

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

Maria Laura Schneider

**GERÊNCIA DE SEGURANÇA PARA DETECÇÃO DE
INTRUSÃO EM TELECOMUNICAÇÕES ATRAVÉS DE
REDES NEURAIIS**

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

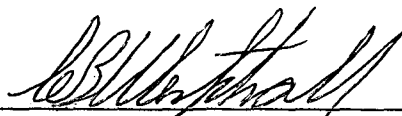
Prof. Dr. Carlos Becker Westphall

Florianópolis, dezembro de 1999.

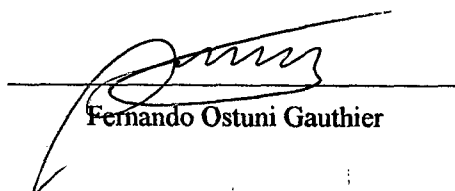
**GERÊNCIA DE SEGURANÇA PARA DETECÇÃO DE INTRUSÃO
EM TELECOMUNICAÇÕES ATRAVÉS DE REDES NEURAIIS**

Maria Laura Schneider

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.

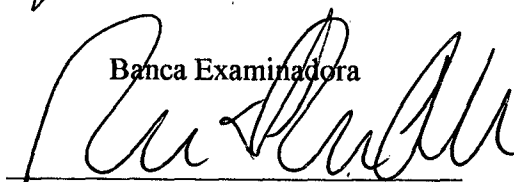


Carlos Becker Westphall

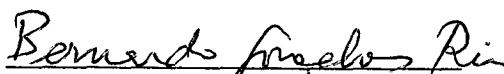


Fernando Ostuni Gauthier

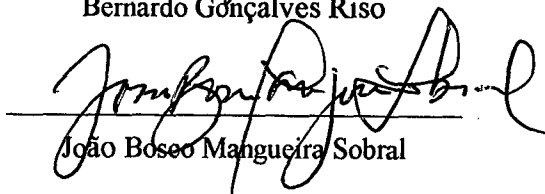
Banca Examinadora



Mirela Sechi Moretti Annoni Notare



Bernardo Gonçalves Riso



João Boseo Manguiera Sobral

Agradecimentos

A realização desta pesquisa foi possível devido a ajuda de algumas pessoas. Estas pessoas tem minha sincera gratidão. Em primeiro lugar, gostaria de dizer o óbvio, isto é, agradeço do fundo do meu coração a minha mãe e meu pai (*in memorium*), que foram os principais responsáveis pela minha felicidade em fazer este mestrado.

Obrigada ao Professor e orientador Carlos Westphall pelo apoio material e psicológico e pela sua grande paciência. Ao Prof. Fernando Cruz minha enorme gratidão pelas tardes inteiras passadas na frente do computador me ajudando a decifrar os emaranhados e complicados significados dos códigos das redes neurais. Não poderia esquecer de agradecer o Prof. Bernardo Riso, que foi o primeiro a acreditar que eu poderia desenvolver uma pesquisa envolvendo assuntos tão novos e promissores e me incentivou neste sentido. A Mirela sou grata por sua alegria de viver, seu bom humor, sua loucura básica ☺, sua sinceridade, suas perguntas diretas e inteligentes, sua sensibilidade, suas dicas, pelas animadas tardes no LRG, mesmo em um ensolarado sábado a tarde, etc, etc e etc... Obrigada também ao meu guru, amigo e conselheiro Julio, que me ensinou muitas coisas preciosas, aos Netfluuuuffiess Tahts, Alex e Ricky, que me proporcionaram muitas lágrimas (de tanto dar risadas ☺) e alegrias. Ao Ricky e ao meu companheiro de sala de estudo (baia) Helio, um obrigada especial de coração, não irei me esquecer do que fizeram por mim!! E como não poderia deixar de ser, para as maravilhosas e necessárias Verinha (Verys) e Valdete um grande, carinhoso e apertado abraço.

Sumário

RESUMO.....	6
ABSTRACT.....	7
LISTA DE SIGLAS.....	8
LISTA DE FIGURAS	9
LISTA DE TABELAS.....	11
1. INTRODUÇÃO.....	12
2. REDES MÓVEIS	15
INTRODUÇÃO.....	15
2.1 TELEFONIA MÓVEL.....	15
2.2 ESTRUTURA.....	16
2.3 SISTEMAS DIGITAL E ANALÓGICO.....	16
2.4 INTRUSÕES NA REDE TELEFÔNICA MÓVEL.....	17
2.4.1 <i>Clonagem</i>	17
2.4.2 <i>Inadimplência</i>	18
CONCLUSÃO.....	18
3. REDES NEURAIS.....	20
INTRODUÇÃO.....	20
3.1 REDES NEURAIS ARTIFICIAIS.....	21
3.1.1 <i>Tipos de Treinamento</i>	24
3.1.1.1 Supervisionado.....	24
3.1.1.2 Não Supervisionado.....	25
3.1.2 <i>Caracterização</i>	25
3.1.3 <i>Agrupamento</i>	26
3.1.3.1 Medidas de Similaridade.....	26
3.1.3.1.1 Distâncias para um par de itens.....	27
3.1.3.1.2 Similaridades e medidas de associação para um par de variáveis... 27	
3.1.3.2 Faces de Chernoff.....	29
3.1.4 <i>Regra de Treinamento de Kohonen</i>	32
3.1.4.1 Mecanismo de Funcionamento.....	33
3.1.4.1.1 Neurônio.....	33
3.1.4.2 Arquitetura.....	34
3.1.4.3 Algoritmo.....	35
3.1.4.3.1 Regra de Aprendizado.....	36
3.1.4.3.2 Preservação da Vizinhança.....	37
3.1.4.4 Testes / Validação.....	37
3.1.5 <i>Aplicações</i>	38
CONCLUSÃO.....	38

4. GERENCIA DE SEGURANÇA PARA A DETECÇÃO DE INTRUSÃO EM TELECOMUNICAÇÕES ATRAVÉS DE REDES NEURAI (DEITREN).	40
INTRODUÇÃO.....	40
4.1 DEITREN.....	41
4.1.1 <i>Amostra de Dados</i>	43
4.1.1.1 Dados Brutos	44
4.1.2 <i>Análise dos Dados</i>	47
4.1.3 <i>Pré Processamento das Variáveis de Entrada</i>	48
4.1.4 <i>Amostra Representativa</i>	54
4.1.5 <i>Kohonen</i>	57
4.1.5.1 Treinamento da Rede - Primeira fase.....	57
4.1.5.1.1 Tecnologia utilizada.....	60
4.1.5.1.2 Treinamento da RN com o algoritmo de Kohonen.....	60
4.1.5.1.3 Análise dos grupos gerados.....	62
4.1.5.2 Definição dos tipos de usuário - Segunda Fase.....	65
4.1.5.3 Controle/Testes - Terceira Fase.....	66
4.1.5.3.1 Falsos positivos e falsos negativos	67
4.1.6 <i>Base de Dados</i>	68
4.1.6.1 Comparação.....	68
4.1.7 <i>Integração dos módulos DEITREN e SSCC do SSTCC</i>	69
4.1.7.1 O funcionamento do Agente.....	70
CONCLUSÃO	71
5 CONCLUSÕES FINAIS E PERSPECTIVAS FUTURAS	72
6. ANEXOS.....	76
7 BIBLIOGRAFIA.....	85

Resumo

Com o rápido crescimento da telefonia móvel, torna-se necessário uma gerência para garantir a segurança desta. A clonagem (cópia ilegal) de telefones celulares e inadimplência (compra de um aparelho celular com intenção premeditada de não pagar a conta) são exemplos de fraudes no sistema telefônico móvel. O prejuízo das companhias telefônicas é de aproximadamente meio milhão de dólares por dia¹. Novas tecnologias de informática são freqüentemente um recurso de obtenção de vantagem competitiva por parte de empresas que utilizaram os benefícios destas antes da concorrência.

O presente trabalho propõe a criação de um sistema que permite monitorar a utilização de telefones celulares para detectar automaticamente fraudes (clonagem, inadimplência). O sistema é baseado no reconhecimento de padrões, através de um processo de agrupamento usando redes neurais. O sistema proposto é reativo, ou seja, não prevê nem impede fraudes, mas alerta a companhia sobre possíveis fraudes, minimizando os prejuízos das empresas de telecomunicações.

Palavras-chave: Telecomunicações, Gerência, Segurança, Telefones Celulares, Fraude, Agrupamento, Redes Neurais.

¹ Dados do IDC – *International Data Corporation*, Setembro, 1999

Abstract

With the fast growth of the mobile telephony, it becomes necessary a management to guarantee the safety of these networks. The cloning mobile phones (illegal copy) problems and impostors/bad-payers (purchase a cellular device with deliberate intention to not pay the account) are examples of frauds in the mobile telecom system. The damage of the telecom companies is of approximately half million dollars per day². New technologies of computer science are frequently the origin of attainment of competitive advantage on the part of companies whom had used the benefices of these before the competition.

The present work proposes the creation of a system that allows to evaluate the use of mobile telephones to detect frauds automatically. The system is based on the pattern recognition, through a grouping process using neural networks. The proposed system is reactive, or either, does not foresee nor it impedes the frauds, but it alerts the company about possible frauds, minimizing the damages of the telecommunications companies.

Keywords: Telecommunications, Management, Security, Mobile Phones, Fraud, Grouping, Neural Networks.

² Data from IDC – *International Data Corporation*, September, 1999.

Lista de Siglas

CCC - Centrais de Comutação e Controle

CORBA - Common Object Request Broker Architecture

DEITREN - Sistema para Detecção de Intrusão em Telecomunicações Através de Redes Neurais

ERB - Estação Rádio Base

ESN - Número Eletrônico de Série

LRG - Laboratório de Redes e Gerência

RN - Redes Neurais

SSCC - Sistema de Segurança contra Clonagem e Inadimplência

SSTCC - Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência

Lista de Figuras

Figura 1 - Esquema do neurônio biológico.....	20
Figura 2 - Esquema de uma Rede Neural Artificial.....	23
Figura 3 - Faces de Chernoff.....	31
Figura 4 - Arquitetura das redes competitivas.....	34
Figura 5 - Esquema genérico do sistema DEITREN: 1 ^a fase.....	41
Figura 6 - Esquema genérico do sistema DEITREN: 2 ^a fase.....	42
Figura 7 - Esquema genérico do sistema DEITREN: 3 ^a fase	43
Figura 8 - Formato do arquivo.....	45
Figura 9 - Descrição dos campos do arquivo da amostra..	46
Figura 10 - Esquema do pré processamento da variável Destino	50
Figura 11 - Variável Data	50
Figura 12 - Variável Hora	51
Figura 13 - Variável Duração	52
Figura 14 - Variável Hora graficamente representada...	53
Figura 15 - Freqüência de Duração na amostra.....	54
Figura 16 - Freqüência de Destino na amostra.....	55
Figura 17 - Freqüência do Dia na amostra.....	56
Figura 18 - Vetores de entrada da RN graficamente Representados	58
Figura 19 - Esquema do funcionamento da rede neural de Kohonen	59
Figura 20 - Estrutura da base de dados	68

Figura 21 - Modelo do SSTCC	69
Figura 22 - Esquema da chamada de função da RN pelo módulo SSCC	71

Lista de Tabelas

Tabela 1 - Tabela de contingência para um par de Variáveis	28
Tabela 2 - Variáveis de entrada	48
Tabela 3 - Tratamento da variável Destino	49
Tabela 4 - Experimentos de treinamento da RN.....	61

1. Introdução

No escopo do estudo da Inteligência Artificial, pesquisas em redes neurais (RN) são caracterizadas em particular pelo foco no reconhecimento de padrões e classificação de uma grande quantidade de dados [GEL]. Acredita-se que o paralelismo massivo oferecido pelas implementações de redes neurais é essencial para soluções de muitos problemas práticos em tempo real [KAN]. Neste contexto, esta pesquisa propõe a criação de um sistema para a detecção de intrusão em redes móveis - pelo uso não autorizado de telefones celulares -, através do algoritmo de classificação proposto por Teuvo Kohonen [HON+KOV]. O sistema proposto é reativo, ou seja, não prevê nem impede a intrusão, mas alerta a companhia sobre possíveis fraudes através de mudanças no perfil de utilização do telefone celular pelos usuários. O desafio em detectar fraudes consiste em fazê-lo antes de haver grandes prejuízos financeiros. O processo consiste em monitorar as ligações de cada usuário em uma determinada companhia telefônica e através do seu perfil de utilização do telefone, detectar alterações consideradas “anormais”. Este processo é realizado utilizando-se uma rede neural competitiva, que pode aprender a detectar correlações e regularidades nas suas entradas e adaptar suas respostas futuras conforme aquelas entradas. Os neurônios de redes competitivas aprendem a reconhecer grupos de vetores de entradas similares, possibilitando assim, que a rede detecte irregularidades nas ligações realizadas.

O sistema para detecção de intrusão em telecomunicações (DEITREN) está inserido no contexto do sistema SSTCC (sistema de segurança para telecomunicações contra clonagem e inadimplência) [NOT]. O SSTCC está dividido em vários módulos e é desenvolvido no LRG (Laboratório de Redes e Gerência) por vários pesquisadores. Envolve a utilização de técnicas de descrição formal e computação distribuída, visando a criação de um método de segurança para a telefonia, principalmente a telefonia móvel com a união dos vários módulos desenvolvidos. A eficiência do SSTCC independe da arquitetura ou tecnologia utilizada pela companhia, pois pelo fato do sistema telefônico ser bastante heterogêneo em termos de ferramentas utilizadas, o sistema está sendo implementado usando-se CORBA como tecnologia de objetos

distribuídos, e a linguagem de programação JAVA para uma completa independência de plataforma [HER].

O DEITREN está dividido em três fases. Definidas e adequadas as informações (variáveis) relevantes através de uma amostra de dados contendo as chamadas telefônicas de vários usuários por um período determinado, elas são submetidas como entradas para treinamento da rede neural (RN). O algoritmo da RN realiza uma auto-classificação, determinando vários grupos. Estes grupos dividem os usuários em tipos, que representam seus perfis de ligações. Após a fase de agrupamento, ou primeira fase, a rede neural é considerada treinada e é usada para definir um tipo para cada usuário através de seu histórico de ligações, que está registrado na base de dados da companhia telefônica. A terceira fase consiste em verificar se um usuário está dentro do perfil definido. Para isto, invoca-se a rede neural treinada passando como conjunto de entrada registros das últimas dez ligações de cada usuário, que estão armazenados em uma base de dados que é atualizada em tempo real. O grupo ativado na saída da rede será comparado com o tipo do usuário definido na segunda fase. Mudanças no tipo podem representar fraude.

O trabalho está organizado em sete capítulos, incluindo esta introdução. O Capítulo 2 apresenta a estrutura e os aspectos de segurança relacionados com telecomunicações, que consiste no tema de pesquisa deste trabalho. São apresentados os tipos de fraudes conhecidas visando o uso ilegal de aparelhos celulares. No Capítulo 3 são abordados os principais conceitos e aplicações relacionadas às redes neurais em geral e mais especificamente na rede neural artificial de Kohonen [KON], por se tratar da tecnologia escolhida para implementação do sistema proposto nesta pesquisa. Além disto, são mostrados conceitos e exemplos de técnicas de agrupamento, pois a implementação da RN de Kohonen é baseada nesta técnica. O Capítulo 4 descreve a gerência de segurança criada com o sistema para detecção de fraudes em telecomunicações usando redes neurais artificiais. Este capítulo contém todos os passos e fases utilizados para elaboração do DETREN. O Capítulo 5 contém as conclusões e perspectivas futuras desta pesquisa, seguidas dos anexos contendo os

códigos fonte usados para a implementação do sistema (Capítulo 6) e das referências bibliográficas (Capítulo 7).

2. Redes Móveis

Introdução

Com os recentes avanços na tecnologia de comunicação sem fio, a computação móvel e a telefonia celular são importantes áreas de pesquisa em crescimento. O presente trabalho aborda a telefonia móvel como objeto de pesquisa para elaboração de uma gerência de segurança.

Mudanças nas telecomunicações não ocorrem isoladamente. Elas desenvolvem-se em conjunto com mudanças em negócios, avanços tecnológicos, e mudam conforme as condições econômicas [DOD].

2.1 Telefonia Móvel

O sistema de telefonia móvel terrestre, também conhecido como telefonia celular, veio complementar os serviços de comunicação por meio da rede fixa. Este tipo de sistema conquistou o mundo da comunicação. Desde a década de 80, milhões de estações foram instaladas pelo mundo. No Brasil o sistema tomou impulso em meados dos anos 90.

A Telefonia celular no Brasil iniciou-se com as empresas do Sistema Telebrás (governamentais) utilizando a Banda A. A banda B foi licenciada para operadoras privadas em meados de 1997. Em 1998 foi completada a privatização das empresas telefônicas no Brasil.

A globalização da economia é um fator chave nas mudanças que estão ocorrendo na indústria das telecomunicações [DOD].

2.2 Estrutura

O telefone celular (ou telefone móvel) funciona de forma semelhante a um telefone convencional. A principal diferença é que o telefone celular é conectado à rede telefônica através de ondas de rádio, permitindo assim sua mobilidade, enquanto o telefone convencional faz uso de fios.

Cada região que tem suporte para o serviço de telefonia móvel é dividida em pequenas áreas, chamadas células, que possuem uma antena celular (ou ERB – Estação Rádio Base), para receber e emitir informações aos telefones celulares que estão em operação naquela célula. Todas as células são ligadas às Centrais de Comutação e Controle (CCC) que, por sua vez, conectam-se à rede telefônica convencional. Nas ERBs estão instalados os equipamentos de transmissão e as antenas que se comunicam com as CCC. Estas centrais controlam o fluxo das ligações efetuadas entre os aparelhos celulares e outras redes de telefonia. À medida que o usuário se movimenta, a Central transfere o sinal de uma ERB para outra, permitindo a continuidade da ligação.

2.3 Sistemas Digital e Analógico

A tecnologia de telecomunicação celular é desenvolvida a partir de sistemas de telefones móveis analógicos e digitais.

O telefone celular pode ser comparado a um rádio, capaz de receber e transmitir ligações por meio de ondas eletromagnéticas. O sistema utiliza estas ondas de rádio para fazer a ligação entre os aparelhos e a ERB mais próxima.

O sistema digital converte os sinais de voz em um sinal digital binário. Este sistema permite que vários sinais de vários usuários utilizem a mesma faixa de frequência. A frequência refere-se ao número de vezes por segundo que uma onda oscila para trás e

para frente em um ciclo completo do seu ponto de partida ao seu ponto final. Um ciclo completo, ocorre quando uma onda começa no ponto de voltagem zero, vai para a parte mais alta positiva da onda, desce para a porção de voltagem negativa e depois volta para o zero [DOD]. O sistema analógico transmite um único sinal por faixa de frequência, sem código de segurança.

2.4 Intrusões na Rede Telefônica Móvel

Intrusão na rede telefônica móvel significa o uso ilegal (não autorizado) de uma linha telefônica.

2.4.1 Clonagem

É a cópia “pirata” de um aparelho legal, contendo o mesmo ESN (Número Eletrônico de Série). Através da frequência, que é como se fosse a identidade do aparelho, pode-se obter o ESN. A frequência é transmitida pelo ar por ondas de rádio. Nos aparelhos analógicos, a clonagem pode ser feita captando a frequência de um telefone celular e transplantando-a para um aparelho diferente. Para isto precisa-se apenas de um equipamento de rádio portátil chamado *scanner*, que registra a frequência na qual telefones celulares nas proximidades operam naquele momento [NOT].

Fazer a clonagem de um telefone celular digital até pouco tempo parecia impossível, pois os sinais sonoros são convertidos em blocos de informações transmitidos na forma de códigos binários de 0 e 1, em uma sequência desordenada. Somente o aparelho receptor tem capacidade de reordenar essas informações. Dois estudantes da Universidade da Califórnia, em *Berkeley*, produziram o clone de um telefone celular digital [VEJ]. Nos aparelhos digitais os fraudadores podem utilizar programas avançados de criptografia para obter o ESN. A operação de rastreamento do código de acesso exige muito mais tempo do que para os aparelhos analógicos, mas é possível.

2.4.2 Inadimplência

O fato de um usuário não pagar as contas telefônicas não se constituiria em um problema incomum, se a linha telefônica estivesse no nome do próprio usuário devedor. Com o rápido crescimento da telefonia móvel no Brasil, são constantes as dúvidas sobre o procedimento correto a ser tomado em casos de venda e transferência da linha ou aparelho telefônico.

Os fraudadores, podem utilizar-se indevidamente das linhas telefônicas de terceiros sem precisar de técnicas que envolvem aparelhos sofisticados e conhecimentos avançados. Para isto, eles se aproveitam da “boa fé” ou da falta de informação das pessoas para obter o aparelho celular, sem passar a linha para seu nome. Desta forma, as ligações são efetuadas e quando a conta telefônica não é devidamente quitada pelo usuário do aparelho, a cobrança recai sobre o dono da linha. Este, se recusa a se responsabilizar pelo acontecido, causando grandes prejuízos para a telefônica.

Conclusão

Relatos como os apresentados acima são suficientes para estimular o crescimento da indústria de proteção aos sistemas de telefonia móvel. Além disso, novas tecnologias de informática são um recurso de obtenção de vantagem competitiva por parte de empresas que utilizaram os benefícios destas antes da concorrência.

O aspecto de segurança das redes telefônicas relevante no contexto desta pesquisa é a maneira como cada usuário costuma utilizar seu telefone. Informações como por exemplo, o tempo de duração de cada ligação, o dia e hora em que foram realizadas e o destino desta são suficientes para determinar um perfil de utilização do telefone para cada usuário. Mudanças neste perfil de utilização do telefone podem indicar fraude.

No próximo capítulo são descritas as Redes Neurais (RN) em geral e a de Kohonen em detalhes por se tratar da técnica de agrupamento escolhida para implementação do sistema proposto nesta pesquisa.

3. Redes Neurais

Introdução

O estudo de Redes Neurais iniciou-se com a inspiração biológica do sistema nervoso dos seres vivos. O elemento básico do sistema nervoso são os neurônios, os quais são responsáveis pela propagação de estímulos. O cérebro é formado por uma coleção de aproximadamente dez bilhões de neurônios interconectados. Cada neurônio é uma célula, como mostra a Figura 1, que usa reações bioquímicas para comunicação com os neurônios vizinhos. A estrutura básica de um sistema nervoso consiste na recepção de estímulos, processamento destas informações e ativação motora. Além disto, sabe-se que o sistema nervoso possui capacidade de memorização.

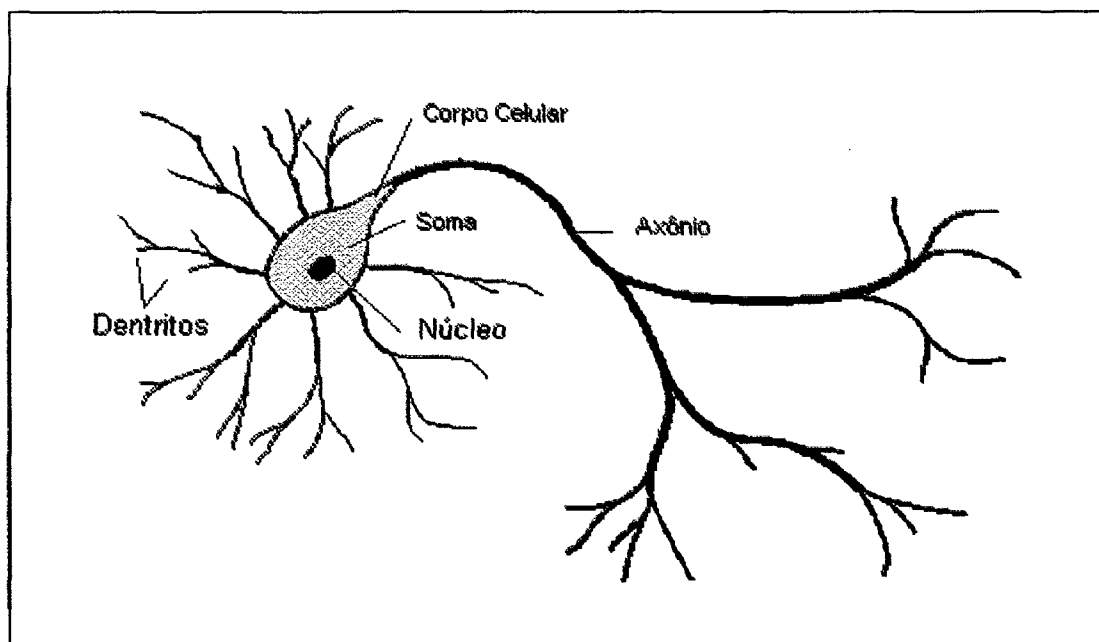


Figura 1 - Esquema do neurônio biológico.
Uma rede neural é um sistema de neurônios ligados por conexões sinápticas. Esse

sistema é dividido em neurônios de entrada, neurônios internos (ou ocultos) e neurônios de saída. Sinapse é a região através da qual os impulsos nervosos são transmitidos entre dois neurônios que estão em contato. Os impulsos recebidos por um neurônio, fazem com que eles produzam substâncias conhecidas como neurotransmissoras que são transferidas do corpo celular para o axônio [KOV]. Este, por sua vez, pode estar conectado a um dendrito de um outro neurônio. Os neurônios de entrada recebem estímulos do meio externo, os neurônios de saída se comunicam com o exterior [BAR]. Se uma pessoa encostar o dedo na água fervente, por exemplo, seus neurônios irão receber esta informação de alta temperatura e o cérebro irá realizar um processamento através das redes neurais, até um resultado ser produzido nos neurônios de saída. Este resultado é interpretado pelo corpo humano e a pessoa em questão realiza o movimento de retirar o dedo rapidamente do contato com a água quente.

3.1 Redes Neurais Artificiais

Algumas definições de redes neurais artificiais são apresentadas a seguir:

'Uma rede neural artificial é primeiramente um gráfico com representação de padrões em termos de valores numéricos fixos para seus nós e transformações entre padrões obtidos através de um simples algoritmo de transcurso de mensagem. Muitas arquiteturas de rede neural, entretanto, são processadores estatísticos, caracterizados por fazer suposições probabilísticas particulares sobre dados.' [JOR]

'Redes neurais artificiais' são redes paralelas de elementos simples (normalmente adaptativos) solidamente interconectadas e sua organização hierárquica tem o objetivo de interagir com os objetos do mundo da mesma maneira que sistemas nervosos biológicos fazem.' [KON]

'Redes neurais artificiais' constituem um campo muito abrangente e diverso de técnicas onde idéias sobre os princípios das redes neurais naturais são utilizados'

[KAN]

Uma rede neural artificial (RNA) é composta por várias unidades de processamento, que estão conectadas por canais de comunicação. Estes canais estão associados a determinados pesos. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões.

O comportamento inteligente de uma RNA vem das interações entre as unidades de processamento da rede. A maioria dos modelos de redes neurais possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. Arquiteturas neurais são organizadas em camadas, com unidades que podem estar conectadas às unidades da camada posterior. A rede neural passa por um processo de treinamento a partir dos casos reais conhecidos, adquirindo, a partir daí, a sistemática necessária para executar adequadamente o processo desejado dos dados fornecidos. Sendo assim, a rede neural é capaz de extrair regras básicas a partir de dados reais, diferindo da computação programada, onde é necessário um conjunto de regras rígidas pré fixadas e algoritmos. Usualmente as camadas são classificadas em três grupos:

- Camada de Entrada: onde os padrões são apresentados à rede;
- Camadas Intermediárias ou Ocultas: onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;
- Camada de Saída: onde o resultado final é concluído e apresentado.

A Figura 2 mostra um exemplo rede neural artificial - onde os quadrados representam os neurônios -, com cinco, oito e um neurônios nas camadas de entrada, intermediária e de saída, respectivamente. As linhas ligando os quadrados da Figura 2 representam os canais de comunicação da rede por onde ocorre o processamento das informações de entrada (camada de entrada) a fim de produzir um resultado (camada de saída).

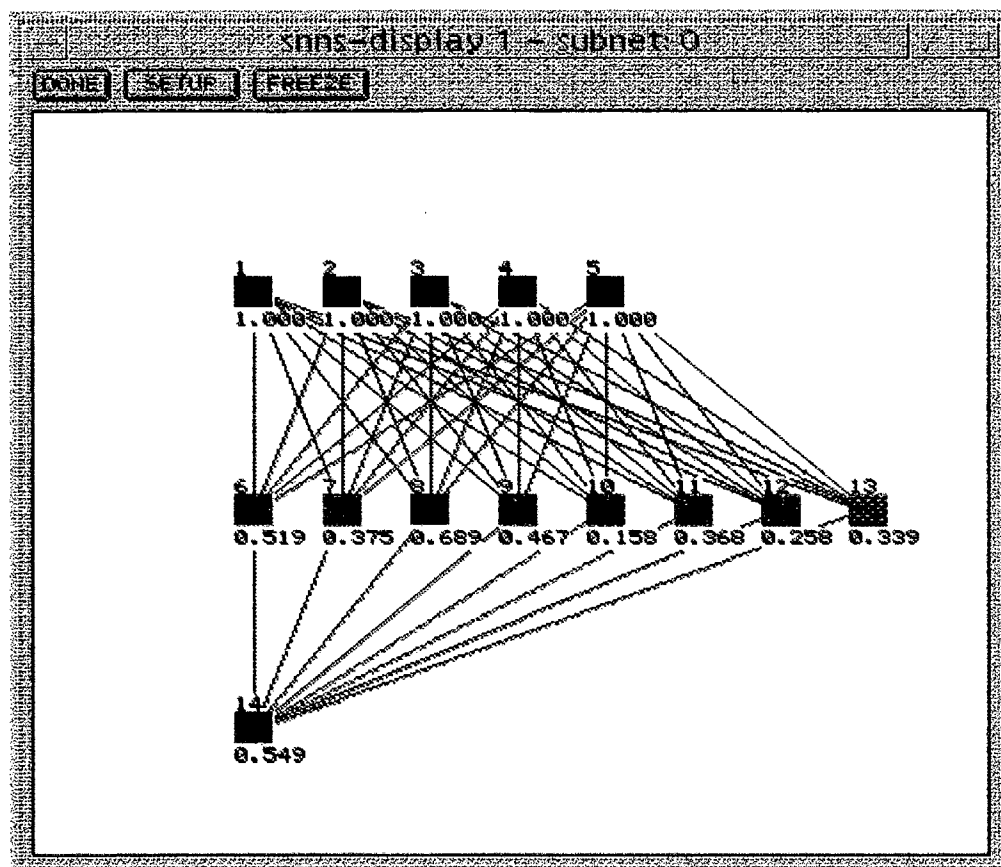


Figura 2 – Esquema de uma Rede Neural Artificial.

Uma rede neural é especificada, principalmente pela sua topologia, pelas características dos elementos computacionais ou nós e pelas regras de treinamento.

As redes neurais também se caracterizam pela capacidade de generalização. Neste caso apresentam-se exemplos sobre um determinado problema para que a rede seja capaz de generalizar quando situações similares se apresentarem.

3.1.1 Tipos de Treinamento

3.1.1.1 Supervisionado

No treinamento supervisionado, para cada vetor de entrada submetido à rede há um vetor de saída associado. A saída produzida é comparada com a saída desejada correspondente. A diferença (ou erro) entre a saída desejada e a saída obtida pela rede é atualizada na entrada, fazendo com que os pesos sejam ajustados, de acordo com um algoritmo de treinamento. Este procedimento tem por objetivo diminuir o erro provocado.

Para as redes neurais com aprendizado supervisionado, pode-se fazer uma analogia a um processo de aprendizado humano. Supondo que queira-se aprender a identificar a origem de vinhos franceses através de sua degustação (*Bordeau, Bourgogne, Côtes du Rhone*, etc.). Para isto pega-se algumas garrafas de cada tipo de vinho. Na primeira etapa, a etapa de aprendizado, experimenta-se os vinhos um a um e tenta-se identificá-los através de seu paladar, perfume, etc. Em seguida verifica-se a etiqueta para ver os acertos. Os erros são levados em conta e o aprendizado é adquirido através dele. Em seguida, volta-se a provár os vinhos a fim de tentar assimilar suas características. Através de um processo iterativo, experimenta-se e re-experimenta-se os vinhos até que seja possível identificar todas as garrafas selecionadas segundo sua origem. Desta forma a etapa de aprendizado chega ao fim. O que aconteceu no cérebro? A rede de neurônios do cérebro armazenou o conhecimento que permite distinguir os tipos de vinho. As redes neurais com aprendizado supervisionado aprendem exatamente da mesma maneira. Introduzindo exemplos sobre o que deseja-se que aprenda. Através de um processo iterativo e automático ela aprende sobre o problema.

3.1.1.2 Não Supervisionado

No treinamento não-supervisionado não exige vetores da saída desejada para cada vetor de entrada da rede. Assim, a atualização dos pesos não depende da comparação entre as saídas obtida e desejada.

O algoritmo de treinamento ajusta os pesos da rede de forma que produzam vetores de saída consistentes. A consistência dos valores significa por exemplo, que a aplicação de dois vetores de treinamento similares produzem a mesma saída. Portanto, o algoritmo de treinamento extrai as propriedades estatísticas do conjunto de treinamento e agrupa os vetores similares em classes específicas [ELV].

3.1.2 Caracterização

Os modelos de redes neurais (as conexões topológicas, capacidade dos elementos computacionais e os algoritmos de treinamento associados) diferem muito. É difícil realizar a caracterização dos algoritmos de redes neurais baseados nas suas propriedades. Uma caracterização é dada em [KAN]:

- Redes *Feedforward* transformam os sinais de entradas em sinais de saída. A transformação é feita geralmente através de um treinamento supervisionado.
- Redes *Feedback* definem as ativações iniciais dos elementos computacionais a partir de sinais de entrada, e executam um ciclo de transições de estado, sendo o resultado o sinal de saída. A intensidade da conexão geralmente é dada à rede.
- Redes de auto-organização, competitivas onde os elementos computacionais competem uns contra os outros nas ativações. O treinamento é feito de maneira não

supervisionada desde que a rede adapte fortemente as conexões baseadas nas propriedades dos dados de entrada.

3.1.3 Agrupamento

Existem várias técnicas que realizam agrupamento, desde estatísticas, até utilizando-se de inteligência artificial, com redes neurais.

A análise de grupos é uma das maiores ferramentas na análise de dados exploratória aplicada em muitas ciências. A análise de dados é caracterizada por muitos testes, validação e complexa aceitação. Normalmente, um grande número de procedimentos e rotinas são usados com o intuito de entender as peculiaridades dos dados em consideração. Em muitos problemas técnicos, pode-se usar uma variedade de técnicas de agrupamento com o objetivo de agrupar pontos em um espaço uni ou multi dimensional de uma maneira que todos os pontos em um único grupo tenham alguma relação natural uns com os outros, e pontos que não pertençam ao mesmo grupo são diferentes de alguma forma. Estas técnicas são ferramentas para descobrir, para gerar idéias e para testar ou formar hipóteses sobre a estrutura natural dos dados em consideração [GEL].

3.1.3.1 Medidas de Similaridade

Na análise de grupos não é feita nenhuma suposição a respeito do número de grupos ou da estrutura destes. Agrupamentos são feitos com base em similaridades ou distancias. O maior esforço para produzir uma estrutura de grupo bastante simples a partir de um conjunto de dados complexo necessariamente requer uma medida de “similaridade”. Há frequentemente muita subjetividade envolvida na escolha de uma medida de similaridade. Considerações importantes incluem a natureza das variáveis e o conhecimento do assunto em questão.

Quando os objetos são agrupados, proximidade é normalmente indicada por alguma

espécie de distância. Por outro lado, variáveis são normalmente agrupadas com base na correlação dos coeficientes ou como medidas de associação.

3.1.3.1.1 Distâncias para um par de itens

A descrição de distancia dada a seguir está em [JOH]. A distância Euclidiana entre duas observações p -dimensionais $x = [x_1, x_2, \dots, x_p]'$ e $y = [y_1, y_2, \dots, y_p]'$ é,

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_p - y_p)^2} = \sqrt{(x - y)'(x - y)} \quad (3 - 1)$$

A distancia estatística entre as mesmas duas observações é dada na forma

$$d(x, y) = \sqrt{(x - y)' A (x - y)} \quad (3 - 2)$$

Geralmente as entradas de A^{-1} são amostras de divergencias e codivergencias. Entretanto, sem um conhecimento prévio dos grupos distintos, estas quantidades não podem ser computadas. Por esta razão, a distancia Euclidiana é frequentemente a mais utilizada para agrupamento.

3.1.3.1.2 Similaridades e medidas de associação para um par de variáveis

Em algumas aplicações são variáveis ao invés de itens que precisam ser agrupados.

Medidas de similaridade para variáveis geralmente tem a forma da correlação dos coeficientes na amostra [JOH].

Quando as variáveis são binárias, os dados podem ser combinados na forma de uma tabela de contingência. Neste caso, as variáveis que delineiam as categorias. Para cada par de variáveis, há n itens categorizados na Tabela 1. Com o código 0 e 1 comum, a Tabela 1 fica:

		Variável k		Totais
		1	0	
Variável i	1	A	b	a+b
	0	C	d	c+d
	Totais	A+c	b+d	n= a+b+c+d

Tabela 1: Tabela de contingência para um par de variáveis.

Por exemplo a variável i é igual a 1 e a variável k igual a 0 para b dos n itens.

A fórmula de correlação aplicada as variáveis binárias na tabela de contingência é dada como

$$r = \frac{ad - bc}{[(a + b)(c + d)(a + c)(b + d)]^{1/2}} \quad (3 - 3)$$

Este número pode ser tido como uma medida de similaridade entre duas variáveis.

A seguir é apresentado um exemplo de agrupamento.

3.1.3.2 Faces de Chernoff

Chernoff [JOH] sugeriu representar observações de faces cujas características (forma da face, curvatura da boca, comprimento do nariz, tamanho do olho, posição da pupila, etc) são determinadas por variáveis para realização de agrupamentos.

As faces de Chernoff, como foram originalmente desenvolvidas, possuem 18 variáveis. A designação das variáveis para os aspectos das faces foi feita pela experimentação. Produzindo escolhas diferentes, obtém-se resultados diferentes. No início, duas ou três variáveis são suficientes para distinguir grupos, estas variáveis podem ser associados às características faciais proeminentes. Associando uma variável “importante” com uma característica como comprimento do nariz é preferível do que uma característica menos proeminente como posição da pupila, pois isto possibilita a seleção de grupos mais facilmente. A Figura 3 mostra a representação das faces de Chernoff de acordo com as variáveis descritas a seguir:

- Altura de meia face;
- Largura da face;
- Posição do centro da boca;
- Inclinação dos olhos;
- Excentricidade $\left(\frac{\text{altura}}{\text{largura}} \right)$ dos olhos;
- Comprimento do olho;
- Curvatura da boca; e
- Comprimento do nariz.

Foram agrupadas faces “similares” em sete grupos. Se fosse desejado um número menor de grupos, provavelmente seriam feitas as combinações dos grupos 5 , 6 e 7 e também dos grupos 2 e 3, para obter 4 ou 5 grupos.

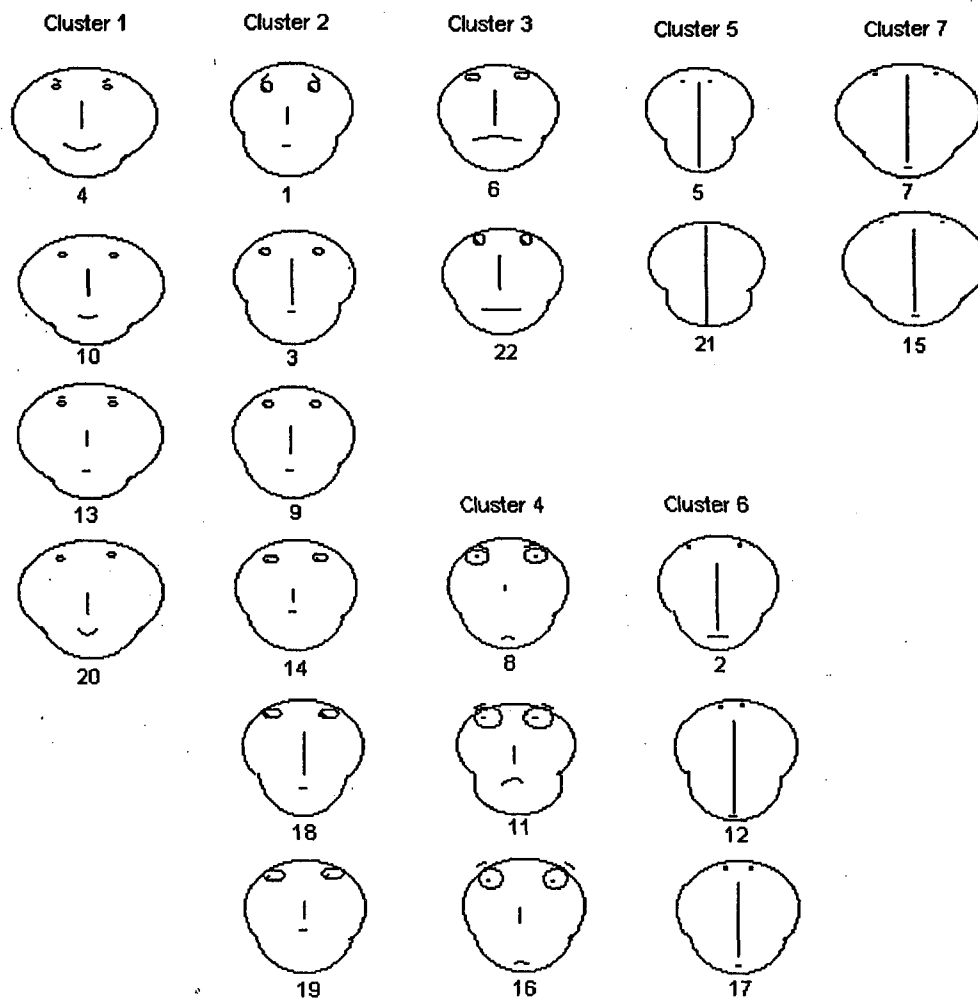


Figura 3 – Faces de Chernoff.

Existem muitas maneiras de se realizar o agrupamento de uma grande quantidade de dados com diferentes medidas de similaridade entre pares de objetos. Porém, a mais utilizada usa o conceito de distancias para realizar o agrupamento de observações de itens, e correlações para agrupar variáveis.

3.1.4 Regra de Treinamento de Kohonen

O modelo kohonen é uma rede neural tipo *Feedforward* e competitiva de treinamento não supervisionado. Considerando uma rede de duas dimensões, o paradigma de kohonen não impõe nenhuma forma topológica, podendo ser hexagonal, retangular, triangular, etc.

O esquema básico de um modelo kohonen tem a propriedade de modificar a si próprio. Assim, os neurônios próximos ao modificado respondem similarmente. Os neurônios da camada competem entre si para serem os vencedores a cada modificação. O neurônio cujo vetor de pesos gerar a menor distância Euclidiana com o vetor de entrada é o vencedor. Dentro deste modelo não supervisionado, não somente os pesos do vencedor, mas também os pesos dos seus vizinhos (dentro de um senso físico) são ajustados.

Este paradigma é baseado na teoria de que as células nervosas corticais estão arranjadas anatomicamente em relação aos estímulos que recebem dos sensores às quais estão ligadas. Com relação a este tema, vale lembrar que alguns cientistas, rastreando a atividade cerebral notaram que, apesar de todas as células nervosas estarem ligadas e coligadas entre si, existem centros de atividade mais intensificados conforme a atividade exercida no momento. Quando se ouve música, por exemplo, ou quando se está efetuando um raciocínio matemático, existem áreas diferentes do cérebro com maior atividade. Este é o paradigma topológico.

Contudo o paradigma usado por Kohonen para descrever os mapas topológicos artificiais não tem o mesmo funcionamento biológico do cérebro, pois os detalhes sobre os ajustes sinápticos e a competição entre os neurônios não é totalmente conhecida na biologia. Porém os mapas descritos por Kohonen fornecem um modelo útil de rede neural para a organização adaptativa de mapas topológicos. Em ambos sistemas biológicos e artificiais, os mapas topológicos fornecem características de funcionamento úteis para um sistema complexo de reconhecimento de padrão [DAY].

3.1.4.1 Mecanismo de Funcionamento

Sinteticamente, o mecanismo de kohonen funciona da seguinte forma: os pesos sinápticos iniciam em *off*, ou seja, contendo valores aleatoriamente baixos, e um sinal de entrada x (com valores que representam uma informação qualquer) é provido para a rede sem que se especifique a saída desejada (característica da rede não supervisionada). O sinal de entrada x é descrito como:

$$x=[x_1, x_2, \dots, x_n]^T \in \mathfrak{R}^n$$

Conforme o sinal de entrada, um neurônio de saída y deverá responder melhor para a entrada respectiva, e este será o vencedor, ou seja, sempre que essa entrada for apresentada, este neurônio será disparado. O neurônio vencedor y , e seus vizinhos (V_y), terão seus pesos sinápticos ajustados e, desta forma, responderão melhor do que antes à entrada apresentada x . A rede é considerada treinada depois de todo o conjunto de treinamento (entradas) ter sido apresentado à rede e os critérios de treinamento (variáveis como o número de iterações, ou seja, quantidade de vezes que o conjunto de entradas é usado, e taxa de aprendizagem) terem sido satisfeitos.

Seguindo a fase de treinamento, tem-se a fase de teste da rede neural. Essa fase é semelhante à fase de treinamento, porém não serão mais alterados os pesos dos neurônios. Para que o teste seja mais exato, a rede poderá receber, como entrada, além dos vetores de entrada que foram apresentados, outros que não fizeram parte do conjunto de treinamento. Reconhecendo as entradas do teste de forma adequada a rede é considerada treinada com sucesso.

3.1.4.1.1 Neurônio

Cada neurônio da rede representa, na verdade, uma saída da rede, ou seja, se a rede possuir um total de 30 neurônios na camada de saída, haverá, conseqüentemente, 30 saídas possíveis para qualquer quantidade de entradas que a rede estiver sendo

submetida.

Outro aspecto fundamental é que o neurônio estará amplamente conectado com as entradas, isso quer dizer que, havendo 10 entradas, haverá, conseqüentemente, 10 conexões sinápticas entre cada neurônio de saída e os 10 pontos da camada de entrada.

Os neurônios de uma camada competitiva se distribuem para reconhecer vetores de entrada apresentados.

3.1.4.2 Arquitetura

A arquitetura para as redes competitivas, é mostrada na Figura 4 e descrita em seguida de acordo com [DEM]:

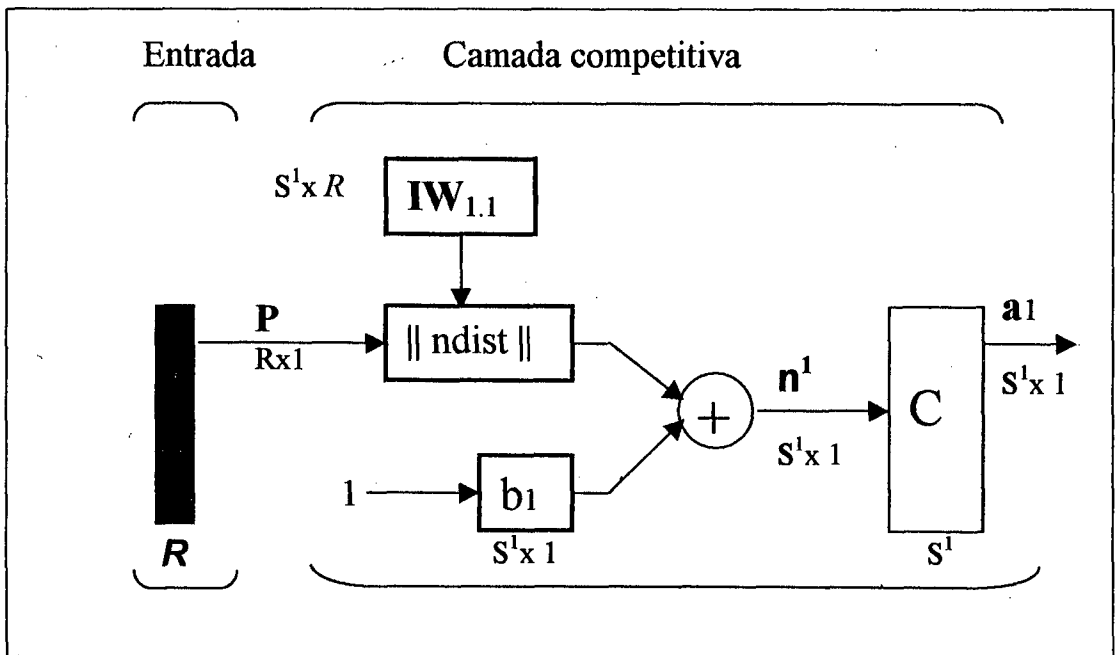


Figura 4 – Arquitetura das redes competitivas.

A caixa $\|dist\|$ da Figura 4 aceita o vetor de entrada P e a matriz de pesos de entrada $IW_{1,1}$, e produz um vetor contendo S^1 elementos. Os elementos são o negativo das distancias entre o vetor de entrada e vetores $iIW_{1,1}$ formados a partir das linhas da matriz de pesos de entrada.

A entrada da rede n^1 de uma camada competitiva é computada achando-se a distancia negativa entre o vetor de entrada \mathbf{P} e os vetores de pesos e adicionando o *bias* (inclinação) \mathbf{b} . Se todos os *bias* são zero, a entrada máxima da rede que um neurônio pode ter é 0. Isto acontece quando o vetor de entrada \mathbf{P} é igual ao vetor de pesos daquele neurônio.

A função de transferência competitiva aceita um vetor de entrada da rede para uma camada e retorna uma saída de neurônios de zeros para todos os neurônios exceto para o vencedor, isto é, o neurônio associado com o maior elemento positivo da entrada da rede n^1 . A saída dos vencedores é 1.

3.1.4.3 Algoritmo

A implementação do algoritmo foi retirada do trabalho de Meyer intitulado *Self-Organizing Processes*. [MEY]. Além da publicação original, descrita em [KOH1], a regra de aprendizado de Kohonen foi descrita várias vezes. A notação apresentada a seguir foi a usada por Ritter [RIT] e Bauer/Paweizik [ING].

O algoritmo descreve um mapa Ω a partir de um espaço de entrada V em um espaço de saída A . O espaço de saída consiste de nós n_j , que estão organizados em uma ordem topológica. Para cada nó n_j em A há um ponteiro (ou indicador) w_j em V . Cada ponto v de V é mapeado para aquele nó n_i , cujo ponteiro w_i mais próximo está de v , isto é

$$n_i : d^V(w_i, v) = \min_{n_j \in A} d^V(w_j, v), \quad (1)$$

onde $d^V(w_j, v)$ denota a distância no espaço V entre w_j e v .

Consequentemente, o mapa Ω é determinado pelos ponteiros w_j . Durante cada passo de iteração estes ponteiros são ajustados.

3.1.4.3.1 Regra de Aprendizado

Um passo simples de iteração começa com a geração de um estímulo $v \in V$ de acordo com uma função de densidade de probabilidade $p(v)$. Então os ponteiros w_i e ponteiros w_j dos nós da vizinhança de n_i são trocados um pequeno passo em direção a v :

$$\delta w_j = \epsilon h_{j,i}^0(d^A(n_j, n_i))(v - w_j) \quad \forall n_j \in A. \quad (2)$$

A extensão da vizinhança de n_i é determinada pela função $h_{j,i}^0$, na qual depende da distância $d^A(n_j, n_i)$ entre n_j e n_i no espaço de saída. Uma escolha típica para é

$$h_{i,j}^0(d) = e^{-d^2/2\sigma^2} \quad (3)$$

A etapa ϵ e o raio da vizinhança σ são parâmetros influenciando a convergência do algoritmo. Eles devem decrescer enquanto o número dos passos de aprendizagem aumenta. Geralmente eles decrescem exponencialmente.

A fase de aprendizado começa com a inicialização de todos w_j , a qual pode ser feita randomicamente ou baseada em uma informação previamente conhecida sobre o mapa. O número de passos de aprendizado podem ser dados explicitamente ou por um

critério de terminação (ex: um raio de vizinhança menor σ_{min}).

3.1.4.3.2 Preservação da Vizinhança

v_1, v_2 e v_3 denotam três estímulos no espaço (ou camada) de entrada e n_{i1}, n_{i2} , e n_{i3} denotam os nós correspondentes do espaço de saída no qual o estímulo é mapeado. Então um mapa Ω é tipo para preservar totalmente a vizinhança, se para cada tripla v_1, v_2 e v_3 o seguinte ocorrer:

$$d^V(v_2, v_1) \geq d^V(v_3, v_1) \Leftrightarrow d^A(n_{i2}, n_{i1}) \geq d^A(n_{i3}, n_{i1})$$

e

$$d^V(v_2, v_1) \leq d^V(v_3, v_1) \Leftrightarrow d^A(n_{i2}, n_{i1}) \leq d^A(n_{i3}, n_{i1}) \quad (4)$$

Por estas condições serem simétricas com respeito a V e A , o mapa inverso Ω^{-1} também preserva totalmente a vizinhança. Bauer e Pawelzik [ING] mostram que a condição (4) é suficiente mas não necessária para a preservação total da vizinhança.

3.1.4.4 Testes / Validação

Acabada a etapa de aprendizado passa-se à etapa de utilização do conhecimento adquirido. É o que se espera da rede neural uma vez que ela passou pela etapa de treinamento. No caso da detecção de intrusão em redes móveis, espera-se que a rede possa distinguir a utilização de telefones celulares por fraudadores da utilização de telefones celulares por seus donos legais, mesmo que estas não tenham feito parte do seu aprendizado.

3.1.5 Aplicações

As aplicações de redes neurais tem destaque em áreas onde podem oferecer uma abordagem mais apropriada do que a computação tradicional. A habilidade em formar mapeamentos não-lineares tem tornado as redes neurais eficientes na modelagem e previsão de sistemas complexos.

Redes neurais são eficientes no reconhecimento de padrões, mapeamento de padrões e teoria de controle em um conjunto de dados, quer estes representem imagens digitalizadas ou o comportamento de uma moeda no mercado de câmbio. Alguns exemplos de áreas específicas onde RN podem ser utilizadas incluem o reconhecimento de fala, processamento e análise de imagens, classificação de sinais sonares e controle adaptativo. As RN são bastante efetivas no aprendizado de padrões a partir de dados incompletos, com ruídos e até compostos de exemplos contraditórios. A habilidade de manipular estes dados imprecisos faz com que as redes neurais sejam eficazes no processamento de informações financeiras e em tarefas onde especialistas não estão à disposição ou um conjunto de regras claras não pode ser facilmente formulado. As redes neurais não são, portanto, apropriadas para as aplicações em que a explicação do raciocínio é fundamental.

O uso de redes neurais tem sido estudado para muitas aplicações, em um grande espectro de campos, desde em tarefas de difícil classificação de padrões, como bioquímica até a gerência de redes de computadores onde as RN são desenvolvidas para substituir algoritmos convencionais.

Conclusão

Pelo motivo das RN aprenderem por exemplos, elas tem o potencial de construir

sistemas de computação que não precisam ser programados. Isto reflete uma abordagem radicalmente diferente da computação comparada com métodos tradicionais, que envolve o desenvolvimento de programas de computador. Em um programa de computador, cada passo que o computador executa é especificado antecipadamente pelo programador, um processo que requer tempo e recursos humanos. Em contraste, as RN possuem uma amostra de entrada e aprendem a dar a saída correta para cada entrada. [JUD]

Como já mencionado anteriormente, uma rede neural é primeiramente um gráfico com representação de padrões em termos de valores numéricos fixos para seus nós e transformações entre padrões obtidos através de um simples algoritmo de transcurso de mensagem. Muitas arquiteturas de rede neural, entretanto, são processadores estatísticos, caracterizados por fazer suposições probabilísticas particulares sobre dados [JOR].

As redes neurais tem vantagem sobre os algoritmos tradicionais devido ao paralelismo massivo de suas implementações, pois seu poder de processamento é medido pelo número de atualizações das interconexões por segundo. Em contraste, o poder de performance dos sistemas tradicionais é caracterizado pelo número de instruções que podem ser executadas por segundo, em sequência, através de um único processador.

Tenta-se formalizar um conhecimento que é muitas vezes subjetivo. Outros elementos além do conhecimento formal, como intuição, sensibilidade ao problema, etc. são utilizados pelo especialista. Estes elementos são dificilmente formalizáveis através de regras [KOH]. As RN podem ser melhor vistas como uma classe de algoritmos para modelagem estatística e predição. Baseado em uma fonte de treinamento de dados, o objetivo é produzir um modelo estatístico do processo no qual os dados foram gerados de maneira a permitir a melhor predição a ser feita para novos dados. Métodos de redes neurais tem grande importância em problemas onde assuntos estatísticos dominam e onde dados são facilmente obtidos.

4. Gerencia de Segurança para a Detecção de Intrusão em Telecomunicações Através de Redes Neurais (DEITREN).

Introdução

Considerando o estado da arte relacionado a segurança na área de telefonia móvel, há bastante interesse em desenvolver novas tecnologias³ relacionadas a clonagem de celulares. Entretanto, pelo que tem-se divulgado, poucos trabalhos a nível de desenvolvimento de *software* tem sido feitos. Em relação ao uso de redes neurais para o reconhecimento de padrões de utilização de telefone não foram encontrados resultados satisfatórios [MIR]. Assim, este trabalho foi baseado em investigação dos resultados de técnicas de reconhecimento de padrão empregadas em outras áreas de pesquisa. Foram testadas diferentes características e diferentes redes neurais que façam o melhor uso destas características para garantir a melhor performance na classificação considerando dados de treinamento e recursos computacionais limitados.

Acredita-se que o paralelismo massivo oferecido pelas implementações de redes neurais é essencial para soluções de muitos problemas práticos em tempo real [KAN]. O desafio em detectar fraudes consiste em fazê-lo antes de haver grandes prejuízos financeiros. O sistema proposto é reativo, ou seja, não prevê nem impede fraudes, mas alerta a companhia sobre possíveis fraudes, minimizando os prejuízos das empresas de telecomunicações.

³ como a tecnologia Boot Block usada pela corporação Intel.

4.1 DEITREN

A implementação de um sistema de detecção de intrusão em redes móveis usando redes neurais pode ser dividida em três fases: 1) treinamento da RN, 2) definição de tipo de usuário e 3) controle de ligações. A 1ª fase consiste no treinamento da RN. Nesta fase são definidos os grupos (tipos de usuários) através do agrupamento dos dados da amostra (Figura 5).

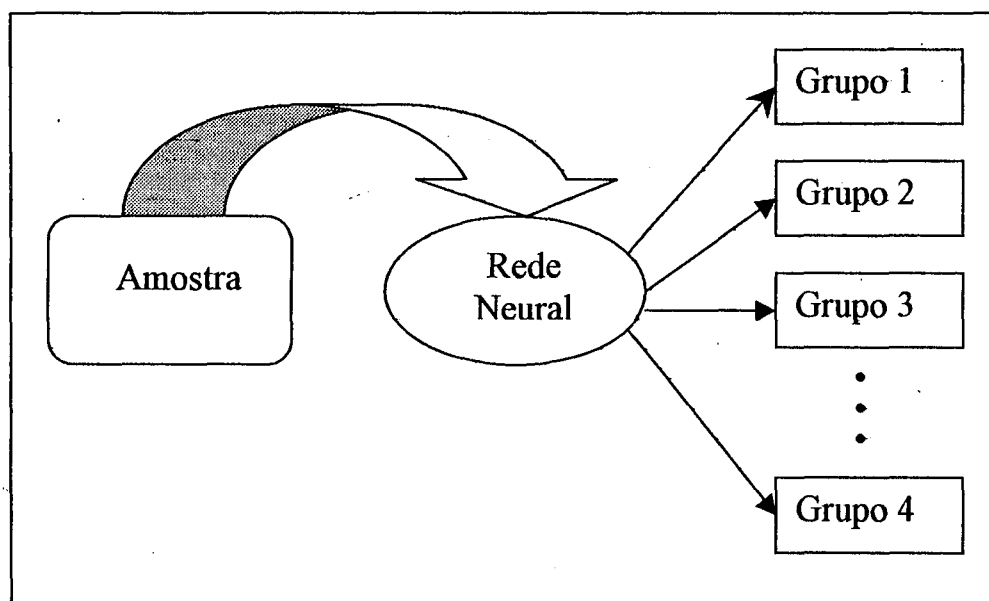


Figura 5: Esquema genérico do sistema DEITREN: 1ª fase.

Na 2ª fase define-se um tipo (grupo) para cada usuário. A definição do tipo de cada usuário é feita usando seu histórico de ligações para invocar a RN treinada. O grupo ativado na saída da rede representa seu tipo, que é armazenado em uma base de dados. O esquema de funcionalidade da 2ª fase está representado na Figura 6.

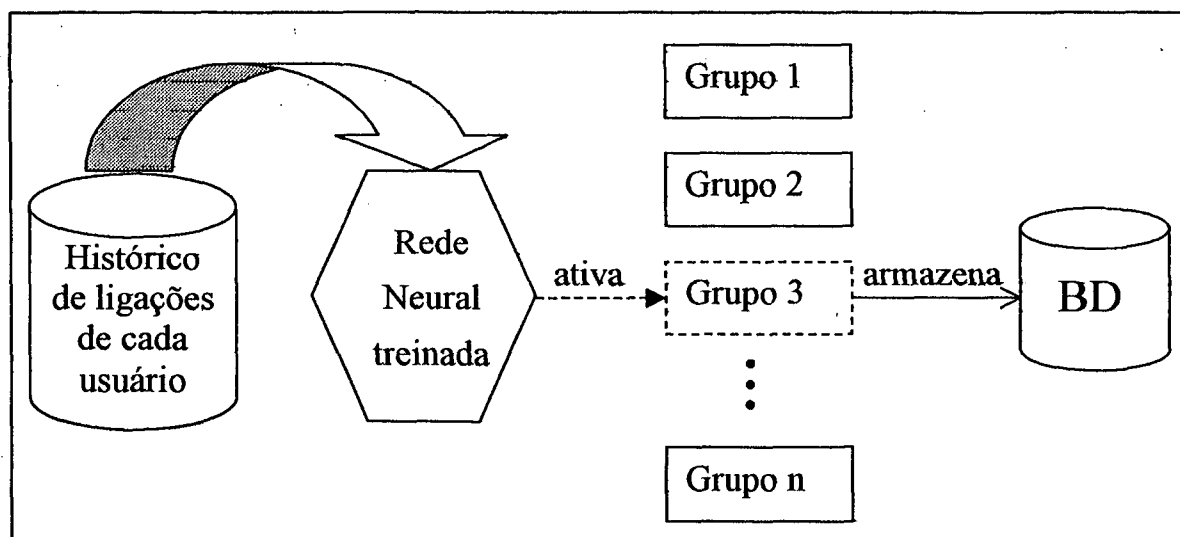


Figura 6: Esquema genérico do sistema DEITREN: 2ª fase.

A 3ª fase, como ilustrada na Figura 7, consiste no controle das ligações efetuadas. A rede neural é invocada para detectar se houve ou não fraudes. A cada dez ligações efetuadas por usuário, a RN é novamente invocada e a saída produzida (tipo) é comparada com o tipo armazenado na BD, definido na 2ª fase. Mudanças no tipo de um período para outro podem indicar intrusões.

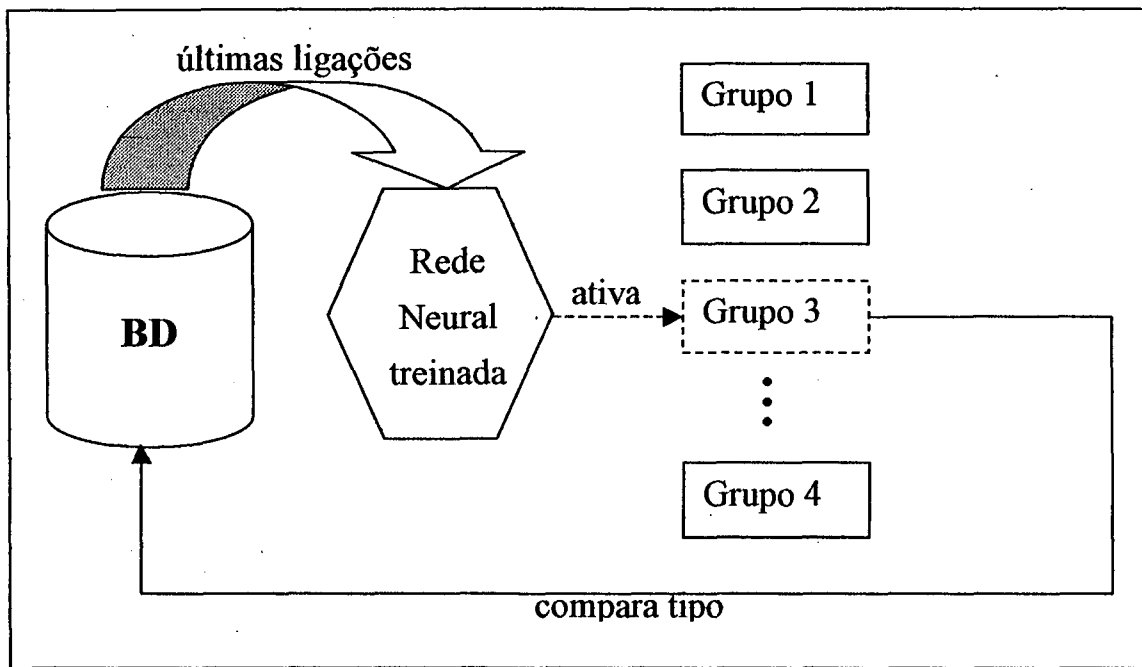


Figura 7: Esquema genérico do sistema DEITREN: 3ª fase.

No estudo, explorou-se diferentes combinações de características extraídas de ligações, bem como, diferentes topologias e parâmetros para a rede.

Os experimentos foram realizados em um PentiumII (64Mb de RAM e 4Gb de espaço disponível em disco). São apresentados os experimentos realizados, passo a passo nos próximos itens.

4.1.1 Amostra de Dados

A seleção dos dados de treinamento apresentados para a RN influenciam no quanto a rede “aprende” ou não uma determinada tarefa. Como uma criança, o quão melhor uma rede irá “aprender” depende dos exemplos apresentados. Um bom conjunto de exemplos é necessário para o processo de aprendizado ser eficiente. Um conjunto pobre de exemplos resultará em um aprendizado pobre por parte da rede. O conjunto de treinamento deve também refletir a variação nos padrões que a rede irá encontrar

após seu treinamento.

A amostra são os dados históricos, que consiste nas chamadas telefônicas dos clientes da telefônica contendo informações detalhadas de cada ligação realizada. Estas informações são submetidas ao treinamento da rede neural. Estes dados são imprescindíveis para que o DEITREN funcione eficientemente, pois o conhecimento da rede para classificação dos tipos depende do quão próximo da realidade estão os dados fornecidos.

Foram coletados aproximadamente cinco milhões de registros com ligações de uma central telefônica localizada no Rio Grande do Sul. Os arquivos de dados foram obtidos da companhia telefônica através de ftp (35Mb), para tornar o processo de envio mais veloz. Os dados foram armazenados em um arquivo texto.

4.1.1.1 Dados Brutos

O arquivo enviado pela telefônica, ao ser aberto, estava como mostrado na Figura 8. Para interpretar o significado das informações foi usada a descrição mostrada na Figura 9. Esta descrição foi fornecida pela telefônica na forma de um arquivo texto juntamente com o arquivo de dados. A manipulação (visualização, edição e organização) das informações foi feita utilizando o banco de dados *Access* e a planilha eletrônica *Excel*.

```

050260499EMBRATEL CRT          182502          240299260499TCI
222REALTFI
190519516023 250499193850598
TR0009051000270000000000000092222200191455
176519535961 11049920545054
3351424069PR00030510000000000000000072722200191597
176519535961 12049915045054
3351424177PP000305100000000000000000083122200191598
176519535961 13049921595054
3351424028PR000405100000000000000000086522200191599
176519535961 14049923495054
3351424028PR0011051000000000000000000182922200191600
176519535961 15049921065054
3351424028PR0009051000000000000000000155422200191601
176519535961 16049920215054
3351424028PR0007051000000000000000000127822200191602
176519550168 02049910505053 7793404
PR0004051000000000000000000086622200191797
176519563239 03049914345054
3341454994PP000305100000000000000000083122200192076
176519569736 05049910485054
3114725097PP000305100000000000000000083122200192290
176519569736 16049913425054
3114731412PP0012051000000000000000000238222200192291
176519587926 12049922475054
3351478492PR000305100000000000000000072722200193404
176519614757 20049911555053 7637646
PP0008051000000000000000000169822200194069
176519636221 13049919185054
3375542117PP000305100000000000000000083122200194552
176519646139 16049920155054
3351424069PR000405100000000000000000086522200194918
176519647011 01049919375054
3354848994PP000305100000000000000000083122200194922
176519649257 15049914255054
3351424995PP0006051000000000000000000134822200195077
176519667303 08049916065054
3381435469PP000305100000000000000000083122200195490
176519667303 09049919545054
3381424441PP000305100000000000000000083122200195491

```

Figura 8 – Formato do arquivo.

```

SQL> DESC TTF
Name                               Null?      Type
-----
TIPO_REG                            VARCHAR2(2)
EST_CHAMADA                          VARCHAR2(1)
NUM_SEQ_REG                          NUMBER(8)
NUM_IDENT_CHAM                       NUMBER(8)
NUM_CHAM_RELAT                       NUMBER(8)
CAUSA_P_SAIDA                        NUMBER(1)
IND_NUM_A                            NUMBER(2)
NUM_DE_A                             VARCHAR2(10) - quem originou
NUM_MS_A                             NUMBER(10)
NUM_SERIE_A                          VARCHAR2(8)
IND_NUM_B                            NUMBER(2)
CAMPOSS                              VARCHAR2(18) - número discado
NUM_MS_B                             NUMBER(10)
NUM_SERIE_B                          VARCHAR2(8)
CATEG_A                              NUMBER(2)
CATEG_B                              NUMBER(2)
LADO_TARIFADO                       NUMBER(1)
DATA_INC_TARIF                      DATE       - data da chamada
HORA_INC_TARIF                      NUMBER(6)  - hora de início da
chamada
DURACAO_TARIFA                      NUMBER(6)
TEMP_OCUP_CAN_A                     NUMBER(6)
TIPO_ULT_CAN_A                      NUMBER(2)
NUM_INT_ULT_CAN_A                   NUMBER(5)
TEMP_OCUP_CAN_B                     NUMBER(6)
TIPO_ULT_CAN_B                      NUMBER(2)
NUM_INT_ULT_CAN_B                   NUMBER(5)
TEMPO_OCUP_RE                       NUMBER(6)
HORA_INI_HAND_A                     NUMBER(6)
HORA_FIM_HAND_A                     NUMBER(6)
ROTA_HANDOFF_A                      VARCHAR2(7)
HORA_INI_HAND_B                     NUMBER(6)
HORA_FIM_HAND_B                     NUMBER(6)
ROTA_HANDOFF_B                      VARCHAR2(7)
TEMP_INTERRUP                       NUMBER(6)
DESC_CHAM_ANORMAL                   NUMBER(1)
CODIGO_FALHA                        NUMBER(5)
INDIC_REENCAMIN                     NUMBER(1)
IDENT_CENTRAL                        VARCHAR2(15)
NUM_REGISTRO                         NUMBER(2)
NUM_ABREVIADO                       VARCHAR2(4)
ROTA_SAIDA                          VARCHAR2(7)
ROTA_ENTRADA                        VARCHAR2(7)
CLASSE_TARIF                        NUMBER(3)
INDC_COMUT_TARIF                    NUMBER(1)
INDC_SERV_IMEDI                     NUMBER(1)
PRECO                                NUMBER(8)
INDC_CHAM_PRIORT                     NUMBER(1)
INDC_PROC_DISC                      NUMBER(1)
ORIG_P_TARIF                        NUMBER(4)
TIPO_PROCED                         NUMBER(2)
INDC_SERV_ASS                       NUMBER(2)
ASSINANTE_X                         VARCHAR2(10)

```

Figura 9 – Descrição dos campos do arquivo da amostra.

Com estes dados disponíveis foi possível realizar uma análise e seleção do que fosse relevante para o sistema proposto. A seguir é apresentada uma descrição do procedimento para seleção das variáveis relevantes relacionadas com o problema em questão.

4.1.2 Análise dos Dados

Objetivos relacionados com a explicação de um fenômeno físico ou social devem ser especificados e então testados pela reunião e análise de dados. Uma análise dos dados reunidos através de observação ou experimentação pode revelar uma explicação diferente do fenômeno. A reunião de um conjunto de dados deve ser feita pelo investigador a partir de variáveis diferentes [JOH].

Um dado se transforma em informação quando este ganha um significado para seu utilizador. Senão ele continua sendo simplesmente um dado. Em suma a informação é um dado tratado. A análise dos dados consiste na definição a adequação das informações relevantes (variáveis de entrada) para submeter como entrada da rede neural. Estas informações devem caracterizar o perfil de utilização do telefone pelos usuários.

O arquivo com a amostra de dados foi importado no *software* Excel (planilha eletrônica) e Access (banco de dados), possibilitando a seleção dos campos importantes e relevantes no escopo deste trabalho.

Feita uma análise dos principais componentes chegou-se a conclusão que as características mais relevantes para resolver o problema em questão seriam as descritos na Tabela 2.

Nome do campo	Descrição
CAMPOSS	Representa o número de telefone discado
DATA_INC_TARIF	Data de realização da ligação
DURACAO_TARIFA	Tempo de duração (em minutos)
HORA_INC_TARIF	Hora de início

Tabela 2. - Variáveis de entrada.

Foram escolhidas estas características por serem informações importantes na definição de um padrão de utilização do telefone pelos usuários. Sua variação e o cruzamento de suas várias combinações são úteis para a RN ser capaz de formar grupos distintos que representam um tipo de usuário. Uma característica interessante das redes neurais é também a sua capacidade de tratar dados qualitativos, por isso foram definidas ambas variáveis: quantitativas (hora e duracao) e qualitativas (número discado e data). A seguir são apresentadas a justificativa e a relevância, bem como o pré processamento realizado para a utilização das informações apresentadas na Tabela 2 pela RN.

4.1.3 Pré Processamento das Variáveis de Entrada

Após a definição das informações relevantes a serem utilizadas pelo sistema, é feito um tratamento dos dados. Este tratamento visa transformar as informações em um formato que possa ser utilizado pela rede neural. Considerações importantes incluem a natureza das variáveis (discreta, contínua binária) e o conhecimento subjetivo do problema [JOH]. O sucesso em relação a eficiência da RN depende fortemente do modo de tratamento das variáveis. Dependendo da grandeza em consideração, as informações são transformadas em valores binários, entre 0 e 1, discretos, divididos intervalos iguais, ou contínuos. Isto é feito através de *scripts* implementados no *software* Matlab 5.0.1 [DEM].

A seguir é apresentado como cada variável foi tratada e a relevância delas para o

sistema.

Destino da ligação

Por ser um dado identificador ou qualitativo, o destino da ligação é importante para, através do número discado, saber que tipo de ligação foi feita. As ligações foram divididas nos seguintes tipos:

- Local: para mesma cidade;
- Interurbana: para cidades diferentes da de origem, mas dentro do território brasileiro;
- Internacional: para cidades localizadas fora do território brasileiro; e
- Especial: serviços como por exemplo Disk Horóscopo, Informações, etc.

A Tabela 3 mostra um esquema da transformação da variável destino da ligação de forma binária. A variável Destino representa quatro entradas da rede neural, baseadas na presença (=1) ou ausência (=0) dos atributos relacionados ao tipo de ligação (local, interurbana, internacional ou especial). Pode parecer mais lógico usar o valor +1 para indicar a presença de um atributo, e -1 para indicar sua ausência, respectivamente. Porém devido a normalização dos vetores de entrada, e sua comparação subsequente, os atributos valendo 0 possuem um efeito qualitativamente similar aos componentes negativos em uma comparação com base nas diferenças vetoriais [HON].

Dados Reais		Rede Neural			
Usuário	Nº Discado	Local	Interurbana	Internacional	Especial
João	2234415	1	0	0	0
Maria	0900213624	0	0	0	1
Pedro	0413352478	0	1	0	0
Ana	005412358745	0	0	1	0

Tabela 3 - Tratamento da variável Destino.

A variável Destino também foi discretizada em intervalos iguais que são passados para a RN onde cada tipo de ligação tem um valor diferente como mostra a Figura 10.

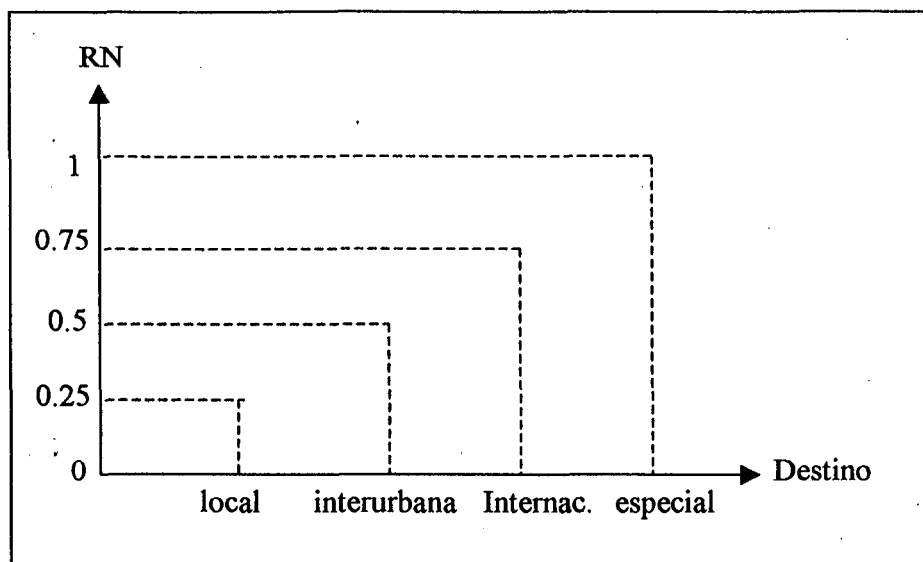


Figura 10: Esquema do pré processamento da variável Destino.

A RN pode ser testada com os dois tipos de pré processamento para uma melhor avaliação de seus resultados.

Data

Através da data no formato dd/mm/aaaa, pode-se determinar o dia da semana correspondente utilizando-se um calendário perpétuo. Deste modo a data corresponde a duas entradas na rede, sendo 1 o valor atribuído para a variável que representa o dia no conjunto de entrada, como está esquematizado na Figura 11.

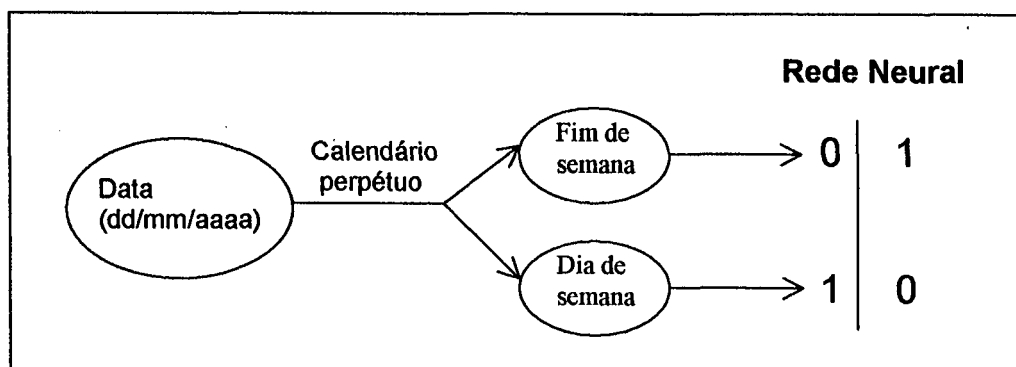


Figura 11 – Variável Data.

A relevância da data para o sistema consiste em saber se a ligação foi realizada em um dia ou final de semana. Isto é devido à alguns fatores que influem na mudança da utilização do telefone nestes dias, já que nos finais de semana as tarifas são mais baratas, a maioria das pessoas não trabalha, etc.

Hora

Para a variável Hora é feita uma transformação linear dos valores de modo que a maior (24:59h) e a menor hora (1:00) representem os valores 1 e 0 respectivamente. A Figura 12 apresenta esta transformação.

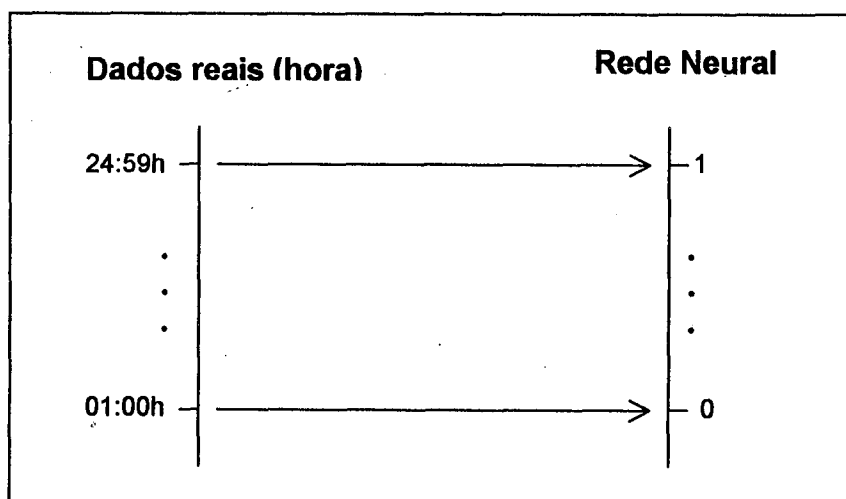


Figura 12 – Variável Hora.

Devido a variação de tarifação de acordo com o horário, é importante ter essa informação para treinar a RN.

Duração

A análise da duração da ligação é feita definindo um tempo considerado dentro da

normalidade das ligações efetuadas pela maioria das pessoas. Esta informação é obtida através da telefônica onde o sistema irá rodar. Se fosse feita simplesmente uma transformação linear dos valores reais para valores entre 0 e 1 usados como entrada da rede, o processo de agrupamento dos dados ficaria incorreto, pois não existe um valor pré definido para o maior tempo de ligação. Sendo assim, em um conjunto de entrada poderia haver uma ligação de dez horas, por exemplo, e em outro conjunto a maior ligação poderia ser de trinta minutos, sendo que o valor atribuído como entrada da rede seria 1 para ambos casos. Desta forma a rede estaria considerando informações completamente diferentes como semelhantes.

O maior tempo que um usuário gasta em uma ligação foi definido aqui como trinta minutos, estabelecendo assim um tempo de ligação limite considerado aceitável. Se a duração da ligação for igual ou maior que 30 minutos, a entrada da rede será igual a 1, como mostra a Figura 13.

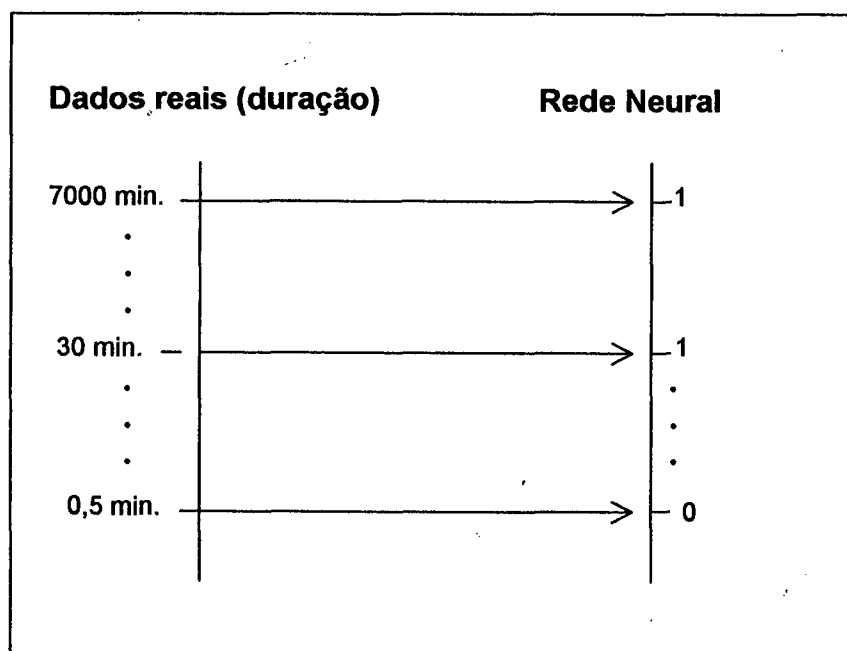


Figura 13 – Variável Duração.

A Figura 14 apresenta a representação gráfica da transformação da variável duração.

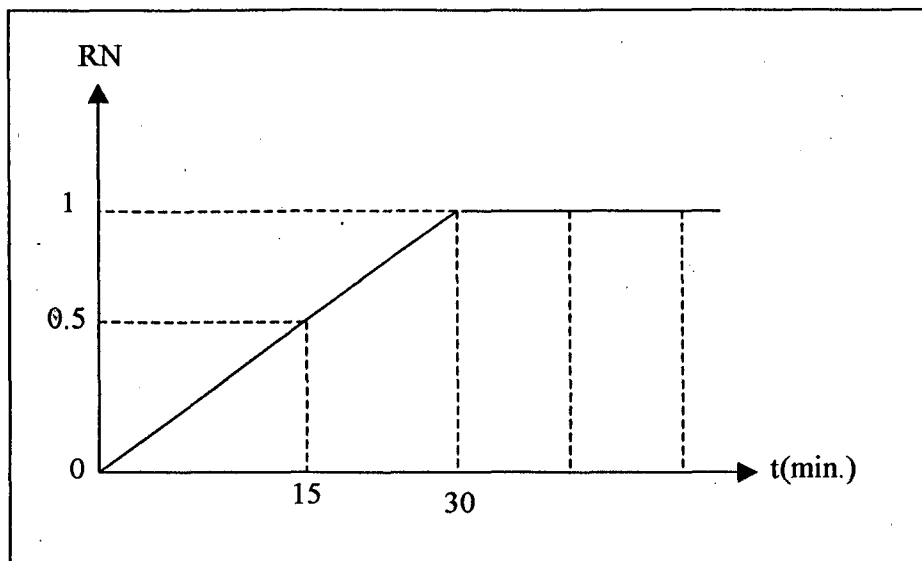


Figura 14 - Variável Hora graficamente representada.

A seleção das características é o primeiro passo para uma aplicação de classificação de padrão. A eficiência do classificador, independente de qual seja ele, é diretamente influenciada pelas características dos parâmetros escolhidos. Uma boa performance de classificação requer a seleção de características relevantes ao problemas em questão e também um classificador que faça um bom uso destas características – considerando dados de treinamento limitados e recursos computacionais.

Feito o pré processamento, os dados são importados no Matlab em forma de uma matriz de dimensões variáveis dependendo do tamanho do arquivo de dados usado e da natureza da variável considerada. Neste estágio, os dados estão preparados para serem submetidos ao treinamento da RN, como é mostrado em seguida (Item 5.1.5.1).

4.1.4 Amostra Representativa

Devido aos recursos computacionais limitados unido a grande massa de dados disponível não foi possível trabalhar com a amostra de dados completa. Assim, através das variáveis do sistema foi feita uma heurística, ou seja, um processo de análise empírico para se definir um arquivo de dados representativo, isto é, menor, mas mantendo as porcentagens de ocorrência de cada variável na amostra. Para isto, através de uma avaliação da frequência - que é mostrada graficamente a seguir - de ocorrência de cada variável na amostra de dados foi gerada uma amostra de tamanho menor. Os gráficos de avaliação da frequência foram feitos com a ferramenta estatística SPSS.

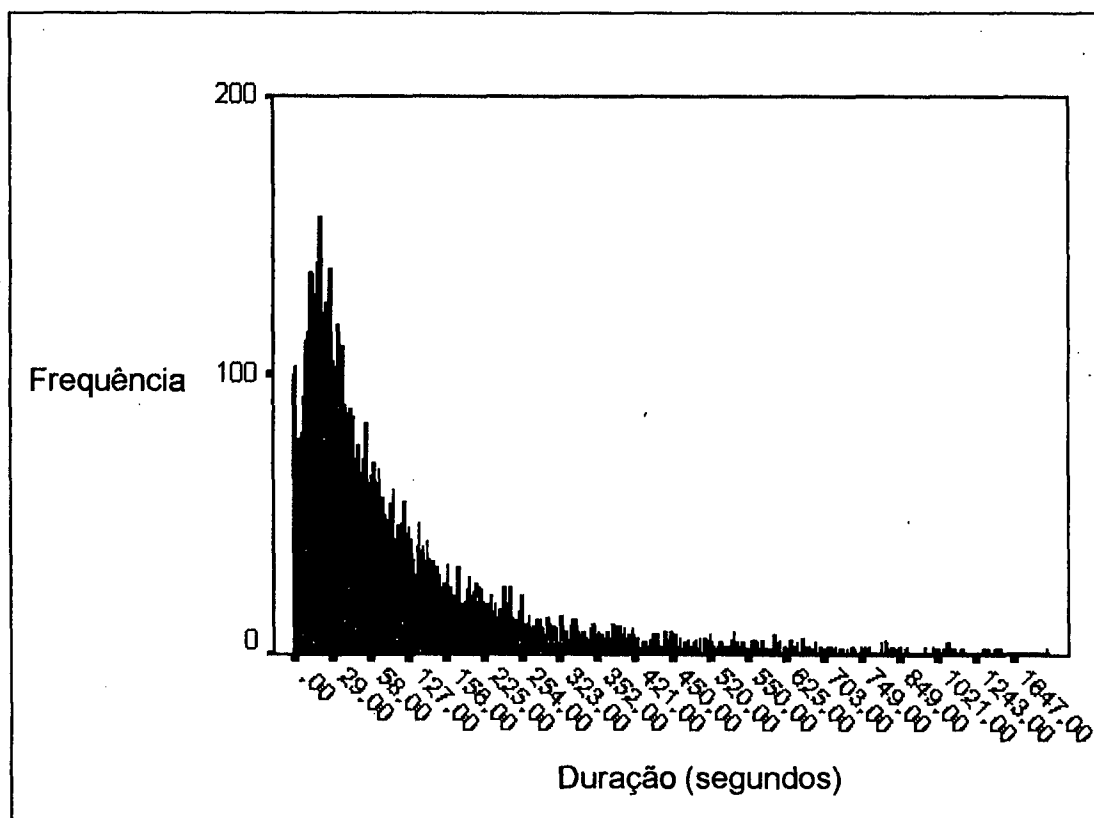


Figura 15: Frequencia de Duração na amostra.

A ocorrência da duração das ligações se concentra entre 0 e 30 minutos aproximadamente.

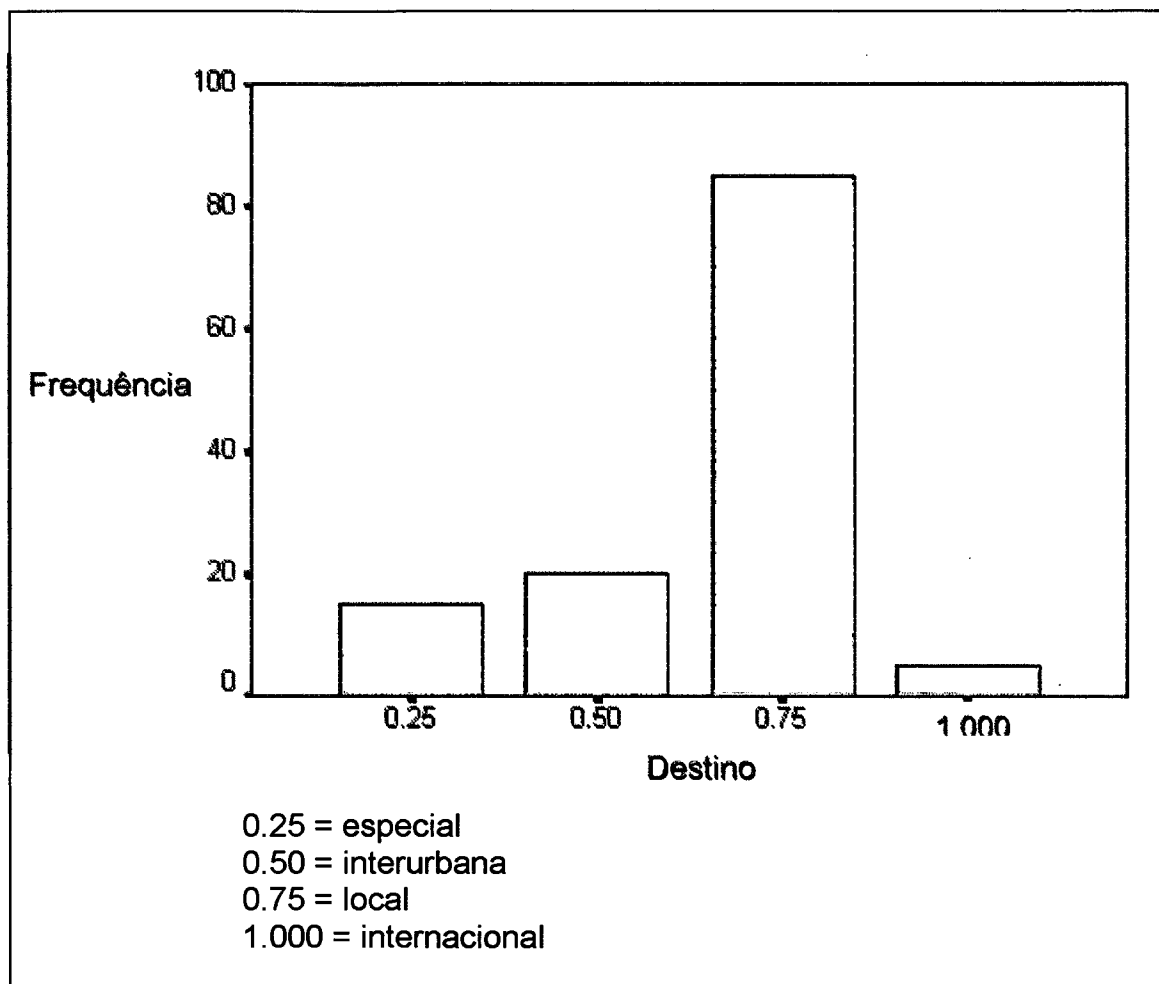


Figura 16: Frequência do Destino na amostra

Observa-se através da Figura 16 que as ligações locais atingiram uma frequência de ocorrência de mais que 80, e os outros tipos de ligações (especiais, interurbanas e internacionais) tiveram no máximo 20 na frequência de ocorrência.

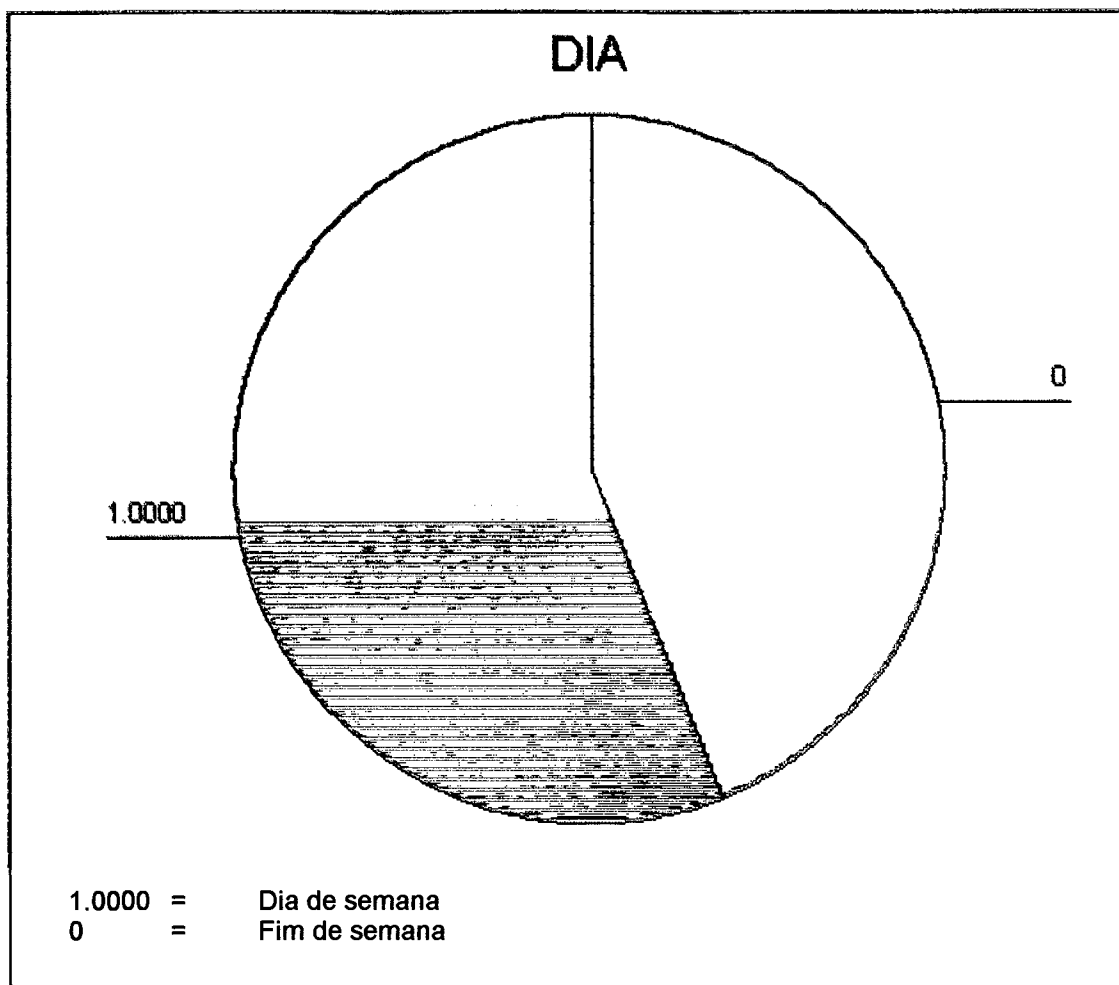


Figura 17: Frequência do Dia na amostra.

Pela Figura 17, mais da metade das ligações na amostra de dados foi realizada em dias de semana.

A partir destas informações, foi gerada uma amostra de dados menor, com 2400 registros de ligações. Foi feito, para esta nova amostra, o mesmo procedimento feito com a amostra grande de dados. Os gráficos gerados pelo SPSS para a nova amostra tiveram as mesmas frequências da amostra de dados original. Deste modo, pode-se dizer que a amostra de dados usada para o treinamento da RN é menor, porém representativa.

4.1.5 Kohonen

A escolha do algoritmo de Kohonen para realizar a auto-organização dos dados, gerando agrupamentos (ou grupos), foi feita considerando a capacidade de treinamento não supervisionado da rede, tornando o processo mais automatizado. Além disso a rede de Kohonen tem vantagens sobre técnicas clássicas de reconhecimento de padrões porque utiliza a arquitetura paralela das redes neurais. Este tipo de rede sempre tem duas camadas. Uma camada de entrada e uma camada de saída. A camada de entrada pode conter um ou mais neurônios dependendo da dimensão dos dados de entrada. Estes neurônios de entrada passam o sinal pela rede até os neurônios da camada de saída. Então, os fatores de pesos são aplicados.

4.1.5.1 Treinamento da Rede – Primeira fase

As camadas competitivas podem ser melhor compreendidas quando seus vetores de entradas são representados graficamente. O gráfico apresentado na Figura 18 mostra os vetores de entradas, compostos pelas variáveis escolhidas com informações sobre as ligações contidas na amostra de dados, representadas pelos pontos na esfera.

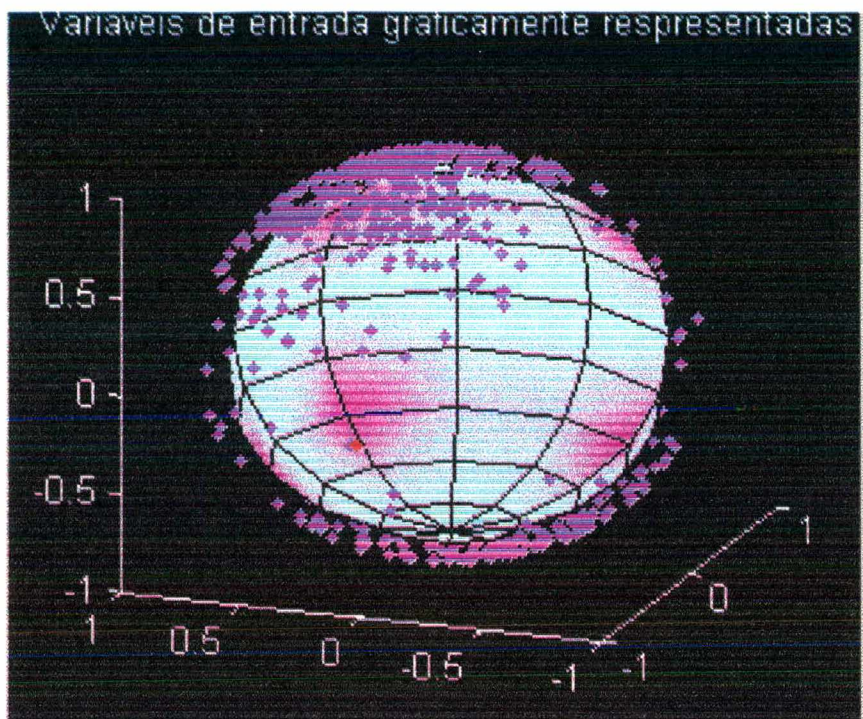


Figura 18: Vetores de entrada da RN graficamente representados

Os vetores de entrada mostrados na Figura 18 serão submetidos à rede neural competitiva, que irá usa-los para definir alguns grupos.

Em uma estrutura uni ou multidimensional dos neurônios da camada de saída, apenas um neurônio para cada conjunto de entrada é ativado quando um vetor de entrada é aplicado na rede. O neurônio ativado indica em qual categoria o vetor de entrada pertence. Os vetores de entrada que diferem pouco entre si fazem com que um mesmo neurônio na saída seja ativado. Isto significa que quando as ligações de um usuário apresentarem poucas modificações, a área do espaço ativada será a mesma.

A Figura 19 apresenta um esquema do funcionamento da rede.

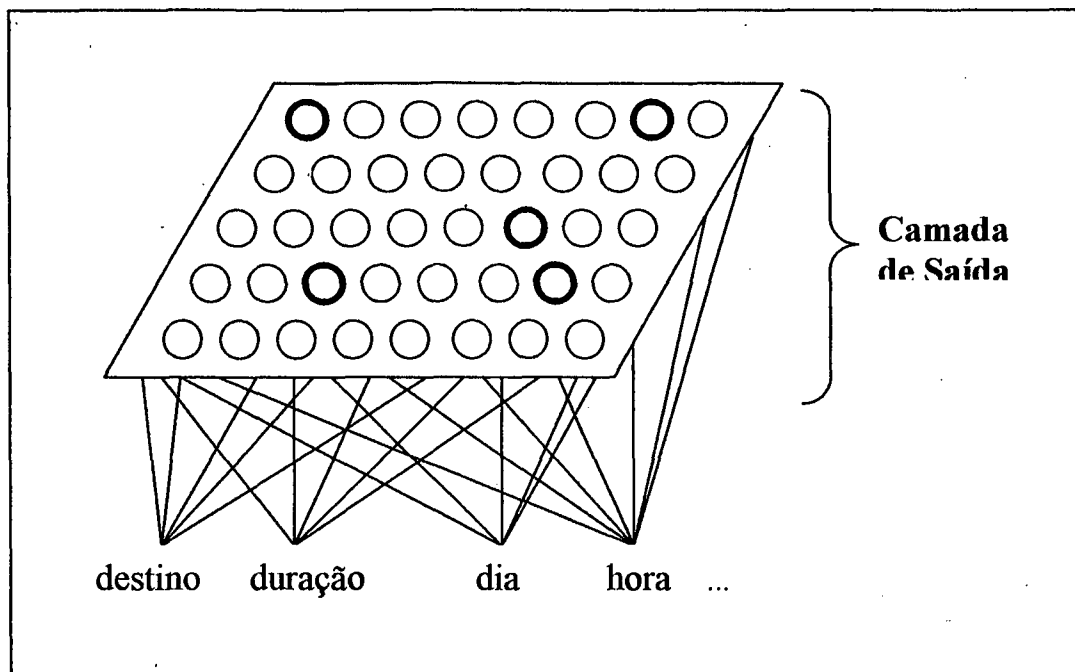


Figura 19 – Esquema do funcionamento da rede neural de Kohonen.

A ativação de apenas um neurônio ocorre devido à competição dos neurônios uns contra os outros. Apenas o neurônio com a menor diferença entre seus fatores de pesos e o vetor de entrada aplicado é ativado. Durante o treinamento da rede, os fatores de pesos dos neurônios são ajustados para reduzir esta diferença. Os fatores de pesos dos neurônios vizinhos também são ajustados para a medida de distância diminuir. Para os neurônios que estão muito distantes os fatores de pesos não são ajustados ou são ajustados de modo a aumentar a diferença de distância.

No começo do treinamento estes ajustes são feitos em uma grande área ao redor do neurônio ativado. Com o progresso do treinamento, esta área torna-se cada vez menor até que se restrinja à área do neurônio ativado. Os ajustes são também reduzidos em tamanho, isto é, após um tempo de treinamento, apenas os fatores de pesos do neurônio ativado são atualizados de forma limitada. Através destes pequenos ajustes, a rede segue mudanças sistemáticas menores a partir dos vetores de entrada.

4.1.5.1.1 Tecnologia utilizada

A forma mais utilizada de se realizar as implementações de RN são os *softwares*, que são programas de computador que implementam a operação das RN. A velocidade da implementação depende da velocidade do *hardware* no qual a RN é executada.

A ferramenta escolhida para desenvolver a RN foi o Matlab [MAT] por possuir uma implementação eficiente do algoritmo de Kohonen, tornando mais fácil estabelecer diferentes escolhas de projeto para o sistema.

4.1.5.1.2 Treinamento da RN com o algoritmo de Kohonen

Neste estágio pode-se rodar o algoritmo de Kohonen com os dados disponíveis. A ferramenta Matlab não foi capaz de treinar a rede usando um arquivo de dados maior do que 200Kb. A rede é então treinada com um arquivo de dados representativo, isto é, menor mas mantendo as porcentagens de ocorrência de cada variável na amostra.

O sucesso do treinamento da rede envolve muitas escolhas e experimentos de treinamento. Foram realizados alguns treinamentos variando as características da rede neural e das variáveis envolvidas. A seguir são apresentadas as redes projetadas, as variáveis usadas e os resultados obtidos para cada treinamento realizado (Tabela 4).

	Nun Iter	Nun Reg	W	Tx Ap.	Nu Var	Natureza da variável				Grupos
						Dest	Durac	Hora	Dia	
RN1	30	30	[1X4]	0.01	4	bin	cont	cont	bin	4
RN2	150	30	[1X4]	0.01	4	bin	cont	cont	bin	3
RN3	4800	2400	[1X4]	0.01	4	bin	cont	cont	bin	6
RN4	4800	2400	[1X4]	0.001	4	bin	cont	cont	bin	4
RN5	7200	2400	[1X10]	0.01	4	bin	cont	cont	bin	9
RN6	7200	2400	[1X10]	0.001	4	bin	cont	cont	bin	7
RN7	12000	2400	[1X4]	0.01	2	bin	cont	-	-	3
RN8	12000	2400	[1X4]	0.001	2	bin	cont	-	-	5
RN9	2400	2400	[1X7]	0.01	4	disc	disc	disc	Bin	4
RN10	2400	2400	[1X7]	0.001	4	disc	disc	disc	Bin	10
RN11	2400	2400	[1X10]	0.01	4	disc	disc	disc	Bin	8
RN12	2400	2400	[1X10]	0.001	4	disc	disc	disc	Bin	9
RN13	4800	2400	[1X7]	0.001	2	disc	disc	-	-	5
RN14	4800	2400	[1X7]	0.01	2	disc	disc	-	-	8
RN15	4700	4700	[1X4]	0.01	4	bin	cont	disc	bin	7
RN16	4700	4700	[1X4]	0.001	4	disc	disc	disc	bin	8
RN17	9400	2400	[1X7]	0.01	2	disc	disc	-	-	4
RN18	9400	2400	[1X7]	0.001	2	disc	disc	-	-	5
.										
.										
.										

Tabela 4: Experimentos de treinamento da RN.

As características apresentadas na Tabela 4 são: número de iterações, número de registros na amostra, dimensão da matriz de pesos, taxa de aprendizagem, número de variáveis, natureza das variáveis de entrada (binárias, contínuas, discretas) e número de grupos. Estas características determinam os critérios de treinamento da RN. Os valores destas características tem influência direta no comportamento da rede, pois a variação destes valores determina o número de grupos gerados na saída da RN. Deve-

se treinar a RN diversas vezes a fim de encontrar a configuração mais apropriada para o problema em questão. Isto é feito através de observação e testes. A seguir é apresentado o procedimento utilizado para a análise dos resultados (grupos) gerados pelo treinamento da RN.

4.1.5.1.3 Análise dos grupos gerados

No final do treinamento, tem-se definidos alguns grupos, que representam os tipos de usuário, isto é, um perfil de utilização do telefone celular. A performance pode ser comparada para diferentes redes projetadas com o objetivo de identificar a melhor rede para a aplicação em questão.

A rede escolhida para realização dos testes foi a RN14, de acordo com a Tabela 4, pois foi treinada com todas as quatro variáveis definidas e obteve na sua saída o número médio de grupos em relação as outras redes.

A saída da rede numericamente representada é uma matriz de dimensão igual ao do arquivo usado para treina-la e tem o seguinte formato:

Columns 1 through 12

```

0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0

```

Columns 13 through 24

```

0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0

```


Columns 97 through 108

0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0

Columns 109 through 120

0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 121 through 132

0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0

Columns 133 through 144

0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0

Columns 145 through 156

0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0

Columns 157 through 168

0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

⋮
⋮
⋮

Columns 2377 through 2388

1	1	1	1	0	1	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	1	0	0	0	1	1	1
0	0	0	0	0	0	1	1	1	0	0	0

Columns 2389 through 2400

```

1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0

```

Estas informações podem ser interpretadas levando-se em consideração as diferentes combinações na posição dos zeros e uns para cada vetor (cada coluna representa um vetor). Cada vetor com combinação diferente representa um grupo. Assim, pode-se notar que foram gerados 8 grupos diferentes. A priori estas informações não tem um significado rotulável, como por exemplo: o grupo 1 representa usuários que fazem muitas ligações locais. Não é conveniente a rotulação destes grupos pois não são conhecidas as “regras” usadas para sua criação. Mas a título de testes e para uma melhor compreensão dos resultados é possível obter-se um significado aproximado em termos do que cada um dos vetores representa. Para isto basta submeter a rede treinada vetores de ligações do mesmo tipo e observar a saída que a rede produz. Isto é, ao se submer um conjunto com ligações somente locais, de pouca duração, em dia comum e horário comercial, o grupo ativado na saída da rede poderá ser rotulado como “Locais de pouca duração”, por exemplo. Este procedimento foi realizado para facilitar a análise do comportamento da RN.

O próximo passo é definir para cada usuário um grupo usando a RN treinada. Esta fase é necessária para ser possível a realização dos testes de eficiência da rede.

4.1.5.2 Definição dos tipos de usuário - Segunda Fase

As informações de ligações de cada usuário, são retiradas de uma base de dados e depois de devidamente tratadas, são submetidas como entradas da rede neural. Estas informações não necessariamente foram usadas na fase de treinamento. A rede deve apontar apenas um grupo para cada conjunto de informação de cada usuário. Este

grupo define o tipo daquele usuário, que representa seu padrão de utilização do telefone. Esta informação é armazenada em uma base de dados (ver ítem 5.1.6).

Com a função *Hardlim* do Matlab - função de saída do algoritmo de Kohonen - é possível passar um conjunto de dados para saber os grupos da saída considerando a matriz de pesos atualizada pelo treinamento da rede. Usando esta função, foram definidos grupos para oito usuários com perfis de utilização do telefone diferentes. Para cada usuário foram feitos vários testes simulando as últimas ligações e verificando a saída gerada pela rede. Os testes realizados são descritos a seguir.

4.1.5.3 Controle/Testes - Terceira Fase

Seguindo a fase de treinamento e após definidos alguns tipos de usuários, tem-se a fase de teste da rede neural, que é semelhante à fase de treinamento, só que não são mais alterados os pesos dos neurônios ao ser submetida uma entrada de valores para a RN. A fase de testes visa avaliar a eficiência e performance da RN. Para que o teste seja mais apurado, a rede poderá receber, como entrada, além dos fatos que foram apresentados, outros que não fizeram parte do conjunto de treinamento. Reconhecendo as entradas do teste de forma adequada a rede é considerada treinada com sucesso.

Os dados de teste devem refletir os dados a serem encontrados quando a RN estiver em uso, e experimentos de testes devem usar o mais amplo espectro de entradas que for possível, incluindo entradas que resultem em circunstâncias não desejáveis. Redes neurais ainda não tem níveis de performance provados teoricamente; a performance deve ser testada [DAY].

As últimas chamadas telefônicas de um usuário são chamadas dados dinâmicos do sistema. Estes dados são usados em tempo real pela rede neural para verificar se o padrão de ligação do usuário continua igual, através da ativação de um tipo e comparação do tipo inicialmente definido pela rede.

Foram testados para os usuários definidos na segunda fase várias combinações de ligações de telefones clonados e não clonados. A seguir são apresentados os resultados dos testes realizados.

4.1.5.3.1 Falsos positivos e falsos negativos

Os falsos positivos são os resultados em que a RN detectou mudança de padrão quando a saída produzida pela rede para um determinado usuário foi diferente da anteriormente definida ao serem passados para rede dados de um telefone clonado. Para isto, foram simulados dados de ligação de fraudadores, ou seja, com muitas ligações internacionais, de longa duração, feitas todos os dias em todos os horários. Neste caso, a RN detectou a clonagem corretamente em 80% dos casos. Somente quando o tipo do usuário em consideração era parecido com o perfil de utilização do telefone do fraudador, a RN não distinguiu a diferença de tipo.

Os falsos negativos representam os resultados que a RN detectou clonagem onde, apesar de ter havido mudança de padrão nas ligações, não foram de fraudadores. Este tipo de teste não foi feito, já que o objetivo principal do sistema é a detecção fraudadores causadores de grandes prejuízos. Os falsos negativos no sistema não constituem uso ilegal do telefone. Desta forma, se a porcentagem de erro para os falsos negativos for alta, alarmes falsos são disparados para o Gerente. O Gerente é a entidade responsável por analisar os telefones considerados clonados e decidir se o alarme será disparado ao usuário final, dono do telefone em questão. Além da intervenção do Gerente, pode ser criada uma hierarquia no padrão das chamadas, onde dependendo do grau de variação, a mudança de padrão é ignorada, ou apenas registrada em um arquivo de *log* para posteriores análises.

4.1.6 Base de Dados

A aquisição de conhecimento das redes neurais pode ser feita automaticamente a partir de exemplos coletados em uma base de dados [JOR].

A base de dados precisa conter as informações de cada usuário, além de suas últimas chamadas telefônicas e o tipo a que ele pertence. A Figura 20 apresenta a estrutura da base de dados do sistema.

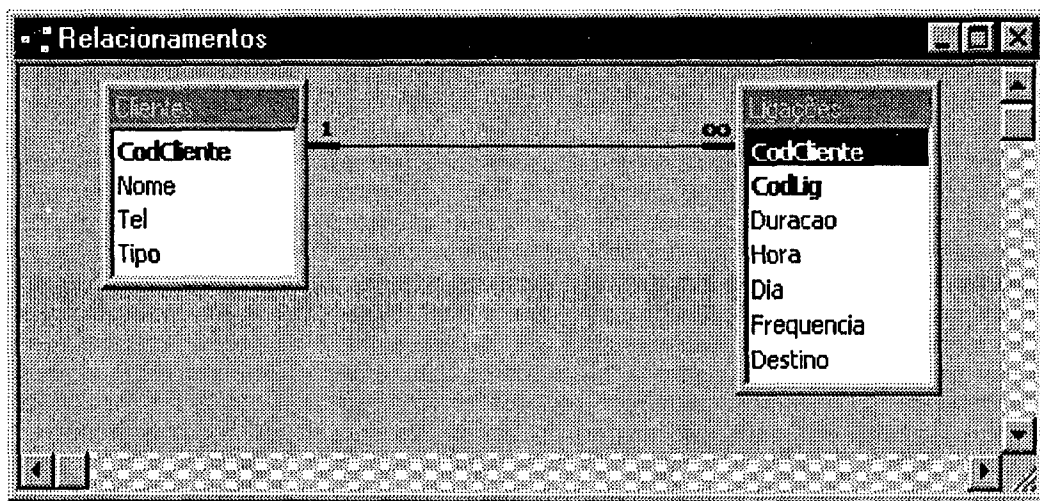


Figura 20 - Estrutura da base de dados.

4.1.6.1 Comparação

Os registros da base de dados são atualizado constantemente, a cada ligação realizada e são usados como entradas da rede neural ao ser invocada. A saída produzida pela rede é comparada com o campo da base de dados que contém o tipo do usuário. Mudanças no tipo podem indicar intrusão. A comparação do tipo do usuário gravado na BD com o tipo gerado pela RN é realizado pelo SSCC (Sistema de Segurança contra Clonagem de Celulares) [HER], que é um dos módulos que fazem parte do SSTCC.

4.1.7 Integração dos módulos DEITREN e SSCC do SSTCC

O SSCC é composto por dois módulos: agente e adaptador. O DEITREN está diretamente inserido no contexto do módulo Agente do SSCC. O SSCC está sendo implementado usando-se CORBA como tecnologia de objetos distribuídos, e a linguagem de programação JAVA para uma completa independência de plataforma [HER].

O modelo do SSTCC completo é mostrado na Figura 21.

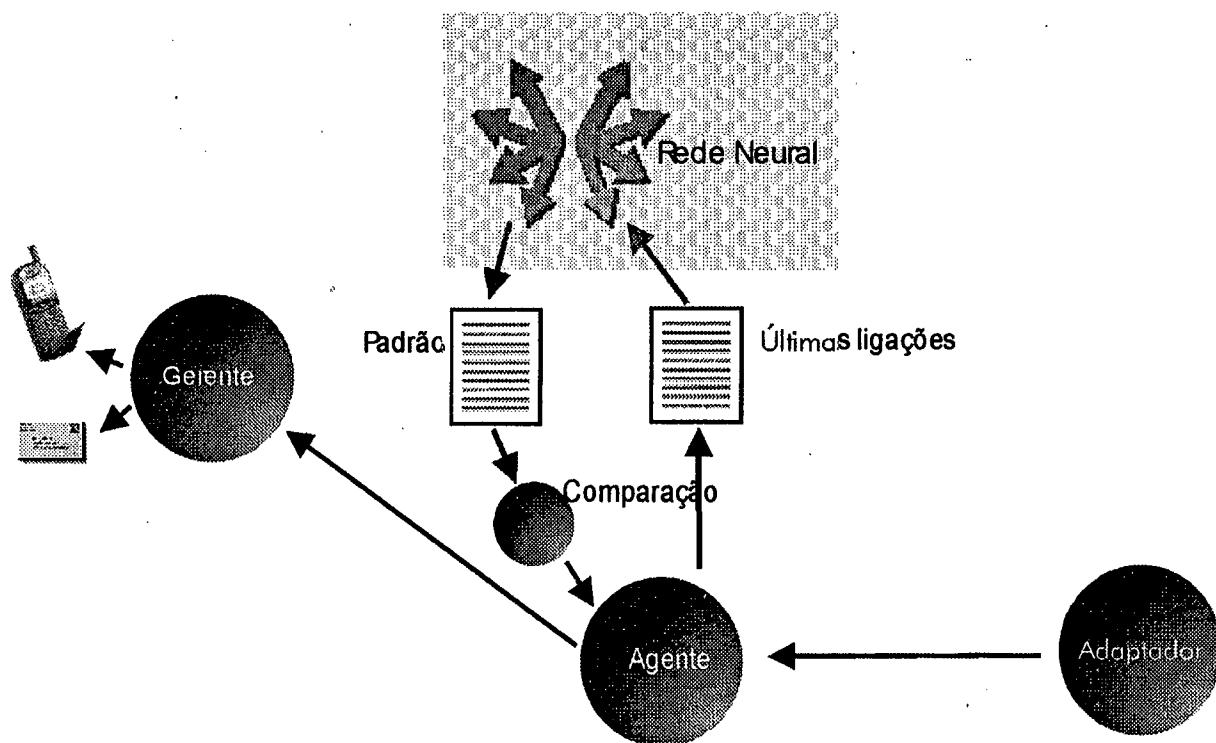


Figura 21: Modelo do SSTCC

4.1.7.1 O funcionamento do Agente

As finalidades básicas do Agente são:

- Ler o número determinado de ligações do banco de dados e excluir os n registros lidos;
- Preparar os dados a serem enviados para a rede neural;
- DEITREN;
- Receber o padrão calculado pela rede neural;
- Comparar o padrão calculado com o padrão previamente determinado para o usuário; e
- Enviar o alarme ao Gerente caso constate fraude de clonagem.

Como a implementação do DEITREN foi feita no Matlab, é necessário traduzi-la para uma linguagem de programação. Através de um programa chamado MCC que está incluso no pacote do Matlab oficial, pode-se traduzir o código em Matlab para uma linguagem de programação como C. Na fase de verificação, a RN será uma função que é invocada pelo código do SSCC passando como parâmetros uma variável com as dez últimas ligações do usuário e os pesos da rede atualizados pelo seu treinamento (fase 1). A rede processa as informações e retorna o padrão do usuário. O esquema do funcionamento da chamada da função da RN é mostrado na Figura 22.

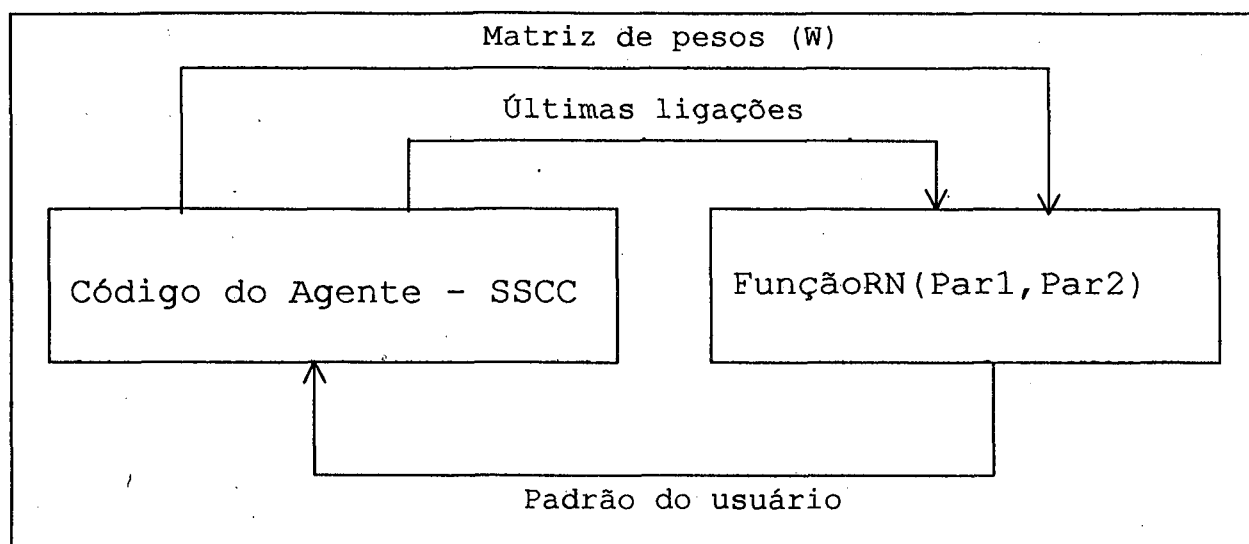


Figura 22: Esquema da chamada de função da RN pelo módulo SSCC.

Com o resultado do retorno da função da RN o SSCC pode verificar, através da comparação na base de dados, se o usuário mudou de padrão. Caso uma mudança de padrão seja verificada, o módulo Agente se encarrega de disparar um alarme de aviso.

Conclusão

A implementação do DEITREN teve várias fases e muitas dificuldades foram encontradas durante seu desenvolvimento. A obtenção de informações foi uma das principais dificuldades na concepção do sistema, pois se tratavam de assuntos muito específicos, porém fundamentais para o sucesso dos resultados e nos quais poucas pessoas tinham conhecimento. A dificuldade no acesso a informação tornou o processo de implementação mais lento.

5 Conclusões Finais e Perspectivas Futuras

A finalidade de testar um método de agrupamento não supervisionado neste trabalho consiste no fato de que em futuras análises pode-se modificar as variáveis de modo que hajam correlações não facilmente identificáveis ao homem. Este tipo de procedimento exploratório é útil para se compreender a natureza complexa das relações multivariadas. Na rede de Kohonen, os grupos são o resultado da relação natural entre os padrões dados a rede e podem fornecer hipóteses, que muitas vezes o especialista do BD (quando existe um) não consegue identificar.

A escolha do processo de agrupamento com o algoritmo de Kohonen foi feita considerando-se a definição e possível modificação das variáveis relevantes no escopo desta pesquisa. Como exemplo, pode-se refinar a variável destino de modo a dividir o território nacional em áreas (Sul, Norte, Sudeste, Nordeste, etc) ou até mesmo em estados através do DDD discado. Além disso, pode existir uma variável com a informação do tipo de uma ligação, isto é, se foi feita a cobrar ou não. Esta informação é importante já que uma pessoa (clonador ou inadimplente) que não precisa se preocupar com o preço da chamada - já que não irá pagá-la - não faria ligações a cobrar. Há possibilidade dessas variáveis possuírem uma relação, como por exemplo, as pessoas que moram no Acre, por exemplo, tem uma média de ligações a cobrar muito maior que o restante dos estados. Existem infinitas hipóteses que podem ser formuladas pelo homem a este respeito, mas esta relação é mais facilmente "perceptível" por uma rede neural como a de Kohonen.

A RN treinada com uma determinada amostra de dados não deve ser generalizada para funcionar em outro lugar que não seja na empresa de onde ela foi retirada. Dependendo de onde a empresa se localiza, haverá diferentes características de utilização do telefone. Por exemplo, pessoas que moram nas fronteiras do Brasil fazem mais ligações internacionais do que a média da população.

Considerando-se a necessidade do sistema de processamento em tempo real, as redes neurais tem vantagem sobre os algoritmos tradicionais devido ao paralelismo massivo de suas implementações, pois seu poder de processamento é medido pelo número de atualizações das interconexões por segundo. Em contraste, o poder de performance dos sistemas tradicionais é caracterizado pelo número de instruções que podem ser executadas por segundo, em sequência, através de um único processador. Desta forma, a limitação de recursos computacionais, influi diretamente na performance do sistema.

Ao final dos experimentos, na fase de testes do sistema, a rede neural é capaz de detectar mudanças de tipo de ligação dos usuários quando necessário, ou seja, quando a simulação de fraudadores com grandes mudanças nas ligações realizadas foi feita. Os resultados dependem de testes - pois não tem níveis de performance provados teoricamente - e variam muito. A variação dos resultados da RN ocorre devido a vários fatores:

✧ A representatibilidade da amostra de dados que é influenciada pelo período de tempo de coleta. Quanto maior for o tempo de coleta, mais a RN “aprende”. Foi usada uma amostra representativa, extraída de um arquivo contendo ligações coletadas durante uma semana, uma quantidade considerada suficiente para o objetivo do sistema proposto.

✧ A quantidade e natureza das variáveis definidas. Como testado, a RN se comportou de forma diferente quando treinada com 2 e 4 variáveis. Com 4 variáveis a RN formou mais grupos ao final do treinamento, tornando o processo de detecção de fraude mais sensível, ou seja, a RN foi capaz de detectar mudanças no perfil de ligação do usuário com mais precisão.

✧ O tratamento das variáveis. Por exemplo, na duração da ligação o tempo máximo considerado normal em uma ligação foi definido como sendo quatro horas. Este tipo de informação provém do diálogo com o especialista que trabalha na telefônica. A comunicação com o especialista no sentido de obtenção de informação é uma fase

importante para implementação do sistema. Porém, como foi o caso desta pesquisa, nem sempre o próprio especialista é capaz de fornecer a informação precisa e completa, ocasionando grandes dificuldades ao implementador.

✧ Sub ou superagrupamento. No caso de subagrupamento, a rede neural criou poucos grupos. Assim, quando um conjunto de entradas que necessite uma classificação diferente das que a rede produziu na fase de treinamento, é apresentado para a rede, o grupo ativado não será correto para aquele conjunto. O sub ou superagrupamento varia de acordo com os critérios de treinamento da RN. Quando treinada com 2 variáveis, por exemplo, a RN produziu um subagrupamento com apenas 2 grupos, o que causou ineficiência na detecção de fraude, já que mesmo com grandes mudanças, a RN continuou ativando o mesmo grupo na saída. Ao contrário, o superagrupamento produz grupos além do necessário, tornando a rede “sensível” às mudanças de tipos de ligação. Assim, a rede poderia detectar mudança de tipo, quando isto não fosse o desejado. Se um usuário mudasse apenas um pouco seu perfil fazendo por exemplo, ligações com duração um pouco acima do seu normal, a rede poderia classificá-lo em um grupo diferente do seu. Este tipo de problema não ocorreu na implementação do DEINTREN.

✧ A dimensão do vetor de pesos (W) é uma importante característica ao se projetar a rede, pois quanto menor sua dimensão, mais limitada a possibilidade do número de grupos a serem criados. Foram testadas diversas dimensões para o vetor de pesos: [1X4], [1X7] e [1X10]. A RN com o W de dimensão [1X7] foi suficiente para o sistema proposto por permitir a criação de mais de 10 grupos.

Devido as características de agrupamento não supervisionado da RN usada nesta pesquisa, a porcentagem de acertos do sistema não pode ser calculada numericamente tendo assim precisão indefinida. Porém, pelo fato de nos testes simulados para detecção de grandes fraudadores - com 100% de mudanças no perfil de ligação - a RN ter ativado corretamente um grupo diferente, sua eficiência foi considerada aceitável. Apesar da possibilidade de alguns alarmes falsos serem disparados, entre o usuário final e a RN existe a intervenção do Gerente[HER] do sistema que tem a função de

analisar cada caso para tomar uma decisão. A pesquisa realizada resultou em uma solução para o problema em questão, já que foi possível a detecção de grandes mudanças no perfil de utilização do telefone celular, permitindo a tomada de alguma providencia por parte da empresa antes de haverem grandes prejuízos. Além disto, a estrutura criada para o funcionamento do sistema (adequação das variáveis de entrada da rede neural, base de dados, comunicação entre os ambientes de implementação), pode ser aproveitada para posteriores testes no intuito de conseguir maior segurança para as empresas de telefonia móvel.

Como perspectivas futuras para esta pesquisa, pode-se usar uma RN de função de base radial, por exemplo, devido sua característica de aprendizado supervisionado. Assim, os grupos na saída são previamente definidos e pode-se fazer uma avaliação mais precisa em relação a eficiência do sistema através do erro gerado no treinamento da rede. A precisão dos resultados é possível já que pode-se determinar um valor numérico na fase de detecção de fraude através da comparação dos resultados com dados de treinamento e de testes. Os resultados gerados pelo treinamento da RN (grupos) com aprendizado não supervisionado podem ser utilizados na definição dos grupos da saída da RN com aprendizado supervisionado.

Além disso, outras abordagens para o desenvolvimento da RN e do sistema devem ser feitas para analisar seu comportamento. Pode-se por exemplo considerar dimensões maiores para os vetores de entrada, com conjuntos de ligações ao invés de vetores com uma ligação.

Na fase de monitoramento e detecção de fraude do sistema, pode ser criada uma hierarquia no padrão das chamadas, onde dependendo do grau de variação, a mudança de padrão é ignorada, ou apenas registrada em um arquivo de *log* para posteriores análises. Isto minimizaria o trabalho do Gerente é a entidade responsável por analisar os telefones considerados clonados e decidir se o alarme será disparado ao usuário final, dono do telefone em questão. Também existe a possibilidade de usar a RN apenas para gerar os padrões de usuários e usar técnicas estatísticas para realizar o controle de intrusão.

6. Anexos

```
*****
*****
*****
% Pré processamento da variável hora de forma contínua.
% Transforma os valores entre 0 e 1 linearmente.
*****
*****
*****
```

```
function [resposta]=horalinear(hora)
```

```
S=mod(hora,100);
M=mod(floor(hora/100),100);
H=mod(floor(hora/10000),100);
resposta=datenum(0,0,0,H,M,S);
```

```
*****
*****
*****
% Pré processamento da variável Duração de forma
% contínua. Transforma os valores linearmente entre 0 e 1
% considerando o valor máximo de duração possível = 4h.
*****
*****
*****
```

```
function [resposta] = duracao (var2)
```

```
x=var2;
```

```
maxx=340;
```

```
minx=0;
```

```
num_x=size(x,1);
```

```
yy=x;
```

```
for i=1:num_x
```

```
    if x(i) >= maxx
```

```
        yy(i) = 1;
```

```
    end
```

```
end
```

```
for i=1:num_x
```

```
    if yy(i) <= minx & yy(i) ~= 1
```

```
        yy(i) = 0;
```

```
    end
```

```
end
```

```
a=(1/(maxx - minx));
b=(-minx/(maxx-minx));
```

```
for i=1:num_x
    if yy(i) ~= 1 & yy(i) ~= 0
        yy(i)=(a*yy(i))+b;
    end
end
```

```
resposta = yy;
```

```
*****
*****
*****
% Pré processamento da variável Destino de forma binária.
% Atribui o valor 1 em um vetor com 4 elementos na
% posição relativa ao tipo de ligação.
*****
*****
*****
```

```
function [retorno] = destino(fone)
%fone=te(:,1);
```

```
n_fone=size(fone,1);
```

```
tipo_ligacao=zeros(n_fone,10);
for i=1:n_fone
    %local
    if fone(i) < 9999999 & fone(i) > 99999
        tipo_ligacao(i,4)=1;
    end
    %especial
    if fone(i) < 99999
        tipo_ligacao(i,2)=1;
    end
    %interurbana
    if fone(i) > 9999999 & fone(i) < 99999999999
        tipo_ligacao(i,3)=1;
    end
    %internacional
    if fone(i) > 99999999999
        tipo_ligacao(i,1)=1;
    end
end
```

```
end
```

```

tipo_ligacao=tipo_ligacao(:,1:4);
retorno = tipo_ligacao;

```

```

*****
*****
*****
% Realiza o pré processamento da variável Data
% atribuindo 0 para dias de semana e 1 para fins de
% semana
*****
*****
*****

```

```

function [retorno] = dia(D)

```

```

D/10000;
a=mod(D,100);
m=mod(floor(D/100),100);
d=mod(floor(D/10000),100);
    %S=datenum(a+1900,m,d,0,0,0);
    S=datenum(a,m,d,0,0,0);
    W = weekday(S);

```

```

    n=size(W,1)
for i=1:n
    if W(i) == 1 | W(i) == 7
        W(i) = 1;
    else
        W(i) = 0;
    end
end
end

```

```

retorno=W;

```

```

*****
*****
*****
% Realiza o pré processamento da variável Hora
% discretizando os valores de acordo com a tarifação para
% intervalos iguais: (08:01h-12:00h)=0.2, (12:01h-
% 14:00h)=0.4, (14:01h-18:00h)=0.6, (18:01h-00:00h)=0.8,
% (00:01h-08:00h)=1
*****
*****
*****

```

```
function [resposta]=horadisc(horad)
```

```

S=mod(horad,100);
M=mod(floor(horad/100),100);
H=mod(floor(horad/10000),100);
R = datenum(0,0,0,H,M,S);

J = R;
hora_x=size(J,1);

f = J;
for i=1:hora_x

    if J(i) > 0 & J(i) <= 0.3333
        f(i) = 0.20;
    end

    if J(i) > 0.3333 & J(i) <= 0.5000
        f(i) = 0.40;
    end

    if J(i) > 0.5000 & J(i) <= 0.5833
        f(i) = 0.60;
    end

    if J(i) > 0.5833 & J(i) <= 0.7500
        f(i) = 0.80;
    end

    if J(i) > 0.7500 & J(i) <= 1
        f(i) = 1;
    end
end

resposta = f;

```

```

*****
*****
*****
% Realiza o pré processamento da variável Duração
% discretizando os valores em intervalos iguais,
% considerando o limite máximo de duração = 6h:
% (0.1'-13'59")= 0.16, (14'-28'.59")= 0.32, (29'-
% 58'.59")= 0.48, (59'-119'59")= 0.64, (120'-358'59")=
% 0.80, (>=359')= 0.96
*****
*****
*****

```

```
function [resposta] = duradisc (var2)
```

```
x=var2;
```

```
minx=0;
```

```
num_x=size(x,1);
```

```
dic=x;
```

```
for i=1:num_x
```

```
    if x(i) > minx & x(i) <= 14
```

```
        dic(i) = 0.16;
```

```
    end
```

```
    if x(i) > 14 & x(i) <= 29
```

```
        dic(i) = 0.32;
```

```
    end
```

```
    if x(i) > 29 & x(i) <= 59
```

```
        dic(i) = 0.48;
```

```
    end
```

```
    if x(i) > 59 & x(i) <= 119
```

```
        dic(i) = 0.64;
```

```
    end
```

```
    if x(i) > 119 & x(i) <= 359
```

```
        dic(i) = 0.80;
```

```
    end
```

```
    if x(i) > 359
```

```
        dic(i) = 0.96;
```

```
    end
```

```
end
```

```
resposta = dic;
```



```

*****
*****
*****
% Realiza o pré processamento da variável Destino,
% discretizando os valores de acordo com o tipo da
% ligação em intervalos iguais: especial=0.25,
% interurbana=0.5, local=0.75 e internacional=1
*****
*****
*****

function [retorno] = destino(fone)
%fone=te(:,1);

n_fone=size(fone,1);

tipo_ligacao=zeros(n_fone,10);
for i=1:n_fone
    %especial
    %if fone(i) < 9999999 & fone(i) > 800000000 & fone(i) <
800999999
    if fone(i) < 9999999
        tipo_ligacao(i)=0.25;
    end
    %especial
    if fone(i) > 90099999 & fone(i) < 900999999
        tipo_ligacao(i)=0.5;
    end
    %interurbana
    if fone(i) > 99999999 & fone(i) > 900000000 & fone(i)
< 900999999
        tipo_ligacao(i)=0.75;
    end
    %internacional
    if fone(i) > 9999999999
        tipo_ligacao(i)=1;
    end
end

tipo_ligacao=tipo_ligacao(:,1:1);
retorno = tipo_ligacao;

```

```

*****
*****
*****
% Algoritmo de Kohonen. Usado para treinamento da RN.
*****
*****
*****

```

```
echo off
```

```

% Define a matriz de entrada pré processada para
% treinar a RN.
R = matriz;
R = R';
V1 = R; %atribui a V1 a matriz com dados reais R.
Q = 240;

```

```
clc
```

```

% a matriz de pesos neste caso é inicializada
% randomicamente
% com dimensão = [6X4].
W = randnr(6,4);
M = W;

% o bias
b = -0.1*ones(6,1);

A = hardlim(W*V1,b) %mudei

```

```
pause
```

```
clc
```

```

% a rede é treinada para cada vetor de entrada da amostra
% de dados
% com a função LEARNK, a regra de aprendizado do
% algoritmo de Kohonen.

```

```
echo off;
```

```
% Taxa de aprendizado:
```

```
lr = 0.01;
```

```
for ind=1:3 % indica o número de vezes que a amostra é
```

```
apresentada p/ treinar a RN.
  for q = 1:Q
    p = V1(:,q);           % Dimensão do vetor de
    entrada.               % Calcula a saída.
    a = hardlim(W*p,b);    % Aplica a regra de
    dW = learnk(W,p,a,lr); % Atualiza matriz de
    W = W + dW;           pesos.

  end
end

W = normr(W);

echo off

disp('Fim do treinamento')
```

```

*****
*****
*****
% Importa arquivo da amostra de dados para Matlab.
% Chama funcoes de entrada da rn.
% Invoca a RN de Kohonen
*****
*****
*****

% abre arquivo da amostra de dados.

fid = fopen ('amostra', 'r');
F = fread(fid);
s = setstr (F');
te=str2num(s);
fclose(fid);

% chama funcoes de pré processamento das variáveis de
% entrada da rn: hora, duracao, dia, destino.

te;
%var1 -> hora
var1 = te(:,2);
%var2 -> duracao
var2 = te(:,3);
%var3 -> dia
var3 = te(:,4);
%var4 -> destino
var4 = te(:,1);

matri      =      [(duracao3(var2)),      (horalinear(var1)),
(dia(var3)), (destino(var4))];

% invoca a RN de Kohonen

kohonen
*****
*****
*****

```

7 Bibliografia

- [BAR] BARRETO, J. M. **Introdução às Redes Neurais Artificiais**, Anais da V Escola Regional de Informática da SBC Região Sul. Florianópolis (SC), maio de 1997.
- [MIR] BOUKERCHE, A.; NOTARE, M.S.M.A. **Neural Fraud Detection in Mobile Phone Operations**. 14th IPPS - International Parallel Processing Symposium & 11th SPDP - Symposium on Parallel & Distributed Processing / Third Workshop on Biologically Inspired Solutions to Parallel Processing Problems (BioSP3). Cancun, May2000.
- [BIS] BISHOP, C.M. **Neural Networks for Pattern Recognition**. Oxford: Oxford University Press, 1995.
- [DAY] DAYHOFF, J. **Neural Network Architectures – An Introduction**. Ed. Van Nostrand Reinhold, 1990.
- [DEM] DEMUTH, H.; BEALE, M. **Neural Network Tollbox – For Use With Matlab**. User's Guide, Version 3.0, 1998. Pg. 7.1-7.33.
- [DOD] DODD, A. Z. **The Essencial Guide to Telecommunications**. Prentice Hall, New Jersey, 1997. Pg. 3-8.
- [DUD] DUDA, R. O.; HART, P.E. **Pattern Classification and Scene Analysis**. Stanford Research Institute, Menlo Park, California. Pg. 211-235.
- [GEL] GELSEMA, E. S.; KANAL, L. N. **Pattern Recognition and Artificial Intelligence**. Machine Intelligence and Pattern Recognition 7. North - Holland. Pg. 395.
- [HER] HERMIDA, A.C.; VALLE, W. **Implementação JAVA dos módulos agentes e adaptador do SSTCC – Sistema de Segurança para Telecomunicações Contra Clonagem e Inadimplência**. Relatório de Projetos, UFSC, 1999. Pg. 49.

- [HON] KOHONEN, T.; RITTER, H. **Self-Organizing Semantic Maps**. Biological Cybernetics, Spring-Verlag, 1989. Pg. 241-254.
- [ING] INGBER, L. **Simulated Annealing: Practive Versus Theory**. Mathl. Comput. Modelling, Vol. 18, Nr. 11, 1993. Pg. 29-57.
- [JOH] JOHNSON, R.A.; WICHERN, D. W. **Applied Multivariate Statistical Analysis**. University of Wisconsin – Madison, Prentice-Hall, New Jersey. Pg. 532-570.
- [JOR] JORDAM, M.I.; BISHOP, C.M. **Neural Networks**. Massachusetts Institute of Technology, 1996. Pg. 2.
- [JUD] DAYHOFF, J. **Neural Network Architectures – An Introduction**. Van Nostrand Reinhold, NY, 1990. Pg. 163-191.
- [KAN] KANGAS, J. **On the Analysis of Pattern Sequences by Self-Organizing Maps**. Helsinki University of Technology, Laboratory of Computer and Information Science, Finlandia. Tede de Doutorado em Tecnologia, 1994. Pg. 7-29.
- [KOH1] Kohonen, T. *Self-organized formation of topologically correct feature maps*, **Biological Cybernetics** 43, 1982. Pg. 59-69.
- [KON] KOHONEN, T.; TORKKOLA, K.; SHOZAKAI, M.; KANGAS, J.; VENTA, O. **Phonetic typewriter for Finnish and Japanese**. In Proceedings of the IEEE 1988 Internacional Conference on Acoustics, Speech and Signal Processing, Nova York, N.Y., 11-14 de Abril, 1988. Pg. 607-610.

- [KOV] Kovács, Z.L. **Redes Neurais Artificiais**. Edição Acadêmica. Pg. 12-20.
- [LAB] **The Math Works Inc. Matlab – The Problem-Solving Tool for Engineers, Mathematicians, and Scientists**. Prentice Hall, Englewood Cliffs, NJ, 1992.
- [LIP] LIPPMANN, R.P. **Pattern Classification Using Neural Networks**. IEEE Communications Magazine, Nov1989. Pg. 47-64.
- [MAT] THE MATH WORKS INC. **The Student Edition of Matlab**. Prentice-Hall Inc., New Jersey, 1992.
- [MEY] Meyer, J. W. **Self-Organizing Processes**. CONPAR 94 – VAPP VI, Lecture Notes in Computer Science 824, Springer-Verlag, Berlin, 1994. Pg. 842-853.
- [NOT] NOTARE, M.S.M.A.; CRUZ, F.A.S.; RISO, B.G.; WESTPHALL, C.B. **Wireless Communications: Security Management for Cloned Cellular Phones**. IEEE WCNC'99 – IEEE Wireless Communications and Networking Conference. New Orleans, LA, USA, 21-24/09/99.
- [RIT] RITTER, H.; SCHULTEN, K. **Neuronale Netze**. Addison-Wesley, Bonn, 1990.
- [TOD] TODESCO, J.L. **Reconhecimento de Padrões usando Rede Neuronal Artificial com uma Função de Base Radial: uma Aplicação na Classificação de Cromossomos Humanos**. Tese de mestrado - Pós-Graduação em Engenharia de Produção, UFSC, Brasil, 1995.
- [VEJ] Revista Veja. **Telefonia – Segredo frágil**. São Paulo, 1998. Pg. 81.
- [ELV] VIEIRA, E. M. **Método para Desenvolver Agentes Adaptativos na Gerência de Redes Usando Redes Neurais**. Tese de Mestrado - Pós-Graduação em Ciência da Computação, UFSC, Brasil, 1997. Pg.25-27.