UFPA

Universidade Federal do Pará

Ramon Villar Monte Palma Pantoja

Reconhecimento de Padrões de Ruído em redes VDSL2 usando Máquinas de Vetor de Suporte

UNIVERSIDADE FEDERAL DO PARÁ

INSTITUTO DE TECNOLOGIA

FACULDADE DE ENGENHARIA ELÉTRICA

BELÉM – PARÁ

2º semestre - 2011

Ramon Villar Monte Palma Pantoja

Reconhecimento de Padrões de Ruído em Redes VDSL2 usando Máquinas de Vetor de Suporte

Trabalho submetido ao Colegiado do Curso de Engenharia Elétrica, do Instituto de Tecnologia da Universidade Federal do Pará (FEE – ITEC – UFPA), para obtenção do grau de Engenheiro Eletricista.

Orientador: Prof. Dr. João Crisóstomo Weyl Albuquerque Costa

Co-orientadora: Prof[a]. Dra. Valquíria Gusmão Macedo.

Belém - PA

2011

Ramon Villar Monte Palma Pantoja

Reconhecimento de Padrões de Ruído em Redes VDSL2 usando Máquinas de Vetor de Suporte

Trabalho submetido ao Colegiado do Curso de Engenharia Elétrica, do Instituto de Tecnologia da Universidade Federal do Pará (FEE – ITEC – UFPA), para obtenção do grau de Engenheiro Eletricista.

Este trabalho foi julgado em ___/___/_____ adequado para obtenção do Grau de Engenheiro Eletricista e aprovado na sua forma final pela banca examinadora que atribuiu o conceito **_____**.

_____
Prof. Dr. João Crisóstomo Weyl Albuquerque Costa
ORIENTADOR

_____
Prof[a]. Dra. Valquíria Gusmão Macedo
CO - ORIENTADORA

_____
Prof. Dr. Aldebaro Barreto da Rocha Klautau Junior
MEMBRO DA BANCA EXAMINADORA

_____
Prof. Dr. Ádamo Lima de Santana
MEMBRO DA BANCA EXAMINADORA

_____
Msc. Vinicius Duarte Lima
MEMBRO DA BANCA EXAMINADORA

_____
Prof. Msc. Ronaldo Nonato Silva Lima
DIRETOR DA FACULDADE DE ENGENHARIA ELÉTRICA

Dedicatória

À minha família, pelo amor com o qual fui criado.

Agradecimentos

À minha família, que sempre me incentivou ao trabalho e ao estudo, e cujo sentimento de amor e fraternidade eu procuro levar aonde quer que eu vá. Agradeço especialmente aos meus irmãos Gabriel e Lucian, por serem as pessoas com as quais tive a maior convivência até hoje, tendo com eles aprendido muito sobre o que é o verdadeiro companheirismo.

À minha namorada, Satie, pelo amor, carinho, paciência, e apoio em todo e qualquer momento. Agradeço também pela ajuda na confecção deste trabalho.

A todos os meus amigos e companheiros, em especial ao Alan, Amagol, Bruno, César, Biro, Diego, Donza, Felipe, Flávio, Luiz Augusto, Mauro André, Pingarilho, Renan, Roberto Medeiros, Thiago, e Wilson.

Ao LEA e seus integrantes, responsável por parte significativa do meu aprendizado profissional e humano enquanto estudante universitário. Agradeço especialmente ao Roberto Menezes, Lamartine Souza, e Vinícius Duarte, que me orientaram durante meu período de estágio.

Aos professores do curso de Engenharia Elétrica, que através dos seus ensinamentos e desafios me fazem sentir orgulho imenso em estar me formando.

Resumo

A tecnologia de acesso em banda larga Linha Digital do Assinante, ou DSL (do inglês *Digital Subscribe Line*) sofre distúrbios originados por basicamente quatro tipos de ruído: *crosstalk,* impulsivo, ruído de radiofreqüência e ruído de fundo. Identificar qual deles afeta o desempenho de um tráfego DSL em um determinado momento pode ser uma informação importante para um melhor gerenciamento do enlace local por parte das operadoras de telefonia, permitindo que algoritmos que atuam nos modems se adaptem durante a conexão, e também facilitando saber qual a origem de certos problemas no serviço prestado ao usuário. Este trabalho propõe o reconhecimento de padrões de ruído na parte do enlace de cobre do VDSL2, a mais recente tecnologia DSL, a partir da aplicação da técnica Máquinas de Vetores de Suporte (*Support Vector Machines* – SVM) sobre um conjunto de informações estatísticas de gerenciamento, disponíveis na camada de aplicação através do Protocolo Simples de Gerência de Rede (*Simple Network Management Protocol*– SNMP).

Abstract

The Digital Subscriber Line (DSL) broadband technology is mainly disturbed by four types of noise: crosstalk, impulse noise, radiofrequency noise and background noise. The identification of which one is limiting the performance of the DSL traffic while the system is running can be useful for a better management of the local loop by the telephone companies, allowing modem algorithms to adapt to different noise situations, and making easier the task of finding some deployment problems sources. This work proposes the noise pattern recognition on the copper local loop of VDSL2, the most recent DSL technology, through the application of Support Vector Machines (SVM) on a set of statistical management information available at the application layer through the Simple Network Management Protocol (Simple Network Management Protocol-SNMP).

Lista de Figuras

Lista de Tabelas

Lista de Siglas

ADSL	Asymmetric Digital Subscriber Line
ADSL2+	Asymmetric Digital Subscriber Line 2
AM	Amplitude Modulation
ATU-C	ADSL Terminal Unit – Central
ATU-R	ADSL Terminal Unit – Remote
AWGN	Additive White Gaussian Noise
CO	Central Office
DMT	Discrete Multi-tone
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
FEXT	Far-end Crosstalk
FFT	Fast Fourier Transform
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
ISDN	Integrated Services Digital Network
MIB	Management Information Base
NEXT	Near-End Crosstalk
PSD	Power Spectral Density
PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
REIN	Repetitive Electrical Impulse Noise
RFC	Request for Comments
RFI	Radio Frequency Interference
SNMP	Simple Network Management Protocol
SNR	Signal-to-Noise Ratio
SVM	Support Vector Machines
VC	Vapnik-Chervonenkis
VDSL2	Very-high bit rate DSL
VTU-R	VDSL Terminal Unit-Remote

Sumário

CAPÍTULO 1

1 INTRODUÇÃO

O ruído em um sistema de comunicação é um dos fatores que causa maior impacto no desempenho do mesmo. No caso de sistemas do DSL, a presença principalmente do *crosstalk* (em português, diafonia, porém o termo em inglês será utilizado devido à sua ampla aceitação) e do ruído impulsivo causam problemas para operadoras e usuários, podendo limitar bastante o uso da tecnologia. A possibilidade da identificação de qual tipo de ruído ocorre em um enlace DSL em um determinado momento se torna deste modo importante para aqueles que vendem o produto, pois terão um meio de monitorar melhor a sua rede nesse aspecto, também permitindo que algoritmos que atuam nos modems se adaptem durante a conexão, e facilitando saber qual a origem de certos problemas no serviço prestado ao usuário (Yang, Dasgupta, Redfer, & Ali).

Uma maneira de resolver esse problema seria através da utilização de técnicas de inteligência computacional, que abrangem uma larga gama de algoritmos e teorias, entre eles a aprendizagem de máquina. Máquinas de Vetor de Suporte (*Support Vector Machines* – SVM) é uma dessas técnicas, sendo baseada na teoria do aprendizado estatístico, desenvolvidas por Vladimir Vapnik e Alexey Chervonenkis (Vapnik, 1998). A ideia principal do SVM é a seguinte: dado um conjunto de vetores de dois padrões diferentes, realizar a projeção dos mesmos em um espaço de igual ou maior dimensão e identificar o subconjunto deles que permita o cálculo de uma função discriminante para realizar a classificação de novos vetores. No caso do problema de identificação de ruído em DSL, o conjunto de vetores são dados da Base de Informação de Gerenciamento, ou MIB (do inglês *Management Information Base*) provenientes do Multiplexador de Acesso DSL, ou DSLAM (do inglês *Digital Subscriber Line Access Multiplexer*), acessíveis através do Protocolo Simples de Gerência de Rede (*Simple Network Management Protocol* – SNMP) da internet. SVM é considerado o estado da arte em aprendizado de máquina e mineração de dados, possuindo uma sólida fundamentação teórica e sendo capaz de lidar com problemas de alta dimensionalidade (Wu, et al, 2008). A aplicação desta técnica será a principal contribuição deste trabalho para encontrar a relação entre a estatística das métricas obtidas e a presença de um determinado tipo de ruído.

1.1 OBJETIVO DO TRABALHO

Dispondo da técnica de aprendizado estatístico SVM desenvolvida computacionalmente (através do software MATLAB), o trabalho tem como objetivo criar uma ferramenta de classificação do ruído presente em uma rede de segunda geração do DSL a taxas muito altas de bit (*Very-high bit rate DSL* – VDSL2), atuando na central telefônica. A classificação desejada possui quatro padrões diferentes: *crosstalk*, Ruído Impulsivo Repetitivo (*Repetitive Electrical Impulse Noise* – REIN), ruído de fundo e Interferência de Rádio Frequência (*Radio Frequency Interference* - RFI).

É importante ressaltar que, apesar de o objetivo deste trabalho ser a classificação do tipo de ruído ocorrendo no DSL, esta pode ser considerada apenas a primeira etapa de um trabalho de pesquisa maior, onde se deseja chegar a estágios de inferência sobre o ruído com o máximo de informações físicas possíveis de serem alcançadas somente a partir das medições na camada de aplicação.

1.2 REVISÃO BIBLIOGRÁFICA E ESTADO DA ARTE

No que se refere a pesquisas já realizadas que possuam semelhança ao presente trabalho, pode-se dividi-los em diferentes categorias: identificação de ruído e aprendizagem de máquina/mineração de dados aplicadas à MIB. Nota-se que ocorre de muitas vezes a identificação do ruído ocorrer junto à estimação do mesmo, sendo que esta não é o objetivo deste trabalho.

Um modo comum de identificação de ruído é através de medidas com equipamentos colocados na casa do usuário. A nota de aplicação (Dunford, 2008) descreve um exemplo deste tipo, que consiste em medidas de potência do ruído em cada portadora. Nota-se a necessidade da interrupção do serviço do assinante para a medição de nível de potência no local do mesmo. Em (Galli, Valenti, 2001) é apresentada uma técnica de identificação de boa precisão baseada em correlação de densidades espectrais de potência medidas, atuando principalmente na camada física de comunicação, o que difere do presente trabalho, que busca realizar inferência em termos de medições da camada de aplicação. Neles a identificação ocorre também quando não está sendo havendo tráfego DSL. Em (Yang, Dasgupta, Redfer, & Ali), a classificação de ruído *crosstalk* é realizada a partir da estimação das densidades espectrais de potência dos ruídos atuando sobre o enlace, sem a interrupção do serviço.

A aplicação de técnicas de aprendizagem de máquina às variáveis MIB consiste basicamente para detecção de erros ou anomalias na rede, como mostrado em (Kulkarni, *et al.*,2006)(Gazineu, 2007). Neles, a gerência de rede através das variáveis MIB é apresentada de maneira mais genérica, com o intuito principal de detectar erros de tráfego, como congestionamento em um nó. Em (Li & Manikopoulos, 2003), é proposto um sistema para detecção de ataques de negação de serviço (*Denial of Service* - DoS). Nele, cada variável MIB é caracterizada por uma densidade de probabilidade padrão, e o monitoramento em tempo real dessas variáveis permite ao sistema realizar, caso o conjunto de variáveis tenha uma estatística diferente daquela de referência, uma classificação entre ocorrência ou não de ocorrência de ataque, através de uma rede neural. Em (Cui-Mei, 2009), o mesmo resultado é desejado, porém desta vez usando SVM como técnica de classificação, o que o torna bastante semelhante ao modo como se pretende classificar os tipos de ruído em DSL neste trabalho. Este artigo propõe o uso de Seleção de Características através da Correlação, ou CFS (do inglês *Correlation Feature Selection*), para selecionar as métricas que vão participar do processo de classificação. Outra ideia interessante deste autor é utilizar SVM em dois níveis de hierarquia: no primeiro nível deseja-se classificar se há ou não ataque DoS, e no nível seguinte deseja-se classificar qual tipo específico de DoS foi realizado, dado que houve um ataque. Finalmente, em (Farias, *et al.*, 2011), artigo do mesmo grupo de pesquisa em DSL do presente trabalho, é proposta uma técnica para classificação e estimação do ruído em tempo real através de regressão linear e lógica fuzzy. Nele, não há necessidade de interrupção do serviço do assinante. O objetivo e os métodos propostos nele são semelhantes aos deste trabalho, diferindo principalmente a técnica de classificação utilizada e as variáveis escolhidas para realizar a classificação.

1.3 ORGANIZAÇÃO DOS CAPÍTULOS

O presente trabalho está dividido nos seguintes capítulos:

- **Capítulo 1 –** Introdução.
- **Capítulo 2 -** Tecnologia xDSL: Neste capítulo é apresentada uma visão geral sobre a tecnologia xDSL, e é feito estudo mais aprofundado sobre a presença do ruído no mesmo. O funcionamento do protocolo SNMP e das MIB em redes xDSL também é introduzido.
- **Capítulo 3 -** Máquinas de vetor de suporte: A técnica de aprendizagem de máquina e mineração de dados é apresentada do ponto de vista do

reconhecimento de padrões, através de suas principais ideias: a construção de um hiperplano de separação ótimo entre dois conjuntos de padrões diferentes, e as funções kernel, que permitem a generalização do algoritmo para problemas não lineares.

- **Capítulo 4 -** Metodologia: é explicado como foram feitas as medições de tráfego em VDSL2 utilizando cabo real, e o modo como foi aplicada a máquina de vetores de suporte.
- **Capítulo 5 -** Resultados: exposição dos resultados obtidos, a partir do uso de matrizes de confusão.
- **Capítulo 6 -** Conclusão: conclusão sobre os resultados obtidos e o trabalho realizado, bem como as propostas de possíveis outros trabalhos que sigam a mesma linha de raciocínio.

CAPÍTULO 2

2 TECNOLOGIA DSL

O sistema DSL é a tecnologia de acesso mais utilizada por usuários residenciais e comerciais (Broadband, 2008). Ela consiste na transmissão de dados, vídeo e áudio, em alta velocidade, através da rede telefônica. A característica principal desta tecnologia é o fato de ela ter sido construída para transmitir sinais digitais em banda larga aproveitando a infra-estrutura de telefonia já existente (que transmite sinais analógicos de voz), cuja existência remete ao começo do século XX. A estrutura da rede telefônica foi criada para operar na estreita faixa de voz, que vai até 4 kHz. Com o advento da eletrônica e dos computadores, tornou-se necessária a pesquisa para encontrar meios de transmitir informações digitais, e a possibilidade de usar faixas maiores de frequência para a composição dos sinais e garantir a qualidade do serviço tornou-se um objetivo a ser alcançado. Devido à infra-estrutura telefônica já existente e popularizada pelo mundo, pensou-se em utilizá-la também para o fim da comunicação digital. A utilização da rede telefônica foi muito importante para a popularização do DSL, principalmente porque permitiu que o acesso em banda larga fosse oferecido a preços muito menores em comparação com a fibra óptica, por exemplo.

De modo a compreender o funcionamento desta tecnologia, é necessário compreender inicialmente como funciona a Rede Telefônica Pública Comutada, ou PSTN (do inglês *Public Switched Telephone Network*). A arquitetura básica de um sistema de telefonia pode ser vista na Figura 1:

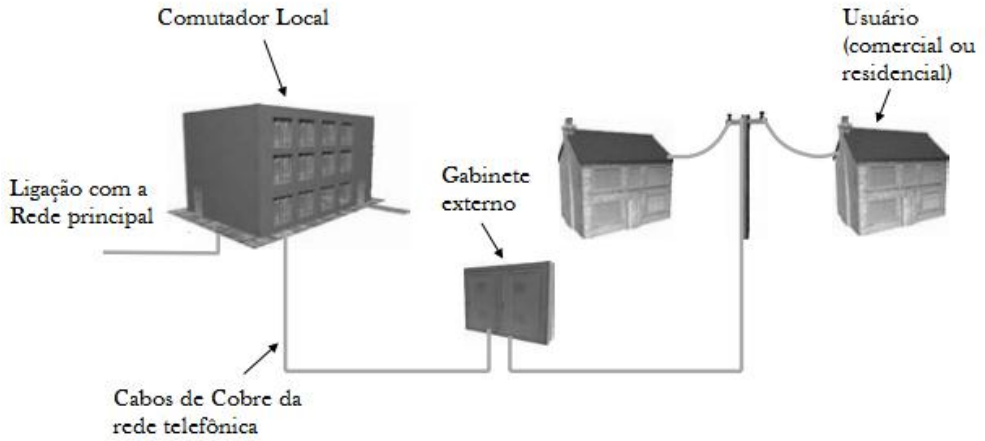

Figura 1 - O enlace local de telefonia (Golden, Dedieu, & Jacobsen, 2004).

A rede telefônica opera na faixa até 4 kHz, funcionando por meio de cabos contendo pares trançados de cobre, enviando sinais analógicos de voz. O comutador local, também chamado de central telefônica, é o responsável pela comutação das

chamadas, realizando o direcionamento das ligações. Esta comutação já foi feita de diversas maneiras, desde operação manual, eletromecânica, e atualmente através de interfaces digitais (Golden, Dedieu, & Jacobsen, 2004). Da central telefônica sai o *backbone*, cabo comportando os diversos pares que correspondem aos usuários sendo atendidos por ela. Esses pares são divididos em direção aos usuários finais através de gabinetes localizados na rua, podendo passar por mais de um em seu caminho. O acréscimo do serviço DSL trouxe mudanças principalmente nas extremidades da linha telefônica, já na central e no usuário, como mostra a Figura 2:

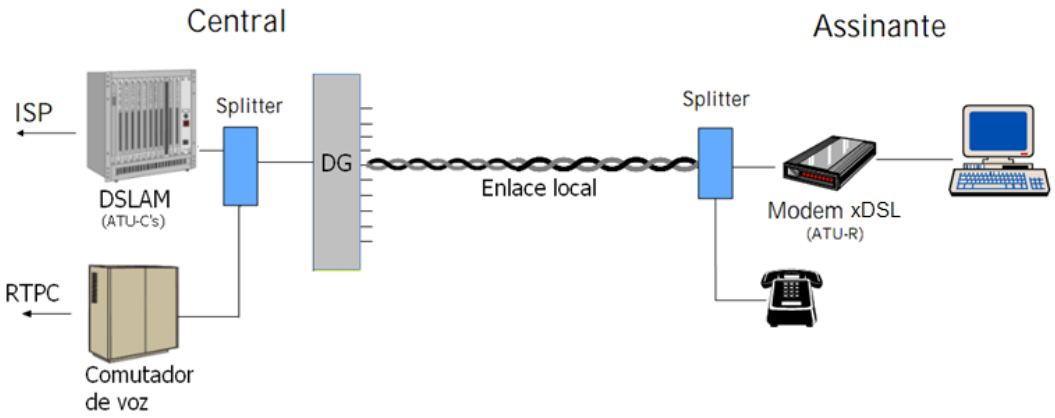

Figura 2 – Diagrama esquemático de uma rede DSL.

As principais adições à estrutura telefônica advindas do DSL foram as seguintes:

Modem xDSL: O modem DSL é um transceptor que fica conectado a um computador ou switch, responsável pelo tratamento analógico e digital do sinal elétrico em DSL. Fica localizado na casa do usuário, onde é normalmente referido como a unidade terminal, ou ATU-R (do inglês ADSL Terminal Unit – Remote), e também está presente na central, como componente do DSLAM, sendo assim referido como Unidade Terminal ADSL–Central, ou ATU-C (do inglês ADSL Terminal Unit – Central).

DSL Access Multiplexer (DSLAM): é o responsável pela multiplexação do tráfego na rede DSL, permitindo que diversos usuários possam usufruir do serviço. Normalmente está presente na Central Telefônica (*Central Office* – CO), que é considerada o começo do enlace. O DSLAM usualmente contém muitos modems ATU-C, servindo a uma grande quantidade de usuários.

***Splitter*:** Como a informação de voz e de dados chega ao consumidor através do mesmo par trançado, é necessário utilizar filtros que dividam a faixa espectral de ambos, para que não haja interferência de um serviço no outro, normalmente através de ecos (G.993.1, 2004). A utilização dos *splitter*'s denota também a possibilidade do usuário de usar o serviço de telefonia e o DSL simultaneamente.

2.1 TIPOS DE DSL:

Um grande número de tipos tecnologias DSL foi e ainda é oferecido no mercado, denotando a versatilidade que é possível alcançar na transmissão de sinais através de cabos de cobre. Entre todas as tecnologias, duas merecem bastante destaque: o ADSL, por ser o tipo mais popular e vendido no mundo, e o VDSL2, que é o seu mais recente estágio desenvolvimento.

2.1.1 ADSL:

O ADSL é o tipo de DSL que permitiu a popularização do serviço, devido à sua capacidade de ser funcional em enlaces de até 4 km, e por ser uma tecnologia de transmissão assimétrica, significando que a banda de *upstream* é diferente da banda de *downstream*, tendo como conseqüência que a taxa de transmissão dos dois também é diferente. Um dos principais motivos para a sua aplicação comercial foi para o de vídeo em demanda (Patrício, 2006). A divisão do seu espectro, como ilustra a Figura 3, favorece o *downstream*, que normalmente é prioritário em termos de usuários residenciais.

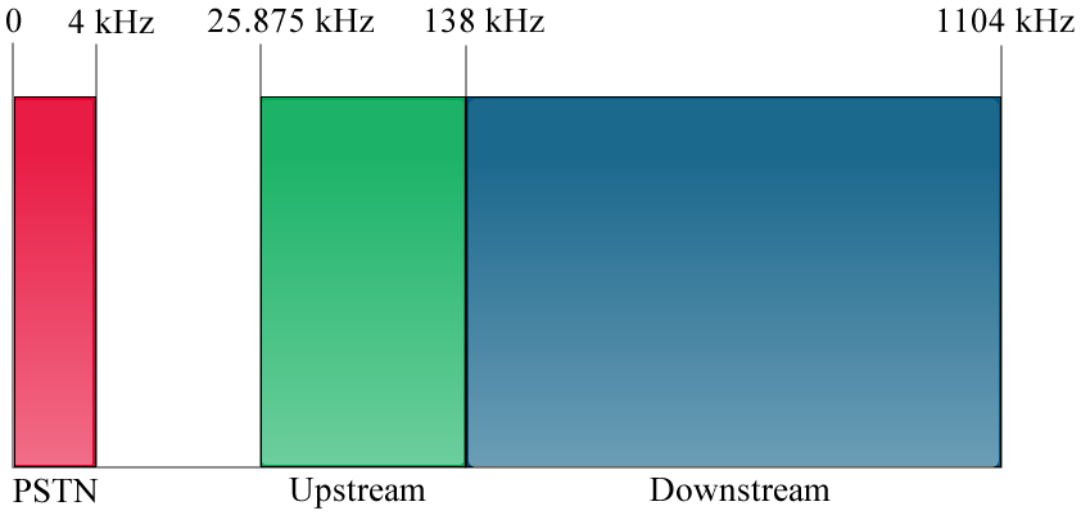

Figura 3 - Divisão do espectro de frequência no ADSL.

Nota-se na Figura 3 que a faixa reservada à voz é dividida daquela reservada à comunicação digital. A última versão do ADSL, o ADSL2+, proporciona até 24 Mbps como taxa de *downstream* e 1Mbps para o *upstream* (Broadband, 2008).

2.1.2 VDSL2

A tecnologia DSL veio como uma maneira de amenizar a necessidade dos consumidores por acesso à internet em banda larga. Entretanto, o avanço das redes de fibra óptica e redes híbridas ópticas-coaxiais faz com que as operadoras de telefonia tenham a constante preocupação com o aumento da taxa de transmissão, de modo a

manter seus usuários fiéis ao serviço oferecido (Papandriopoulos & Evans, SCALE: A Low-Complexity Distributed Protocol for Spectrum Balancing in Multiuser DSL Networks, 2009). A criação do VDSL2 representa esta tentativa de manter a tecnologia ainda bastante competitiva, com a faixa de frequência se estendendo até 30 MHz e prometendo até 100 Mbps em um tráfego simétrico para pequenos enlaces (até 1500 m). O tráfego assimétrico também é permitido, com velocidade de 150 Mbps para downstream e 50 Mbps para o upstream. A expansão da fibra óptica a partir da central nos enlaces é um dos fatores que permite ao VDSL2 alcançar taxas tão grandes em relação às tecnologias DSL anteriores, especificamente na modalidade de fibra até o gabinete (*Fiber to the Curb* - FTTC), com o VDSL2 ligando a fibra óptica a consumidores residenciais, e na fibra até o prédio (*Fiber to the Building* - FTTB), ligando a fibra óptica a consumidores comerciais principalmente. VDSL2 traz mudanças significativas em termos de infra-estrutura, visto que agora os DSLAMs (antes normalmente localizados dentro das CO's) terão que ser posicionados em gabinetes situados próximos ao usuário (Eriksson & Odenhammar, 2006).

2.2 Ruído em Sistemas de Comunicação

O ruído é um dos principais fatores limitantes de um sistema de comunicação, e uma descrição breve do seu impacto faz-se interessante, já que identificá-lo é um dos objetivos deste trabalho. Em seu artigo seminal (Shannon, 1948), Shannon descreveu matematicamente a influência do ruído sobre a capacidade do canal de comunicação. Considerando o sistema de comunicação digital descrito na Figura 4:

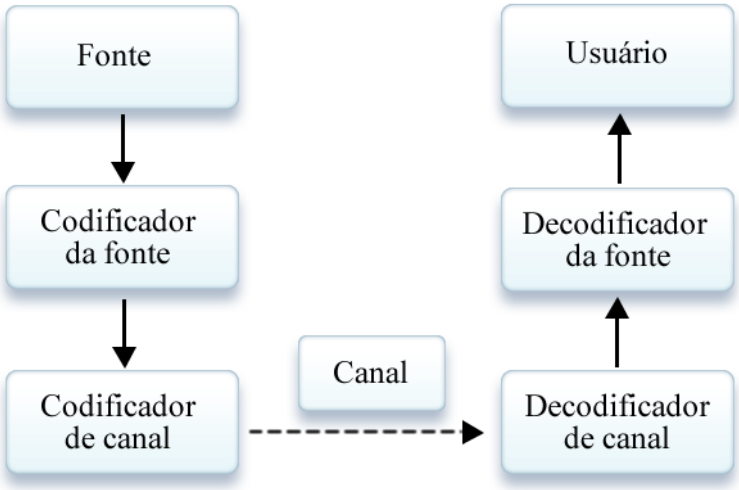

Figura 4 - Diagrama esquemático geral de um sistema de comunicação digital.

A capacidade do canal em um sistema digital foi descrita na Equação 2.1 como sendo:

$$C = \max_{\mathcal{P}_{S_a}} Im(S_a, S_b), \quad (2.1)$$

onde $Im(Sa, Sb)$ é a informação mútua entre o emissor (S_a) e o receptor (S_b). Pode-se dizer então que a capacidade do canal é a taxa máxima de informação que se pode transmitir através do mesmo, dada em bits/Hz, alcançada em função da distribuição de probabilidade do emissor. Logo, caso a taxa do canal seja $R < C$, pode haver um aumento no comprimento dos blocos de informação que garanta que sua chegada ao receptor sem erro (MacKay, 2003). Caso $R > C$, a probabilidade de erro do bloco aumenta proporcionalmente ao comprimento do mesmo, fazendo com que a comunicação comece a se tornar impraticável. Ao estender a Equação 2.1 para um canal na presença de ruído, chegou-se à seguinte Equação 2.2 para a capacidade:

$$C = W.\log_2(1 + SNR), \quad (2.2)$$

Onde W é a largura de banda do canal em Hz, e SNR é a Razão Sinal-Ruído, RSR (*Signal-to-Noise Ratio* – SNR), mostrando a relação entre a potência do sinal contendo a informação que se deseja transmitir, e a potência do ruído presente no canal afetando a informação. Logo, é possível aumentar a capacidade de transmissão em um canal aumentando a largura de banda ou melhorando a razão sinal-ruído do mesmo. O SNR varia de acordo com o tipo de ruído que se considera atuando sobre o sistema. Será visto adiante que é possível obter informações interessantes para realizar uma caracterização do ruído (como a margem de SNR, número de blocos corrigidos, e taxa de transmissão da central para o usuário) através do protocolo SNMP.

2.3 Tipos de Ruído em Sistemas DSL

2.3.1 *Crosstalk*

O *crosstalk* em DSL é o ruído causado devido ao acoplamento eletromagnético entre os fios de cobre que compõem o canal de comunicação telefônico, e é o principal tipo de ruído presente em um enlace DSL (Cendrillon, 2004). Um dos motivos é a má qualidade dos tradicionais fios telefônicos, que agravam o efeito do *crosstalk* no desempenho do serviço DSL. Este tipo de ruído pode ocorrer mesmo em enlaces curtos, e depende da topologia do enlace (Papandriopoulos, *et AL*, 2009). O *crosstalk* quando dois pares próximos estão trabalhando na mesma faixa de freqüência.

O *crosstalk* pode ser categorizado em dois tipos diferentes: NEXT e FEXT, definidos a seguir.

2.3.1.1 NEXT

O *crosstalk* do tipo NEXT (do inglês *Near-End Crosstalk*) é o acoplamento que ocorre em outro par trançado no sentido contrário ao do sinal original. Sua ocorrência é mais intensa nas extremidades do enlace, mas pode ocorrer também no meio dele (Golden, Dedieu, & Jacobsen, 2004). Este tipo de ruído ocorre em transmissões DSL simétricas. A Figura 5 exemplifica a ocorrência do NEXT:

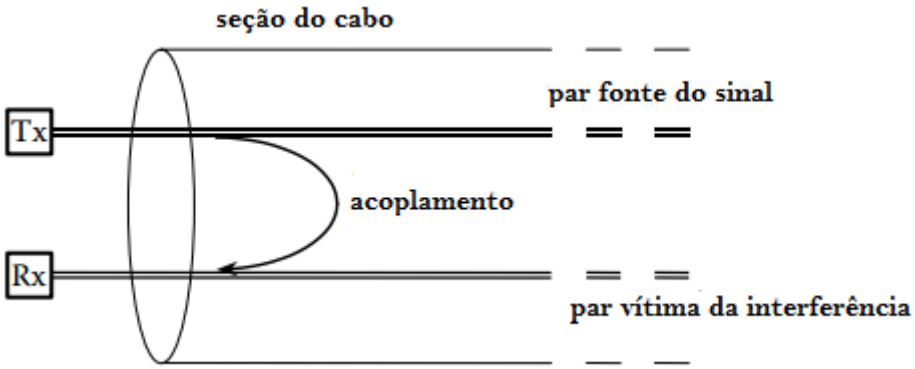

Figura 5 – NEXT (Golden, Dedieu, & Jacobsen, 2004).

O modelo (empírico) da densidade espectral de potência do ruído NEXT é dado pela Equação 2.3 (Golden, Dedieu, & Jacobsen, 2004):

$$PSD_{NEXT}(f)^2 = PSD_{SINAL}.K_{NEXT}.|f|^{1,5}. \quad (2.3)$$

Este modelo indica a relação entre o sinal transmitido por N pares interferentes, e o *crosstalk* sobre um cabo sofrendo a interferência, pois K_{NEXT} é proporcional ao número de pares interferentes no cabo.

2.3.1.2 FEXT

O FEXT (do inglês *Far-end Crosstalk*) é o acoplamento que ocorre no mesmo sentido do sinal transmitido. O FEXT, ilustrado na Figura 6, pode ocorrer em transmissões simétricas e assimétricas.

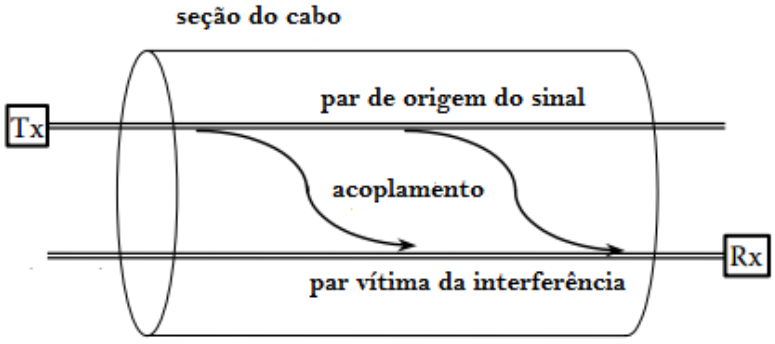

Figura 6 – FEXT (Golden, Dedieu, & Jacobsen, 2004).

O modelo (empírico) da potência do ruído FEXT é dado pela Equação 2.4 (Golden, Dedieu, & Jacobsen, 2004):

$$PSD_{FEXT}(f)^2 = PSD_{SIGNAL}.K_{FEXT}.C(f,L).|f|^2, \quad (2.4)$$

onde $C(f)$ é proporcional ao comprimento do cabo e ao número de interferentes, e K_{FEXT} é uma constante.

O NEXT é muito mais pernicioso que o FEXT, pois ele afeta um sinal que já sofreu perdas devido à distância percorrida, enquanto que o FEXT sofre atenuação ao longo da linha e depois afeta o sinal. A partir dos modelos apresentados, percebe-se que a o nível de interferência aumenta à medida que aumenta a frequência de operação do DSL, e também quanto maior a quantidade de interferentes conectados no mesmo *binder* (cabo de proteção revestindo um conjunto de pares trançados), maior a potência do ruído (Golden, Dedieu, & Jacobsen, 2004).

2.3.2 Ruído Elétrico Impulsivo Repetitivo

O Ruído Elétrico Impulsivo Repetitivo, ou REIN (do inglês *Repetitive Electric Impulsive Noise*) é o ruído que se caracteriza por pulsos elétricos de curta duração, mas de potências muito elevadas, ocorrendo com periodicidade. Sua interferência é gerada por qualquer pulso eletromagnético que ocorra nas proximidades do enlace, e, portanto sua fonte não é facilmente identificada, sendo normalmente proveniente da atividade humana e transitórios causados por chaveamento (Golden, Dedieu, & Jacobsen, 2004). A potência do ruído impulsivo pode ser grande o suficiente para muitas vezes interromper a transmissão em sistemas DSL.

2.3.3 Ruído de Radiofrequência

É o ruído causado pela interferência de ondas de rádio, tendo como origem normalmente transmissões de emissoras usando modulação AM ou de rádio amador. É um ruído que possui uma faixa estreita de frequência, entre 2,5 e 5 kHz, sendo que a faixa do AM localiza-se entre 0,5 e 1,6 MHz, e a do amador entre 1,8 e 29 MHz, que também é compartilhada pelo ADSL e pelo VDSL2. Em comparação com o *crosstalk* e o ruído impulsivo, o impacto do ruído de radiofrequência possui menor importância. No entanto, na frequência do VDSL2, a potência do ruído pode chegar a -30 dBm e 0 dBm, para rádio AM e amador respectivamente, considerando a transmissão em modo diferencial, o que leva o RFI estar incluído na padronização do VDSL2 (Nedev, 2003).

2.3.4 Ruído de Fundo

O ruído de fundo caracteriza-se por estar presente no sistema mesmo quando a fonte não está emitindo. Ele é criado pela composição das interferências de diversas fontes externas ao sistema de comunicação. Em DSL, ele possui como padrão a

densidade espectral de potência de -140 dBm/Hz, podendo ocupar diferentes faixas de frequência (Golden, Dedieu, & Jacobsen, 2004) (Brost & Aspell, 2002).

2.4 Métricas MIB em DSL

Como as medições da camada física podem causar a interrupção do serviço de internet, uma alternativa para inferência do ruído seria buscar nas camadas superiores um meio de realizá-lo, a partir dos dados apresentados por elas. Existe um conjunto de ferramentas de gerenciamento de rede na camada de aplicação. Entre elas, o SNMP, que foi criado por um grupo do *Internet Engineering Task Force* (IETF) justamente para realizar o monitoramento e gestão das redes de computadores e dispositivos de Protocolo de Internet (*Internet Protocol* – IP). Ele segue basicamente um conjunto de regras que permite a um computador obter informações estatísticas (como pacotes perdidos, número de erros, margem de ruído, entre muitas outras) a respeito de outro computador. Ele permite, por exemplo, que um administrador de rede possa diagnosticar e corrigir problemas na rede a partir de servidores remotos. O SNMP foi definido a partir de três documentos RFC (*Request for Comments*) (Gaïti, 2005):

RFC 1156 *Management Information Base* – MIB

RFC 1157 SNMP *Protocol*

RFC 1213 *Management Information Base* – MIB II

Cada informação obtida pelo SNMP é estocada em um MIB, que é um banco de dados virtual criado para a tarefa da gestão de rede de comunicações. Esse banco de dados possui diversas variáveis, chamadas de variáveis MIB, que neste trabalho serão também chamadas métricas. Os MIB são informações relacionadas ao gerenciamento de dispositivos, como impressoras, roteadores, e modems. Eles são organizados de acordo com a árvore descrita na Figura 7:

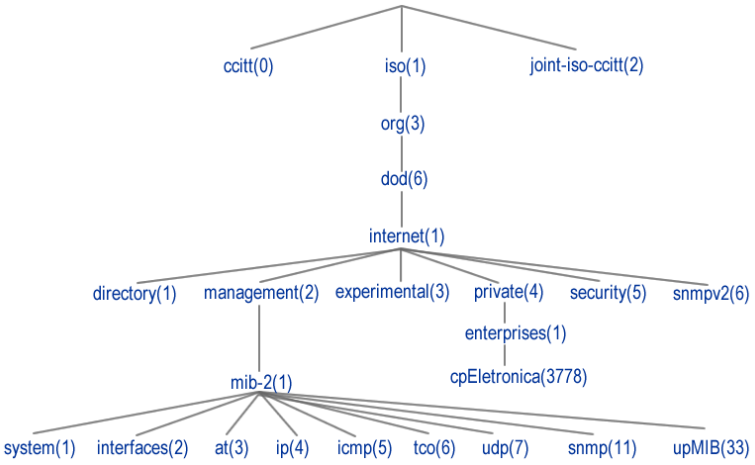

Figura 7 - Estrutura da árvore MIB.

O fato de o protocolo SNMP e o MIB serem ferramentas que permitem monitorar o funcionamento de uma rede através de métricas estatísticas, aliado à explicação dada na seção (ruído em sistemas DSL), propicia uma possibilidade de obter um conhecimento adicional àquele simplesmente descrito pelas mesmas, justificando a realização dos MIB como fonte de dados para o processo de aprendizagem de máquina que será visto adiante. No caso da classificação de ruído, deve-se tentar descrever o comportamento dessas métricas na presença do mesmo.

O MIB em sistemas DSL é obtido a partir de um módulo do DSLAM, que obtém informações de gerenciamento para cada par de modems situados nas extremidades do enlace, como mostra a Figura 8 (Ericsson, 2009).

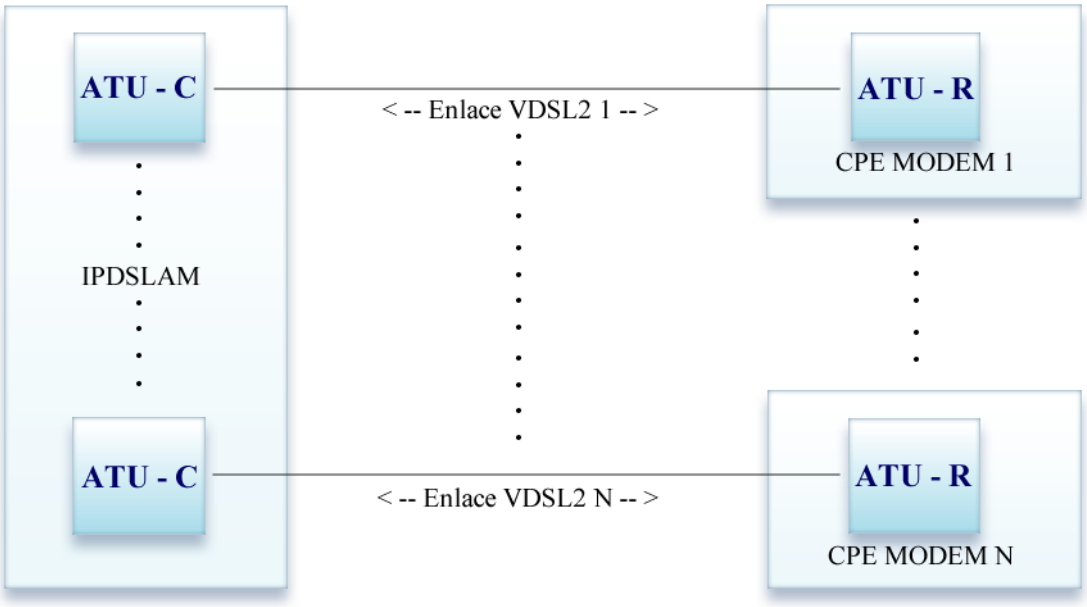

Figura 8 – DSLAM obtém as métricas MIB de cada enlace DSL (Ericsson, 2009).

CAPÍTULO 3

3 MÁQUINAS DE VETOR DE SUPORTE

3.1 Aprendizado de Máquina

Máquinas de vetores de suporte é uma técnica que pertence a um ramo da inteligência artificial chamado de *Aprendizado de Máquina*. De modo a entender melhor certos conceitos genéricos utilizados em SVM, uma introdução ao aprendizado de máquina se faz útil. O aprendizado de máquina é composto por diversas técnicas que possuem em comum um objetivo: fazer com que determinado sistema execute certa tarefa com base em uma série de experiências relacionadas com a mesma, ou seja, fazer com que o sistema *aprenda*. O processo de aprendizagem ocorre analisando um conjunto de m dados de entrada $\boldsymbol{x}_i$, com $i = 1,2,\dots,m$, onde $\boldsymbol{x}$ é um vetor de dimensão n, e a partir desta análise inferir um determinado padrão de saída y. A arquitetura tradicional de uma máquina de aprendizado pode ser conferida na Figura 9, onde $h = h(x)$ é uma hipótese (modelo) representando a relação aproximada entre x e y, e $\tilde{y}$ é a saída estimada pela hipótese

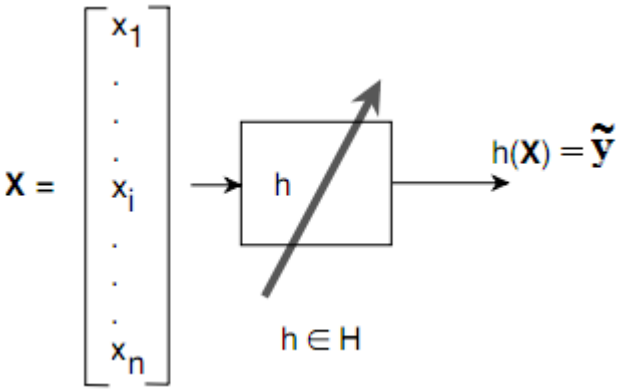

Figura 9 - Máquina de aprendizado.

Uma descrição formal para a problemática do aprendizado de máquina pode ser a seguinte (Mitchell, 1997):

"Um programa de computador aprende, baseado em uma experiência E, com respeito a uma tarefa T, e uma medida de desempenho P, se o desempenho em T, medido por P, aumenta com a experiência E."

O processo do aprendizado de máquina pode ser resumido nas 5 etapas seguintes:

1 – Obtenção do conjunto de *dados de treino* $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_m)$

2 – Formulação de uma (ou várias) hipótese $h(\boldsymbol{x}) = \tilde{y}$, com $h(\boldsymbol{x})$ representando um conhecimento a priori da relação entre os dados de entrada $\boldsymbol{x}$ e a saída y.

3 – Treinamento do algoritmo de aprendizado utilizando os dados obtidos em 1, adaptando a hipótese h a medida que a máquina recebe mais dados. Esta é a etapa de aprendizagem do algoritmo, existindo uma grande quantidade de técnicas diferentes que podem ser usadas para treinamento.

4 - Teste da hipótese obtida a partir dos dados de treino utilizando os *dados de validação.* Se no começo desta etapa houver mais de uma hipótese plausível para um modelo definitivo, apenas uma será escolhida para a etapa 5. A hipótese final também pode ser modificada nesta parte.

5- Aplicação da hipótese final obtida nos *dados de teste*, de modo a verificar se ela generaliza bem para novos dados, ou seja, se ela fornece a resposta correta para dados para os quais o algoritmo nunca foi apresentado e não conhece a saída y. O objetivo final do aprendizado é sempre obter uma hipótese que se adéqüe a esse critério, se não perfeitamente, pelo menos aproximadamente, portanto ela não deve ser modificada nesta etapa. Os dados de treino, validação e teste fazem parte do mesmo conjunto de dados obtidos de um experimento. Recomenda-se que os dados tenham um comportamento diversificado (valores não tão parecidos, ou que tenham um número diversificado de saídas, e não várias de somente um tipo, por exemplo) de modo a gerar uma hipótese generalista.

Técnicas de aprendizado de máquina são aplicadas para reconhecimento de padrões, visão computacional, robótica, economia, e várias outras áreas onde se deseja descobrir relações matemáticas dentro de um conjunto de dados. Duas categorias de aprendizado são apresentadas abaixo: supervisionado e não-supervisionado.

3.1.1 Aprendizado Supervisionado

No aprendizado supervisionado, o conjunto de dados de entrada é formado por $(\boldsymbol{x}_i, y_i)$, onde i é a i-ésima entrada da máquina de aprendizado, ou seja, é fornecida à máquina a entrada junto à sua respectiva saída. Este tipo de situação é favorável para o algoritmo de aprendizado, já que ele é "ensinado" a dar o resultado correto para determinada entrada. O aprendizado supervisionado pode ser de dois tipos: classificação ou regressão. Na classificação, busca-se classificar a saída de uma máquina de aprendizagem em termos qualitativos, normalmente discretos. O problema atual de classificação de ruído se enquadra neste caso, já que a saída pertence ao conjunto

$\{crosstalk, REIN, RFI, ruído\ de\ fundo\}$. Na regressão, busca-se obter uma saída quantitativa na máquina de aprendizado, geralmente na forma de uma função contínua $f(x)$. O aprendizado supervisionado também é chamado "aprendizado com professor". Um exemplo deste tipo de aprendizado é o reconhecimento de manuscritos, mostrado na Figura 10. Nele, é fornecida uma imagem digital representando números ou letras, com 16x16 pixels, onde cada pixel é formado por 8 bits, que representam a intensidade na escala do cinza, indo de 0 a 255. Juntamente com a imagem digital é fornecido precisamente o dígito que a entrada representa, fazendo com que o algoritmo aprenda a relacionar as imagens digitais com a sua respectiva saída. (Hastie, Tibshirani, & Friedman, 2009).

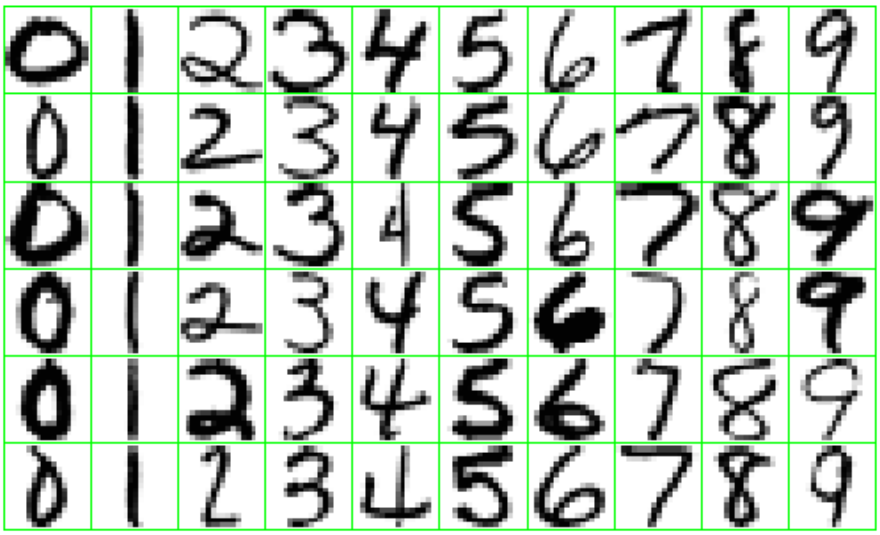

Figura 10 - Alguns exemplos de dígitos manuscritos do serviço postal americano.

3.1.2 Aprendizado Não-supervisionado

No aprendizado não-supervisionado, a saída não é fornecida juntamente com a de entrada, mas o número de classes ao qual as entradas pertencem podem ser conhecidas. Nesta modalidade de aprendizado mais difícil já que o algoritmo terá que aprender por si só quais os diferentes padrões ocultos no conjunto de dados (Hastie, Tibshirani, & Friedman, 2009). Um exemplo de problema de aprendizado não-supervisionado é o da estimação das densidades de probabilidade presentes em um conjunto de dados, sendo que o mesmo é composto por uma ou mais densidades.

3.2 Máquinas de Vetor de Suporte

Máquinas de vetor de suporte são um conjunto de algoritmos para aprendizado estatístico relativamente recentes, desenvolvidos por Vapnik et AL (Vapnik, 1998). Máquinas de vetor de suporte fazem parte da classe de algoritmos de aprendizado estatístico supervisionado, não-paramétricos e podem ser usado tanto para classificação como para regressão.

A principal ideia por trás dos algoritmos SVM consiste em encontrar o hiperplano ótimo que proporcione a máxima separação entre dois conjuntos de dados de padrões conhecidos e diferentes. Desse modo, podemos determinar o padrão de uma amostra não identificada com base em sua localização em relação ao hiperplano ótimo. Quando o conjunto de treino é composto por dados linearmente separáveis, ou seja, dados que podem ser divididos por pelo menos um hiperplano, o procedimento para obtenção do hiperplano ótimo consiste na solução direta de um problema de otimização convexa. Caso o conjunto de dados não seja linearmente separável, faz-se necessária a utilização de uma transformação que aumente a dimensão do mesmo, de modo que neste novo espaço os dados estejam linearmente separáveis e possa-se encontrar um hiperplano de separação. Esta transformação é realizada através do uso da função de *kernel*, o que torna o SVM uma técnica bastante robusta e eficiente na resolução de problemas de classificação não lineares. O fato de minimizar uma função convexa, para a qual existe uma boa quantidade de métodos analíticos de solução, facilita a sua implementação, bem como previne o algoritmo de sofrer com problemas relacionados a mínimos locais. Outra importante característica das SVM's é a possibilidade de utilizá-los em problemas com número infinito de dimensões (Cristianini & Shawe-Taylor, 2000).

3.2.1 Complexidade da Hipótese e Dimensão de Vapnik-Chervonenkis (VC)

Como já mencionado, o problema básico em aprendizado supervisionadona supervisionado é aquele de encontrar a estrutura presente nos dados de entrada $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$. Dado um novo x, deseja-se encontrar o seu y correspondente com base nas saídas já fornecidas pelos dados de treino. Um meio comum de alcançar este objetivo é a minimização do *risco empírico* (ou erro empírico) em relação aos dados de treino, dado pela Equação 3.1:

$$R_{emp}(\boldsymbol{w},\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}|f(\boldsymbol{w},\boldsymbol{x_i}) - y_i|, \qquad (3.1)$$

onde $f(\boldsymbol{w},\boldsymbol{x})$ é a hipótese considerada. No caso de SVM, $f(\boldsymbol{w},\boldsymbol{x_i})$ e y podem assumir somente os valores 1 e -1. Porém, nem sempre um erro pequeno para os dados de treino significa que nos dados de teste o erro também será pequeno, o que pode causar problemas na generalização do algoritmo, causando o chamado *overfitting*[1]. Isso pode ser mostrado da seguinte maneira (Weston): dados um conjunto de treino $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)$ e um conjunto de teste $\overline{\boldsymbol{x}_1}, \ldots, \overline{\boldsymbol{x}_k}$, ambos pertencendo ao mesmo espaço X, para todo f existe f^* de modo que

$$f(\boldsymbol{x}_i) = f^*(\boldsymbol{x}_i) \qquad \text{e} \qquad f\left(\overline{\boldsymbol{x}_j}\right) \neq f^*\left(\overline{\boldsymbol{x}_j}\right), \qquad (3.2)$$

onde $i = 1, \ldots, m$ e $j = 1, \ldots, k$. Ou seja, existe uma infinidade de funções possíveis que podem ser relacionadas aos dados de treino e suas saídas, entretanto deve-se encontrar a verdadeira função relacionando x e y, ou pelo menos boas estimativas da verdadeira função $y = f(x)$. A Figura 11 exemplifica a situação:

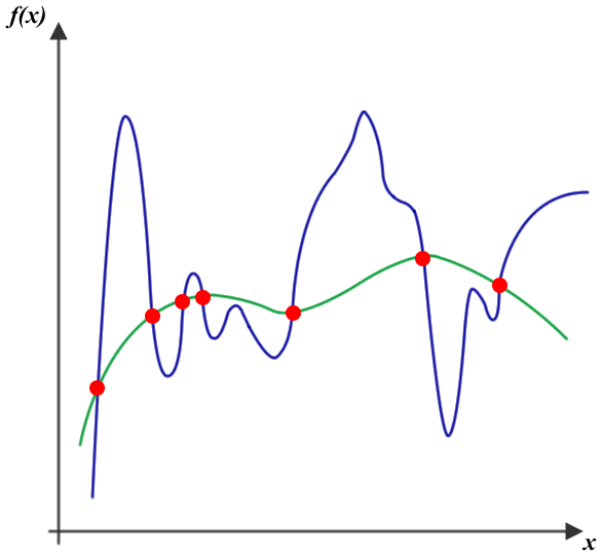

Figura 11 - Os pontos vermelhos podem pertencer a um número infinito de funções.

Deve-se, portanto, restringir o número de funções possíveis para resolver o problema, para uma quantidade que se adéque ao número de dados de treino disponível, de modo a restringir a própria complexidade da hipótese (Schölkopf, 2000).

Em um de seus principais resultados, a teoria do aprendizado estatístico determina que, de modo a generalizar eficientemente um determinado problema de

[1] O *overfitting* é o fenômeno que ocorre quando, após o treinamento, a máquina de aprendizado obtém um desempenho muito bom na classificação dos dados de treino (erro empírico tendendo a zero), porém obtém um desempenho ruim na classificação dos dados de teste. Esta má classificação ocorre porque a complexidade da hipótese final gerada acaba por depender somente da forma como os dados de treino estão distribuídos

aprendizado, o algoritmo deve ser capaz de minimizar o risco estrutural, dado pela seguinte Equação 3.3:

$$R(\boldsymbol{w},\boldsymbol{x}) \leq R_{emp}(\boldsymbol{w},\boldsymbol{x}) + \Phi(h,m), \tag{3.3}$$

onde $R_{emp}(\boldsymbol{w},\boldsymbol{x})$ é o risco empírico e m é o número de dados de treino. O termo Φ depende de m e do parâmetro h, chamado dimensão de Vapnik-Chernonenkis (VC), que determina o grau de complexidade de uma determinada classe de funções.

A dimensão VC é uma propriedade de uma família de funções $f(\boldsymbol{w})$, onde $\boldsymbol{w}$ define os parâmetros de uma determinada função. Considerando um caso de classificação binária, onde $y = \{\pm 1\}$, ela é definida como sendo o maior número h de pontos possíveis de serem divididos pela função $f(\boldsymbol{w})$ (Schölkopf, 2000) (Burges, 1998).

. De maneira mais geral, a dimensão VC é considerada uma medida da *capacidade* de uma função, sendo capacidade uma medida da complexidade de uma função. Se uma família de funções de alta capacidade for usada para classificação, é possível que ocorra *overfitting*, enquanto que famílias de funções com baixa capacidade podem acarretar em um risco empírico relativamente alto, como exemplificado na Figura 13.

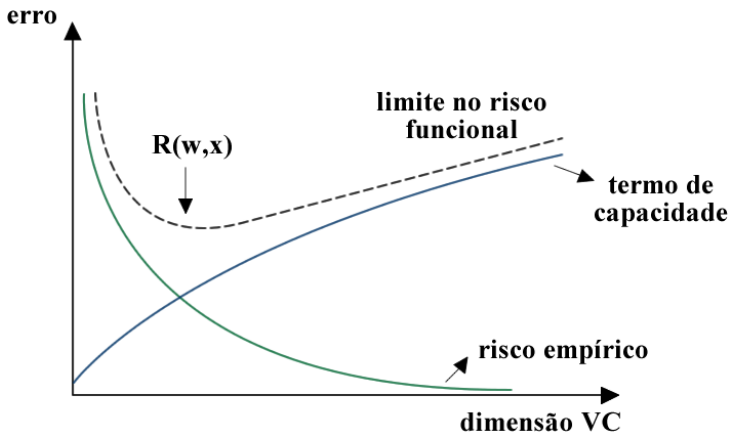

Figura 12 - Variação do risco estrutural em função da dimensão VC.

Na Figura 14, esta situação fica mais explícita para o conjunto de dados não linearmente separável das bolas opacas e das bolas vazadas. Nota-se que, da esquerda para a direita, funções de dimensão VC maior são utilizadas para se ajustarem melhor aos dados de treino.

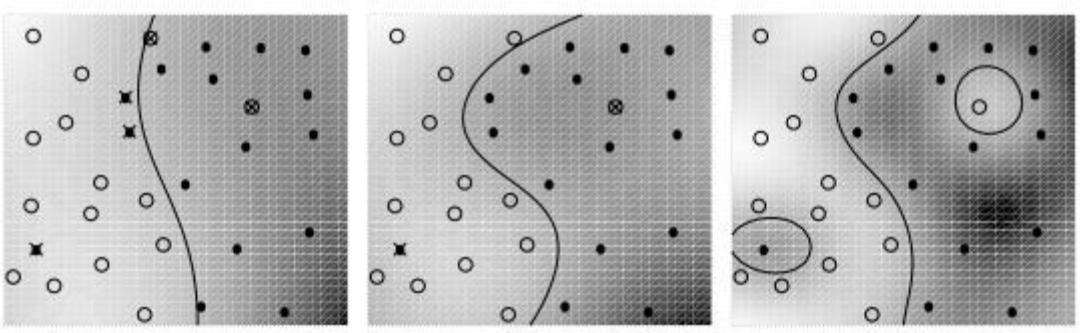

Figura 13 - Funções diferentes possuem capacidades diferentes (Weston).

A minimização do risco estrutural busca proporcionar um balanceamento entre a complexidade do espaço de hipóteses e o conjunto finito de dados de treino, fazendo com que o desempenho do algoritmo seja bom não somente nos dados de treino, mas também nos de teste.

3.2.2 Classificador de Margem Rígida e o Caso Linearmente Separável

Para compreender o funcionamento básico de SVM's, pode-se tomar como exemplo o problema mais simples, em que se treina uma máquina com dados linearmente separáveis, também chamada de máquina linear (Burges, 1998). A Figura 15 ilustra um conjunto de padrões linearmente separáveis, e como é possível traçar um ou mais hiperplanos para separá-los.

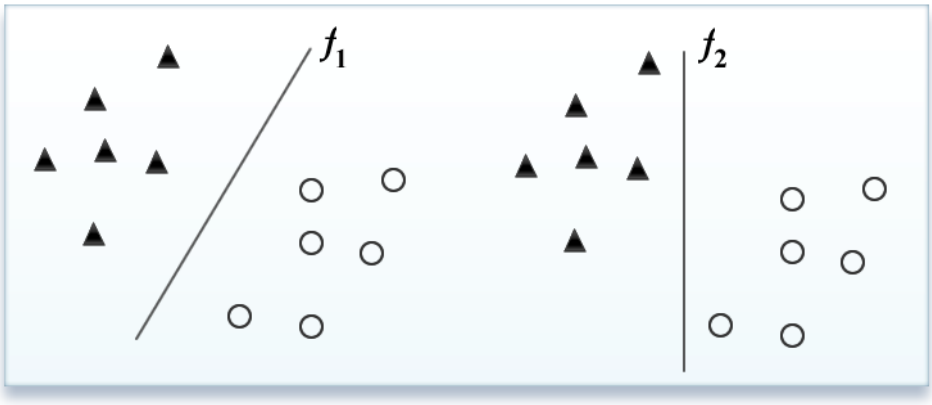

Figura 14 - Conjuntos de dados linearmente separáveis.

Supondo o problema de classificação binária onde se deseja estimar f de modo que:

$$f : \boldsymbol{X} \rightarrow \{+1, -1\}, \tag{3.4}$$

Onde $\boldsymbol{X} = (\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_3), \ldots, (\boldsymbol{x}_m, y_m)$ é um conjunto de dados linearmente separáveis independentes e identicamente distribuídos (i.i.d). Dada a classe de hiperplanos:

$$\boldsymbol{w}.\boldsymbol{x} + b = 0 \qquad \{\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{n}}, \quad b \in \mathbb{R}\} \tag{3.5}$$

Deseja-se encontrar o hiperplano ótimo que proporcione a máxima margem possível de divisão entre os dados de classes diferentes, obedecendo à regra de decisão:

$$f(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}.\boldsymbol{x} + b) \tag{3.6}$$

Entre todos os hiperplanos possíveis, existe um único que irá corresponder a este critério (Schölkopf, 2000), que pode ser descrito da seguinte maneira:

$$\max_{\boldsymbol{w},b} \min\{\|\boldsymbol{x} - \boldsymbol{x_i}\| : \boldsymbol{x} \in \mathbb{R}^{\boldsymbol{n}}, \boldsymbol{w}.\boldsymbol{x} + b = 0, i = 1,2,\dots,m\} \tag{3.7}$$

Chamando de x^+ o conjunto de dados para o qual $f(\boldsymbol{x})$ será positivo, e x^- o conjunto de dados para $f(\boldsymbol{x})$ negativo, supõe-se que se deseje encontrar uma *margem funcional* entre os dados e o hiperplano ótimo como mostrado na Figura 16.

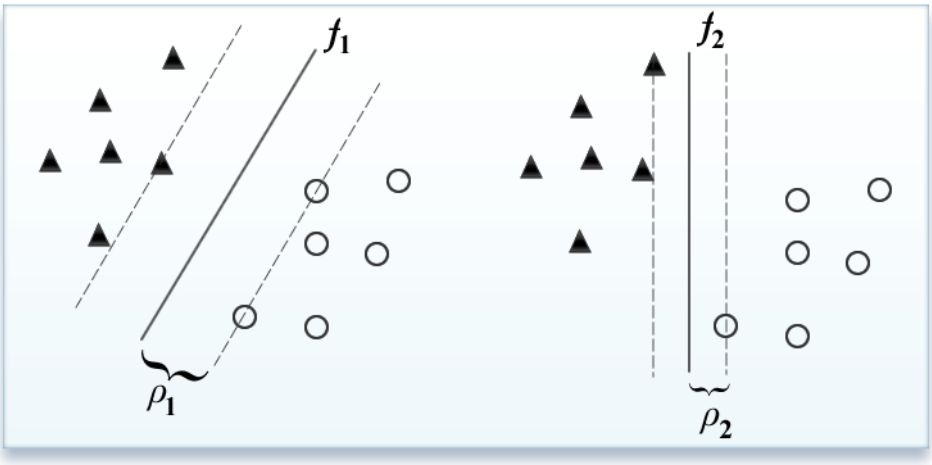

Figura 15 - Definindo a margem do classificador.

Por convenção, esta margem possui valor unitário, de modo que:

$$\boldsymbol{w}.\boldsymbol{x}^+ + b \geq +1 \quad \text{e}$$
$$\boldsymbol{w}.\boldsymbol{x}^- + b \leq -1 \tag{3.8}$$

Nos casos extremos ($\boldsymbol{w}.\boldsymbol{x} + b = \pm 1$), a margem geométrica ρ pode ser encontrada normalizando $\boldsymbol{w}$ (Cristianini & Shawe-Taylor, 2000)**,**

$$\rho = \frac{1}{2}\left(\langle \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}.\boldsymbol{x}^+\rangle - \langle \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}.\boldsymbol{x}^-\rangle\right) =$$
$$= \frac{1}{2\|\boldsymbol{w}\|}(\langle \boldsymbol{w}, \boldsymbol{x}^+\rangle - \langle \boldsymbol{w}, \boldsymbol{x}^-\rangle) =$$
$$= \frac{1}{\|\boldsymbol{w}\|} \tag{3.9}$$

Já que a margem deve ser maximizada, o hiperplano ótimo pode ser encontrando resolvendo-se o seguinte problema de otimização:

$$\text{minimizar } \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{3.10}$$
$$\text{sujeito à } \{ y_i.(\boldsymbol{w}.\boldsymbol{x_i} + b) \geq 1 \, , i = 1,2,\dots,m$$

O problema descrito pode ser mais bem visualizado através Figura 17, que apresenta um caso de classificação bidimensional:

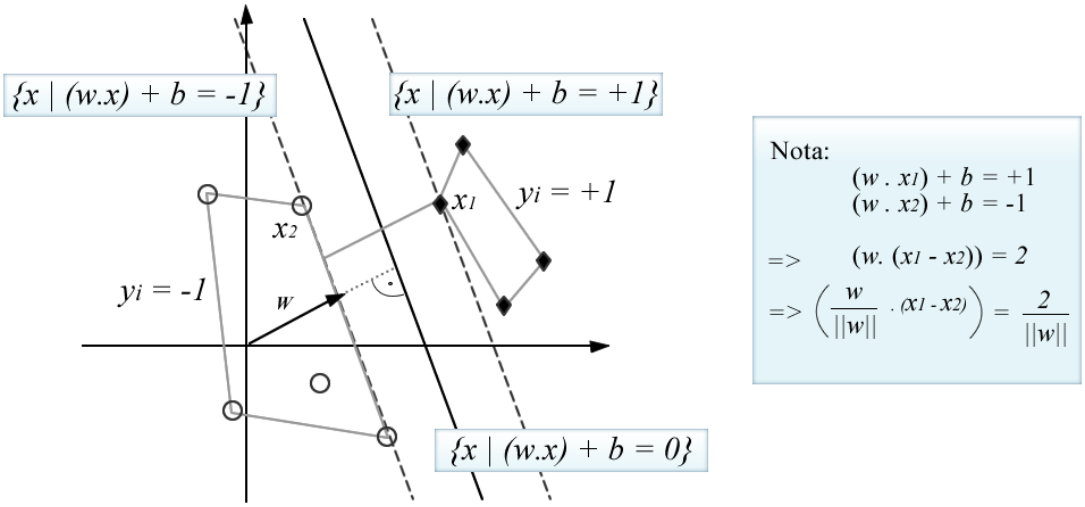

Figura 16 - Classificador de margem rígida. Os vetores de suporte são aqueles situados em cima da margem (Schölkopf, 2000).

Este é o chamado *classificador de margem rígida*, pois é definido por um hiperplano que busca maximizar somente a separação entre conjuntos de dados de padrões diferentes. O problema descrito é um problema de otimização convexa, e pode ser resolvido através da chamada formulação primal do problema, obtida inserindo multiplicadores de Lagrange (um para cada restrição), resultando na seguinte função de custo a ser minimizada:

$$L(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{\mathrm{i}=1}^{m} \alpha_i \left[y_i(\boldsymbol{w}.\boldsymbol{x_i} + b) - 1\right] \qquad (3.11)$$

A solução precisa deste problema não entra no escopo deste trabalho. Apesar disso, parte do procedimento de busca da solução do mesmo permite entender o que são os *vetores de suporte.* Primeiramente, nota-se que nesta nova formulação os dados de treino aparecem somente na forma de produtos internos na função custo. Mais à frente será visto que este é um importante fato para a generalização do algoritmo para classificação de padrões não linearmente separáveis. A função custo deve ser minimizada em relação à $\boldsymbol{w}$ e b e maximizada em relação aos α_i, caracterizando um ponto de sela para o mínimo da função, resultando em:

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{m} y_i \alpha_i = 0 \qquad (3.12)$$

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{m} y_i \alpha_i \boldsymbol{x}_i = 0 \quad \rightarrow \quad \boldsymbol{w} = \sum_{i=1}^{m} y_i \alpha_i \boldsymbol{x}_i \tag{3.13}$$

Uma importante relação em problemas de otimização convexa decorre do teorema demonstrado por Kuhn e Tucker (Cristianini & Shawe-Taylor, 2000), conhecida como condição de complementaridade de Karush-Kuhn-Tucker (KKT), que estabelece, no caso de SVM, que a solução ótima do problema deve obedecer à seguinte relação:

$$\alpha_i . [y_i(\boldsymbol{w} . \boldsymbol{x_i} + b - 1)] = 0, \qquad i = 1, \dots, m \tag{3.14}$$

Esta condição determina que, para as restrições ativas, $\alpha_i > 0$ e desempenha um papel na otimização, enquanto que para as restrições inativas $\alpha_i = 0$, não tendo importância para a otimização. Caso uma das restrições ativas sofra alguma variação, seu α_i correspondente também irá mudar, significando que este representa a sensibilidade da solução ótima face à restrição. Caso ocorra uma mudança em uma restrição inativa, seu correspondente α_i não irá mudar. Finalmente, chega-se a conclusão que o problema de encontrar o hiperplano ótimo de separação entre dois conjuntos de classes diferentes depende somente dos vetores cujas restrições no problema de otimização possuem multiplicadores de Lagrange $\alpha_i > 0$, e, portanto, se dá a esses o nome de vetores de suporte, pois a determinação do hiperplano depende apenas deles.

Substituindo as Equações 3.12 e 3.13 na equação primal (Equação 3.11), obtém-se uma nova função custo, dependente somente dos multiplicadores de Lagrange, chamada formulação dual, resultando finalmente em:

$$\text{maximizar} \quad L(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \boldsymbol{x}, \boldsymbol{x}' \rangle \tag{3.15}$$

$$\text{sujeito à} \quad \alpha_i \geq 0, i = 1, \dots, m \text{ e } \sum_{i=1}^{m} \alpha_i y_i = 0$$

Cuja solução agora é encontrada em função das variáveis duais α_i. A função de decisão toma a seguinte forma:

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{m} \alpha_i y_i \langle \boldsymbol{x}, \boldsymbol{x}' \rangle + b \right) \tag{3.16}$$

Uma analogia (Schölkopf, 2000) (Burges, 1998) pode ser feita entre o classificador apresentado e a mecânica. Através da solução apresentada, pode-se considerar que cada vetor de suporte exerce uma força perpendicular sobre uma folha

rígida localizada sobre o hiperplano ótimo, de modo a manter o sistema sobre equilíbrio. A restrição dada pela Equação 3.12 determina que a soma das forças sobre a folha é igual a zero, e a Equação 3.13 também determina que o torque seja igual a zero, já que

$$\sum_i \boldsymbol{x}_i \times y_i \alpha_i . \boldsymbol{w}/\|\boldsymbol{w}\| = \boldsymbol{w} \times \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} = 0 \qquad (3.17)$$

A função de decisão e a função custo do problema dual dependem somente do produto interno. Esta importante propriedade permite a aplicação de SVM's em espaços de características de dimensão maior que o problema original, que também sejam espaço de produto interno. Este produto interno do espaço de características desempenha um grande papel em SVM's, e permite a aplicação do algoritmo descrito em uma ampla gama de problemas de aprendizagem de máquina.

3.2.3 Kernels

Quando dois padrões não são linearmente separáveis, e sua distribuição no espaço é complexa o suficiente para não permitir o uso adequado de classificadores de hiperplano de separação ótimo, faz-se necessário realizar um mapeamento dos dados de treino em um espaço de características de maior dimensão, de modo que no novo espaço criado eles sejam linearmente separáveis, como mostrados na Figura 18.

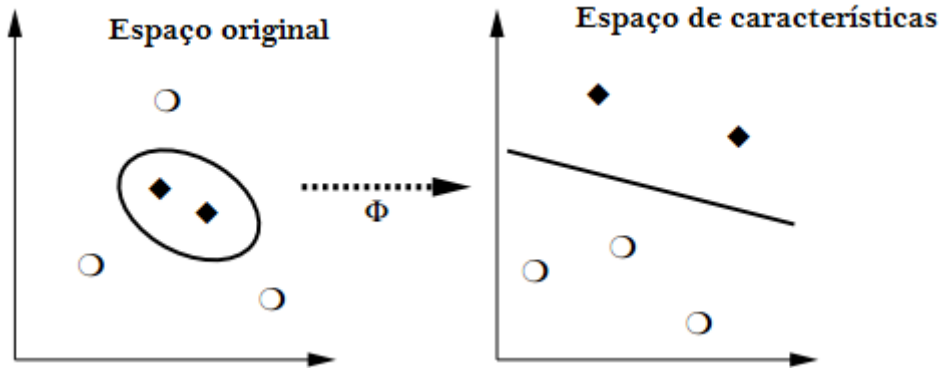

Figura 17 - Mapeamento no espaço de características (Schölkopf, 2000).

O mapeamento é realizado por uma determinada função φ. A realização do mesmo baseia-se no teorema de Cover: Um problema complexo de classificação de padrões, projetado não-linearmente em um espaço de alta dimensão, é mais provável de ser linearmente separável do que em um espaço de menor dimensão, dada que a densidade populacional do novo espaço não é tão grande (Cover, 1965).

Supondo, por exemplo, que se deseje mapear um espaço de dimensão n em um espaço de dimensão d $(d > n)$, usando um polinômio de ordem d. No caso simples do mapeamento de um espaço bidimensional para um espaço tridimensional, tem-se:

$$\boldsymbol{x} = (x_1, x_2) \rightarrow \varphi(x_1, x_2) = (x_1^2, x_2^2, x_1 x_2) = \varphi(\boldsymbol{x}), \tag{3.18}$$

onde $d = 3$. O número de termos (monômios) do vetor original no novo espaço de características será igual a:

$$N = \binom{n + d - 1}{d} \tag{3.19}$$

Para problemas de pequena dimensionalidade, o mapeamento direto não acarreta grandes dificuldades computacionais. No entanto, o uso de uma expressão fatorial mostra que para problemas de maior dimensionalidade (que são os casos mais comuns), o número de monômios dos vetores do espaço de características será muito grande, tornando o mapeamento impraticável em termos computacionais. Por exemplo, no caso de reconhecimento de imagens de 16 × 16 pixels, usando um polinômio de 5º grau para mapeamento, os vetores no espaço de características possuirão 10^{10} monômios. Esta dificuldade sugere o uso de um *mapeamento implícito* para facilitar o problema.

Apesar disso, a teoria de máquinas de vetor de suporte estabelece que o conhecimento ou não da função φ não é necessário para encontrar o hiperplano ótimo, mas sim o produto interno dos diversos dados de treino no espaço de características. A esse produto interno é dado o nome de *kernel*. A construção de hiperplanos ótimos no espaço de características com o auxílio de kernels é a principal ideia das SVM.

De maneira mais intuitiva, o kernel pode ser considerado como uma medida de similaridade num determinado espaço vetorial com produto interno X. Sendo uma medida de similaridade entre dois elementos, o kernel pode ser considerado uma função, na forma:

$$\begin{gathered} k: X \times X \rightarrow \mathbb{R} \\ (x, x') \rightarrow k(x, x') \end{gathered} \tag{3.20}$$

onde

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle \tag{3.21}$$

Em que φ é o mapeamento de X para um determinado espaço de características F. Ou seja, o kernel é uma função que toma dois vetores do espaço X e retorna um número real que mede a similaridade entre os mesmos. Em um espaço euclidiano, o produto interno pode ser considerado como um kernel, ou seja:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle \tag{3.22}$$

Este é o chamado kernel linear, e é utilizado justamente na situação de dados de treino linearmente separáveis. Outros tipos de kernel não-lineares são dados na Tabela 1.

Tabela 1 - Tipos de Kernel.

Kernel	Tipo de classificador
$K(\boldsymbol{x}, \boldsymbol{x}') = (\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + c)^d$	Polinômio de grau d
$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma^2}\right)$	Gaussiano – Função de Base Radial
$K(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\langle \boldsymbol{x}, \boldsymbol{x}' \rangle) - \theta$	Perceptron Multicamadas

O kernel polinomial de ordem 2

$$k(\boldsymbol{x}, \boldsymbol{x}') = (\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + 1)^2, \tag{3.23}$$

por exemplo, pode ser mostrado como o produto de dois mapeamentos polinomiais, considerando o espaço original tendo duas dimensões, como segue abaixo:

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{x}') &= (\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + 1)^2 \\ &= (1 + x_1 x_1' + x_2 x_2')^2 \\ = 1 + 2x_1 x_1' + 2x_2 x_2' + (x_1 x_1')^2 + (x_2 x_2')^2 + 2(x_1 x_1' x_2 x_2') &= \varphi(\boldsymbol{x}).\varphi(\boldsymbol{x}') \end{aligned} \tag{3.24}$$

Onde se pode afirmar que $\varphi(\boldsymbol{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)$

3.2.4 Condição de Existência de um Kernel

A criação de um kernel, como dito anteriormente, não depende do conhecimento da função de mapeamento φ. Será visto adiante que isso ocorre devido ao fato de a regra de decisão para a representação dual

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \alpha_i y_i \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle + b \tag{3.25}$$

depender somente do produto interno $\langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle$, ou seja, do próprio kernel.

O teorema de Mercer determina quando uma função $k(\boldsymbol{x}, \boldsymbol{x}')$ pode ser considerada um kernel (Cristianini & Shawe-Taylor, 2000):

Seja X um espaço de dimensão finita onde $k(\boldsymbol{x}, \boldsymbol{x}')$ é uma função simétrica em X. Então $k(\boldsymbol{x}, \boldsymbol{x}')$ é uma função kernel se e somente se a sua respectiva matriz de Gram

$$\boldsymbol{K} = (k(x_i, x_j))_{i,j=1}^{m} \tag{3.26}$$

For positiva definida, ou seja:

$$\sum_{i,j=1}^{m} c_i c_j K_{ij} \geq 0 \tag{3.27}$$

Para todo $c_i \in \mathbb{R}$.

Respeitada a condição, k é chamado um kernel de Mercer. Em casos onde um determinado kernel não segue a condição de Mercer, o problema de otimização

quadrática pode não ter solução. Ainda assim, caso um kernel que não siga a condição de Mercer resulte numa matriz positiva definida para um determinado conjunto de dados de treino, o problema de otimização quadrática terá uma solução ótima (Burges, 1998). As funções de kernel possuem as seguintes propriedades, dado que K_1 e K_2 são kernels definidos em $X \times X, X \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^+$, $f(.)$ é uma função real em X, e B é uma matriz simétrica positiva semi-definida de dimensão $n \times x$ (Cristianini & Shawe-Taylor, 2000):

1. $K(\boldsymbol{x}, \boldsymbol{y}) = K_1(\boldsymbol{x}, \boldsymbol{y}) + K_2(\boldsymbol{x}, \boldsymbol{y})$
2. $K(\boldsymbol{x}, \boldsymbol{y}) = a.K(\boldsymbol{x}, \boldsymbol{y})$
3. $K(\boldsymbol{x}, \boldsymbol{y}) = K_1(\boldsymbol{x}, \boldsymbol{y}).K_2(\boldsymbol{x}, \boldsymbol{y})$
4. $K(\boldsymbol{x}, \boldsymbol{y}) = f(\boldsymbol{x}).f(\boldsymbol{y})$
5. $K(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T.B.\boldsymbol{y}$

Estas propriedades demonstram a capacidade de reproduzir kernels a partir de outros kernels.

Ainda não se possui uma boa noção teórica de quando um determinado tipo de kernel deve ou não ser aplicado em um problema específico, e a aplicação de um kernel em um problema complexo é um fator de sucesso ou fracasso na realização do mesmo, sendo este um grande trunfo e ao mesmo tempo um limitador para o SVM (Burges, 1998). Ainda assim em muitos casos o uso de diferentes kernels pode resultar em ótimos resultados. O conceito de kernel como medida de similaridade é bastante amplo, e possui muitas propriedades demonstradas (Hastie, Tibshirani, & Friedman, 2009).

3.2.5 Classificadores de Vetor de Suporte

Dada a noção de hiperplano de separação ótimo e kernels, é possível agora fornecer a representação geral de SVM, que é obtida realizando a substituição os produtos internos da formulação dual por kernels. Também serão adicionadas variáveis de folga ε_i ao problema (citar o problema primal), já que em casos reais é comum a ocorrência de algumas amostras muito próximas à amostra de outra classe, de modo a não respeitar as restrições (restrições do problema primal), como mostra a Figura 19.

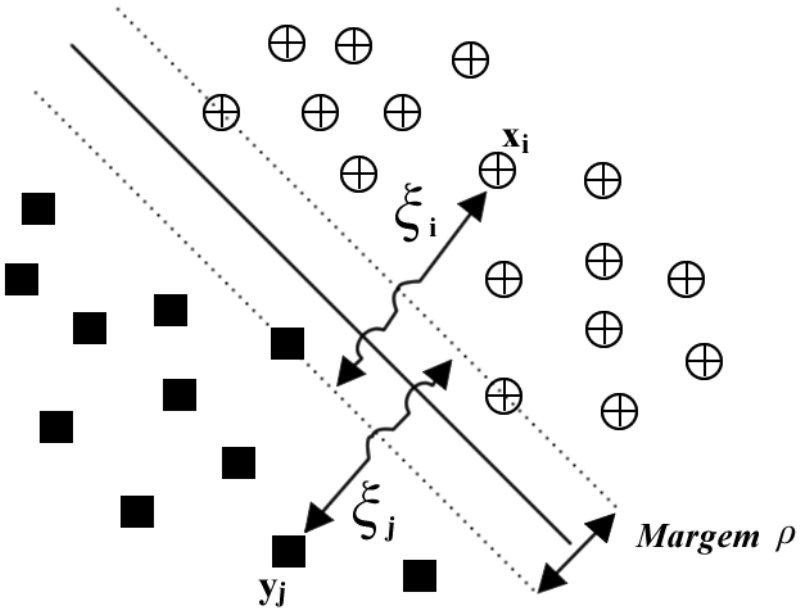

Figura 18 - Variáveis de folga .

A inclusão das variáveis de folga deixa as restrições da seguinte forma:

$$y_i.\left((\boldsymbol{w}.\boldsymbol{x_i}) + b\right) \geq 1 - \varepsilon_i, \qquad i = 1, \dots, m \tag{3.28}$$

Os ε_i representam o erro de cada amostra em relação ao hiperplano de separação. Incorporando as variáveis de folga ao problema original, chega-se à seguinte função custo:

$$\text{minimizar} \quad L(\boldsymbol{w}, b, \varepsilon) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\varepsilon_i \tag{3.29}$$

$$\text{sujeito à} \begin{cases} y_i.\left((\boldsymbol{w}.\boldsymbol{x_i}) + b\right) \geq 1 - \varepsilon_i, & i = 1, \dots, m \\ \varepsilon_i \geq 0 & i = 1, \dots, m \end{cases}$$

Este é o chamado *classificador de margem suave*, pois nele não se busca somente encontrar o hiperplano de máxima separação, mas também reduzir o erro das amostras de treino que estão erroneamente classificadas. As condições de complementaridade de Karush-Kuhn-Tucker determinam que, para este classificador, para as amostras de treino que se encontram do lado correto do hiperplano (amostras corretamente classificadas), seu respectivo ε_i será igual a zero, enquanto aquelas que se encontram do lado errado do hiperplano terão seu ε_i minimizado de modo a encontrar um balanceamento entre o erro de classificação dos dados de treino e a máxima separação possível entre as duas classes. O parâmetro C é o responsável por controlar o peso que os ε_i terão na otimização: quanto maior o seu valor, mais importância será dada à minimização do erro de treino. Esta função custo mostra demonstra bem a capacidade de SVM de encontrar um balanceamento entre a complexidade da hipótese (através do termo $\frac{1}{2}\|\boldsymbol{w}\|^2$) e a minimização do risco empírico (dado por $C\sum_{i=1}^{m}\varepsilon_i$).

Ao passar o problema para a forma dual, e utilizando kernels, obtém-se:

$$\text{maximizar} \quad L(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}, \boldsymbol{x}') \tag{3.30}$$

$$\text{sujeito à} \quad 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m \text{ e } \sum_{i=1}^{m} \alpha_i y_i = 0$$

cuja função de decisão continua sendo a mesma

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{m} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x_i}) + b\right) \tag{3.31}$$

A presença do fator C na restrição dos α_i restringe a influência de outliers (dados que não correspondem à estatística padrão do conjunto total), padrões que na verdade não representem um comportamento normal da classe. Para o cálculo de b, volta-se novamente à condição de Karush-Kuhn-Tucker, fazendo, para qualquer vetor $\boldsymbol{x_i}$:

$$\sum_{j=1}^{m} \alpha_j y_j k(\boldsymbol{x_i}, \boldsymbol{x_j}) + b = y_i \tag{3.32}$$

A fim de evitar problemas numéricos, recomenda-se utilizar a média do valor de b obtido no conjunto inteiro de dados de treino. Percebe-se que a função custo necessita dos dados de treino apenas para o cálculo do kernel, logo, os mesmos dados precisam entrar no algoritmo apenas na forma da matriz de Gram $\boldsymbol{K}$ (Equação 3.26)**.**

$$\begin{pmatrix} K_{11} & \cdots & K_{1m} \\ \vdots & \ddots & \vdots \\ K_{m1} & \cdots & K_{mm} \end{pmatrix} \tag{3.33}$$

O problema de otimização na forma acima descrita é a forma mais comum do SVM em reconhecimento de padrões.

A Figura 20 é um exemplo da capacidade do SVM em lidar com dados não linearmente separáveis, ilustrando a ideia centra da técnica. Nela, um kernel gaussiano foi usado para separar o conjunto de bolas vazadas das bolas opacas. Percebe-se que, se forem tomados os dados no espaço original, não é possível traçar um hiperplano de separação dividindo os dois conjuntos. Foi usado então o kernel gaussiano (que pertence a uma classe maior de funções chamadas de funções de base radial) para aumentar a dimensão do problema. As curvas tracejadas são as projeções do hiperplano ótimo de separação (encontrado no espaço de características) no espaço original dos dados, e a

região em branco é a margem projetada do espaço de características. As amostras que contém um círculo ao redor indicam os vetores de suporte.

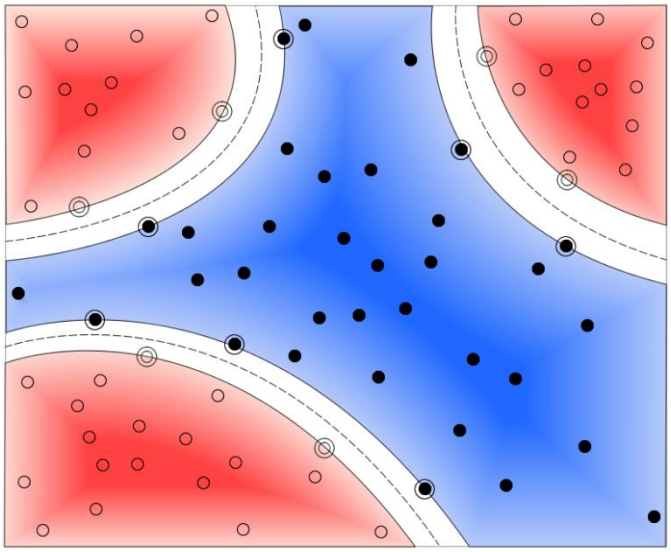

Figura 19 - Aplicação do SVM em classificação de dados não linearmente separáveis.

CAPÍTULO 4

4 METODOLOGIA

O objetivo deste trabalho é classificar corretamente qual o tipo de ruído está presente na parte do cobre de um enlace VDSL2. É importante considerar que neste trabalho, embora seja uma simplificação, é considerado que cada tipo de ruído ocorre em momentos diferentes (não são simultâneos). Os tipos de ruído a serem classificados são:

- Ruído de fundo;
- *Crosstalk* (NEXT/FEXT);
- Ruído elétrico repetitivo impulsivo;
- Ruído RFI.

A escolha dos tipos de ruído foi feita com base na norma G-993.1 da ITU (G.993.1, 2004).

O procedimento para realizar o trabalho divide-se em dois momentos:

1. Montagem e execução do cenário de medição para extração das MIB.
2. Aplicação do algoritmo de aprendizagem de máquina (SVM) para realizar a classificação do ruído.

4.1 Cenário de Medição

A montagem do cenário de medição baseia-se em parte na norma G-993.1 (G.993.1, 2004). O esquema geral de medição é exemplificado pela Figura 22.

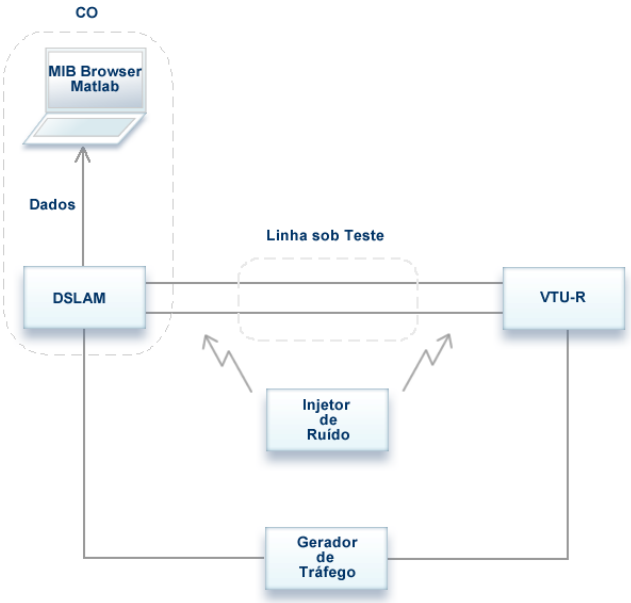

Figura 20 - Disposição dos equipamentos no cenário de medição.

A descrição de como cada equipamento é utilizado no cenário de medições segue abaixo:

DSLAM: Objetiva-se realizar a classificação do ruído através do DSLAM. Para simular esta situação, um computador foi conectado a ele para coletar os dados das medições (através do software MIB Browser, descrito mais adiante), e realizar a classificação do ruído (através do MATLAB). O modelo do DSLAM usado foi o Ericsson EDN312xp.

Gerador de tráfego: é o equipamento que simulará a geração de tráfego no enlace experimental. É ligado ao DSLAM e ao usuário para efetuar um tráfego bidirecional. O tráfego gerado é do tipo HTTP. A taxa de transmissão foi escolhida de acordo com a taxa do VDSL2 assimétrico. O modelo utilizado para o gerador foi o SPIRENT AX-4000.

Gerador de ruído: contém os arquivos de ruído dos diversos tipos descritos, e permite simular a presença do mesmo no enlace. Será ligado diretamente ao enlace. O ruído foi injetado tanto do lado da central como do lado do usuário. Os arquivos de ruído utilizados são dados no Apêndice C. O modelo do gerador de ruído utilizado foi o DLS 5500 da Spirent Communications.

VTU-R (VDSL Terminal Unit - Remote): Representa o usuário na outra extremidade do enlace VDSL2.

Linha sob Teste: é o enlace telefônico físico real usado para os experimentos. Está ligado ao DSLAM, ao modem (que representa o usuário remoto), e também ao injetor de ruído, já que deve sofrer influência do mesmo. No presente trabalho, os comprimentos de enlace selecionados são 50, 150, e 450 metros, respeitando os limites do VDSL2. As bitolas dos cabos escolhidas no para medição foram as de 0,4 mm e 0,5 mm, por serem mais as usadas (Golden, Dedieu, & Jacobsen, 2004). É importante notar que neste trabalho serão usados enlaces de uma única seção de cabo ligando os terminais.

As medições foram realizadas utilizando os comprimentos de enlace e bitola do cabos descritos na Tabela 2.

Tabela 2 -Tipos de ruído e enlaces utilizados.

Enlace	Tipo de Ruído	Comprimento(m)
Enlace 1 (bitola 0,4 mm)	*Crosstalk*	50,150,450
	REIN	
	RFI	
	Ruído de Fundo	
Enlace 2 (bitola 0,5 mm)	*Crosstalk*	
	REIN	
	RFI	
	Ruído de Fundo	

O *crosstalk* foi injetado com base no número de interferentes e na potência do ruído, conforme a Tabela 3.

Tabela 3 - Variações de *crosstalk* utilizados.

Número de interferentes	Potência do ruído
5	-25.6dBm
10	-23.4dBm
15	-22.4dBm
20	-21.6dBm
25	-21.1dBm
30	-20.6dBm
35	-20.2dBm
40	-19.8dBm
45	-19.5dBm
49	-19.3dBm

O ruído impulsivo utilizado tem potência de 0 dBm, e o ruído RFI escolhido possui potências de -44 dBm e -54 dBm. A situação do ruído de fundo foi criada não injetando artificialmente nenhum tipo de ruído nos cabos reais.

A medição ocorre na sequência descrita:

1. O modem e o DSLAM devem estar devidamente sincronizados, e o gerador de tráfego deve estar realizando a transmissão.

2. O tipo de ruído selecionado é injetado no enlace escolhido.
3. A extração de cada amostra das MIB é realizada de 30 em 30 segundos, até um total de 30 amostras, totalizando 15 minutos de medição.
4. Ao fim da extração das amostras, o tráfego é encerrado, a injeção de ruído é interrompida e o modem é desligado.

4.2 Aplicação do Algoritmo de Aprendizagem

Estabelecido o método de obtenção dos dados, a parte mais física do problema, é necessário em seguida estabelecer as etapas do processo de aprendizagem de máquina.

4.2.1 Ferramentas Computacionais

A ferramenta escolhida como fonte dos dados foi o MIB Browser. O MIB Browser é um programa gratuito que obtém suas métricas MIB de um determinado hardware através do protocolo SNMP. A ferramenta escolhida para realizar a aplicação do SVM foi o MATLAB, pela facilidade do mesmo em lidar com matrizes e álgebra linear. Existe um bom número de softwares onde SVM`s são implementados: LIBSVM (Chang & Lin, 2001), SVMLight (Joachims, 2008), SVMTorch (Collobert & Bengio, 2001) etc., porém, por fins de aprendizagem, e com o propósito específico de utilizar a técnica para a classificação de ruído, optou-se realizar uma implementação própria no MATLAB. De modo a obter as métricas MIB mostrados em tempo real no MIB Browser, foi usado o programa SNMPGet (SNMPGET, 2009) junto à um script MATLAB criado com o propósito de armazenar as métricas em arquivos de extensão “.csv”.

4.2.2 Fase de Seleção dos Dados Relevantes

Para tanto, os dados, que estão originalmente contidos em um arquivo “.csv”, serão convertidos em matrizes do MATLAB (arquivos.m). Existe um universo de 59 métricas MIB. A disposição das métricas MIB no arquivo “.csv” está de acordo com a Figura 23, onde cada coluna representa os valores em sequência de uma métrica, e cada linha representa uma amostra.

	A	B	C	D	E	F	G	H	I	J	K
90	325	71	86	15	6	1	5	1	3	53	13
91	324	71	86	15	6	1	5	1	3	53	13
92	326	71	86	15	6	1	5	1	3	53	13
93	326	71	86	15	6	1	5	1	3	53	13
94	325	71	86	15	6	1	5	1	3	53	13
95	196	60	89	20	9	4	17	4	3	132	52
96	287	60	89	20	9	4	17	4	3	132	52
97	285	60	89	20	9	4	17	4	3	132	52
98	281	60	89	20	9	4	17	4	3	132	52
99	284	60	89	20	9	4	17	4	3	132	52
100	282	60	89	20	9	4	17	4	3	132	52
101	283	60	89	20	9	4	17	4	3	132	52
102	285	60	89	20	9	4	17	4	3	132	52
103	284	60	89	20	9	4	17	4	3	132	52
104	285	60	89	20	9	4	17	4	3	132	52
105	283	60	89	20	9	4	17	4	3	132	52
106	289	60	89	20	9	4	17	4	3	132	52
107	283	60	89	20	9	4	17	4	3	132	52
108	195	60	89	20	10	14	20	5	4	156	66
109	278	60	89	20	10	14	20	5	4	156	66
110	280	60	89	20	10	14	20	5	4	156	66
111	282	60	89	20	10	14	20	5	4	156	66
112	283	60	89	20	10	14	20	5	4	156	66
113	279	60	89	20	10	14	20	5	4	156	66
114	282	60	89	20	10	14	20	5	4	156	66

Figura 21 Arquivo ".csv" contendo as MIB.

De modo a escolher as métricas mais relevantes para o processo de aprendizagem, o conjunto inteiro de dados medidos foi analisado, e optou-se pelo seguinte método de seleção:

- As métricas que possuem variância nula no conjunto total de amostras são eliminadas, pois a covariância de cada uma delas com os quatro tipos de ruído é nula, se considerarmos estes como variáveis aleatórias.
- A matriz dos coeficientes de correlação linear das métricas foi calculada, tomando por base o conjunto inteiro de amostras, e procurou-se selecionar aquelas que tivessem grande correlação com o maior número de métricas possível sem ser correlacionadas entre si. O coeficiente de correlação linear é calculado através da seguinte Equação 4.1, onde X e Y representam os valores de duas métricas quaisquer:

$$\rho_{XY} = \frac{cov(X,Y)}{\sqrt{var(X)var(Y)}} \tag{4.1}$$

O Apêndice A contém a tabela 11, listando todas as 59 métricas MIB escolhidas inicialmente, assim como uma explicação das convenções de nomenclatura utilizadas nas MIB em DSL.

Realizado o processo de seleção, foram escolhidas as seguintes métricas, descritas em (Ericsson, 2009):

adslAturCurrSnrMgn: É a margem de ruído do lado das instalações do cliente com relação ao respectivo sinal recebido em décimos de dB.

adslAturCurrOutputPwr: Medida da potência total de saída transmitida pelas instalações do usuário.

adslAtucPerfCurr1DayESs: contagem dos segundos de erro durante o dia corrente. O segundo de erro é um parâmetro que conta o número de intervalos de um segundo contendo uma ou mais anomalias no código de redundância cíclica, ou um ou mais defeitos de perda de sinal ou *frame* severamente errado.

4.2.3 Fase de Treinamento do SVM

Nesta fase busca-se realizar o treinamento do algoritmo de modo a determinar os vetores de suporte que irão permitir a classificação dos dados de teste e validação.

Selecionadas as métricas, o conjunto de treino foi separado do conjunto de teste, usando 25% dos dados para treino e 75% para teste, optando por deixar os dois conjuntos com dados de medições diversificadas, e não com muitos dados de poucas medições, de modo a garantir ao algoritmo de classificação melhores chances de generalização. A escolha não comum de apenas 25% do conjunto total para os dados de treino se deve ao fato de que a otimização quadrática realizada pelo MATLAB não era resolvida corretamente para matrizes de Gram relativamente grandes(com dimensão maior que 250x250), o que implicou neste reduzido conjunto de treino.

Após a seleção, os dados das métricas foram utilizados para encontrar os vetores de suporte que determinarão como os mesmos estão divididos no espaço multidimensional. A obtenção desses vetores depende da escolha do kernel a ser utilizado no problema. Optou-se então por utilizar o kernel linear, o gaussiano, e o polinomial para avaliar o desempenho do SVM. Como existem quatro tipos de ruído diferentes, e o SVM é um classificador binário, decidiu-se realizar a classificação de novas amostras utilizando o método “todos contra todos”. Neste método, é realizada uma classificação para cada dois padrões diferentes, e escolhido é aquele que obteve o maior número de amostras corretamente classificadas. Ressalta-se aqui que não houve normalização das amostras em momento algum do processo de aprendizagem de máquina. A Figura 24 ilustra esta fase:

Figura 22 - Fase inicial da treinamento para determinação dos vetores de suporte.

4.2.4 Fase de Classificação

No segundo momento, busca-se realizar a classificação dos dados de teste, onde as métricas coletadas serão continuamente classificadas, indicando para cada amostra do conjunto qual o tipo de ruído está presente no enlace. Para esta fase, coleta-se apenas os dados das métricas relevantes na fase 1, que serão operados em conjunto com os vetores de suporte (também obtidos na fase 1), e realiza-se a classificação dos mesmos. A Figura 25 ilustra esta fase:

Figura 23 - Fase de classificação.

CAPÍTULO 5

5 RESULTADOS

Para apresentar os resultados obtidos na classificação dos dados, optou-se por utilizar a matriz de confusão, comumente utilizada em problemas de classificação. Para auxiliar na compreensão dos valores usados para medir o desempenho do SVM, a tabela de confusão foi dada na Tabela 4, cujos elementos são os seguintes:

Tabela 4 - Tabela de confusão.

Classe	Preditos como (+1)	Preditos como (-1)
+1	Verdadeiros Positivos (Tp)	Falsos Negativos (Fn)
-1	Falsos Positivos (Fp)	Verdadeiros Negativos (Tn)

- Verdadeiros positivos (Tp): número de elementos que possuem classe +1 e que foram corretamente classificados com a classe +1 pelo SVM.
- Falsos positivos (Fp): número de elementos que possui classe -1 e que foram erroneamente classificados com a classe +1 pelo SVM.
- Falsos negativos (Fn): número de elementos que possui classe +1 e que foram erroneamente classificados com a classe -1 pelo SVM.
- Verdadeiros negativos (Tn): número de elementos que possui classe -1 e que foram classificados com a classe -1 pelo SVM.

Esta matriz permite o cálculo de inúmeras medidas de desempenho do classificador. As medidas de desempenho escolhidas para o este trabalho foram as seguintes:

Exatidão: proporção do número total de predições corretas. Calculado através da seguinte Equação 5.1:

$$AC = \frac{(Tp + Tn)}{Tp + Fp + Tn + Fn} \tag{5.1}$$

Precisão: proporção dos casos preditos positivos que estão corretos.

$$P = \frac{Tp}{Tp + Fp} \tag{5.2}$$

Taxa de verdadeiros positivos: proporção de casos positivos que foram corretamente classificados:

$$Taxa_{Tp} = \frac{Tp}{Tp + Fn} \tag{5.3}$$

Ressalta-se que o desempenho do classificador será avaliado para o problema de classificação de ruído como um todo e também para cada classe específica. Exceção é feita à precisão, pois ela existe apenas para as classes específicas. As matrizes de confusão descritas para os kernels linear, gaussiano, e polinomial de ordem 2 (Tabelas 5 – 10), representam os resultados para os dados de teste do problema. Os parâmetros dos kernels escolhidos correspondem ao melhor desempenho alcançado, com $C = 2$ (variável que controla o risco empírico no SVM).

Tabela 5 – Matriz de confusão para o kernel Gaussiano (com σ=2).

Predição / **Classe**	***Crosstalk***	**REIN**	**RFI**	**Ruído de Fundo**	**Taxa de Verdadeiros Positivos**
Crosstalk	450	3	0	0	99,33%
REIN	2	59	0	0	96,29%
RFI	0	0	277	33	88,08%
Ruído de Fundo	0	0	0	164	100,00%
Média					95,92%

Tabela 6 - Exatidão e precisão para o kernel gaussiano.

	Exatidão	**Precisão**
Geral	96,15%	--
Crosstalk	98,90%	99,55%
REIN	92,18%	96,72%
RFI	89,35%	89,35%
Ruído de Fundo	83,24%	83,24%

Tabela 7 Matriz de confusão para o kernel Polinomial (com d=2).

Predição / Classe	*Crosstalk*	REIN	RFI	Ruído de Fundo	Taxa de Verdadeiros Positivos
Crosstalk	449	4	0	0	99,11%
REIN	0	61	0	0	100%
RFI	0	0	281	29	90,64%
Ruído de Fundo	0	0	20	144	87,80%
Média					94,38%

Tabela 8 - Exatidão e precisão para o kernel polinomial.

	Exatidão	Precisão
Geral	94,63%	--
Crosstalk	99,11%	100%
REIN	93,84%	93,84%
RFI	85,15%	93,35%
Ruído de Fundo	74,61%	83,23%

Tabela 9 – Matriz de confusão para o kernel linear.

Predição / Classe	*Crosstalk*	REIN	RFI	Ruído de Fundo	Taxa de Verdadeiros Positivos
Crosstalk	449	4	0	0	99,11%
REIN	0	61	0	0	100%
RFI	0	0	261	49	84,19%
Ruído de Fundo	0	0	16	148	90,24%
Média					93,38%

Tabela 10 - Exatidão e precisão para o kernel linear.

	Exatidão	Precisão
Geral	93,01%	--
Crosstalk	99,11%	100%
REIN	93,84%	93,84%
RFI	80,06%	94,22%
Ruído de Fundo	69,48%	75,12%

O kernel gaussiano foi aquele que obteve as melhores medidas de desempenho, seguido do kernel polinomial de ordem 2, e por último o kernel linear. A exatidão geral dos 3 kernels foi muito semelhante, com valores acima de 90%. Este resultado se deve bastante ao fato de que a classificação do *crosstalk* e do REIN provou-se tarefa simples, já que os dois obtiveram precisão e Exatidão acima de 90% nos três casos. Por outro lado, a classificação do REIN e do RFI foi menos satisfatória, com exatidão e precisão entre 80% e 90% em média, com um desempenho um pouco melhor para o RFI. O desempenho mais fraco do kernel linear pode ser explicado pelo fato de que, em situações de aplicação real de SVM's, supor que os dados estejam linearmente separados seria um caso bastante otimista. Uma última análise é feita para a taxa de verdadeiros positivos. Os quatro tipos de situações de ruído consideradas possuem um comportamento em regime permanente, sendo que as rajadas do REIN atuam periodicamente (Golden, Dedieu, & Jacobsen, 2004). Logo, em um cenário VDSL2 no qual somente um tipo de ruído ocorra por vez, o normal é que haja bastantes amostras consecutivas de um mesmo tipo, e como a taxa de verdadeiros positivos foi alta, espera-se que a classificação de ruído tenha um bom desempenh

CAPÍTULO 6

6 CONCLUSÃO

Os experimentos em laboratório mostram que o objetivo inicial da classificação do ruído em uma rede VDSL2 é realizável. O reduzido número de dados de treino e os resultados satisfatórios obtidos nas medições realizadas em diferentes cenários mostram também que cada tipo de ruído influencia a estatística das métricas MIB de uma maneira bastante característica, o que facilita o processo de aprendizagem de máquina. Atenta-se também para o fato de que os resultados obtidos não podem ser considerados como gerais, podendo a correta classificação ser alcançada seguindo o esquema proposto na metodologia das medições.

Em relação ao método proposto, foi mostrado, com base nos resultados alcançados, que a aplicação de técnicas de aprendizagem de máquina às métricas MIB pode gerar mais conhecimento sobre a rede do que aquele sendo explícito pelo valor delas. O fato de essas técnicas computacionais serem relativamente simples de serem aplicadas e não interferirem no oferecimento do serviço é um atrativo para as operadoras de telefonia e fornecedores do serviço VDSL2.

6.1 Propostas de Trabalhos Futuros

- O ruído *crosstalk*, sendo predominante em DSL, poderia ser analisado mais profundamente através da sua influencia sobre a estatística das métricas MIB. Para tanto, possivelmente o número de métricas a ser analisado deverá ser aumentado. Informações como faixa de freqüência, número de tons afetados, bit loading, e potência do ruído poderiam ser estimadas utilizando técnicas de aprendizagem de máquina, tornando mais rica a classificação de ruído.
- Aplicação do método proposto e do classificador obtido em linhas telefônicas em redes envolvendo usuários reais.
- A classificação desenvolvida neste trabalho envolvia os dados obtidos apenas para um par de modems, VTU-O e seu respectivo VTU-R. Redes de computadores, como o próprio nome já diz, envolvem uma grande quantidade de usuários e terminais. Caso se aplique técnicas de aprendizado de máquina em

dados que envolvam informações de diversos enlaces, é possível estudar o caso de gerenciamento inteligente da rede VDSL2 em maior escala.

- Realizada a classificação para o caso de somente um tipo de ruído por vez, o passo seguinte seria a realização da classificação de mais de um tipo de ruído ocorrendo simultaneamente na rede, tornando o método proposto mais abrangente.

REFERÊNCIAS BIBLIOGRÁFICAS

Broadband, F. Acesso em 19 de outubro de 2011, disponível em Broadband Forum: http://www.broadband-forum.org/downloads/About_DSL.pdf, 2008

Brost, R., & Aspell, S. "ADSL Interoperability Test Plan", 2002

Burges, C. J. "A Tutorial on Support Vector Machines for Pattern Recognition". Bell Laboratories, Lucent Technologies, 1998

Cendrillon, R. "Multi-user Signal and Spectra Co-ordination for Digital Subscriber Lines",2004

Chang, C.-C., & Lin, C.-J. "LIBSVM: a library for support vector machines", 2001

Collobert, R., & Bengio, S. "SVMTorch: support vector machines for large-scale regression problems". The Journal of Machine Learning Research, 2001

Cover, T. M.. "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition". IEEE Transactions on Electronic Computers, Julho de 1965

Cristianini, N., & Shawe-Taylor, J. "An Introduction to Support Vector Machines and Other Kernel-based Learning Method", 2000.

Cui-Mei, B. "Intrusion Detection Based on One-class SVM and SNMP MIB data" . Fifth International Conference on Information Assurance and Security, (p. 4), 2009

Dunford, C. "Measuring NEXT and FEXT". Telecom Test and Measurement. 2008

Ericsson. "EDN312x, EDN312, EDN110 Proprietary MIB Description EDA 1200", 2009

Eriksson, P.-E., & Odenhammar, B. "VDSL2: Next important broadband technology", Ericsson, 2006

Farias, F. S., Borges, G. S., Moritsuka, N. S., Costa, J. C., Francês, C. R., Souza, L. V., et al. "Noise Estimation in DSL Networks using Linear Regression and Fuzzy Systems",2011

G.993.1, I.-T. "Very high speed digital subscriber line transceivers", International Telecommunication Union, 2004

Gaïti, D. "Intelligence dans les réseaux". Lavoisier, 2005

Galli, Stefano; Valenti, Craig; "A Frequency-Domain Approach to Crosstalk Identification in xDSL Systems", 2001

Gazineu, D. S. "VISDAMAGE Ferramenta de Mineração Visual de Dados Aplicada à Gerência de Redes". Trabalho de Conclusão de Curso, Graduação em Sistemas de Informação, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2007

Ginsburg, D. "Implementing ADSL", 1999

Golden, P., Dedieu, H., & Jacobsen, K. S. "Fundamentals of DSL Technology" (1 ed.). Auerbach Publications, 2004

Hastie, T., Tibshirani, R., & Friedman, J. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction". (2 ed.). Springer. 2009

Joachims, T., "SVMLight: Support Vector Machine". Acesso em 8 de Novembro de 2011, disponível em SVMLight: http://svmlight.joachims.org/, 2008

Kulkarni, P. G., McClean, S. I., Parr, G. P., & Black, M. M., "Deploying MIB Data Mining for Proactive Network Management". 3rd International IEEE Conference Intelligent Systems, 2006

Li, J., & Manikopoulos, C."Early Statistical Anomaly Intrusion Detection of DOS Attacks Using MIB Traffic Parameters". *IEEE* , 2003

MacKay, D. J.,"Information Theory, Inference, and Learning Algorithms", 2003

Mitchell, T. M., "Machine Learning", McGraw Hill, 1997

Nedev, N. H. "Analysis of the Impact of Impulse Noise in Digital Subscriber Line Systems". The University of Edinburgh, 2003

Papandriopoulos, J. (s.d.). Acesso em 29 de Novembro de 2011, disponível em John Papandriopoulos: http://jpap.andriopo.ulos.org/

Papandriopoulos, J., & Evans, J. S. "SCALE: A Low-Complexity Distributed Protocol for Spectrum Balancing in Multiuser DSL Networks". IEEE Transactions on Information Theory , *55*, 2009

Patrício, É. T. "Software para Qualificação de Enlaces em Sistemas xDSL: Abordagem por Algoritmos Genéticos", Trabalho de Conclusão de Curso, Graduação em Engenharia da Computação, Universidade Federal do Pará - UFPA, Belém, 2006

Schölkopf, B. "Statistical Learning and Kernel Methods". Microsoft Research, 2000

Shannon, C. E. "A Mathematical Theory of Communication". The Bell System Technical Journal , *27*, 1948

Siqueira, R. G. (2010). "Analysis and Mitigation of the Effect of Repetitive Impulsive Noises on Digital Subscriber Lines". Trabalho de Conclusão de Curso, Graduação em Engenharia da Computação, Universidade Federal de Pernambuco, Recife.

SNMPGET. (2009). Acesso em 21 de Dezembro de 2011, disponível em SNMPGET: http://www.net-snmp.org/docs/man/snmpget.html

Vapnik, V. N. "Statistical Learning Theory" . John Wiley & Sons, Inc., 1998

Weston, J. "Support Vector Machine (and Statistical Learning Theory) Tutorial". NEC Labs America, Princeton, USA.

Wu, X., *et al.*, *"*Top 10 Algorithms in Data Mining*"*, Knowledge and Information Systems, 14, 2008

Yang, Z., Dasgupta, U., Redfer, A., & Ali, M. "Noise Identification in ADSL Modems"

APÊNDICE A – CONJUNTO TOTAL DAS MÉTRICAS MIB

Tabela 11 Conjunto das 59 métricas MIB selecionadas

Métricas MIB	Métricas MIB
adslIfAdminStatus	adslAturPerfCurr1DayLoss
adslIfOperStatus	adslAturPerfCurr1DayLprs
adslAtucChanCurrTxRate	adslAtucPerfCurr1DayLols
adslAturChanCurrTxRate	adslAtucPerfCurr1DayInits
adslAtucCurrAttainableRate	adslAtucChanConfInterleaveMaxTxRate
adslAturCurrAttainableRate	adslAturChanConfInterleaveMaxTxRate
adslAtucCurrSnrMgn	adslAtucChanConfInterleaveMinTxRate
adslAturCurrSnrMgn	adslAturChanConfInterleaveMinTxRate
adslAtucCurrAtn	adslAtucConfMaxSnrMgn
adslAturCurrAtn	adslAturConfMaxSnrMgn
adslAtucCurrOutputPwr	adslAtucConfTargetSnrMgn
adslAturCurrOutputPwr	adslAturConfTargetSnrMgn
adslAtucChanInterleaveDelay	adslAtucConfMinSnrMgn
adslAturChanInterleaveDelay	adslAturConfMinSnrMgn
adslAtucProprietaryChanActualInp	adslAtucChanConfMaxInterleaveDelay
adslAturProprietaryChanActualInp	adslAturChanConfMaxInterleaveDelay
adslAtucChanPerfCurr1DayUncorrectBlks	adslAtucProprietaryChanConfXINPminIlvB0
adslAturChanPerfCurr1DayUncorrectBlks	adslAturProprietaryChanConfXINPminIlvB0
adslAtucPerfCurr1DayESs	adslAtucProprietaryPhysXActualLineBitRate
adslAturPerfCurr1DayESs	adslAturProprietaryPhysXActualLineBitRate
adslAtucPerfCurr1DaySesL	loopDiagLoopAttenuationFE
adslAturPerfCurr1DaySesL	loopDiagLoopAttenuationNE
adslAtucChanCrcBlockLength	loopDiagSignalAttenuationFE
adslAturChanCrcBlockLength	loopDiagSignalAttenuationNE

adslAtucChanPerfCurr1DayCorrectedBlks	loopDiagSnrMarginFE
adslAturChanPerfCurr1DayCorrectedBlks	loopDiagSnrMarginNE
adslAtucPerfXCurr1DayEcs	loopDiagAttainableBitRateFE
adslAturPerfXCurr1DayEcs	loopDiagAttainableBitRateNE
adslAtucPerfCurr1DayUasL	adslLineXStatusActPsdUs
adslAturPerfCurr1DayUasL	adslLineXStatusActPsdDs
adslAtucPerfCurr1DayLoss	-

A nomenclatura (incompleta) utilizada para descrever as métricas segue a seguinte convenção:

- Atuc : unidade terminal na central telefônica.
- Atur : unidade terminal na extremidade do usuário (remota).
- Curr: corrente (adjetivo).
- Prev: precedente.
- Atn: Atenuação.
- Es: Segundos com erro.
- Lof: Perda de *frame*
- Lol: Perda de ligação
- Los: Perda de sinal
- Lpr: Perda de potência
- Max: Máximo
- Min: Mínimo
- Mgn:Margem
- Psd: Densidade espectral de potência
- Snr: Razão Sinal Ruído
- Tx: Transmissor

- Rx: Receptor
- Blks: Blocos

APÊNDICE B – DIAGRAMAS DE DISPERSÃO PARA O CASO DO *CROSSTALK*

De modo a ter uma noção do comportamento estatístico das métricas MIB, decidiu-se plotar o diagrama de dispersão 134 amostras de *crosstalk* de 12 métricas (por simplificação). As 134 medições foram escolhidas de variados cenários de transmissão VDSL2 utilizados neste trabalho:

1. adslAtucCurrSnrMgn;
2. adslAturCurrSnrMgn;
3. adslAturCurrOutputPwr;
4. adslAtucChanPerfCurr1DayUncorrectBlks;
5. adslAtucPerfCurr1DayESs;
6. adslAtucChanPerfCurr1DayCorrectedBlks;
7. adslAtucPerfXCurr1DayEcs;
8. adslAturPerfXCurr1DayEcs;
9. adslAtucPerfCurr1DayUasL;
10. adslAturPerfCurr1DayUasL;
11. adslAturPerfCurr1DayLprs;
12. adslAtucPerfCurr1DayInits.

As amostras estão normalizadas em relação à sua média e variância. Devido à quantidade de métricas, foi necessário dividir o diagrama em quatro blocos diferente, na seguinte ordem:

- No primeiro bloco estão plotadas as métricas de 1 a 6 (abscissas) pelas métricas de 1 à 6 (ordenadas).
- No segundo bloco estão plotadas as métricas de 1 a 6 pelas métricas de 7 à 12
- No terceiro bloco estão plotadas as métricas de 7 a 12 pelas métricas 1 a 6
- No quarto bloco estão plotadas as métricas de 7 a 12 pelas métricas de 7 a 12

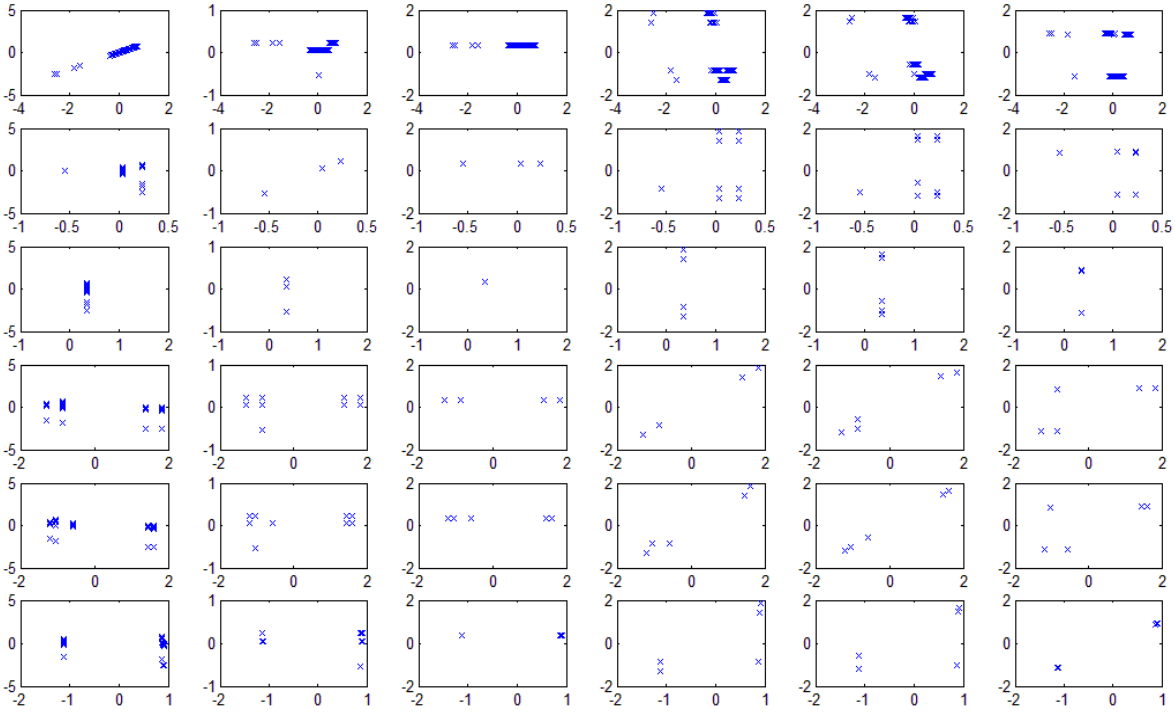

Tabela 12 Métricas de 1 a 6 (abscissas) pelas métricas de 1 à 6 (ordenadas)

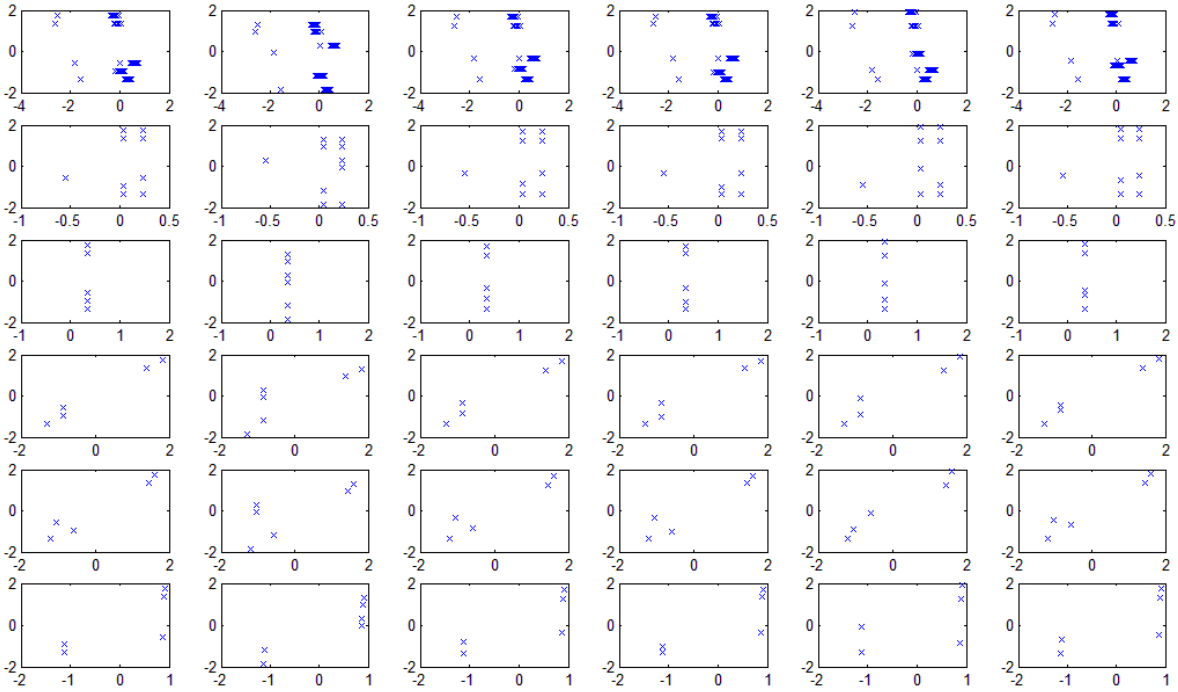

Tabela 13 Métricas de 1 a 6 pelas métricas de 7 à 12

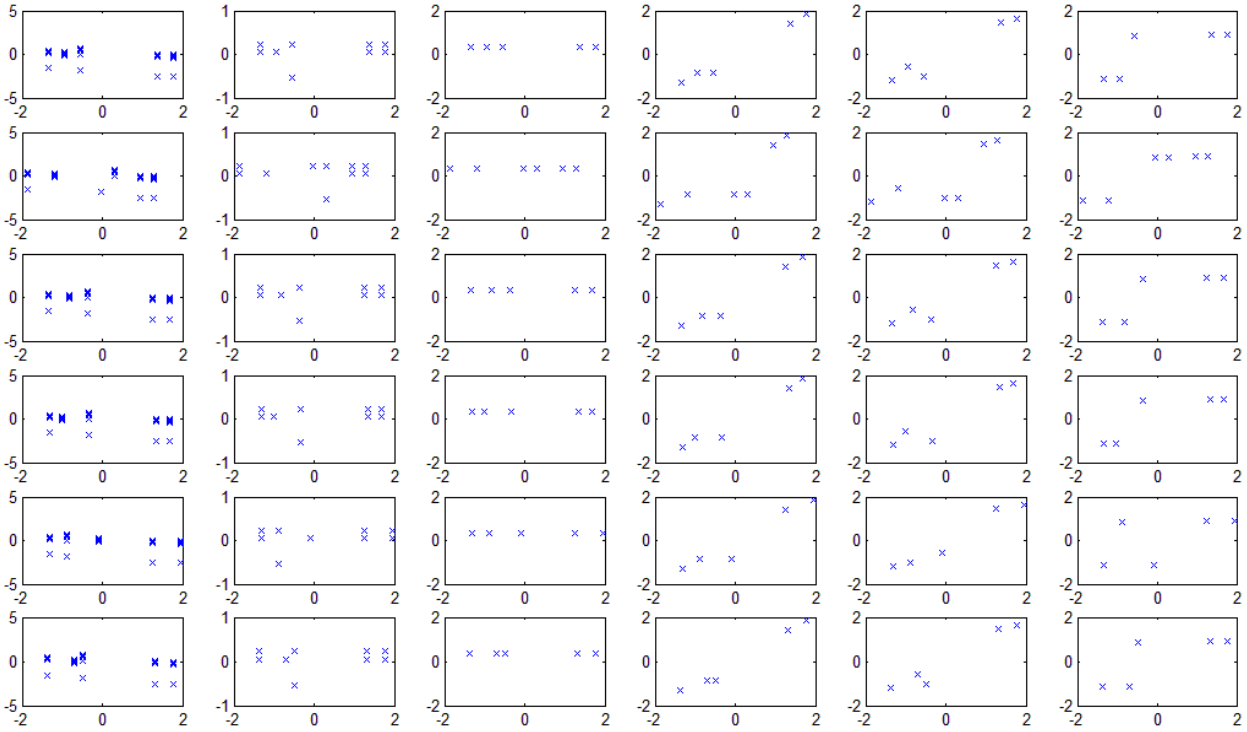

Tabela 14 Métricas de 7 a 12 pelas métricas 1 a 6

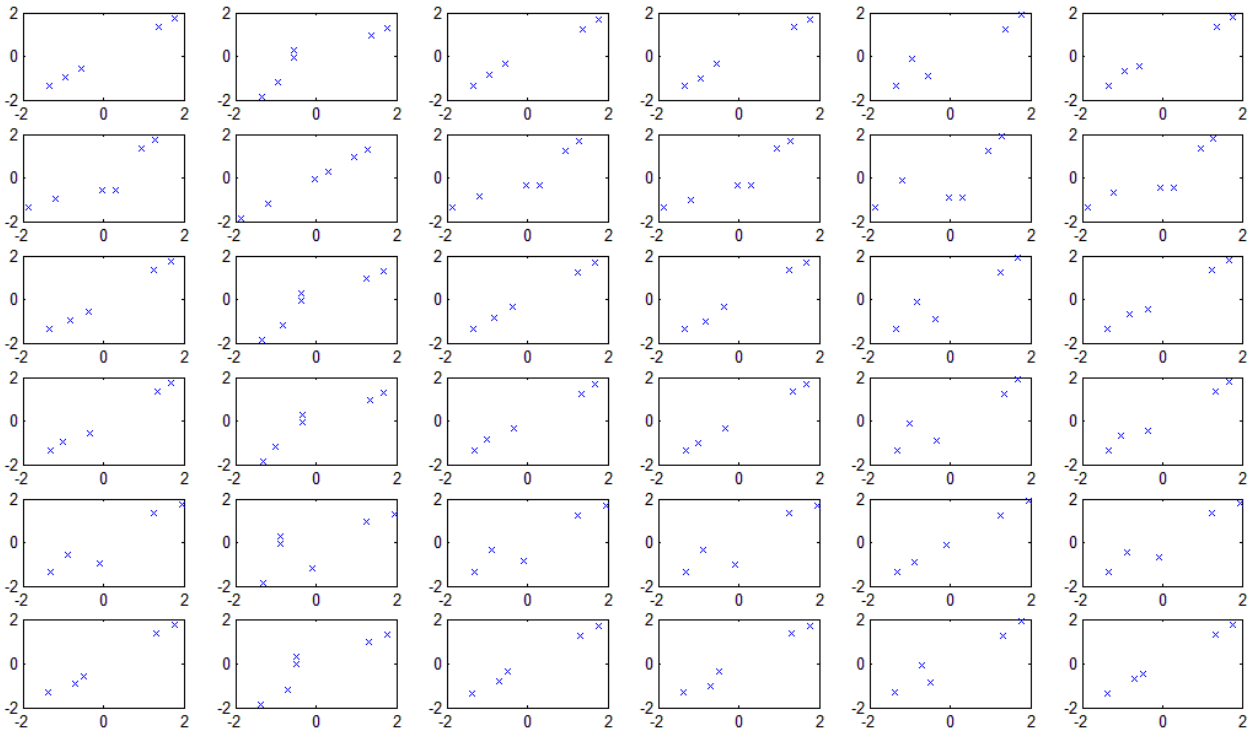

Tabela 15 Métricas de 7 a 12 pelas métricas de 7 a 12

APÊNDICE C – ARQUIVOS DE RUÍDO UTILIZADOS NAS MEDIÇÕES

Os nomes dos arquivos de ruído indicam algumas características físicas deles, bem como a norma da qual foram originados. No ruído impulsivo, nota-se o intervalo entre de cada impulso (100 μs). No ruído

Crosstalk:
Para 50m
Potência: variável
ITU-T/VDSL2_(North_America)_v1.0/G993-2_Annex_A/POTS/at_VTU-O/Loop1/G9932VDSL2-APOTS_VTU-O_Loop1-0100ft_xtk.enc

Potência: variável
ITU-T/VDSL2_(North_America)_v1.0/G993-2_Annex_A/POTS/at_VTU-O/Loop2/G9932VDSL2-APOTS_VTU-O_Loop2-0100ft_xtk.enc

Para 150m
Potência: variável
ITU-T/VDSL2_(North_America)_v1.0/G993-2_Annex_A/POTS/at_VTU-O/Loop1/G9932VDSL2-APOTS_VTU-O_Loop1-0500ft_xtk.enc

Potência: variável
ITU-T/VDSL2_(North_America)_v1.0/G993-2_Annex_A/POTS/at_VTU-O/Loop2/G9932VDSL2-APOTS_VTU-O_Loop2-0500ft_xtk.enc

Para 450m
Potência: variável
ITU-T/VDSL2_(North_America)_v1.0/G993-2_Annex_A/POTS/at_VTU-O/Loop1/G9932VDSL2-APOTS_VTU-O_Loop1-1500ft_xtk.enc

Potência: variável
ITU-T/VDSL2_(North_America)_v1.0/G993-2_Annex_A/POTS/at_VTU-O/Loop2/G9932VDSL2-APOTS_VTU-O_Loop2-1500ft_xtk.enc

Impulsivo
Potência: 0 dbm
REIN/Differential_Mode/Europe-100Hz/-85dBm-hz/-Rein-85dBm-Hz_100us-100Hz_td.enc

RFI
Potência: -44 dbm
TS101_270-1v2-0-10_5B19v2-0/Broadcast_RF/-ETSI-VDSL_RF_Diff-Mode_Up-A_rfi.enc

Potência: -54 dbm
TS101_270-1v2-0-10_5B19v2-0/Broadcast_RF/-ETSI-VDSL_RF_Diff-Mode_Up-B_rfi.enc