UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Adinarte Correa da Silva

Modelo para análise de dados de gerência de redes utilizando técnicas de KDD

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de mestre em Ciência da Computação

Orientadora: Prof. Dra. Elizabeth Specialski

Florianópolis-SC, Dezembro de 2002

Modelo para análise de dados de gerência de redes utilizando técnicas de KDD

Adinarte Correa da Silva

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Fernando A. Ostuni Gauthier, Dr. (Coordenador do Curso

Banca Examinadora

Elizabeth Specialski, Dra. (orientadora)

João Bosco da Mota Alves, Dr.

Luiz Fernando J. Maia, Dr.

**"Não somos o que deveríamos ser; não somos o que iremos ser.
mas graças a Deus, não somos o que éramos."
(Martin Luther King)**

AGRADECIMENTOS

Gostaria de agradecer a minha Orientadora Prof. Dra Elizabete Specialski, pela paciência, pelos valiosos conselhos e demonstração de amizade.

Aos professores membros da banca, Dr. João Bosco da Mota Alves e Dr. Luiz Fernando J. Maia, pelo incentivo e disponibilidade.

Um enorme agradecimento a minha esposa Diva, pelo inesgotável e constante apoio e incentivo recebido durante todo este período, e pela incrível paciência e atenção que sempre tive. Estendo o agradecimento aos meus cunhados Ademir e Odete, pela força, carinho e auxilio.

Aos meus amigos Moacir, André e Clausa, que não mediram esforços para me auxiliar quando foi necessário.

Aos professores, colegas de mestrado e trabalho, que tiveram participação na realização do trabalho.

A Deus como fonte de inspiração quanto tudo parecia perdido

SUMÁRIO

LISTA DE SIGLAS

CMIP - Common Management Information Protocol

CMISE - Common Management Information Services

HTML - HyperText Markup Language

IA - Inteligência Artificial

IEE - The Institution of Electrical Engineers

IEEE - Institute of Electrical and Electronics Engineers

IETF - International Engineering Task Force

IP - Internet Protocol

ISO - International Organization for Standardization

KDD - Knowledge Discovery in DataBase

KDSE - Knowledge Discovery Support Environment

LAN - Local Area Networks

MIB - Management Information Base

ML - Machine Learning

OSI - Open Systems Interconnection

PDU - Protocol Data Unit

SGBD - Sistema Gerenciador de Banco de Dados

SNMP - Simple Network Management Protocol

SQL - Structured Query Language

TCP - Transport Control Protocol

TMN - Telecommunications Management Network

UDP - User Datagram Protocol

WAN - Wide Area Networks

WWW - World Wide Web

LISTAS DE FIGURAS

RESUMO

O presente trabalho propõe um modelo de apoio à análise de dados, coletados pelo protocolo SNMP em uma rede de computadores, independente da tecnologia de rede ou tipo de MIB utilizada. O modelo é baseado no processo KDD, segundo FAYAD e BRACHMAN, o qual é caracterizado por atividades distribuídas em etapas iterativas e interativas. O modelo foi desenvolvido e testado utilizando Regras de associação (algoritmo APRIORI), porém, permite que outras técnicas de datamining possam ser utilizadas. Para a validação do Modelo foi utilizado o software WEKA.

Palavras chave: *Regras de associação, SNMP, DataMining, KDD.*

ABSTRACT

The present work proposes a support model to the analysis of data, collected by the protocol SNMP in a network, independent of the network technology or type of MIB. The model is based on the process KDD, according to FAYAD and BRACHMAN, which it is characterized by activities distributed in stages iterative and interactive. The model was developed and tested using association Rules (algorithm APRIORI), even so, it allows that other datamining techniques can be used. For the validation of the Model the software used was WEKA.

Key words, SNMP, DataMining, KDD.

1 - INTRODUÇÃO

As redes de computadores estão se tornando um recurso imprescindível em instituições comerciais, centros de pesquisas e instituições educacionais. A sua utilização permite o compartilhamento de aplicativos, periféricos e banco de dados independente da distância entre os recursos.

Em uma empresa, os sistemas de informações indispensáveis são desenvolvidos, desde o seu projeto original, sobre a plataforma de alguma rede, confiantes da integridade da mesma, para que processos vitais não sejam prejudicados. Da mesma forma, outros sistemas de grande difusão no mercado tais como o correio eletrônico e os sistemas bancários somente são possíveis através das redes de computadores.

Segundo Harnedy [30], "as redes e recursos de computação distribuídos estão se tornando vitais para a maioria das organizações. Sem um controle efetivo, os recursos não proporcionam o retorno que a corporação requer".

As instituições que já utilizam as redes de computadores estão se beneficiando do avanço contínuo desta tecnologia. No entanto, o número de estações conectadas vem aumentando, o que pode causar a heterogeneidade de equipamentos. Em conjunto com estes fatores aparece a implementação de softwares que consomem mais recursos, aumentando os problemas de comunicação. Ainda segundo Harnedy [30], o contínuo crescimento da rede em termos de componentes, usuários, interfaces, protocolos e fornecedores, ameaça o gerenciamento com perda de controle sobre o que está conectado na rede e como os recursos estão sendo utilizados.

Devido à importância das redes de computadores para uma organização, uma alta disponibilidade é requerida. Isto significa que o nível de falhas e de degradação de desempenho, considerado aceitável, está cada vez menor, podendo chegar a zero em alguns sistemas críticos.

Quando um projeto de rede é desenvolvido, é necessário que o mesmo seja baseado nas características computacionais de cada local, pois o fator econômico tem

grande influência na aprovação do projeto. Além disso, não se pode prever todas as implementações futuras, mas, sim prever apenas uma margem de crescimento. Isto é corroborado por Redisul[22], que estabelece esta margem em 30%. Os usuários, conforme Harnedy [30], esperam uma melhoria dos serviços oferecidos (ou no mínimo, a mesma qualidade), quando novos recursos são adicionados ou quando são distribuídos.

A atividade de gerenciamento de redes foi a solução encontrada para antever ou mesmo corrigir possíveis problemas em tempo hábil. Conforme Franceschi [5], as redes são equipamentos que operam de forma distribuída e estão sujeitos a diversos tipos de falhas:

- Falhas nos equipamentos devido à ação do tempo, como umidade e calor excessivo;
- Falhas operacionais ou de uso indevido dos equipamentos;
- Falha de sobrecarga; etc.

Além da gerência de falhas existem outras preocupações para aqueles que utilizam e administram os serviços de rede. A ISO (International Organization for Standardization) define cinco áreas funcionais de gerenciamento além da gerência de falhas, são elas: gerência de configuração, gerência de desempenho, gerência de contabilização e a gerência de segurança. FRANCESCHI[5].

O universo de objetos gerenciáveis é amplo e cada categoria de objetos possui suas características de gerência, tornando volumosa a base de dados coletada para análise. Devido ao número de informações (registros) e à variedade de tipos de informações (campos), dificultando o entendimento dos dados. Para que os dados coletados se transformem em informações úteis para o gerente da rede, é necessário utilizar técnicas que facilitem o entendimento dos dados.

De acordo com PIATESKI [33], os dados produzidos e armazenados em larga escala pelas diversas gerências tornam-se inviáveis para a leitura, bem como para análise através de métodos manuais tradicionais, onde, o especialista testa sua hipótese

contra a base de dados. Por outro lado, sabe-se que grandes quantidades de dados equivalem a um maior potencial de informação. Diante deste cenário, surge a necessidade de encontrar uma técnica que explore esta grande base de dados e consiga extrair o conhecimento embutido nos dados.

O processo capaz de descobrir conhecimento em banco de dados chama-se *Knowledge Discovery Database* (KDD). O processo de KDD foi proposto em 1989, referindo-se às etapas que produzem conhecimentos a partir dos dados e, principalmente, à etapa de mineração dos dados, que é a fase que transforma dados em informações FAYAD [17]. Este processo envolve encontrar e interpretar padrões nos dados, de modo iterativo e interativo, através da repetição dos algoritmos e da análise de seus resultados.

O Segundo Andrassoyvá [34], o processo de KDD, apesar de possuir diferentes abordagens no tratamento ou na ordem de execução das tarefas, utiliza mecanismos para preparação, apresentação dos dados e técnicas de Datamining. Conforme Carvalho[27], "A utilização de técnicas automáticas de exploração de grandes quantidades de dados de forma a descobrir novos padrões e relações que, devido ao volume de dados, não seriam facilmente descobertos a olho nu pelo seres humanos."

1.2 - OBJETIVOS

O Datamining desponta como uma tecnologia capaz de cooperar amplamente na busca do conhecimento embutido nos dados. Geralmente, a base de dados não está devidamente organizada para a aplicação da técnica, sendo necessária a criação de um modelo de análise de dados. Este modelo, além de incluir a mineração dos dados (Datamining), deve se preocupar, também, com a preparação da base de dados e da análise dos resultados, permitindo gerar um resultado final claro e objetivo.

Sendo assim, o objetivo deste trabalho é propor um modelo de análise de dados relacionados à gerência de redes, independente da tecnologia utilizada, classe de objetos gerenciada e da área funcional pertencente. O modelo proposto é baseado em técnicas

de KDD, e deve permitir encontrar associações que possam indicar a ocorrência de algum problema de rede.

Considera-se que a busca por estas associações seja realizada em uma base de dados que possui informações teoricamente distintas em um grande volume de dados.

O modelo proposto visa possibilitar executar o processo de dataminig a partir de qualquer informação pertinente a gerencia de redes, objetivando encontrar informações ocultas na base de dados que venha a solucionar algum problema existente na rede ou antever algum problema potencial futuro.

O modelo possui como principais características:

- **Flexibilidade** – As etapas do modelo possibilitam que sejam adequadas a processar informações não padronizadas;
- **Escalabilidade** – O modelo não foi desenvolvido para Equipamentos, Sistemas operacionais ou softwares específicos, desta forma, dependendo da situação, o modelo permite que seja utilizado em qualquer tipo de ambiente computacional;
- **Modularidade** – Permite a utilização de diversos algoritmos de Dataminig diferentes, dependendo da situação imposta.

1.3 - Organização do trabalho

O trabalho está organizado em seis capítulos. O capítulo 2 contém a revisão bibliográfica. No capítulo 3 o modelo proposto é apresentado. No capítulo 4 é descrito o estudo de caso realizado com o programa WEKA. O capítulo 5 apresenta a conclusão e sugestões para trabalhos futuros e no capítulo 6 a bibliografia utilizada é listada.

2 – REVISÃO DE LITERATURA

2.1 – GERÊNCIA DE REDES DE COMPUTADORES

Neste capítulo serão abordados os conceitos básicos referentes às arquiteturas de gerenciamento, os protocolos para transferência de informação de gerenciamento e os modelos de informação utilizados pelas arquiteturas.

Segundo SPECIALSKI [35], todos os equipamentos da rede que fazem parte do sistema de gerenciamento possuem um conjunto de softwares destinados às tarefas de coletar informações sobre as atividades relacionadas à rede, armazenar estatísticas localmente e responder aos comandos do centro de controle da rede. Estes nodos são referenciados como AGENTES. No mínimo um hospedeiro da rede é designado para as tarefas de controlador da rede (GERENTE) e possui uma coleção de software chamada Aplicação de Gerenciamento da Rede. A aplicação de gerenciamento da rede possui uma interface que permite a um usuário autorizado gerenciá-la.

2.1.1 – Modelos de gerenciamento

Segundo SPECIALSKI [35], um sistema de gerenciamento de rede apresenta um conjunto de ferramentas integradas que permitem monitorar e controlar a rede. A autora ressalta, ainda, que o sistema deve apresentar uma interface amigável onde o operador disponha de um conjunto de comandos para executar as tarefas de gerenciamento. Também é necessário que a maioria do hardware e software a serem utilizados para o gerenciamento da rede já esteja incorporado nos equipamentos de usuários existentes.

A gerência de redes de computadores é uma tarefa complexa, envolvendo a monitoração e o controle de diferentes componentes de hardware e software, tais como UDUPA [36]:

- Equipamentos de nível físico e de enlace;
- Componentes de computadores (processadores, impressoras, etc.);
- Componentes de interconexão (*bridges*, roteadores, *switches*, etc.);

- Hardware de telecomunicações (modems, multiplexadores, etc.);
- Sistemas operacionais;
- Aplicações e ferramentas de softwares;
- Softwares de interconexão (presentes nas *bridges*, roteadores, etc.);
- Aplicações cliente/servidor (servidores de banco de dados, servidores de arquivos, servidores de impressão, etc.);
- Software de comunicação de dados.

Atualmente a gerência de redes pode ser realizada de acordo com três modelos abertos, a saber:

- A arquitetura de gerenciamento de acordo com o modelo OSI;
- A arquitetura de gerência da Internet ou SNMP;
- A arquitetura de gerência de redes de telecomunicações – TMN.

Para FRANCESCHI [5], cada uma dessas abordagens possui um protocolo para definir quais informações deverão ser coletadas, como serão obtidas e quais as operações que poderão ser realizadas. Conforme SPECIALSKI [35], um software genérico, utilizando qualquer uma das abordagens acima é composto por:

- Elementos gerenciados;
- Agentes;
- Gerentes;
- Bancos de Dados de Informações;
- Protocolos para troca de informações de gerenciamento;
- Interfaces para programas aplicativos;
- Interfaces com o usuário.

Independentemente da arquitetura de gerenciamento utilizada, os sistemas de gerenciamento abordam um subconjunto de funcionalidades que podem ser classificados em uma das cinco áreas funcionais definidas pela ISO: gerência de falhas, gerência de configuração, gerência de contabilização, gerência de desempenho e gerência de segurança. FRANCESCHI[5].

Independente do modelo implementado, a homogeneidade da solução oferecida por um único fabricante, facilitaria a análise final dos dados. Entretanto, com as constantes ampliações naturais da rede e o avanço tecnológico dos dispositivos, a heterogeneidade é inevitável.

2.1.2 – Áreas Funcionais

O objetivo do gerenciamento OSI é o de resolver os problemas relativos as anormalidades que podem ocorrer em uma rede de computadores, com o intuito de facilitar o gerenciamento, estas anomalias foram categorizadas em cinco áreas distintas, chamadas: **Áreas Funcionais de Gerência.**

2.1.2.1 - GERÊNCIA DE FALHAS

Gerência de falhas é a área definida para tratar especificamente de problemas ou falhas na rede, localizando o problema, isolando-o e solucionando-o, quando for possível FRANCESCHI [5].

Segundo SPECIALSKI [35], "Falhas não são o mesmo que erros. Uma falha é uma condição anormal cuja recuperação exige ação de gerenciamento. Uma falha normalmente é causada por operações incorretas ou um número excessivo de erros".

É possível utilizar algumas ferramentas para realizar o gerenciamento de falhas, LEINWAND [12]:

- ***Sistema de gerenciamento de redes*** – Entre outras funcionalidades pode sinalizar para o administrador, quando algum dispositivo falhar e armazenar estas informações de possíveis falhas em um LOG, para análise futura;
- ***Analisadores de Protocolo*** - Analisando os protocolos trafegados na rede é possível se identificar se existe algum dispositivo que esteja trafegando algum tipo de protocolo não padronizado na concepção da rede;
- ***Verificador de Cabo*** – permite identificar possíveis falhas no cabeamento da rede, desde cabos interrompidos até perda de pacotes causados por algum tipo de diafonia ou ruídos;
- ***Sistemas Redundantes*** – Equipamentos de rede idênticos (por exemplo: switches, RAIDs, roteadores) que permanecem "stand alone", até que algum

equipamento entre em estado de falha. A partir deste momento, o equipamento entra em funcionamento. Este processo fica transparente para o usuário, não ocasionando parada de rede ou segmento.

2.1.2.2 - GERÊNCIA DE CONFIGURAÇÃO

O gerenciamento de configuração está relacionado com a inicialização da rede e com uma eventual desabilitação de parte ou de toda a rede. Também está relacionado com as tarefas de manutenção, adição e atualização de relacionamentos entre os componentes e do status dos componentes durante a operação da rede. SPECIALSKI [35]

A gerência de configuração através de suas variáveis pode documentar as configurações iniciais ou valores default dos atributos de todos os dispositivos configuráveis na rede.

O gerente da rede deve ser capaz de, inicialmente, identificar os componentes da rede e definir a conectividade entre eles. Também deve conseguir modificar a configuração, em resposta à avaliações de desempenho, recuperação de falhas, problemas de segurança, atualização da rede ou a fim de atender à necessidades dos usuários. Segundo STALLINGS [4], o gerenciamento de configuração inclui as funções que seguem:

- Visualizar as configurações definidas;
- Configurar e modificar valores dos atributos dos objetos gerenciáveis;
- Definir e Modificar relacionamentos;
- Inicializar e finalizar operações de rede;
- Distribuir Software;
- Examinar de valores e relacionamentos;
- Emitir Relatório do status da Configuração.

O gerenciamento de configuração consiste de três etapas segundo LEINWAND [12]:

- Obter as informações sobre o ambiente de rede corrente;
- Utilizar os dados obtidos para modificar a configuração dos dispositivos de redes;
- Armazenar os dados, documentando a configuração atual dos dispositivos.

A gerência de configuração disponibiliza dados importantes para as outras áreas de gerenciamento pois alguns dos problemas de falhas, desempenho e segurança, ocorridos em redes de computadores, podem ser ocasionados pela má configuração dos dispositivos.

2.1.2.3 - GERÊNCIA DE DESEMPENHO

O gerenciamento do desempenho de uma rede consiste na monitoração das atividades da rede e no controle dos recursos através de ajustes e trocas. Conforme SPECIALSKI [35], algumas das questões relativas ao gerenciamento do desempenho são:

- Qual éIdentificar o nível de capacidade de utilização de um recurso
- O Verificar se o volume de tráfego é excessivo
- O Identificar se o throughput foi reduzido e se o nível é aceitável
- Existem Investigar a existência de gargalos
- O Monitorar o tempo de resposta

Para realizar a gerência de desempenho em uma rede, é necessário coletar algumas variáveis de rede definidas na MIB dos agentes, tais como:

- O número de pacotes trafegados;
- O número de pacotes com erros;
- A porcentagem de colisão de pacotes;
- A Matriz de comunicação (hosts de origem x hosts de destino dos pacotes);
- A identificação do segmento ao qual pertence cada pacote trafegado.

Segundo LEINWAND [12], as etapas envolvidas nesse processo são:

- Coletar os dados de utilização dos equipamentos e enlaces de rede;

- Analisar os dados relevantes, de modo a identificar tendências de utilização elevadas;
- Estabelecer limiares de utilização e usar procedimentos de simulação para identificar como a rede pode ser alterada para maximizar sua performance.

Na análise dos dados relevantes, é possível analisar, mediante cálculo, outras informações importantes, como:

- Detectar falhas de algum componente da rede;
- Prever expansões da rede;
- Permitir a segmentação da rede;
- Em nível de Custeio da rede, verificar qual área utiliza mais a rede (ligando Nº de pacotes trafegado X Origem/Destino X Grupo de usuários).

A gerência de desempenho é umas das áreas funcionais que mais necessitam atenção para gerenciá-la, pois muitos fatores diferentes em uma rede podem causar queda de performance.

2.1.2.4 - GERÊNCIA DE CONTABILIZAÇÃO

O gerenciamento de contabilização diz respeito à análise de como os recursos de rede disponíveis são utilizados e os custos envolvidos nesta utilização. Envolve o rastreamento e geração de relatórios da utilização por usuário ou grupo de usuários, a fim de estabelecer métricas, verificar cotas, determinar custos e cobrar dos usuários. Permite um aumento do entendimento da utilização da rede, o que pode auxiliar também no desenvolvimento de uma rede mais produtiva MELCHIORS [64]

SPECIALSKI [35], cita que, mesmo nos casos onde não exista cobrança sobre a contabilização da rede, o administrador deverá estar habilitado a controlar os recursos de rede. Isto evita que o usuário abuse de seus privilégios de acesso e assim monopolize a rede, prejudicando os demais. Também é possível conhecer os procedimentos dos usuários, vislumbrando a utilização futura de recursos para poder planejar o crescimento da rede. Também é possível utilizar esta função para dividir o custo da área de

informática, entre os departamentos, conforme a utilização de recursos da rede efetivamente consumidos.

Segundo STALLINGS [4], alguns exemplos de recursos que podem ser objeto de contabilização:

- Linhas de Comunicação – LANs, WANs, linhas alugadas, comunicação dial-up e sistema PABX;
- Computadores – Estações de trabalho e servidores instalados;
- Sistemas e Software – Sistemas, aplicativos e utilitários instalados nos servidores e Estações de trabalho;
- Services Serviços – Incluem todas as comunicações comerciais e informações sobre serviços disponibilizados na rede para os usuários.

A gerência de contabilização pode ser vista, portanto, como um mecanismo para a identificação de demandas por recursos, possibilitando um planejamento pró-ativo de expansões da rede e de equipamentos de rede.

2.1.2.5 - GERÊNCIA DE SEGURANÇA

Esta gerência envolve o controle de pontos de acesso às informações sensíveis em uma rede. Estas informações geralmente são armazenadas nos equipamentos de rede e devem ser mantidas em segurança e não disponíveis para todos os usuários. O gerenciamento de segurança é responsável pelo monitoramento e relatório das tentativas de intrusão na rede ocorridas com sucesso ou não. Também inclui facilidades para procedimentos de controle, tais como: LEWIS [95]

- Estabelecer políticas para a utilização da rede;
- Estabelecer e manter chaves criptografadas e códigos de autorização;
- Manter um registro (log) do acesso à rede;
- Prevenir e relatar acessos não autorizados;
- Iniciar procedimentos de investigação em resposta a acessos não autorizados;
- Detectar e prevenir vírus de computadores.

Segundo BRISA [3], a norma de referência da Arquitetura de Segurança do Modelo OSI, trata exclusivamente da segurança dos canais de comunicação que permitem que os sistemas nas extremidades destes canais de comunicação se comuniquem entre si, de modo seguro e protegido.

Para isso, ela define cinco serviços de segurança:

- Autenticação tanto de entidades pares como origem dos dados (authentication);
- Controle de acesso aos recursos da rede (access control);
- Confidencialidade dos Dados (confidenciality);
- Integridade dos dados (integrity);
- A não rejeição ou não repudiação (non repudiation).

Cada um destes serviços apresenta um conjunto de procedimentos que devem ser seguidos a fim de garantir a segurança necessária para cada tipo de aplicação existente na rede. Dependendo da aplicação, pode ser requerida a implantação de apenas um dos serviços ou então, de um conjunto destes serviços.

2.1.3 – Modelo de Informação

As áreas funcionais de gerenciamento foram definidas como uma forma de quebrar a complexidade do gerenciamento de uma rede MELCHIORS [64]. Cada uma delas pressupõe um conjunto de informações que devem ser organizadas e utilizadas no processo de tomada de decisões.

Conforme foram descritas anteriormente, as ferramentas de gerenciamento permitem a coleta de dados relevantes para cada área. A extração de informações úteis destes dados, a classificação (seleção) destes dados e a definição da freqüência com que estes dados devem ser coletados constitui tarefas adicionais. Além disso, dependendo do modelo adotado (OSI ou Internet), são aplicados diferentes modelos de informação.

Segundo SPECIALSKI [35], de uma maneira geral e independente do modelo de gerenciamento, pode-se classificar as informações coletadas em 3 tipos:

- **Informação estática:** são informações que caracterizam a configuração dos elementos, como o endereço de cada hosts e a rota;

- **Informações Dinâmicas:** são informações sobre o tráfego de pacotes e a taxa de erros de transmissão, que podem mudar a cada vez que são coletadas;
- **Informações Estatísticas:** são as mesmas informações dinâmicas, mas armazenadas, com formato de porcentagens ou médias, para posterior comparação.

2.1.4 – Arquitetura SNMP para gerência de redes

O SNMP (Simple Network Management Protocol) é um protocolo da camada de aplicação designado para facilitar a troca de informações de gerenciamento entre dispositivos de rede. A primeira proposta do protocolo foi lançada em 1988, com a definição do protocolo SNMP – ou SNMPv1. Este protocolo tornou-se rapidamente o esquema de gerenciamento independente de fabricante STALLINGS [66].

Conforme figura 2.1, o modelo utilizado para gerenciamento de redes SNMP é composto pelos seguintes elementos STALLINGS [66]:

- Estação de Gerenciamento - serve como interface para o gerente humano num sistema de gerenciamento de rede;
- Agente de Gerenciamento - responde às solicitações de informações e de ações da estação de gerenciamento e devem também prover, de maneira assíncrona, informações importantes que não foram solicitadas por esta estação;
- Protocolo de Gerenciamento de Redes - A forma de comunicação entre a estação de gerenciamento e o agente é definido pelo protocolo de gerenciamento de rede, o SNMP.
- Base de Informações Gerenciais (MIB) - os recursos a serem gerenciados são representados como objetos, a MIB é a coleção destes objetos;

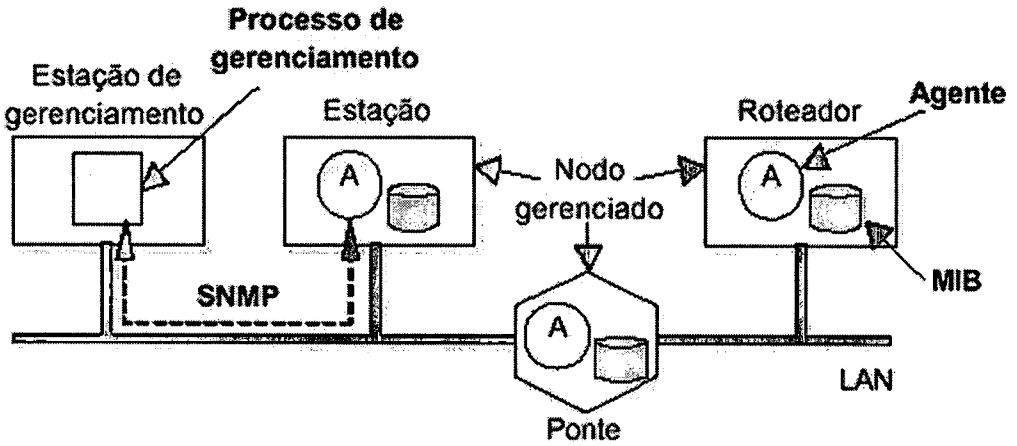

Figura 2.1– Arquitetura de gerenciamento TCP/IP - fonte: GASPARY [67]

O gerenciamento de rede é feito a partir das estações de gerenciamento que, por sua vez, são computadores que executam um software de gerência de rede. As estações de gerenciamento contêm um ou mais processos que se comunicam com os agentes espalhados pela rede, emitindo comandos e obtendo respostas. Muitas dessas estações possuem uma interface gráfica para permitir que o gerente da rede inspecione o estado dos recursos sob gerenciamento e efetue as ações necessárias. Cada agente mantém uma ou mais variáveis que descrevem o estado do recurso onde ele está hospedado. Essas variáveis são chamadas de objetos. TANENBAUM [1].

A interação entre a estação de gerenciamento e os agentes é feita através do protocolo SNMP. Esse protocolo permite que a estação de gerenciamento consulte o estado dos objetos locais de um agente, alterando-os se necessário STALLINGS [66].

O SNMP é um protocolo simples que foi concebido com o objetivo de ser fácil de implementar, utilizando um mínimo de recursos de rede e de processamento. A utilização em massa do protocolo fez com que surgissem alguns problemas no SNMP original. STALLINGS [3] e GASPARY [67], apresentam alguns dos problemas do SNMP v1:

- O SNMP pode não ser adequado no gerenciamento de redes muito grandes, devido às limitações de desempenho impostas pelas consultas. Com a 1ª versão do SNMP (SNMP v 1), faz-se necessário enviar um pacote para receber outro com informação. Esse tipo de consulta resulta em grande volume de mensagens

de gerenciamento e acarreta problemas de tempo de resposta que podem degradar o desempenho da rede;

- O SNMP não é adequado para recuperar grandes volumes de dados, como os existentes em tabelas;
- O protocolo SNMP provê apenas um mecanismo trivial de autenticação. Assim, o SNMP é mais adequado para monitoração do que para controle;
- O SNMP não suporta comandos imperativos. A única forma de disparar um evento em um agente é indiretamente, determinando o valor de um objeto da sua MIB. Essa solução é menos flexível e poderosa do que uma baseada em algum tipo de chamada de procedimento remoto, onde é possível passar parâmetros, definir condições e obter resultados oriundos da execução do procedimento;
- O modelo da MIB é limitado e não suporta aplicações que façam consultas de gerenciamento sofisticadas, baseadas em valores de objetos ou tipos;
- O SNMP não suporta comunicações do tipo gerente-gerente. Por exemplo, não há um mecanismo que permita a um sistema de gerenciamento aprender sobre dispositivos e redes gerenciadas por outro sistema de gerenciamento.

Para solucionar estes problemas, foram propostas novas versões do protocolo, conhecidas como SNMPv2 (versão 2) e SNMPv3 (versão 3).

2.1.5 – Protocolo SNMPV2

Como o SNMPv1 possui algumas deficiências nas áreas de segurança e funcionalidade, foi criada a segunda versão do protocolo, conhecida como o SNMPv2. Esta nova versão encontra-se descrita nos RFCs 1902 - 1907. Os assuntos relacionados com as versões SNMPv1 e SNMPv2 estão apresentados no RFC 1098.

O SNMPv2 apresenta algumas vantagens sobre o SNMPv1, segundo LIMA [26]:

- Comunicação gerente-gerente;
- Gerenciamento hierárquico;
- Eliminação de ambigüidades nas definições dos objetos;
- Acréscimo de duas novas PDUs (Protocol Data Unit), que são mensagens para serem trocadas durante uma comunicação entre o gerente e o agente.

- get-bulk-request-PDU - Permitindo que uma grande quantidade de informações possa ser transferida do agente para o gerente eficientemente ;
- inform-request-PDU – Permitindo que um gerente envie ou eventualmente solicite informações a outro gerente.
- Melhoria na segurança - o SNMPv2 acrescentou novos conceitos e serviços que trouxeram mais segurança ao protocolo. Os conceitos incluídos foram:
- Conceito de visão de MIB definido em termos de sub-árvores, restringindo o acesso a porções predefinidas da MIB;
- Conceito de contexto, que é uma coleção de objetos e seus respectivos agentes, e a especificação dos privilégios envolvidos. Os serviços incluídos são de integridade, autenticação e confiabilidade dos dados;

2.1.6 - Protocolo SNMPv3

Conforme SPECIALSKI [35], uma das características chave do SNMPv3 é a modularidade da documentação e arquitetura. O projeto da arquitetura do SNMPv3 é totalmente compatível com as especificações SNMPv1 e SNMPv2. Esta integração permite a continuação de uso do legado de SNMP por agentes e gerentes SNMPv3.

O RFC 2571, documento que definiu a arquitetura do SNMPv3, define os seguintes objetivos que guiaram o seu desenvolvimento:

- Utilizar o trabalho existente. Os conceitos de segurança do SNMPv3 se baseiam fortemente no SNMPv2u e SNMPv2;
- Resolver o problema de segurança, principalmente para a operação *set-request*, considerada a deficiência mais importante no SNMPv1 e SNMPv2;
- Ser modular para possibilitar o desenvolvimento de parte da arquitetura, mesmo que o consenso não tenha sido atingido no todo;
- Definir uma arquitetura que permita longevidade ao *framework* SNMP que já tenha sido definido e que venha a ser definido no futuro;
- Manter o SNMP o mais simples possível;

- Projetar uma arquitetura modular que permita a implementação sobre diversos ambientes operacionais;
- Acomodar modelos de segurança alternativos.

2.1.7 – Management Information Base (MIB)

Um dos fatores importantes em sistema de gerenciamento é a forma como as informações sobre os elementos de rede estão armazenadas.Tais informações precisam estar disponíveis segundo um determinado padrão, para que possam ser reconhecidas e utilizadas por qualquer aplicação. Este padrão foi definido como *Management Information Base* (MIB).

Para PROENÇA [72], a *Management Information Base* (MIB) é o local onde estão definidas e armazenadas as informações que podem ser acessadas através de um protocolo de gerenciamento. Estas informações ou variáveis poderão ser lidas ou alteradas pela aplicação de gerenciamento, sendo confirmada por SPECIALSKI [35], que diz ser a MIB uma coleção estruturada de objetos gerenciados.

Os objetos gerenciados representam os recursos sujeitos ao gerenciamento. Cada nodo do sistema de gerenciamento mantém uma MIB que reflete o estado dos recursos gerenciados naquele nodo. Uma entidade de gerenciamento pode monitorar os recursos de um nodo, lendo os valores dos objetos na MIB e pode controlar os recursos de um nodo, modificando estes valores.

Uma MIB é apresentada como uma árvore de dados estruturada composta de diversos grupos e subgrupos. Grupos intermediários não possuem valores mas, caso o grupo não possua subgrupos, ele é chamado de objeto e possui um valor associado. Cada grupo possui um OID (Object Indentifier), ou seja, um número que identifica o objeto, que é composto do OID do seu grupo pai mais o seu próprio OID, como mostrado na Figura 2.2., desta forma, conclui-se que o OID para o objeto sysDescr, que faz parte do grupo system, é: 1.3.6.1.2.1.1.1, enquanto seu nome simbólico é a concatenação de todos os nodos que estão acima dele, ou seja:

iso.org.dod.internet.mgmt.mib.system.sysDescr. SPECIALSKI [35] MENDONÇA [68]

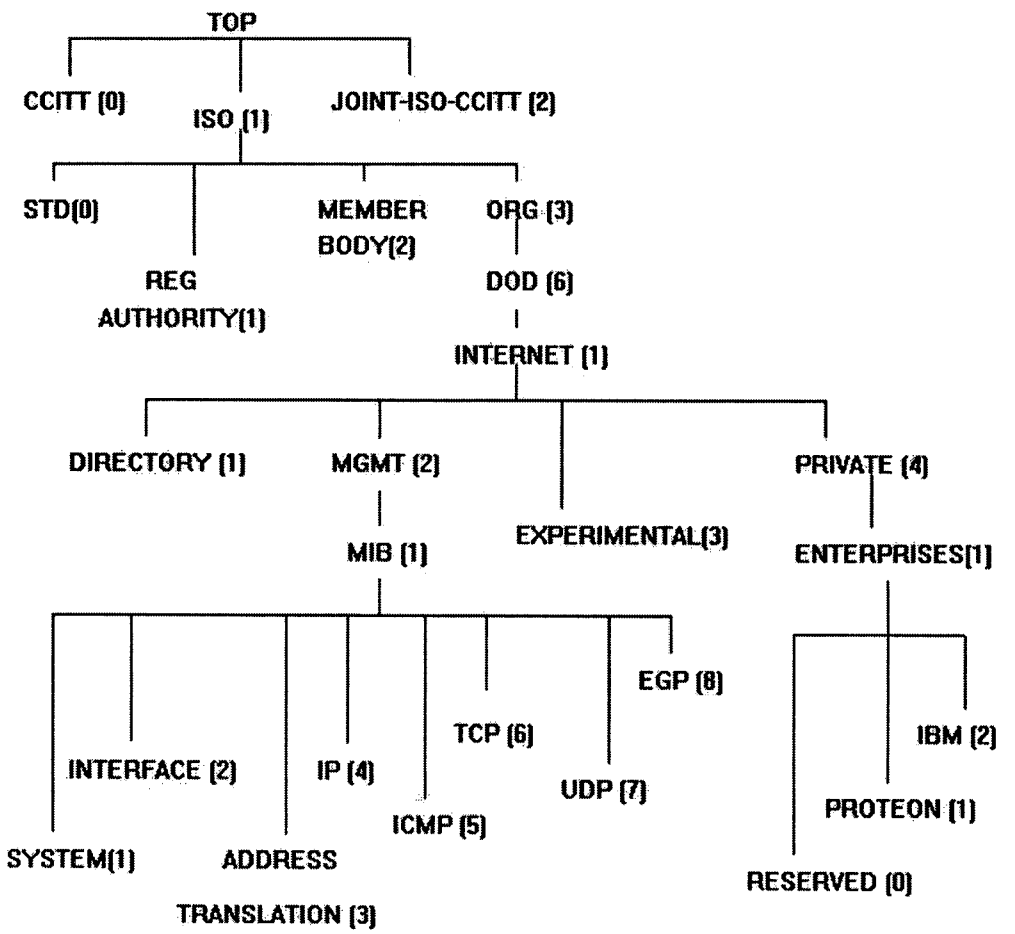

Figura 2.2- Árvore de registro de tipos de objetos – fonte: SPECIALSKI [35]

A sub-árvore MGMT, Figura 2.2, contém a definição das bases de informação de gerenciamento que foram aprovadas pelo IAB. Atualmente, existem duas versões da MIB, conforme SPECIALSKI [35]:

- MIB I - fornece informações gerais sobre o equipamento gerenciado, sem levar em conta as características específicas deste equipamento, possibilitando obter informações como tipo e status da interface (Ethernet, Token-Ring), número de pacotes transmitidos, número de pacotes com erros, informações sobre protocolo de transmissões, etc. Apresentada pela RFC 1066 para uso com o protocolo TCP/IP, explicando e definindo a base de informações necessárias para controlar e monitorar redes baseadas no protocolo TCP/IP. A RFC 1066 foi aceita como padrão pela Interactive Active Board (IAB) na RFC 1156.

- MIB II - é uma evolução da MIB I que introduziu novas informações além daquelas encontradas na MIB I. A MIB II foi proposta pela RFC 1158 e formalizada como padrão de fato pela RFC 1213. Sendo uma evolução da MIB I, a MIB II apresenta o mesmo OID da MIB I, pois apenas uma poderá estar presente em qualquer configuração. SPECIALSKI [35]

Os objetos da MIB II são subdivididos nos grupos como mostra a figura 2.3: SPECIALSKI [35] MENDONÇA [68]

Grupo	Nº de Objetos	Descrição
System	7	Nome, Local e descrição dos Equipamentos.
Interfaces	23	Interface de rede e seu tráfego
AT	3	Conversão de endereços (obsoleto)
IP	42	Estatísticas de Pacotes IP
ICMP	26	Estatísticas sobre as mensagens ICMP recebidas
TCP	19	Algoritmos TCP, parâmetros e estatísticas
UDP	6	Estatísticas de tráfego UDP
EGP	20	Estatísticas de tráfego e protocolo de gateway externo
CMOT	0	Protocolo CMOT
Trasmission	0	Reservado para MIBs de meios físicos específicos
SNMP	29	Estatísticas de tráfego SNMP

Figura 2.3 - Tabela de categorias da MIB II.

Cada grupo apresentado na figura 2.3 possui um conjunto de variáveis, que foram definidas para armazenar os valores dos objetos que se deseja gerenciar, onde, de acordo com SPECIALSKI [35] e MENDONÇA [68], tem-se que:

- O grupo **System** permite que o gerente descubra o nome do dispositivo, quem o fabricou, o que o hardware e o software contém, onde está localizado e o que deverá fazer, entre outras coisas;
- O grupo **Interfaces** lida com os adaptadores de rede. Ele controla o número de pacotes e bytes enviados e recebidos na rede, o número de descartes, o número de difusões e o tamanho da fila de saída;
- O grupo **AT** fornece informações sobre o mapeamento de endereços (por exemplo, Ethernet em IP);

- O grupo **IP** trata do tráfego IP recebido e emitido pelo nó. Ele é possui contadores que controlam o número de pacotes descartados por diversas razões. Estão disponíveis estatísticas sobre a fragmentação e a remontagem de dados, sendo especialmente importantes para o gerenciamento de roteadores;
- O grupo **ICMP** se refere a mensagens de erro IP. Basicamente dispondo de um contador que registra quantas mensagens de erro ICMP de um determinado tipo foram encontradas;
- O grupo **TCP** monitora o número atual e o cumulativo de conexões abertas, segmentos enviados e recebidos e diversas estatísticas de erros;
- O grupo **UDP** registra o número de datagramas UDP enviados e recebidos, e quantos dos enviados não foram entregues devido a uma porta desconhecida ou a algum outro erro;
- O grupo **EGP** é usado para roteadores compatíveis com o protocolo de gateway externo. Ele controla quantos pacotes de um determinado tipo foram enviados, quantos foram recebidos e encaminhados corretamente, e quantos foram recebidos e descartados;
- O grupo **CMOT** existe somente por razões históricas. O CMOT é um protocolo que ajuda na transição do SNMP para o CMIS/CMIP
- O grupo **Transmission** é um marcador de lugar para as MIBs de meios físicos específicos. Por exemplo, nesse grupo é possível manter estatísticas especificamente relacionadas a Ethernet;
- O grupo **SNMP** se destina ao cálculo de estatísticas sobre a operação do próprio SNMP.

Na MIB I e na MIB-II, as categorias possuem 175 objetos, como descrito em ROSE [13].

Para estruturar as MIBs, é necessário a definição de regras de construção da base de dados, estas regras são definidas em um documento chamado SMI, conforme apresentado no próximo item.

2.1.8 – SMI - Structure of Management Information e ASN.1

As regras de construção das estruturas da MIB são descritas através da SMI – Structure of Management Information. A estrutura de informações de gerência SMI é um conjunto de documentos que definem:

- Forma de identificação e agrupamento das informações;
- Sintaxes permitidas;
- Tipos de dados permitidos.

Os objetos de uma MIB são especificados de acordo com a ASN.1 - Abstract Syntax Notation One. A notação sintática abstrata é uma forma de descrição abstrata dos dados com o objetivo de não levar em consideração à estrutura e restrições do equipamento no qual está sendo implementada. Para cada objeto são definidos: nome (Object Name), identificador (Object Identifier), sintaxe (Syntax), definição e acesso. As instâncias do objeto são chamadas de variáveis:

- Object Name é o nome do objeto, é composto por uma string de texto curto.
- Object Identifier é o identificador do objeto, é formado por números que são separados por pontos.
- Syntax, sintaxe do objeto, descreve o formato, ou o valor, da informação. Ela pode ser:
    - Sintaxe do tipo simples que pode ser um inteiro, uma string de octetos, um Object Identifier ou nulo;
    - Sintaxe de aplicação, podendo ser um endereço de rede, um contador, uma medida, um intervalo de tempo ou incompreensível.
- A definição é uma descrição textual do objeto.
- O acesso é o tipo de controle que se pode ter sobre o objeto, podendo ser: somente leitura, leitura e escrita ou não acessível DIAS [71] PROENÇA [72].

2.1.9 - RMON (Remote Monitoring)

Para se obter uma monitoração eficiente, o intervalo de tempo entre duas leituras de valores de objetos em uma rede deve ser pequeno. No entanto, quanto menor o tempo de *polling*, mais tráfego é gerado na rede. Conseqüentemente, mais problemas poderão surgir, tais como, baixa de desempenho da rede. Para solucionar o problema de

tráfego gerado entre agente/gerente, mantendo a freqüência de monitoração das variáveis, foi desenvolvido o RMON, que armazena os dados capturados no próprio agente, possibilitando aumentar o intervalo de *polling* sem perder informações.

No agente RMON, o processamento para cálculos e contabilizações estatísticas é executado localmente no agente, sendo que os dados são enviados para a estação de gerenciamento, já refinados. "Os benefícios são claros, pois a estação de gerenciamento irá fazer menos *polling* ao agente gerando menos tráfego na rede" PROENÇA [72].

O RMON resolve o problema dos agentes SNMP tradicionais que não são capazes de analisar seus próprios dados, forçando a estação de gerenciamento a ficar inspecionando (fazendo *polling*) regulares em todos os dispositivos gerenciados, causando um tráfego excessivo na rede SPECIALSKI [35].

O RMON segundo, CARVALHO [14] e WALDBUSSER [15], oferece suporte à implementação de um sistema de gerenciamento distribuído. Nele, atribui-se aos diferentes elementos, tais como estações de trabalho, hubs, switches ou roteadores, das redes locais remotas, a função de monitor remoto.

Cada elemento RMON tem, então, como tarefas, coletar, analisar, tratar e filtrar informações de gerenciamento da rede e apenas notificar à estação gerente os eventos significativos e situações de erro. No caso de existirem múltiplos gerentes, cada elemento RMON deve determinar quais informações de gerenciamento devem ser encaminhadas para cada gerente.

Sendo assim, segundo CARVALHO [14], os objetivos do RMON são:

- Reduzir a quantidade de informações trocadas entre a rede local gerenciada e a estação gerente conectada a uma rede local remota;
- Possibilitar o gerenciamento contínuo de segmentos de redes locais, mesmo quando a comunicação entre o elemento RMON e a estação gerente estiver, temporariamente, interrompida;

- Permitir o gerenciamento pró-ativo da rede, diagnosticando e registrando eventos que possibilitem detectar o mau funcionamento e prever falhas que interrompam sua operação;
- Detectar, registrar e informar à estação gerente, condições de erro e eventos significativos da rede;
- Enviar informações de gerenciamento para múltiplas estações gerentes, permitindo, no caso de situações críticas de operação da rede gerenciada, que a causa da falha ou mau funcionamento da rede possa ser diagnosticada a partir de mais de uma estação gerente.

É possível haver agentes RMON em vários segmentos da rede se reportando a uma ou mais estações de gerenciamento, conforme mostra a **figura 2.4** PROENÇA [72].

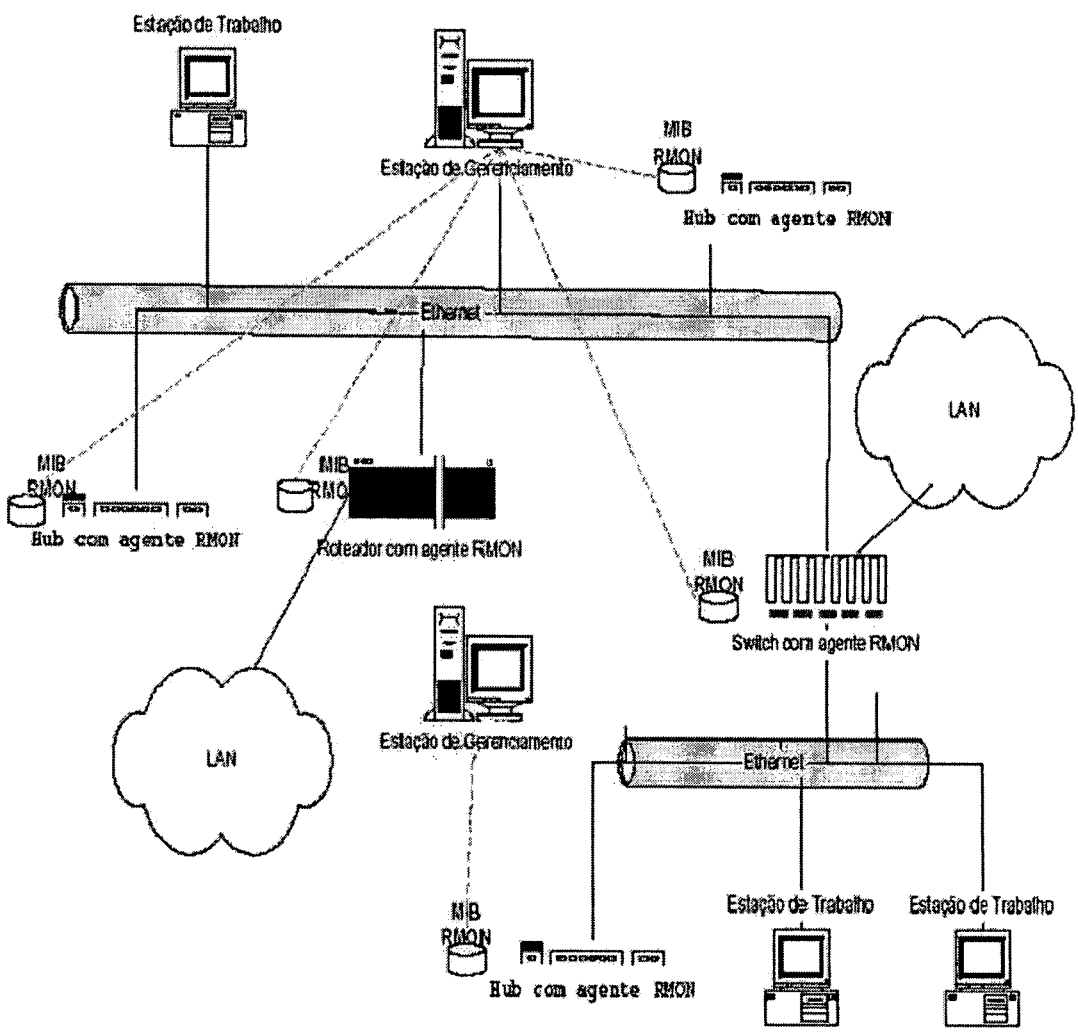

Figura 2.4 – Modelo de gerenciamento com o modelo RMON – PROENÇA [72]

2.1.10 - BASELINE

Para facilitar a percepção de alguma anormalidade na rede, ou seja, perceber quando um segmento monitorado está tendendo a um estado crítico, é necessário comparar a situação atual do segmento com os dados estatísticos sobre o comportamento considerado normal para este segmento VERONEZ [69].

Segundo NETO [16], baseline, é uma caracterização estatística do funcionamento normal do segmento monitorado de uma rede. Geralmente é criado a partir de monitoração da rede no período normal ou a partir de simulações.

Segundo ROCHA [70], Uma proposta para gerar uma Baseline monitorada é o monitoramento da máquina gateway durante um período determinado e, a partir dos dados colhidos, estimar o comportamento para cada hora de cada dia da semana, gerando vários arquivos, um para cada parâmetro utilizado em algum sistema especialista de verificação.

2.2 - KDD – Knowledge Data Discovery

O KDD é um processo que envolve a automação da identificação e do reconhecimento de padrões em um banco de dados COMPUTER [23]. Trata-se de uma pesquisa de fronteira que começou a se expandir mais rapidamente na segunda metade da década de 90, sua principal característica é a extração não-trivial de informações a partir de uma base de dados de grande porte SADE [24]. Essas informações são necessariamente implícitas, previamente desconhecidas, e potencialmente úteis.

O KDD mostra-se como uma ferramenta semi-automática que possibilita a análise de grandes conjuntos de dados. Propõe-se como o descobridor de informação útil a partir de uma grande base de dados. A informação descoberta pode ser representada por

regras, descrevendo propriedades dos dados, padrões que ocorrem freqüentemente, agrupamentos de objetos na base de dados, etc... MANNILA [72].

O processo de KDD não requer que os usuários criem hipóteses sobre os relacionamentos e correlações entre variáveis. É possível ao usuário, por exemplo, simplesmente perguntar quais são as variáveis que afetam as vendas de um produto específico. Em todo o processo de KDD, o Datamining representa 20 % os outros 80 % são atribuídos à preparação dos dados MEGAPUNTER [19].

Segundo FRAWLEY [20], KDD é o processo não trivial de identificar padrões nos dados válidos, novos, potencialmente úteis e compreensíveis. Já para FAYAD [17], KDD é o processo de extrair (identificar) conhecimento, usando métodos (algoritmos) de mineração de dados, de acordo com o que é julgado conhecimento, pelas métricas e saídas esperadas, usando uma base de dados onde são requeridos alguns: pré-processamentos, exemplificações da base e transformações.

Inicialmente, as pesquisas foram centradas na construção, implementação e otimização de técnicas com capacidade de extração de padrões a partir de uma grande massa de dados, denominados algoritmos de mineração HAN [37] AGRAWAL [38] JAI [39]. Porém, quando estes algoritmos eram aplicados em uma tarefa de descoberta de conhecimento em um ambiente real, produziam resultados que não eram suficientemente confiáveis. Isto ocorre, principalmente, devido às possíveis inconsistências existentes nos dados, como valores nulos ou incoerentes e a necessidade de adequação destes dados à técnica de mineração implementada pelo algoritmo GHEDINI [28].

Neste capítulo são apresentadas duas abordagens para o processo de KDD, apresentando suas principais características e as dificuldades em sua aplicação.

A Seção 2.2.1 apresenta a abordagem de FAYAD, FAYAD [40] FAYAD [41] FAYAD [42] que está mais voltada para a definição e características das etapas do processo de KDD. A Seção 2.2.2 descreve a abordagem de Brachmann. BRACHMAN

[43], que além da estrutura do processo, enfoca a tarefa do analista na construção de aplicações. A seção 2.2.3 mostra uma tabela comparativa entre dois outros modelos menos utilizados, o Modelo de SIMOUDIS e o modelo de MANNILA. Os problemas com os Modelos segundo FAYAD e segundo BRACHMAN no contexto do projeto, são detalhados na seção 2.2.4.

2.2.1 - Abordagem de FAYAD

O autor, em FAYAD [40], define KDD como o processo não trivial de identificação de padrões válidos, novos, potencialmente utilizáveis e finalmente compreensíveis a partir dos dados. Esta definição está relacionada ao pressuposto de que este processo deve envolver pesquisa de estruturas, padrões, modelos ou parâmetros com algum grau de autonomia, caracterizando-o como não trivial. A necessidade dos padrões descobertos serem válidos para novos dados com algum grau de certeza, novos e potencialmente utilizáveis para usuários e tarefas; e, compreensíveis, se não imediatamente, após algum processamento.

A utilização do termo "processo" implica que há vários passos iterativos e interativos envolvendo a preparação dos dados, a busca por padrões, a avaliação do conhecimento obtido e seu refinamento. Ele é *Interativo* porque a maioria dos passos envolve avaliação e tomada de decisão pelos usuários e, *iterativo* porque de qualquer etapa pode-se decidir por retomar etapas anteriores do processo, até que o resultado esperado seja alcançado.

Para o processo de KDD segundo FAYAD [40] as tarefas são descritas como na figura 2.5

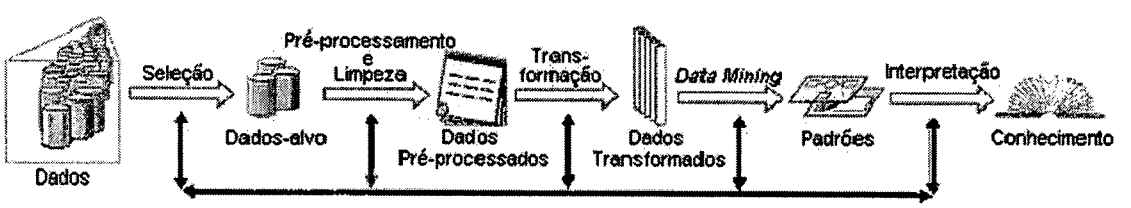

Figura 2.5 – Modelo KDD Segundo Fayad - MESQUITA[45]

2.2.1.1 - Definição de Objetivos

O primeiro passo para a definição dos objetivos é definir o objetivo em termos de negócio, ou seja, qual o retorno esperado, o que se deseja atingir, por exemplo: definir o que utilizar em uma promoção com mala direta, qual o segmento do mercado consumidor a atingir.

O segundo passo são as definições dos objetivos próprios do KDD, como por exemplo: identificar clientes de alto risco e identificar afinidades entre os clientes.

O Terceiro passo é a definição da ação a ser tomada com os resultados obtidos, como por exemplo: alterar o plano de mercado, criar campanha promocional.

2.2.1.2 - Seleção de Dados

Deve-se definir o volume adequado de dados, se existirem muitos dados o processo será mais demorado, por outro lado, poucos dados implicariam na falha de conclusão. A base de dados deve ser trabalhada de forma a eliminar as colunas inúteis, como por exemplo, o nome dos clientes, quando será definido o perfil do consumidor. As linhas com valores errados também deverão ser excluídos, pois isto poderia alterar a conclusão do processo de KDD.

2.2.1.3 - Pré Processamento (Entendimento, Limpeza e Enriquecimento dos dados)

Esta fase pode ser decomposta em três sub-fases:

- *Entendimento dos dados* - pode-se utilizar diferentes ferramentas para compreender uma base de dados: ferramentas de Query determinam valores estatísticos de colunas relevantes; ferramentas Estatísticas que podem, por exemplo: traçar a média e sua variação nos dados de uma coluna. Ferramentas OLAP permitem determinar relações entre colunas e linhas; ferramentas de Visualização que mostram em gráficos e imagens bidimensionais ou tridimensionais as informações da base de dados;
- *Limpeza dos Dados* - deve-se verificar, por exemplo, a duplicação de casos ou registros na base de dados, a plausibilidade das informações contidas na base de dados;

- *Enriquecimento dos Dados* - descobertas as falhas, pela limpeza dos dados, deve-se enriquecer a base de dados, trocando os dados falsos por verdadeiros. Por exemplo, quando da existência de três códigos diferentes para o mesmo cliente devido a diferenças na digitação dos nomes: Adinarte C. da Silva, Adinarte Correa da Silva e Adinarte Silva. Deve-se verificar qual o nome correto e alterar os registros.

2.2.1.4 - Transformação dos dados

Dependendo da técnica de Datamining utilizada, os dados devem ser preparados de formas diferentes. Alguns passos podem ser necessários como, por exemplo, realizar cálculos, agrupar valores, transformar o conteúdo de determinado campo, nivelamento de valores, etc.

2.2.1.5 – Dataminning

Deve ser criado o modelo levando em conta três etapas:

- *Definição da tarefa de mineração de dados*: definir o objetivo do modelo a ser gerado por uma técnica de mineração de dados, considerando o objetivo geral do processo de KDD, ou seja, decidir a classe do algoritmo AGRAWAL [38] que será aplicado, como classificação, segmentação, associação entre outras;
- *Definição do algoritmo de mineração de dados:* cada tarefa de mineração pode consistir da aplicação de vários algoritmos. Nesta etapa o(s) método(s) a ser (em) utilizado(s) para pesquisa dos padrões, bem como os modelos e parâmetros que podem ser apropriados, são selecionados;
- *Mineração*: aplicar o algoritmo sobre os dados selecionados, ou seja, buscar os padrões de interesse. Esta etapa envolve o constante ajuste dos parâmetros para refinamento do modelo.

2.2.1.6 – Interpretação / Avaliação

Nesta etapa são executadas a visualização, interpretação e avaliação dos padrões extraídos; remoção de valores redundantes ou padrões irrelevantes e a tradução destes para uma linguagem que possa ser entendida pelo usuário. Como destacado por CABENA [44], os resultados da etapa de mineração não têm efeito algum até que sejam validados e, somente depois de verificado seus graus de interesse é que podem ser classificados como "conhecimento".

2.2.1.7 - Consolidação do conhecimento descoberto

A função desta etapa é incorporar este conhecimento dentro da execução do sistema seja como documentação e relatórios, ou execução de ações baseadas nesse conhecimento.

Segundo GHEDINI [28], a abordagem de Fayad, apesar de conseguir adequar as principais atividades dentro de etapas no contexto de um processo, se detém ao fluxo de procedimentos a serem realizados entre estas etapas. A principal crítica a esta abordagem é que ela define o processo como interativo e iterativo, mas não explora questões importantes que determinam estas características, tais como:

- A importância do conhecimento prévio do domínio dos dados e da aplicação como um fator determinante na condução do processo;
- As atividades de definição das tarefas a serem realizadas e do objetivo global do processo, que estão sempre sendo refinados à medida que os dados vão sendo explorados com maior profundidade e são responsáveis pelos redirecionamentos no processo;
- A importância do papel do usuário dentro do processo;
- A utilização de um conjunto heterogêneo de ferramentas, geralmente não integradas.

2.2.2- Abordagem de Brachman

O processo de KDD proposto em BRACHMAN [43], parte da descoberta da tarefa e dos dados centrados no objetivo geral do processo e das informações da base de dados. Após, passa por diversas etapas que são apoiadas por um conjunto de ferramentas de consulta, estatística e I.A., visualização, apresentação e transformação, que interagem com o usuário. Brachman considera que o empreendimento é extremamente complicado e, apesar de existir uma ordem em suas etapas, o processo é iterativo e o analista pode se mover de uma etapa para outra a qualquer momento, sem seguir uma ordem determinada. As etapas definidas estão apresentadas na figura 2.6:

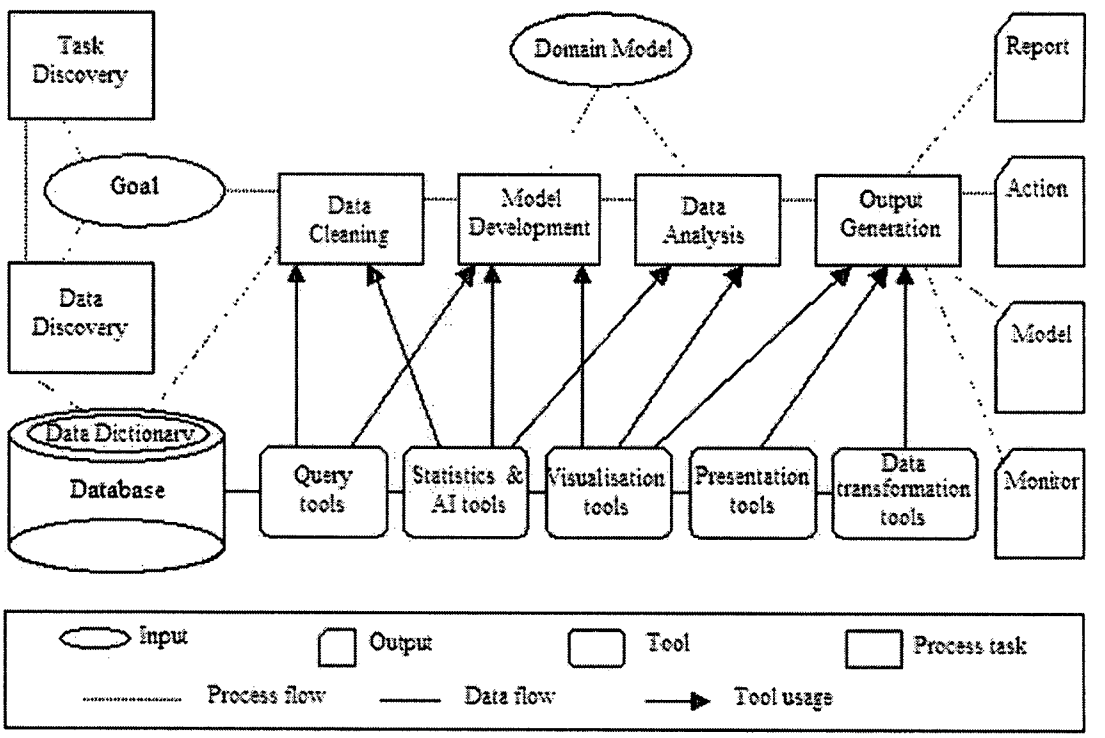

Figura 2.6 – Modelo de Brachman - ANDRASSOYVÁ [34]

Sendo:

2.2.2.1 - Task Discovery

A realização desta tarefa, em conjunto com o usuário, permite a definição e clareza dos objetivos globais do processo.

2.2.2.2 - Data Discovery

É necessário conhecer e entender a estrutura, o conteúdo e a qualidade dos dados. Outra questão importante é avaliar se os dados podem satisfazer o objetivo definido. É importante ressaltar que esta etapa e a de descoberta da tarefa, citada acima, são executadas quase que simultaneamente. Isto porque a definição dos objetivos da aplicação e as tarefas que devem ser realizadas para alcançar este objetivo estão muito ligadas aos dados existentes e ao seu domínio. Conseqüentemente, o aprimoramento da visão sobre os dados pode refinar os objetivos e tarefas definidas anteriormente. Estas duas etapas são intercaladas a todo o momento: em um primeiro estágio, quando da definição da aplicação; e no decorrer do processo.

Isto acontece porque o aprimoramento é conseguido nas etapas posteriores de limpeza e análise dos dados. Desta forma, pode ser necessária à redefinição dos objetivos e, naturalmente, das tarefas a serem realizadas para alcançá-los.

2.2.2.3 - Data Cleaning

Dados podem conter ruídos, como valores inexistentes e/ou inconsistentes e, um trabalho de limpeza nestes dados é necessário. Esta tarefa é geralmente inevitável, mas deve-se fazer uma análise cuidadosa, porque muitas vezes o que pode parecer uma anomalia pode indicar um fenômeno crucial do domínio.

2.2.2.4 - Model Development

O primeiro trabalho do analista nesta etapa é encontrar um subconjunto na população da base de dados que se comporte de uma forma tal, que seja significativa o bastante para ser o foco de análise. Os dados devem ser visualizados e ajustados, se necessário. Nesta etapa são realizadas as tarefas de:

- Segmentação dos dados geralmente utilizando-se de técnicas não supervisionadas como agrupamento;
- Seleção do modelo a ser gerado, tais como regressão, árvores de decisão, redes neurais;

- Seleção dos parâmetros. O principal objetivo desta etapa é explorar os possíveis modelos de mineração disponíveis e ajustar os parâmetros de acordo com os dados e o objetivo do processo de KDD.

2.2.2.5 - Data analysis

Neste momento, o analista tem uma hipótese sobre os dados e vários tipos de ferramentas para desenvolver o modelo. Nesta etapa é realizada uma especificação formal do modelo, sua avaliação e um possível refinamento. Quando necessário, os parâmetros específicos para o modelo são ajustados. Geralmente é necessário utilizar-se iterativamente de ferramentas de análise baseadas em visualização e em algoritmos.

2.2.2.6 - Output generation:

As saídas podem ser geradas de diversas formas, como relatórios, descrição textual, gráficos, etc. Podem-se também desenvolver gatilhos no banco de dados que disparam ações ou alertas quando determinadas condições forem detectadas.

Segundo GHEDINI [28], a abordagem segundo Brachman, tem o mérito de detalhar o processo de KDD de uma forma bem mais abrangente e realista. Preocupa-se com fatores que têm grande influência nos resultados de uma aplicação, principalmente com as deficiências apontadas na abordagem de Fayyad, entre elas:

- A inclusão das etapas de descoberta dos dados e da tarefa, as quais são fundamentais para a definição do escopo da aplicação. Estas etapas e suas atividades são altamente iterativas e interativas, e o seu resultado é a especificação dos objetivos da aplicação e a definição das atividades que deverão ser executadas para alcançar este objetivo. Durante qualquer outra etapa do processo, o aprimoramento do conhecimento sobre o domínio dos dados pode gerar uma redefinição das atividades a serem executadas, reestruturando o processo, bem como o ajuste dos objetivos;
- A natureza da tarefa do analista, que na prática é extremamente interativa, envolve a manipulação de vários arquivos, tabelas, consultas e diversas

ferramentas de apoio, as quais nem sempre estão projetadas para trabalhar em conjunto. Existem ainda inúmeros sub-problemas que devem ser tratados antes do problema principal ser atacado. Dentre estes estão a necessidade de analisar minuciosamente os dados, realizar conversões de dados, trabalhar constantemente com ruídos, dados confusos e complexos consultas SQL. Além de todas estas dificuldades, a condução de uma aplicação é muito dependente de seu conhecimento prévio sobre o domínio dos dados;

- A falta de apoio à tarefa do analista, por parte das ferramentas ou técnicas de descoberta de conhecimento, ficando clara a necessidade de enfatizar mais o processo centrado no usuário (analista) e nas suas tarefas, pois o processo de descoberta de conhecimento é muito mais complexo que simplesmente a descoberta de padrões interessantes.

2.2.3- Modelos de Simoudis e Mannila.

ANDRASSOYVÁ [34], além de comparar os modelos de Fayyad e Brachman, compara, em nível de processo, os modelos de Simoudis e Mannila, conforme apresentado na figura 2.7. Observando-se esta comparação e as características dos modelos de KDD, apresentadas anteriormente, verifica-se que o processo considerado mais importante em todos os modelos é o Datamining. Isto ocorre por ser o processo que realmente busca a solução do problema apresentado. Desta forma, a seção 2.3 descreve esta técnica com mais detalhes.

<table>
<tr><th>Simoudis</th><th>Mannila</th><th>Fayyad et al.</th><th>Brachman & Anand</th></tr>
<tr><td></td><td>Entendendo o dominio</td><td>Definição dos objetivos</td><td>Descoberta das tarefas</td></tr>
<tr><td>Seleção dos dados</td><td></td><td>Seleção dos dados</td><td>Descoberta dos dados</td></tr>
<tr><td rowspan="3">Transformação dos Dados</td><td rowspan="3">Preparação dos dados</td><td>Limpeza dos dados e pré-processamento</td><td>Limpeza dos dados</td></tr>
<tr><td>Redução dos dados e projeção</td><td rowspan="2">Desenvolvimento do modelo</td></tr>
<tr><td>Escolhendo o modelo de datamining</td></tr>
<tr><td rowspan="2">Data mining</td><td rowspan="2">Descobrindo os padrões (datamining)</td><td>Escolhendo o algoritmo de dataminig</td><td rowspan="2">Analise dos dados</td></tr>
<tr><td>Datamining</td></tr>
<tr><td>Interpretação dos resultados</td><td>Pós-processamento dos padrões descobertos</td><td>Interpretação</td><td>Geração das Saidas</td></tr>
<tr><td></td><td>Colocando os resultados em uso</td><td>Utilizando os dados descobertos</td><td></td></tr>
</table>

Figura 2.7 – Modelo Simoudis e Manilla - ANDRASSOYVÁ [34]

2.2.4- Problemas com os modelos segundo FAYAD e segundo BRACHMAN no contexto do projeto

Para elaborar o modelo KDD teoricamente ideal, no processo que o trabalho compreende, e que trata especificamente sobre informações pertinentes a redes de computadores, foram analisados as principais virtudes e defeitos dos modelos de BRACHMAN e de FAYAD, que são os mais utilizados conforme mostra a figura 2.8.

O Que	Observação	Efeito	FAYAD	BRACHMAN
Informações não correlatas na base de dados	O Objetivo da tarefa e o conhecimento dos dados necessitam estar suficientemente claros	O Algoritmo poderá processar atributos da base de dados que não são necessários, gerando lentidão e um número excessivo de regras desnecessárias.	Não faz referência à descoberta dos dados e nem à descoberta da tarefa	Possui duas funções específicas para estas tarefas: o Task Discovery e o Data Discovery
Ferramentas integradas no processo	Para que a tarefa seja realizada, é necessário à utilização de um conjunto de ferramentas que não são integradas no processo.	Não padronização na utilização das ferramentas auxiliares	Não fazem nenhuma referência as ferramentas auxiliares	Especifica em paralelo ao fluxo de dados as ferramentas que serão utilizadas em cada tarefa, como ferramentas de consulta, Ferramentas Estatísticas, etc...
Importância da interação do analista no processo	Evidencia a interação do analista no processo	Dificuldade em entender o modelo do ponto de vista do analista	Não evidencia a importância da interação do analista no processo	Demonstra a interação entre as ferramentas auxiliares e a tarefa do modelo
Definição da tarefa	Divisão do modelo em tarefas	Tarefas importantes estão agrupadas com outras tarefas	Com exceção do Pré-processamento e Limpeza, as outras tarefas estão bem estratificadas. Ex: Seleção de dados, transformação, dataminig, etc...	Possui muitas tarefas conceitualmente diferentes em uma mesma tarefa. Ex: não a seleção criteriosa dos dados, na base de dados como uma tarefa separada.

Figura 2.8 – diferenças entre o Modelo de FAYAD e o Modelo de BRACHMAN

Conforme os detalhes observados na figura 2.8, os dois modelos possuem qualidades e defeitos, o ideal seria a junção dos modelos em um único. Isto porque, a base de dados gerados pela rede possui diversos atributos e outros de grande importância, resultantes de uma equação derivada.

O modelo de BRACHMAN prima pelo entendimento do modelo do domínio, enquanto o de FAYAD, prima pela estratificação das tarefas. O modelo proposto, decomposto no capítulo 3, busca a junção entre BRACHMAN e FAYAD, visando facilitar o processo de descobrimento de dados provenientes do protocolo SNMP.

A tarefa considerada mais importante do KDD é o datamining, que é incumbida de, efetivamente, encontrar as relações na base de dados. Devido a sua importância, será tratada de forma mais aprofundada no capítulo 2.3.

2.3 - DATAMINING

2.3.1- Considerações Iniciais

A capacidade de coletar e gerar dados têm crescido rapidamente conforme pode ser observado na figura 2.9 FÉLIX [47]. A difusão do uso de código de barras nos produtos comerciais, a informatização de muitos negócios e de transações governamentais e os avanços das ferramentas de coleções de dados têm fornecido uma enorme quantidade de dados.

Milhões de base de dados têm sido usadas em: gerenciamento de negócios, administração governamental, gerenciamento de dados científicos e relativos à engenharia e em muitas outras aplicações. Estas bases de dados continuam com uma curva de rápido crescimento devido à disponibilidade de poderosos sistemas geradores MESQUITA [45], como conseqüência, estas bases de dados possuem verdadeiros tesouros de informação escondidas. Toda esta informação pode ser usada para melhorar procedimentos, permitindo que a empresa detecte tendências e características disfarçadas, reagindo rapidamente a um evento futuro.

Este crescimento de volume nas bases de dados tem gerado uma necessidade de novas técnicas e ferramentas. Estas podem, de maneira inteligente e automática, transformar o dado processado em informação útil e de fácil compreensão. MESQUITA [45].

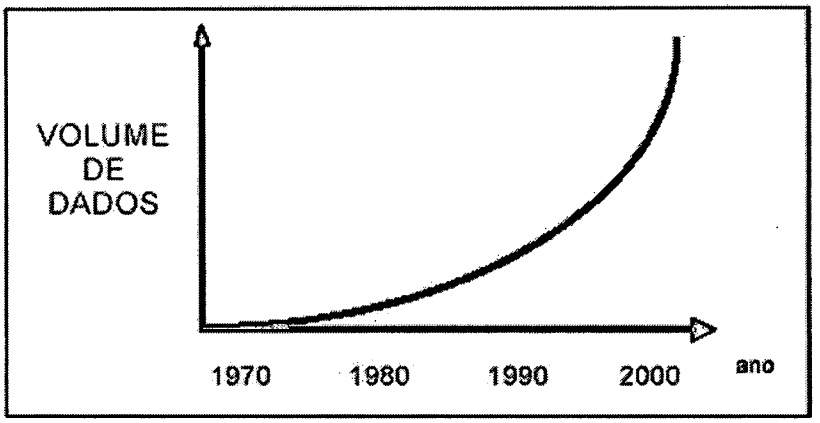

Fig 2.9 – Crescimento do Volume de Dados – FÉLIX[47]

Ultimamente, umas das técnicas utilizadas para extrair informações úteis de uma grande base de dados é o datamining. Segundo PIATETSKY [46], o datamining também é referenciada como Knowledge Data Discovery. Já FAYAD [17], afirma que o Datamining é simplesmente uma das etapas do KDD. FAYAD [17] diz ainda, que "o Datamining consiste de algoritmos particulares de mineração de dados que, sob algumas limitações de eficiência computacionais aceitáveis, produz uma enumeração particular de padrões sobre uma base de dados".

Um Exemplo prático e comum da utilização de técnicas de Datamining é a análise de dados coletados de uma Vídeo Locadora, para extrair informações sobre os clientes que permitirão rentabilizar a empresa Figura 2.10.

Seria interessante saber quais filmes os clientes vão querer ver e ter um modo de avisá-los, que chegou um filme o qual eles provavelmente vão querer locar. Para saber quais são os filmes favoritos dos clientes, poder-se-ia armazenar informação sobre todos os filmes que cada cliente alugou. Através de datamining ou de outras técnicas como OLAP (On-Line Analytical Processing) ou mesmo simples queries SQL, seria possível poder fazer uma previsão sobre quais os melhores filmes a adquirir, quais as quantidades e que época seria conveniente adquirir uma quantidade maior de filmes.

Para avisar aos clientes, uma simples carta, com as últimas novidades seria suficiente. No entanto, uma carta para todos os clientes de todas as lojas, pode ficar um tanto oneroso, tendo em vista que muitos dos clientes não estariam interessados em

muitos dos filmes anunciados. Também é possível usar Datamining para tentar descobrir para quem vale a pena enviar a carta.

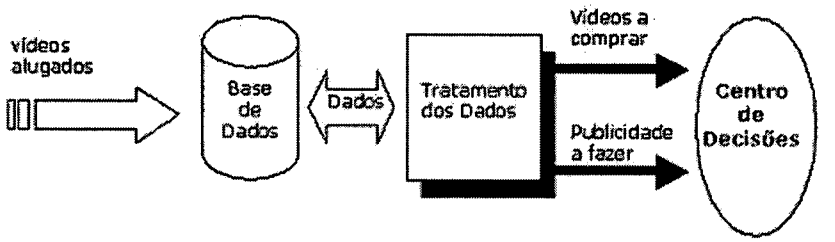

Fig 2.10 – Aquisição de Informação utilizando Datamining – SANTOS [48]

Este exemplo de Vídeo Locadora é transportável para outros negócios, pois grandes empresas, mesmo algumas empresas menores, teriam enormes vantagens na implementação de técnicas de datamining, para tratar informações de um banco de dados. Assim, a partir de uma base de dados existente na empresa, pode-se utilizar Datamining para refinar os dados e obter informações que possam ser utilizadas nos centros de decisão.

Entendendo que os dados pertinentes à rede de computadores geralmente constituem uma grande base de dados e que muitas informações relevantes estão na verdade escondidas sob esta montanha de dados e podem ser descobertas utilizando técnicas de datamining

2.3.2- Técnicas Convencionais x Técnicas de Datamining

Diferentes técnicas existem para analisar os dados dos clientes. Há técnicas convencionais, como OLAP, ferramentas de consulta (query) e estatística, e novas técnicas como datamining. O valor de datamining pode ser mais bem compreendido se comparado a técnicas convencionais. Abaixo são tratadas separadamente as técnicas convencionais e datamining. (posteriormente, é apresentada uma comparação entre as técnicas).

2.3.2.1- Técnicas Convencionais

Segundo DATADISTILLIERS [50], técnicas estatísticas são as mais utilizadas na construção de modelos preditivos. Técnicas estatísticas têm a desvantagem de necessitar de premissas sobre os modelos. A maior parte das bases de dados contém muitas informações diferentes, geralmente armazenada em algumas centenas de variáveis. Na construção de um modelo é necessário selecionar as variáveis a serem incorporadas, além de indicar como devem ser as relações entre estas variáveis. Por causa desta restrição, relações potencialmente interessantes são facilmente esquecidas.

Por exemplo, um analista pode querer determinar os fatores que levam clientes a solicitar indenização das seguradoras de automóvel nos casos de acidentes. Ele pode, inicialmente, formular hipóteses de que motoristas homens são clientes de risco e, então, analisar a base de dados com ferramentas estatísticas para comprovar ou refutar a hipótese. Se a hipótese não for comprovada pelos dados, ele deverá procurar outro fator como, por exemplo: idade, como elemento indicador de risco. Se os dados ainda não oferecerem suporte à hipótese, ele poderá utilizar os fatores idade e sexo, juntos como fatores de predição.

No entanto, quando existem dezenas ou mesmo centenas de variáveis, torna-se mais difícil e trabalhoso formular uma boa hipótese e analisar as bases de dados com técnicas estatísticas para comprová-la ou refutá-la.

2.3.2.2- Técnicas Datamining

Segundo DATADISTILLIERS [50], o datamining difere de técnicas estatísticas porque, ao invés de verificar padrões hipotéticos, utilizam os próprios dados para descobrir padrões. Por exemplo, supondo que o analista que queria identificar os fatores de risco de solicitação de indenização utilizasse uma ferramenta de datamining. A ferramenta poderia descobrir que motoristas com menos de 24 anos de idade são clientes de risco, mas a ferramenta pode ir além e descobrir um padrão que o analista não pensou inicialmente. Por exemplo, que o tipo de carro em combinação com a idade também são determinantes do risco.

2.3.2.3 - Comparação entre Técnicas convencionais e Datamining

Bases de dados armazenam conhecimento que podem auxiliar a melhorar negócios. Técnicas tradicionais permitem a verificação de hipóteses. Aproximadamente 5% de todas as relações podem ser encontradas por este método. Datamining pode descobrir outras relações anteriormente desconhecidas: os 95% restantes, conforme figura 2.11.

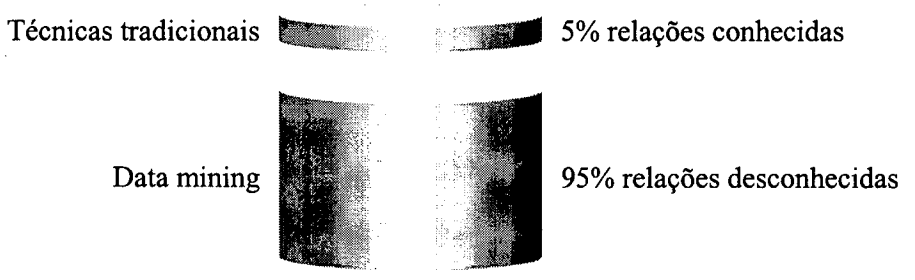

Fig: 2.11 – Técnicas Tradicionais X Técnicas Datamining – DATADISTILLIERS [50]

Enquanto as técnicas convencionais necessitam que analista informe anteriormente o que será pesquisado o datamining busca correlação entre os dados existentes na base de dados sem interferência externa.

2.3.3 – Metodologia do Datamining

Segundo CARVALHO [27], a mineração de dados pode ser realizada em três diferentes formas, em função do nível de conhecimento que se tenha do problema estudado:

- **Descoberta não supervisionada de relações** - Esta técnica é utilizada quando não existe um problema específico a ser resolvido. Todo o banco de dados é vasculhado sem qualquer relação pré-determinada, acarretando num grande número de novas relações. Estas relações são disponibilizadas, mas

necessita de uma ação analítica manual para identificar as relações que possam ser realmente úteis;

- **Testagem de Hipótese** - Quando existe um certo conhecimento da relação que ocasiona o problema, mas não é sabido se a relação é verídica, utiliza-se esta técnica para validar a hipótese;
- **Modelagem Matemática dos Dados** - Quando existe uma boa base de conhecimento para o problema, mas existe a necessidade de expandir este conhecimento, criando novas relações, utiliza-se a Modelagem Matemática dos dados.

2.3.4 – Técnicas de Datamining

Segundo CARVALHO quaisquer umas das três possíveis metodologias do datamining citadas no capítulo 2.3.3, necessitam basicamente das mesmas técnicas para sua realização:

- **Previsão** - aprender um padrão a partir de exemplos e usar o modelo desenvolvido para predizer futuros valores das variáveis selecionadas. É possível entrar com um registro de dados semipreenchidos e esperar que a ferramenta de Datamining preencha os valores que estão faltando, baseado nos padrões encontrados;
- **Classificação** - encontrar uma função que mapeia um caso em uma dentre diversas classes discretas de classificação;
- **Detecção de relações** - estabelecer relações entre variáveis;
- **Explicitação de modelos** - encontrar fórmulas explícitas descrevendo dependências entre várias variáveis;
- **Clustering** - identificar um conjunto finito de categorias ou setores para descrever dados;
- **Detecção de chaves** - determinar as escolhas mais significantes dentre algumas chaves obtidas nos dados a partir de previsões ou valores esperados;
- **Descoberta de padrões** - é o processo onde padrões são procurados em uma base de dados sem ter idéia do que se espera encontrar. O conhecimento extraído da base de dados é geralmente colocado na forma de regras IF THEN

ou na forma de associações. Por exemplo, quando o primeiro evento ocorre, o segundo evento ocorrerá também;

- **Forensync analysis** - é o processo de extrair elementos de dados que fogem dos padrões encontrados pelo processo de descoberta. Estes elementos podem significar uma exceção ao padrão.

2.3.5 – Algoritmos de Datamining

Existem muitos algoritmos que podem ser utilizados em Datamining. Em parte, estes algoritmos não são novos, mas sim originários de métodos estatísticos de Inteligência Artificial. Uma infinidade de ferramentas pode ser utilizada, dentre elas podem ser citadas: Métodos Estatísticos, Redes Neurais, Programação Evolucionária, Raciocínio Baseado em Casos, Arvores de decisão, Métodos de decisão não Linear e Regras de Associação. Os itens de 2.3.5.1 a 2.3.5.7 descrevem cada uma destas ferramentas.

2.3.5.1 – Métodos Estatísticos

Os métodos estatísticos utilizam geralmente a força bruta para extrair resultado. Segundo AURELIO [32], a comunidade estatística está geralmente interessada em interpretar seus modelos e, possivelmente, sacrificando-os de alguma performance para ser capaz de extrair significado de sua estrutura. Daí surgem às teorias da probabilidade penalizada e formulações Bayesianas. Existem diversos métodos estatísticos, alguns clássicos e outros mais recentes, sendo que todos assumem a existência de uma variável (atributo) resposta, y, e uma coleção de variáveis preditoras, $\mathbf{x} = (x1, x2, \ldots xj)$, além da disponibilidade de dados para treinamento. A meta é encontrar um modelo para prever y de **x** que funcione bem quando aplicado a um novo dado.

Alguns métodos estatísticos comumente utilizados:

- Classificadores Bayesianos;
- Redes Bayesianas;
- Arvores de decisão.

2.3.5.2 – Redes Neurais

Informalmente uma rede neural artificial é um sistema composto por vários neurônios de modo que as propriedades de sistemas complexos sejam usadas. Estes neurônios estão ligados por conexões, chamadas conexões sinápticas BARRETO [21].

Esta é uma classe grande de diversos sistemas cuja arquitetura imita estrutura do tecido neural vivo, até certo ponto construída por neurônios separados. Uma das arquiteturas mais difundidas, multilayered perceptron with back propagation, emula o trabalho de neurônios incorporados em uma rede hierárquica, onde cada neurônio de um nível é conectado com as saídas de todos os neurônios do nível anterior.

Os dados são tratados e analisados como parâmetros de excitação dos neurônios, vistos como verdadeiro alimento para os neurônios do primeiro nível. Estas excitações são propagadas a partir dos neurônios do primeiro nível para os do próximo nível, sendo ampliado ou debilitado de acordo com pesos (coeficientes numéricos) designados para as conexões intraneurais correspondentes.

Como resultado final deste processo, um único neurônio detém o conhecimento dos neurônios de nível mais alto e adquire um pouco de valor (força de excitação), sendo considerada uma predição - a reação da rede inteira para os dados processados FRAWLEY [20].

Para fazer predições significantes, primeiramente uma rede neural tem que ser treinada com dados que descrevem situações prévias. Devem ser introduzidos parâmetros e reações corretas para os neurônios. Treinamento consiste em selecionar pesos designados às conexões, o que provêem a proximidade máxima das reações produzidas pela cadeia de neurônios às reações corretas conhecidas FRAWLEY [20].

Esta aproximação provou ser efetiva em problemas de reconhecimento de imagem. Porém, pode não ser tão efetivas em aplicações que exigem maior grau de

certeza, como aplicações médicas ou financeiras. Existem várias razões para esta dificuldade.

Primeiramente, as redes neurais construídas para analisar um sistema complexo, como mercados financeiros, também são redes complexas. Elas incluem dúzias de neurônios com centenas de conexões entre eles. Como resultado, o numero de graus de liberdade do modelo (estes são os pesos de todas as conexões entre os neurônios da cadeia) freqüentemente fica maior que o número de exemplos (dados registrados separadamente) usados para treinar a rede. Isto priva o modelo de qualquer possibilidade de previsão correta. FRAWLEY [20]

2.3.5.3 – Programação Evolucionária

No momento este é o método mais jovem e conseqüentemente o algoritmo mais promissor do Datamining. A idéia subjacente do método é que o sistema formule hipóteses automaticamente, sobre a dependência da variável em relação a outras variáveis, na forma de programas expressos em uma linguagem de programação interna. Utilizando uma linguagem de programação universal a aproximação assegura que qualquer dependência ou algoritmo pode ser expresso, em princípio, nesta linguagem. FRAWLEY [20]

O paradigma evolucionário de desenvolvimento de programas é aquele em que não se consegue especificar exatamente o que se deseja antes de começar a programar. Segue a inspiração da Natureza segundo a teoria da evolução, em que não se sabe onde vai chegar. BARRETO [21]

O processo de produção de programas internos, hipóteses, é organizado como evolução no mundo de todos os possíveis programas. Quando um programa encontra uma hipótese que descreve razoavelmente bem a dependência observada, começa a apresentar várias modificações leves a este programa e selecionam os melhores programas filhos, alcançados por este processo. Assim, melhora-se a precisão da predição. Deste modo o sistema desenvolve várias linhas genéticas de programas, que

competem entre si procurando expressar com maior precisão a dependência procurada. FRAWLEY [20]

Quando o melhor programa, ou hipótese, com um nível de precisão desejado é obtido, um módulo especial do sistema traduz a dependência descoberta da linguagem interna para uma forma explícita entendida pelo ser humano: fórmulas matemáticas, tabelas, etc. Isto proporciona para o usuário uma perspicácia e controle da dependência obtida, bem como permite a visualização dos resultados. FRAWLEY [20]

2.3.5.4 – Raciocínio baseado em casos (CBR)

A idéia principal deste método é simples: para prever uma situação futura, ou para tomar uma decisão correta, os sistemas que utilizam este método encontram um caso análogo e escolhem a mesma solução que foi aplicada no caso anterior. Devido a esta razão este método é também chamado de "nearest neighbour method". Sistemas de CBR demonstram bons resultados em problemas diversos.

Por outro lado, uma grande desvantagem é que estes sistemas não criam qualquer modelo ou regras que resumem a experiência prévia. Suas predições estão baseadas em processar o conjunto inteiro de dados históricos disponíveis e, assim, para o CBR é difícil contar quais fatores específicos influenciaram na predição do sistema. FRAWLEY [20]

2.3.5.5 – Árvores de decisão

Segundo NETO [51], árvores de classificação ou de decisão são técnicas de indução usadas para descobrir regras de classificação para um atributo a partir da subdivisão sistemática dos dados contidos no repositório sendo analisado.

Segundo ANTUNES [52], as árvores de decisão são um dos métodos mais simples e bem sucedidos da aprendizagem. Uma árvore de decisão é uma representação de um conjunto de regras de classificação. Estas classificam as instâncias, ordenando-as de acordo com a árvore, desde o nó raiz até a algum dos nós terminais (folhas), que

fornece a classificação para a instância. Cada nó da árvore especifica um teste para algum dos atributos da instância (variáveis), e cada ramo descendente desse nó corresponde a um dos valores possíveis para esse atributo. Uma instância é classificada começando por testar o atributo especificado pelo nó raiz, e depois seguindo o ramo correspondente ao valor do atributo na instância. Este processo é então repetido para a sub-árvore com raiz neste novo nó.

De um modo geral, uma árvore de decisão representa uma disjunção de conjunções de restrições no valor dos atributos. Cada um dos caminhos desde o nó raiz até uma folha corresponde a uma conjunção de testes de atributos e a própria árvore a uma disjunção de conjunções. O objetivo é, portanto, gerar a árvore que melhor se identifique ao problema, ou seja, que melhor classifique as instâncias do domínio considerado.

Uma vantagem deste método é que esta forma de representação de regras é intuitiva e facilmente entendida pelo ser humano. Porém, a determinação do significado de uma regra toma-se problemática. Isto se origina no fato de que há um número menor de registros a cada nível da árvore de classificação que está sendo construída. A árvore está dividindo dados em um grande número de pequenos conjuntos de casos específicos.

Se a árvore construída está bastante "fechada": se contiver um número grande de filiais pequenas, então tal uma árvore não provê uma solução significante e estatisticamente justificada. Para aplicações em problemas complexos este método não acha uma solução satisfatória. FRAWLEY [20]

2.3.5.6 – Regras de Associação

Um dos tipos comuns de padrões que podem ser extraídos através da mineração de dados são as regras de associação. Elas representam a probabilidade de que um item apareça em um conjunto, ou transação, dado que outro está presente. BRUSSO [53]

As regras associativas foram introduzidas em AGRAWAL [54], como resultado de pesquisas conduzidas independentemente no projeto **Quest IBM** Almaden

Research Center, QUEST [55] e na Universidade de Helsinki, MANNILA [56]. Os algoritmos existentes para tal mineração buscam encontrar todas as regras que satisfaçam certos parâmetros, não somente verificar se uma regra em particular é válida. Isso porque a simples verificação tem a limitação de poder desconsiderar regras novas e mudanças nas tendências de um universo hipotético. AGRAWAL [54]

Regras associativas, simplificadamente, são declarações do tipo: "98% dos clientes que compram pneus e acessórios também efetuam revisão nos seus veículos".A regra é composta de um antecedente e um conseqüente, como se fosse uma implicação lógica; sem as conseqüências formais desta. AGRAWAL [54]

2.3.5.7 – Considerações Finais

O Modelo foi desenvolvido para que o gerente de rede encontre alguma anormalidade em dados trafegados em uma rede de computadores. Cada técnica de datamining possui características próprias e podem ser utilizados para diversas finalidades. O modelo proposto, Independe do algoritmo utilizado, pois trata o processo de datamining como uma tarefa isolada, mas, parte do pré-suposto que será utilizado o algoritmo de Regras de Associação, tendo em vista que:

- Representam a probabilidade de que um item apareça em um conjunto, ou transação, visto que outro está presente. Essas regras têm sido utilizadas principalmente no comércio varejista para a análise dos itens adquiridos pelos consumidores em uma cesta de compra. Um exemplo de tal padrão em uma rede de computadores é a declaração de que "40% dos erros no segmento X, ocorre quando a estação Y esta trafegando dados".
- Problemas de rede considerados grandes (envolvem parada total ou de algum segmento da rede), são facilmente percebidos e resolvidos. Problemas menores (impacto na performance da rede, excesso de broadcast em determinado segmento, etc.), dependendo do número de dispositivos conectados a rede, dificilmente são detectados. Os algoritmos de Regras de Associações, pelas suas características, mostraram-se úteis na percepção deste tipo de problema sem que o gerente utilize padrões preditivos.
- Disponibilidade de softwares Freeware, que trabalham com este algoritmo;

- Facilidade de compreensão, relacionando-o aos demais algoritmos;
- Facilidade na implementação, pois, os algoritmos das RA's são de teoricamente simples.

2.3.6 – Ferramentas de Datamining

Existe grande disponibilidade de ferramentas de datamining que incrementam praticamente todas as técnicas disponíveis, mas a grande maioria, possui um alto custo de aquisição e não possui demos, fatores que dificultam a análise da ferramenta.

Algumas empresas possuem versões Demos (demonstração), mas não habilitam a utilização de dados próprios (Clementine) ou é limitada no número de registros da base de dados (WizRule – 1000 registros), que impossibilita um processamento eficaz do Datamining.

A seguir são apresentadas algumas ferramentas disponíveis e uma breve característica, segundo AURÉLIO [32] e CID [75]:

- **AC2 (www.isoft.fr)** - AC2 é um conjunto de bibliotecas em C/C++ que permite a desenvolvedores e profissionais da área de TI embutir as funcionalidades de DM em seus próprios softwares. O AC2 cobre todos os passos do processo de datamining, da modelagem (modelo OO) até a validação cruzada.

- **ALICE d'ISoft (www.isoft.fr)** - É uma ferramenta de datamining de fácil utilização. Utiliza árvores de decisão para explorar os dados. Gera relatórios, queries, análise "What if", etc.

- **Clementine (www.isl.co.uk/clem.html)** - É uma ferramenta para datamining. Vencedora por duas vezes do UK Government's (Department of Trade & Industry) SMART award for innovation. As aplicações do Clementine incluem segmentação de clientes, detecção de fraudes, análise de crédito, etc.

- **Data Surveyor (www.ddi.nl)** - Data Surveyor é uma ferramenta de DM para usuários especializados. Consiste de vários algoritmos e provê suporte para

todos os passos do processo de KDD. Data Surveyor permite que o usuário descubra conhecimento interativamente, inspecione os resultados durante a descoberta e guie o processo de descoberta. As aplicações do Data Surveyor incluem banco de dados de marketing, análise de crédito e risco.

- **WizRule-** WizRule é um software para análise de dados que utiliza algoritmos de regras de inferência, exibe relatórios de regras, Pronúncia e Desvios. O WizRule é considerado um software amigável para detentores de conhecimentos básicos de regras de inferência. A versão de demonstração é limitada em (1000) mil registros.

- **XpertRule Miner** - Software gráfico que trabalha com regras de associação. Trabalho com um modelo conceitual de KDD para a preparação dos arquivos. É compatível com um grande número de banco de dados, inclusive com o Microsoft Access. A versão Demo é limitada a base de dados fornecida. Não é informado o algoritmo de mineração utilizado

- **Poly Analyst** - Software desenvolvido pela Megapunter Intelligence, necessita que o usuário tenha conhecimentos aprofundados nas técnicas de Datamining. É oferecido em quatro versões para todos sistemas Microsoft Windows: Lite, Professional, Power Knowledge. Possibilita a utilização de vários bancos de dados, inclusive formato "CSV".

- **WEKA(www.cs.waikato.ac.nz/ml/weka)** - O Framework WEKA é a sigla para "*Waikato Environment for Knowledge Analysis*". Software desenvolvido pela "University of Waikato" da Nova Zelândia. O framework WEKA, suporta vários técnicas de dataminig: decision tree inducers, rule learners, naive Bayes, decision tables, locally weighted, regression, support vector machines, instance-based learners, logistic regression, voted perceptrons. O capítulo 2.4 possui mais informações sobre o aplicativo.

2.4 – Framework WEKA

O Framework WEKA foi escolhido para o desenvolvimento deste trabalho por ser um software Freeware, desenvolvido em Java, o que o torna uma ferramenta portável e por permitir que se trabalhe com várias técnicas de datamining, podendo-se citar, entre elas, as Regras de associações, que será a técnica utilizada neste trabalho.

2.4.1 – Considerações Iniciais

O Framework Weka é uma coleção de algoritmos de aprendizagem para resolver problemas concretos de datamining, é um *pacote* em Java desenvolvido no meio acadêmico numa universidade neozelandesa e é a sigla para *Waikato Environment for Knowledge Analysis*.WAIKATO [60]

O Framework WEKA inclui implementações de vários algoritmos de aprendizagem. Estes permitem extrair informações sobre um arquivo de dados de treino, ou seja, dados preparados para o treinamento do aplicativo ou de arquivos originários de uma base de dados qualquer. JUNIOR [57]

2.4.2 – Estrutura do WEKA

O Framework WEKA é constituído por um conjunto de pacotes em Java, cada um dos pacotes possui uma função específica. O conjunto de pacotes do Framework WEKA é composto dos seguintes pacotes:

- **Weka.core** – define os objetos sobre os quais os esquemas de aprendizagem são processados. Por exemplo:
    - **Attribute** – Campos **Name, Type, Value** – corresponde a um atributo de uma tabela; Um atributo pode ser do tipo **Nominal**, **Numérico** ou **String**.
    - **Instance** – conjunto de objetos do tipo *Attribute* – corresponde a um registo de uma tabela;

- **Instances** – conjunto de objetos do tipo *Instance* – corresponde ao conjunto de dados, por exemplo uma base de dados, ou apenas a uma tabela;

- **Weka.classifiers** – *pacote* onde são definidos os algoritmos que implementam os esquemas de aprendizagem que funcionam como classificadores, ou seja, aprendizagem supervisionada.
- **Weka.clusterers** – *pacote* onde são definidos os algoritmos que implementam esquemas de aprendizagem não-supervisionada.
- **Weka.filters** – *pacote* que implementa um conjunto de ferramentas para pré-processar os dados recolhidos das bases de dados.
- **Weka.experimenter** – *pacote* com ferramentas para manipular bases de dados.
- **Weka.gui** – Interface de interação com o usuário.

O pacote weka.classifiers é formado por implementações de algoritmos de aprendizagem, tais como IBK (K-nearest neighbor), ZeroR, o Weka.classifiers.J48 que é uma implementação do algoritmo de arvore de decisão C4.5 e é o mais popular algoritmo do Framework Weka. O pacote weka.associations contém duas classes, ItemSet e Apriori, que juntos implementam esse algoritmo. O Weka.cluster contém uma implementação de dois métodos de aprendizagem não supervisionada: Cobweb e o algoritmo EM. O pacote weka.estimators contém subclasses que são utilizadas pelo Naive Bayes para computar os diferentes tipos de distribuição de probabilidade. O pacote weka.filters permite aos usuários selecionar um subconjunto de atributos ou selecionar um subconjunto de instancias de dados baseada em algum critério. QUEIROZ [59]

O Framework WEKA possui alguns itens possíveis de configuração, que tem como função modificar o entendimento das regras pelo algoritmo, estes itens são:

Sendo:

- **Delta** – Decréscimo para o suporte mínimo. Reduz o nível de suporte, até que o suporte mínimo seja alcançado;
- **LowerBoundMinSupport** – limite para o suporte mínimo;
- Metric Type – Ajusta o nível de interesse em uma determinada regra. O Frame work WEKA possui quatro opções:

- Confidence – é o percentual de dados que os itens com suporte mínimo ocorrem, quando o item de comparação (conseqüência) também ocorre.
- Lift – é a confiança(confidence), dividido pela proporção de todos os exemplos cobertos pelo item de comparação (conseqüência). A medida LIFT, independe do suporte mínimo. Exemplo:
  - Freq(A)=200, Freq(B)=100, Freq(A,B)=50, total de registros=1000
  - Confiança(A=>B) 50 /1000 = 0,05
  - Freq_rel(B) 100/1000 = 0,01
  - Lift(A=>B) 0,05/0,01 = 5
- Leverange – é a proporção dos itens adicionais cobertos pela premissa e pela conseqüência.
- Conviction – é o contrário da confiança, indica independência das regras.

O Framework Weka armazena os dados em objetos do tipo Instances, para que os dados sejam compreendidos pelo Framework WEKA. A base de dados deverá possuir um dos seguintes formatos: WAIKATO [60]

- Arquivo local no formato .arff;
- Arquivo em URL no formato .arff;
- Tabelas de Banco de Dados via JDBC.

2.4.3 – Formato .ARFF

O Framework Weka utiliza o padrão ARFF para seus arquivos de entrada, independente do algoritmo utilizado. Este padrão de representação de bases de dados garante independência e a não obrigatoriedade de ordens de precedência entre as instâncias, bem como a inexistência de inter-relacionamentos entre as mesmas. Ao mesmo tempo, permite-se que os arquivos de dados sejam verificados quanto à sua consistência.

A Figura 2.12 apresenta o formato de um arquivo com extensão ARFF. O símbolo % significa que a linha é um comentário, não tendo validade junto ao processamento realizado pelos algoritmos.

```
% 1. Title: Iris Plants Database
%
% 2. Sources:
%     (a) Creator: R.A. Fisher
%     (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%     (c) Date: July, 1988
%
@RELATION iris

@ATTRIBUTE sepallength  REAL
@ATTRIBUTE sepalwidth   REAL
@ATTRIBUTE petallength  REAL
@ATTRIBUTE petalwidth   REAL
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}
```

Figura 2.12 – Format ARFF – WAIKATO[60]

- **@relation name {nome do processo}** – Nome do processo;
- **@attribute {nome do atributo} {especificação}** - Lista de atributos do conjunto de dados, incluindo o atributo classe. Essa declaração tem a forma:
    - **@attribute nominal_atributo {primeiro_valor, segundo_valor, terceiro_valor}** - Se um atributo é nominal, a especificação contém uma lista de valores possíveis para cada atributo entre chaves.
    - **@attribute numérico_atributo numérico** - Se o atributo for numérico, a especificação é substituída por uma palavra chave numérica (valores Inteiros no Weka são tratados como números reais.)
    - **@attribute string_atributo string** - Existe também um outro tipo de atributo que é que o atributo string. Esse atributo fornece a possibilidade de armazenar um comentário ou um ID campo para cada instancia em um conjunto de dados.
- **@data**
    - Depois da declaração dos atributos os dados são iniciados por uma declaração de inicio dos dados.

- **Lista de instancias** - Uma lista de todas as instâncias. Estas instâncias são listadas em formato separado por vírgula, com uma interrogação representando os valores faltando. Comentários são representados por "%".

2.4.4 – Algoritmos Datamining

Segundo AMARAL [61], existem diversos algoritmos disponíveis que efetuam a busca de relações em um banco de dados, entre eles pode-se destacar:

- **Algoritmo DHP** - tenta reduzir o número de candidatos ao coletar contagens aproximadas no passo anterior. São necessárias tantas leituras quanto a maior cardinalidade dos conjuntos de itens freqüentes.
- **Algoritmo PARTITION** - minimiza acessos I/O ao ler a base de dados somente duas vezes. Este particiona a base de dados em pequenos pedaços que podem ser processados em memória principal. Na primeira leitura, são gerados todos os potenciais conjuntos de itens freqüentes, os quais são confirmados na segunda leitura através da contagem do suporte.
- **Algoritmo DLG** - usa um vetor de bits por item, ou seja, em quais transações cada item ocorre. Os conjuntos freqüentes são gerados através de operações lógicas AND nos vetores de bits. Entretanto, DLG assume que os vetores de bits cabem em memória principal, o que pode vir a ser um problema de escalabilidade, quando se trata de bases de dados de milhões de transações.
- **Algoritmo DIC** - Conta dinamicamente candidatos de tamanhos variados à medida que a leitura da base acontece, conseguindo assim reduzir o número de acesso aos dados. Outra contribuição desse algoritmo é a introdução do conceito de *convicção* na geração das regras propriamente ditas.
- **Algoritmo APRIORI** - criado por AGRAWAL, e utilizado pelo framework WEKA para processar as regras de associações. Será apresentando com mais detalhes no item 2.4.5.

O Framework WEKA é formado por um conjunto de implementações de algoritmos de diversas técnicas de *DataMining*, como pode ser observado na Figura 2.13. Nesta hierarquia representam-se apenas algumas das classes implementadas pelo framework Weka. Existem ainda classes que implementam outros algoritmos para

classificação, estimativas, filtragens, além de classes para interfaces gráficas. JUNIOR [57]

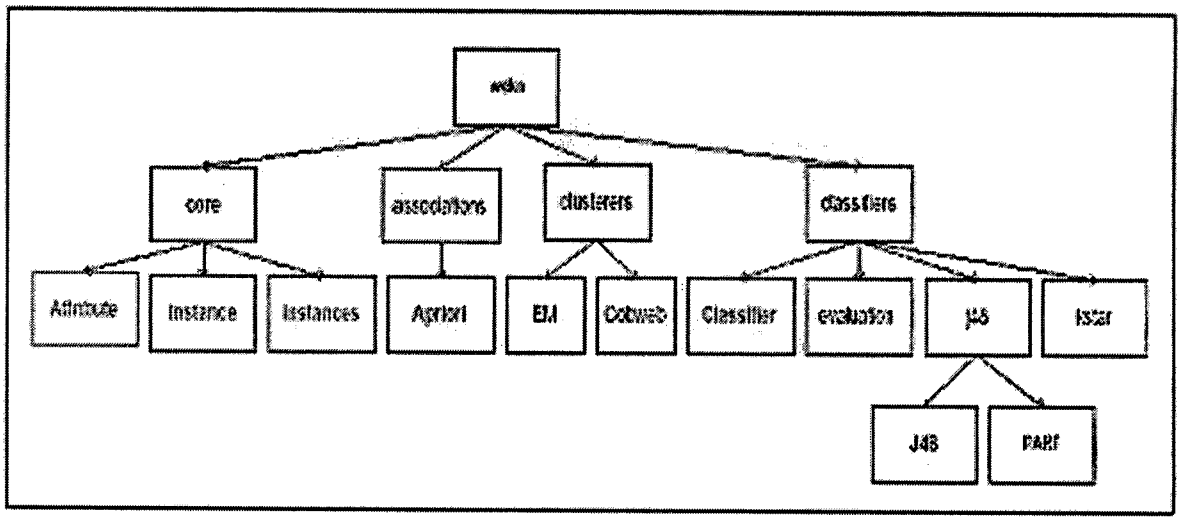

Figura 2.13 – Algoritmos suportados pelo Framework WEKA – JUNIOR[57]

Dentre os algoritmos disponibilizados pelo Framework WEKA, mostrados na figura 2.13, foi escolhida a técnica de Regras de associações, utilizando o algoritmo APRIORI, único algoritmo de Regras de Associação, disponível pelo framework WEKA, conforme será descrito no item 2.4.5

2.4.5 – Algoritmo APRIORI

O algoritmo APRIORI pode trabalhar com um grande número de atributos, gerando diversas alternativas combinatórias WESLEY [63]. Para gerar as regras, o algoritmo APRIORI consulta o banco de dados repetidas vezes, analisando os conjuntos de dados e separando-os pela freqüência de registros e atribuindo um valor de confiança para cada conjunto de dados encontrado. A partir deste ponto, gera um percentual de probabilidade de ocorrência destes conjuntos na base de dados. Segundo CARVALHO [74], Podemos resumir o procedimento em dois passos:

- **Geração** – É feita uma varredura em todo o arquivo com o intuito de gerar todos os conjuntos de combinação dos dados que estão no arquivo.
- **Poda** - São considerados apenas os conjuntos existentes, que satisfazem a freqüência mínima pré-definida, que pode ser chamada de conjuntos mais freqüentes.

A medida da freqüência de dado conjunto será chamada de *suporte*, que pode ser definida como na figura 2.14.

$$Suporte(X) = \frac{\text{Nº de registros do arquivo que contêm os elementos do conjunto X}}{\text{Nº total de registros do arquivo}}$$

Figura 2.14 – Cálculo do Suporte da regra APRIORI – CARVALHO[74]

O algoritmo APRIORI funciona como no exemplo abaixo: CARVALHO [74]

Considerando um fragmento dos dados de uma farmácia, figura 2.15, o arquivo é formado por Itens (colunas) e Venda do item para determinado cliente (linha). O valor do campo é representado pelo número 1 se o cliente adquiriu o produto; caso contrário, pelo número 0.

Transação	Gaze	Esparadrapo	Mercúrio
1	1	1	0
2	1	1	0
3	1	1	1
4	1	1	1
5	1	1	1
6	1	1	1
7	1	0	1
8	1	1	1
9	0	1	1
10	1	1	1

Figura 2.15– Tabela de transações de venda a Clientes –CARVALHO [74]

Considerando que o suporte mínimo é igual 0,6, o campo "*Gaze*" e o campo "*Esparadrapo*", possuem um suporte de 0,9 (em dez linhas, nove, possuem o valor 1), maior que o Suporte mínimo exigido (0,6). A figura 2.16 mostra os outros conjuntos, que possuem o suporte mínimo, igual ou maior 0,6.

Podemos observar na Figura 2.16, que as três tabelas possuem características diferentes:

- A tabela A: contém o suporte de um único produto;
- A tabela B: contém o suporte da ocorrência de dois produtos simultâneos;
- A tabela C: contém o suporte da ocorrência de três produtos simultâneos.

Coluna	Suporte
Gaze	0,9
Esparadrapo	0,9
Mercúrio	0,8

(a)

Colunas	Suporte
Gaze, esparadrapo	0,8
Gaze, mercúrio	0,7
Esparadrapo, mercúrio	0,7

(b)

Colunas	Suporte
Gaze, esparadrapo, mercúrio	0,6

(c)

Figura 2.16 – Subgrupo de suporte mínimo – CARVALHO [74]

Para cada grande conjunto de itens X, todas as regras com fator de confiança acima de um mínimo especificado (C_{min}) são geradas como: $\forall\ Y \subset X$, *se suporte(X) / suporte* $(X - Y) \geq C_{min}$ *então gera a regra* $X - Y \Rightarrow Y$, onde o fator de confiança de uma regra R: X $\Rightarrow$ Y, é definido pela fórmula mostrada na Figura 2.17.

$$Confiança(R) = \frac{\text{Nº de registros com X e Y}}{\text{Nº de registros com X}}$$

Figura 2.17 – Fórmula da Confiança dos dados – CARVALHO [74]

Então, se |Y| = 1 e todas as transações tiverem o mesmo valor para o item Y, então Confiança(R) é igual a 1.

Supondo um fator de confiança mínimo de 0,8 para o exemplo, tem-se como válidas apenas as regras sombreadas da Figura 2.18.

Regra	Fator de confiança
{Gaze} ⇒ Esparadrapo	0,88
{Esparadrapo} ⇒ Gaze	0,88
{Gaze} ⇒ Mercúrio	0,77
{Mercúrio} ⇒ Gaze	0,87
{Esparadrapo} ⇒ Mercúrio	0,77
{Mercúrio} ⇒ Esparadrapo	0,87
{Gaze,Esparadrapo} ⇒ Mercúrio	0,75
{Gaze,Mercúrio} ⇒ Esparadrapo	0,85
{Esparadrapo,Mercúrio} ⇒ Gaze	0,85

Figura 2.18 – Regras de confiança além do Mínimo – CARVALHO [74]

O Algoritmo APRIORI pode ser desenvolvido como mostra a figura 2.19 onde observa-se que, como são realizadas diversas consultas à base de dados, pode gerar um alto custo de I/O em grandes bases de dados.

```
Procedure Apriori

   L1 = {frequent 1-itemsets};
   for (k=2; Lk-1 ≠ ∅; k++) do
      Ck = apriori_gen(Lk-1);
      forall transactions t ∈ D do
         Ct = subset(Ck, t);
         forall candidates c ∈ Ct do
            c.count++;
         od
      od
      Lk = {c ∈ Ck | c.count ≥ min_sup};
   od
   Answer = ∪k Lk;
end
```

Figura 2.19 – Algoritmo Apriori - BRUSSO[53]

3 – PROPOSTA DO MODELO

Para elaborar o modelo KDD, teoricamente ideal ao escopo do trabalho, tratando especificamente sobre informações pertinentes a redes de computadores, conforme visto no capitulo 2.2.4, o modelo proposto será uma junção dos modelos de BRACHMAN e de FAYAD, em virtude dos prós e dos contras apresentados por cada um deles.

3.1 – Modelo Proposto

O modelo proposto, apesar de ser desenvolvido para a utilização da técnica de regras de associação, é independente da técnica e da ferramenta de datamining utilizada.

O Modelo proposto é mostrado na figura 3.2 e possui as características listadas na figura 3.1, herdadas dos modelos de BRACHMAN e FAYAD. O modelo proposto altera a ordem de algumas tarefas e busca confrontar o domínio dos dados disponíveis com o problema que se deseja sanar, gerando uma meta (objetivo) de busca. Para isto, utiliza as tarefas de descoberta da Meta e Descoberta dos dados, tarefas estas existentes no modelo segundo BRACHMAN.

Tarefa	Modelo Herdado	Observação
Descoberta da Meta e Descoberta dos dados	BRACHMAN	Exportadas do Modelo de BRACHMAN pois FAYAD não enfatiza o domínio sobre o modelo
Seleção dos Dados	FAYAD	Exportadas do Modelo FAYAD, já que o modelo BRACHMAN não deixa claro em que tarefa será feita a seleção dos dados.
Pré-Processamento e Limpeza dos dados.	BRACHMAN e FAYAD	Tanto FAYAD como BRACHMAN incluem as tarefas de Pré-Processamento e limpeza dos dados em uma única tarefa.
Limpeza dos Dados	BRACHMAN e FAYAD	Veja item acima
Mineração dos dados	BRACHMAN e FAYAD	
Pós Mineração	FAYAD e BRACHMAN	
Documentação	BRACHAM	FAYAD trata a geração de Saídas como: "Consolidação do conhecimento descoberto"

Figura 3.1 – Tarefas Herdadas

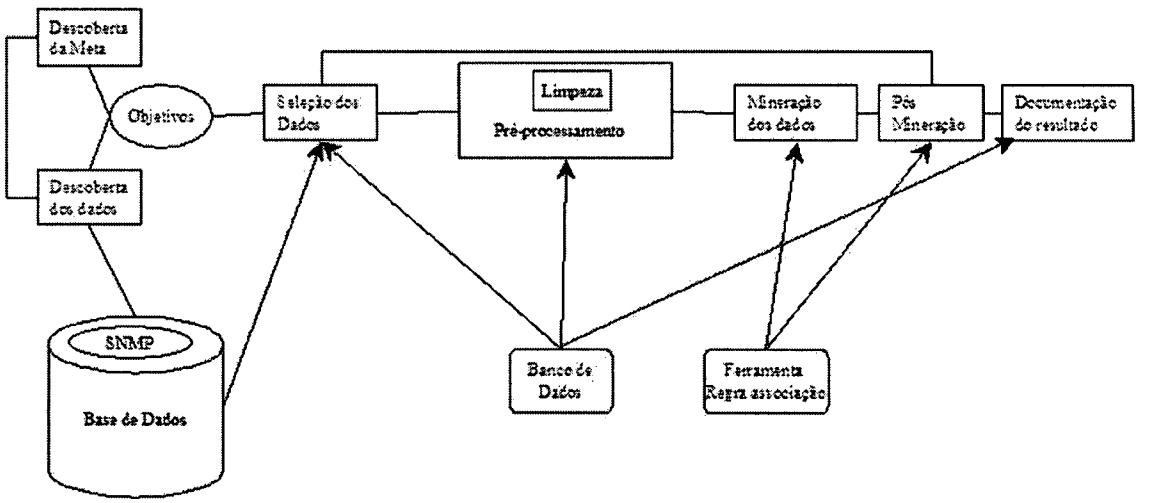

Figura 3.2 Modelo Proposto

3.2 – Estrutura do Modelo.

O Modelo proposto busca reunir a importância dada ao domínio dos dados e as ferramentas auxiliares de BRACHMAN com a versatilidade e clareza das tarefas segundo FAYAD, desta forma, é composto das seguintes tarefas:

3.2.1 - Descoberta da Meta

O objetivo desta tarefa é identificar o objetivo da execução do modelo proposto de forma clara. Como o Modelo visa encontrar problemas que estão ocultos entre os dados, pressupõe que o gerente de rede não encontrou nenhum problema evidente na rede, através do recurso de gerenciamento disponível.

3.2.2 - Descoberta dos dados:

Nesta tarefa confronta-se a meta proposta, conforme seção 3.2.1, e os dados existentes na base de dados disponíveis. Caso faltar algum dado que seja de importância fundamental à meta proposta, estes deverão ser incluídos na base de dados. Também pode ser necessário rever a meta, alterando-a e deixando-a compatível com a realidade da base de dados.

É importante ressaltar que esta etapa e a etapa de descoberta da tarefa, citada acima, são executadas quase que simultaneamente, pois a definição da meta na aplicação está intrinsecamente ligada aos dados existentes e ao seu domínio.

No caso específico das redes de computadores, a execução da tarefa de descoberta dos dados exige que o gerente de rede possua, no mínimo, conhecimentos sobre:

Arquitetura SNMP;

- Domínio dos dados disponíveis na base de dados;
- Tempo de polling dos dados (consistência da informação);
- Teor das variáveis existentes;
- Possíveis cálculos entre variáveis, que resultam em alguma informação relevante. (utilizadas na tarefa de Pré-Processamento). Ex: figura 3.4

3.2.3 - Seleção dos Dados

É nesta tarefa que se inicia o processamento das informações propriamente dito. Tem por função gerar uma base de dados otimizada, contendo somente os dados necessários para atender o objetivo proposto, conforme demonstra a figura 3.3 Como ferramenta para a consolidação da nova base de dados, poderá ser utilizado qualquer programa de banco de dados, como o Microsoft Access, por exemplo.

Para otimizar a base de dados, reduzindo o custo de processamento e, conseqüentemente, o custo de análise, poderá ser feita uma analogia às técnicas empregadas em **Banco de Dados Distribuídos**, utilizando fragmentação do banco de dados. Segundo BORTONE [64] GASPERIN [65], estas podem ser de três tipos:

- **Fragmentação Horizontal** – Divide a tabela através da divisão de suas linhas sob os critérios determinados pela meta;
  - **Ex**: na aplicação de um critério temporal, eliminando todos os registros que antecedem ao problema ocorrido.
- **Fragmentação Vertical** – Divide a tabela através da divisão de colunas em vários fragmentos;
  - **Ex**: Em uma análise associativa, na aplicação de um critério de escopo de variáveis, dependendo do objetivo estabelecido,

eliminando duas categorias que são distintas por natureza como os dados pertencentes à categoria TCP (SNMP) e UDP (SNMP)

- **Fragmentações Mistas** – São utilizadas as duas técnicas acima, onde os dados são fragmentados horizontalmente e verticalmente.

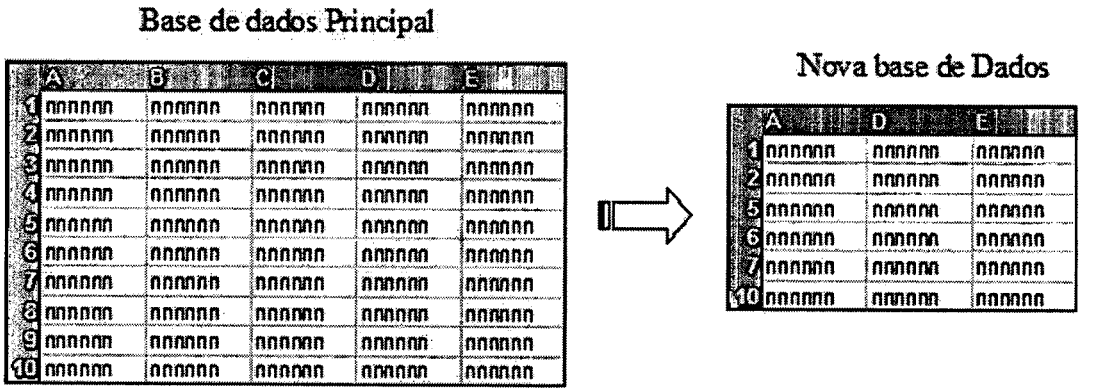

Figura 3.3 – Seleção de dados

3.2.4 - Pré-Processamento e Limpeza dos dados

Os dados selecionados da base de dados são originários de variáveis SNMP, porém, algumas informações relevantes, quando se trata de análise de dados para gerência de redes, não são fornecidos diretamente pela arquitetura SNMP, e sim, resultados de alguma operação aritmética que envolve duas ou mais variáveis como ilustrado na **Figura 3.4.**

Para calcular:

•Alto trafego IP de entrada na rede:
(ifInUcastPkts + ifInNUcastPkts) / ipInReceives

•Alta taxa de utilização de uma interface
((((ifInOctets_y - ifInOctets_x) + (ifOutOctets_y - ifOutOctets_x))/(y-x))/8)/ifSpeed

Figura 3.4 – Operação aritmética envolvendo duas ou mais variáveis SNMP.

O objetivo desta tarefa é a formatação dos dados para a entrada no mecanismo de dataminig, baseado na meta proposta, e pode ser efetuada utilizando algumas técnicas:

- **Pré-Limpeza:** Efetuar a limpeza de algumas anomalias que foram importadas juntamente com o arquivo na fase de Seleção de dados
- **Enriquecimento dos dados:** Adição de novos atributos (variáveis), derivados de atributos já existentes, conforme figura 3.4.
- **Tradução:** As regras são melhores definidas sobre valores categorizados em relação a valores quantitativos. Ex: baseado no endereço IP, pode-se definir o tipo de dispositivo de rede que os dados foram capturados, pois. em endereços Classes B, pode-se definir que o terceiro octeto diferencie os dispositivos:
    - xxx.xxx.50.xxx = Servidores
    - xxx.xxx.60.xxx = Servidores de impressão
    - xxx.xxx.70.xxx = Impressoras
    - xxx.xxx.30.xxx = Estações Clientes
- **Compatibilização de valores:** Serve para transformar todos os dados quantitativos em uma mesma unidade de análise.
    - Ex: caso os dados provenientes de alguns agentes sejam acumulativos (RMON), subtrair a variável do registro atual do registro anterior, como na **figura 3.5**.

	A	B	C
1	ipInReceives	ipInReceives Real	Fórmula
2	36980272		
3	37004343	24071	= A3 - A2
4	37026104	21761	= A4 - A3
5	37048498	22394	= A5 - A4
6	37072358	23860	= A6 - A5
7	37095842	23484	= A7 - A6
8	37122931	27089	= A8 - A7
9	37151695	28764	= A9 - A8
10	37224488	72793	= A10 - A9
11	37248205	23717	= A11 - A10
12	37271221	23016	= A12 - A11

Figura 3.5 – Cálculo de compatibilidade

- **Compatibilidade:** O modelo proposto foi concebido com o intuito de utilizar diversas ferramentas Datamining disponíveis no mercado. Cada produto pode

utilizar formatos diferentes de representação dos dados ou cabeçalho e a tarefa de pré-processamento tem como função compatibilizar a base de dados com o aplicativo Datamining a ser utilizado. Ex:

- O aplicativo WEKA trabalha com a base de dados no formato ARFF que necessita de um cabeçalho específico para efetuar o processamento, como demonstra a seção 2.4.3.

- **Discretização:** tratamento desses atributos caso haja necessidade: atributos do tipo numérico podem ser transformados de modo a tornarem-se nominais;

Para efetuar o Pré-processamento, poderão ser utilizados: SGBDs, Querys ou planilhas de cálculo.

A base otimizada pode conter informações que não serão utilizadas na fase de Datamining. Além de não servirem como parâmetros, podem acarretar um número muito grande de regras desnecessárias, caso a técnica de datamining utilizada for a de Regras de Associações. Estas informações geralmente são causadas por:

- **Problemas no dispositivo e na rede** - causada pela queda de energia elétrica, sobrecarga elétrica, problemas no módulo agente, excesso de colisão na rede, etc... Estes problemas podem gerar diversos registros com problemas, como:
  - **Valores inconsistentes** – valores fora da normalidade, como por exemplo: a média de trafego de rede é de 1000 pacotes, em dado momento em um dispositivo, o número de pacotes é de 1.000.000;
  - **Valores Nulos** – Campos que deveriam conter valores estão em branco;
  - **Caracteres Estranhos** – caracteres estranhos na base de dados, geralmente gravados em virtude da transmissão dos dados. Os dados SNMP trafegam sob o protocolo UDP;
  - **Cabeçalhos** – Cabeçalhos gerados pelo software de captura dos dados deixando os dados inconsistentes;
- **Dados utilizados na tarefa de Pré-processamento** – muitos dados, oriundos da base de dados principal, são utilizados somente para efetuar alguma equação derivada e não serão utilizados no processo de datamining.

A Limpeza dos dados é uma tarefa necessária e deve ser encarada com atenção, pois alguns dados, que a princípio parecem ser anomalias, podem ser cruciais na análise final.

A tarefa de limpeza dos dados deve ser realizada junto com a tarefa de pré-processamento, pois, pode ser utilizada, antes (eliminando cabeçalhos, caracteres estranhos, etc..), durante (eliminando cálculos auxiliares) ou no final (deixando somente os atributos e registros necessários para a tarefa de Datamining) da tarefa de pré-processamento.

3.2.5 - Mineração dos Dados (RA's)

Esta é a principal tarefa do Modelo e tem a função de extrair informações úteis de dentro de uma base de dados. O projeto utiliza como técnica de datamining, Regras de associação, mas não impede que seja utilizada qualquer outra ferramenta de datamining.

O projeto utiliza como ferramenta de datamining o framework WEKA, o qual processa as regras de associações, utilizando o algoritmo APRIORI para testar o modelo. Dependendo da semântica dos dados, bem como do tamanho da base de dados, o algoritmo APRIORI gera uma enorme quantidade de regras. Estas regras devem ser analisadas de forma criteriosa, eliminando as regras desnecessárias. Este problema é tratado na tarefa chamada Pós Mineração.

3.2.6 - Pós-Mineração

Nesta etapa são analisados os dados resultantes na tarefa de mineração e comparados ao objetivo traçado inicialmente. Caso o objetivo não tenha sido atendido, o processo deve ser reiniciado, refazendo a base de dados otimizada com outros dados ou até mesmo revendo o objetivo traçado anteriormente e alterando-o. É importante o critério na análise, pois os dados podem revelar informações importantes que não eram esperadas e não faziam parte do objetivo inicial.

Para a tarefa de Pós-mineração, as regras serão analisadas na própria ferramenta de Datamining e exportadas para um SGBD, como o Microsoft Access ou até mesmo para planilhas eletrônicas.

3.2.7 – Documentação

A principal função da tarefa de documentação é a tradução da tarefa de pós-mineração para uma linguagem comum, que possa ser entendida pelo usuário. CABENA [44] destaca que as informações mineradas pelo processo não têm validade se não for de entendimento geral, para a validação dos resultados.

A Documentação pode ser gerada em diversos formatos, desde relatórios textuais até gráficos. Também é possível, mediante análise do resultado, desenvolver ferramentas de alerta, quando determinada condição, descoberta no processo for detectada.

Para que seja efetuada a validação do modelo, o capítulo 4, demonstra um teste em uma base de dados, que teve seus dados intencionalmente alterados, para simular algum problema de rede.

4 – ESTUDO DE CASO

4.1 – Ambiente de Captura dos Dados

Os dados utilizados para a análise, foram coletados em uma rede composta de dois switches principais, provendo conexão para 150 estações de forma direta, conectadas no próprio switch, e de forma indireta, conectados através de 12 segmentos dispostos a partir dos switches principais por módulos de fibra ótica, conforme a figura 4.5. Todos os dispositivos de redes possuem agentes SNMP para gerenciamento. Os dispositivos de redes na sua grande maioria são da marca LANNET, o restante, da marca 3Com.

A rede possui quatro servidores de dados, conectados ao switch denominado: switch central.

A rede utiliza uma topologia mista, conforme observado na Figura 4.1, onde os servidores estão conectados em módulos 100 Mbits, enquanto as estações clientes em módulos de 10 Mbits.

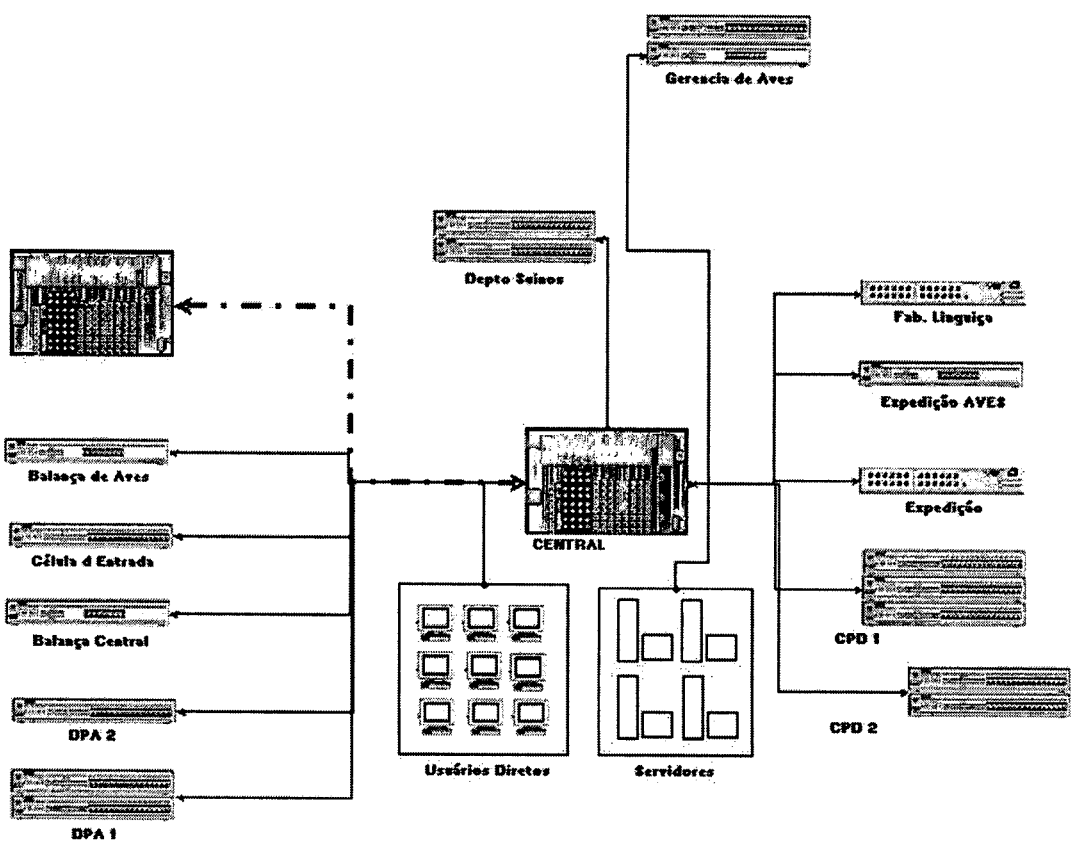

Figura 4.1 – Topologia da rede

Todos os dispositivos de redes possuem agentes SNMP, porém, os dados foram capturados a partir do switch central, pois possui todos os servidores conectados.

Os switches principais estão configurados conforme mostram as tabelas nas figuras 4.2 e 4.3 e a configuração dos distribuidores é apresentada na tabela da figura 4.4.

CPD		
Switch Chassis LET-36 LANNET		
Slot	Descrição	Tecnologia
1	Vago	
2	LSE 1008	Fastethernet
3	LHB	Propritaria 100 Mbits
4	LFE 404	Ethernet
5	LFE 404	Ethernet
6	LFE 404	Ethernet
7	LFE 404	Ethernet
8	LSE 808	Ethernet
9	LSE 108	Ethernet
10	LSE 108	Ethernet
11	LSE 108	Ethernet
12	LSE 108	Ethernet
13	LSE 108	Ethernet
14	LSE 108	Ethernet
15	LSE 108	Ethernet

Figura Figura 4.2 – Configuração Switch Central

GA		
Switch Chassis LET-36 LANNET		
Slot	Descrição	Tecnologia
1	LSE 808	
2	LSE 808	Fastethernet
3	LSE 808	Propritaria 100 Mbits
4	LFE 404	Ethernet
5	LFE 404	Ethernet
6	LSE 108	Ethernet
7	LSE 108	Ethernet
8	LSE 108	Ethernet
9	LSE 108	Ethernet
10	vago	
11	vago	
12	vago	
13	vago	
14	vago	
15	vago	

Figura 4.3 – Configuração Switch GA

Distribuidores			
Equipamento	Qtidade	Local	Marca
HUB	3	CPD 1	Lannet SHE-16
HUB	2	DPA 1	Lannet SHE-16
HUB	2	CPD 2	Lannet SHE-16
HUB	1	DPA 2	Lannet SHE-16
HUB	2	Depto Suinos	Lannet SHE-16
HUB	2	Gerencia de Aves	Lannet SHE-16
HUB	1	Fab. Lingüiça	3COM 24 Portas
HUB	1	Celula de Entrada	Lannet SHE-8
HUB	1	Expedição	3COM 24 Portas
HUB	1	Expedição Aves	Lannet SHE-8
HUB	1	Balança Aves	Lannet SHE-8
HUB	1	Balança Central	Lannet SHE-16

Figura 4.4 – Configuração Distribuidores

Slot	Porta	pacotes	Equipamento	Local
2	1	27812790	Servidor SCOCDA01 (princiapl)	CPD
3	1	5379151	SWITCH	Gerencia de Apoio
4	1	517674	HUB	Expedição
4	2	0	Livre	
4	3	533577	HUB	CelulaCélula de Entrada
4	4	0	Livre	
5	1	3086766	HUB	DPA 1
5	2	2387513	HUB	CPD 2
5	3	1670821	HUB	Depto Suinos
5	4	557157	HUB	Fab. Lingüiça
6	1	2180941	HUB	DPA 2
6	2	1297517	HUB	Gerencia de Aves
6	3	551364	HUB	Balança Central
6	4	345279	HUB	Expedição Aves
7	1	3238138	HUB	CPD 1
7	2	0	Livre	vazio
7	3	159983	HUB	Balança Aves
7	4	0	Livre	vazio
8	1	4627	Maquina cliente	CPD
8	2	96	Maquina cliente	CPD
8	3	1459813	Servidor SCOCDA02 (producao)	CPD
8	4	0	Vazio	
8	5	78674	Servidor SCOCDA03 (DPA)	CPD
8	6	954	Maquina cliente	CPD
8	7	1129145	Servidor E-mail (CC:Mail)	CPD
8	8	1601	Maquina cliente	CPD
9	1	663939	Maquina cliente (8)	Área Administrativa
10	1	822768	Maquina cliente (8)	Área Administrativa
11	1	678115	Maquina cliente (8)	Área Administrativa
12	1	433752	Maquina cliente (8)	Área Administrativa
13	1	3900235	maquina cliente (8)	Área Administrativa
14	1	455260	maquina cliente (8)	Área Administrativa
15	1	106447	maquina cliente (8)	Área Administrativa

Figura 4.5 – Lista de Segmentos da rede

4.2 – Softwares Utilizados

A rede utiliza como gerente SNMP, o software HP OPENVIEW, responsável pelo monitoramento e a captura dos dados para análise. O HP OPENVIEW permite escolher as variáveis SNMP desejadas e o tempo de polling a ser utilizado. Os dados capturados são gravados em arquivos no formato Texto.

Para testes das equações com as variáveis SNMP capturadas, foi utilizada a planilha de calculo Microsoft Excel.

Como banco de dados para manipulação da base de dados e efetivação nas equações entre as variáveis SNMP foi utilizado o Microsoft Access.

Para a edição do arquivo final e transformá-lo no formato ARFF, foi utilizado o Microsoft Notepad.

4.3 – Aplicação do Modelo

Para o teste do modelo foi utilizada uma base de dados com cinco variáveis e 2.682.718 registros, coletados entre 01/12/1997 e 31/12/1997, com um polling a cada 20 minutos. A base de dados utilizada, mostrada na Figura 4.6, possui informações oriundas da MIB privada da MADGE, conforme mostrado na figura 4.7.

Madge SMONMaster Port Statistics for 129.126.11.11 saved on Mon Dec 01 00:05:09 1997
. Values are in Pkts/Sec or Octets/Séc

Time	Slot	Port	Good Pkts	Error Pkts.
11:49:31	2	1	964	10
11:49:31	3	1	218	0
11:49:31	4	1	27	1
11:49:31	4	2	0	0
11:49:31	4	3	6	0
11:49:31	4	4	0	0
11:49:31	5	1	118	0
11:49:31	5	2	114	12
11:49:31	5	3	70	0
11:49:31	5	4	3	0
11:49:31	6	1	49	0
11:49:31	6	2	34	0
11:49:31	6	3	14	0
11:49:31	6	4	12	0
11:49:31	7	1	138	0

Figura 4.6 – Estrutura da Base de dados

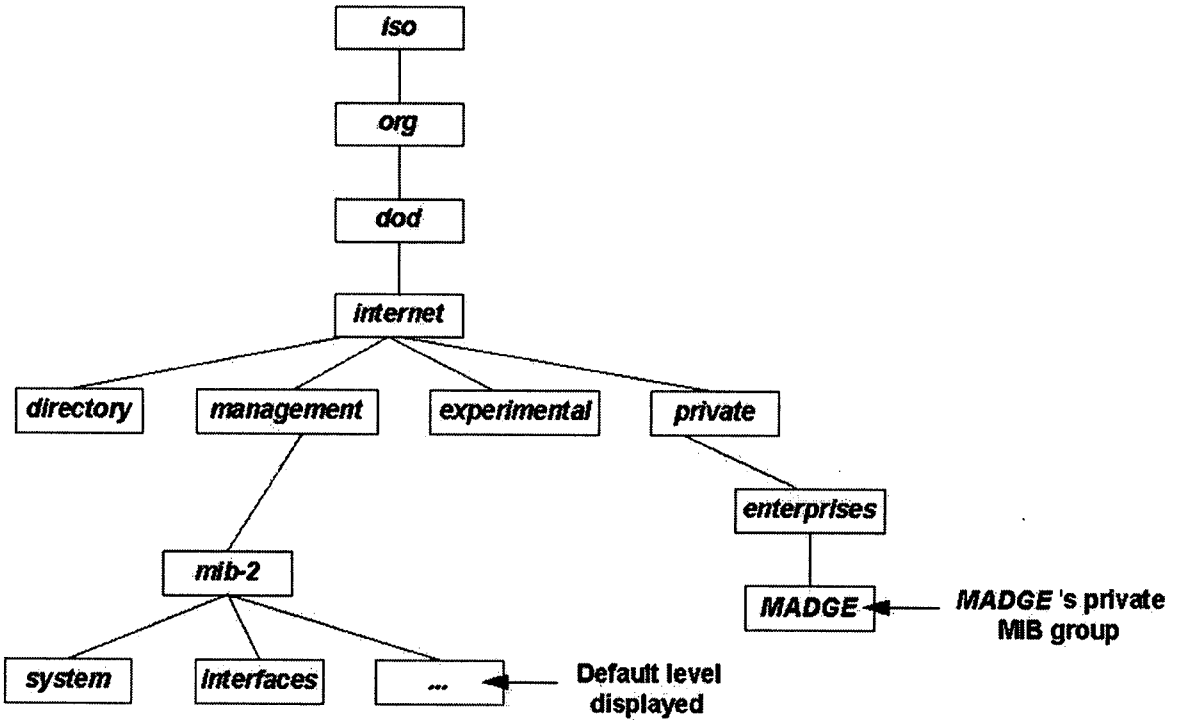

Figura 4.7 – Arvore MIB Private – Fonte: HPOPENVIEW

A partir da base descrita acima, iniciou-se a análise dos dados coletados para o teste da hipótese, utilizando o modelo proposto, conforme mostra-se nos itens de 4.3.1 a 4.3.7

4.3.1 – Descoberta da Meta

Norteando o objetivo do trabalho que é a utilização do datamining e o teste com o modelo proposto, foi utilizada uma base de dados com cinco atributos, conforme mostra a figura 4.6, e sem problemas aparentes. Para se validar o modelo, bem como a técnica de datamining aplicada, foram simulados alguns problemas na base de dados:

- Problema na máquina conectada no Slot 8 Porta 1: taxa de pacotes com erros de 70%;
- Problema no Slot 4: 40 % de pacotes com erros;
- 70% de pacotes com erros no Slot 6 porta 2: No horário que compreende o turno 3 (das 01:00 até às 08:00);

4.3.2 – Descoberta dos Dados

O arquivo gerado pelo HPOPENVIEW, conforme figura 4.6, além de possuir um cabeçalho que contém a data do início da captura dos dados e o endereço IP do dispositivo, possui 2.682.718 registros e cinco atributos como segue:

- Time: horário que foi efetuado o polling;
- Slot: Número do Slot no Chassi, a qual os módulos Switch estão conectados;
- Port: Porta de conexão no módulo Switch;
- Good pkts: Pacotes trafegados sem erros;
- Error pkts: Pacotes trafegados com erro.

4.3.3 – Seleção dos dados

A cada polling executado, o HP OPENVIEW cria um arquivo texto contendo os dados coletados. Foi efetuada uma junção de todos os arquivos gerados em um único arquivo texto, depois importado para uma base de dados estruturada, utilizando o Microsoft Access. Todos os atributos e todos os registros da base de dados foram utilizados para o processamento.

4.3.4 – Pré-processamento dos dados

Para otimizar a base de dados foram efetuadas as seguintes manipulações:

- Somente no cabeçalho de cada arquivo, existia a informação sobre o dia da semana e o dia do mês:
  - Foi desenvolvido um módulo em Visual Basic para extrair estes dados e disponibilizar como um atributo da base de dados, anexo 4.
- Foram eliminados todos os cabeçalhos e valores nulos;
- O arquivo possuía 2.682.718 registros e, por questões de desempenho, foi otimizado em número de registros, mantendo o mesmo teor da informação. O arquivo ficou com 5.544 registros:
  - Utilizando querys de agrupamento do Microsoft Access, Conforme figura 4.8, foram somados todos os pacotes que atendiam ao critério abaixo:
    - Dentro do mesmo dia da semana;
    - Utilizavam a mesma porta e o mesmo slot;

- Foram capturados dentro da mesma hora. O campo "*time*" foi truncado, mantendo somente as horas.

- O arquivo ficou com os seguintes atributos:
  - Diasemana – contendo os dias da semana(Sun, Mon, Tue, Wed, Thu, Fri, Sat);
  - Hora – contendo a somatória dos pacotes trafegados durante o período de uma hora (0, 1 ,2 ,3 ,4 ,5 ,6 ,7 ,8 ,9 ,10 ,11 ,12 ,13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23);
  - Slot – contendo o número do Slot que trafegaram os dados (2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15);
  - Porta – contendo o número da porta que o dispositivo está conectado (1, 2, 3, 4, 5, 6, 7, 8);
  - Vl_bom e vl_ruim – Pacotes bons e pacotes com erros.

```
SELECT [Dados com horas].Campo2 AS Expr1, [Dados com horas].Campo3, [Dados com
horas].Campo4, [Dados com horas].Campo5, Sum([Dados com horas].Campo6) AS
SomaDeCampo6, Sum([Dados com horas].Campo7) AS SomaDeCampo7 INTO [ports
somados]
FROM [Dados com horas]
GROUP BY [Dados com horas].Campo2, [Dados com horas].Campo3, [Dados com
horas].Campo4, [Dados com horas].Campo5;
```

Figura 4.8 – Comando SQL para agrupamento dos dados

- Simulação 1: Foi simulado um problema em uma máquina, contendo as seguintes características:
  - A máquina cliente está conectada no Slot 8(LSE-808), na porta 01;
  - Foi atualizado o atributo vl_ruins, com 70% do valor do atributo vl_bons, utilizando o critério: Slot=8 e port=1;
  - Foram atualizados 168 registros;

  Para efetivar a simulação, foi utilizado o comando SQL, como mostra a figura 4.9.

```
UPDATE [dados teste] SET [dados teste].Ruins = ([bons]*70)/100
WHERE ((([dados teste].Slot)=8) AND (([dados teste].Porta)=1));
```

Figura 4.9 – Comando SQL para simulação 1

- Simulação 2: Foi simulado um módulo com problemas, com as seguintes características:
    - Problema no módulo conectado no Slot 4 (LFE 404), este módulo segmenta dois distribuidores, conforme figura 4.5;
    - Foi atualizando o atributo vl_ruins com 40% do valor do atributo vl_bons, utilizando o critério: slot=4;
    - Foram atualizadas 672 linhas;

  Para efetivar a simulação, foi utilizado o comando SQL, como mostra a figura 4.10.

```
UPDATE [dados teste] SET [dados teste].Ruins = ([bons]*40)/100
WHERE ((([dados teste].Slot)=4));
```

Figura 4.10 – comando SQL para simulação 2

- Simulação 3: Foi simulado um problema no HUB do turno 3 (trabalha das 01:00 hs até as 08:00 hs), na gerência de aves, possuindo as seguintes características:
    - Problema no HUB conectado no Slot 6 (LFE 404), Port 2;
    - Foi atualizando o atributo vl_ruins com 70% do valor do atributo vl_bons, utilizando o critério: slot=4, port =2 e horas=entre 1 e 8;
    - Foram atualizadas 56 linhas;

  Para efetivar a simulação, foi utilizado o comando SQL como mostra a figura 4.11.

```
UPDATE [dados teste] SET [dados teste].Ruins = ([bons]*70)/100
WHERE ((([dados teste].Hora) Between 1 And 8) AND (([dados teste].Slot)=6) AND
(([dados teste].Porta)=2));
```

Figura 4.11 – Comando SQL para a simulação 3

- O algoritmo APRIORI não aceita valores e, desta forma, os valores foram substituídos por texto, conforme segue:
    - Os dados foram copiados para uma planilha do software Microsoft Excel;

        - O software aceita no máximo 65.536 linhas, o arquivo em otimização possui 5.544 linhas.
    - No atributo vl_bom:
        - Foi efetuada a média de tráfego total (soma_de_pacotes_bons/número de registro);
        - Foi definido um valor mínimo e um valor máximo, para o trafego ser considerado "dentro da média". Estes valores foram obtidos, calculando 10% para mais ou para menos do valor médio;
            - Ex: se o valor médio de tráfego for igual a 100, o valor mínimo de consideração é 90 e o máximo é de 110.
        - Foi atribuído o texto:
            - Menor – caso o valor real esteja abaixo da média;
            - Média – caso o valor esteja na faixa considerada média;
            - Maior – caso o valor estiver acima da média;
            - Zero – caso o valor for 0.
        - Fórmula utilizada: =SE(vl_bons=0;0;SE(vl_bons<vl_média;"menor";SE(vl_bons>vl_média;"maior";"media"))).
    - No atributo vl_ruim
        - Foi atribuído o texto:
            - Menor – caso o valor do atributo vl_ruim, esteja abaixo de 15% do atributo vl_bons;
            - Maior – caso o valor do atributo vl_ruim, esteja acima de 15% do atributo vl_bons;
            - Zero – caso o valor do atributo vl_ruim for 0.
        - Fórmula utilizada: =SE(vl_ruim=0;0;SE(vl_ruim<(vl_bons*15%);"menor";"maior"))
- O arquivo foi exportado para o formado Microsoft Access para facilitar a limpeza dos dados e a importação para o arquivo no formato texto com os atributos separados por vírgula;

- Utilizando o Microsoft Access, foram eliminadas as colunas: Vl_bons e a vl_ruins
- O arquivo foi exportado para o formato texto, delimitando os atributos com o sinal de virgula e retirando os qualificadores de texto.

- Utilizando um editor de Texto, foi criado o arquivo no formato ARFF, compatível com o Framework WEKA;
  - O cabeçalho do arquivo no formato ARFF, foi criado conforme a figura 4.12:

```
@RELATION dados

@ATTRIBUTE sem      {Fri,Mon,Sat,Sun,Thu,Tue,Wed}
@ATTRIBUTE hora     {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23}
@ATTRIBUTE slot     {2,3,4,5,6,7,8,9,10,11,12,13,14,15}
@ATTRIBUTE port     {1,2,3,4,5,6,7,8}
@ATTRIBUTE bom      {media,maior,menor,0}
@ATTRIBUTE ruim     {media,maior,menor,0}


@DATA
Fri,0,10,1,menor,menor
Fri,0,11,1,menor,menor
Fri,0,12,1,menor,menor
Fri,0,13,1,menor,menor
Fri,12,14,1,menor,menor
Fri,8,15,1,menor,menor
Fri,1,2,1,maior,menor
```

Figura 4.12– Arquivo dados_real.arff, no formato ARFF

Com a execução dos processos descritos acima, o arquivo está preparado para a tarefa de mineração.

4.3.5 – Mineração

Para a verificação da integridade do arquivo no formato ARFF, foi executado o programa *Weka Experiment Environment*, o qual acusou erro nos dados. Os valores do atributo bons estavam com o conteúdo "Menor" e "Média", enquanto na declaração do atributo bom (@ATTRIBUTE bom), o valor estava respectivamente: "menor" e

"media". Todo o processo foi refeito e os dados do arquivo foram alterados, pois o software não reconhece a acentuação.

No Framework Weka optou-se pela técnica de regras de associação (associate), configurado com as métricas de reconhecimento das regras conforme figura 4.13:

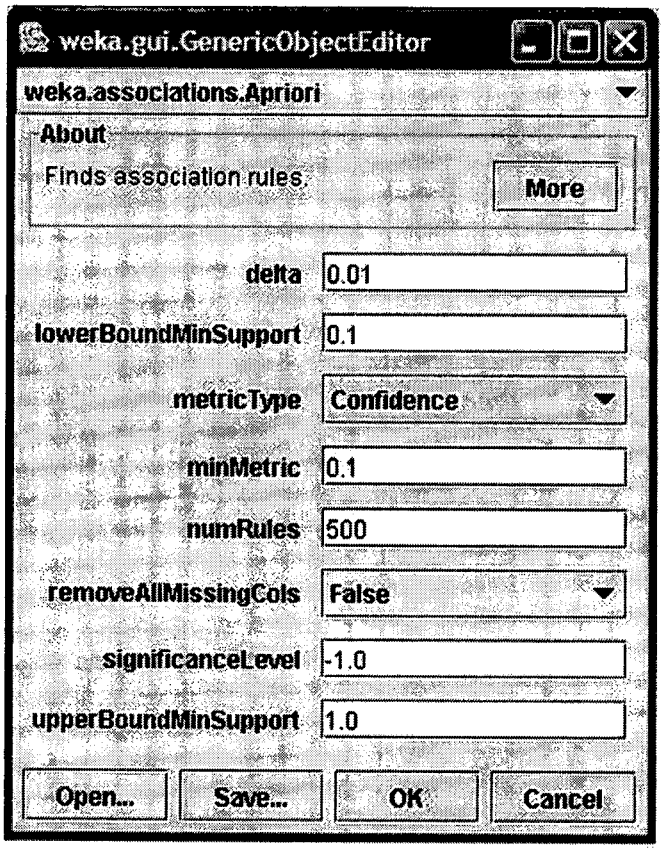

Figura 4.13 – Configuração Framework WEKA

O algoritmo foi aplicado na base de dados e demorou 1 minuto para a sua execução finalizar, utilizando um computador P III com 256 Mb de memória RAM e rodando sob o Sistema Operacional Windows XP Professional. O algoritmo APRIORI, utilizando as configurações mostradas na figura 4.12, gerou 500 regras. Destas, 70,6% (353) das regras continham na sua formação a variável "*BOM*". O Objetivo da análise foi encontrar os problemas simulados, desta forma, a variável que contém os dados de pacotes bons trafegados, pode ser eliminada.

A variável "*BOM*" foi eliminada, utilizando uma opção de filtro do próprio Framework WEKA e o processo foi refeito. Sem a variável "*BOM*", o processo, utilizando a mesma máquina, demorou iguais 1 minuto para o processamento e o

algoritmo APRIORI gerou 430 regras que serão analisadas na tarefa de pós-mineração, a seguir.

4.3.6 – Pós-mineração

O framework WEKA, gerou o relatório contendo a configuração utilizada e as regras descobertas conforme a amostra do relatório na figura 4.13 e o relatório completo no anexo V.

```
=== Run information ===
Scheme:       weka.associations.Apriori -N 500 -T 0 -C 0.1 -D 0.01 -U 1.0 -M 0.01 -S -1.0
Relation:     dados-weka.filters.AttributeFilter-V-R1-4,6
Instances:    5544
Attributes:   5
              sem
              hora
              slot
              port
              ruim
=== Associator model (full training set) ===

Apriori
=======
Minimum support: 0.01
Minimum metric <confidence>: 0.1
Number of cycles performed: 98

Generated sets of large itemsets:
Size of set of large itemsets L(1): 56
Size of set of large itemsets L(2): 134
Size of set of large itemsets L(3): 47

Best rules found:
 1. ruim=0 285 ==> sem=Sun 285    conf:(1)
 2. slot=14 168 ==> port=1 ruim=menor 168    conf:(1)
 3. slot=14 port=1 168 ==> ruim=menor 168    conf:(1)
 4. slot=14 ruim=menor 168 ==> port=1 168    conf:(1)
 5. slot=10 168 ==> port=1 ruim=menor 168    conf:(1)
...
...
...
...
 430. slot=9 ruim=menor 168 ==> port=1 168    conf:(1)
```

Figura 4.14 – Resultado da mineração

Para facilitar a busca das regras que satisfazem a condição simulada, foram utilizadas as querys do banco de dados Microsoft Access. Para reduzir a amostragem, foram selecionadas, da lista de regras, somente as regras que contivessem a string

"RUIM". Com isso, a amostra de regras foi reduzida para 308. A partir desta lista, foi iniciada a busca das regras que confirmassem a simulação efetuada.

4.3.6.1 – Simulação 1

Foram selecionadas somente as regras que atendiam o seguinte critério: Contivessem a string "RUIM", a string "SLOT=8" e a string "PORT=1". Foram encontradas cinco regras:

- Regra número 53. slot=8 ruim=maior 146 ==> port=1 146 conf:(1)
  - Conhecimento:
    - O slot 8 trafegou 146 pacotes ruins considerados "maiores" e deste tráfego total, 146 foram trafegados na Porta 1. Confiabilidade da regra de 100%
  - A regra comprova a simulação efetuada, pois os 22 registros restantes, foram no domingo, onde não existiu tráfego de rede.

- Regra número 87. slot=8 port=1 168 ==> ruim=maior 146 conf:(0.87)
  - Conhecimento:
    - De 168 ocorrências na base de dados que compreendem dados trafegados no slot 8 e na porta 1, 146 estavam com pacotes ruins considerados maiores. Confiabilidade da regra de 87%.
  - A regra também comprova a simulação, pois novamente os 22 registros faltantes eram pacotes que compreendiam o Domingo, sem tráfego na rede.

- Regra número 149. port=1 ruim=maior 307 ==> slot=8 146 conf:(0.48)
  - Conhecimento:
    - De 307 ocorrências com a porta 1 e pacotes ruins, considerados maiores, 146 ocorrências foram no slot 8. Confiabilidade da regra de 48%.
  - Regra descartada como Ideal, pois a porta 1 se encontra em outros Slots além do Slot 8. Figura 4.4.

- Regra número 284. ruim=maior 810 ==> slot=8 port=1 146 conf:(0.18)
    - Conhecimento
        - Das 810 ocorrências com pacotes considerados ruins, 146 ocorreram no slot 8 porta 1. confiabilidade da regra de 18%.
    - Regra descartada, pois outros dispositivos trafegaram pacotes ruins.

- Regra número 423. slot=8 1344 ==> port=1 ruim=maior 146 conf:(0.11)
    - Conhecimento
        - Das 1344 ocorrências no slot 8, 146 foram na porta 1 e eram pacotes ruins. Confiabilidade da regra de 11%
    - Regra descartada, pois existem pacotes bons trafegados em outras portas do slot 8.

4.3.6.2 – Simulação 2

Foram selecionadas somente as regras que contivessem a string "RUIM" e a string "SLOT=4". Foram encontradas 25 regras:

- **Conjunto de Regras 1:**
    - 21. slot=4 port=3 168 ==> ruim=maior 168 conf:(1)
    - 63. slot=4 port=1 168 ==> ruim=maior 161 conf:(0.96)
    - 104. slot=4 port=4 168 ==> ruim=maior 144 conf:(0.86)
    - 105. slot=4 port=2 168 ==> ruim=maior 144 conf:(0.86)

Estas regras foram descartadas pois a somatória das 4 regras, elucidaria o problema da simulação 2, mas, tratadas individualmente é improvável.

- **Conjunto de Regras 2:**
    - 179. slot=4 ruim=maior 617 ==> port=3 168 conf:(0.27)
    - 184. slot=4 ruim=maior 617 ==> port=1 161 conf:(0.26)
    - 240. slot=4 ruim=maior 617 ==> port=4 144 conf:(0.23)
    - 241. slot=4 ruim=maior 617 ==> port=2 144 conf:(0.23)

Regras descartadas, pois, tendo como um dos precedentes da regra, os registros com pacotes ruins (617), juntamente com o slot 4, é improvável que seja possível

caracterizar o problema da simulação 2, pois a representatividade do problema, que está individualizado por porta, é muito pequena, tendo em vista que o problema ocorre em todo o modulo.

- **Conjunto de Regras 3:**
  - 22. port=3 ruim=maior 168 ==> slot=4 168 conf:(1)
  - 262. port=3 840 ==> slot=4 ruim=maior 168 conf:(0.2)
  - 262. port=3 840 ==> slot=4 ruim=maior 168 conf:(0.2)
  - 56. port=4 ruim=maior 144 ==> slot=4 144 conf:(1)
  - 294. port=4 840 ==> slot=4 ruim=maior 144 conf:(0.17)
  - 140. port=2 ruim=maior 191 ==> slot=4 144 conf:(0.75)
  - 295. port=2 840 ==> slot=4 ruim=maior 144 conf:(0.17)
  - 148. port=1 ruim=maior 307 ==> slot=4 161 conf:(0.52)

Regras descartadas, pois o problema abrange todo o slot 4, que é composto de 4 portas, entretanto, outros dispositivos (slots) possuem portas nominadas como 1,2,3 e 4 conforme figura 4.5, e não fazem parte da simulação.

- **Conjunto de Regras 4:**
  - 255. ruim=maior 810 ==> slot=4 port=3 168 conf:(0.21)
  - 280. ruim=maior 810 ==> slot=4 port=1 161 conf:(0.2)
  - 286. ruim=maior 810 ==> slot=4 port=4 144 conf:(0.18)
  - 287. ruim=maior 810 ==> slot=4 port=2 144 conf:(0.18)
  - 139. ruim=maior 810 ==> slot=4 617 conf:(0.76)

Descartadas, pois, tendo como único precedente da regra, pacotes ruins (810), não fica caracterizado o problema no Slot 4, pois analisa todos os pacotes ruins existentes e os confronta individualmente com as portas do slot 4.

- **Conjunto de Regras 5:**
  - 206. slot=4 672 ==> port=3 ruim=maior 168 conf:(0.25)
  - 234. slot=4 672 ==> port=1 ruim=maior 161 conf:(0.24)
  - 247. slot=4 672 ==> port=4 ruim=maior 144 conf:(0.21)
  - 248. slot=4 672 ==> port=2 ruim=maior 144 conf:(0.21)

Regras descartadas, pois a somatória das 4 regras (206, 234, 247 e 248), elucidaria o problema da simulação 2, mas, tratadas individualmente, não caracteriza o problema simulado.

- **Conjunto de Regras 6:**
  - 139. ruim=maior 810 ==> slot=4 617 conf:(0.76)
    - Não é possível definir com clareza o problema da simulação 2, mas, devido a regra comparar individualmente os pacotes ruins e o slot 4 e, o percentual de erros do slot 4 é bastante alto (82%), é provável que esta regra exija uma análise mais aprofundada.

  - 79. slot=4 672 ==> ruim=maior 617 conf:(0.92)
    - A regra 79, **comprova claramente** o problema da simulação 2, pois, dos 672 registros do suporte do slot 4, 617 eram pacotes ruins.

4.3.6.3 – Simulação 3

Foram selecionadas somente as regras que atendiam o seguinte critério: Contivessem a string "RUIM", a string "SLOT=6", a string "PORT 2" e a strings "1,2,3,4,5,6,7,8". Foram encontradas 8 regras:

- 131. hora=8 232 ==> ruim=menor 184 conf:(0.79)
- 132. hora=7 231 ==> ruim=menor 183 conf:(0.79)
- 133. hora=2 231 ==> ruim=menor 183 conf:(0.79)
- 134. hora=6 231 ==> ruim=menor 179 conf:(0.77)
- 135. hora=4 231 ==> ruim=menor 178 conf:(0.77)
- 136. hora=3 231 ==> ruim=menor 178 conf:(0.77)
- 137. hora=1 232 ==> ruim=menor 177 conf:(0.76)
- 138. hora=5 231 ==> ruim=menor 176 conf:(0.76)

Nenhuma das regras evidenciou a simulação 3, pois o suporte da simulação 3 era muito baixo, 56 registros, não apropriado para o algoritmo APRIORI.

4.3.7 – Documentação

Os resultados obtidos com o estudo de caso foram considerados positivos, pois das três simulações efetuadas, duas foram claramente evidenciadas nas regras:

- Simulação 1 - As regras 53 e 87 evidenciam claramente o problema simulado;
- Simulação 2 – A regra 79 evidencia claramente o problema simulado, enquanto a regra 139, sugere que pode existir um problema, mas necessita de uma análise mais específica;
- Simulação 3 – Nenhuma regra apresenta o problema com clareza, pois o número de registros alterados, em relação ao número de registros existentes, era muito pequeno, para que o algoritmo pudesse gerar uma regra clara do problema.

A tarefa de documentação, não foi levada em consideração nesta análise de dados, pois visa transpor o resultado da pós-mineração para uma linguagem comum, que possa ser entendida pelo usuário não especialista, somente foi relatado o resultado encontrado de forma sucinta.

5.0 – CONCLUSÃO

A realização deste trabalho envolveu o estudo de duas tecnologias teoricamente distintas, Gerência de redes e Datamining. Foi desenvolvido um modelo KDD, baseado nos modelos conceituais de FAYAD e BRACHAMN, para a aplicação de algoritmos de datamining sobre uma base de dados em gerência de redes.

A heterogeneidade dos dispositivos de rede, das tecnologias de redes e dos aplicativos de gerência de redes, dificulta a padronização de um procedimento para descoberta de problemas, bem como o grande número de informações disponíveis, acabam por comprometer o entendimento destas informações, tendo em vista que muitas são relevantes, estando escondidas dentro dos próprios dados. Uma das alternativas foi utilizar técnicas de Datamining para extrair estas informações e dar ciência do problema a quem de direito.

Para que fosse possível aplicar esta técnica, foi necessário organizar as informações coletadas em estado bruto, independente do software de captura dos dados, e disponibilizá-las de forma organizada, transparente, homogênea, e completa em relação ao objetivo desejado. Para este fim, existem algumas técnicas de KDD disponíveis no mercado; entre as mais utilizadas estão a abordagem do KDD segundo FAYAD e BRACHMAN, mas, desenvolvidas para trabalhar com informações comerciais. Desta forma, foi criado um novo modelo de KDD, voltado para a gerência de redes.

O presente trabalho, optou pela técnica de Regras de associação, pois o objetivo secundário, foi encontrar a correlação de dados potenciais para a solução de algum tipo de problema, revelados por meio de regras criadas a partir da base de dados. Para o processamento das RA's, foi utilizado o algoritmo APRIORI, base dos demais algoritmos de RA's. Esse algoritmo, apesar possibilitar que o objetivo traçado, fosse alcançado, não se mostrou o ideal para casos onde a base de dados é formada por valores quantitativos, caso da base de dados utilizada. Devido a esta fragilidade, foi necessário a substituição dos valores trafegados (numéricos), por valores representativos(Maior, Média, Menor e Zero), conforme item 4.3.4.

Para que fosse possível a análise dos dados, foi necessário um software que aplicasse o algoritmo. Um grande problema, foi encontrar algum software disponível e

freeware; a maioria dos softwares encontrados, eram versões Trial ou Demos, limitadas ao número de linhas (registros) para o processamento.

O software escolhido foi o Framework WEKA, que trabalha com diversas técnicas de datamining e possui uma performance aceitável (1 minuto pra processar 5456 linhas). O maior problema na utilização do Framework WEKA, foi encontrar a documentação integral, já que o software é freeware, mas os autores comercializam o livro.("Datamining: Pratical machine learning tools with Java implementations" by Ian H. Witten, Eibe Frank e Morgan Kaufmann), mas este livro não foi encontrado para a aquisição..

O modelo proposto tratou de aliar o comprometimento com o domínio do problema (conhecimento do problema e conhecimento dos dados) e a ação das ferramentas auxiliares do modelo segundo BRACHMAN, com a definição de forma clara e a ordenação das tarefas do modelo segundo FAYAD.

Para validar o modelo desenvolvido, foi efetuada uma análise de dados, utilizando dados capturados em uma rede gerenciada. Foi escolhida uma base de dados simples, contendo cinco variáveis de fácil compreensão. A partir desta base, foram simulados alguns problemas na rede e a base de dados foi submetida ao modelo proposto.

Foram efetuadas três simulações de problemas na base de dados, e o algoritmo APRIORI, documentou, em forma de regra, claramente, dois problemas. O terceiro problema não foi encontrado, pois a representatividade dos dados era muito pequena em relação à base de dados, levando a consideração que o algoritmo APRIORI, gera as regras, baseado no número de ocorrências de determinado conjunto de dados, na base de dados.

O modelo passou por adequações no decorrer da análise dos dados, pois algumas tarefas que teoricamente possuíam estava na ordem correta de execução, tiveram que ser alteradas, é o caso da tarefa de limpeza dos dados, que inicialmente estava precedendo a tarefa de pré-processamento e que, com o decorrer dos testes, foi colocada após a tarefa de Pré-processamento; e no modelo final, se adequou melhor junto a tarefa de pré-processamento.

5.1 – Sugestões para trabalhos futuros

A partir da experiência obtida com o desenvolvimento deste trabalho, podem-se sugerir alguns experimentos, testes e pesquisas que servirão para melhorar os resultados aqui obtidos:

- Utilizando o Modelo proposto e uma base de dados simulada, aplicar outras técnicas de dataminig existentes, tentando encontrar qual técnica é realmente a mais indicada para a gerência de redes;
- Utilizando o modelo proposto, testar uma base de dados, utilizando um algoritmo de regras de associações, que entendam valores quantitativos;
- Utilizando o modelo proposto, testar uma base de dados, utilizando os diversos algoritmos de Regras de associações disponíveis, para verificar qual algoritmo é o mais indicado para a gerência de redes.

Este trabalho não esgotou o universo de pesquisas e experimentos que podem ser realizados na união das áreas de gerência de redes de computadores e de datamining; ao contrário, espera-se que ele sirva de base para outros desenvolvimentos e que possa ser utilizado por administradores de redes ou mesmo que possa ser utilizado como exemplo da aplicação do datamining em sistemas de gerenciamento em outras áreas.

6.0 - FONTES BIBLIOGRÁFICAS

[1] TANENBAUM, A. S. **Redes de Computadores,** 1997, Editora Campus, 4ª edição.

[2] SOARES, L.F.G, LEMOS, G, COLCHER, S. **Rede de Computadores das LANs MANs e WANs às Redes ATM,** 2ª Edição Revisada e Ampliada, Rio de Janeiro, Campus, 1995.

[3] BRISA, **Gerenciamento de Redes – uma abordagem de sistemas abertos**, 1993, Makron Books, São Paulo.

[4] STALLINGS, W. **SNMP, SNMPv2, SNMPv3 and RMON 1 and 2**, 1999, 3ª edição, Addison Wesley Longman, Inc.

[5] FRANCESCHI, A. S. M. **Desenvolvimento de Agentes Autônomos para Gerência de redes de Computadores**, Tese de Doutorado, Florianópolis, Maio de 2000.

[6] Da SILVA, A. C., **"Sistema de Relatório de Auxilio à Gerência de Redes"**, Sistema desenvolvido na Sadia/Concórdia, em 1998, utilizando o software Microsoft Access, organizado em 1 MB (vazio) e 700 MB (contendo dados de 1 mês).

[7] ZACKER, C, DOYLE, P, **Redes de Computadores Configuração, Manutenção e Expansão**, 1ª edição, São Paulo, MAKRON Books, 2000.

[8] MAZZOLA, V.B., **Apostila Arquitetura de Redes de Computadores**, DEZ/99, CPGCC-UFSC, Florianópolis.

[9] SILVA, Mário Gomes, **Novell Netware 3.12**, São Paulo-Sp, Ed. Érica.

[10] ROESLER, Valter, **Apostila Redes Locais de Computadores**, UNISINOS, Março 1999.

[11] SAMPAIO, S.,C., **Plataforma para concepção de aplicações de gerencia de redes utilizando o SNMP**, Projeto Específico, UNIFACS, Bahia, 1997.

[12] LEINWAND, A.; CONROY, K. F. **Network Management. A Practical Perspective**. 2.ed. Menlo Park: Addison-Wesley, 1996.

[13] ROSE, M., MCCLOGHRIE, K. **Management Information Base for Network Management of TCP/IP-based internets**. RFC 1156. Network Working Group. 1990.

[14] CARVALHO C. C.. **RMON permite gerenciar redes remotas**. LANTIMES Brasil, vol. 3, edição 1, pp. 24-25, Abril 1997.

[15] WALDBUSSER, S. **Remote Network Monitoring Management Information Base**. RFC 1271. Carnegie Mellon University. 1991.

[16] NETO, F. W., **Aplicando técnicas de Série temporais em gerenciamento pró-ativo de Redes de Computadores**. Anais do simpósio de Redes de Computadores. Rio de Janeiro(RJ). Maio 1998.

[17] FAYAD, U. M., PIATETSKI-SHAPIRO, G., SMITH, P., UTHURUSAMY, R. **Advanced in Knowledge Discovery and Datamining**, American Associates for Artificial Intelligence, Menlo Park, California, EUA, 1996.

[18] LANGLEY, P, & SIMON, H. A., **Application of machine learning an rule induction. Communications of ACM**, Vol 38, No 11, Novembro 1995.

[19] MEGAPUNTER. URL: http://www.megapunter.com, Julho de 2001.

[20] FRAWLEY, W. J., PIATETSKI-SHAPIRO, G., AND MATTHEUS, C.J. 1991. **Knowledje discovery in Databases**, ed Piatetsky-Shapiro and B. Frawley. Cambridge, Mass, AAAI/MITpress 1-27.

[21] BARRETO, J. M., **Inteligência artificial – No limiar do Século XXI**, Florianópolis-SC, 1997.

[22] REDISUL, **ASS-BUILD Rede Sadia**, Concórdia, 1996.

[23] COMPUTER WORLD, http://www.uol.com.br/computerworld/technology/qstudy/9908datamining.htm, Acesso em: 30/05/2001.

[24] Sade, Alberto Sualaiman & Souza, Jano Moreira de, **Prospecção de Conhecimento em Base de Dados Ambientais.**

[25] Mannila, H. Et alii. **Finding Interesting Rules From Large Sets of Discovered Association Rules**, 3 rd International Conference on Information and Knowledge Management, novembro 1994.

[26] Lima, Michele Mara de A. Espíndula, **RNP News Generations Vol 1/ N° 7 - Gerenciamento de redes**, http://www.rnp.br/newsgen/9712/gerencia.shtml.

[27] CARVALHO, Luis Alfredo Vidal de,**Datamining: a mineração de dados no marketing, medicina, economia, engenharia e administração**, São Paulo: Érica, 2001.

[28] GHEDINI, Guellner Cinara, **Um Modelo de Apoio à Documentação de Aplicações de Descoberta de Conhecimento em Base de Dados**, Dissertação de Mestrado, Pontífica Universidade Católica do Rio grande do Sul, Porto Alegre, 2000.

[29] PRAS, A. **Network Management Architectures**. CTIT Ph. Thesioras no. 95-02,

Centre for Telematics and Information Technology, The Netherlands, 1995.

[30] Harnedy, S., Total SNMP - **Exploring the Simple Network Management Protocol**, Prentice Hall, Upper Saddle River, NJ, 1997.

[31] VORUGANTI, R.R. **A Global Network Management for the 90's**. IEEE Communications Magazine, pp. 74-83, Aug., 1994.

[32] Aurélio, M. Vellasco, M. Lopes, Carlos Henrique, **Descoberta de conhecimento e mineração de dados**, Pontífica Universidade Católica do Rio de Janeiro, iro, 1999.

[33] G. Piatetsky-Shapiro. **Knowledge discovery in real databases: A report on the IJCAI**-89 Workshop. AI Magazine, Vol. 11, No. 5, Jan. 1991, Special issue, 68-70.

[34] Andrassoyvá, Eva, Parlic, J, **Knowledge Discovery in Databases: A Comparison of Different Views**. Technical University of Košice, Slovakia, Project 977091.

[35] Specialski, Elizabeth S., **Gerência de redes de computadores e telecomunicações**, UFSC – Universidade Federal de Santa Catarina, Florianópolis.

[36] UDUPA, D. K. **Network Management Systems Essentials**. New York: McGraw-Hill, 1996.

[37] HAN Jiawei, CAI Yandong, CERCONE Nick. **Data-Driven Discovery of Quantitative Rules in Relational Databases**. IEEE Transaction on Knowledge and Data Engineering, v. 5, n.1, p. 29-40, Dez. 1993.

[38] AGRAWAL, Rakesh, RAMAKRISHNAN, Srikant. **Fast Algorithms for Mining Association Rules. In: VLDB CONFERENCE**, 20th,1994, Santiago, Chile. Proceedings. Santiago : [s.n.], 1994. P. 487-499.

[39] JAIN, Anil, DUBES, R. **Algorithms for Clustering Data**. Englewood Cliffs, New Jersey, Prentice-Hall, 1988.

[40] FAYYAD, Usama, PIATETSKY-SHAPIRO, Gregory, SMYTH Padhraic. **The KDD Process for Extracting Useful Knowledge from Volumes of Data**. Communications of the ACM, New York : ACM, Nov 96, v. 39, N. 11. p. 27-34.

[41] FAYYAD, Usama, PIATETSKY-SHAPIRO, Gregory, SMYTH Padhraic. **From Data Mining to Kwonledge Discovery : An Overview. Advances in Knowledge Discovery and Data Mining**. AAAI/MIT Press, Califórnia, USA, 1996. p. 1-30

[42] FAYYAD, Usama, PIATETSKY-SHAPIRO, Gregory, SMYTH Padhraic. **From Data Mining to Knowledge Discovery in Databases**. American Association of Artificial Intelligence. p.37-54, 1996.

[43] BRACHMAN, Ronald, KHABAZA, Tom, KLOESGEN, Willi, **Mining Business Databases**. Comunication of the ACM, New York : ACM, v. 39, n. 11, p. 42-48, nov. 1996.

[44] CABENA Peter; HADJINIAN Pablo. **Discovery Data Mining from Concept to Implementation**. New Jersey : Prentice-Hall, 1998. p. 195.

[45] MESQUITA Ricardo L. S., Goldzmit Daniel S., **Datamining**, Universidade Estadual Paulista, São Paulo-SP, 1999.

[46] Piatetsky-Shapiro and W.J. **Farewell Knowledge Discovery in Data mining,** AAAI/MIT Press, 1991

[47] FÉLIX Luis Carlos M., **Dataminning no processo de extração do conhecimento dos dados,** USP, São Carlos-SP, 1998.

[48] SANTOS Jose, Henriques N, Reis Vanda, **DataMinning/DataWarehouse,** http://students.fct.unl.pt/users/nuno/dmdw/, Universidade Nova de Lisboa, Lisboa-PT, 1999.

[49] Loyolla W., **Avaliação do Estado da Arte e Produtos Datamining**, Universidade Católica de Brasil, UCB, Brasilia, DF, 2000.

[50] DW, Equipe - DataWarehouse – Data Mining", http://www.datawarehouse.inf.br/, artigo: **Uma breve história do Data Mining?**, visita em 22/10/2002.

[51] Neto, Manoel G. M., **Mineração de dados**, Material publicado nos Anais da 6a Escola Regional de Informática da Regional 2 do Estado de São Paulo, São Carlos, São Paulo, Abril de 2001.

[52] ANTUNES, Claudia M., **Sistema de aquisição de conhecimento para apoio à consulta de sub-divisão**, Dissertação para a obtenção do grau de Mestre em Engenharia Electrotécnica e de Computadores, Lisboa, Portugal, abril/2001.

[53] BRUSSO, Marcos J., Navaux, Philippe O. A. , Geyer, Claudio F.R., **Um Modelo para a Descoberta de Regras de Associação Aplicado à Mineração do Uso da Web**, Universidade Federal do Rio Grande do Sul, Porto Alegre-RS, 2000.

[54] AGRAWAL, R. & IMIELINSKI, T. & SWAMI, A., **Mining Association Rules Between Sets of Items in Large Databases**, disponível em ACM SIGMOD Conf. Management of Data, 1993.

[55] **QUEST DATA MINING PROJECT**, IBM Almaden Research Center, disponível em http://www.almaden.ibm.com/cs/quest, 1999.

[56] MANNILA, H. & TOIVONEN, H. & VERKAMO, A. I., Efficient Algorithms for Discovering Association Rules, disponível nos anais do KDD94, 1994.

[57] JUNIOR, Olival G. F, Martins, Jefferson G., Rodrigues, Alejandro M., Barcia, Ricardo M., **Sistema de apoio a decisão usando a tecnologia datamining com estudo da Universidade Estadual de Maringá** , artigo I Congresso Brasileiro de Computação, Maringa, PR, 2001

[58] Site da biblioteca WEKA. Disponível em http://www.cs.waikato.ac.nz/~ml/weka/, 2002.

[59] QUEIROZ, Ana E. M., **Identificação do tipo de estratégia de análise de dados mais adequada para ser implementada no desenvolvimento do servidor ADeCUI**, Universidade Federal de Pernambuco, Recife, PE, 2002

[60] WAIKATO, URL http://www.cs.waikato.ac.nz/~ml/weka/arff.html, visitada em 2002.

[61] AMARAL, Eduardo P., OLIVEIRA, Claudia M. G. M. De, **O Problema da Mineração de Regras Associativas em Bases de Dados**, Instituto Militar de Engenharia Departamento de Engenharia de Sistemas, Rio de Janeiro-RJ, 2000.

[62] AGRAWAL, R., SHAFER, J. C. **Parallel mining of association rules**. IEEE Transactions on Knowledge and Data Engineering, vol. 8, NO. 6, December 1996.

[63] WESLEY Romão, Niederauer Carlos A. P. , Martins Alejandro, Tcholakian Aran, Pacheco Roberto C. S., Barcia Ricardo M, **Extração de Regras de associação em C&T: O algoritmo APRIORI.**, Universidade Federal de Santa Catarina, Centro Tecnológico, Florianopolis-SC.

[64] Bortone, Carlos R., **Estudo sobre a distribuição de dados no sistema de gerenciamento de contratos**, Pontifica Universidade Católica de Minas Gerais, Belo Horizonte-MG, 2000.

[65] GASPERIN, Caroline V., **Banco de Dados Distribuídos**. Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre-RS.

[66] STALLINGS, WILLIAM. **SNMP and SNMPv2: The Infrastructure for Network Management**. IEEE Communications Magazine. New York, v. 36, n. 3, p. 37-46, March 1998.

[67] GASPARY Luciano P., **Gerenciamento de protocolos de alto nível e aplicações distribuídas**, E.Q. n. 37 - PGCC-UFRGS, Porto Alegre – RS.

[68] MENDONÇA, Flavio D., **SNMP: Estrutura, protocolo e aplicação**, Universidade Federal do Rio de Janeiro.

[69] VERONEZ, Cleversson A., **Gerência de Desempenho do tráfego de redes utilizando redes bayesiana**, UFSC – Universidade Federal de Santa Catarina, Florianópolis-SC, 2000.

[70] ROCHA, M.A., WESTPHALL, C.B, **Proactive management of computers networks using Artificial Intelligence Agents and Techniques**. Proceedings of the Simposium on Integrated Network Management. San Diego-CA, USA, 1997.

[71] Dias Zanella B., Alves Nilton, **Protocolo de gerenciamento SNMP**, artigo documento CBFT-NT-006/01, UFRGS, Porto Alegre-RS, 2002.

[72] PROENÇA Mario L., **Uma Ferramenta para Auxílio no Gerenciamento de Redes com Backbone ATM**, UFRGS – Universidade federal do Rio grande do sul, Porto Alegre-RS, 1998.

[73] MANNILA, H., **Dataminig : machine learning, statistcs and databases**, Eight International conference on Scientific and Statistical Database Management, Stockholm-Swedem, 1996.

[74] CARVALHO, Juliano V., SAMPAIO, marcus C., MONGIOVI, Giuseppe, **Utilização de Técnicas de "Data Mining" para o Reconhecimento de Caracteres Manuscritos**, Centro de tecnologia da Universidade Federal da Paraíba, João Pessoa, PA.

[75] CID, Dante J. A., PASSOS, Emannuel, P. L., **Análise de Ferramentas de Dataminig**, Núcleo de Pesquisa em Inteligência Computacional Aplicada, Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro – Brasil.

ANEXO I – Algoritmo APRIORI - I

ALGORITMO APRIORI

```
%INICIALIZAÇÃO
arquivo = 'dados4.txt'
minsup = 0.0; maxsup = 100; minconf = 0.0%
k = 1; Natributos = 50; NL = 0;
%LEITURA DO ARQUIVO
arq = fopen(arquivo,'rt');
i = 1;
linha = str2num(fgetl(arq));
for j = 1:Natributos,
Sup(j) = linha(j);
end;
while ~feof(arq)
i = i + 1;
linha = str2num(fgetl(arq));
for j = 1:Natributos,
Sup(j) = Sup(j) + linha(j); %Cálculo do Suporte Geral
end;
end;
st = fclose(arq);
Npesquisadores = i
fator = 100/Npesquisadores;
Sup = Sup*fator;
%Passo (1)--> L = {1-itemset}
%=====================================
Nlarges = 0;
for col=1:Natributos
if Sup(col) >= minsup & Sup(col) < maxsup,
Nlarges = Nlarges + 1;
L(Nlarges,1) = col;
L(Nlarges,2) = Sup(col);
end; %if
end; %for col
clear Sup
%Passo (2) --> Laço geral para obter L
%=====================================
while Nlarges(k) > 1,
k = k + 1;
%Passo 3 --> Chamar a função apriori_gen para obter
c
%=====================================
=
apriori_gen;
for i = 1:Ncandid
soma = 0;
%LEITURA DO ARQUIVO
arq = fopen(arquivo,'rt');
for j = 1:Npesquisadores
linha = str2num(fgetl(arq));
for w = 1:k
if linha(c(i,w)) == 1
adic = 1;
else
adic = 0;
break;
end; %if
end; %for w
soma = soma + adic;
end; %for j
st = fclose(arq);
c(i,k+1) = soma*fator; %c recebe a coluna de Suporte
end; %for i
%Passo 9 - OBTER L(k)
%====================
j = 0;
clear large
for i = 1:Ncandid
if c(i,k+1)>= minsup & c(i,k+1)< maxsup
j = j + 1;
large(j,:) = c(i,:);
end; %if
end; %for i
Nlarges(k) = j;
if Nlarges(k) > 0
L(1:Nlarges(k),1:k+1,k) = large;
NL = k;
end;
end; %while
%Chamar a função genrules para extrair as regras
genrules
'FIM DO ALGORITMO APRIORI'
```

ANEXO II – Algoritmo APRIORI - II

FUNÇÃO APRIORI_GEN

```
%OBTER OS CANDIDATOS E PODA (join step e
prune step)
Ncandid = 0;
if k==2,
for fixo = 1:Nlarges-1
for varia = (fixo+1):Nlarges
ss = 1;
item(ss) = L(fixo,1);
ss = ss + 1;
item(ss) = L(varia,1);
Ncandid = Ncandid + 1;
c(Ncandid,:) = item;
end; %for varia
end; %for fixo
else %para k>2
for i=1:(k-1)
flag(i) = 1;
end; %for i
for fixo = 1:(Nlarges(k-1)-1)
for ss = 1:(k-1)
item_ant(ss) = L(fixo,ss,k-1);
end; %for ss
if item_ant(ss) < Natributos
for i = 2:(k-1)
test(i-1) = item_ant(i);
end; %for i
for varia = (item_ant(ss)+1):Natributos
test(i) = varia;
pert = 0;
for j = (fixo+1):Nlarges(k-1) %Procura em L
for w=1:(k-1)
if test(w) == L(j,w,k-1)
pert(w) = 1;
else
pert(w) = 0;
end; %if
end; %for w
if pert == flag
break;
end; %if
end; %for j
if pert == flag %item pertence a L
item = item_ant;
item(k) = varia; %novo componente
Ncandid = Ncandid + 1;
c(Ncandid,:) = item;
end; %if
end; %for varia
end; %if
end; %for fixo
end; %else
'FIM DA FUNÇÃO APRIORI_GEN'
```

ANEXO III – Algoritmo APRIORI - III

```
FUNÇÃO GENRULES
%ESTRAÇÃO DE REGRAS A --> B
if NL < 2,
NL
'Não há Regras.'
else
'Extraindo Regras...'
Nregras = 0;
for k = NL:-1:2
for i = 1:Nlarges(k)
itemset = L(i,:,k);
m = k - 1;
while m > 0
a = itemset(1:m);
b = itemset(m+1:k);
tamb = 0;
for w = m+1:k
tamb = tamb + 1;
end;
for j = 1:Nlarges(m)
if L(j,1:m,m) == a
break;
end;
end; %for j
sup_a = L(j,m+1,m);
conf = (itemset(k+1)/sup_a)*100;
if conf >= minconf
Nregras = Nregras + 1;
A(Nregras,1:m) = a; %antecedente da regra
B(Nregras,1:tamb) = b; %consequente da regra
S(Nregras) = itemset(k+1); %suporte da regra
C(Nregras) = conf; %confiança da regra
m = m - 1; %próximo subset
else
break; %não precisa avaliar subsets
end; %if conf
end; %while
end; %for i
end; %for k
clear L
Nregras
if Nregras > 0,
A
B
format bank;
'Suporte = ',S'
'Confiança = ',C'
end; %if Nregras
end; %if NL
'FIM DA FUNÇÃO GENRULES'
```

ANEXO IV – Programa conversão de cabeçalho

```
VERSION 5.00
Begin VB.Form Form1
  Caption       = "Converter Registros Base Sistema
Data Mining"
  ClientHeight  = 2220
  ClientLeft    = 1650
  ClientTop     = 1545
  ClientWidth   = 6450
  LinkTopic     = "Form1"
  ScaleHeight   = 2220
  ScaleWidth    = 6450
  Begin VB.CommandButton botao
    Caption     = "Clique Aqui!!!!"
    Height      = 1005
    Left        = 1800
    TabIndex    = 0
    Top         = 900
    Width       = 2625
  End
  Begin VB.Data Data1
    Caption     = "Data1"
    Connect     = "Access"
    DatabaseName  = "C:\Dina\ports.mdb"
    DefaultCursorType= 0 'DefaultCursor
    DefaultType   = 2 'UseODBC
    Exclusive     = 0 'False
    Height      = 345
    Left        = 4710
    Options     = 0
    ReadOnly    = 0 'False
    RecordsetType  = 1 'Dynaset
    RecordSource  = "ports"
    Top         = 1020
    Visible     = 0 'False
    Width       = 1140
  End
```

```
End
Attribute VB_Name = "Form1"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
Private dados As Database
Private tabela As Recordset
Private letra, semana As String

Private Sub botao_Click()
  tabela.MoveFirst
  Do While Not (tabela.EOF)
    letra = Left(tabela("Campo1"), 1)
    If letra = "M" Then
      semana = Mid(tabela("Campo1"), 62, 3)
    Else
      tabela.Edit
      tabela("Diasemana") = semana
      tabela.Update
    End If
    tabela.MoveNext
  Loop
  MsgBox ("Registros Atualizados!!!")
End Sub

Private Sub Form_Load()
Dim dbname As String
  dbname = "\ports.mdb"
  Set dados =
DBEngine.Workspaces(0).OpenDatabase(App.Path &
dbname)
  Set tabela = dados.OpenRecordset("Ports",
dbOpenTable)
  tabela.Index = "PrimaryKey"
End Sub
```

ANEXO V – Regras Geradas pelo algoritmo APRIORI

=== Run information ===

Scheme: weka.associations.Apriori -N 500 -T 0 -C 0.1 -D 0.01 -U 1.0 -M 0.01 -S -1.0
Relation: dados-weka.filters.AttributeFilter-V-R1-4,6
Instances: 5544
Attributes: 5
sem
hora
slot
port
ruim
=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.01
Minimum metric <confidence>: 0.1
Number of cycles performed: 98

Generated sets of large itemsets:

Size of set of large itemsets L(1): 56

Size of set of large itemsets L(2): 134

Size of set of large itemsets L(3): 47

Best rules found:

1. ruim=0 285 ==> sem=Sun 285 conf:(1)
2. slot=14 168 ==> port=1 ruim=menor 168 conf:(1)
3. slot=14 port=1 168 ==> ruim=menor 168 conf:(1)
4. slot=14 ruim=menor 168 ==> port=1 168 conf:(1)
5. slot=10 168 ==> port=1 ruim=menor 168 conf:(1)
6. slot=10 port=1 168 ==> ruim=menor 168 conf:(1)
7. slot=10 ruim=menor 168 ==> port=1 168 conf:(1)
8. slot=9 168 ==> port=1 ruim=menor 168 conf:(1)
9. slot=9 port=1 168 ==> ruim=menor 168 conf:(1)
10. slot=9 ruim=menor 168 ==> port=1 168 conf:(1)
11. port=7 168 ==> slot=8 ruim=menor 168 conf:(1)
12. slot=8 port=7 168 ==> ruim=menor 168 conf:(1)
13. port=7 ruim=menor 168 ==> slot=8 168 conf:(1)
14. slot=7 port=3 168 ==> ruim=menor 168 conf:(1)
15. slot=7 port=1 168 ==> ruim=menor 168 conf:(1)
16. slot=6 port=3 168 ==> ruim=menor 168 conf:(1)
17. slot=6 port=1 168 ==> ruim=menor 168 conf:(1)
18. slot=5 port=3 168 ==> ruim=menor 168 conf:(1)
19. slot=5 port=2 168 ==> ruim=menor 168 conf:(1)
20. slot=5 port=1 168 ==> ruim=menor 168 conf:(1)
21. slot=4 port=3 168 ==> ruim=maior 168 conf:(1)
22. port=3 ruim=maior 168 ==> slot=4 168 conf:(1)
23. slot=3 168 ==> port=1 ruim=menor 168 conf:(1)
24. slot=3 port=1 168 ==> ruim=menor 168 conf:(1)
25. slot=3 ruim=menor 168 ==> port=1 168 conf:(1)

26. slot=2 168 ==> port=1 ruim=menor 168 conf:(1)
27. slot=2 port=1 168 ==> ruim=menor 168 conf:(1)
28. slot=2 ruim=menor 168 ==> port=1 168 conf:(1)
29. port=7 168 ==> ruim=menor 168 conf:(1)
30. slot=15 168 ==> port=1 168 conf:(1)
31. slot=14 168 ==> ruim=menor 168 conf:(1)
32. slot=14 168 ==> port=1 168 conf:(1)
33. slot=13 168 ==> port=1 168 conf:(1)
34. slot=12 168 ==> port=1 168 conf:(1)
35. slot=11 168 ==> port=1 168 conf:(1)
36. slot=10 168 ==> ruim=menor 168 conf:(1)
37. slot=10 168 ==> port=1 168 conf:(1)
38. slot=9 168 ==> ruim=menor 168 conf:(1)
39. slot=9 168 ==> port=1 168 conf:(1)
40. port=8 168 ==> slot=8 168 conf:(1)
41. port=7 168 ==> slot=8 168 conf:(1)
42. port=6 168 ==> slot=8 168 conf:(1)
43. port=5 168 ==> slot=8 168 conf:(1)
44. slot=3 168 ==> ruim=menor 168 conf:(1)
45. slot=3 168 ==> port=1 168 conf:(1)
46. slot=2 168 ==> ruim=menor 168 conf:(1)
47. slot=2 168 ==> port=1 168 conf:(1)
48. port=5 ruim=menor 167 ==> slot=8 167 conf:(1)
49. slot=13 ruim=menor 158 ==> port=1 158 conf:(1)
50. slot=12 ruim=menor 157 ==> port=1 157 conf:(1)
51. slot=11 ruim=menor 157 ==> port=1 157 conf:(1)
52. slot=15 ruim=menor 156 ==> port=1 156 conf:(1)
53. slot=8 ruim=maior 146 ==> port=1 146 conf:(1)
54. port=8 ruim=menor 145 ==> slot=8 145 conf:(1)
55. port=6 ruim=menor 145 ==> slot=8 145 conf:(1)
56. port=4 ruim=maior 144 ==> slot=4 144 conf:(1)
57. slot=8 ruim=0 117 ==> sem=Sun 117 conf:(1)
58. port=5 168 ==> slot=8 ruim=menor 167 conf:(0.99)
59. slot=8 port=5 168 ==> ruim=menor 167 conf:(0.99)
60. slot=8 port=3 168 ==> ruim=menor 167 conf:(0.99)
61. port=5 168 ==> ruim=menor 167 conf:(0.99)
62. slot=5 672 ==> ruim=menor 663 conf:(0.99)
63. slot=4 port=1 168 ==> ruim=maior 161 conf:(0.96)
64. slot=5 port=4 168 ==> ruim=menor 159 conf:(0.95)
65. slot=13 168 ==> port=1 ruim=menor 158 conf:(0.94)
66. slot=13 port=1 168 ==> ruim=menor 158 conf:(0.94)
67. slot=6 port=4 168 ==> ruim=menor 158 conf:(0.94)
68. slot=13 168 ==> ruim=menor 158 conf:(0.94)
69. slot=12 168 ==> port=1 ruim=menor 157 conf:(0.93)
70. slot=12 port=1 168 ==> ruim=menor 157 conf:(0.93)
71. slot=11 168 ==> port=1 ruim=menor 157 conf:(0.93)
72. slot=11 port=1 168 ==> ruim=menor 157 conf:(0.93)
73. slot=12 168 ==> ruim=menor 157 conf:(0.93)
74. slot=11 168 ==> ruim=menor 157 conf:(0.93)
75. slot=7 672 ==> ruim=menor 624 conf:(0.93)
76. slot=15 168 ==> port=1 ruim=menor 156 conf:(0.93)
77. slot=15 port=1 168 ==> ruim=menor 156 conf:(0.93)
78. slot=15 168 ==> ruim=menor 156 conf:(0.93)
79. slot=4 672 ==> ruim=maior 617 conf:(0.92)
80. slot=6 672 ==> ruim=menor 613 conf:(0.91)
81. sem=Wed slot=8 192 ==> ruim=menor 168 conf:(0.88)
82. sem=Tue slot=8 192 ==> ruim=menor 168 conf:(0.88)
83. sem=Thu slot=8 192 ==> ruim=menor 168 conf:(0.88)

84. sem=Sat slot=8 192 ==> ruim=menor 168 conf:(0.88)
85. sem=Mon slot=8 192 ==> ruim=menor 168 conf:(0.88)
86. sem=Fri slot=8 192 ==> ruim=menor 168 conf:(0.88)
87. slot=8 port=1 168 ==> ruim=maior 146 conf:(0.87)
88. port=8 168 ==> slot=8 ruim=menor 145 conf:(0.86)
89. slot=8 port=8 168 ==> ruim=menor 145 conf:(0.86)
90. port=6 168 ==> slot=8 ruim=menor 145 conf:(0.86)
91. slot=8 port=6 168 ==> ruim=menor 145 conf:(0.86)
92. slot=8 port=2 168 ==> ruim=menor 145 conf:(0.86)
93. port=8 168 ==> ruim=menor 145 conf:(0.86)
94. port=6 168 ==> ruim=menor 145 conf:(0.86)
95. sem=Wed port=1 336 ==> ruim=menor 288 conf:(0.86)
96. sem=Tue port=1 336 ==> ruim=menor 288 conf:(0.86)
97. sem=Thu port=1 336 ==> ruim=menor 288 conf:(0.86)
98. sem=Sat port=1 336 ==> ruim=menor 288 conf:(0.86)
99. sem=Mon port=1 336 ==> ruim=menor 288 conf:(0.86)
100. sem=Fri port=1 336 ==> ruim=menor 288 conf:(0.86)
101. slot=8 port=4 168 ==> ruim=menor 144 conf:(0.86)
102. slot=7 port=4 168 ==> ruim=menor 144 conf:(0.86)
103. slot=7 port=2 168 ==> ruim=menor 144 conf:(0.86)
104. slot=4 port=4 168 ==> ruim=maior 144 conf:(0.86)
105. slot=4 port=2 168 ==> ruim=maior 144 conf:(0.86)
106. sem=Sat 792 ==> ruim=menor 666 conf:(0.84)
107. sem=Wed 792 ==> ruim=menor 665 conf:(0.84)
108. sem=Tue 792 ==> ruim=menor 665 conf:(0.84)
109. sem=Thu 792 ==> ruim=menor 665 conf:(0.84)
110. sem=Mon 792 ==> ruim=menor 665 conf:(0.84)
111. sem=Fri 792 ==> ruim=menor 665 conf:(0.84)
112. port=1 2352 ==> ruim=menor 1972 conf:(0.84)
113. hora=17 231 ==> ruim=menor 191 conf:(0.83)
114. hora=15 231 ==> ruim=menor 191 conf:(0.83)
115. hora=9 231 ==> ruim=menor 191 conf:(0.83)
116. hora=12 232 ==> ruim=menor 191 conf:(0.82)
117. hora=16 231 ==> ruim=menor 190 conf:(0.82)
118. hora=14 231 ==> ruim=menor 190 conf:(0.82)
119. hora=13 231 ==> ruim=menor 190 conf:(0.82)
120. hora=11 231 ==> ruim=menor 190 conf:(0.82)
121. hora=10 231 ==> ruim=menor 190 conf:(0.82)
122. hora=19 231 ==> ruim=menor 189 conf:(0.82)
123. hora=18 231 ==> ruim=menor 189 conf:(0.82)
124. hora=20 231 ==> ruim=menor 186 conf:(0.81)
125. slot=8 1344 ==> ruim=menor 1081 conf:(0.8)
126. port=3 840 ==> ruim=menor 671 conf:(0.8)
127. hora=23 231 ==> ruim=menor 184 conf:(0.8)
128. hora=22 231 ==> ruim=menor 184 conf:(0.8)
129. hora=21 231 ==> ruim=menor 184 conf:(0.8)
130. hora=0 228 ==> ruim=menor 181 conf:(0.79)
131. hora=8 232 ==> ruim=menor 184 conf:(0.79)
132. hora=7 231 ==> ruim=menor 183 conf:(0.79)
133. hora=2 231 ==> ruim=menor 183 conf:(0.79)
134. hora=6 231 ==> ruim=menor 179 conf:(0.77)
135. hora=4 231 ==> ruim=menor 178 conf:(0.77)
136. hora=3 231 ==> ruim=menor 178 conf:(0.77)
137. hora=1 232 ==> ruim=menor 177 conf:(0.76)
138. hora=5 231 ==> ruim=menor 176 conf:(0.76)
139. ruim=maior 810 ==> slot=4 617 conf:(0.76)
140. port=2 ruim=maior 191 ==> slot=4 144 conf:(0.75)
141. sem=Sun port=1 336 ==> ruim=menor 244 conf:(0.73)

142. port=4 840 ==> ruim=menor 605 conf:(0.72)
143. slot=6 port=2 168 ==> ruim=menor 119 conf:(0.71)
144. port=2 840 ==> ruim=menor 576 conf:(0.69)
145. sem=Sun slot=8 192 ==> ruim=0 117 conf:(0.61)
146. sem=Sun 792 ==> ruim=menor 458 conf:(0.58)
147. sem=Sun ruim=menor 458 ==> port=1 244 conf:(0.53)
148. port=1 ruim=maior 307 ==> slot=4 161 conf:(0.52)
149. port=1 ruim=maior 307 ==> slot=8 146 conf:(0.48)
150. ruim=menor 4449 ==> port=1 1972 conf:(0.44)
151. sem=Wed ruim=menor 665 ==> port=1 288 conf:(0.43)
152. sem=Tue ruim=menor 665 ==> port=1 288 conf:(0.43)
153. sem=Thu ruim=menor 665 ==> port=1 288 conf:(0.43)
154. sem=Mon ruim=menor 665 ==> port=1 288 conf:(0.43)
155. sem=Fri ruim=menor 665 ==> port=1 288 conf:(0.43)
156. sem=Sat ruim=menor 666 ==> port=1 288 conf:(0.43)
157. sem=Wed 792 ==> port=1 336 conf:(0.42)
158. sem=Tue 792 ==> port=1 336 conf:(0.42)
159. sem=Thu 792 ==> port=1 336 conf:(0.42)
160. sem=Sun 792 ==> port=1 336 conf:(0.42)
161. sem=Sat 792 ==> port=1 336 conf:(0.42)
162. sem=Mon 792 ==> port=1 336 conf:(0.42)
163. sem=Fri 792 ==> port=1 336 conf:(0.42)
164. ruim=0 285 ==> sem=Sun slot=8 117 conf:(0.41)
165. sem=Sun ruim=0 285 ==> slot=8 117 conf:(0.41)
166. ruim=0 285 ==> slot=8 117 conf:(0.41)
167. ruim=maior 810 ==> port=1 307 conf:(0.38)
168. sem=Wed 792 ==> port=1 ruim=menor 288 conf:(0.36)
169. sem=Tue 792 ==> port=1 ruim=menor 288 conf:(0.36)
170. sem=Thu 792 ==> port=1 ruim=menor 288 conf:(0.36)
171. sem=Sat 792 ==> port=1 ruim=menor 288 conf:(0.36)
172. sem=Mon 792 ==> port=1 ruim=menor 288 conf:(0.36)
173. sem=Fri 792 ==> port=1 ruim=menor 288 conf:(0.36)
174. sem=Sun 792 ==> ruim=0 285 conf:(0.36)
175. sem=Sun 792 ==> port=1 ruim=menor 244 conf:(0.31)
176. port=2 ruim=menor 576 ==> slot=5 168 conf:(0.29)
177. slot=6 ruim=menor 613 ==> port=3 168 conf:(0.27)
178. slot=6 ruim=menor 613 ==> port=1 168 conf:(0.27)
179. slot=4 ruim=maior 617 ==> port=3 168 conf:(0.27)
180. slot=7 ruim=menor 624 ==> port=3 168 conf:(0.27)
181. slot=7 ruim=menor 624 ==> port=1 168 conf:(0.27)
182. port=4 ruim=menor 605 ==> slot=5 159 conf:(0.26)
183. port=4 ruim=menor 605 ==> slot=6 158 conf:(0.26)
184. slot=4 ruim=maior 617 ==> port=1 161 conf:(0.26)
185. slot=6 ruim=menor 613 ==> port=4 158 conf:(0.26)
186. slot=5 ruim=menor 663 ==> port=3 168 conf:(0.25)
187. slot=5 ruim=menor 663 ==> port=2 168 conf:(0.25)
188. slot=5 ruim=menor 663 ==> port=1 168 conf:(0.25)
189. sem=Wed ruim=menor 665 ==> slot=8 168 conf:(0.25)
190. sem=Tue ruim=menor 665 ==> slot=8 168 conf:(0.25)
191. sem=Thu ruim=menor 665 ==> slot=8 168 conf:(0.25)
192. sem=Mon ruim=menor 665 ==> slot=8 168 conf:(0.25)
193. sem=Fri ruim=menor 665 ==> slot=8 168 conf:(0.25)
194. sem=Sat ruim=menor 666 ==> slot=8 168 conf:(0.25)
195. port=2 ruim=menor 576 ==> slot=8 145 conf:(0.25)
196. port=3 ruim=menor 671 ==> slot=7 168 conf:(0.25)
197. port=3 ruim=menor 671 ==> slot=6 168 conf:(0.25)
198. port=3 ruim=menor 671 ==> slot=5 168 conf:(0.25)
199. slot=7 672 ==> port=3 ruim=menor 168 conf:(0.25)

200. slot=7 672 ==> port=1 ruim=menor 168 conf:(0.25)
201. slot=6 672 ==> port=3 ruim=menor 168 conf:(0.25)
202. slot=6 672 ==> port=1 ruim=menor 168 conf:(0.25)
203. slot=5 672 ==> port=3 ruim=menor 168 conf:(0.25)
204. slot=5 672 ==> port=2 ruim=menor 168 conf:(0.25)
205. slot=5 672 ==> port=1 ruim=menor 168 conf:(0.25)
206. slot=4 672 ==> port=3 ruim=maior 168 conf:(0.25)
207. slot=7 672 ==> port=4 168 conf:(0.25)
208. slot=7 672 ==> port=3 168 conf:(0.25)
209. slot=7 672 ==> port=2 168 conf:(0.25)
210. slot=7 672 ==> port=1 168 conf:(0.25)
211. slot=6 672 ==> port=4 168 conf:(0.25)
212. slot=6 672 ==> port=3 168 conf:(0.25)
213. slot=6 672 ==> port=2 168 conf:(0.25)
214. slot=6 672 ==> port=1 168 conf:(0.25)
215. slot=5 672 ==> port=4 168 conf:(0.25)
216. slot=5 672 ==> port=3 168 conf:(0.25)
217. slot=5 672 ==> port=2 168 conf:(0.25)
218. slot=5 672 ==> port=1 168 conf:(0.25)
219. slot=4 672 ==> port=4 168 conf:(0.25)
220. slot=4 672 ==> port=3 168 conf:(0.25)
221. slot=4 672 ==> port=2 168 conf:(0.25)
222. slot=4 672 ==> port=1 168 conf:(0.25)
223. port=2 ruim=menor 576 ==> slot=7 144 conf:(0.25)
224. port=3 ruim=menor 671 ==> slot=8 167 conf:(0.25)
225. ruim=menor 4449 ==> slot=8 1081 conf:(0.24)
226. sem=Wed 792 ==> slot=8 192 conf:(0.24)
227. sem=Tue 792 ==> slot=8 192 conf:(0.24)
228. sem=Thu 792 ==> slot=8 192 conf:(0.24)
229. sem=Sun 792 ==> slot=8 192 conf:(0.24)
230. sem=Sat 792 ==> slot=8 192 conf:(0.24)
231. sem=Mon 792 ==> slot=8 192 conf:(0.24)
232. sem=Fri 792 ==> slot=8 192 conf:(0.24)
233. slot=5 ruim=menor 663 ==> port=4 159 conf:(0.24)
234. slot=4 672 ==> port=1 ruim=maior 161 conf:(0.24)
235. port=4 ruim=menor 605 ==> slot=8 144 conf:(0.24)
236. port=4 ruim=menor 605 ==> slot=7 144 conf:(0.24)
237. slot=5 672 ==> port=4 ruim=menor 159 conf:(0.24)
238. ruim=maior 810 ==> port=2 191 conf:(0.24)
239. slot=6 672 ==> port=4 ruim=menor 158 conf:(0.24)
240. slot=4 ruim=maior 617 ==> port=4 144 conf:(0.23)
241. slot=4 ruim=maior 617 ==> port=2 144 conf:(0.23)
242. slot=7 ruim=menor 624 ==> port=4 144 conf:(0.23)
243. slot=7 ruim=menor 624 ==> port=2 144 conf:(0.23)
244. port=2 840 ==> ruim=maior 191 conf:(0.23)
245. slot=7 672 ==> port=4 ruim=menor 144 conf:(0.21)
246. slot=7 672 ==> port=2 ruim=menor 144 conf:(0.21)
247. slot=4 672 ==> port=4 ruim=maior 144 conf:(0.21)
248. slot=4 672 ==> port=2 ruim=maior 144 conf:(0.21)
249. sem=Wed 792 ==> slot=8 ruim=menor 168 conf:(0.21)
250. sem=Tue 792 ==> slot=8 ruim=menor 168 conf:(0.21)
251. sem=Thu 792 ==> slot=8 ruim=menor 168 conf:(0.21)
252. sem=Sat 792 ==> slot=8 ruim=menor 168 conf:(0.21)
253. sem=Mon 792 ==> slot=8 ruim=menor 168 conf:(0.21)
254. sem=Fri 792 ==> slot=8 ruim=menor 168 conf:(0.21)
255. ruim=maior 810 ==> slot=4 port=3 168 conf:(0.21)
256. ruim=maior 810 ==> port=3 168 conf:(0.21)
257. port=2 ruim=menor 576 ==> slot=6 119 conf:(0.21)

258. port=3 840 ==> slot=7 ruim=menor 168 conf:(0.2)
259. port=3 840 ==> slot=6 ruim=menor 168 conf:(0.2)
260. port=3 840 ==> slot=5 ruim=menor 168 conf:(0.2)
261. port=2 840 ==> slot=5 ruim=menor 168 conf:(0.2)
262. port=3 840 ==> slot=4 ruim=maior 168 conf:(0.2)
263. port=3 840 ==> ruim=maior 168 conf:(0.2)
264. port=4 840 ==> slot=8 168 conf:(0.2)
265. port=3 840 ==> slot=8 168 conf:(0.2)
266. port=2 840 ==> slot=8 168 conf:(0.2)
267. port=4 840 ==> slot=7 168 conf:(0.2)
268. port=3 840 ==> slot=7 168 conf:(0.2)
269. port=2 840 ==> slot=7 168 conf:(0.2)
270. port=4 840 ==> slot=6 168 conf:(0.2)
271. port=3 840 ==> slot=6 168 conf:(0.2)
272. port=2 840 ==> slot=6 168 conf:(0.2)
273. port=4 840 ==> slot=5 168 conf:(0.2)
274. port=3 840 ==> slot=5 168 conf:(0.2)
275. port=2 840 ==> slot=5 168 conf:(0.2)
276. port=4 840 ==> slot=4 168 conf:(0.2)
277. port=3 840 ==> slot=4 168 conf:(0.2)
278. port=2 840 ==> slot=4 168 conf:(0.2)
279. port=3 840 ==> slot=8 ruim=menor 167 conf:(0.2)
280. ruim=maior 810 ==> slot=4 port=1 161 conf:(0.2)
281. slot=6 ruim=menor 613 ==> port=2 119 conf:(0.19)
282. port=4 840 ==> slot=5 ruim=menor 159 conf:(0.19)
283. port=4 840 ==> slot=6 ruim=menor 158 conf:(0.19)
284. ruim=maior 810 ==> slot=8 port=1 146 conf:(0.18)
285. ruim=maior 810 ==> slot=8 146 conf:(0.18)
286. ruim=maior 810 ==> slot=4 port=4 144 conf:(0.18)
287. ruim=maior 810 ==> slot=4 port=2 144 conf:(0.18)
288. ruim=maior 810 ==> port=4 144 conf:(0.18)
289. slot=6 672 ==> port=2 ruim=menor 119 conf:(0.18)
290. port=2 840 ==> slot=8 ruim=menor 145 conf:(0.17)
291. port=4 840 ==> slot=8 ruim=menor 144 conf:(0.17)
292. port=4 840 ==> slot=7 ruim=menor 144 conf:(0.17)
293. port=2 840 ==> slot=7 ruim=menor 144 conf:(0.17)
294. port=4 840 ==> slot=4 ruim=maior 144 conf:(0.17)
295. port=2 840 ==> slot=4 ruim=maior 144 conf:(0.17)
296. port=4 840 ==> ruim=maior 144 conf:(0.17)
297. sem=Wed 792 ==> ruim=maior 127 conf:(0.16)
298. sem=Tue 792 ==> ruim=maior 127 conf:(0.16)
299. sem=Thu 792 ==> ruim=maior 127 conf:(0.16)
300. sem=Mon 792 ==> ruim=maior 127 conf:(0.16)
301. sem=Fri 792 ==> ruim=maior 127 conf:(0.16)
302. sem=Sat 792 ==> ruim=maior 126 conf:(0.16)
303. ruim=maior 810 ==> sem=Wed 127 conf:(0.16)
304. ruim=maior 810 ==> sem=Tue 127 conf:(0.16)
305. ruim=maior 810 ==> sem=Thu 127 conf:(0.16)
306. ruim=maior 810 ==> sem=Mon 127 conf:(0.16)
307. ruim=maior 810 ==> sem=Fri 127 conf:(0.16)
308. ruim=maior 810 ==> sem=Sat 126 conf:(0.16)
309. slot=8 ruim=menor 1081 ==> port=7 168 conf:(0.16)
310. slot=8 ruim=menor 1081 ==> sem=Wed 168 conf:(0.16)
311. slot=8 ruim=menor 1081 ==> sem=Tue 168 conf:(0.16)
312. slot=8 ruim=menor 1081 ==> sem=Thu 168 conf:(0.16)
313. slot=8 ruim=menor 1081 ==> sem=Sat 168 conf:(0.16)
314. slot=8 ruim=menor 1081 ==> sem=Mon 168 conf:(0.16)
315. slot=8 ruim=menor 1081 ==> sem=Fri 168 conf:(0.16)

316. slot=8 ruim=menor 1081 ==> port=5 167 conf:(0.15)
317. slot=8 ruim=menor 1081 ==> port=3 167 conf:(0.15)
318. sem=Wed 792 ==> port=4 120 conf:(0.15)
319. sem=Wed 792 ==> port=3 120 conf:(0.15)
320. sem=Wed 792 ==> port=2 120 conf:(0.15)
321. sem=Tue 792 ==> port=4 120 conf:(0.15)
322. sem=Tue 792 ==> port=3 120 conf:(0.15)
323. sem=Tue 792 ==> port=2 120 conf:(0.15)
324. sem=Thu 792 ==> port=4 120 conf:(0.15)
325. sem=Thu 792 ==> port=3 120 conf:(0.15)
326. sem=Thu 792 ==> port=2 120 conf:(0.15)
327. sem=Sun 792 ==> port=4 120 conf:(0.15)
328. sem=Sun 792 ==> port=3 120 conf:(0.15)
329. sem=Sun 792 ==> port=2 120 conf:(0.15)
330. sem=Sat 792 ==> port=4 120 conf:(0.15)
331. sem=Sat 792 ==> port=3 120 conf:(0.15)
332. sem=Sat 792 ==> port=2 120 conf:(0.15)
333. sem=Mon 792 ==> port=4 120 conf:(0.15)
334. sem=Mon 792 ==> port=3 120 conf:(0.15)
335. sem=Mon 792 ==> port=2 120 conf:(0.15)
336. sem=Fri 792 ==> port=4 120 conf:(0.15)
337. sem=Fri 792 ==> port=3 120 conf:(0.15)
338. sem=Fri 792 ==> port=2 120 conf:(0.15)
339. ruim=menor 4449 ==> port=3 671 conf:(0.15)
340. ruim=menor 4449 ==> sem=Sat 666 conf:(0.15)
341. ruim=menor 4449 ==> sem=Wed 665 conf:(0.15)
342. ruim=menor 4449 ==> sem=Tue 665 conf:(0.15)
343. ruim=menor 4449 ==> sem=Thu 665 conf:(0.15)
344. ruim=menor 4449 ==> sem=Mon 665 conf:(0.15)
345. ruim=menor 4449 ==> sem=Fri 665 conf:(0.15)
346. ruim=menor 4449 ==> slot=5 663 conf:(0.15)
347. sem=Sun 792 ==> slot=8 ruim=0 117 conf:(0.15)
348. port=1 ruim=menor 1972 ==> sem=Wed 288 conf:(0.15)
349. port=1 ruim=menor 1972 ==> sem=Tue 288 conf:(0.15)
350. port=1 ruim=menor 1972 ==> sem=Thu 288 conf:(0.15)
351. port=1 ruim=menor 1972 ==> sem=Sat 288 conf:(0.15)
352. port=1 ruim=menor 1972 ==> sem=Mon 288 conf:(0.15)
353. port=1 ruim=menor 1972 ==> sem=Fri 288 conf:(0.15)
354. port=1 2352 ==> sem=Wed 336 conf:(0.14)
355. port=1 2352 ==> sem=Tue 336 conf:(0.14)
356. port=1 2352 ==> sem=Thu 336 conf:(0.14)
357. port=1 2352 ==> sem=Sun 336 conf:(0.14)
358. port=1 2352 ==> sem=Sat 336 conf:(0.14)
359. port=1 2352 ==> sem=Mon 336 conf:(0.14)
360. port=1 2352 ==> sem=Fri 336 conf:(0.14)
361. slot=8 1344 ==> sem=Wed 192 conf:(0.14)
362. slot=8 1344 ==> sem=Tue 192 conf:(0.14)
363. slot=8 1344 ==> sem=Thu 192 conf:(0.14)
364. slot=8 1344 ==> sem=Sun 192 conf:(0.14)
365. slot=8 1344 ==> sem=Sat 192 conf:(0.14)
366. slot=8 1344 ==> sem=Mon 192 conf:(0.14)
367. slot=8 1344 ==> sem=Fri 192 conf:(0.14)
368. port=4 840 ==> sem=Wed 120 conf:(0.14)
369. port=3 840 ==> sem=Wed 120 conf:(0.14)
370. port=2 840 ==> sem=Wed 120 conf:(0.14)
371. port=4 840 ==> sem=Tue 120 conf:(0.14)
372. port=3 840 ==> sem=Tue 120 conf:(0.14)
373. port=2 840 ==> sem=Tue 120 conf:(0.14)

374. port=4 840 ==> sem=Thu 120 conf:(0.14)
375. port=3 840 ==> sem=Thu 120 conf:(0.14)
376. port=2 840 ==> sem=Thu 120 conf:(0.14)
377. port=4 840 ==> sem=Sun 120 conf:(0.14)
378. port=3 840 ==> sem=Sun 120 conf:(0.14)
379. port=2 840 ==> sem=Sun 120 conf:(0.14)
380. port=4 840 ==> sem=Sat 120 conf:(0.14)
381. port=3 840 ==> sem=Sat 120 conf:(0.14)
382. port=2 840 ==> sem=Sat 120 conf:(0.14)
383. port=4 840 ==> sem=Mon 120 conf:(0.14)
384. port=3 840 ==> sem=Mon 120 conf:(0.14)
385. port=2 840 ==> sem=Mon 120 conf:(0.14)
386. port=4 840 ==> sem=Fri 120 conf:(0.14)
387. port=3 840 ==> sem=Fri 120 conf:(0.14)
388. port=2 840 ==> sem=Fri 120 conf:(0.14)
389. port=2 840 ==> slot=6 ruim=menor 119 conf:(0.14)
390. ruim=menor 4449 ==> slot=7 624 conf:(0.14)
391. ruim=menor 4449 ==> slot=6 613 conf:(0.14)
392. ruim=menor 4449 ==> port=4 605 conf:(0.14)
393. slot=8 ruim=menor 1081 ==> port=8 145 conf:(0.13)
394. slot=8 ruim=menor 1081 ==> port=6 145 conf:(0.13)
395. slot=8 ruim=menor 1081 ==> port=2 145 conf:(0.13)
396. slot=8 ruim=menor 1081 ==> port=4 144 conf:(0.13)
397. port=1 2352 ==> ruim=maior 307 conf:(0.13)
398. ruim=menor 4449 ==> port=2 576 conf:(0.13)
399. slot=8 1344 ==> port=7 ruim=menor 168 conf:(0.13)
400. slot=8 1344 ==> sem=Wed ruim=menor 168 conf:(0.13)
401. slot=8 1344 ==> sem=Tue ruim=menor 168 conf:(0.13)
402. slot=8 1344 ==> sem=Thu ruim=menor 168 conf:(0.13)
403. slot=8 1344 ==> sem=Sat ruim=menor 168 conf:(0.13)
404. slot=8 1344 ==> sem=Mon ruim=menor 168 conf:(0.13)
405. slot=8 1344 ==> sem=Fri ruim=menor 168 conf:(0.13)
406. slot=8 1344 ==> port=8 168 conf:(0.13)
407. slot=8 1344 ==> port=7 168 conf:(0.13)
408. slot=8 1344 ==> port=6 168 conf:(0.13)
409. slot=8 1344 ==> port=5 168 conf:(0.13)
410. slot=8 1344 ==> port=4 168 conf:(0.13)
411. slot=8 1344 ==> port=3 168 conf:(0.13)
412. slot=8 1344 ==> port=2 168 conf:(0.13)
413. slot=8 1344 ==> port=1 168 conf:(0.13)
414. slot=8 1344 ==> port=5 ruim=menor 167 conf:(0.12)
415. slot=8 1344 ==> port=3 ruim=menor 167 conf:(0.12)
416. port=1 ruim=menor 1972 ==> sem=Sun 244 conf:(0.12)
417. port=1 2352 ==> sem=Wed ruim=menor 288 conf:(0.12)
418. port=1 2352 ==> sem=Tue ruim=menor 288 conf:(0.12)
419. port=1 2352 ==> sem=Thu ruim=menor 288 conf:(0.12)
420. port=1 2352 ==> sem=Sat ruim=menor 288 conf:(0.12)
421. port=1 2352 ==> sem=Mon ruim=menor 288 conf:(0.12)
422. port=1 2352 ==> sem=Fri ruim=menor 288 conf:(0.12)
423. slot=8 1344 ==> port=1 ruim=maior 146 conf:(0.11)
424. slot=8 1344 ==> ruim=maior 146 conf:(0.11)
425. slot=8 1344 ==> port=8 ruim=menor 145 conf:(0.11)
426. slot=8 1344 ==> port=6 ruim=menor 145 conf:(0.11)
427. slot=8 1344 ==> port=2 ruim=menor 145 conf:(0.11)
428. slot=8 1344 ==> port=4 ruim=menor 144 conf:(0.11)
429. port=1 2352 ==> sem=Sun ruim=menor 244 conf:(0.1)
430. ruim=menor 4449 ==> sem=Sun 458 conf:(0.1)