambos os segmentos (BIANCHO et al., 2006).

A proposta para automatização das operações espaciais do segmento solo é a arquitetura *Multi-Agent Ground-operation Automation (MAGA)* (BIANCHO et al., 2006), que tem como objetivo gerenciar a alocação dos recursos em solo para o rastreio e controle da operação de múltiplos satélites por uma mesma estação terrena de rastreio. Nessa arquitetura, os agentes atuam de forma hierárquica e distribuída para a geração de planos para o controle de missões espaciais. Esta característica hierárquica presente na arquitetura deve-se à relação de dependência funcional entre os agentes. A Figura 2.4 ilustra a arquitetura MAGA, apresentando a hierarquia de planejamento existente entre os agentes que constituem essa arquitetura.

Estes agentes planejadores são especialistas na geração de planos do segmento de solo. Estes planos se inter-relacionam, interagindo durante o processo de planejamento de forma seqüencial e ordenada para a obtenção de

um plano único e final, pronto para ser executado por um agente responsável pela automatização da execução.

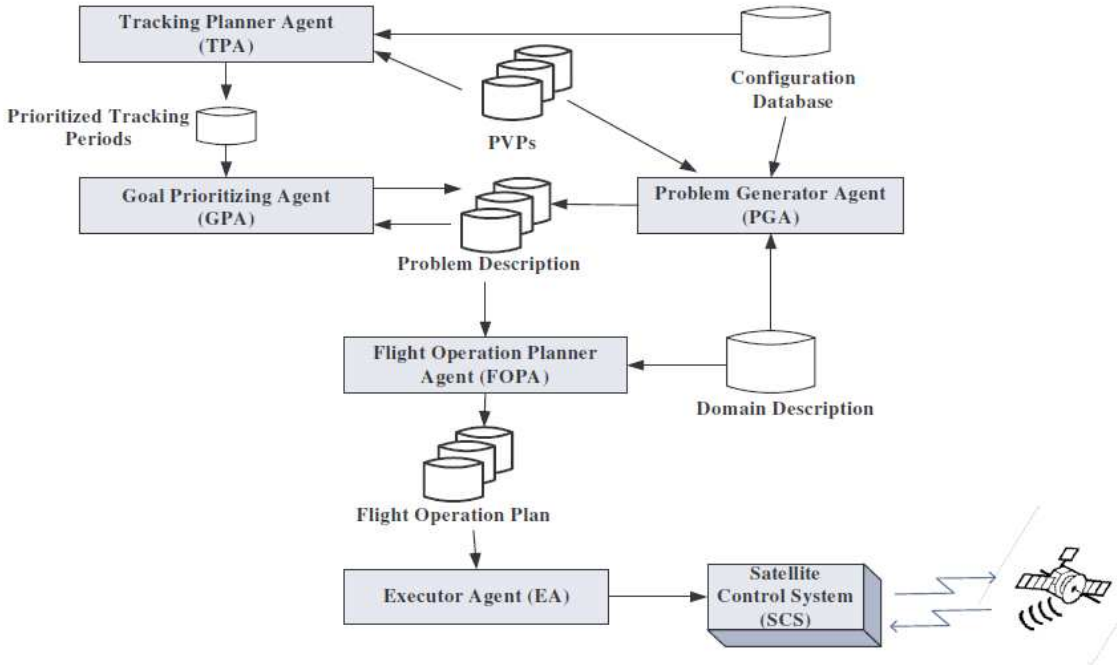

Figura 2.4 - Arquitetura de automação de operações solo multi-agente (MAGA)
Fonte: Biancho et al. (2006).

Os módulos de geração do problema, configuração, planejamento e visualização da arquitetura MAGA tiveram seus agentes implementados dentro de outro projeto de planejamento usando inteligência artificial no INPE. O resultado foi o desenvolvimento do sistema de Planejamento Inteligente de Planos de Operação de Vôo (PlanIPOV) (CARDOSO et al., 2006). Este sistema utiliza técnicas de planejamento temporal baseado em inteligência artificial (planejador temporal LPG-TD) na geração automática de planos de operação de voo para o suporte às atividades de controle de satélites em solo.

Cada plano de operação de voo gerado deve atender à fase operacional de rotina de um satélite no Centro de Rastreio e Controle de Satélites (CRC) do INPE, conforme apresentado na Figura 2.5.

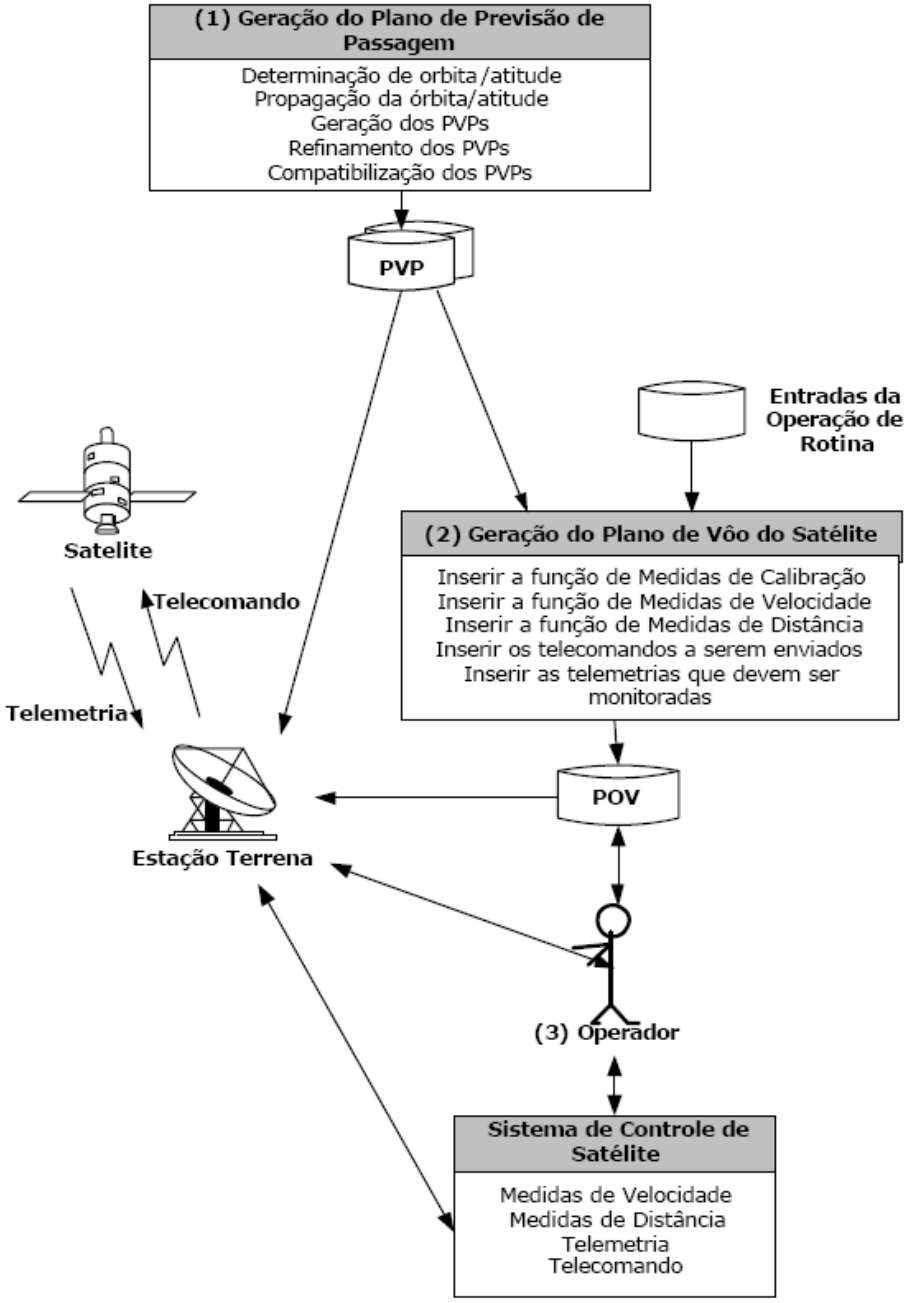

Figura 2.5 - Preparação e Execução do plano de operação de voo

Fonte: Cardoso (2006).

O desafio na geração automática de planos para a operação de satélites do INPE está em garantir que todas essas ações a serem executadas sejam inseridas no plano em instantes adequados, visto que, cada ação do plano deve respeitar tanto as restrições de tempo como de recursos.

Em suma, o conjunto de ações contidas em um plano de operação de voo seja gerado manualmente ou de forma automática, age diretamente sobre dados críticos para manutenção da integridade do satélite em órbita. Sendo, portanto necessária uma avaliação cuidadosa destes planos, que dependendo da demanda de satélites em operação, pode tornar-se inviável.

2.4 Segurança no Planejamento da Operação de Sistemas Espaciais

Nos projetos espaciais desenvolvidos pelas principais agências internacionais têm se utilizado ferramentas baseadas em técnicas de inteligência artificial para o controle de operações de missão espacial. Essas ferramentas vêm sofrendo melhorias contínuas decorrentes da própria experiência ou como resultado de restrições impostas aos novos conceitos de planejamento de missão.

O objetivo principal dessas ferramentas é a obtenção de um plano de execução que contemple as operações rotineiras e de contingências, ou seja, um conjunto de operações de rotina que busca balancear requisitos e recursos, escalonando um grande número de atividades de curto, médio e longo prazo (RABENAU e PESCHKE, 2004; RABENAU et al., 2002).

De acordo com os órgãos de pesquisas das principais agências espaciais internacionais é inegável que o desenvolvimento de sistemas autônomos tem sido a chave para solucionar os problemas relacionados às aplicações que envolvem exploração espacial, tais como: as grandes distâncias, os limites de comunicação, bem como os altos custos de operação. No entanto, o risco e o custo dessas missões espaciais levam a certa resistência para empregar uma tecnologia nova, complexa e de difícil entendimento. Em função dessa resistência existe forte coação para que o processo de desenvolvimento demonstre prover a segurança e confiança exigidas nesse tipo de aplicação (BLANQUART et al., 2004).

O processo de desenvolvimento dos sistemas de solo segue procedimentos e práticas clássicas existentes na Engenharia de Software, além de atender as normas e padronizações exigidas pelas agências espaciais internacionais para garantia de qualidade do software. Entre as práticas, a verificação dos sistemas de solo e das operações de controle de missões espaciais baseia-se na simulação.

A simulação pode ocorrer por meio da experiência ou por meio de ensaios realizados com o auxílio de modelos, pela utilização de sinais e dados simulados e ainda pela utilização de simuladores. Os simuladores são programas especiais de computador capazes de analisar e simular o comportamento de um sistema físico. A simulação de um satélite pode conter, por exemplo, intervalos de rotina durante a pré-passagem ou pós-passagem, ou decorrentes de uma manutenção ou atualização do sistema.

A verificação por simulação também possibilita diagnosticar, depurar, localizar falhas e calibrar equipamentos. Quando implementada na totalidade, a verificação por simulação permite não somente testar elementos e componentes do segmento solo, mas o sistema como um todo. No entanto, como este tipo de simulação tem custo bastante elevado, na prática a simulação é empregada nos elementos considerados críticos (WERTZ e LARSON, 1999).

Para sistemas de solo desenvolvidos com autonomia para controlar as operações de veículos espaciais, as agências espaciais internacionais referem-se à engenharia dita “similar”, em que os princípios básicos do processo de verificação e validação (V&V) de componentes são muito próximos ao da engenharia de software clássica (PRESSMAN, 2006; SOMMERVILLE, 2003). Além dos procedimentos e práticas clássicas, as atividades de V&V nos sistemas com autonomia de software devem incluir uma metodologia de testes específicos para ganho de confiabilidade (POWELL e THÉVENOD-FOSSE, 2002).

Estes desafios crescentes em relação à segurança e confiabilidade nos sistemas espaciais com maior autonomia nas operações de controle de veículos espaciais, decorrentes da complexidade e criticalidade motivaram a *European Space Agency* (ESA) a coordenar um grande projeto, visando à garantia do produto de software dos sistemas espaciais autônomos que realizam operações de solo e bordo.

O projeto *Software Product Assurance for Autonomy on-board Spacecraft* (SPAAS), resultou de uma parceria entre os seguintes centros de pesquisa e tecnologia europeus: *European Aeronautic Defence and Space company* (EADS ASTRIUM), *Laboratoire d'Architecture et d'Analyse des Systèmes* (LAAS-CNRS), *European Space Research and Technology Centre* (ESTEC) e o *Société d'ingénierie experte en informatique industrielle technique et scientifique* (AXLOG Ingénierie).

O objetivo do projeto consistiu em investigar as medidas para garantia do produto de software que utiliza técnicas avançadas em inteligência artificial na geração de autonomia.

Durante o desenvolvimento do projeto SPAAS, diversos métodos relacionados à segurança foram analisados, sendo recomendadas normas e padronizações. Os integrantes do projeto desenvolveram ainda componentes de software relacionados à segurança para tratamento de situações imprevistas provenientes desta maior autonomia nas operações de controle em veículos espaciais. As principais recomendações foram (BLANQUART et al., 2004):

- Uso de testes exaustivos de simulação;
- Uso de técnicas de segurança monitoradas, tais como: um supervisor ou validador.

O componente de software desenvolvido no SPAAS para o tratamento de situações imprevistas, nomeado *plausibility checker* (BLANQUART et al., 2004) tem como função suportar e complementar a validação das ações de um software de solo, especialmente no controle dos procedimentos de bordo gerados de forma automática antes de serem enviados para execução real. A arquitetura geral e a forma como atua esse componente é apresentada na Figura 2.6.

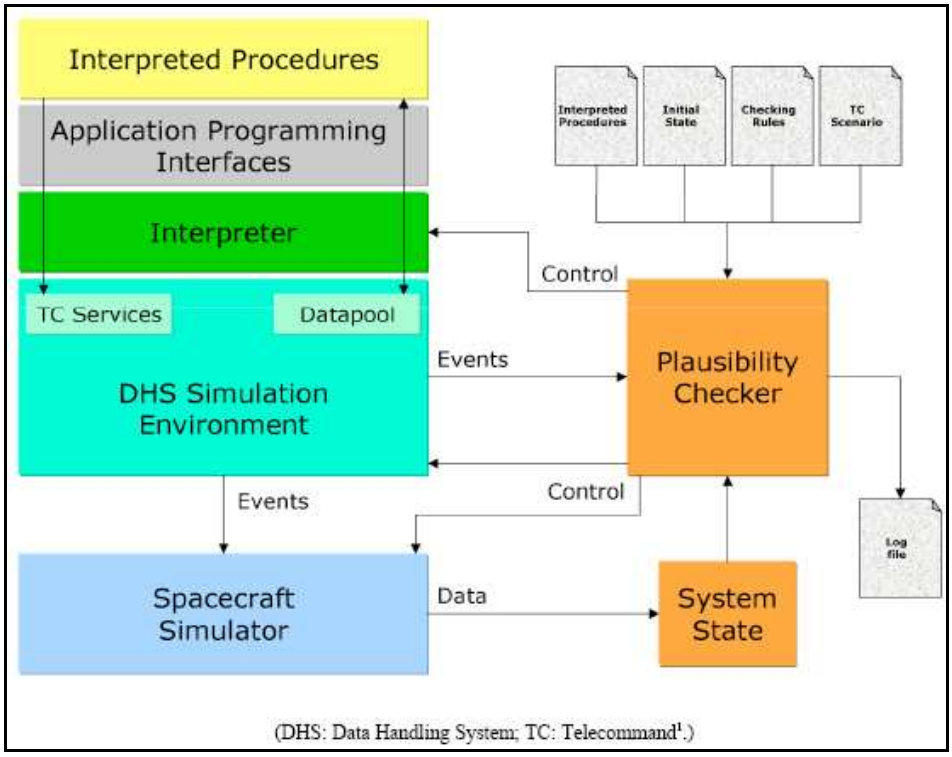

Figura 2.6 - Arquitetura do verificador de plausibilidade

Fonte: Blanquart et al. (2004).

O *plausibility checker* controla os comandos a serem enviados para o simulador da espaçonave ou para o simulador *Data Handling System* (DHS), baseando-se nas informações recebidas do próprio simulador da espaçonave (*Spacecraft Simulator*) sobre o estado simulado da espaçonave. Para realizar o controle dos comandos, o componente *plausibility checker* também recebe de arquivos

de dados, as informações relativas às operações de solo, gerando um arquivo de registro de eventos.

O componente *plausibility checker* foi desenvolvido em Java, sendo testado em *workstations* e microcomputadores dentro de um ambiente simulado do DHS e de simulação da espaçonave. O DHS é responsável pela manipulação de dados de todos os sistemas a bordo da espaçonave, inclusive o próprio computador de bordo.

O resultado dos testes com o *plausibility checker* demonstrou, que o controle realizado por este componente aliado ao uso de testes exaustivos de simulação são fundamentais para a validação dos comandos de solo gerados de forma automática antes da execução real.

O resultado do projeto SPAAS recomenda a elaboração e inclusão de um componente de software específico semelhante ao *plausibility checker* como um componente genérico a ser instanciado e usado em projetos espaciais reais com automatização nas operações de missões espaciais para desempenhar o suporte à segurança e confiabilidade.

Para estar em conformidade com as recomendações fornecidas pelo projeto, que envolveu vários centros de pesquisas espaciais internacionais e desenvolver um componente de software similar ao *plausibility checker*, torna-se necessário que este componente seja capaz de manipular grandes quantidades de dados gerados pelo simulador e também capaz de realizar a validação dos comandos de operação a serem enviados. E, assim, garantir a segurança das operações que controlam satélites antes da execução real.

As recomendações não incluem detalhes de projeto para o desenvolvimento do componente de software similar ao *plausibility checker*, entretanto, a arquitetura apresentada na Figura 2.6 indica que para realizar a validação dos comandos,

o validador faz uso de uma base de conhecimento do satélite desenvolvida por especialistas.

Na sua totalidade, os artigos da área espacial relacionados à segurança no planejamento das operações fazem referência ao desenvolvimento de componentes de hardware e software para simuladores voltados a realização de testes de verificação e validação.

Um simulador de satélite é concebido para representar fielmente o comportamento do satélite, por este motivo o desenvolvimento de simuladores envolve custo bastante elevado em função da construção da modelagem de um ou mais subsistemas e da geração da base de conhecimento do satélite por especialistas. Mesmo decorrida uma década (WERTZ e LARSON, 1999), os simuladores ainda apresentam custo bastante elevado.

Na busca de uma solução simplificada como alternativa ao desenvolvimento de simuladores para realizar predições de estados operacionais de satélites, vislumbra-se à possibilidade de se utilizar dados operacionais do satélite para classificar o estado do satélite. Nesta direção, o conceito de mineração de dados torna-se bastante apropriado por utilizar um modelamento matemático para extrair conhecimento de uma base de dados operacionais do satélite classificados por especialistas e prever estados futuros a partir dos dados coletados nas operações de rotina, evitando a construção de uma base de conhecimento do satélite inerente ao desenvolvimento de um simulador.

No próximo capítulo será apresentada uma visão geral da mineração de dados, uma descrição da metodologia e técnicas existentes a serem empregadas para o desenvolvimento de um componente de software voltado a segurança no planejamento das operações de satélites em órbita.

3 MINERAÇÃO DE DADOS

A mineração de dados consiste em uma área que combina métodos tradicionais de análise de dados com algoritmos sofisticados para processar dados. Em função desta característica, ela abriu oportunidades para exploração e análise de dados. Neste capítulo, será apresentada uma visão geral da mineração de dados, incluindo a metodologia, descrevendo as fases ou passos de transformação dos dados, que compõem o processo de exploração e análise dos dados, bem como técnicas para detectar e descrever padrões estruturais de dados, dependendo do objetivo pretendido para a aplicação.

Também será apresentado um método empregado pelos algoritmos classificadores para validação dos modelos de classificação construídos a partir da base de dados, as métricas para avaliação do desempenho de um classificador, a estatística Kappa e as funções estatísticas para avaliação de classificadores. Por fim, serão apresentados trabalhos correlatos na área espacial voltado a segurança no planejamento das operações de satélites em órbita.

3.1 Mineração de Dados

A prospecção de dados ou mineração de dados (*Data Mining*) consiste no processo de descoberta automática de informações úteis em depósitos de dados, podendo revelar informação estratégica escondida nos dados, principalmente nas situações que envolvem a análise de grandes conjuntos de dados. Além da busca por padrões consistentes, a mineração de dados permite o estabelecimento de relacionamentos sistemáticos entre variáveis, que podem ser validados pela aplicação de novos subconjuntos de dados. O processo consiste basicamente em 3 etapas: exploração; construção do modelo ou definição do padrão; verificação e validação (WITTEN e FRANK, 2005).

Algumas etapas da mineração de dados estão intimamente ligadas ao aprendizado de máquina, que consiste em uma sub-área da Inteligência artificial preocupada com a criação de algoritmos que aprendem com o ambiente no qual está exposto. Se fornecermos a um algoritmo de aprendizado uma grande massa de dados, ele será capaz de retirar algumas conclusões sobre as relações existentes nestes dados. Os algoritmos de aprendizado de máquina transformam dados em regras que expressam o que há de importante nos dados (CARVALHO, 2005).

A mineração de dados constitui parte de um processo maior de descoberta de conhecimento em banco de dados - *Knowledge Discovery in Databases* (KDD). Trata-se do processo de conversão, por meio de passos de transformação de dados brutos em informações úteis. KDD consiste, fundamentalmente, na estruturação; na seleção, preparação e pré-processamento dos dados, bem como na transformação, adequação e redução da dimensionalidade dos dados. A dimensão de um conjunto de dados consiste no número de atributos que os objetos desse conjunto de dados possuem. Durante o processo KDD de descoberta de conhecimento, modelos computacionais são construídos para a descoberta automática de novos fatos e relacionamentos entre dados, a partir da utilização repetida e muitas vezes interativa, de algoritmos de busca.

O processo KDD é interativo e iterativo, envolvendo uma série de etapas onde cada uma pode requerer do usuário capacidade de análise e de tomada de decisão. As principais fases do processo são descritas a seguir e representadas na Figura 3.1 (GOLDSCHMIDT e PASSOS, 2005).

- Seleção: análise dos dados existentes e seleção daqueles a serem utilizados na busca por padrões e na geração de conhecimento novo;

- Pré-processamento: tratamento e preparação dos dados para uso pelos algoritmos. Nesta etapa deve-se identificar e retirar valores inválidos, inconsistentes ou redundantes.

- Transformação: utilização, quando necessário, de alguma transformação linear ou mesmo não linear nos dados, de forma a encontrar aqueles mais relevantes para o problema em estudo. Nesta etapa geralmente são aplicadas técnicas de redução de dimensionalidade e de projeção dos dados.

- Mineração: busca por padrões a partir da aplicação de algoritmos e técnicas computacionais específicas.

- Interpretação: análise dos resultados da mineração e na geração de conhecimento pela interpretação e utilização dos resultados em benefício do negócio.

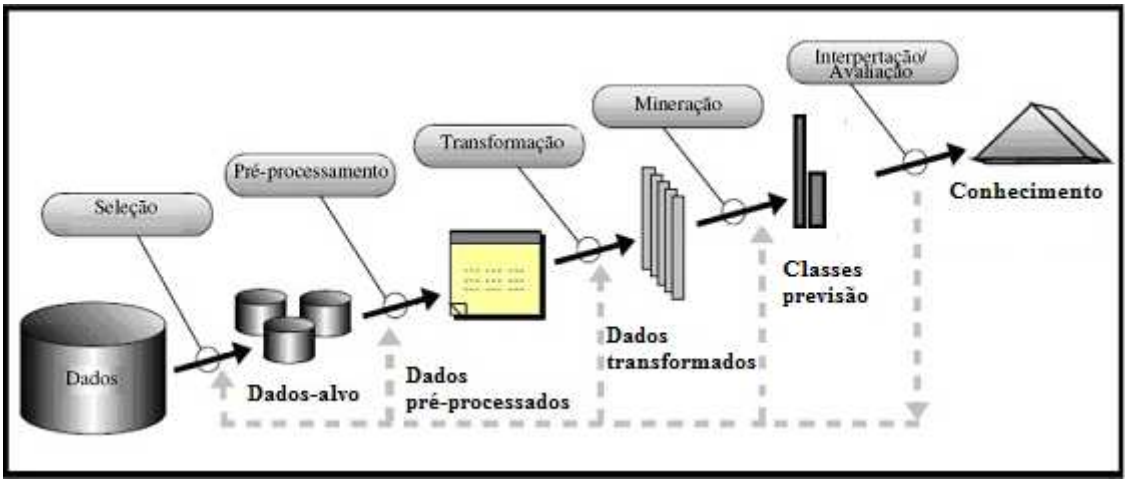

Figura 3.1 - Fases do processo KDD

Fonte: Adaptada de Herden (2007).

A fase de mineração envolve metodologias e técnicas específicas para a descoberta de novas relações por meio de procedimentos mecânicos fornecidos a partir da sub-área de aprendizado de máquina, que realizam uma análise sistemática e exaustiva sobre inúmeros registros contidos em uma base

de dados. Cada uma das metodologias de mineração necessita basicamente das mesmas técnicas para a sua realização. Sendo as técnicas de mineração organizadas de forma a agir sobre grandes bancos de dados com o intuito de descobrir padrões úteis e recentes, que de outra forma poderiam permanecer ignorados. As técnicas também fornecem capacidade de previsão do resultado de uma observação futura.

3.2 Metodologia e Técnicas

A fase de mineração pode ser realizada de três diferentes formas, onde cada uma delas corresponde a uma metodologia diretamente relacionada ao nível de conhecimento que se tenha da aplicação (CARVALHO, 2005):

- Descoberta não-supervisionada das relações: emprega-se esta metodologia quando não se tem conhecimento sobre as relações que se deseja estudar. Neste caso, é possível simplesmente deixar que as técnicas automáticas de mineração procurem nos dados relações escondidas.

- Teste de hipótese: emprega-se esta metodologia quando existe algum tipo de conhecimento sobre o campo de atuação ou alguma ideia sobre que relação se deseja buscar. Neste caso, define-se uma hipótese e verifica-se sua confirmação ou refutação.

- Modelagem de dados: emprega-se esta metodologia quando se tem um conhecimento maior da área de atuação e da relação que se deseja estudar.

As técnicas de mineração são de caráter genérico, podendo ser cada uma delas implementadas por diversas ferramentas. De uma forma geral, 5 técnicas abrangem todas as outras. São elas: classificação (descritiva ou preditiva), análise de associação, análise de agrupamentos, detecção de anomalias.

A classificação consiste na tarefa de organizar objetos em uma entre diversas categorias pré-definidas. Trata-se de um problema universal que abrange muitas aplicações diferentes, sendo por isso, uma das técnicas mais utilizadas em mineração de dados e que envolve uma abordagem sistemática para a construção de modelos de classificação de um conjunto de dados de entrada (CARVALHO, 2005).

Conforme o propósito, um modelo de classificação pode ser descritivo ou preditivo. A modelagem descritiva busca derivar padrões para evidenciar os relacionamentos entre os dados e requerem, além do pré-processamento, técnicas de pós-processamento para explicar os dados, principalmente quando sua natureza tem cunho exploratório. Enquanto a modelagem preditiva tem como propósito prever o valor de um determinado atributo baseado nos valores de outros atributos. O atributo a ser previsto é conhecido como a variável dependente ou alvo, enquanto que os atributos usados para fazer a previsão são conhecidos como as varáveis independentes ou explicativas. Existem 2 tipos de tarefas de modelagem de previsão: classificação, a qual é usada para variáveis alvo discretas com argumento categórico, e regressão, utilizada para variáveis alvo contínuas (TAN et al., 2009). A regressão consiste na busca por uma função que represente, de forma aproximada, o comportamento apresentado pelo fenômeno em estudo. A forma mais conhecida de regressão é a linear, mas também pode ser não-linear.

Aplicações onde o modelo preditivo exige um juízo necessário para as decisões futuras; emprega-se um esquema de aprendizagem de classificação, em que a partir de um conjunto de exemplos classificados (dados de treinamento), espera-se aprender uma maneira de classificar amostras não conhecidas (dados de teste) (WITTEN e FRANK, 2005).

A Análise de associação é usada para descobrir padrões que descrevam características fortemente associadas dentro dos dados. Os padrões descobertos são normalmente representados na forma de regras de implicação

ou subconjuntos de características. Devido ao tamanho exponencial do seu espaço de busca, o objetivo da análise de associação é extrair os padrões mais interessantes de uma forma eficiente (TAN et al., 2009).

A análise de agrupamentos procura encontrar grupos (clusters) de observações intimamente relacionados de modo que as observações que pertençam ao mesmo grupo sejam mais semelhantes entre si do que com as que pertençam a outros grupos. Esta técnica busca similaridades entre os dados para definir um conjunto finito de classes ou categorias que os contenha e os descreva. Por ser o objetivo dessa implementação a tarefa de encontrar clusters, essa técnica tornou-se conhecida como clusterização (*clustering*), sendo usadas também denominações como agrupamento, segmentação ou agregação (TAN et al., 2009).

A detecção de anomalias é a tarefa de identificar observações cujas características sejam significativamente diferentes do resto dos dados.Tais observações são conhecidas como anomalias ou fatores estranhos. O objetivo de um algoritmo de detecção de anomalias consiste em descobri as anomalias verdadeiras e evitar rotular erroneamente objetos normais como anômalos (TAN et al., 2009).

3.3 Construção de Modelos de Classificação para Predição

Cada técnica de classificação (ou classificadora) utiliza um algoritmo de aprendizado para identificar o modelo que melhor se adapta a relação entre o conjunto de atributos (entrada) e o rótulo de classe (saída) do conjunto de dados (Figura 3.2). O modelo gerado por um algoritmo de aprendizagem deve ser capaz de prever corretamente os rótulos de classe de registros desconhecidos. Portanto, o objetivo de um algoritmo de aprendizagem é a construção de modelos com considerável capacidade de generalização, ou seja, modelos que permitam prever com precisão os rótulos de classe de registros desconhecidos (TAN et al., 2009).

Figura 3.2 – Classificação: mapear um conjunto de atributos no seu rótulo de classe
Fonte: Adaptada de Tan et al. (2009).

Os dados de entrada para tarefa de classificação são um conjunto de registros. Cada registro ou instância ou exemplo, caracteriza-se por uma dupla (x,y), onde x é o conjunto de atributos e y o atributo especial, designado como rótulo da classe, também chamado de atributo alvo, variável alvo ou categorização. Técnicas de classificação são mais apropriadas para prever ou descrever um conjunto de dados com categorias nominais ou binárias.

Para a construção do modelo de classificação a partir de um conjunto de dados de entrada, utiliza-se uma técnica de classificação, onde cada técnica, emprega um algoritmo de aprendizagem para identificar o modelo mais apropriado para o relacionamento entre o conjunto de atributos e o rótulo da classe dos dados de entrada. O modelo gerado pelo algoritmo de aprendizagem deve se adaptar bem aos dados de entrada e prever corretamente os rótulos de classe de registros ou amostras não conhecidos previamente.

Para a resolução de problemas de classificação, um conjunto de treinamento deve ser fornecido, contendo registros, cujos rótulos de classe sejam conhecidos. O conjunto de treinamento é utilizado para construir o modelo de classificação, que é aplicado ao conjunto de teste, contendo registros com rótulos de classe desconhecidos. Ao conjunto de treinamento, aplica-se uma das técnicas para a construção do modelo de classificação, a partir da seleção de um dos classificadores existentes. Nas seqüentes subseções serão

apresentados os classificadores voltados à construção de modelos de classificação preditiva.

3.3.1 Classificadores

Classificadores para construção de modelos de classificação devem ser selecionados de acordo com critérios como (HAN e KAMBER, 2006):

- Precisão do modelo: capacidade do modelo de prever corretamente o rótulo da classe de amostras desconhecidas;
- Velocidade: refere-se ao custo computacional associado à geração e uso do modelo;
- Robustez: capacidade do modelo de fazer previsões corretas na presença de dados anômalos ou faltantes;
- Escalabilidade: capacidade de construção de um modelo eficiente para uma grande quantidade de dados;
- Interpretabilidade: refere-se ao nível de entendimento e discernimento provido pelo modelo.

Segundo Tan et al. (2009) existem 6 tipos de classificadores com abordagens distintas na execução da tarefa de classificação. São eles:

- Classificadores baseados em árvore de decisão: apresentam uma abordagem determinística, cuja construção da árvore ocorre conforme a informação extraída dos dados;
- Classificadores baseados em redes neurais artificiais: não conhecem a distribuição, extraindo assim, a estatística dos dados;

Classificadores bayesianos: apresentam uma abordagem não determinística, baseiam-se na inferência probabilística;

Classificadores baseados em vetores de suporte: baseiam-se na teoria de aprendizado estatístico para minimizar erros da classificação empírica e maximizar a margem geométrica entre os resultados;

Classificadores de vizinho mais próximo: procuram encontrar todos os exemplos de treinamento, que sejam relativamente semelhantes aos atributos do exemplo de teste;

Classificadores baseados em regras: classificam os registros a partir de um conjunto de regras.

As subseções seguintes apresentam em detalhes, cada um dos classificadores e os algoritmos desenvolvidos para implementação.

3.3.1.1 Classificadores Baseados em Árvore de Decisão

Um classificador de árvore de decisão, consiste em uma técnica de classificação baseada em algoritmos de particionamento sucessivo. Também conhecida como indução de árvore de decisão, decorre do princípio de dividir para conquistar para produzir árvores de decisão. A partir da informação contida nos dados, cada ramo da árvore surge em função de um questionamento de classificação e cada folha é uma partição do conjunto de dados contendo a sua respectiva classificação. Uma árvore de decisão é formada pelo nó raiz, arestas e outros nós internos e terminais, que resultam da separação de registros com características diferentes (TAN et al., 2009).

A forma de execução é simples: dado um conjunto de dados com uma das variáveis como objeto de saída, o algoritmo encontra o atributo mais fortemente relacionado à variável de saída, determinando-o como o primeiro ramo chamado de raiz. Os demais atributos são subseqüentemente classificados como nós até que se chegue ao último nível, a folha. Assim, a árvore de decisão utiliza a estratégia de dividir para conquistar, de forma que um

problema complexo é decomposto em problemas mais simples, sendo recursivamente, a mesma estratégia aplicada a cada problema simplificado.

Uma vez construída a árvore de decisão, um dado novo pode ser classificado por ela em um de seus subgrupos, desde que , iniciando seu trajeto no nó raiz da árvore, este seja dirigido para a direita ou esquerda a cada nó de acordo com a regra heurística (decisão) associada ao mesmo até chegar a um nó terminal (folha) da árvore.

Os algoritmos de árvore decisão aprendem com os exemplos, gerando uma árvore de decisão e um conjunto de regras associadas . No entanto, conforme o problema, algumas árvores de decisão obtidas podem ser mais precisas que outras, visto que encontrar a árvore ótima seria computacionalmente inviável por causa do tamanho exponencial do espaço de pesquisa. Apesar disto, algoritmos eficientes têm sido desenvolvidos para induzir a uma árvore de decisão razoavelmente precisa, embora não perfeita (TAN et al., 2009).

Estes algoritmos ainda incluem a poda da árvore, ou seja, o algoritmo de desenvolvimento da árvore é parado antes de gerar uma árvore totalmente desenvolvida, mas que satisfaça perfeitamente todos os dados de treinamento. Um desses algoritmos é o algoritmo de Hunt (TAN et al., 2009), que constitui a base de muitos algoritmos de indução de árvores de decisão existentes, como o *Induction Decision Tree* (ID3) (TAN et al., 2009), que foi o primeiro algoritmo sobre o assunto, C4.5 (Figura 3.3), um melhoramento do anterior, *Chi-Squared Automatic Induction* (CHAID) e *Classification and Regression Trees*.(CART) (WITTEN e FRANK, 2005).

Por muitos anos a seleção do subconjunto de teste foi feita pelo algoritmo ID3 unicamente com base no critério de ganho de informação. Este critério está baseado na redução da entropia causada pelo particionamento das amostras de acordo com os seus atributos (HAN e KAMBER, 2006). A entropia ou variância é uma medida de quanto um sistema é desorganizado e em relação à

quantidade de informação, mede o grau de impureza em uma coleção de amostras, ou seja, mede a homogeneidade dos subgrupos da árvore (CARVALHO, 2005).

```
Algoritmo C4.5
- repetir várias vezes (aproximadamente 10)
        CONSTRUIR
        Escolher conjunto de trabalho do conjunto de treinamento
        REPETIR
                formar árvore para conjunto de trabalho
                        SE critério de parada satisfeito
                                escolher melhor classe
                        SENÃO
                                escolher melhor teste de atributo
                                dividir conjunto de treinamento em concordância
                                formar árvore nos sub-conjuntos
                testar no resto do conjunto de treinamento
                adicionar itens mal classificados ao conjunto de treinamento
        ATÉ não haver melhorias
        PODAR
        ENQUANTO a árvore de decisão contiver sub-árvores complexas e com pouco benefício
                Substituir sub-árvores por folhas
- selecionar a árvore podada mais promissora
```

Figura 3.3 - Pseudocódigo do algoritmo C4.5 de indução de árvore de decisão.
Fonte: Adaptada de Witten e Frank (2005).

A entropia de um particular nó de uma árvore de decisão é a soma, sobre todas as classes representadas no nó, da proporção de registros pertencente a uma particular classe multiplicada pelo logaritmo base dois daquela proporção, cujo valor é dado por (Fórmula 3.1):

$$\text{EntropiaEsperada} = -\sum_{i=1}^{m} p_i \log_2(p_i) \qquad (3.1)$$

Na Fórmula 3.1, p_i é igual à proporção da classe i, no nó, i=1,2,...,m, onde m é o número de categorias da variável alvo. Para medir a Entropia real de uma determinada variável A, utiliza-se a Fórmula 3.2:

$$\text{EntropiaReal(A)} = \sum_{j=1}^{v} \frac{S_{1j} + \ldots + S_{mj}}{S} \text{EntropiaEsperada} \qquad (3.2)$$

Na Fórmula 3.2, S_{ij} representa os subconjuntos (quantidade de registros) da amostra S (o número total de registros), onde j=1,...,v, corresponde às categorias da variável de entrada. Então, o ganho de informação (A) para um determinado ramo da variável A pode ser obtido pela diferença entre a entropia esperada e a entropia real. O valor de ganho de informação (A) permite uma redução da entropia causada pelo conhecimento do valor da variável A.

Uma das otimizações encontradas no algoritmo C4.5 e versões recentes como o J48 é a introdução de um novo critério para este tipo de seleção, definido como taxa de ganho, onde uma normalização é efetuada sobre o ganho de informação para ajustar testes com muitas saídas.

3.3.1.2 Classificadores Baseados em Redes Neurais Artificiais

O interesse em empregar um classificador baseado em redes neurais artificiais para criar um modelo de classificação, surge por se tratar de técnicas não-paramétricas e não-lineares, que permitem o mapeamento dos dados de entrada em associação com os dados de saída, de forma que a saída da rede consiste justamente na classe associada à amostra.

Análoga à estrutura do cérebro humano, uma Rede Neural Artificial (RNA) é composta de um conjunto interligado de nodos e links direcionados. Consiste em um conjunto de elementos de processamento (neurônios formal), agrupados em diversas topologias e regido por procedimentos matemáticos, tais como: clusterização de vetores, otimização discreta, minimização de erros, entre outros (HAYKIN, 2001).

As redes neurais artificiais constituem-se em modelos computacionais paralelos baseados numa unidade atômica, o neurônio (Figura 3.4). Em geral,

estes modelos possuem inspiração neurobiológica, porém, na prática, são algoritmos computacionais representando, de maneira bastante elementar, o mecanismo de funcionamento cerebral. Atualmente, existe uma extensa variedade de RNAs disponíveis.

Uma rede neural artificial é composta por várias unidades de processamento, cujo funcionamento é bastante simples. Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede, conforme ilustra a Figura 3.4.

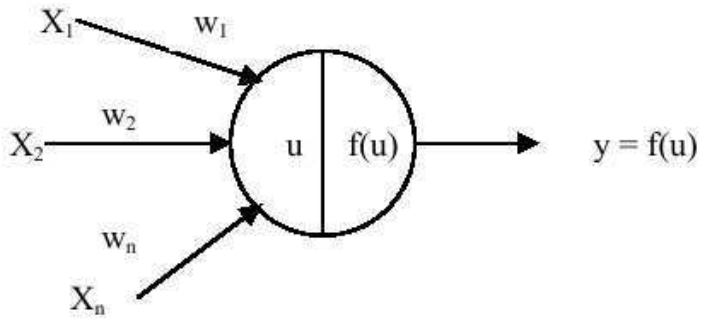

Figura 3.4 – Modelo de um neurônio artificial.
Fonte: Adaptada de Karrer et al (2005).

A operação de uma unidade de processamento, pode ser resumida nos seguintes passos:

- Sinais são apresentados à entrada;
- Cada sinal é multiplicado por um número, ou peso, que indica a sua influência na saída da unidade;
- Realizada a soma ponderada dos sinais que produz um nível de atividade;

- Se este nível de atividade exceder a certo limite (*threshold*) a unidade produz uma determinada resposta de saída.

As entradas do neurônio correspondem a um vetor $X = [X_1; X_2;...; X_n]$ de dimensão n. Para cada entrada X_i, há um peso correspondente w_i que simula a concentração de neurotransmissores da conexão sináptica. A soma ponderada das entradas X_i por seus respectivos pesos w_i é chamada de saída linear u. Esta saída deve ser submetida a uma função de ativação f para obter-se a saída de ativação y do neurônio, isto é, y = f(u). A função de ativação pode assumir várias formas, geralmente não-lineares.

A maioria dos modelos de redes neurais possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. Em outras palavras, as RNAs aprendem por meio de exemplos. Esta capacidade de aprender e de generalizar a informação aprendida credita as RNAs características de adaptabilidade, generalização e tolerância a ruídos, dentre outras (HAYKIN, 2001).

Outra característica importante das redes neurais está na capacidade de auto-organização, em que a rede atribui um padrão de entrada a uma classe dentre um conjunto de classes conhecidas. Este procedimento empregado no aprendizado de uma rede neural, denominado de algoritmo de treinamento e sua função principal corresponde a modificação dos pesos sinápticos visando atingir o objetivo (HAYKIN, 2001).

Cada vetor de entrada, que representa um padrão, pode ser visto como um estímulo aplicado à rede e corresponde a um só neurônio na camada de saída, que também se denomina neurônio vencedor. Ou seja, a rede funciona como um mapeador de um conjunto de atributos de entrada representados por neurônios que são ativados na camada de saída (KOHONEN, 2001).

A aprendizagem por quantização vetorial (Learning Vector Quantization – LVQ) consiste em uma rede neural artificial supervisionada baseada em competição onde cada unidade de saída representa uma classe ou categoria particular (KOHONEN, 1995). As redes LVQ aprendem a classificar os vetores de entrada nas classes definidas pelo usuário.

Uma rede LVQ consiste em duas camadas. A primeira camada, denominada de *Competitive Layer*, responsável por aprender a classificar os vetores de entrada. A segunda camada, denominada de *Liner Layer*, que define em qual classificação os valores se encaixam. A Figura 3.5 ilustra as camadas da arquitetura da rede LVQ.

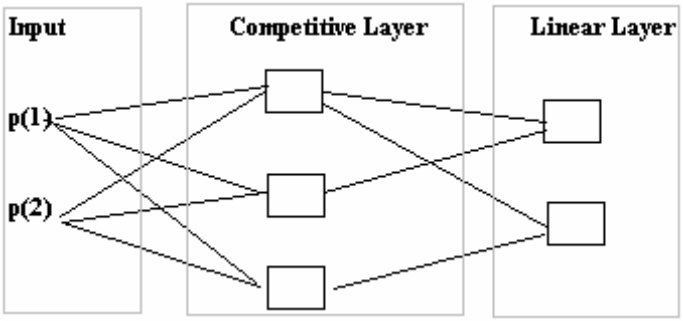

Figura 3.5 – Representação das camadas da arquitetura da LVQ
Fonte: Kohonen (1995).

Seu funcionamento difere bastante das outras redes, pois ao invés de aprender através de correções graduais baseadas nas saídas desejadas, as unidades da LVQ recebem já no início do aprendizado atribuições de qual classe representarão. A partir daí têm seus pesos ajustados de maneira a definir as fronteiras entre as classes, utilizando um conjunto de exemplos de padrões (FAUSETT, 1994; FREEMAN, 1991).

O vetor dos pesos das conexões de uma unidade chama-se também *codebook*. O vetor peso pode ser aleatório ou proveniente do conjunto de treinamento. Apenas a unidade que corresponde ao padrão, ou unidade vencedora (*Winner Takes All*), será analisada quando apresentado o padrão.

Se a classe a ela atribuída for a mesma que a da entrada, haverá um reforço do *codebook* da unidade de maneira que ele se torne um pouco mais próximo à entrada apresentada, de acordo com uma taxa de aprendizado. Se por outro lado as classes da unidade e da entrada forem diferentes, o *codebook* será ajustado de forma que a unidade se distancie da entrada. Com isso, torna-se gradualmente mais provável que as unidades certas reajam àquela entrada da próxima vez (KOHONEN, 1995). Os *codebooks* ficarão dispostos pelo espaço dos padrões deforma a cobrirem a área que reconhecem.

O LVQ busca ajustar o mapa de características para melhorar o desempenho da rede (BRAGA et al., 2007). A característica de método supervisionado possibilita que, após o treinamento da rede SOM, seja possível avaliar a classificação gerada por esta rede para cada padrão. Nessa avaliação, o LVQ ajusta os pesos para melhorar a classificação obtida pela rede SOM. Os passos para treinamento com algoritmo LVQ são apresentados a seguir:

1. Os pesos e parâmetros dos vetores são inicializados;
2. Para cada padrão de treinamento é definido o neurônio vencedor;
3. Os pesos deste e de seus vizinhos são atualizados, enquanto a taxa de aprendizado é reduzida.
4. Esse processo se repete até que o erro seja menor que um dado valor.

Para o ajuste dos pesos, este algoritmo usa a Equação 3.3, que compara, em cada padrão de entrada, a saída produzida com a saída desejada. Como LVQ pode calcular o erro, esta medida pode ser utilizada como critério de parada do algoritmo.

$$w_{ji}(t+1) = \{ \begin{matrix} w_{ji}(t) + \eta(t)(x_i(t) - w_{j_i}(t0), \text{se } j \in \Lambda(t), \text{classe correta} \\ w_{ji}(t), \text{caso contrário classe incorreta} \end{matrix} \quad (3.3)$$

Na Equação 3.3, $w_{ji}(t)$ corresponde ao peso da conexão entre o elemento de entrada x(t) e o neurônio j, $\eta(t)$ a taxa de aprendizado e Λ a vizinhança do neurônio vencedor em um instante de tempo t.

Os principais algoritmos de treinamento LVQ são LVQ1, LVQ2 e LVQ3. O processo de classificação é o mesmo para os três métodos, o que difere cada um deles é a forma de treinamento, ou seja, como as células são atualizadas. No algoritmo LVQ1, somente uma célula é atualizada por vez, enquanto no algoritmo LVQ2 duas células são atualizadas simultaneamente em cada iteração do processo de treinamento (BRAGA et al., 2007; DEMUTH et al., 2008; HAYKIN, 2001).

3.3.1.3 Classificadores Bayesianos

Existem aplicações em que o relacionamento entre o conjunto de atributos e o rótulo da classe pode não ser determinístico, isto é, o rótulo da classe de um registro de teste (vide seção 3.3.1.1) não pode ser previsto com certeza, embora seu conjunto de atributos seja idêntico a alguns dos exemplos de treinamento. Situações desta natureza podem surgir devido a dados com ruídos ou devido à presença de determinados fatores de confusão que afetam a classificação, mas que não são incluídos na análise.

Para resolver problemas de classificação com essa característica, adota-se uma abordagem baseada no teorema de Bayes para a construção de modelos fundamentados em um princípio estatístico que combina o conhecimento prévio das classes com novas evidências colhidas dos dados, ou seja, combina conhecimento a priori (probabilidade a priori ou incondicional) com dados de observação, requerendo a probabilidades à priori (RUSSELL e NORVIG, 2004).

Classificadores Bayesianos são classificadores estatísticos que classificam um objeto numa determinada classe, baseando-se na probabilidade deste objeto pertencer a esta classe. Quando aplicados a grandes volumes de dados,

apresentam desempenho comparável aos resultados produzidos por árvores de decisão e redes neurais (HAN e KAMBER, 2006).

Para descrever como o teorema de Bayes é utilizado na classificação, torna-se necessário formalizar o problema do ponto de vista estatístico: suponha X e Y um par de variáveis aleatórias. Sua probabilidade junta, P(X = x, Y = y), se refere à probabilidade da variável X receber o valor x e a variável Y receber o valor y. Uma probabilidade condicional é a de que uma variável aleatória receba um determinado valor dado que o resultado de outra variável aleatória seja conhecido. Por exemplo, a probabilidade condicional P(Y = y |X = x,) se refere à probabilidade da variável Y receber o valor y, dado que a variável X tenha o valor x. As probabilidades condicionais juntas levam a Fórmula 3.4, conhecida como teorema de Bayes (TAN et al., 2009):

$$P(Y / X) = \frac{P(X / Y)}{P(X)} \qquad (3.4)$$

O teorema de Bayes pode ser utilizado para classificação da seguinte forma: considera-se novamente uma perspectiva estatística, onde X denota o conjunto de atributos e Y a variável de classe. Se esta tiver um relacionamento não determinístico com os atributos, então se torna possível tratar X e Y como variáveis aleatórias e obter seu relacionamento utilizando probabilisticamente P(Y|X). Esta probabilidade condicional também é conhecida como probabilidade posterior de Y, em oposição a sua probabilidade anterior ou prévia ou a priori, P(Y).

Durante a fase de treinamento, torna-se necessário descobrir as probabilidades posteriores P(Y|X) para cada combinação X e Y baseada em informações coletadas a partir dos dados de treinamento. Conhecendo estas probabilidades, um registro de teste X’ pode ser classificado encontrando-se a classe Y’ que maximize a probabilidade posterior, P(Y’|X’).

Avaliar as probabilidades posteriores com precisão para cada combinação possível de rótulo de classe e valor de atributo torna-se um problema difícil,

porque requer um conjunto de treinamento muito grande, mesmo para um número moderado de atributos. O teorema de Bayes é bastante útil por permitir expressar a probabilidade posterior em termos da probabilidade anterior P(Y), a probabilidade condicional de classe P(X|Y) e a evidência P(X), conforme descreve a Equação 3.5:

$$P(Y / X) = \frac{P(X / Y) \times P(Y)}{P(X)} \tag{3.5}$$

Uma comparação entre as probabilidades posteriores para diferentes valores de Y, mostra que o termo denominador, P(X) se mantém constante e, portanto, pode ser ignorado. Sendo então, a probabilidade anterior P(Y) facilmente avaliada a partir do conjunto de treinamento, calculando-se a fração de registros de treinamento que pertence a cada classe. Para avaliar as probabilidades condicionais de classe P(X|Y), será apresentado o método de implementação de classificação bayesiano denominado Naive Bayes.

O classificador Naive Bayes também denominado Bayes simples ou classificador bayesiano ingênuo consiste na implementação de um método de classificação bayesiano, assim chamado por assumir que os atributos são condicionalmente independentes, ou seja, a informação de um evento não interfere na informação de nenhum outro.

Apesar desta premissa ingênua, simplista e pouco realista, o classificador Naive Bayes se apresenta eficaz em muitos casos práticos de aprendizado supervisionado, apresentando inclusive o melhor desempenho em várias tarefas de classificação quando comparado a outros classificadores com diferentes abordagens. Aliás, o classificador Naive Bayes apresenta um bom desempenho, apesar da hipótese de independência condicional ser freqüentemente violada. Mas sem esse tipo de suposição em algum nível o problema se torna intratável (HAN e KAMBER, 2006).

Um classificador bayesiano simples ou ingênuo avalia a probabilidade condicional de classe supondo que os atributos sejam condicionalmente

independentes, dados o rótulo de classe y. A independência condicional pode ser declarada da forma descrita pela Equação 3.6:

$$P(X \mid Y = y) = \prod_{i=1}^{d} P(X_i \mid Y = y) \tag{3.6}$$

Na Equação 3.6 cada conjunto de atributos X = $\{X_1, X_2, ...X_d\}$ consiste de *d* atributos. Sendo assim, com a suposição da independência condicional de classe para cada combinação de X (TAN et al., 2009; RUSSELL e NORVIG, 2004), apenas estima-se a probabilidade condicional de cada X_i, dado Y. Esta abordagem leva a uma significativa simplificação, uma vez que o conjunto de treinamento requerido não precisa ser muito grande para obtenção uma boa estimativa da probabilidade.

Para classificar um registro de teste, o classificador Naive Bayes, calcula a probabilidade posterior para cada classe Y, a partir da Equação 3.7

$$P(Y \mid X) = \frac{P(Y)\prod_{i=1}^{d} P(X_i \mid Y)}{P(X)} \tag{3.7}$$

Uma vez que P(X) mantém-se constante para cada Y, então basta escolher a classe que maximiza o numerador.

Classificadores Naive Bayes são robustos para pontos de ruídos isolados, porque calculam a média de tais pontos ao avaliar probabilidades condicionais a partir de dados. Também manipulam valores faltantes, ignorando o exemplo durante a construção e classificação do modelo (TAN et al., 2009).

3.3.1.4 Classificadores Baseado em Máquina de Vetor de Suporte

Um classificador baseado em máquina de vetor de suporte consiste em uma técnica fundamentada na teoria de aprendizagem estatística. A máquina de vetores de suporte, do inglês Support Vector Machine - SVM foi desenvolvida por (VAPNIK, 1995), com o intuito de resolver problemas de classificação de padrões. Esta técnica originalmente desenvolvida para classificação binária,

busca a construção de um hiperplano como superfície de decisão, de tal forma que a separação entre exemplos seja máxima. Isso considerando padrões linearmente separáveis.

A técnica SVM pertence a uma categoria de classificadores lineares e foi desenvolvida com o objetivo de auxiliar na solução de problemas de classificação e no reconhecimento de padrões. O seu conceito é baseado na ideia de minimizar riscos estruturais, ou seja, minimizar erros da classificação empírica e maximizar a margem geométrica entre os resultados.

Um classificador SVM linear, por exemplo, pode ser treinado para procurar um hiperplano em dados separáveis linearmente e assim, ser utilizado na classificação de padrões ou em regressões não lineares. Basicamente, o SVM é um algoritmo linear que constrói hiperplanos como superfícies de decisão de maneira que a fronteira de separação entre classes positivas e negativas seja maximizada (VAPNIK, 1998),

O funcionamento do algoritmo ocorre geometricamente por um hiperplano no espaço, o qual separa pontos que representam instâncias positivas da categoria dos pontos que representam instâncias negativas. O hiperplano é escolhido durante a etapa de treinamento como único e com margens máximas definidas. A margem consiste na distância do hiperplano entre o ponto mais próximo dos conjuntos positivos e negativos. Os hiperplanos são escolhidos por meio de um sub-conjunto das instâncias de treinamento que são chamados de vetores de suporte. Apenas parte dos dados de treinamento é utilizada para escolha do hiperplano, sendo que a outra parte não influencia nessa escolha. O classificador SVM tem como ponto forte a rapidez de execução, pois não depende da dimensionalidade do espaço amostral.

Segundo Tan et al. (2009) existem diversas classes de algoritmos SVM que atuam dependendo do tipo de problema de separação, mas os mais citados na literatura são:

- SVM linear com dados separáveis (Hard-margin);
- SVM linear com dados não separáveis (Soft-margin);
- SVM não linear.

A formulação inicial por Vapnik (1998) é conhecida como *Hard-margin* SVM. Esse algoritmo foi evoluído para o algoritmo *Soft-margin* SVM com tolerância a ruídos, erros, dados não linearmente separáveis, em que há uma escolha entre o tamanho da margem e o número de erros de treinamento na fronteira de separação.

A ideia básica por trás da técnica SVM está na escolha de um hiperplano de margem máxima possível. Conforme a Figura 3.6, a separação entre os casos apresentados pode ser realizada por duas margens B_1 e B_2, dado que as duas margens separam os casos sem que ocorram erros de classificação. Cada margem B_i está associada a um par de hiperplano b_{i1} e b_{i2}. O termo b_{i1} é obtido, movendo-se o hiperplano paralelamente da margem até que chegue perto de um ponto (círculo ou quadrado). O separador com maior margem, no caso B_1, é escolhido como o hiperplano de margem máxima para as instâncias de treinamento.

O problema pode ser definido como o problema de classificação de N exemplos de treinamento. Cada exemplo é definido por uma tupla $(x_i; y_i)$, (i = 1, 2, ... ,N), em que $x_i = (x_{i1}, x_{i2},...,x_{id})$ corresponde ao conjunto de atributos para o i-ésimo exemplo. Por convenção, seja $y_i \in \{-1,1\}$ o conjunto dos rótulos. A decisão de uma margem de um classificador linear (hiperplano separador) pode ser obtida pela Equação 3.8, em que w representa vetor normal perpendicular ao hiperplano e b são parâmetros do modelo (TAN et al.,2009).

$$w \cdot x + b = 0 \tag{3.8}$$

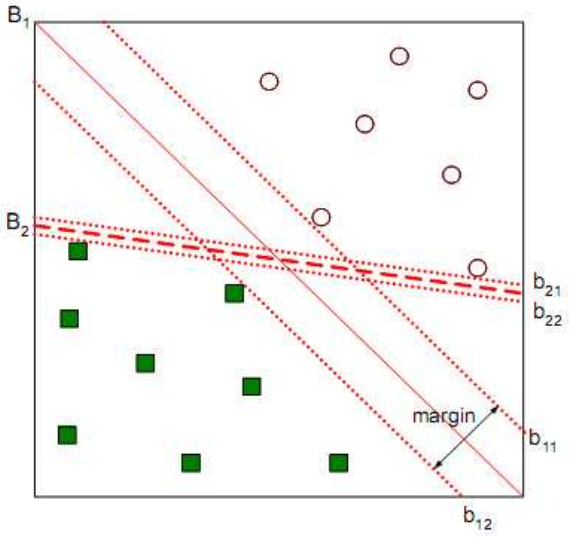

Figura 3.6 – Margem de uma margem de decisão

Fonte: Adaptada de Tan et al. (2009).

Sendo assim para o cálculo da margem de separação, definida por b_{11} e b_{12} na Figura 3.6, a Função 3.9 mostra os rótulos com valores -1 e 1 e assim eles podem ser previstos para qualquer exemplo de teste z. A Equação 3.10, mostra como pode ser obtida a margem d da margem de decisão, ou seja a distância entre a margem e as margens máximas.

$$y = \begin{cases} 1, & se \quad x \cdot z + b > 0; \\ -1, & se \quad x \cdot z + b < 0. \end{cases} \tag{3.9}$$

$$d = \frac{2}{\|w\|} \tag{3.10}$$

O aprendizado do modelo SVM linear compreende a fase de treinamento que envolve estimar os parâmetros w e b. da margem de decisão nos dados de treinamento (TAN et al., 2009).

No caso proposto, maximizar a margem d recai em um problema conhecido como otimização convexa, que pode ser solucionado, utilizando o método multiplicativo de Lagrange. Em alguns casos faz-se necessário que os multiplicadores de Lagrange sejam restritos a valores não negativos. Essa

transformação implica nas condições de Karush-Kuhn-Tucker (KKT) (TAN et al.,2009).

O caso SVM *hard-margin* até então considerado não permite a ocorrência de erros de classificação. No entanto, os dados reais não estão livres de ruídos. A Figura 3.7 mostra os pontos circulados para os quais não é possível obter uma margem separadora dos rótulos isenta de erros. Constatado esse problema, foi desenvolvida a abordagem *soft-margin*. Esse caso nomeado SVM linear não separável é definido pela situação em que é tolerável a ocorrência de erros nos dados de treinamento (TAN et al.,2009).

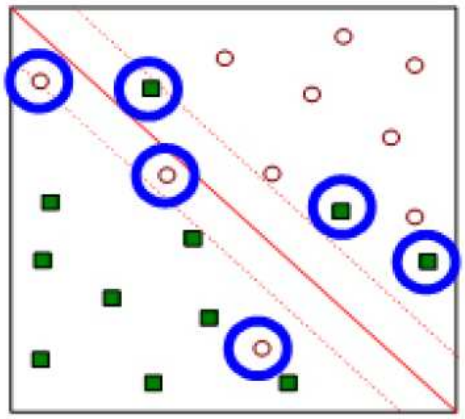

Figura 3.7 – SVM linear com dados não separáveis
Fonte: Adaptada de Tan et al. (2009).

Devido a sua eficiência em trabalhar com dados de alta dimensionalidade, tem sido reportada na literatura como uma técnica altamente robusta, muitas vezes comparada às redes neurais (SUNG e MUKKAMALA, 2003; DING e DUBCHAK, 2001).

Um algoritmo eficiente para implementação da técnica SVM seria o de Otimização Sequencial Mínima (*Sequential Minimal Optimization* - SMO) (PLATT, 1998), cuja utilização de memória é linear para realizar os treinamentos. Com isso, o SMO permite lidar com grande quantidade de dados para treinamento, além de prover a implementação da técnica SVM por meio do uso de polinômios ou núcleos gaussianos.

O algoritmo SMO surgiu da necessidade de um algoritmo SVM com implementação rápida, simples e capaz de tratar conjuntos de dados mais extensos. Além disso, tem capacidade de tratar conjuntos de dados esparsos, que possuem um número substancial de elementos com valor zero, pois empregam maior ênfase na etapa de avaliação da função de decisão. A otimização realizada no algoritmo SMO está na programação quadrática analítica, invés da abordagem numérica utilizada tradicionalmente nas implementações SVM.

O algoritmo SMO escolhe a resolução dos problemas de otimização, optando pelas menores otimizações possíveis em cada passo. Nos problemas de programação quadrática em SVM, a menor otimização possível envolve os multiplicadores de Lagrange, porque eles devem obedecer à restrição de igualdade linear. Em cada passo, o método SMO escolhe a otimização dos dois multiplicadores de *Lagrange*, buscando valores ótimos para eles e atualizando-os para refletir os novos valores ótimos. A vantagem está em utilizar um otimizador analítico, ao invés de ser chamada toda uma biblioteca de rotinas de programação quadrática. Além disso, não é necessário armazenamento de matrizes extras, o que permite manipular problemas com conjunto de treinamento volumoso (PLATT, 1998).

O algoritmo SMO consiste na sequência de três componentes principais:

- O método analítico, para solucionar os dois multiplicadores;
- Uma heurística para escolher quais multiplicadores otimizados;
- Um método para computar um dos parâmetros da margem de decisão.

No funcionamento do SMO, ocorre o cômputo das restrições de apenas dois multiplicadores de Lagrange escolhidos. Em seguida, soluciona-se o problema de maximização das restrições. (PLATT, 1998).

As escolhas de heurísticas podem ocorrer no primeiro ou segundo multiplicador de Lagrange. Na primeira escolha heurística, o algoritmo procura casos em que ocorre violação das condições de Karush-Kuhn-Tucker (KKT). Com isso, não é necessário percorrer completamente o conjunto de treinamento, diminuindo o tempo de processamento. O exemplo que viola as condições KKT é sujeito a uma otimização imediata. Uma vez descoberta à violação, um segundo multiplicador é escolhido, utilizando a segunda escolha, os dois multiplicadores são submetidos à otimização conjunta. O SVM é atualizado com os valores dos novos multiplicadores, procurando por outros casos de violação para o primeiro multiplicador (PLATT, 1998).

O algoritmo SMO permite que as condições KKT sejam cumpridas dentro de uma margem de tolerância (PLATT, 1998). Quanto maior a acurácia da saída, mais tempo levará para que o algoritmo chegue à convergência. Uma vez resolvidos os multiplicadores, é necessário determinar um valor limite para as margens b (Figura 3.6), em que as condições KKT são satisfeitas. Ele é computado, mantendo um *cache* para cada vetor de suporte e escolhe o erro para maximizar o passo.

3.3.1.5 Classificadores de Vizinho Mais Próximo

Um classificador de vizinho mais próximo (*Nearest Neighbor*- NN) tem por objetivo encontrar todos os exemplos de treinamento que sejam relativamente semelhantes aos atributos do exemplo de teste. Estes exemplos são conhecidos como vizinhos mais próximos e podem ser utilizados para determinar o rótulo da classe como exemplo de teste. Uma desvantagem óbvia de abordagem relaciona-se ao fato de que alguns registros de testes podem não ser classificados por não corresponderem a nenhum exemplo de treinamento.

Um classificador de vizinho mais próximo representa cada exemplo como um ponto dado em um espaço *d*-dimensional, onde *d* é o número de atributos.

Dado um exemplo de teste, calcula-se sua proximidade com o resto dos pontos de dados no conjunto de treinamento. Os vizinhos mais próximos *k* de um determinado exemplo z se referem aos k pontos mais próximos de z. Os classificadores de vizinho mais próximo fazem suas previsões baseadas em informações locais, sendo as decisões de classificação tomadas localmente, tornando-os suscetíveis a ruídos para valores pequenos de *k*.

A Figura 3.8 ilustra os 1, 2, 3 vizinhos mais próximos de um ponto de dados localizado no centro de cada círculo . O ponto de dado é classificado com base nos rótulos de classe dos seus vizinhos. No caso onde os vizinhos têm mais de um rótulo, atribui-se o ponto do dado a classe majoritária dos seus vizinhos mais próximos . Na Figura 3.8(a), o vizinho mais próximo 1 do ponto de dados é um exemplo negativo, atribuído a classe negativa. Na Figura 3.8(c), a vizinhança contém 2 exemplos positivos e 1 negativo. Então, atribui-se à classe majoritária, ou seja, atribui-se o ponto de dado a classe positiva (TAN et al., 2009).

Figura 3.8 – Os 1, 2 e 3 vizinhos mais próximos desta instância
Fonte: Adaptada de Tan et al. (2009).

Portanto, assim que a lista de vizinhos mais próximos for obtida pelo algoritmo de classificação, o exemplo de teste é classificado baseado na classe majoritária dos seus vizinhos mais próximos.

O algoritmo *Kstar* consiste em um classificador baseado em exemplos, onde a classe de um exemplo teste é baseada no treinamento de classes de exemplos similares a ele, determinado por alguma função de similaridade. O *Kstar*,

implementa o classificador de vizinho mais próximo, utilizando funções de distância baseadas na entropia (CLEARY e TRIGG, 1995). Classificadores baseados em exemplo tais como *Kstar* (FRANK et al., 1999), assumem que os exemplos similares terão classes similares.

3.3.1.6 Classificadores Baseado em Regras

Um classificador baseado em regras consiste naquele, cuja técnica para classificar registros baseia-se em um conjunto de regras “se...então”. Cada regra de classificação pode ser expressa da seguinte forma:

$$r_i: (\text{Condição}) \rightarrow y_i \qquad (3.11)$$

Na expressão 3.11, o lado esquerdo da regra é chamado de antecedente da regra ou pré-condição e contém um conjunto de testes de atributos. Enquanto o lado direito é chamado de consequência das regras, que contém a classe y_i prevista. Em suma, uma regra r cobre um registro x se a pré-condição de r corresponder aos atributos de x. As regras são diretamente obtidas a partir de algoritmos de cobertura seqüencial.

A expressividade de um conjunto de regras é praticamente equivalente àquela de uma árvore de decisão, porque uma árvore de decisão pode ser representada por um conjunto de regras completas e mutuamente excludentes. Tanto os classificadores de árvores de decisão quanto os baseados em regras criam partições retilíneas do espaço de atributos e atribuem uma classe a cada partição.

A ordenação das regras pode ser implementada regra a regra ou classe a classe. O esquema de ordenação baseado em regras garante que cada registro de teste seja classificado pela “melhor” regra que o cobrir. Uma potencial desvantagem seria que regras com menor prioridade são mais difíceis de interpretar, porque supõem a negação das regras que as precedem.

No esquema de ordenação baseado em classes, as regras que pertençam à mesma classe aparecem juntas no conjunto de regras R. As regras muitas vezes são ordenadas coletivamente com base na sua informação de classe. A ordenação relativa entre as regras da mesma classe não é importante; desde que uma das regras seja disparada, a classe será atribuída ao registro de teste. Isto torna a interpretação das regras um pouco mais fácil. No entanto, existe a possibilidade de uma regra de qualidade alta não ser percebida em favor de uma regra inferior que faça o prognóstico da classe de prioridade mais alta.

A maioria dos classificadores baseados em regras bem conhecidos como o C4.5rules, JRip (desenvolvido na linguagem de programação Java) e RIPPER (*Repeated Incremental Pruning to Produce Error Reduction* ou poda incremental repetida para produzir redução de erro, proposto por William W. Cohen (WITTEN e FRANK, 2005), empregam o esquema de ordenação baseada em classes.

Funcionamento do algoritmo *Ripper* ou *JRip* (desenvolvido na linguagem de programação *Java*) para um problema de duas classes. Escolha uma das classes como positiva e a outra como classe negativa (WITTEN e FRANK, 2005):

1. Aprenda regras para a classe positiva;
2. A classe negativa será a classe *default* (padrão);

Para um problema de várias classes:

1. Ordene as classes de acordo com a prevalência crescente da classe (iniciar com a classe que contém a menor quantidade de exemplos ou padrões);
2. Aprenda primeiro o conjunto de regras para a menor classe (menor número de exemplos), considerada como classe positiva; trate o restante como classes negativas;

3. Repita com a menor classe seguinte, tratando-a como classe positiva.

Construindo um conjunto de regras:

1. Use o algoritmo de cobertura seqüencial (TAN et al., 2009);
2. Encontre a melhor regra que cubra o conjunto atual de exemplos positivos;
3. Elimine tanto os exemplos positivos quanto negativos cobertos pela regra;
4. Cada vez que uma regra é colocada no conjunto de regras, calcule o novo comprimento da descrição;
5. Pare de adicionar novas regras quando o novo comprimento da descrição for *d* bit maior que o menor comprimento de descrição encontrado até então.

Crescendo uma regra:

1. Inicie com a regra vazia;
2. Adicione conjunções enquanto elas melhorarem o ganho de informação *FOIL* (*First-Order Induction Learning* ou Aprendizado por Indução de Primeira Ordem;
3. Pare quando a regra não cobrir mais exemplos negativos;
4. Pode a regra imediatamente usando o incremento da poda do erro reduzido (*reduced error pruning incremental*).

Medida para poda: $v = (p - n) / (p + n)$, em que: p = número de exemplos positivos cobertos pela regra no conjunto de validação; e n = número de exemplos negativos cobertos pela regra no conjunto de validação. Método de Poda: retire qualquer seqüência final de condições que maximize v.

Otimizando o conjunto de regras:

1. Para cada regra *r* no conjunto de regras *R*, considere 2 regras alternativas:
    - Regra de substituição (r*): cresça nova regra a partir do zero;
    - Regra de revisão (r'): adicione conjunções para estender *r*.
2. Compare a regra *r* com as regras r* e r'.
3. Escolha o conjunto de regras que minimize o Princípio da Descrição de Mínimo Comprimento (*Minimum Description Length Principle - MDL*), que representa o modelo da forma mais compacta possível com o máximo de informações dos dados
4. Repita geração e otimização de regras para o restante dos exemplos positivos.

Existe ainda um método para gerar um conjunto de regras a partir de uma árvore de decisão (vide seção 3.3.1.1). No princípio, cada caminho a partir do nodo raiz até o nodo folha de uma árvore de decisão pode ser expresso como uma regra de classificação. As condições de teste encontradas pelo caminho formam os conjuntos de antecedentes da regra, enquanto que o rótulo da classe no nodo folha é atribuído a sequência da regra. A Figura 3.9 mostra um exemplo de um conjunto de regras gerado a partir de uma árvore de decisão.

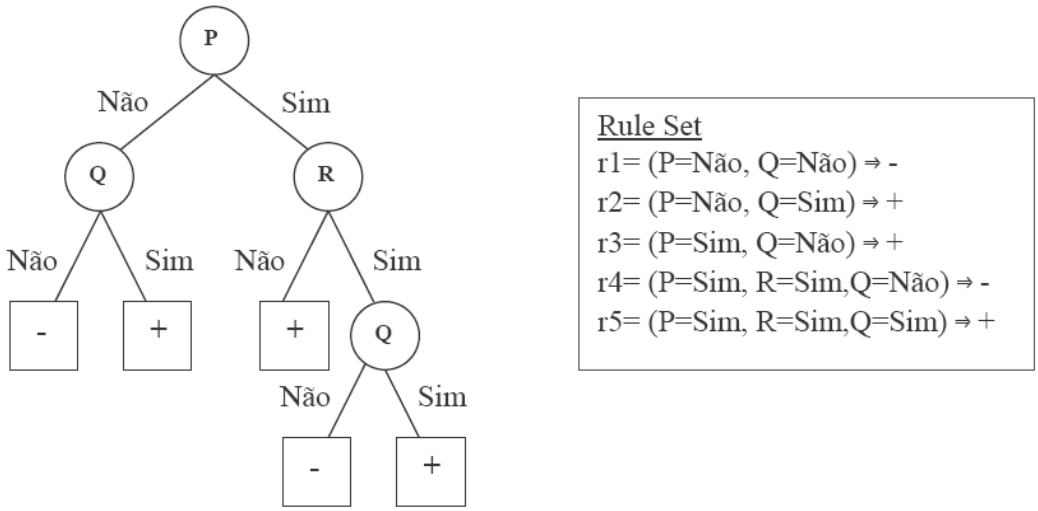

Figura 3.9 – Convertendo uma árvore de decisão em regras de classificação

Fonte: Adaptada de Tan et al. (2009).

3.3.2 Validação Cruzada

Trata-se de uma abordagem para validação dos modelos de classificação construídos pelos classificadores a partir da amostragem contida na base de classificação supervisionada. Uma ferramenta padrão em estatística conhecida como *cross-validation* (GEISSER, 1993) e que se apresenta como uma alternativa a subamostragem aleatória .

Neste método, o conjunto de dados disponível (são os exemplos ou amostras) é particionado aleatoriamente em um conjunto de treinamento e um conjunto de teste. O conjunto de treinamento é depois particionado em dois conjuntos disjuntos: o subconjunto de estimação e o subconjunto de validação.

O subconjunto de estimação é usado para estimação do modelo, isto é, treinamento da rede. E o subconjunto de validação ou teste é utilizado para avaliação do desempenho do modelo.

O particionamento pode ser feito a partir de cada registro, utilizado o mesmo número de vezes para treinamento e exatamente uma vez para teste. Ao

generalizar o método, a validação cruzada de N partes, segmenta os dados em N partições de tamanho igual. Assim, durante cada execução, uma das partições é escolhida para teste, enquanto as outras são utilizadas para treinamento. Este procedimento é repetido N vezes de modo que cada partição seja utilizada exatamente uma vez para teste.

Uma rodada do cross-validation envolve o particionamento de uma amostra de dados em subconjuntos complementares, executando a análise de um subconjunto (chamado de conjunto de treinamento), e validando a análise em outro subconjunto (chamado de conjunto de validação ou teste). Para reduzir a variabilidade, múltiplas rodadas da validação cruzada são executadas usando diferentes partições, em que o resultado da validação é a média das rodadas.

A validação cruzada é uma ferramenta padrão para análise, sendo um importante recurso para ajudar a desenvolver e ajustar os modelos de mineração de dados. A validação cruzada é utilizada depois de se criar uma estrutura de mineração e os modelos de mineração relacionados para assegurar a validade do modelo. Sendo utilizada em aplicações, onde o objetivo é a predição para estimar o quão correto um modelo preditivo deverá ser executado na prática.

3.3.3 Análise Estatística de Modelos de Classificação

Para avaliar qual o classificador, que representa o modelo de classificação, com melhor desempenho na tarefa de classificação de um determinado conjunto de instâncias, torna-se necessário utilizar medidas para estimar a qualidade do classificador. Estas medidas são extraídas de funções estatísticas calculadas a partir dos dados de treinamento e teste. Estas funções estatísticas são apresentadas nas seções subsequentes.

3.3.3.1 Métricas para Avaliação de Desempenho

Avaliação de desempenho de um modelo de classificação, baseia-se na contagem de registros de teste corretamente e incorretamente previstos pelo modelo. Estas contagens são tabuladas em uma tabela conhecida como matriz de confusão. A Tabela 3.1 mostra a matriz de confusão para um problema de classificação binária.

Tabela 3.1 – Matriz de confusão para um problema de 2 classes

		Classe Prevista	
		Classe = 1	Classe = 0
Classe Real	Classe = 1	e_{11}	e_{10}
	Classe = 0	e_{01}	e_{00}

Fonte: Adaptada de Tan et al. (2009).

Cada elemento eij da matriz de confusão indica o número de registros da classe 0 previsto incorretamente como classe 1. Enquanto eji indica o número de registros da classe 1 previstos incorretamente como classe 0. Assim, com base nos elementos da matriz de confusão, o número total de previsões corretas feitas pelo modelo é calculado por: (e_{11}+e_{00}), enquanto o número total de previsões incorretas é calculado por: (e_{10}+e_{01}).

Embora uma matriz de confusão forneça as informações necessárias para determinar o quanto um modelo de classificação é bem executado, resumir estas informações em um único número tornaria mais conveniente para realizar uma comparação de desempenho entre diferentes modelos. Esta comparação pode ser obtida a partir de métricas de desempenho como a exatidão (*accuracy*) e a taxa de erro (TAN et al., 2009). A precisão e a taxa de erro são obtidas a partir da Equação 3.12 e Equação 3.13 respectivamente, sendo definidas da seguinte forma:

$$\text{Exatidão} = \frac{\text{Número de previsões corretas}}{\text{Número total de previsões}} = \frac{e_{11} + e_{00}}{e_{11} + e_{10} + e_{01} + e_{00}} \quad (3.12)$$

$$\text{Taxa de erro} = \frac{\text{Número de previsões incorretas}}{\text{Número total de previsões}} = \frac{e_{10} + e_{01}}{e_{11} + e_{10} + e_{01} + e_{00}} \quad (3.13)$$

A maioria dos algoritmos de classificação procura modelos que atinjam a maior exatidão ou, equivalentemente, a menor taxa de erro quando aplicados ao conjunto de testes.

A precisão consiste em outra métrica definida em função do valor da predição positiva (número de casos positivos por total de casos cobertos). Casos positivos (*True Positives* - TP): consistem nos valores classificados verdadeiramente positivos. Enquanto os casos falsos positivos (*False Positives* - FP) consistem nos dados classificados erroneamente como positivos pelo classificador. O valor da precisão é obtido por: precisão = TP/ (TP+FP). A Tabela 3.2 apresenta a matriz de confusão em termos de casos positivos e negativos por classe.

Tabela 3.2 – Matriz de confusão: casos positivos e negativos

	Predição Positiva	Predição Negativa
Classe Positiva	Positivo Verdadeiro (TP)	Falso Negativo (FN)
Classe Negativa	Falso Positivo (FP)	Negativo Verdadeiro (TN)

Fonte: Adaptada de Tan et al. (2009).

A taxa de verdadeiros positivos corresponde ao número de casos positivos que são verdadeiramente positivos, ou seja a taxa de acerto ou taxa de verdadeiros positivos é dada por: positivos = TP/ (TP+FN). Enquanto a taxa de falsos positivos corresponde ao número de casos considerados alarmes falsos ou taxa de falsos positivos, sendo dada por: taxa de falsos positivos = FP/ (FP+TN) (FERREIRA, 2006).

A cobertura (*recall*) corresponde ao valor da cobertura de casos para uma determinada classe, isto é, o número de instâncias corretamente classificadas em relação ao total de exemplos pertencentes a esta classe. Calcula-se de forma idêntica a taxa de verdadeiros positivos, sendo dada por: cobertura = TP/ (TP+FN). Quanto maior a cobertura, maior são os acertos do classificador para uma determinada classe. Um classificador com valor alto de cobertura para uma determinada classe, identifica um maior número de exemplos para esta classe (FERREIRA, 2006).

A *F-measure* corresponde à média harmônica entre a precisão e a cobertura. Mede a capacidade de generalização do modelo gerado, permitindo verificar se durante o treinamento este assimilou do conjunto de treinamento características significativas que permitem um bom desempenho em outros conjuntos de dados, ou se concentrou em peculiaridades (FERREIRA, 2006). A F-measure é descrita pela Equação 3.14:

$$F - measure = 2 \cdot \frac{precisão \cdot cobertura}{precisão + cobertura} \tag{3.14}$$

3.3.3.2 Estatística ou Coeficiente Kappa

O coeficiente ou estatística Kappa (*k*) consiste em uma medida estatística da concordância real, (indicado pelos elementos diagonais da matriz de confusão de cada classificador), menos a concordância pelo acaso (indicado pelo produto total da linha e coluna). Sendo assim, o cálculo da estimativa do coeficiente Kappa (k) é descrito pela Equação 3.15:

$$K = \frac{P_0 - P_C}{1 - Pc} \tag{3.15}$$

Na Equação 3.14 P_0 representa a concordância observada, isto é, a proporção de observações corretamente classificadas. Enquanto P_c representa a concordância esperada, isto é, a proporção esperada de acerto por acaso.
Os valores de Kappa são interpretados como o máximo de 1 quando a concordância é considerada perfeita e 0 quando a concordância não é melhor que o acaso. A Tabela 3.3 apresenta os valores de Kappa a serem interpretados (SHESKIN, 2003).

O coeficiente de concordância *Kappa* deve ser empregado para determinação da variação entre classificadores, que classificam separadamente uma amostra empregando a mesma escala de categoria.

Tabela 3.3 – Interpretação dos valores de Kappa

Valor de kappa (*k*)	Interpretação
< 0	Nenhuma concordância
0 – 0,20	Leve concordância
0,21 – 0,40	Concordância regular
0,41 – 0,60	Concordância moderada
0,61 – 0,80	Concordância substancial
0,81 – 1	Concordância quase perfeita

Fonte: Sheskin (2003).

3.3.3.3 Erro Absoluto Médio

O erro absoluto médio (*Mean Absolute Error – MAE*) consiste em uma medida que indica a média do afastamento de todos os valores fornecidos pelos classificadores e o seu real valor. Pode ser calculada utilizando a Equação 3.16, na qual *n* representa o número de amostras, x_i *representa* o valor fornecido pelo classificador (previsto) para a i-ésima amostra e $\bar{x}$ representa a média dos valores de todas as amostras (FERREIRA, 2006).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - \bar{x}\right| \quad (3.16)$$

3.3.3.4 Raiz do Erro Quadrático Médio

A raiz do erro quadrático médio (*Root Mean Squared Error – RMSE*) consiste em uma medida utilizada para estimar o sucesso da predição numérica. Este valor é calculado pela média da raiz quadrada da diferença entre o valor calculado e o valor correto, fornecendo o valor do erro entre os valores atuais e os valores preditos. Pode ser calculada a partir da Equação 3.17, na qual *n* representa o número de amostras, x_i representa o valor fornecido pelo classificador (previsto) para a i-ésima amostra e $\bar{x}$ representa a média dos valores de todas as amostras (FERREIRA, 2006).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2} \quad (3.17)$$

3.3.3.5 Erro Absoluto Total Normalizado

O erro total absoluto (*Relative Absolute Error – RAE*) consiste em mais uma das medidas utilizada para estimar a qualidade de um classificador, em que valores mais baixos significam maior precisão do modelo, com o valor próximo de zero temos o modelo estatisticamente perfeito. Pode ser calculada utilizando a Equação 3.18, na qual *n* representa o número de amostras, x_i representa o valor fornecido pelo classificador para a i-ésima amostra, $\bar{x}$ representa a média dos valores de todas as amostras e $\overset{*}{x}_i$ representa o erro absoluto total do classificador normalizado (dividido) pelo erro absoluto total da medida amostrada ou seja, representa o valor correto, que deve ser fornecido pelo classificador para a amostra em questão (FERREIRA, 2006).

$$RAE = \frac{\sum_{i=1}^{n} \left| x_i - x_i^* \right|}{\sum_{i=1}^{n} \left| x_i - \bar{x} \right|} \tag{3.18}$$

3.3.3.6 Raiz do Erro Quadrático Relativo

A raiz do erro quadrático relativo (Root *Relative Squared Error – RRSE*) reduz o quadrado do erro relativo na mesma dimensão da quantidade sendo predita, incluindo raiz quadrada. Assim como a raiz quadrada do erro significativo (root mean-squared error), este exagera nos casos em que o erro da predição foi significativamente maior que o erro significativo. Pode ser calculada utilizando a Equação 3.19, na qual *n* é o número de amostras, x_i é o valor fornecido pelo classificador para a i-ésima amostra, $\bar{x}$ é a média dos valores de todas as amostras e x_i^* representa o valor correto, que deve ser fornecido pelo classificador para a amostra em questão (FERREIRA, 2006).

$$RRSE = \sqrt{\frac{\sum_{i=1}^{n} \left(x_i - x_i^* \right)^2}{\sum_{i=1}^{n} \left(x_i - \bar{x} \right)^2}} \tag{3.19}$$

3.4 Trabalhos Correlatos

A exemplo do que ocorre em diversas áreas, a motivação na utilização de técnicas de mineração de dados para a construção de um modelo de classificação preditiva ou modelo preditivo surge em função do interesse por métodos diagnósticos.

O diagnóstico e predição de estados do satélite têm o propósito de auxiliar especialistas no processo de decisão em relação às ações nos planos de voo e, assim, garantir a integridade do satélite. Além disto, um modelo preditivo baseado em conceitos de mineração de dados, ou seja, um modelamento

matemático, se apresenta como uma alternativa ao uso dos simuladores de custo bastante elevado para segurança no planejamento das operações de controle de satélite.

No entanto, a seleção de uma técnica de mineração de dados mais adequada para construção do modelo de preditivo para realização de diagnósticos e predições depende essencialmente da tarefa específica a ser executada, dos dados disponíveis para análise e dos critérios de avaliação das técnicas (vide seção 3.1).

A seleção de uma técnica de mineração de dados poderia basear-se também em uma análise da literatura associada ao emprego de mineração de dados para a realização de predição de estados de satélites. Porém, os trabalhos correlatos voltados à predição de estados de satélites fazem referência somente ao uso de simuladores (vide seção 2.4).

Todavia, para todo simulador, que tem como característica ser um sistema reativo, ou seja, baseado em evento (estímulo) e resposta (PRESSMAN, 2006), torna-se necessário a construção por especialistas da modelagem de um ou mais subsistemas e da geração da base de conhecimento do satélite. Construir todo o modelamento considerando todas as regras e restrições, torna complexo e oneroso o desenvolvimento de um simulador.

A complexidade associada ao desenvolvimento de simuladores pode ser observada nas arquiteturas de simuladores desenvolvidos ou em desenvolvimento, encontradas na literatura por Barreto (2010). A Figura 3.10 e Figura 3.11 mostram, como exemplo, as arquiteturas de dois simuladores em desenvolvimento: o *Cosmic Simulator* (CSIM) (HOMEM et al., 2006) e o Simulador *LISA Pathfinder* (DELHAISE e BRU, 2006).

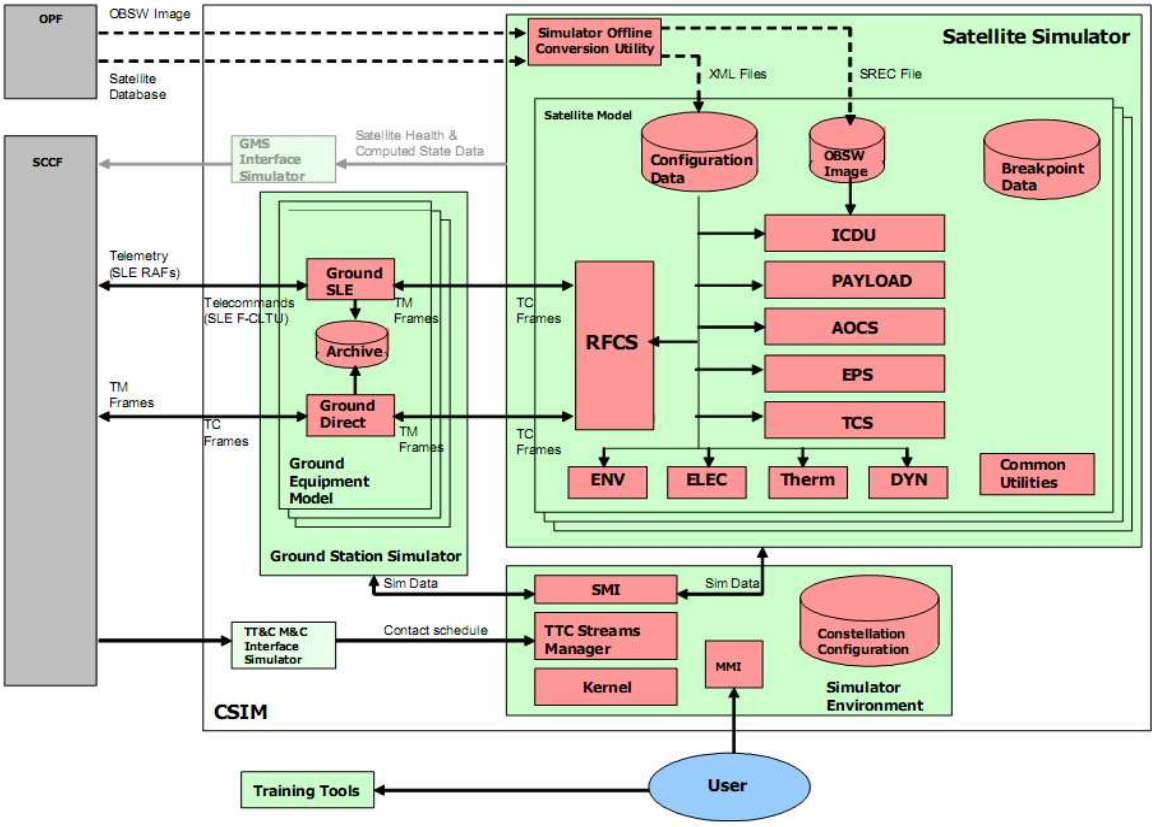

Figura 3.10 – Arquitetura do Cosmic Simulator (CSIM)

Fonte: Homem et al. (2006).

O Cosmic Simulator (CSIM) foi projetado para prover um ambiente completo para validação do segmento solo responsável pelo controle da constelação de satélites GALILEO.

O Simulador LISA (Laser Interferometer Space Antenna) Pathfinder (DELHAISE e BRU, 2006) está sendo desenvolvido para a missão LISA, de detecção de ondas gravitacionais no espaço. O objetivo do simulador consiste em validar e testar o segmento solo, provendo telemetrias de forma realística, em resposta a telecomandos enviados.

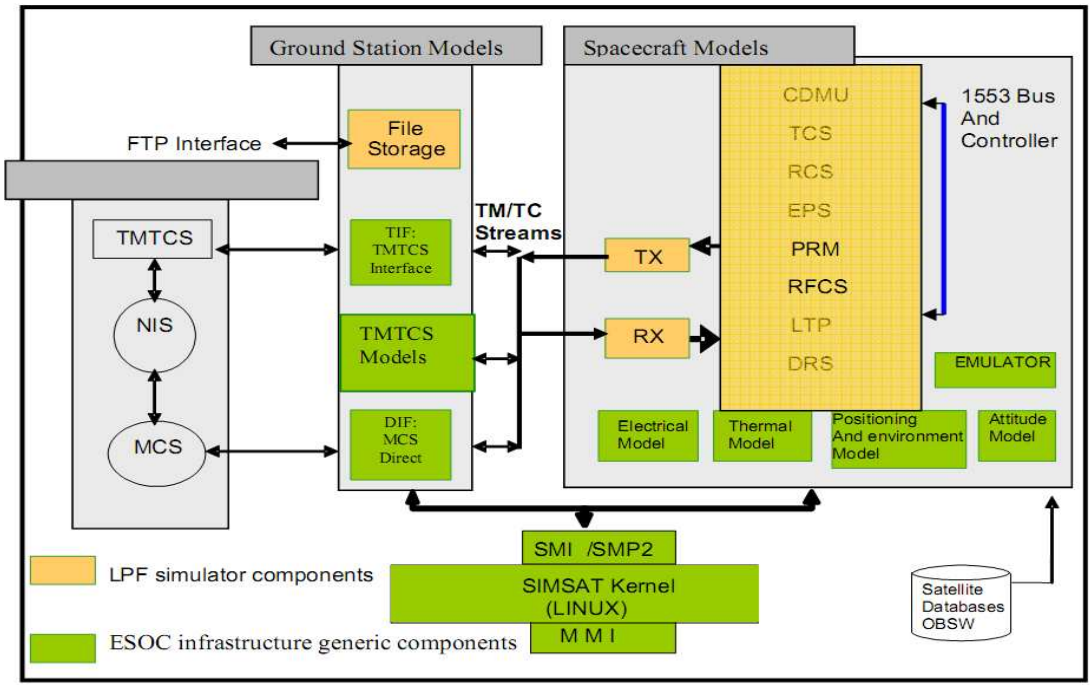

Figura 3.11 – Arquitetura do simulador Lisa Pathfinder (LPF)

Fonte: Delhaise e Bru (2006).

Outros dados referentes ao desenvolvimento de simuladores pesquisados na literatura por Barreto (2010) são mostrados na Tabela 3.4.

Tabela 3.4 – Dados sobre o desenvolvimento de simuladores de satélites.

Simulador	Autor/Empresa	Inicio/Uso	Status	Satélite	Custo
CSIM	SciSys,Veja, Critical, SkyTek	2011	Dois satélites lançados para validação	Constelação Galileo	Mais de 3.4 bilhões de euros
XMM (*X-ray Multi Mirror*) (CÔME e IRVINE, 1998).	ESA/ESOC, Vega	1999	Concluído	XMM	689 milhões de euros
LPF	ESOC, EADS Astrium	2011	Não finalizado	LPF	240 milhões de dólares

Fonte: Adaptada de Barreto (2010).

No INPE, o desenvolvimento de simuladores vem sendo realizado pelo grupo de simulação da Engenharia e Tecnologia Espacial (ETE) (AMBROSIO et al., 2006; BARRETO et al., 2010), com o interesse de fortalecer a arte na área de simulação do INPE.

Em função do alto custo dada a complexidade inerente ao desenvolvimento de um simulador, propõe-se uma alternativa ao uso de simuladores para a realização de predição de estados de satélites. A alternativa proposta consiste em uma ferramenta de software, que utiliza análise de dados usando técnicas de mineração de dados para a construção de sistema de previsão de estados futuros de satélites, desde que sejam fornecidas as telemetrias do respectivo satélite, bem como o modelo que descreve o comportamento do sistema.

No próximo capítulo o projeto desta ferramenta de software para validação será apresentado. A ferramenta constitui uma parte relevante da estratégia para validação de um plano de operação de voo a partir de predições de estados dos satélites.

4 ESTRATÉGIA PARA VALIDAR PLANO DE OPERAÇÃO DE VOO

No INPE, a perspectiva de múltiplos lançamentos e o aumento da demanda de satélites em operação, poderia tornar inviável uma validação criteriosa dos planos de voo que controlam estes satélites. Sendo assim, propõe-se uma solução para avançar na melhoria da segurança no planejamento das operações de rotina que controlam os satélites em órbita. Trata-se de uma estratégia para a validação de plano de operação de voo de satélites do INPE, tendo como parte relevante uma ferramenta concebida com o propósito de gerar diagnóstico e realizar predições de estados do satélite em função das ações contidas no plano.

O objetivo da estratégia consiste em empregar os conceitos de mineração de dados na análise dos dados para prever estados do satélite, auxiliando os especialistas na validação dos planos de operação de voo, que podem ser gerados manualmente ou de forma automática por um planejador. E, a partir desta melhoria na segurança no planejamento das operações também garantir a integridade dos satélites em órbita. Assim, com base em um modelamento matemático contribuir na direção de oferecer uma alternativa aos onerosos simuladores na tarefa de realizar predições de estados operacionais de satélites.

Para contextualizar o trabalho, será apresentada uma visão geral da estratégia, que inclui a arquitetura concebida para validação de um plano de operação de voo. Em seguida, a arquitetura para geração de diagnóstico de estados do satélite, que se baseia em análise matemática para realizar a tarefa de predição dos estados do satélite a partir de estados simulados ou de telemetrias recebidas pela estação terrena.

4.1 Arquitetura para Validação

Desafios crescentes em relação à segurança e confiabilidade nos projetos espaciais desenvolvidos pelas principais agências espaciais internacionais, que visam maior autonomia nas operações de controle de veículos espaciais vêm sendo apresentados por diversos artigos recentes na área espacial. Em busca de soluções para estes desafios, as principais agências internacionais fazem referência ao desenvolvimento de componentes de hardware e software para a realização de testes de verificação e validação (vide seção 2.4).

Em conformidade com as recomendações das agências e visando alcançar este avanço em confiabilidade, uma estratégia para validar planos de operação de voo está sendo proposta. A estratégia de validação consiste em uma arquitetura composta por diversos componentes de software para verificação e validação de um plano de operação gerado automaticamente, antes da execução real propriamente dita ou durante uma operação real (Figura 4.1). A estratégia foi concebida com o objetivo de avaliar a partir dos estados simulados ou telemetrias vindas do satélite em operação, o impacto do plano sobre o estado operacional do satélite. A estratégia baseia-se em técnicas adequadas de garantia de sistemas espaciais (BLANQUART et al., 2004).

A Figura 4.1 ilustra dois módulos pontilhados dentro da estratégia, que indicam duas seqüencias distintas de execução: a primeira ocorre a partir de uma execução off-line do plano gerado pelo primeiro componente de software, o planejador de operações, cada ação do plano é executada e uma simulação do comportamento do satélite é realizada pelo segundo componente de software, um simulador de satélite (vide seção 2.4). A saída fornecida pelo simulador consiste em um conjunto de parâmetros e telemetrias, que formam o estado operacional do satélite simulado, resultante da execução das ações do plano gerado pelo planejador de operações.

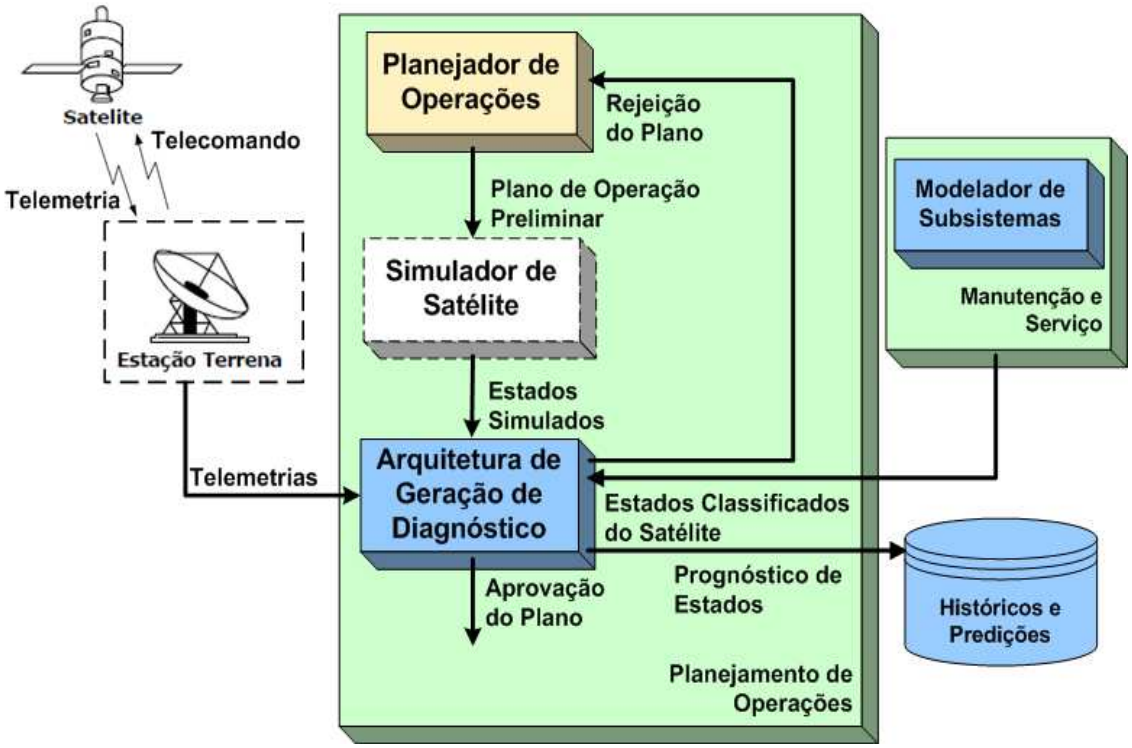

Figura 4.1 – Arquitetura de validação do plano de operação de voo.

Uma das sequências definida pela arquitetura apresentada na Figura 4.1, mostra que cada saída do simulador, ou seja, cada estado de operação gerado pelo simulador se torna dado de entrada para o terceiro componente de software da estratégia: a arquitetura de geração de diagnóstico, que a partir da ferramenta de validação denominada gerador de diagnóstico realiza diagnóstico e predição dos estados do satélite, por meio da classificação dos parâmetros e telemetrias simuladas, indicando a evolução do estado do satélite em função das ações do plano e sugerindo a aprovação ou rejeição do plano.

Outra sequência de execução prevista para a estratégia acontece durante uma operação real do satélite, a partir de telemetrias recebidas pela estação terrena e enviadas como dados de entrada para a arquitetura de geração de diagnóstico, que conforme a sequência anterior, utiliza a ferramenta de validação gerador de diagnóstico para realizar diagnóstico e predição dos estados do satélite, por meio da classificação dessas telemetrias, indicando a

evolução do estado do satélite em função das ações do plano e sugerindo a aprovação ou rejeição do plano.
No caso em que a ferramenta de validação gerador de diagnóstico realiza o diagnóstico do estado do satélite a partir de telemetrias vindas de um satélite em operação, a classificação dos estados ocorre da mesma forma que no conjunto de dados fornecidos pelo simulador, visto que as telemetrias constituem um subconjunto de parâmetros previstos igualmente para cada subsistema do satélite e o simulador deste subsistema.

4.2 Ferramenta de Validação de Plano de Operação de Voo Baseada em Modelo Preditivo

Como parte relevante da estratégia de validação (vide Figura 4.1), o componente nomeado arquitetura de geração de diagnóstico, contém a ferramenta gerador de diagnóstico, concebida para fornecer predição de estados futuros do satélite a partir de parâmetros e telemetrias críticas afetadas diretamente pelas ações do plano (vide seção 4.2), indicando o impacto do plano sobre o nível de segurança operacional do satélite e, também como o estado geral do satélite deve evoluir, sugerindo a aprovação ou rejeição do plano.
A ferramenta de validação que está sendo proposta consiste em um componente de software capaz de analisar grande quantidade de dados fornecidos por um subsistema crítico para manutenção do satélite em órbita e de realizar diagnósticos e predições e sobre estados do satélite, utilizando para isto técnicas de mineração de dados.

A ideia de se projetar uma ferramenta de validação baseada em técnicas de mineração de dados surgiu da intenção de trazer para a área espacial, a exemplo de outras áreas, uma alternativa baseada em modelamento matemático para a realização de diagnósticos de estados do satélite, em relação às propostas atuais baseadas em simuladores (vide seção 3.4). A mineração de dados usa técnicas para extrair conhecimento sobre os dados

para derivar padrões e tendências existentes nos dados. Normalmente, esses padrões não podem ser descobertos com a análise tradicional de dados pelo fato das relações serem muito complexas ou por haver muitos dados (vide seção 3.1).

Uma vez que a ferramenta baseia-se em conceitos de mineração de dados, os estados simulados ou dados reais do satélite são utilizados para gerar a base de dados de classificação supervisionada dentro do processo de descoberta de conhecimento em base de dados (KDD) (vide seção 3.1), cujas fases dentro deste estudo tiveram como objetivo construir o modelo de predição mais adequado para classificação dos estados operacionais de um satélite.

Para representar o processo de descoberta de conhecimento de uma base de classificação supervisionada de um satélite, foi projetada uma arquitetura para a geração de diagnóstico (Figura 4.2).

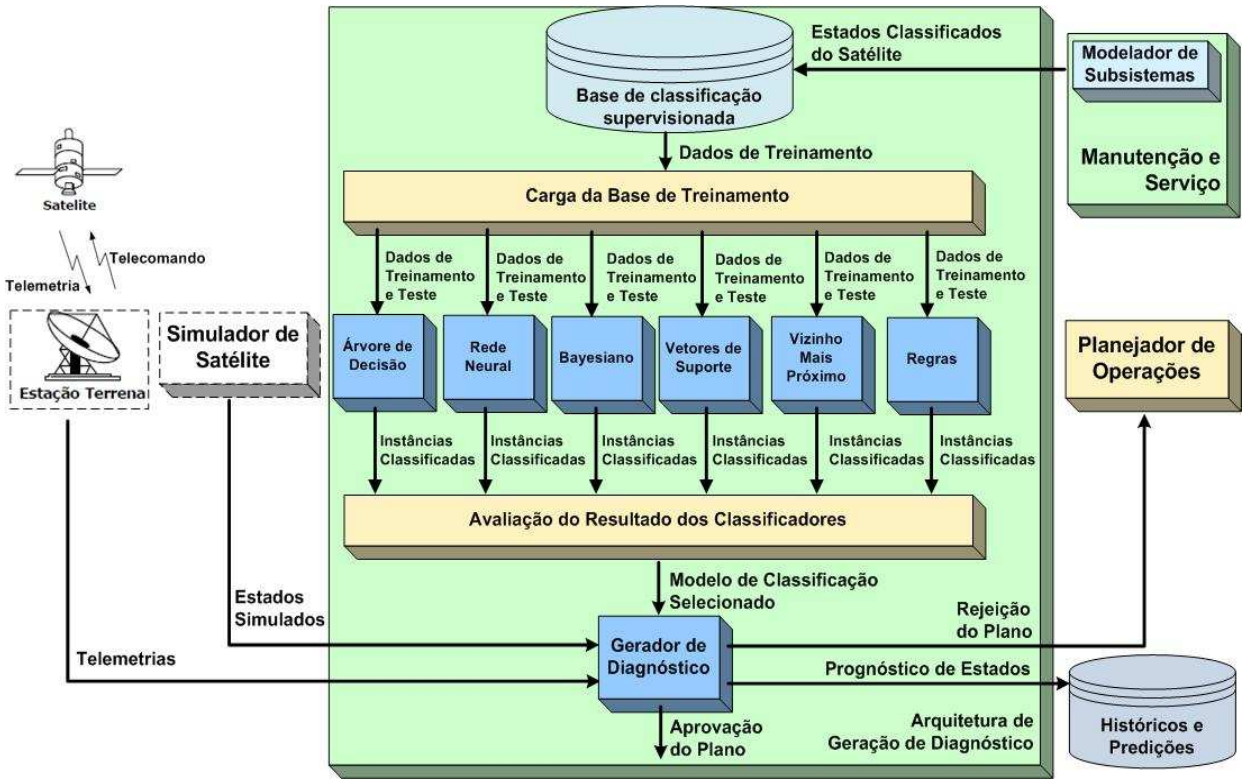

Figura 4.2 – Arquitetura para geração de diagnóstico de estado do satélite.

A arquitetura da Figura 4.2 representa os componentes de software e a sequência entre eles, que compõem as fases do processo de construção do modelo de classificação preditiva a ser selecionado. A definição das fases está em conformidade com as fases do processo de descoberta de conhecimento em mineração de dados (vide seção 3.1). A seguir, são descritas as atividades associadas a cada uma dessas fases.

1. Fase de construção da base de classificação supervisionada: geração da base de dados, contendo amostras de treinamento e argumento categórico com capacidade de julgamento. As amostras serão utilizadas na busca por padrões e na geração de conhecimento novo. Esta fase envolve as seguintes atividades:

    - Seleção dos atributos, ou seja, seleção e análise das telemetrias e parâmetros que compõem o modelo do subsistema de suprimento de energia, determinando-se também o atributo rótulo de classe ou variável alvo ou variável categórica (vide seção 3.1), que deve conter um dos estados de operação do satélite, discretizados e definidos em níveis de segurança pelo especialista;

    - Preparação dos dados, que se caracteriza pelo tratamento e preparação dos dados para uso pelos algoritmos. Nesta etapa deve-se identificar e retirar valores inválidos, inconsistentes ou redundantes;

    - Transformação dos dados com a utilização, quando necessário, de alguma transformação linear ou mesmo não linear nos dados, de forma a encontrar aqueles mais relevantes para o problema em estudo. Nesta etapa geralmente são aplicadas técnicas de redução de dimensionalidade, ou seja, redução do número de atributos utilizados.

2. Fase de construção dos modelos de classificação preditiva:

    - Fase de mineração com a construção de diferentes modelos de classificação preditiva de estados do satélite a partir da base de classificação supervisionada gerada na fase anterior e aplicação dos algoritmos de classificação. Com base na literatura foram selecionadas 6 técnicas distintas de classificação (vide seção 3.3.3.1 a 3.3.3.6), adequadas à classificação binária. São elas: árvore de decisão, rede neural, bayesiano, máquina de vetor de suporte, vizinho mais próximo e regras.

3. Fase de avaliação do resultado dos classificadores:

    - Fase de interpretação e avaliação dos resultados fundamentada em um pacote estatístico, definido para avaliar classificadores. Trata-se de um conjunto de funções estatísticas (vide seção 3.3.3), com a finalidade de realizar uma análise estatística dos modelos de classificação construídos pelos algoritmos classificadores. Esta fase visa selecionar o modelo preditivo que apresenta maior acurácia na classificação dos estados do satélite em amostras desconhecidas (dados de teste).

4.3 Estratégia para Atualização da Ferramenta de Validação

A arquitetura da ferramenta de validação de plano de operação de voo, gerador de diagnóstico, apresentada na Figura 4.2 (vide seção 4.3), mostra que o modelo preditivo foi selecionado a partir da avaliação da acurácia do modelo de classificação construído em relação à base de classificação supervisionada.

Esta seleção do modelo preditivo adequado retrata o caráter dinâmico do modelo, que está associado ao caráter dinâmico da base supervisionada, que pode ser alterada, por exemplo, em função da necessidade de adequá-la para

diferentes satélites ou para atualização do modelo do satélite em operação, quando degradado pelo uso.

No entanto, as alterações para aumentar, diversificar ou até substituir a base supervisionada devem ser realizadas manualmente no protótipo da ferramenta, desenvolvido neste trabalho. O ideal para um dinamismo maior, seria implementar uma base configurável para diversos modelos de satélite, com a inclusão de um módulo que importasse o modelo de subsistema do satélite para assim, gerar a base supervisionada.

O próximo capítulo apresenta a implementação do protótipo baseado na arquitetura para geração de diagnóstico de estado do satélite, apresentada na Figura 4.2. O capítulo inicia apresentando a fase de construção da base de classificação supervisionada, com a implementação do aplicativo desenvolvido para gerar o conjunto de dados da base de classificação supervisionada para treinamento, baseado em um estudo de caso do subsistema de energia do satélite virtual XSAT (TOMINAGA et al., 2009). Apresenta ainda, a fase de mineração para a construção dos modelos de classificação preditiva, envolvendo a configuração de parâmetros para a execução dos algoritmos classificadores selecionados e a abordagem adotada para validação dos modelos construídos a partir dos dados contidos na base supervisionada.

5 CONSTRUÇÃO DA BASE DE CLASSIFICAÇÃO SUPERVISIONADA E DOS MODELOS DE CLASSIFICAÇÃO PREDITIVA

Neste capítulo será apresentada a forma como foi implementado o protótipo baseado na arquitetura para geração de diagnóstico de estado do satélite (vide seção 4.2). São descritas as atividades desenvolvidas na fase de construção da base de classificação supervisionada, com a implementação do aplicativo para gerar o conjunto de dados de classificação supervisionada para treinamento. Os dados foram gerados a partir de um modelo teórico de um sistema de energia do satélite virtual XSAT, utilizado como estudo de caso.

Em seguida, apresenta-se a fase de construção dos modelos de classificação preditiva, ou seja a fase de mineração propriamente dita, com a seleção da abordagem usada para validação dos modelos construídos a partir dos dados contidos na base supervisionada. Também são apresentados valores atribuídos aos parâmetros de configuração para a execução dos algoritmos classificadores selecionados para a construção dos modelos de classificação preditiva.

5.1 Estudo de Caso: Sistema de Suprimento de Energia do Satélite Virtual (XSAT)

Como estudo de caso para geração dos estados simulados foi utilizado um modelo simplificado do subsistema de suprimento de energia do satélite virtual (XSAT) (TOMINAGA et al., 2009).

Inicialmente foi feita a implementação do modelo do subsistema de suprimento de energia do satélite XSAT, com a simulação de várias órbitas e passagens sobre as estações terrenas, que incluiu a simulação das tarefas de recepção de telemetrias e envio de telecomandos (vide seção 2.2) durante as passagens. Estes telecomandos com a função de ligar e desligar as cargas úteis, enviados durante os períodos de visada simulam as ações de um plano de operação de

voo, atuando sobre dados críticos para manutenção da vida útil do satélite, contidos no subsistema de suprimento de energia.

A simulação deste modelo utilizado como estudo de caso, resultou na geração dos estados simulados, ou seja um conjunto composto por telemetrias, parâmetros e limites operacionais do modelo do subsistema de suprimento de energia do satélite XSAT. Uma descrição destas telemetrias, destes parâmetros e dos respectivos valores operacionais é apresentada nas Tabelas 5.1, 5.2, 5.3, 5.4 e 5.5.

Tabela 5.1 – Resumo das operações de missão do satélite virtual XSAT

Carga útil	Descrição	Dados	Dados recebidos na estação	Critério de operação	Consumo de Energia
PL1	Câmera	Imagens de satélite para monitorar superfície terrestre	Estação de recepção de imagens	Sobre a estação, na região iluminada ou durante eclipse durante calibração	PPL1 ON = 800 W OFF = 100 W
PL2	Subsistema de Coleta de Dados	Dados ambientais aquisição da plataforma de coleta de dados	Estação de coleta de dados	Sobre a estação ou em modo contínuo, na região iluminada e eclipse	PPL2 ON = 15 W OFF = 5 W

Fonte: Adaptada de Tominaga et al. (2009).

A Tabela 5.1 apresenta uma descrição das cargas-úteis, os critérios de operação e o consumo de energia das cargas-úteis durante a operação, que constituem as operações de missão do satélite virtual XSAT, nas quais se baseia o plano de operação de voo.

Tabela 5.2 – Parâmetros e telemetrias do sistema de suprimento de energia -XSAT

Telemetria	Descrição	Parâmetro	Descrição
SAG	*Solar Array Generator* – Painel solar	PAV	Potência disponível para o satélite
PSAG	Energia fornecida pelo SAG	CBAT	Capacitância da Bateria
VBAT	Voltagem da bateria	BAT	Bateria
QBAT	Carga da bateria	DOD	Profundidade de descarga da bateria
IBAT	Corrente de carga na bateria		

Fonte: Adaptada de Tominaga et al. (2009).

A Tabela 5.2 apresenta uma descrição dos parâmetros e telemetrias (vide seção 2.1), que compõem o subsistema de suprimento de energia do satélite virtual XSAT.

Tabela 5.3 – Valores de potência fornecida e consumida no XSAT

Status a bordo		Descrição	Potência fornecida (W)	Potência consumida (W)
SAG	SUN	*Sunlight* – Região iluminada da órbita	1600	0
	ECL	*Eclipse* – Região escura da órbita	0	0
PL1	ON	PL1 Em operação	0	800
	OFF	PL1 Em espera	0	100
PL2	ON	PL2 Em operação	0	15
	OFF	PL2 Em espera	0	5
SM	-	Módulo de serviço	0	780

Fonte: Adaptada de Tominaga et al. (2009).

A Tabela 5.3 apresenta o valor da potência fornecida para a bateria pelo painel solar na região iluminada da órbita e os valores de potência consumida pelas

cargas-úteis e pelo módulo de serviço durante a operação do satélite virtual XSAT.

Tabela 5.4 – Valores de potência em cada modo de operação do XSAT

Plano	Status a bordo			Potência (W)		
	SAG	PL1	PL2	Consumida	Fornecida	Disponível
A	SUN	ON	ON	1595	1600	5
B	SUN	ON	OFF	805	1600	795
C	SUN	OFF	ON	115	1600	1485
D	SUN	OFF	OFF	885	1600	715
E	ECL	ON	ON	1595	0	-1595
F	ECL	ON	OFF	1585	0	-1585
G	ECL	OFF	ON	895	0	-895
H	ECL	OFF	OFF	885	0	-885

Fonte: Adaptada de Tominaga et al. (2009).

A Tabela 5.4 apresenta o valor da potência total consumida, fornecida e disponível em cada modo de operação do satélite virtual XSAT, definido pelo plano de operação de voo.

Tabela 5.5 – Critério para controle da profundidade de descarga da bateria do XSAT

DOD (%)	Estado do DOD	Estado de operação
< 15	BAIXO	SAFE
15 ~ 20	ALTO	UNSAFE
> 20	EXTREMO	FORBIDDEN

Fonte: Adaptada de Tominaga et al. (2009).

A Tabela 5.5 apresenta a classificação em níveis de segurança do estado de operação do satélite virtual XSAT, definida pelo especialista em função do valor da profundidade de descarga da bateria (*Depth of Discharge* - DOD). O DOD é considerado um parâmetro extremamente crítico por estar diretamente relacionado à manutenção da integridade do satélite em órbita.

Os estados simulados do satélite para este estudo de caso são os estados de operação do satélite XSAT gerados por simulação, por meio da implementação do modelo do subsistema de suprimento de energia. Sendo constituídos pelos valores dos parâmetros, das telemetrias e dos limites operacionais apresentados nesta seção.

5.2 Construção da Base de Classificação Supervisionada

A implementação do modelo do subsistema de suprimento de energia do satélite virtual (XSAT) e de operação do satélite XSAT foi feita na linguagem Visual Basic Version. 6.0. O *Microsoft Visual Basic* consiste em uma linguagem de programação de desenvolvimento de aplicações visuais para o ambiente *Windows* baseado em *Basic* (*Beginners All-purpose Symbolic Instruction Code*). Mas poderia ser feito em qualquer ambiente visual.

A Figura 5.1 mostra a interface em Visual Basic do aplicativo XSAT *database generator*, desenvolvido para gerar a base de dados classificados para treinamento. O aplicativo disponibiliza para o usuário as opções de iniciar a operação do XSAT e gerar os registros, além de outras opções como finalização da operação do XSAT.

A partir de uma configuração inicial dos parâmetros do subsistema de suprimento de energia do satélite XSAT (TOMINAGA et al., 2009), a simulação da operação do satélite XSAT, durante a execução do aplicativo, segue o plano de previsão de eclipse e órbita para cada 24 horas com total de 12 órbitas.

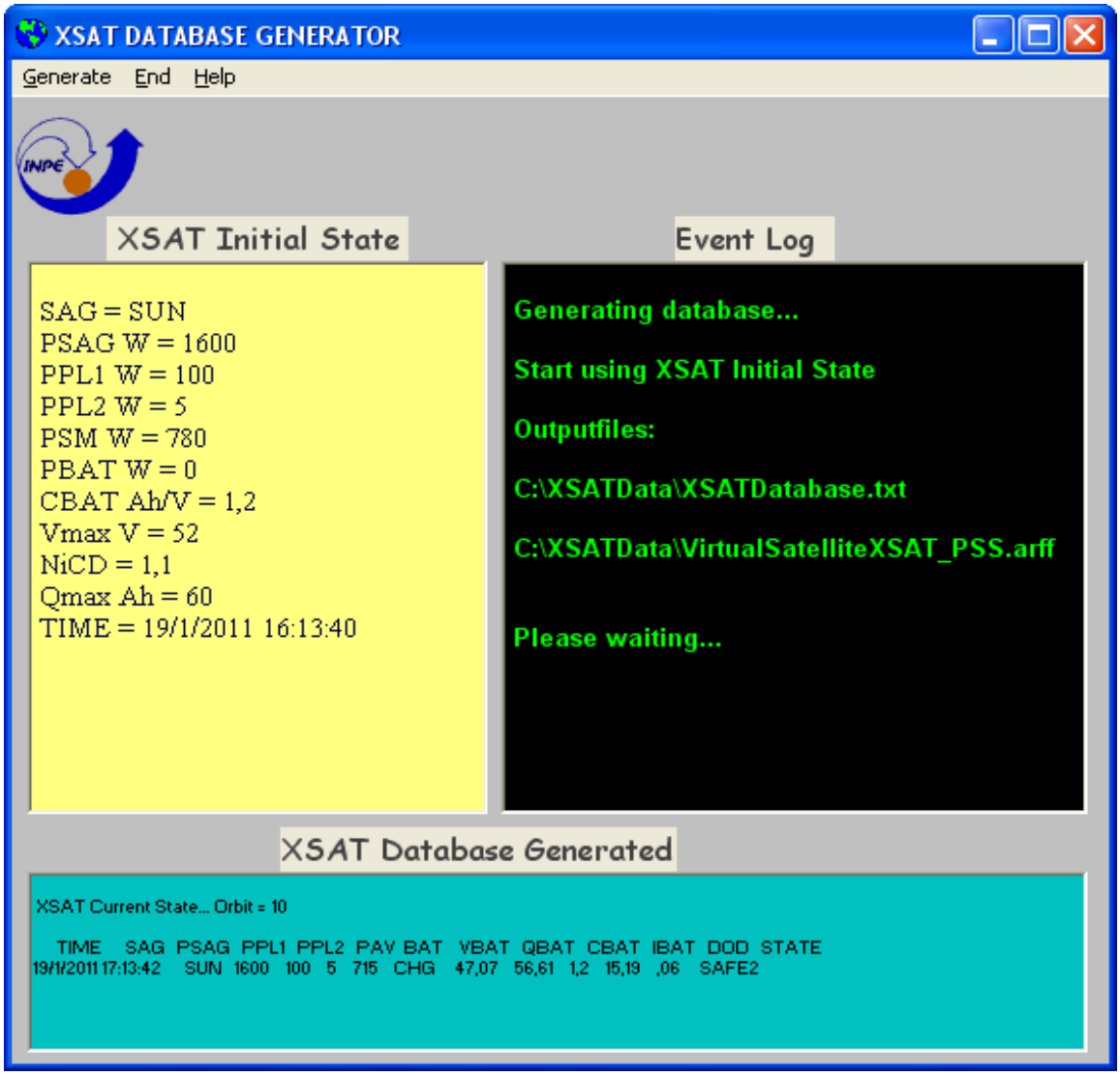

Figura 5.1 – Gerador de registros contendo os estados de operação do XSAT.

O aplicativo XSAT *database generator* foi desenvolvido com base no plano de previsão de eclipse e órbita gerado para o segundo Satélite Sino-Brasileiro de Recursos Terrestres (*China-Brazil Earth-Resources Satellite* -CBERS-2), pelo Software de Análise e Previsão de Passagem para Satélites Artificiais (MILANI et al., 2005), desenvolvido pelo Grupo de Dinâmica Orbital (GDNO) da Divisão da mecânica Espacial e Controle (DMC) do INPE e utilizado pelo Centro de Controle de Satélites (CCS) do Centro de Rastreio e Controle de Satélites (CRC) do INPE.

No aplicativo XSAT *database generator*, foram incluídos os planos de operação de voo, contendo cada um dos modos de operação definidos para o satélite virtual XSAT (vide Tabela 5.4). Durante cada uma das 4 passagens sobre a estação terrena (vide seção 2.2), as ações contidas no plano de operação de voo foram executadas.

Conforme a execução das ações do plano, os valores de cada um dos 12 atributos foram sendo atualizados, gerando os registros. Estes atributos são parâmetros e telemetrias escolhidos por fornecerem um panorama do sistema de energia do satélite (vide seção 5.1).

Com base em um dos atributos, o parâmetro de controle da profundidade de descarga da bateria (DOD), definido por especialistas (vide Tabela 5.5), os registros gerados foram classificados em 2 níveis de segurança de estados operacionais do satélite XSAT, discretizados a partir do estado SAFE como SAFE3 e SAFE2 (Tabela 5.6). No entanto, torna-se importante destacar que poderiam ser selecionados outros níveis de segurança dentro de qualquer um dos limites operacionais definidos pelo especialista para o DOD (vide Tabela 5.5).

O número total de 214 registros ou instâncias geradas pelo aplicativo XSAT para formar a base supervisionada, considerou uma quantidade balanceada de classificações SAFE3 e SAFE2. Registros posteriores correspondiam a outras classificações, que não seriam de interesse para este trabalho.

Tabela 5.6 – Critério de controle do DOD para treinamento dos dados.

DOD (%)	Estado de operação - XSAT
< 4	SAFE3
5 ~ 9	SAFE2

O conjunto de registros classificados como SAFE3 ou SAFE2, gerados pelo aplicativo XSAT database generator é, então, armazenado nos arquivos de saída XSATDatabase.txt e VirtualSatelliteXSAT_PSS.arff.

A Tabela 5.7 apresenta os primeiros registros contidos no arquivo XSATDatabase.txt para serem utilizados como dados ou amostras ou exemplos de treinamento (base de classificação supervisionada) pelos algoritmos classificadores (vide seção 3.3.2).

Tabela 5.7 – Os primeiros registros dos estados de operação do XSAT gerados.

DATE/TIME		SAG	PSAG	PPL1	PPL2	PAV	BAT	VBAT	QBAT	CBAT	IBAT	DOD	STATE
19/4/2010	12:30:10	SUN	1600	100	5	715	FULL	50,00	60,00	1,2	0,00	0,00	SAFE3
19/4/2010	12:30:40	SUN	1600	100	5	715	FULL	50,00	60,00	1,2	0,00	0,00	SAFE3
19/4/2010	12:31:10	SUN	1600	100	5	715	FULL	50,00	60,00	1,2	0,00	0,00	SAFE3
19/4/2010	12:31:40	SUN	1600	100	5	715	FULL	50,00	60,00	1,2	0,00	0,00	SAFE3
19/4/2010	12:32:10	SUN	1600	100	5	715	FULL	50,00	60,00	1,2	0,00	0,00	SAFE3
19/4/2010	12:32:40	ECL	0	100	5	-885	DIS	50,00	59,84	1,2	-19,47	0,00	SAFE3
19/4/2010	12:33:10	ECL	0	100	5	-885	DIS	49,86	59,68	1,2	-19,52	0,01	SAFE3
19/4/2010	12:33:40	ECL	0	100	5	-885	DIS	49,73	59,51	1,2	-19,58	0,01	SAFE3
19/4/2010	12:34:10	ECL	0	100	5	-885	DIS	49,59	59,35	1,2	-19,63	0,01	SAFE3
19/4/2010	12:34:40	ECL	0	100	5	-885	DIS	49,46	59,18	1,2	-19,68	0,01	SAFE3
19/4/2010	12:35:10	ECL	0	100	5	-885	DIS	49,32	59,02	1,2	-19,74	0,02	SAFE3
19/4/2010	12:35:40	SUN	1600	100	5	715	CHG	49,18	59,14	1,2	14,54	0,01	SAFE3
19/4/2010	12:36:10	SUN	1600	100	5	715	CHG	49,28	59,26	1,2	14,51	0,01	SAFE3
19/4/2010	12:36:40	SUN	1600	100	5	715	CHG	49,38	59,38	1,2	14,48	0,01	SAFE3
19/4/2010	12:37:10	SUN	1600	100	5	715	CHG	49,49	59,5	1,2	14,45	0,01	SAFE3
19/4/2010	12:37:40	SUN	1600	100	5	715	CHG	49,59	59,62	1,2	14,42	0,01	SAFE3
19/4/2010	12:38:10	SUN	1600	100	5	715	CHG	49,69	59,74	1,2	14,39	0,00	SAFE3
19/4/2010	12:50:10	SUN	1600	100	5	715	CHG	49,25	59,23	1,2	14,52	0,01	SAFE3
19/4/2010	12:50:40	ECL	0	800	5	-1585	DIS	49,35	58,93	1,2	-35,33	0,02	SAFE3
19/4/2010	12:51:10	ECL	0	100	5	-885	DIS	49,11	58,77	1,2	-19,82	0,02	SAFE3
19/4/2010	12:51:40	ECL	0	100	5	-885	DIS	48,97	58,6	1,2	-19,88	0,02	SAFE3
19/4/2010	12:52:10	ECL	0	100	5	-885	DIS	48,83	58,43	1,2	-19,93	0,03	SAFE3
19/4/2010	12:52:40	ECL	0	100	5	-885	DIS	48,7	58,27	1,2	-19,99	0,03	SAFE3
19/4/2010	12:53:10	ECL	0	100	5	-885	DIS	48,56	58,1	1,2	-20,05	0,03	SAFE3
19/4/2010	12:53:40	SUN	1600	100	5	715	CHG	48,42	58,22	1,2	14,77	0,03	SAFE3
19/4/2010	12:54:10	SUN	1600	100	5	715	CHG	48,52	58,35	1,2	14,74	0,03	SAFE3
19/4/2010	12:54:40	SUN	1600	100	5	715	CHG	48,62	58,47	1,2	14,71	0,03	SAFE3
19/4/2010	12:55:10	SUN	1600	100	5	715	CHG	48,72	58,59	1,2	14,67	0,02	SAFE3
19/4/2010	12:55:40	SUN	1600	100	5	715	CHG	48,83	58,71	1,2	14,64	0,02	SAFE3
19/4/2010	12:56:10	SUN	1600	100	5	715	CHG	48,93	58,84	1,2	14,61	0,02	SAFE3
19/4/2010	13:44:40	ECL	0	800	15	-1595	DIS	47,25	56,39	1,2	-37,13	0,06	SAFE2
19/4/2010	13:45:10	ECL	0	100	5	-885	DIS	47	56,22	1,2	-20,71	0,06	SAFE2
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....

5.3 Construção dos Modelos de Classificação Preditiva

A tarefa de mineração de dados para a construção dos modelos de classificação preditiva com diferentes abordagens foi realizada por um algoritmo classificador, representando cada técnica de classificação. A seleção das técnicas baseou-se naquelas adequadas à classificação binária. A Tabela 5.8 mostra cada técnica de classificação e o respectivo algoritmo classificador selecionado.

Cada algoritmo classificador utilizou como entrada (Tabela 5.8), a base de classificação supervisionada, que corresponde ao conjunto de dados de treinamento e teste armazenado no arquivo VirtualSatelliteXSAT_PSS.arff (vide seção 5.1), conforme mostra a Figura 5.2.

Tabela 5.8 –. Técnica de classificação e o algoritmo classificador selecionado.

Técnica de Classificação	Algoritmo Classificador
Árvore de Decisão	J48
Rede Neural Artificial	LVQ2_1
Bayesiano	NaiveBayes
Máquina de Vetor de Suporte	SMO
Vizinho Mais Próximo	KStar
Baseado em Regras	JRip

O formato *attribute-relation file format* - arff, (Figura 5.2) constitui um padrão exigido para a execução dos algoritmos classificadores, que integram um pacote desenvolvido para análise do conhecimento (*Waikato Environment for Knowledge Analysis* - WEKA), Version.3.4.11, ©1999-2007. Um software livre, escrito em Java e disponibilizado sob a licença *GNU General Public License (GPL)*, com o objetivo de adicionar algoritmos de abordagens diferentes na sub-área da Inteligência Artificial, dedicado ao estudo da aprendizagem de máquina (FRANK et al., 1999).

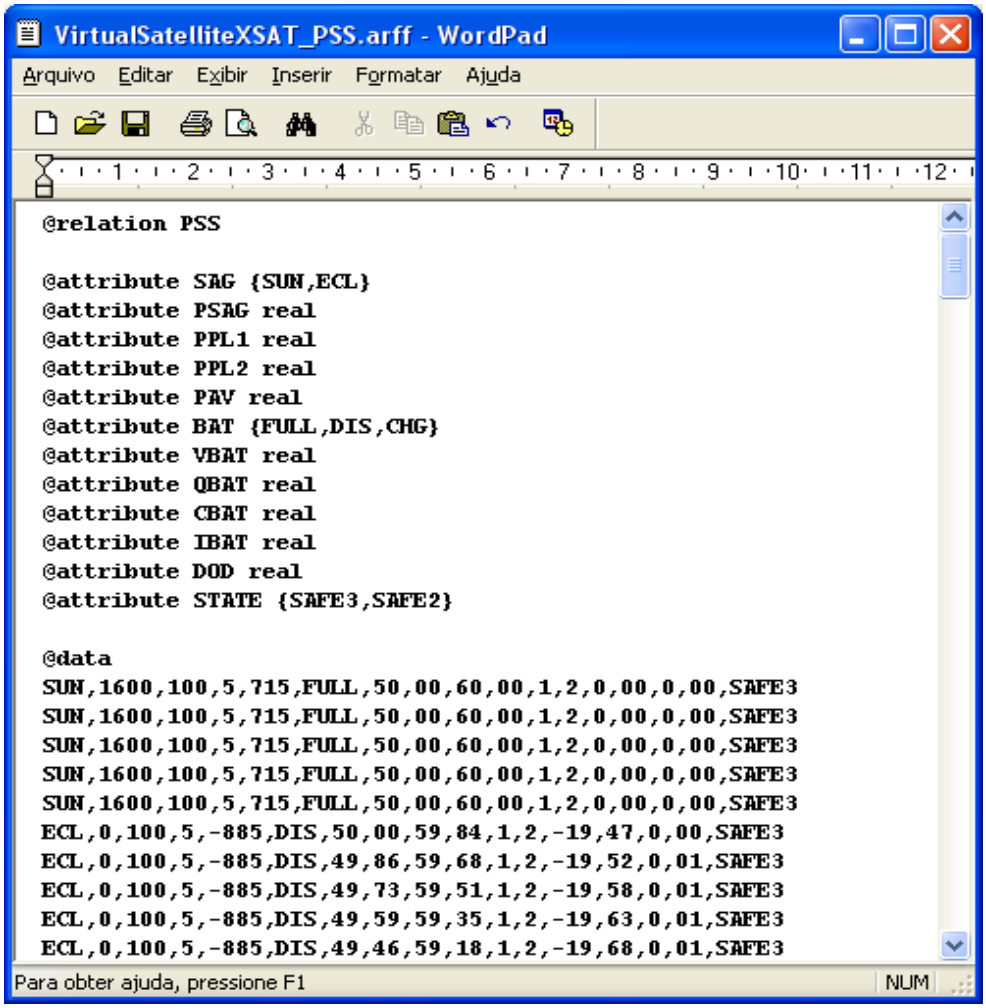

Figura 5.2 – Arquivo de dados de treinamento/teste em formato arff.

A Figura 5.3 Ilustra o ambiente Weka, preparado para a execução dos algoritmos classificadores a partir do carregamento do arquivo VirtualSatelliteXSAT_PSS.arff, contendo dados de treinamento e teste (Figura 5.2). Não foi necessário alterar os valores pré-definidos dos parâmetros de configuração para a execução de cada algoritmo classificador.

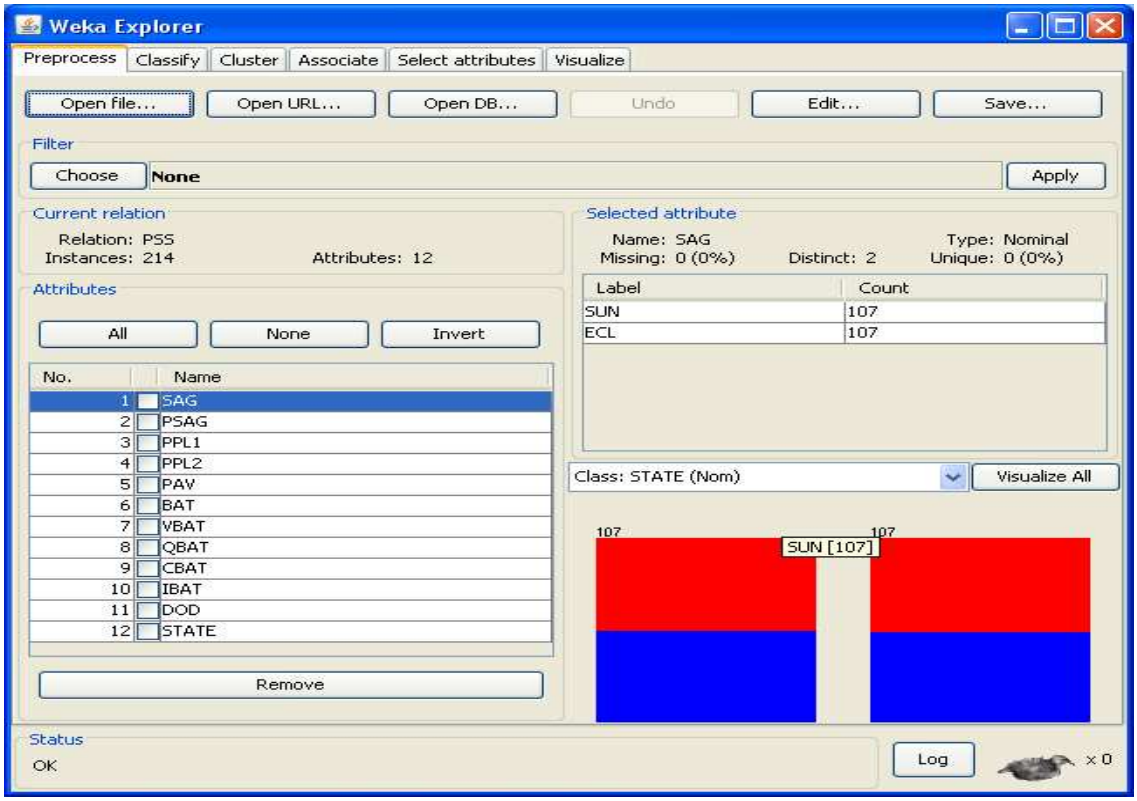

Figura 5.3 – Ambiente weka preparado com os dados da base supervisionada do subsistema de suprimento de energia do satélite virtual XSAT.

5.3.1 Algoritmo Classificador J48

O algoritmo de classificação J48 foi selecionado para a construção do modelo de classificação baseado em árvore de decisão, que apresenta uma abordagem determinística, cuja construção da árvore ocorre conforme a informação extraída dos dados.

O algoritmo J48 utilizado consiste na implementação em Java de uma classe para gerar árvore de decisão C4.5 sem poda ou podada (QUINLAN, 1993), release 8 (vide seção 3.3.1.1). Sendo o weka.classifiers.trees.J48 parte integrante do pacote Weka, Version. 3.4.11, ©1999-2007 (FRANK et al., 1999).

Parâmetros de configuração para a execução:

- confidenceFactor - Fator de confiança utilizado para poda da árvore, em que valores menores correspondem a mais poda (C= 0.25). Não se observa alterações no erro para valores de C> 0.5;
- minNumObj - Número mínimo de instâncias por folha (M=2);
- unpruned – Indica se a árvore gerada será ou não podada (*true* e *false*).

5.3.2 Algoritmo Classificador LVQ2_1

Classificadores baseados em redes neurais artificiais não conhecem a distribuição dos dados, extraindo assim, a estatística dos dados. Para representar um modelo de redes neurais artificiais de classificação, foi utilizada a Rede LVQ de quantização vetorial por aprendizagem, que define uma família de algoritmos adaptativos para a quantificação de vetores, originalmente proposto por Kohonen (1995).

O algoritmo de classificação selecionado para a implementação de redes LVQ foi o LVQ2_1 (FRANK et al., 1999), que consiste num algoritmo iterativo, cujo princípio básico é o de reduzir a distância entre os vetores de entrada na mesma classe e afastar-se do vetor de entrada na classe errada (vide seção 3.3.1.2).
O processo de classificação em todos os métodos LVQ é idêntico, diferindo somente quanto ao processo de treinamento. O LVQ2 atualiza duas células a cada iteração durante o treinamento, enquanto o LVQ1 atualiza somente 1 célula.

Na versão do classificador LVQ2_1 utilizada, as duas melhores unidades de correspondência (*Best Matching Units* - BMU) são selecionadas por um vetor de dados. Uma das classes do BMU deve coincidir com o vetor de dados, que estão ajustados as dimensões da janela. O algoritmo LVQ2_1 consiste em uma

implementação em Java, sendo o weka.classifiers.neural.lvq.Lvq2_1 parte integrante do pacote Weka, Version.3.4.11, ©1999-2007 (FRANK et al., 1999).

Parâmetros de configuração para a execução:

- initialisationMode - Modo de inicialização do modelo (vetor codebook). Opções:1=Random Training Data Proportional, 2=Random Training Data Even, 3=Random Values In Range, 4=Simple KMeans, 5=Farthest First, 6==K-Nearest Neighbour Even (M=1);

- learningFunction - Função para taxa de aprendizagem, utilizada durante o treinamento. A função linear normalmente responde melhor. Opções: 1=Linear Decay, 2=Inverse, 3=Static (L=1);

- learningRate - Valor da taxa inicial de aprendizagem. Recomendam-se os valores 0.3 ou 0.5 (R=0,3);

- totalCodebookVectors - Número total de vetores codebook no modelo (C=20 – número mínimo de codebooks). Objetivo do algoritmo consiste em aproximar a distribuição de uma classe usando um número reduzido de vetores codebook, e assim, procura minimizar erros de classificação;

- totalTrainingIterations - Número total de iterações para treinamento. Recomenda-se de 30 a 50 vezes o número de vetores codebook (I=1000);

- useVoting - Permite selecionar dinâmicamente a classe atribuída para cada vetor codebook. Prevê o tratamento automático para instâncias erroneamente classificadas (G=false);

- seed – Semente utilizada para randomização dos dados (S=1);

- windowSize - Tamanho da janela para serem ajustados os vetores *codebook*. Recomenda-se 0,2 ou 0,3 (W=0).

5.3.3 Algoritmo Classificador Naive Bayes

Classificadores bayesianos apresentam uma abordagem não determinística, baseando-se na inferência probabilística. Em outras palavras, o rótulo da classe de um registro de teste não pode ser previsto com certeza, embora seu conjunto de atributos seja idêntico a alguns dos exemplos de treinamento (vide seção 3.3.1.3).

Classificador bayesiano selecionado foi o algoritmo NaiveBayes (JOHN e LANGLEY, 1995), A versão utilizada, que consiste em uma implementação de uma classe em Java, realiza uma estimativa das classes a partir do modelo probabilístico Naive Bayes. A precisão dos valores numéricos deste “estimador” são escolhidos com base na análise dos dados de treinamento. O algoritmo weka.classifiers.bayes.NaiveBayes, constitui parte integrante do pacote Weka, Version.3.4.11, ©1999-2007 (FRANK et al., 1999).

Parâmetros de configuração para a execução:

- useKernelEstimator – Utiliza um estimador *kernel* para atributos numéricos ao invés de uma distribuição normal, onde calcula-se a média e o desvio padrão. Em alguns casos a estimativa do kernel pode ser interessante para aproximar a formas de distribuição mais complexas. Foi utilizada a distribuição normal (*false*);

- useSupervisedDiscretization – Utiliza uma discretização supervisionada para converter atributos numéricos para os nominal (*false*).

Para todos os atributos numéricos foi utilizada a opção default de distribuição normal.

5.3.4 Algoritmo Classificador SMO

Classificadores baseados em máquina de vetor de suporte (SVM) estão fundamentados na teoria do aprendizado estatístico para minimizar erros da classificação empírica e maximizar a margem geométrica entre os resultados, visando estabelecer condições matemáticas que permitem escolher um classificador, com bom desempenho, para o conjunto de dados disponíveis para treinamento e teste (vide seção 3.3.1.4).

Esta técnica originalmente desenvolvida para classificação binária, busca a construção de um hiperplano como superfície de decisão, de tal forma que a separação entre exemplos seja máxima. Isso considerando padrões linearmente separáveis. Em outras palavras esta teoria busca encontrar um bom classificador levando em consideração todo o conjunto de dados, buscando uma separação ótima entre as classes

O algoritmo de classificação selecionado para a implementação do modelo de classificação baseado em máquina de vetor de suporte foi o algoritmo de otimização sequencial mínima (SMO) (PLATT, 1998) (KEERTHI et al., 2001).

O algoritmo SMO utilizado consiste em uma implementação em Java, sendo o weka.classifiers.functions.SMO parte integrante do pacote Weka, Version.3.4.11, ©1999-2007 (FRANK et al., 1999).

Esta implementação do classificador SMO, substitui todos os valores faltantes e transforma os atributos nominais em binário, além de normalizar todos os atributos. Os coeficientes da saída são baseados em dados originais, não normalizados.

Parâmetros de configuração para a execução:

- C - Parâmetro complexidade. Relacionado ao aumento do tempo computacional (C=1.0);
- cacheSize - Tamanho do cache do kernel (A=250007);
- epsilon - Fator para arredondamento de erro, que não deve ser alterado (P= 1.0E-12);
- exponent – Expoente para o kernel polinomial (E=1.0);
- gamma - Valor do parâmetro gama para kernels RBF (G=0.01);
- lowerOrderTerms – Se termos de ordens inferiores são utilizados (L= 0.0010);
- numFolds - Número de dobras para validação cruzada, utilizados para gerar dados de treinamento de modelos logísticos (V=-1: significa utilizar os dados de treinamento);
- randomSeed - Semente de números aleatórios para utilizada para o método de validação cruzada (W=1);
- toleranceParameter - Parâmetro de tolerância, relacionado a quantidade de iterações para aprendizagem da classe (weka). Não deve ser alterado (L=0.0010).

5.3.5 Algoritmo Classificador KStar

Classificadores de vizinho mais próximo procuram encontrar todos os exemplos de treinamento, que sejam relativamente semelhantes aos atributos do exemplo de teste (vide seção 3.3.1.5).

O algoritmo de classificação selecionado para a implementação do modelo de classificadores de vizinho mais próximo foi o algoritmo KStar (K*), um classificador baseado em exemplo, isto é, baseia-se na classe das instâncias

de formação semelhante, conforme determinado por uma função de similaridade (CLEARY e TRIGG,1995). Ele difere de outros por utilizar funções de distância baseada na entropia e assume que os exemplos similares terão classes similares (vide seção 3.3.1.5).
O algoritmo Kstar utilizado consiste em uma implementação em Java, sendo o weka.classifiers.lazy.KStar parte integrante do pacote Weka, Version.3.4.11, ©1999-2007 (FRANK et al., 1999).

Parâmetros de configuração para a execução:

- globalBlend – Parâmetro correspondente a mistura, relacionado ao uso de cálculos de entropia, com valores definidos de 0 a 100 e expressos em porcentagem (B=20);

- missingMode - Determina como os valores de atributos faltantes são tratados (M=a - *average columns entropy curves*).

5.3.6 Algoritmo Classificador JRip

Os classificadores baseados em regras são induzidos a partir dos dados de treinamento e decidem a que classe pertence cada instância de predição a partir de uma série de regras do tipo SE <condição preposicional> E < condição preposicional > E .... ENTÃO<classe> (vide seção 3.3.1.6).

O algoritmo de indução JRip representa o método direto de criação das regras. Este algoritmo escala quase que linearmente o número de exemplos de treinamento, sendo apropriado para construção de modelos a partir de um conjunto de dados com distribuições de classes desequilibradas. O funcionamento está baseado na escolha da classe majoritária como padrão e em seguida descobrir as regras para detectar a classe minoritária (vide seção 3.3.1.6).

O algoritmo JRip utilizado consiste em uma classe em Java, que implementa uma aprendizagem baseada em regra proposicional, com Poda Incremental Repetida para Produzir Redução do Erro (RIPPER), proposto por William W. Cohen como uma versão otimizada do IREP. O weka.classifiers.rules.JRip constitui parte integrante do pacote Weka, Version.3.4.11, ©1999-2007 (FRANK et al., 1999).

Parâmetros de configuração para a execução:

- Folds - Determina a quantidade de dados utilizados para a poda. Uma vez é usado para realizar a poda, o restante para o cultivo das regras (F= 3);
- minNo - Peso mínimo total das instâncias em uma regra (N=2.0);
- optimizations – Número para otimização. Utilizado na fase de otimização após a fase de produção das regras e da fase de poda (O=2);
- seed - Semente utilizada para randomização dos dados (S=1).

5.4 Método de Validação dos Modelos Obtidos a partir dos Dados

O conjunto de registros ou instâncias ou amostras contendo os estados de operação do XSAT, que formam a base de classificação supervisionada apresentados na Tabela 5.7 serão utilizados pelos algoritmos classificadores como dados ou instâncias de treinamento e de teste, uma vez que os dados de treinamento e os dados de teste são amostras representativas do problema (vide seção 3.3.2).

Os arquivos com dados de treinamento e de teste podem ter naturezas diferentes. Mas, o conjunto de teste deve ser um conjunto de exemplos

independentes que não foram usados na geração do modelo. Sendo possível assim trazer confiabilidade na validação do modelo.

A abordagem utilizada por todos os algoritmos classificadores para validação dos modelos obtidos dos dados gerados e que representam o estado do sistema de energia do XSAT, consiste no método padrão atual do ambiente Weka de avaliação chamado validação cruzada (*cross-validation*) 10 vezes ou 10 dobras, como alternativa a subamostragem aleatória. Este tipo de validação estima a precisão de um classificador, uma vez que neste método os dados da amostragem são segmentados ou estratificados em 10 partições de mesmo tamanho. A Figura 5.4 ilustra o processo de validação cruzada de 4 dobras no ambiente Weka.

Durante cada execução, uma das partições é escolhida para teste, enquanto o restante delas passa a ser utilizada para treinamento. Este procedimento é repetido 10 vezes de modo que cada partição é utilizada exatamente uma vez para testar o modelo (vide seção 3.3.2).

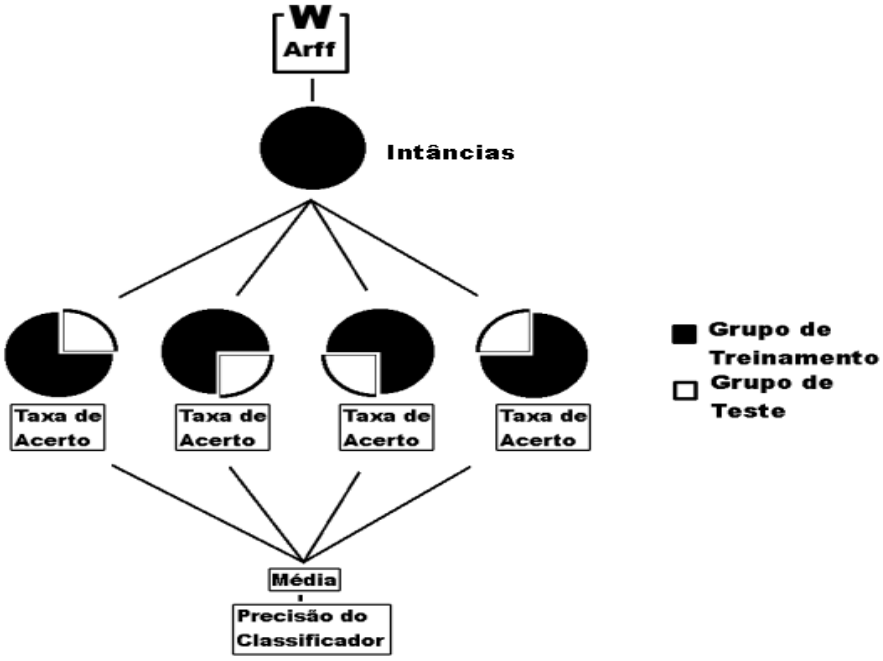

Figura 5.4 – Representação no Weka da validação cruzada de 4 dobras.

Duas vantagens estão relacionadas ao uso do método de validação cruzada: a primeira relaciona-se a forma de manusear os dados da base supervisionada inerente ao método, que evita a superposição dos conjuntos de teste. A segunda vantagem relaciona-se a validação cruzada com o número 10 de partes estratificadas ou dobras, em que evidencias experimentais por meio de testes extensivos em conjuntos de dados diversos e utilizando diferentes técnicas de aprendizagem, demonstraram ser este o número adequado de estratificação da amostragem para obtenção da melhor estimativa de erro para a média calculada e redução da variância.

A partir da execução dos algoritmos classificadores selecionados, apresentados neste capítulo e que representam os modelos de classificação selecionados (vide seção 3.3), serão avaliados os modelos de classificação gerados pelos algoritmos classificadores (vide seção 3.3.3). O melhor resultado obtido desta análise indica o modelo de predição que apresenta maior acurácia para classificar os estados operacionais do satélite (vide seção 4.2). O próximo capítulo apresenta a análise estatística dos resultados obtidos pelos modelos de classificação na busca do modelo mais adequado para a geração de diagnóstico (vide seção 4.2).

6 AVALIAÇÃO DOS MODELOS DE CLASSIFICAÇÃO PREDITIVA GERADOS

Neste capítulo são apresentados os modelos de classificação gerados pelos algoritmos classificadores e os resultados da classificação fornecida por cada classificador, sendo a partir destes resultados realizada uma análise estatística destes modelos de classificação construídos para a realização de predição dos estados do satélite XSAT.

O método utilizado para validação dos modelos de classificação obtidos a partir dos dados foi à validação cruzada 10-fold repetida 10 vezes, padrão no ambiente Weka (vide seção 3.3.2). Neste método os classificadores são testados com todas as instâncias do conjunto de treinamento. Neste trabalho foi adotada uma estratificação da amostragem (base supervisionada) de 10 partes ou dobras, onde se reserva uma parte estratificada da amostragem qualquer para teste, deixando-se as demais para treinamento. O processo se repete alternando a parte estratificada de teste. O número 10 para estratificação da amostragem resulta de evidencias experimentais, que demonstraram uma redução na variância da estimativa (vide seção 5.4).

A análise dos resultados representa a fase de avaliação dos resultados obtidos pelos classificadores (vide seção 4.2). Estes resultados são tabulados em uma tabela conhecida como matriz de confusão e utilizados para obtenção das métricas de desempenho e das funções estatísticas designadas para avaliação de modelos de classificação (vide seção 3.3.3).

O resultado da análise determina qual o modelo de classificação preditiva apresenta maior precisão no diagnóstico e predição dos estados do satélite, indicando o modelo de classificação mais adequado para o desenvolvimento da ferramenta gerador de diagnóstico.

6.1 Modelos de Classificação Gerados

A definição das siglas dos atributos que representam o sistema de energia do satélite XSAT, referenciados nos modelos de classificação gerados, encontra-se nas Tabelas: 4.1, 4.2 e 4.3.

O algoritmo de classificação J48 foi executado com o parâmetro configurado para a construção do modelo de árvore com poda (vide seção 5.3.1).

O modelo de árvore construído indicou que entre o conjunto de atributos, a telemetria relacionada com a voltagem da bateria (VBAT) aparece de forma determinante na classificação do estado operacional que se encontra o satélite XSAT (vide seção 5.1).

A Figura 6.1 mostra o modelo de árvore gerado, ilustrando como o modelo atua na predição de estados futuros do satélite a partir de valores de VBAT desconhecidos, contidos nos dados de teste ou contidos em novos registros de dados com amostras desconhecidas.

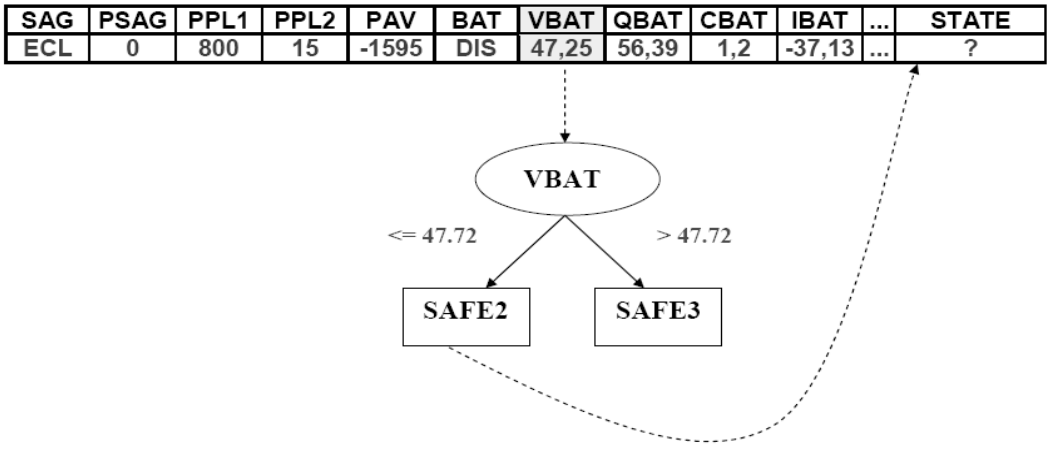

Figura 6.1 – Modelo baseado em árvore de decisão com poda gerado para classificação de amostras desconhecidas.

Com o objetivo de avaliar o custo computacional e o desempenho em relação ao modelo anterior, um segundo modelo de árvore de decisão sem poda foi

gerado pelo algoritmo de classificação J48 (vide seção 5.3.1). A Figura 6.2 mostra o modelo de árvore gerado, sem poda.

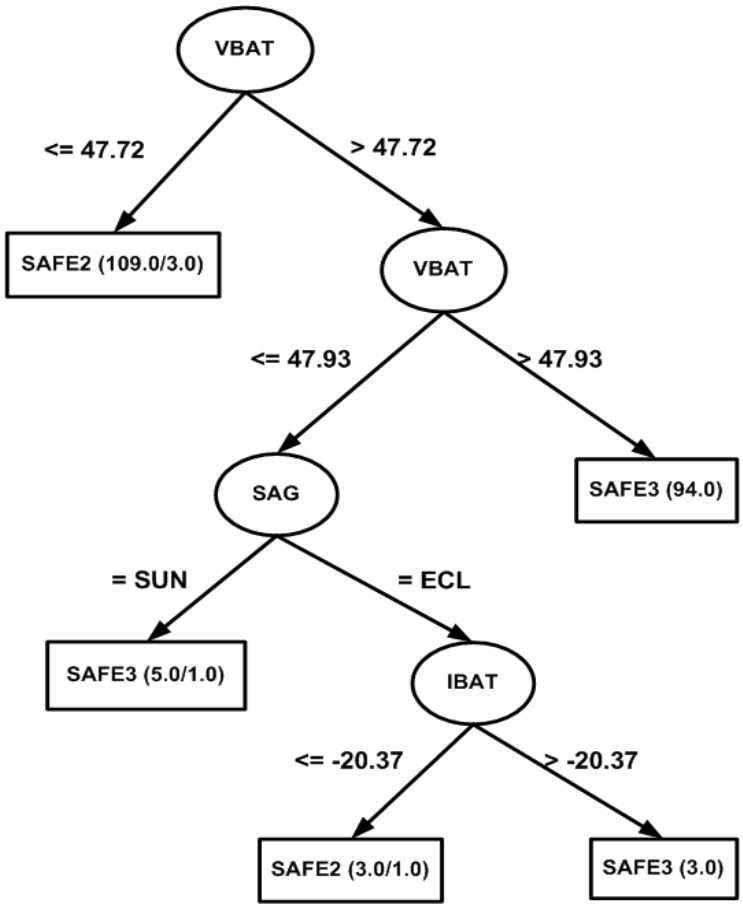

Figura 6.2 – Modelo baseado em árvore de decisão sem poda, gerado para classificação de amostras desconhecidas.

A distribuição estatística das classes fornecida pelo algoritmo de classificação LVQ2_1 (vide seção 5.3.2) foi de SAFE3: 50% e SAFE2: 50% para os vetores de entrada, que representam os 12 atributos (vide seção 3.3.3.2).

Os classificadores Naive Bayes realizam os cálculos assumindo uma condição forte de independência, ou seja, todos os atributos que representam o sistema de energia do XSAT são condicionalmente independentes dado o valor da classe SAFE3 ou SAFE2 (vide seção 5.3.3). Sendo, então fornecido o valor da probabilidade prévia para a classe SAFE3: P(SAFE3)=49% e o valor da probabilidade prévia para a classe SAFE2: P(SAFE2)=51%.

Assim, para classificar cada instância ou registro, o classificador calcula a partir das probabilidades prévias de SAFE3 e SAFE2, as probabilidades posteriores P(SAFE3|X) e P(SAFE2|X).

O algoritmo de classificação SMO, baseado em vetores de suporte (vide seção 5.3.4), utiliza o BinarySMO (voltado a classificação binária) para classificar nas classes: SAFE3 e SAFE2. Pesos foram atribuídos aos atributos, conforme apresenta a Figura 6.3 a seguir:

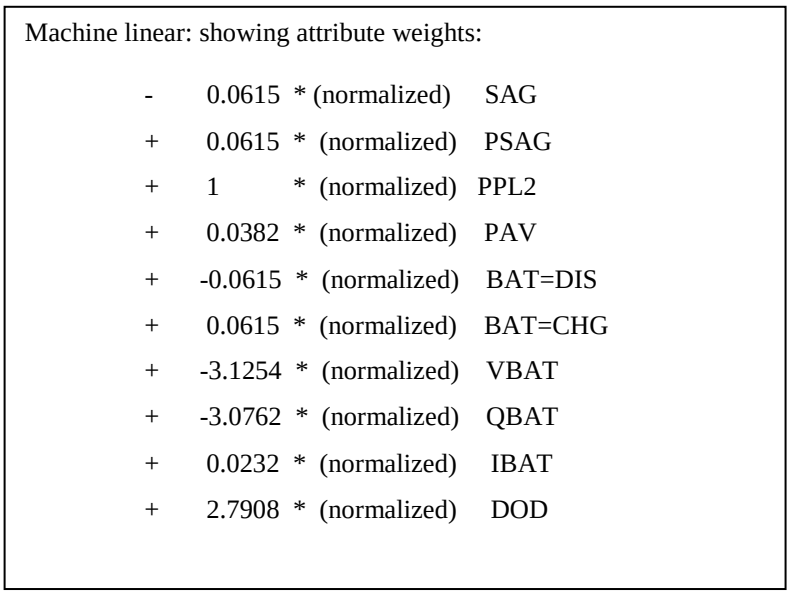

Figura 6.3 – Pesos atribuídos aos atributos pelo classificador SMO.

O resultado mostra a fronteira de decisão linear encontrada pela SVM. Para linear *Support Vector Machines*, esse limite pode ser expresso em termos de atributos originais, bem como os vetores de suporte. O resultado mostra que foram um total de 5810 avaliações da função de kernel e que 78,583% já haviam sido computadorizadas, ou seja, retornaram do cache.

O algoritmo de classificação KStar (vide seção 5.3.5), que utiliza funções de distância baseado na entropia, após a execução forneceu diretamente as contagens das instâncias corretamente e incorretamente classificadas.

O algoritmo de classificação JRip (vide seção 5.3.6), gerou um total de 2 regras:

1. (VBAT >= 47.73) => STATE=SAFE3 (105.0/3.0)

2. STATE=SAFE2 (109.0/3.0)

6.2 Análise Estatística dos Modelos de Classificação

A análise estatística dos modelos de classificação baseia-se nas métricas de desempenho e nas funções estatísticas específicas para avaliação da qualidade dos modelos de classificação (vide seção 3.3.3). Os valores das métricas resultaram do cálculo de cada função estatística, realizado a partir dos dados classificados e tabulados na matriz de confusão (vide seção 3.3.3). O resultado desta análise estatística determinou entre os modelos de classificação selecionados (vide seção 4.2), qual apresentou maior precisão no diagnóstico e predição dos estados do satélite.

6.2.1 Matriz de Confusão Resultante

A avaliação de desempenho de um modelo de classificação baseia-se na contagem de registros ou instâncias de teste corretamente e incorretamente previsto pelo modelo. Estas contagens são tabuladas em uma tabela conhecida como matriz de confusão (vide seção 3.3.3.1). A Tabela 6.1 mostra cada matriz de confusão resultante da classificação dos registros de teste como SAFE3 e SAFE2, realizada pelos algoritmos classificadores.

A Tabela 6.1 apresenta ainda a técnica de classificação associada ao algoritmo classificador na construção do modelo, bem como o total de instâncias classificadas e o total de classificação por classes. Estas contagens são necessárias para o cálculo de outras funções estatísticas de avaliação da qualidade de cada classificador (vide seção 3.3.3).

Tabela 6.1 – Matriz de confusão dos seis algoritmos classificadores.

Técnica de Classificação		Classes Previstas		
Árvore de Decisão	J48	Classe = SAFE3	Classe = SAFE2	Total
	Classe = SAFE3	e_{ii} = 101	e_{ij} = 4	105
	Classe = SAFE2	e_{ji} = 8	e_{jj} = 101	109
	Total	109	105	214
Rede Neural Artificial	LVQ2_1	Classe = SAFE3	Classe = SAFE2	Total
	Classe= SAFE3	e_{ii} = 99	e_{ij} = 6	105
	Classe = SAFE2	e_{ji} = 5	e_{jj} = 104	109
	Total	104	110	214
Bayesiano	NaiveBayes	Classe = SAFE3	Classe = SAFE2	Total
	Classe = SAFE3	e_{ii} = 99	e_{ij} = 6	105
	Classe = SAFE2	e_{ji} = 2	e_{jj} = 107	109
	Total	101	113	214
Máquina de Vetores de Suporte	SMO	Classe = SAFE3	Classe = SAFE2	Total
	Classe= SAFE3	e_{ii} = 100	e_{ij} = 5	105
	Classe = SAFE2	e_{ji} = 4	e_{jj} = 105	109
	Total	104	110	214
Vizinho Mais Próximo	KStar	Classe = SAFE3	Classe = SAFE2	Total
	Classe= SAFE3	e_{ii} = 100	e_{ij} = 5	105
	Classe = SAFE2	e_{ji} = 5	e_{jj} = 104	109
	Total	105	109	214
Baseado em Regras	JRip	Classe = SAFE3	Classe = SAFE2	Total
	Classe= SAFE3	e_{ii} = 100	e_{ij} = 5	105
	Classe = SAFE2	e_{ji} = 5	e_{jj} = 104	109
	Total	105	109	214

Cada elemento e_{ii} da matriz de confusão indica o número de registros da classe SAFE3 previsto corretamente como classe SAFE3, enquanto o elemento e_{jj} indica o número de registros da classe SAFE2 previsto corretamente como classe SAFE2. O elemento e_{ij} da matriz de confusão indica o número de registros da classe SAFE3 previsto incorretamente como classe SAFE2, enquanto eji indica o número de registros da classe SAFE2 previstos incorretamente como SAFE3.

Portanto, com base nos elementos da matriz de confusão da Tabela 6.1 foram calculados o número total de predições corretas e de predições incorretas de cada modelo para o número total de 214 instâncias. Sendo o número total de predições corretas feitas por cada classificador, obtido pela soma dos elementos $e_{ii}+e_{jj}$ e o número total de predições incorretas, obtido pela soma dos elementos: $e_{ij}+e_{ji}$, conforme apresentado na Tabela 6.2.

Tabela 6.2 – Classificação das instâncias conforme cada classificador.

Classificador	Instâncias Corretamente Classificadas	Instâncias Erroneamente Classificadas	Total de Instâncias
J48	202	12	214
LVQ2_1	203	11	214
NaiveBayes	206	8	214
SMO	205	9	214
KStar	204	10	214
JRip	204	10	214

Observa-se na Tabela 6.2, que para o total de 214 instâncias o algoritmo classificador estocástico Naive Bayes classificou corretamente o maior número de instâncias de teste (206 instâncias) e, consequentemente, apresentou o menor número de instâncias classificadas erroneamente (8 instâncias). O classificador SMO apresentou o segundo maior número de instâncias classificadas corretamente, seguido dos classificadores KStar e JRip, que apresentaram o mesmo número de instâncias classificadas corretamente (204 instâncias).

O classificador LVQ2_1, aparece em seguida, apresentando a diferença de 1 instância classificada corretamente a menos (203 instâncias) dos classificadores KStar e JRip.

Por fim, o algoritmo J48 aparece como o classificador que apresentou o maior número de instâncias classificadas erroneamente. O modelo de arvore de decisão sem poda, gerado pelo algoritmo J48 não foi nem considerado em

função do resultado do número de instâncias classificadas corretamente ser ainda inferior ao modelo com poda (201 instâncias).

6.2.2 Métricas de Desempenho

A partir dos valores da matriz de confusão (Tabela 6.1) e dos valores de classificações das instâncias (Tabela 6.2) foram calculadas métricas para comparação do desempenho entre os diferentes modelos (vide seção 3.3.3.1). Foram obtidos os valores da exatidão ou acurácia (*accuracy*) e taxa de erro associados a cada modelo. Os valores calculados são apresentados na Tabela 6.3.

Tabela 6.3 – Métricas de desempenho para cada algoritmo classificador.

Classificador	Exatidão (%)	Taxa de erro (%)
J48	94,3925 %	5,6075 %
LVQ2_1	94,8598 %	5,1402 %
NaiveBayes	96,2617 %	3,7383 %
SMO	95,7944 %	4,2056 %
KStar	95,3271 %	4,6729 %
JRip	95,3271 %	4,6729 %

Observa-se na Tabela 6.3, que os valores para a exatidão ou acurácia dos classificadores avaliados apresentam pouca diferença no que diz respeito à acurácia. Com base nesta métrica, a Figura 6.4 apresenta a comparação entre os classificadores.

De acordo com os valores das métricas o modelo que atingiu a maior exatidão ou, equivalentemente, a menor taxa de erro quando aplicado o conjunto de testes foi o modelo gerado pelo classificador estocástico Naive Bayes (96,26% e 3,73%). O modelo que apresentou o menor percentual de exatidão foi o classificador determinístico J48.

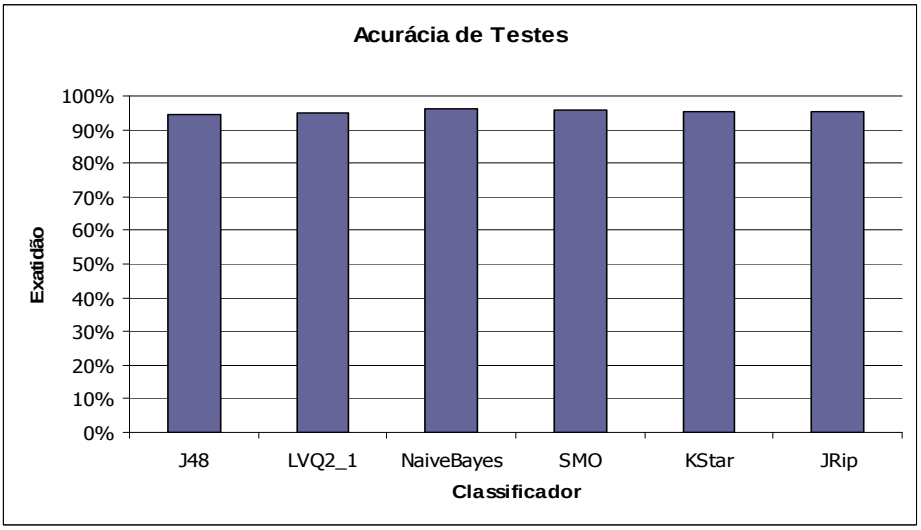

Figura 6.4 – Acurácia de testes dos classificadores utilizados.

Os resultados extraídos da matriz de confusão em termos de verdadeiro positivo (TP), falso positivo (FP), verdadeiro negativo (TN) e falso negativo (FN) (vide seção 3.3.3.1.). Estes valores foram utilizados por servirem de base para os cálculos das demais métricas indicadas para aferir a qualidade de cada classificador. São elas: taxa de verdadeiros positivos, taxa de falsos positivos, precisão, cobertura e *F-Measure* por classe (vide seção 3.3.3.1.).

A Tabela 6.4 apresenta os valores das métricas por classe (SAFE3 e SAFE2) de cada classificador. A métrica de comparação utilizada foi a *F-measure*, definida como a média harmônica entre a precisão (*Precision*) e a cobertura (*Recall*) (vide seção 3.3.3.1). A precisão corresponde à razão entre o número de instâncias classificadas corretamente em uma classe e o número de instâncias classificadas como aquela classe (tanto corretamente quando incorretamente). Enquanto a cobertura corresponde à razão entre o número de instâncias classificadas corretamente em uma classe e o numero total de instancias da classe (vide seção 3.3.3.1.).

Os resultados apresentados na Tabela 6.4 mostram que de uma forma geral as taxas de cobertura e precisão para ambas as classes SAFE3 e SAFE2 foram

altas para todos os classificadores, indicando uma boa distribuição dos dados utilizados para o treinamento.

Os valores da taxa TP ou a cobertura para a classe SAFE3 foi maior no classificador J48 (0.962), sendo menor nos classificadores Naive Bayes e LVQ2_1 (0.943). Enquanto o valor da cobertura para a classe SAFE2, ocorreu justamente o inverso, sendo maior no classificador Naive Bayes (0.982) e menor no classificador J48 (0.927).

Tabela 6.4 – Métricas de desempenho por classe.

J48	Taxa TP	Taxa FP	Precisão	Cobertura	F-Measure	Classe
	0,962	0,073	0,927	0,962	0,944	SAFE3
	0,927	0,038	0,962	0,927	0,944	SAFE2
LVQ2_1	Taxa TP	Taxa FP	Precisão	Cobertura	F-Measure	Classe
	0,943	0,046	0,952	0,943	0,947	SAFE3
	0,954	0,057	0,945	0,954	0,95	SAFE2
Naive Bayes	Taxa TP	Taxa FP	Precisão	Cobertura	F-Measure	Classe
	0,943	0,018	0,98	0,943	0,961	SAFE3
	0,982	0,057	0,947	0,982	0,964	SAFE2
SMO	Taxa TP	Taxa FP	Precisão	Cobertura	F-Measure	Classe
	0,952	0,037	0,962	0,952	0,957	SAFE3
	0,963	0,048	0,955	0,963	0,959	SAFE2
KStar	Taxa TP	Taxa FP	Precisão	Cobertura	F-Measure	Classe
	0,952	0,046	0,952	0,952	0,952	SAFE3
	0,954	0,048	0,954	0,954	0,954	SAFE2
JRip	Taxa TP	Taxa FP	Precisão	Cobertura	F-Measure	Classe
	0,952	0,046	0,952	0,952	0,952	SAFE3
	0,954	0,048	0,954	0,954	0,954	SAFE2

Os valores de precisão do modelo gerado pelo classificador Naive Bayes foi o maior para classe SAFE3 (0,98) e o menor para a classe SAFE2. Enquanto a precisão do modelo gerado pelo classificador J48 foi a menor para classe SAFE3 (0,98) e a maior para a classe SAFE2.

Como métrica de comparação utilizou-se então, a medida de desempenho F-measure para medir a capacidade de generalização de cada modelo gerado, verificando se durante o treinamento, o modelo assimilou do conjunto de dados características significativas para um bom desempenho ou se concentrou em peculiaridades. Sendo assim, o modelo gerado que apresentou o maior valor da métrica F-measure para ambas as classes foi o modelo do classificador Naive Bayes (0,961 e 0,964), seguido do classificador SMO (0,957 e 0,959).

6.2.3 Estatística ou Coeficiente Kappa

A partir da matriz de confusão, também foi calculado o coeficiente ou estatística Kappa para cada classificador. A partir dos valores de kappa obtidos foi feita a interpretação da medida da concordância real das instâncias classificadas por cada classificador (vide seção 3.3.3.2). Na avaliação dos modelos gerados, a estatística kappa corresponde ao resultado da comparação entre a classificação feita por cada classificador e o conjunto de treinamento, que representa o classificador perfeito. A Tabela 6.5 apresenta o valor de cada coeficiente Kappa obtido de cada algoritmo classificador e a interpretação da concordância correspondente:

Tabela 6.5 – Coeficiente kappa e a respectiva interpretação.

Classificador	Kappa	Interpretação
J48	0,8879	Concordância quase perfeita
LVQ2_1	0,8971	Concordância quase perfeita
NaiveBayes	0,9252	Concordância quase perfeita
SMO	0,9158	Concordância quase perfeita
KStar	0,9065	Concordância quase perfeita
JRip	0,9065	Concordância quase perfeita

Os resultados apresentados na Tabela 6.5 mostram que de uma forma geral todos os classificadores apresentaram uma concordância considerada excelente em comparação com a classificação existente no conjunto de

treinamento. Todos os valores ficaram dentro do intervalo (0,81 a 1), cuja interpretação define neste caso, a concordância bem próxima ao valor máximo, definido como 1 (vide seção 3.3.3.2).

O classificador estocástico Naive Bayes apresentou o maior valor de kappa (0,92), seguido do classificador SMO (0,9158). O classificador J48 apresentou o menor valor para a estatística kappa (0,89).

6.2.4 Outras Funções Estatísticas para Avaliação dos Classificadores

Os valores das estatísticas obtidas até esta etapa da análise dos modelos, mostram resultados idênticos para os classificadores KStar e JRip. O algoritmo KStar utiliza funções de distância baseada na entropia (vide seção 5.3.5) e JRip, baseado em regras, utiliza poda incremental repetida para produzir redução de erro (vide seção 5.3.6). No entanto, apesar das abordagens distintas, ambos os classificadores tiveram desempenho idêntico para o conjunto de dados de treinamento.

Quando os resultados da acurácia de testes entre classificadores mostram-se muito próximos ou até mesmo idênticos, torna-se necessário utilizar funções estatísticas para obtenção de medidas relacionadas ao cálculo de erro, que também avaliam a qualidade de cada classificador. São elas: o erro absoluto médio (vide seção 3.3.3.3), a raiz do erro quadrático médio (vide seção 3.3.3.4), o erro absoluto total normalizado (vide seção 3.3.3.5) e a raiz do erro quadrático (vide seção 3.3.3.6). Os valores obtidos de cada uma destas medidas são apresentados na Tabela 6.6 a seguir:

Tabela 6.6 – Estatísticas dos modelos de classificação.

Classificador	Erro Absoluto Médio	Raiz do Erro Quadrático Médio	Erro Absoluto Total Normalizado	Raiz do Erro Quadrático Relativo
J48	0,0792	0,2317	15,8356 %	46,3509 %
LVQ2_1	0,0514	0,2267	10,2828 %	45,3468 %
Naive Bayes	0,0427	0,1662	8,5375 %	33,249 %
SMO	0,0421	0,2051	8,4132 %	41,0177 %
KStar	0,0601	0,1948	12,0209 %	38,9689 %
JRip	0,0732	0,2132	14,634 %	42,643 %

Observa-se nos valores apresentados na Tabela 6.6, que não ocorreram resultados idênticos. Portanto, estas funções de cálculo de erro para avaliação dos modelos de classificação, que podem atuar como um critério de "desempate" na decisão do modelo mais adequado para classificar o conjunto de dados da aplicação. A Figura 6.5 apresenta um gráfico comparativo das estatísticas dos classificadores KStar e JRip, em que o classificador KStar apresenta valores de erros menores.

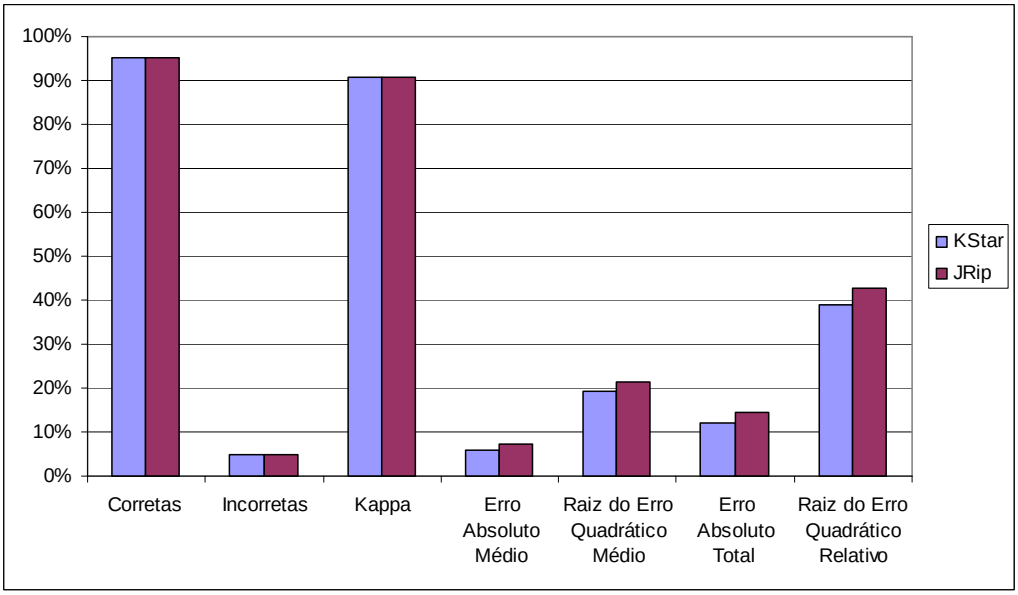

Figura 6.5 – Gráfico comparativo das estatísticas dos classificadores KStar e JRip.

6.3 Modelo de Classificação Selecionado

As seções anteriores apresentaram um estudo comparativo dos modelos de classificação gerados, baseado na análise estatística de desempenho dos algoritmos classificadores selecionados para representarem os métodos de classificação. A estatística utilizada tem como objetivo determinar o modelo de classificação, que apresenta maior acurácia na predição de estados operacionais de satélites, ou seja, o modelo preditivo mais adequado em função da base de dados.

Os resultados alcançados na fase de análise estatística dos modelos de classificação gerados, mostraram que o classificador estocástico Naïve Bayes (96,26% de acertos) apresentou melhor desempenho na tarefa de classificar em estados as instâncias de teste do satélite virtual XSAT.

O classificador SMO (95,79% de acertos), também apresentou um desempenho bem próximo ao Naive Bayes. Em seguida os dois classificadores KStar e JRip (95,33% de acertos) apresentaram valores de desempenho idênticos em várias métricas, apenas diferindo em valores de erro, em que o classificador KStar obteve índices menores.(vide seção 6.2.4), obtendo no geral um melhor desempenho.

Por fim, o classificador LVQ2_1 (94,86% de acertos) obteve valores de desempenho pouco superior aos valores apresentados pelo classificador J48, que apresentou o pior desempenho em relação a todos os demais algoritmos classificadores (94,39% de acertos)

O resultado da análise estatística realizada, indica que o modelo de classificação preditiva Naïve Bayes, apresentou maior acurácia no diagnóstico do estado do satélite virtual XSAT e , consequentemente, na predição de estados futuros do satélite.

O modelo de classificação gerado pelo classificador estocástico Naïve Bayes, se mostrou o mais adequado para o conjunto de dados, utilizado neste trabalho como estudo de caso. Sendo, portanto, o mais indicado para o desenvolvimento da ferramenta gerador de diagnóstico (vide seção 4.2).

7 CONCLUSÃO

A criação de uma ferramenta de validação de planos de operação de voo a partir de predições de estados dos satélites, para a fase operacional de rotina dos satélites do INPE, pode ser considerada o primeiro passo na melhoria em segurança para planejamento de missões espaciais, seja para atender a uma maior automatização das operações ou atender a uma demanda maior de satélites em órbita.

A predição de estados de satélites, baseou-se em análise matemática, onde o caminho encontrado durante a pesquisa e posteriormente aplicado foram as técnicas de mineração de dados em IA, de forma a construir um modelo adequado para o diagnóstico destes estados. A exatidão no diagnóstico auxilia os especialistas na tomada de decisão em relação às alterações nos planos de voo, visando manter a integridade do satélite.

A seguir são apresentadas as principais contribuições deste trabalho, os caminhos a serem explorados em trabalhos futuros e as possíveis formas de utilização da ferramenta no INPE.

7.1 Principais Contribuições

As contribuições deste trabalho a serem destacadas são apresentadas a seguir:

Para o INPE, este trabalho contribuiu com um modelo empírico que se baseia em dados para realizar diagnóstico do estado de satélites, que pode ser usado *stand-alone* ou em parceria com um simulador. O modelo poderia ser executado em paralelo com o Simulador, ambos fazendo predições. Os resultados dos estados futuros listados por ambos poderiam contribuir para aferir o simulador ou o modelo.

O aspecto de independência da ferramenta, em que para aumentar ou diversificar a base supervisionada, os dados podem ser obtidos diretamente dos testes realizados em laboratório com o satélite antes do lançamento ou do satélite em tempo real, ou por simulação em laboratório.

Este caráter dinâmico da base supervisionada, possibilita ainda que a base seja alterada, por exemplo, em função da necessidade de adequá-la para diferentes satélites ou para atualização do modelo do satélite em operação, quando degradado pelo uso.

O suporte à tomada de decisão, em que a partir da predição de estados do satélite, os especialistas podem avaliar o impacto decorrente da ação de cada plano no comportamento do satélite.

Nas pesquisas realizadas para esta tese, não foi encontrado na literatura relacionada à área espacial, nenhum trabalho que fizessem uso de um modelo preditivo baseado em conceitos de mineração de dados para a realização de tarefas de diagnóstico sobre o estado operacional de um satélite. Todos os dados encontrados vieram de artigos na área, que fazem referência somente ao uso de simuladores (vide seção 3.4).

No entanto, o desenvolvimento de simuladores envolve um trabalho demorado de análise para construção da modelagem e geração da base de conhecimento do satélite, impactando portanto, no custo. A seção 3.4 apresenta alguns artigos com as atuais tendências, relacionados ao desenvolvimento de simuladores. Onde se conclui que o modelo passa a ser para o INPE uma ferramenta interessante para fazer predições futuras, com um custo relativamente baixo se comparado com a construção de um simulador.

A aceitação de artigos oriundos deste trabalho em revista e conferência internacionais mostra o interesse da comunidade da área de missão de operações espaciais nas ideias aqui implementadas, principalmente a

contribuição para a segurança das atividades de controle dos satélites em órbita.

Um artigo apresentando uma visão parcial deste trabalho foi selecionado para apresentação oral na Conferência Internacional sobre Operações Espaciais (*International Conference on Space Operations - S*paceOps 2010) (SOUZA et al., 2010-a), que ocorreu em abril de 2010.

Outro artigo relacionado foi aceito e publicado no *Journal of Aerospace Computing, Information, and Communication* (JACIC), em dezembro de 2010 (SOUZA et al., 2010-b). Com publicação prevista para o próximo mês de maio, mais um artigo foi aceito no *SpaceOps Committee Quarterly,* um periódico em meio eletrônico relacionado com as operações na área espacial.

Outros artigos derivados deste trabalho de tese foram submetidos para o *Journal of Computational Interdisciplinary Sciences* (JCIS) e Revista Controle & Automação da Sociedade Brasileira de Automática, que se encontram atualmente em processo de revisão.

7.2 Trabalhos Futuros

A partir do protótipo da ferramenta desenvolvido neste trabalho para realizar predições dos estados de satélites, transformar por meio de trabalhos futuros, o protótipo em projeto com a perspectiva de torná-lo operacional. E, assim contribuir de forma efetiva para a segurança das operações espaciais para controlar os satélites do INPE, incorporando as operações de rotina procedimentos relacionado à segurança, compatíveis aos utilizados nas missões das principais agências internacionais (BLANQUART et al., 2004).

Neste trabalho, a seleção das técnicas baseou-se naquelas adequadas à classificação binária, portanto com a definição apenas das classes SAFE3 e SAFE2. No entanto, poderia ser realizada em trabalhos futuros uma

diversificação da base supervisionada com uma discretização dos estados operacionais do satélite.

As alterações na base supervisionada devem ser realizadas manualmente. No entanto, por meio de contribuições futuras, poderia ser alcançado o ideal para um dinamismo maior com a inclusão de um módulo que importasse o modelo de um satélite para assim, gerar a base supervisionada.

As técnicas de classificação utilizadas neste trabalho para a construção dos modelos preditivos e seleção do modelo mais adequado, considerou apenas um único classificador induzido pelo conjunto de exemplos de treinamento (com exceção do método de vizinho mais próximo). No entanto, poderia ser explorada em trabalhos futuros, uma melhoria na acurácia da classificação dos estados, por exemplo, agregando as técnicas de predições com múltiplos classificadores. Estas técnicas são conhecidas como método de grupos ou combinação de classificadores. Um método de grupo constrói um conjunto de classificadores básicos a partir dos dados de treinamento e executa a classificação recebendo um voto sobre a predição feita por cada um dos classificadores básicos (TAN et al., 2009).

O protótipo da ferramenta foi desenvolvido para realizar predições dos estados de satélites a partir de estados simulados, quando integrada com o simulador ou a partir de dados reais (vide seção 4.3). Neste trabalho foram utilizados dados simulados, gerados a partir do modelo teórico que descreve o subsistema de suprimento de energia de um satélite virtual (Vide seção 4.2).

No caso da base de dados adquirida a partir de um satélite real, ruídos podem estar presentes entre os dados. Neste caso, a detecção e remoção de ruídos tornam-se parte do pré-processamento dos dados, na qual uma preparação da base de dados poderia ser necessária antes da execução do modelo de classificação. Isto não levaria a uma alteração do modelo utilizado nas predições.

Porém, nos casos em que intencionalmente a anomalia poderia se tornar o foco dentro da base de dados, então passa a ser necessário o uso de algoritmos específicos para detecção destes ruídos (TAN et al., 2009). Nestes casos, tratamentos mais específicos para estes dados poderiam ser explorados em trabalhos futuros.

7.3 Considerações Finais

No INPE, há um enorme esforço no desenvolvimento de simuladores, com o interesse de fortalecer a arte na área de simulação do INPE (BARRETO et al, 2010), considerando tanto o alto custo de aquisição, quanto à exigência de que toda missão espacial deve fazer uso de simuladores.

No entanto, a construção de um simulador de satélites, envolve a forma como seus elementos (subsistemas representados por modelos) interagem, o grau de fidelidade na representação, além da definição de um acoplamento perfeito entre os elementos a serem simulados, entre outros fatores relacionados à construção da modelagem e geração da base de conhecimento do satélite.

Ao ser considerada a complexidade e o custo envolvidos na construção de simuladores, conforme apresentado na seção 3.4, constata-se que o desenvolvimento de simuladores no INPE, também envolve alto custo. Enquanto o protótipo desenvolvido neste trabalho pode vir a se tornar um produto, com custo comparativamente bem menor às cifras atuais que alcança a casa de bilhões de euros (vide Tabela 3.4).

Portanto, sobre o ponto de vista da engenharia de satélite do INPE, a maior contribuição decorrente do desenvolvimento deste trabalho seria a redução drástica de custos com a substituição de um simulador caro (seja em função da aquisição ou do desenvolvimento) por um modelamento, que emule um simulador na tarefa de realizar predições. Este, apresenta-se como um

caminho bastante promissor para redução de custos nas atividades de controle de satélites no INPE.

REFERÊNCIAS BIBLIOGRÁFICAS

AMBROSIO, A. M.; CARDOSO, P. E.; BIANCHI NETO, J. Brazilian satellite simulators: previous solutions trade-off and new perspectives for the CBERS program. In: INTERNATIONAL CONFERENCE ON SPACE OPERATIONS WILL BE HOSTED BY THE, 9TH, 2006, Rome, Italy. **Proceedings...** 2006. p. 7. CD-ROM. (INPE-14068-PRE/9237). Disponível em: <http://urlib.net/sid.inpe.br/mtc-m16@80/2006/08.21.15.01>. Acesso em: 19 abr. 2011.

BARRETO, J. P.; HOFFMANN, L. T.; AMBROSIO, A. M. Using SMP2 standard in operational and analytical simulators. In: INTERNATIONAL CONFERENCE ON SPACE OPERATIONS (SPACEOPS), 11., Apr. 25 – 30, 2010, Huntsville, Alabama, USA. **Proceedings…** American Institute of Aeronautics and Astronautics, 2010. 8 p. (AIAA-2010-2267-433). Disponível em: <http://www.spaceops2010.org>. /Acesso em: 31 de maio de 2010.

BARRETO, J. P. **Uma arquitetura para subsistemas de computação de bordo de um simulador de satélites**. São José dos Campos, Brasil: INPE, 2010. 31 p.

BLANQUART, P.; FLEURY, S.; HERNEK, M.; HONVAULT, C.; INGRAND, F.; PONCET, J. C.; POWELL, D.; STRADY-L_ECUBIN, N.; THEVENOD-FOSSE, P. Software safety supervision on-board autonomous spacecraft. In: EUROPEAN CONGRESS ON EMBEDDED REAL TIME SOFTWARE *(ERTS-2)*, 2., 2004,Toulouse, France. **Proceedings...** Toulouse, 2004. 11 p.

BRAGA, A. P.; CARVALHO, A. C.; LUDERMIR, T. B. **Redes neurais artificiais: teoria e aplicações**. 2a. ed. Rio de Janeiro: LTC, 2007, 248 p.

BRAGA, L. P. V. **Introdução à mineração de dados**. 2. ed. Rio de Janeiro: E-papers, 2005. 212 p.

BRAT, G.; DENNEY, E.; FARELL, K.; GIANNAKOPOULOU, D.; JONSSON, A.; FRANK, J.; BODDY, M.; CARPENTER, T.; ESTLIN, T.; PIVTORAIKO, M. A Robust Compositional Architecture for Autonomous Systems. In: IEEE AEROSPACE CONFERENCE, 2006, Big Sky, MT. **Proceedings…** March 2006.

BIANCHO, A. C.; AQUINO, A. C.; FERREIRA, M. G. V.; SILVA, J. D. S.; CARDOSO, L. A. Multi-agent ground-operations automation architecture. In: INTERNATIONAL CONFERENCE ON SPACE OPERATIONS, 9., 2006, Rome. **Proceedings...** Rome, Italy, June 2006.

CARDOSO, L. S. **Aplicação da tecnologia de agentes de planejamento nas operações de satélites** . 2006. 167p. (INPE-14092-TDI/1075). Dissertação (Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2006.

CARDOSO, L. S.; FERREIRA, M. G. V.; ORLANDO, V. An intelligent system for generation of automatic flight operation plans for the satellite control activities at INPE. In: INTERNATIONAL CONFERENCE ON SPACE OPERATIONS, 9., 2006, Rome, Italy. **Proceedings...** Rome, June, 2006.

CARNIELLO, A.; FERREIRA, M. G. V.; SILVA, J. D. S. **Uma arquitetura de automação de operações solo multi-agente**. São José dos Campos: INPE, 2005. 57 p. (INPE-12911-PUD/169).

CARVALHO, L. A. V. **Datamining a mineração de dados no marketing, medicina, economia, engenharia e administração**. 1. ed. Rio de Janeiro: Editora Ciência Moderna, 2005.

CLEARY, J.G.; TRIGG, L.E. K*: an instance-based learner using an entropic distance measure. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 12., 1995, Tahoe City, California, US . **Proceedings…** Tahoe City, 1995. p.108-114.

CÔME, H. ;IRVINE, M. The XMM simulator - the technical challenges. **ESA Bulletin**, v 96, 1998.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS (INPE). **Resumo do roteiro de desenvolvimento de missões e tecnologias espaciais para o período 2008-2020**. São José dos Campos, Brasil: INPE, 2008, 26 mar. 2008. 31 p. (CPA-068-2008).

DELHAISE, F.; BRU, T. Innovative concepts to reduce costs of mission control and simulator for lisa pathfinder. In: CONFERENCE ON SPACE OPERATIONS, 9., 2006, Rome. **Proceedings…** Rome, 2006.

DEMUTH, H.; BEALE, M.; HAGAN, M. **Neural network toolbox 6**: user`s guide. Natic, MA, USA: The MathWorks, 2008. 907 p.

DING, C. H.; DUBCHAK, I. Multi-class protein fold recognition using support vector machines and neural networks. **Bioinformatics**, v. 17, n. 4, p. 349-358, 2001.

EUROPEAN SPACE AGENCY (ESA). **Space engineering**: ground systems and operations. Part 1: principles and requirements. Netherlands: ESA Publications Division, Apr. 2000. (ECSS-E-70 A).

FAUSETT, L. V. **Fundamentals of neural networks:** architectures, algorithms and applications. New Jersey, USA: Prentice Hall International Inc.,1994.

FERREIRA, C. **Gene expression programming:** mathematical modeling by an artificial intelligence. 2. ed. Berlin: Springer-Verlag, 2006.

FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The WEKA data mining software: an update. **SIGKDD Explorations**, v. 11, n. 1, University Mark Hall, 1999.

FREEMAN, J. A. **Neural networks:** algorithms, applications, and programming techniques. Reading, MA: Addison-Wesley Publishing Company Inc, 1991.

GEISSER, S. **Predictive inference**: an introduction. London: CRC Press, 1993, 280p. (ISBN 0-412-03471-9).

GOLDSCHMIDT, R.; PASSOS, E. **Data mining**: um guia prático: conceitos, técnicas, ferramentas, orientações e aplicações. Rio de Janeiro, RJ: Elsevier, 2005, 261 p.

HAN, J.; KAMBER M. **Data mining:** concepts and techniques. London: Morgan Kaufmann, 2. ed., 2006.

HARRISON, T. **Intranet data warehouse**. São Paulo: Editora Berkeley Brasil, 1998.

HAYKIN, S. **Redes neurais** - princípios e prática. 2. ed. Porto Alegre: Bookman Companhia Editora, 2001. 900 p.

HERDEN, A. **UPKDD:** um processo para desenvolvimento de sistemas de descoberta de conhecimento em banco de dados. Dissertação (Mestrado) - Universidade Estadual de Maringá, 2007.

HOMEM, M. T; PIGNÉDE, M; MERRI, M.; REGGESTAD, V; PIDGEON, A.; MATUSSI, S. The GALILEO simulator: a major step in software technology from single spacecraft to constellation simulators. In: Conference on Space Operations, 9., 2006. Rome. **Proceedings…** Rome, 2006.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 14950:2004**: space systems – unmanned spacecraft operability, ACE/68, Apr 02 2004. p.34. (ISBN 0-580-43607-1).

JOHN G. H.; LANGLEY P. Estimating continuous distributions in bayesian classifiers. In: Conference on Uncertainty in Artificial Intelligence, 11., 1995, San Mateo. **Proceedings…** San Mateo: Morgan Kaufmann, San Mateo, 1995. p. 338-345.

KARRER, D.; CAMEIRA, R.; VASQUES, A.; BENZECRY, M. **Redes neurais artificiais**: conceitos e aplicacoes. - Encontro de Engenharia de Producao da UFRJ - IX Profundão, 9., 2005, Rio de Janeiro: UFRJ, 2005.

KEERTHI, S.S.; SHEVADE, S.K.; BHATTACHARYYA, C.; MURTHY, K.R.K. Improvements to latt's SMO algorithm for SVM classifier design. **Neural Computation**, v.13, n. 3, p. 637-649, 2001.

KOHONEN, T. Learning vector quantization. In: ARBID, M. A. (ed.). **Handbook of brain theory and neural networks**. Cambridge: MIT Press, 1995. p. 537-540.

KOHONEN, T. **Self-organizing maps**. 3. extended ed. Berlin: Springer, 2001.

KUGA, H.; KONDAPALLI, R. Satellite orbit determination: a first-hand experience with the first Brazilian satellite SCD1. In: INTERNATIONAL ASTRONAUTICAL FEDERATION CONGRESS, 44. (IAF), Oct 16-22 1993, Graz, Austria. **Proceedings…** Graz: IAF,1993.

MILANI, P. G., MARTINS NETO, A. F., LOPES, R. V. F., SOUZA, P. N., PALEROSI, A. C., DURÃO, O., CARRARA, V., RICCI, M. C.; FONSECA, I. M., KUGA, H. K. **Uma apresentação dos sistemas, equipamentos, recursos e estudos em guiagem e controle desenvolvidos na divisão de mecânica espacial e controle - DMC do INPE (versão 1)**. São José dos Campos: Instituto Nacional de Pesquisas Espaciais, 2005. 101 p. (INPE-10119-RPQ/246). Disponível *em*: <http://mtc-m16.sid.inpe.br/rep-/sid.inpe.br/iris@1916/2005/07.21.17.51>. Acesso em: 13 jul. 2006.

PLATT, J. C. **Sequential minimal optimization**: a fast algorithm for training support vector machines. [S.l.]: Microsoft Research, 1998. (Technical Report 98-14).

POWELL, D.; THÉVENOD-FOSSE, P. Dependability issues in ai-based autonomous systems for space applications. In: IARP/IEEE-RAS JOINT WORKSHOP ON TECHNICAL CHALLENGE FOR DEPENDABLE ROBOTS IN HUMAN ENVIRONMENTS, 2., October 7-8 2002, Toulouse, France. **Proceedings…** Toulouse: IEEE, 2002. p.163-177.

PRESSMAN, R. S., **Engenharia de software**. 6. Ed. São Paulo: Mc Graw Hill/Nacional, 2006, 752 p.

QUINLAN, R. **C4.5**: programs for machine learning. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

RABENAU, E.; DENIS, M.; JAYARAMAN, P. Implementation of a mission planning system for an interplanetary mission. In: INTERNATIONAL CONFERENCE ON SPACE OPERATIONS, (SPACEOPS), 7., 2002, Huntsville. **Proceedings…** American Institute of Aeronautics and Astronautics, 6 p, AIAA 2002-T3-07. Disponível em: <http://www.spaceops2010.org>./Acesso em: 31 de maio de 2010.

RABENAU, E.; PESCHKE, S. Experience gained with a mission planning system for mission to Mars. In: INTERNATIONAL CONFERENCE ON SPACE OPERATIONS, (SPACEOPS), 8., May 2004, Montreal, Canada. **Proceedings…** American Institute of Aeronautics and Astronautics, 4 p, AIAA 2010-2285. Disponível em: <http://www.aiaa.org/spaceops2004archive>. /Acesso em: 20 de abril de 2007.

RUSSELL, S.; NORVIG, P. **Inteligência artificial**. Tradução da 2a. Edição. [S.l.]: Editora Campus, Brazil, 2004.

SHESKIN, D. J. **Handbook of parametric and nonparametric statistical procedures**. 3. ed. Boca Raton: Chapman&Hall/CRC, 2003. 1232 p. ISBN 1584884401.

SOMMERVILLE, I. **Engenharia de software**. 6 ed. São Paulo: Addison Wesley, 2003.

SOUZA, P. B. **Melhoria do software embarcado em satélites do INPE:** proposta para um passo a mais**.** 2002, 138 p. (INPE-14616-TDI/1195). Dissertação (Mestrado em Computação Aplicada) – INPE, São José dos Campos, 2002.

SOUZA, P. B.; FERREIRA, M. G. V.; SILVA, J. D. S. Decision support tool for prediction of critical data to the satellite integrity. In: INTERNATIONAL CONFERENCE ON SPACE OPERATIONS, (SPACEOPS), 11., Apr. 25 – 30 2010, Huntsville, Alabama, USA. **Proceedings…** American Institute of Aeronautics and Astronautics, 4 p, AIAA 2010-2285. Disponível em: <http://www.spaceops2010.org>. /Acesso em: 31 de maio de 2010.

SOUZA, P. B.; FERREIRA, M. G. V.; SILVA, J. D. S. A tool for prediction of satellite future states. **Journal of Aerospace Computing, Information, and Communication,** v.7, n. 12, p. 406-414., 2010.

SOUZA, P. B.; FERREIRA, M. G. V.; SILVA, J. D. S.; AMBROSIO A.M. Decision support tool for prediction of critical data to the satellite integrity., **SpaceOps Committee Quarterly**, Mar 2011. Aceito.

SUNG, A. H.; MUKKAMALA, S. Identifying important features for intrusion detection using support vector machines and neural networks. In: INTERNATIONAL SYMPOSIUM ON APPLICATIONS AND THE INTERNET TECHNOLOGY, 2003. **Proceedings...** 2003. p. 209-216.

TAN, P-N.; STEINBACH, M.; KUMAR, V. **Introdução ao data mining -** mineração de dados. 1. ed. São Paulo: Editora Ciência Moderna, 2009.

TOMINAGA, J.; SILVA, J. D. S.; FERREIRA, M. G. V. An implementation of a satellite simulator as a fuzzy rulebased inference machine. In: WORKSHOP DOS CURSOS DE COMPUTAÇÃO APLICADA DO INPE, 9. (WORCAP), 2009, São José dos Campos. **Anais...** São José dos Campos: INPE, 2009. On-line. Disponível em: <http://urlib.net/sid.inpe.br/mtc-m18@80/2010/06.21.18.26>. Acesso em: 20 abr. 2011.

VAPNIK, V. N. **The nature of statistical learning theory.** Berlin: Springer-Verlag, 1995.

VAPNIK, V. N. **Statistical learning theory:** inference from small samples. New York: Wiley, 1998.

WERTZ, J.; LARSON, W. **Space mission analysis and design.** 3. ed. [S.l.]: Microcosm, Inc. and Kluwer Academic Publishers, 1999.

WITTEN, I. H.; FRANK E. **Review:** data mining: practical machine learning tools and techniques. 2. ed. San Francisco, Califórnia: Morgan Kaufmann, 2005. 525 p.

GLOSSÁRIO

dimensionalidade dos dados - relacionada ao número de dimensões ou atributos.

inferência - processo de extrair uma conclusão baseada somente no que já se conhece. Em Inteligência Artificial, corresponde a derivação de novas sentenças a partir de sentenças antigas.

medida de distância - medida do tempo de propagação de um sinal eletromagnético emitido por um equipamento instalado, por exemplo, em uma estação terrena de rastreio, no percurso total de ida até um transponder instalado, por exemplo, em um satélite, e retorno até o local de onde foi emitido, após ser retransmitido de volta pelo transponder. Como o sinal eletromagnético se propaga a velocidade da luz, uma vez obtida a medida do tempo de propagação do sinal no percurso entre a estação terrena e o satélite, em um dado instante, basta multiplicar o valor dessa medida pela velocidade da luz para transformá-lo no valor de uma medida da distancia da estação ao satélite, nesse instante.

medida de velocidade - o processo de geração baseia-se na medição do desvio Doppler de freqüência sofrido por um sinal eletromagnético no percurso entre um transmissor e um receptor, que é proporcional à velocidade radial relativa entre o transmissor e o receptor. Esse tipo de medida é dito ser de um caminho, quando o transmissor se encontra no satélite e o receptor em solo, onde é feita a medição. Quando tanto o transmissor quanto o receptor se encontram em solo, e o satélite é equipado com um transponder que apenas recebe o sinal enviado de solo e o retransmite de volta (o que implica em uma medida de velocidade baseada na medição do desvio Doppler total sofrido pelo sinal nos percursos de subida ao satélite e retorno ao solo), a medida é dita ser de dois caminhos.

telecomando - comandos remotos enviados de solo para os subsistemas do satélite.

telemetria - dados transmitidos pelo satélite para o solo que informam o estado dos seus subsistemas.

APÊNDICE A – SAÍDA GERADA PELOS CLASSIFICADORES

```
=== Run information ===
Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     PSS
Instances:    214
Attributes:   12
              SAG
              PSAG
              PPL1
              PPL2
              PAV
              BAT
              VBAT
              QBAT
              CBAT
              IBAT
              DOD
              STATE
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

VBAT <= 47.72: SAFE2 (109.0/3.0)
VBAT > 47.72: SAFE3 (105.0/3.0)

Number of Leaves  :       2

Size of the tree :        3


Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         202               94.3925 %
Incorrectly Classified Instances        12                5.6075 %
Kappa statistic                          0.8879
Mean absolute error                      0.0792
Root mean squared error                  0.2317
Relative absolute error                 15.8356 %
Root relative squared error             46.3509 %
Total Number of Instances              214

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.962     0.073      0.927     0.962     0.944    SAFE3
  0.927     0.038      0.962     0.927     0.944    SAFE2


=== Confusion Matrix ===
```

```
   a   b   <-- classified as
 101   4 |   a = SAFE3
   8 101 |   b = SAFE2

=== Run information ===

Scheme:       weka.classifiers.trees.J48 -U -M 2
Relation:     PSS
Instances:    214
Attributes:   12
              SAG
              PSAG
              PPL1
              PPL2
              PAV
              BAT
              VBAT
              QBAT
              CBAT
              IBAT
              DOD
              STATE
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 unpruned tree
------------------

VBAT <= 47.72: SAFE2 (109.0/3.0)
VBAT > 47.72
|   VBAT <= 47.93
|   |   SAG = SUN: SAFE3 (5.0/1.0)
|   |   SAG = ECL
|   |   |   IBAT <= -20.37: SAFE2 (3.0/1.0)
|   |   |   IBAT > -20.37: SAFE3 (3.0)
|   VBAT > 47.93: SAFE3 (94.0)

Number of Leaves  :       5

Size of the tree :        9

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         201               93.9252 %
Incorrectly Classified Instances        13                6.0748 %
Kappa statistic                          0.8785
Mean absolute error                      0.0669
Root mean squared error                  0.2222
Relative absolute error                 13.3884 %
Root relative squared error             44.4438 %
Total Number of Instances              214

=== Detailed Accuracy By Class ===
```

```
TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.952     0.073      0.926     0.952     0.939    SAFE3
  0.927     0.048      0.953     0.927     0.94     SAFE2

=== Confusion Matrix ===

   a   b   <-- classified as
 100   5 |   a = SAFE3
   8 101 |   b = SAFE2

=== Run information ===

Scheme:       weka.classifiers.neural.lvq.Lvq2_1 -M 1 -C 20 -I 1000 -L
1 -R 0.3 -S 1 -G false -W 0.3
Relation:     PSS
Instances:    214
Attributes:   12
              SAG
              PSAG
              PPL1
              PPL2
              PAV
              BAT
              VBAT
              QBAT
              CBAT
              IBAT
              DOD
              STATE
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

-- Training Time Breakdown --
Model Initialisation Time   : 16ms
Model Training Time         : 0ms
Total Model Preparation Time: 16ms

-- Cass Distribution --
SAFE3 :  10 (50%)
SAFE2 :  10 (50%)

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         203               94.8598 %
Incorrectly Classified Instances        11                5.1402 %
Kappa statistic                          0.8971
Mean absolute error                      0.0514
Root mean squared error                  0.2267
Relative absolute error                 10.2828 %
Root relative squared error             45.3468 %
Total Number of Instances              214

=== Detailed Accuracy By Class ===
```

```
TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.943     0.046      0.952     0.943     0.947    SAFE3
  0.954     0.057      0.945     0.954     0.95     SAFE2

=== Confusion Matrix ===

   a   b   <-- classified as
  99   6 |   a = SAFE3
   5 104 |   b = SAFE2

=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     PSS
Instances:    214
Attributes:   12
              SAG
              PSAG
              PPL1
              PPL2
              PAV
              BAT
              VBAT
              QBAT
              CBAT
              IBAT
              DOD
              STATE
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

Class SAFE3: Prior probability = 0.49

SAG:  Discrete Estimator. Counts =  54 53  (Total = 107)
PSAG:  Normal Distribution. Mean = 807.619 StandardDev = 799.9637
WeightSum = 105 Precision = 1600.0
PPL1:  Normal Distribution. Mean = 6.6667 StandardDev = 116.6667
WeightSum = 105 Precision = 700.0
PPL2:  Normal Distribution. Mean = 0 StandardDev = 1.6667 WeightSum =
105 Precision = 10.0
PAV:  Normal Distribution. Mean = -286 StandardDev = 873.5482
WeightSum = 105 Precision = 577.5
BAT:  Discrete Estimator. Counts =  6 53 49  (Total = 108)
VBAT:  Normal Distribution. Mean = 48.7475 StandardDev = 0.6547
WeightSum = 105 Precision = 0.02476683937823835
QBAT:  Normal Distribution. Mean = 58.4703 StandardDev = 0.7836
WeightSum = 105 Precision = 0.028627450980392172
CBAT:  Normal Distribution. Mean = 1.2 StandardDev = 0.0017 WeightSum
= 105 Precision = 0.01
IBAT:  Normal Distribution. Mean = -3.3108 StandardDev = 17.1374
WeightSum = 105 Precision = 0.3065517241379311
DOD:  Normal Distribution. Mean = 0.0254 StandardDev = 0.0135
WeightSum = 105 Precision = 0.01
```

```
Class SAFE2: Prior probability = 0.51

SAG:  Discrete Estimator. Counts =  55 56  (Total = 111)
PSAG:  Normal Distribution. Mean = 792.6606 StandardDev = 799.9663
WeightSum = 109 Precision = 1600.0
PPL1:  Normal Distribution. Mean = 12.844 StandardDev = 116.6667
WeightSum = 109 Precision = 700.0
PPL2:  Normal Distribution. Mean = 0.367 StandardDev = 2.6842
WeightSum = 109 Precision = 10.0
PAV:  Normal Distribution. Mean = -307.2936 StandardDev = 880.0697
WeightSum = 109 Precision = 577.5
BAT:  Discrete Estimator. Counts =  1 56 55  (Total = 112)
VBAT:  Normal Distribution. Mean = 46.5855 StandardDev = 0.7104
WeightSum = 109 Precision = 0.02476683937823835
QBAT:  Normal Distribution. Mean = 55.8755 StandardDev = 0.8591
WeightSum = 109 Precision = 0.028627450980392172
CBAT:  Normal Distribution. Mean = 1.2 StandardDev = 0.0017 WeightSum
= 109 Precision = 0.01
IBAT:  Normal Distribution. Mean = -3.2427 StandardDev = 18.5531
WeightSum = 109 Precision = 0.3065517241379311
DOD:  Normal Distribution. Mean = 0.0685 StandardDev = 0.0147
WeightSum = 109 Precision = 0.01

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         206               96.2617 %
Incorrectly Classified Instances         8                3.7383 %
Kappa statistic                          0.9252
Mean absolute error                      0.0427
Root mean squared error                  0.1662
Relative absolute error                  8.5375 %
Root relative squared error             33.249 %
Total Number of Instances              214

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.943     0.018      0.98      0.943     0.961    SAFE3
  0.982     0.057      0.947     0.982     0.964    SAFE2

=== Confusion Matrix ===

   a   b   <-- classified as
  99   6 |   a = SAFE3
   2 107 |   b = SAFE2

=== Run information ===

Scheme: weka.classifiers.functions.SMO -C 1.0 -E 1.0 -G 0.01 -A 250007
-L 0.0010 -P 1.0E-12 -N 0 -V -1 -W 1
Relation:     PSS
Instances:    214
Attributes:   12
              SAG
              PSAG
```

```
              PPL1
              PPL2
              PAV
              BAT
              VBAT
              QBAT
              CBAT
              IBAT
              DOD
              STATE
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

SMO

Classifier for classes: SAFE3, SAFE2

BinarySMO

Machine linear: showing attribute weights, not support vectors.

        -0.0615 * (normalized) SAG
 +       0.0615 * (normalized) PSAG
 +       1      * (normalized) PPL2
 +       0.0382 * (normalized) PAV
 +      -0.0615 * (normalized) BAT=DIS
 +       0.0615 * (normalized) BAT=CHG
 +      -3.1254 * (normalized) VBAT
 +      -3.0762 * (normalized) QBAT
 +       0.0232 * (normalized) IBAT
 +       2.7908 * (normalized) DOD
 +       1.9296

Number of kernel evaluations: 5810 (78.583% cached)

Time taken to build model: 0.2 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         205               95.7944 %
Incorrectly Classified Instances         9                4.2056 %
Kappa statistic                          0.9158
Mean absolute error                      0.0421
Root mean squared error                  0.2051
Relative absolute error                  8.4132 %
Root relative squared error             41.0177 %
Total Number of Instances              214

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.952     0.037      0.962     0.952     0.957    SAFE3
  0.963     0.048      0.955     0.963     0.959    SAFE2

=== Confusion Matrix ===
```

```
   a   b   <-- classified as
 100   5 |   a = SAFE3
   4 105 |   b = SAFE2

=== Run information ===

Scheme:       weka.classifiers.lazy.KStar -B 20 -M a
Relation:     PSS
Instances:    214
Attributes:   12
              SAG
              PSAG
              PPL1
              PPL2
              PAV
              BAT
              VBAT
              QBAT
              CBAT
              IBAT
              DOD
              STATE
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

KStar Beta Verion (0.1b).
Copyright (c) 1995-97 by Len Trigg (trigg@cs.waikato.ac.nz).
Java port to Weka by Abdelaziz Mahoui (am14@cs.waikato.ac.nz).

KStar options : -B 20 -M a

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         204               95.3271 %
Incorrectly Classified Instances        10                4.6729 %
Kappa statistic                          0.9065
Mean absolute error                      0.0601
Root mean squared error                  0.1948
Relative absolute error                 12.0209 %
Root relative squared error             38.9689 %
Total Number of Instances              214

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.952     0.046      0.952     0.952     0.952    SAFE3
  0.954     0.048      0.954     0.954     0.954    SAFE2

=== Confusion Matrix ===
   a   b   <-- classified as
 100   5 |   a = SAFE3
   5 104 |   b = SAFE2

=== Run information ===
```

```
Scheme:       weka.classifiers.rules.JRip -F 3 -N 2.0 -O 2 -S 1
Relation:     PSS
Instances:    214
Attributes:   12
              SAG
              PSAG
              PPL1
              PPL2
              PAV
              BAT
              VBAT
              QBAT
              CBAT
              IBAT
              DOD
              STATE
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

JRIP rules:
===========

(VBAT >= 47.73) => STATE=SAFE3 (105.0/3.0)
 => STATE=SAFE2 (109.0/3.0)

Number of Rules : 2


Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         204               95.3271 %
Incorrectly Classified Instances        10                4.6729 %
Kappa statistic                          0.9065
Mean absolute error                      0.0732
Root mean squared error                  0.2132
Relative absolute error                 14.634  %
Root relative squared error             42.643  %
Total Number of Instances              214

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.952     0.046      0.952     0.952     0.952    SAFE3
  0.954     0.048      0.954     0.954     0.954    SAFE2

=== Confusion Matrix ===

   a   b   <-- classified as
 100   5 |   a = SAFE3
   5 104 |   b = SAFE2
```

ANEXO A - ARTIGO PUBLICADO

JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 7, December 2010

A Tool for Prediction of Satellite Future States

Primavera Botelho de Souza*, Mauricio Gonçalves Vieira Ferreira†, and José Demisio Simões da Silva‡
National Institute for Space Research, São José dos Campos, SP 12227-010

DOI: 10.2514/1.52045

The prospect of multiple launches by National Institute for Space Research's satellite program has motivated the development of an application using techniques based on artificial intelligence techniques for automatic generation of flight operation plans to control satellite activities. However, making a critical analysis of these plans before real-world implementation is not possible. We propose a different approach as part of a strategy to validate these plans. This will use a decision support tool based on machine learning concepts to generate prognosis of satellite states for assisting experts in evaluating the performance of the plan. To build the tool, a comparative study of performance between classic data mining classifiers is accomplished to determine the classification model that provides greater accuracy to predict satellite future states.

Nomenclature

$P(Y)$	prior probability for Y
$P(Y\|X)$	posterior probability for Y
X	attribute set
Y	class variable

I. Introduction

THERE is general interest in automating satellite control operations related to the task of controlling multiple satellites in National Institute for Space Research's (INPE) space program. In addition, it is generally accepted that the automation of satellite control activities represents a way of reducing in-orbit satellite maintenance costs. At INPE, autonomous systems to control satellite operations employing artificial intelligence (AI) are being developed to automate ground segment operations.

However, this increased autonomy in satellite control operations can lead to distrust of the automatic control system behavior as compared with that of the well known and routine manual control system. In such cases, these systems still require an improvement in reliability to become operational.

In order to achieve this breakthrough in reliability, predictability, and safety, an AI-based strategy for automatic validation of a flight operation plan generated by a planner is presented. This is an architecture composed of software components, resulting from the combination of verification and validation techniques. As a relevant part of this strategy, a decision support tool is proposed in this paper, to assist experts in evaluating the actions of the plan, aiming at guaranteeing the integrity of the satellite. This tool consists of software using AI techniques aimed at predicting the behavior of critical platform satellite subsystems, such as the power supply subsystem (PSS), directly affected by the actions contained in each flight operation plan.

Received 20 August 2010; accepted for publication 3 October 2010.

* Doctorate student, Applied Computing Postgraduate Program (CAP), Av. Dos Astronautas 1758 Jd. Granja, prima@laser.inpe.br.
† Doctor researcher, Satellite Control and Tracking Center (CRC), Av. dos Astronautas 1758 Jd. Granja, mauricio@ccs.inpe.br.
‡ Doctor researcher, Applied Mathematics and Computing Associated Laboratory (LAC), Av. dos Astronautas 1758 Jd. Granja, demisio@lac.inpe.br.

This paper presents in the following section some concepts related to the automation of the control activities of the satellite in orbit. Section 3 describes the strategy for validation of a flight operation plan, an overview of the software architecture and the tool proposed for validation. Section 4 discusses some data mining techniques of classification for data prediction to design the tool. Section 5 presents a comparative study of performance between classifiers algorithms to determine the classification model that provides greater accuracy to predict satellite future states. Conclusions are presented in Section 6.

II. Satellite Flight Operation Plan

The flight operation plan includes the planning of control operations of space missions and ground segment activities for the planning, execution, and control of the satellite in orbit. Each flight operation plan aims to maintain the satellite in orbit, working to achieve the goals of the mission, containing all the necessary information to control the satellite in orbit, such as: procedures for flight control, procedures for recovery of contingencies, rules, plans, and schedules. All activities included in a flight operation plan have as their starting point the passage of the satellite over the Earth station. The amount of time that a satellite is visible to a given Earth station determines the set of flight operations that should be performed during each pass. Among the activities to control for this period is the sending of commands from the ground (telecommand), and the reception of telemetry which indicates the general state of the satellite.

To meet the growing demand for satellites in orbit and reduce costs significantly, recent studies in AI-based planning have been aimed at the development of tools that automate the tasks of controlling ground operations in INPE. The system, called intelligent planning of flight operation plans (*PlanIPOV*) [1], uses temporal planning AI techniques (temporal planner) applied to the automatic generation of flight operation plans to support the activities of controlling satellites in orbit.

At the same time, the use of automatically generated flight operation plan leads to many doubts. These are partly related to the new technologies involved, but the greatest resistance is related to reliability in the execution of these actions, the predictability and safety of satellites. This increase in autonomy can lead to suspicion about the behavior, often well known and routine. The set of actions contained in a plan acts directly on data critical to maintain of the satellite integrity. Furthermore, depending on the demand for satellites in orbit, a careful validation of these plans can become unviable. In other words, this increased autonomy in satellite control operations still require an improvement in reliability to become operational.

III. Strategy for Validation of Flight Operation Plan

For this advance in reliability, a strategy for validation of flight operation plans is being proposed. The strategy of validation consists of an architecture composed of several software components for validation of an operation plan generated automatically, to be executed in simulation before actual execution (Fig. 1). Designed with the aim of evaluating the impact of the plan from the simulated state of the satellite, the strategy is designed on the basis of appropriate assurance techniques for space systems [2].

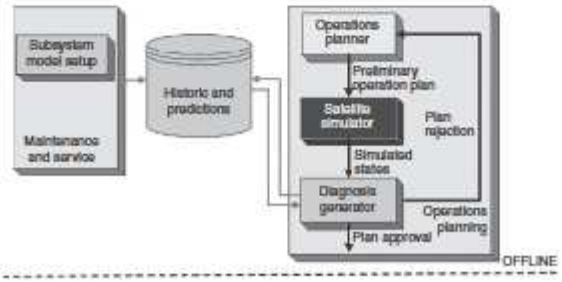

Fig. 1 Validation of flight operations plan: architecture and situation.

As the relevant part of this strategy, a validation tool called the diagnosis generator has being developed to provide prediction about future satellite states from the parameters and critical telemetries, indicating how the general satellite state should evolve, suggesting the adoption, or rejection of the plan.

Through an execution offline of the generated plan by the operations planner [1], each action of the plan is executed and a simulation of the satellite behavior is performed by a satellite simulator [3]. The simulator is based on a virtual satellite, with simplified models, which is also part of the strategy for validation of the generated plan [3].

The simulator returns to the diagnosis generator, parameters, and telemetries, containing the simulated state of the satellite, resulting from the execution of the plan's actions by simulator. As a study case, a simplified model of telemetries, parameters, and operational limits of the PSS of a virtual satellite XSAT is being used. The power supply is a critical subsystem for the satellite integrity [3]. Tables 1–5 present a description of these XSAT parameters and telemetries provided by satellite simulator and used as input data for diagnosis generator.

Upon receiving the data from the XSAT virtual satellite PSS model due to an implementation of the plan's actions, the diagnosis generator tool provides prediction from these parameters and telemetries, generating prognosis of the satellite states indicating how the general state of the satellite will evolve, indicating the impact of the plan in the security level of the satellite operation status.

IV. Techniques for Data Prediction

Computational prediction models are based on probabilistic reasoning over time, interpreting the present and understanding the past and future forecast [4]. The prediction is one of the basic inference tasks in time models, in

Table 1 Virtual Satellite (XSAT) mission operations summary

Payload	Description	Payload data	Data receiving station	Operation criteria	Power consumption
PL1	Optical camera	Satellite imagery for land surface monitor	Image receiving station	Over station, at sunlight or at night if calibration requested	PPL1 ON = 800 W OFF = 100 W
PL2	Data collection subsystem	Environmental data acquired by data collection platforms	Data collection station	Over station or continuous, at sunlight and eclipse	PPL2 ON = 15 W OFF = 5 W

Table 2 XSAT PSS parameters

Identifier	Description	Identifier	Description
SAG	Solar array generator	PAV	Power available to the satellite
PSAG	SAG power	IBAT	BAT charging current
BAT	Battery	VBAT	BAT voltage
QBAT	BAT charge	DOD	BAT depth-of-discharge

Table 3 XSAT power values

Onboard status		Description	Generated power (W)	Consumed power (W)
SAG	SUN	Sunlight—sun illuminated phase	1600	0
	ECL	Eclipse—eclipse phase	0	0
PL1	ON	PL1 operating	0	800
	OFF	PL1 standby	0	100
PL2	ON	PL2 operating	0	15
	OFF	PL2 standby	0	5
SM	–	Service module	0	780

Table 4 XSAT power in each operation mode

Operation mode (defined in the plan)	Onboard status			Power (W)		
	SAG	PL1	PL2	Consumed	Generated	Available
A	SUN	ON	ON	1595	1600	5
B	SUN	ON	OFF	805	1600	795
C	SUN	OFF	ON	115	1600	1485
D	SUN	OFF	OFF	885	1600	715
E	ECL	ON	ON	1595	0	−1595
F	ECL	ON	OFF	1585	0	−1585
G	ECL	OFF	ON	895	0	−895
H	ECL	OFF	OFF	885	0	−885

Table 5 XSAT battery DOD control criteria

DOD (%)	DOD status	Operation status
<15	Low	Safe
15–20	High	Unsafe
>20	Extreme	Forbidden

which the posterior distribution on the future state is calculated, given all the evidence to date. Predictive models have been widely used for building tools to support decision making.

Data mining is a method, in which the ultimate goal is prediction, and represents a process developed to examine routinely large amounts of data collected in search of consistent patterns and systematic relationships between variables. Techniques for finding and describing structural patterns in data have developed within a field known as machine learning, where different styles of learning appear, depending on the data mining application. Those applications where the predictive model requires a judgment needed to inform future decisions, a classification learning scheme takes a set of classified examples (training data) from which it is expected to learn a way of classifying unseen examples (test data) [5].

A classification technique (or classifier) is a systematic approach to building classification models from an input data set. Each technique employs a learning algorithm to identify a model that best fits the relationship between the attribute set (*input*) and class label (*output*) of the input data. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before. Therefore, a key objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previously unknown records [5]. We approach the classical techniques of classification, including decision tree classifiers, Bayesian classifiers, and neural networks.

Following the general approach to solving a classification problem, it was used as a case study, a training data; i.e., a data set with 156 records (instances) of classified examples (Table 6). These input data consist on attribute set of telemetries, parameters, and operational limits of a simplified model of a PSS [3], based on a virtual satellite (see Sec. III), as a result of the action set of a flight operation plan. Each data record is associated with classification of satellite security levels SAFE2 and SAFE3 (STATE class label). For this input data was applied a classifier algorithm, representing each classical classification learning scheme, which each algorithm produces a classification model.

The method used to handle the input data for all classifiers algorithm was one of the methods to random subsampling called cross-validation. We used the 10-fold cross-validation, which the data were segmented into 10 equal-sized partitions. During each run, one of the partitions is chosen for testing, whereas the rest of them are used for training. This procedure is repeated 10 times so that each partition is used for test exactly once.

As mentioned in Sec. III, the diagnosis generator tool should be able to generate data prediction for this satellite subsystem considered critical, based on the classification model that provides greater accuracy to predict satellite future states. So, aiming to provide adequate reasons, the following sections present the main features of these classifiers and associated algorithms used to build the classification models for the diagnosis generator tool.

Table 6 Input data from XSAT operation

DATE/TIME		SAG	PSAG	PPL1	PPL2	PAV	BAT	VBAT	QBAT	CBAT	IBAT	DOD	STATE
19/4/2010	12:30:10	SUN	1600	100	5	715	FULL	50	60	1.2	0	0	SAFE3
19/4/2010	12:30:40	SUN	1600	100	5	715	FULL	50	60	1.2	0	0	SAFE3
19/4/2010	12:31:10	SUN	1600	100	5	715	FULL	50	60	1.2	0	0	SAFE3
19/4/2010	12:31:40	SUN	1600	100	5	715	FULL	50	60	1.2	0	0	SAFE3
19/4/2010	12:32:10	SUN	1600	100	5	715	FULL	50	60	1.2	0	0	SAFE3
19/4/2010	12:32:40	ECL	0	100	5	−885	DIS	50	59.84	1.2	−19.47	0	SAFE3
19/4/2010	12:33:10	ECL	0	100	5	−885	DIS	49.86	59.68	1.2	−19.52	0.01	SAFE3
19/4/2010	12:33:40	ECL	0	100	5	−885	DIS	49.73	59.51	1.2	−19.58	0.01	SAFE3
19/4/2010	12:34:10	ECL	0	100	5	−885	DIS	49.59	59.35	1.2	−19.63	0.01	SAFE3
19/4/2010	12:34:40	ECL	0	100	5	−885	DIS	49.46	59.18	1.2	−19.68	0.01	SAFE3
19/4/2010	12:35:10	ECL	0	100	5	−885	DIS	49.32	59.02	1.2	−19.74	0.02	SAFE3
19/4/2010	12:35:40	SUN	1600	100	5	715	CHG	49.18	59.14	1.2	14.54	0.01	SAFE3
19/4/2010	12:36:10	SUN	1600	100	5	715	CHG	49.28	59.26	1.2	14.51	0.01	SAFE3
19/4/2010	12:36:40	SUN	1600	100	5	715	CHG	49.38	59.38	1.2	14.48	0.01	SAFE3
19/4/2010	12:37:10	SUN	1600	100	5	715	CHG	49.49	59.5	1.2	14.45	0.01	SAFE3
19/4/2010	12:37:40	SUN	1600	100	5	715	CHG	49.59	59.62	1.2	14.42	0.01	SAFE3
19/4/2010	12:38:10	SUN	1600	100	5	715	CHG	49.69	59.74	1.2	14.39	0	SAFE3
19/4/2010	12:50:10	SUN	1600	100	5	715	CHG	49.25	59.23	1.2	14.52	0.01	SAFE3
19/4/2010	12:50:40	ECL	0	800	5	−1585	DIS	49.35	58.93	1.2	−35.33	0.02	SAFE3
19/4/2010	12:51:10	ECL	0	100	5	−885	DIS	49.11	58.77	1.2	−19.82	0.02	SAFE3
19/4/2010	12:51:40	ECL	0	100	5	−885	DIS	48.97	58.6	1.2	−19.88	0.02	SAFE3
19/4/2010	12:52:10	ECL	0	100	5	−885	DIS	48.83	58.43	1.2	−19.93	0.03	SAFE3
19/4/2010	12:52:40	ECL	0	100	5	−885	DIS	48.7	58.27	1.2	−19.99	0.03	SAFE3
19/4/2010	12:53:10	ECL	0	100	5	−885	DIS	48.56	58.1	1.2	−20.05	0.03	SAFE3
19/4/2010	12:53:40	SUN	1600	100	5	715	CHG	48.42	58.22	1.2	14.77	0.03	SAFE3
19/4/2010	12:54:10	SUN	1600	100	5	715	CHG	48.52	58.35	1.2	14.74	0.03	SAFE3
19/4/2010	12:54:40	SUN	1600	100	5	715	CHG	48.62	58.47	1.2	14.71	0.03	SAFE3
19/4/2010	12:55:10	SUN	1600	100	5	715	CHG	48.72	58.59	1.2	14.67	0.02	SAFE3
19/4/2010	12:55:40	SUN	1600	100	5	715	CHG	48.83	58.71	1.2	14.64	0.02	SAFE3
19/4/2010	12:56:10	SUN	1600	100	5	715	CHG	48.93	58.84	1.2	14.61	0.02	SAFE3
19/4/2010	13:44:40	ECL	0	800	15	−1595	DIS	47.25	56.39	1.2	−37.13	0.06	SAFE2
19/4/2010	13:45:10	ECL	0	100	5	−885	DIS	47	56.22	1.2	−20.71	0.06	SAFE2
...	...	...	...	...	...	...	...	...	...	...	...	...	...

A. Decision Tree Classifiers

A decision tree classifier, which is a simple yet widely used classification technique also known as decision tree induction, derives from the simple divide-and-conquer algorithm for producing decision trees [6]. A decision tree contains a root and others internal nodes, beyond attribute test conditions to separate records that have different characteristics.

A decision tree classification learning algorithm was applied to data set (Table 6) to generate the decision tree model for classification of the satellite state. The algorithm chosen for building the decision tree was a well-known and frequently used over the years the C4.5 and J48 as a class for generating a pruned or unpruned C4.5 decision tree [6].

The output of learning algorithm J48 indicating a pruned decision tree model for the training set used with only two (SAFE2 and SAFE3) leaf nodes classification of states (STATE class label). Furthermore, the resulting tree model indicates that the telemetry related with the battery voltage (VBAT) (See Sec. III) is critical to classify the security level of the satellite operation status. Figure 2 shows the decision tree classification model generated used to prognosis of the satellite state for unknown values in a new data record (record test).

B. Bayesian Classifiers

Following a different approach, we consider the relationship between the attribute set and the class variable being nondeterministic. In other words, it is when the class label of a test record cannot be predicted with certainty, even

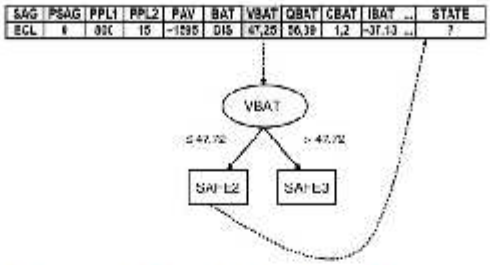

Fig. 2 Classification model based in decision tree applied a test record.

though its attribute set is identical to some of the training examples (see Fig. 2). For solving these classification problems, an approach based on the Bayes theorem is used for modeling probabilistic relationships between the attribute set (X) and the class variable (Y). Consist in a statistical principle for combining prior knowledge of the classes with new evidence gathered from data

$$\frac{P(Y \mid X) = P(X \mid Y P(Y))}{P(X)} \tag{1}$$

Describing how the Bayes theorem was used for classification, let us formalize the classification problem from a statistical perspective. If the class variable (Y) has a nondeterministic relationship with the attributes (X), then we can treat X and Y as random variables and capture their relationship probabilistically using $P(Y \mid X)$. This conditional probability is also known as the posterior probability for Y, as opposed to its prior probability, $P(Y)$. During the training phase, it need to learn the posterior probabilities $P(Y \mid X)$ (Equation (1)) for every combination of X and Y based on information gathered from the training data [7].

The classifier algorithm used to implementation of this model was a naïve Bayes classifier [5], which works using for classification each test record from training data (Table 6), needed to compute the posterior probabilities $P(\text{SAFE2} \mid X)$ and $P(\text{SAFE3} \mid X)$ based on the prior probability obtained for class SAFE3 ($P(\text{SAFE3}) = 67\%$) and the prior probability for class SAFE2 ($P(\text{SAFE2}) = 33\%$). So, the classification is based on the result of the condition: if $P(\text{SAFE3} \mid X) > P(\text{SAFE2} \mid X)$, then the record is classified as SAFE3; otherwise, it is classified as SAFE2.

C. Vector Quantization

Following one more different approach to build a classification model, we are interested in models of artificial neural networks for classification, because it is a nonparametric and nonlinear technique, which allows the mapping of input data associated with output data. Therefore, the output of the network is the class associated to the sample [8].

For representing a model of artificial neural networks for classification, we choose learning vector quantization (LVQ) networks, which define a family of adaptive algorithms for quantifying vector, originally proposed by Kohonen [9]. LVQ networks define methods for supervised training employing a self-organizing network approach which uses the training vectors to recursively tune placement of competitive hidden units that represent categories of the inputs. Once the network is trained, an input vector is categorized as belonging to the class represented by the nearest hidden unit [8].

The classifier algorithm used to implementation of LVQ networks was the LVQ2_1 classifier algorithm [5]; it consists on iterative algorithm, whose basic principle is to reduce the distance of the input vectors in the same class, and to move away input vector in the wrong class. The classes distribution obtained as output were SAFE3: 16 (80%) and SAFE2: 4 (20%) for the input vectors representing 12 attributes.

In the next section, a performance evaluation of each classification model generated and comparison between three classifiers is accomplished based on performance metrics such as accuracy and error rate values, being the results presented and discussed.

All the classifiers algorithms used are an integral part of the Waikato environment for knowledge analysis (WEKA), a suite of machine learning software was written in Java [6]. WEKA is free software available under the GNU General Public License (GPL), aiming at adding algorithms from different approaches in the subarea of AI, dedicated to the study of learning by machines [6].

V. Results and Discussion

The performance evaluation of a classification model is based on the counts of test records correctly and incorrectly predicted by the model. These counts are tabulated in a table know as confusion matrix. Table 7 depicts the confusion matrix of classifiers: J48, naïve Bayes, and LVQ2_1.

Each entry e_{ij} in Table 7 denotes the number of records from class SAFE3 predicted to be class SAFE2. For instance, e_{ji} is the number of records from class SAFE2 predicted incorrectly predicted as SAFE3. Thus, based on the entries in the confusion matrix, the total number of correct predictions and total number of incorrect predictions of each model was calculated and presented on Table 8. From these matrix elements is possible also get the performance metrics such as accuracy for each model and the error rate values (Table 8).

The most classification algorithms seek models that attain the highest accuracy, or equivalently, the lowest error rate. Then, evaluating in terms of percentages, the accuracy and error rate values for each classifier, we can say that the classifier naïve Bayes shows the better accuracy value (95%) and minor error rate (5%) followed of the decision tree classifier (91%) and (9%). The worse accuracy and error rate associated was the neural classifier LVQ2_1 (86%) and (13%).

Other key measure for evaluating classifiers is Kappa statistics or Kappa coefficient. A measure of agreement used in nominal scale, that gives us an idea of how much the observations deviate from those expected due to chance, giving us so how legitimate interpretations are. This observer disagreement is indicated by how observers classify individual subjects into the same category on the measurement scale. During in run, each classifier assigned items

Table 7 Confusion matrix of tree classifiers: J48, naïve Bayes, and LVQ2_1

	Class = SAFE3	Class = SAFE2	Total
J48			
Class = SAFE3	$e_{ii} = 99$	$e_{ij} = 6$	105
Class = SAFE2	$e_{ji} = 8$	$e_{jj} = 43$	51
Total	107	49	156
Naïve Bayes			
Class = SAFE3	$e_{ii} = 99$	$e_{ij} = 6$	105
Class = SAFE2	$e_{ji} = 2$	$e_{jj} = 49$	51
Total	101	55	156
LVQ2_1			
Class = SAFE3	$e_{ii} = 96$	$e_{ij} = 9$	105
Class = SAFE2	$e_{ji} = 12$	$e_{jj} = 39$	51
Total	108	48	156

Table 8 Accuracy and error rate performance metrics for each classifier

Classifiers	Accuracy (%)	Error rate (%)
J48	91.02	8.97
Naïve Bayes	94.87	5.13
LVQ2_1	86.53	13.46

Table 9 Kappa coefficient values provided by the three classifiers

Classifiers	Kappa	Agreement
J48	0.7940	Good
Naïve Bayes	0.8858	Very good
LVQ2_1	0.6894	Good

to one of the two classes SAFE3 and SAFE2, but the number of individuals assigned to each class by classifier are disagree (see Table 7).

The values of Kappa are interpreted as the maximum of 1 when agreement is perfect, 0 when agreement is no better than chance, and negative values when agreement is worse than chance. Other values can be roughly interpreted as [10]:

1) insufficient agreement (<0.20)
2) fair agreement (0.20–0.40)
3) moderate agreement (0.40–0.60)
4) good agreement (0.60–0.80)
5) very good agreement (0.80–1.00)

Kappa measures the percentage of data values in the main diagonal of the confusion matrix (Table 7) and then adjusts these values for the amount of agreement that could be expected due to chance alone. In Table 9, the Kappa coefficient values of each classifier are reported and interpreted.

The Kappa coefficient value obtained form naïve Bayes classifier presented a perfect agreement, whereas the other classifiers present a good agreement. Overall, the classifier algorithm naïve Bayes showed better results, indicating the Bayesian method as the best classification model generated to predict satellite future states.

VI. Conclusion

This paper presented a comparative study of performance between algorithms classifiers, selected to represent each of the three different methods for classification models generation. The use of methods with different approaches to learning proved to be a significant contribution, in selecting the most appropriate classification model to the set of application data. In addition, the statistic became possible to determine with certainty, which the classification model that provides greater accuracy for predicting satellite future states, once the classification model is the core of the tool designed to assist experts in impact analysis of each plan's action on the satellite behavior to decision support making.

In this study, we simulated data generated from the theoretical model that describe the PSS of a virtual satellite. But, where the data are acquired from a real satellite, which the data may contain missing or anomalies data, the detection and removal of anomalies is often part of the preprocessing of data before the classification model execution. An exception may occur when the anomaly becomes the focus in an application database, requiring the use of specific algorithms to detect anomalies that could be included in future works.

Acknowledgment

The authors wish to thank CNPq/MCT (an institute of the Brazilian goverment) for supporting this research through grant 142063/2006-1.

References

[1] Cardoso, L. S., Ferreira, M. G. V., and Orlando, V., "An Intelligent System for Generation of Automatic Flight Operation Plans for the Satellite Control Activities at INPE," *Proceedings of the 9th International Conference on Space Operations*, AIAA, Inc., Reston, Virginia, VA, June 2006.

[2] Blanquart, P., Fleury, S., Hernek, M., Honvault, C., Ingrand, F., Poncet, J. C., Powell, D., Strady-Lécubin, N., and Thevenod-Fosse, P., "Software Safety Supervision On-board Autonomous Spacecraft," *2nd European Congress on Embedded Real Time Software* (*ERTS*-2), Toulouse, France, 2004, 11p.

[3] Toninaga, J., Ferreira, M. G. V., and Silva, J. D. S., "An Implementation of a Satellite Simulator as a Fuzzy Rule-Based Inference Machine," *WORCAP IX*, INPE, São José dos Campos, Brazil, 2009.

[4] Russell, S., and Norvig, P., *Inteligência Artificial*, Tradução da 2nd ed., Editora Campus, Brazil, 2004.
[5] Tan, P.-N., Steinbach, M., and Kumar, V., *Introduction to Data Mining*. Addison-Wesley, Reading, MA, 2005.
[6] Witten, I. H., and Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, 1st ed., The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, San Mateo, CA, 1999.
[7] Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H., "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, Vol. 11, No. 1, 1999, pp. 10–18.
[8] Haykin, S., *Redes Neurais: Princípios e Prática*, 2nd ed., Makron Books, São Paulo, 2001, 900 pp.
[9] Kohonen, T., *Self-Organizing Maps*. 3rd, extended ed., Springer, 2001.
[10] Sheskin, D. J. *Handbook of Parametric and Nonparametric Statistical Procedures*, 3rd ed., Chapman & Hall/CRC, Boca Raton, FL, 2003, ISBN 1584884401.

Reinaldo Perez
Associate Editor

ÌNDICE POR ASSUNTO