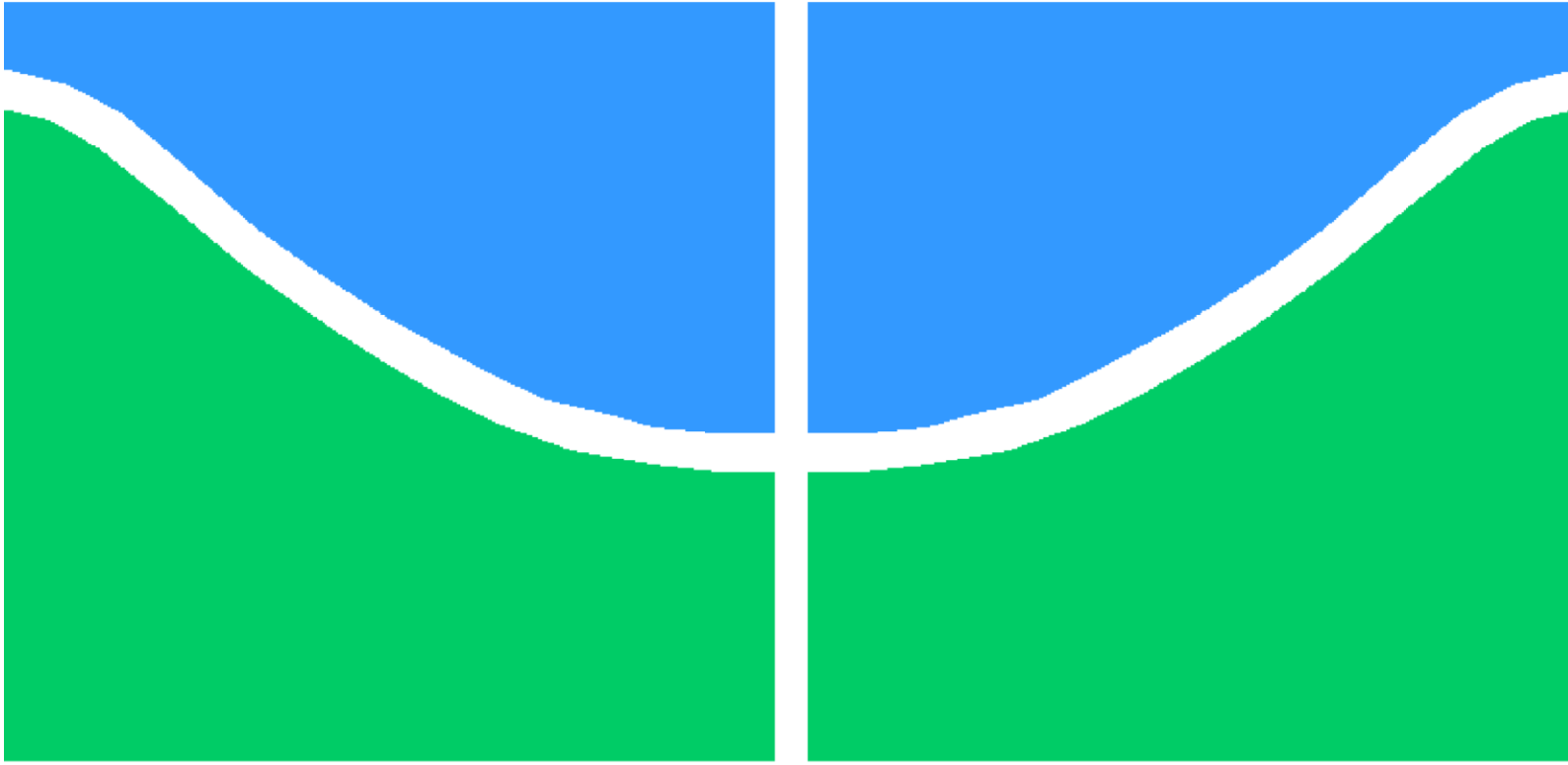


DETECÇÃO DE COMPRESSÃO DUPLA AMR USANDO CARACTERÍSTICAS DE VOZ NO DOMÍNIO DA COMPRESSÃO

JOSÉ FABRIZIO PEREIRA SAMPAIO

TESE DE DOUTORADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

DETECÇÃO DE COMPRESSÃO DUPLA AMR USANDO CARACTERÍSTICAS DE VOZ NO DOMÍNIO DA COMPRESSÃO

JOSÉ FABRIZIO PEREIRA SAMPAIO

TESE DE DOUTORADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA DA UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR.

APROVADA POR:

Prof. Francisco Assis de Oliveira Nascimento, Doutor (UFRJ)
(Orientador)

Prof. José Antonio Apolinário Junior, Doutor (UFRJ)
(Examinador Externo)

Prof. Li Weigang, Doutor (ITA)
(Examinador Interno)

Prof. Ricardo Lopes de Queiroz, D. Sc. (UTA, EUA)
(Examinador Interno)

PUBLICAÇÃO: PPGENE.TD – Nº 168A/2020

BRASÍLIA/DF: DEZEMBRO – 2020

FICHA CATALOGRÁFICA

SAMPAIO, JOSÉ FABRIZIO PEREIRA	
Detecção de Compressão Dupla AMR usando Características de Voz no Domínio da Compressão [Distrito Federal] 2020.	
xviii, 133p, 210 x 297 mm (ENE/FT/UnB, Doutor, Tese de Doutorado – Universidade de Brasília. Faculdade de Tecnologia.	
Departamento de Engenharia Elétrica	
1. Codificador AMR	2. Áudio forense
3. Compressão dupla	4. Domínio da compressão
5. Escalonamento robusto	6. Seleção de características
I. ENE/FT/UnB	II. Título

REFERÊNCIA BIBLIOGRÁFICA

SAMPAIO, J. F. P (2020). Detecção de Compressão Dupla AMR usando Características de Voz no Domínio da Compressão. Tese de Doutorado em Engenharia Elétrica, Publicação PPGENE.TD – Nº 168A/2020, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 132p.

CESSÃO DE DIREITOS

AUTOR: José Fabrizio Pereira Sampaio.
TÍTULO: Detecção de Compressão Dupla AMR usando Características de Voz no Domínio da Compressão.
GRAU: Doutor ANO: 2020

José Fabrizio Pereira Sampaio
SPO Lote 7 - Setores Complementares
70610-902 Brasília – DF – Brasil

AGRADECIMENTOS

Meus sinceros e eternos agradecimentos ao Prof. Dr. Francisco Assis Nascimento pela honra da sua companhia nesta jornada, pela sua paciência e ensinamentos transmitidos, além da sua fé inabalável na minha capacidade de vencer esse desafio.
Ao Departamento de Engenharia Elétrica pela oportunidade oferecida e à Polícia Federal pelo apoio institucional durante o curso.

Dedico esta tese a Deus, por ter permitido chegar tão longe; aos meus pais, pela base escolar sólida e pelo sacrifício pessoal para oferecê-la; à minha esposa Fernanda, pela paciência sem limites, apoio incondicional e amor sincero; à minha pequena filha Laura, privada de tantos momentos comigo, mas sempre com um sorriso iluminado quando com ela podia estar; ao Prof. Francisco Assis, pela amizade leal, conselhos equilibrados e entusiasmo contaminante pela ciência.

RESUMO

DETECÇÃO DE COMPRESSÃO DUPLA AMR USANDO CARACTERÍSTICAS DE VOZ NO DOMÍNIO DA COMPRESSÃO

Autor: José Fabrizio Pereira Sampaio

Orientador: Francisco Assis de Oliveira Nascimento

Programa de Pós-graduação em Engenharia Elétrica.

Brasília, mês de dezembro (2020)

O codec AMR (*adaptive multirate*) é um padrão para compressão de sinal de voz na rede móvel celular e também para armazenar áudio como um formato de arquivo com extensão AMR em gravadores digitais e smartphones. O fácil acesso a programas para adulterar arquivos AMR elevou a demanda por exames de autenticação de áudio nos processos judiciais. Um dos procedimentos de triagem mais úteis é a detecção de compressão dupla, pois, em termos gerais, um arquivo duplamente comprimido é incompatível com um arquivo original. Nesta tese, um novo método baseado em máquina de vetor suporte (SVM) é proposto para detectar arquivos AMR duplamente comprimidos usando apenas características (*features*) no domínio da compressão, em contraste com os métodos existentes que usam a forma de onda descomprimida. Parâmetros específicos do áudio codificado são extraídos por desempacotamento, como os coeficientes de predição linear, e então usados para computar um conjunto de características estatísticas. Para melhorar o desempenho da SVM, um procedimento robusto é usado para escalonar as características. A seleção do modelo SVM consiste em uma busca em grade seguida por um algoritmo recursivo de eliminação de características com redução de polarização de correlação para determinar o melhor número de características que maximiza a acurácia de validação cruzada. A organização dos experimentos foi implementada usando o *corpus* de voz TIMIT, permitindo comparar o método proposto com o estado da arte, revelando que ele supera os métodos publicados. Uma análise de robustez exaustiva também foi feita para quatro condições adversas: um *corpus* diferente (em português brasileiro), arquivos com duração variável, ataque de descolamento de quadro (*frame offset*) e adição de ruído. Tais experimentos demonstram que o método proposto é robusto, assim como apresenta alto desempenho para arquivos de áudio AMR contaminados por ruído.

ABSTRACT

DETECTION OF AMR DOUBLE COMPRESSION USING COMPRESSED-DOMAIN SPEECH FEATURES

Author: José Fabrizio Pereira Sampaio

Supervisor: Francisco Assis de Oliveira Nascimento

Programa de Pós-graduação em Engenharia Elétrica.

Brasília, December (2020)

The adaptive multi-rate (AMR) codec is a speech signal compression standard designed for mobile networks and to store audio as AMR extension file format in digital recorders and smartphones. Easy access to software to tamper with AMR files has increased audio authentication demand in court trials. One of the most useful screening procedures is the double compression detection because, in general terms, a double compressed file is incompatible with an original audio file. In this thesis, a new method based on support vector machine (SVM) is proposed to detect double compressed AMR audio files by using only compressed-domain speech features, in contrast to existing methods which use decompressed waveform. Specific parameters from encoded audio are extracted by unpacking, like linear prediction coefficients, and then used to compute a set of statistical features. To improve SVM performance, a robust scaling procedure is used to scale features. The SVM model selection consists of a grid search followed by a recursive feature elimination with correlation bias reduction algorithm to determine the best number of features that maximizes cross-validation accuracy. The experimental setup was implemented using the TIMIT speech *corpus* to compare the proposed method with state-of-the-art, revealing that it outperforms the published methods. An extensive robust analysis was performed for four different adverse conditions: different corpus (in Brazilian Portuguese), variable duration files, frame offset attack and noise addition. Such experiments show that the proposed method is robust, as well as presents high performance for noise contaminated AMR audio files.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE SÍMBOLOS E ABREVIAÇÕES

a	coeficientes de predição linear agregados
a_i	coeficientes de predição linear individuais
AMR	*Adaptive Multirate Codec*
AWGN	*Additive White Guassian Noise*
β	Ganho de *pitch*
BNF	*Best number of features* (melhor número de características)
BR1	primeira taxa de compressão num arquivo com compressão dupla
BR2	segunda taxa de compressão num arquivo com compressão dupla ou taxa de compressão de um arquivo com compressão simples
C	constante de penalidade ou regularização do *kernel* RBF para SVM
CBR	*Correlation Bias Reduction*
CDF	*Cumulative Density Function*
CELP	*Code Excited Linear Prediction*
γ	constante do *kernel* RBF para SVM
DCT	*Discrete Cosine Transform*
DFT	*Discrete Fourier Transform*
E	energia do quadro de voz
ENF	*Electric Network Frequency*
$\boldsymbol{F_{Rank}}$	vetor de ordenamento das características
FS	*Feature Selection*
g_c	ganho do dicionário fixo
GL	*Generalized Logistic*
GSM	*Global System for Mobile Communications*
LP	*Linear prediction*
LPC	*Linear predictive coding*
LSF	*Line Spectral Frequencies*
LSP	*Line Spectrum Pairs*
LTA	*Long Term Average*
$m_{a_i}(x)$	distribuição de probabilidade do primeiro dígito x do coeficiente a_i
MDCT	*Modified Discrete Cosine Transform*

$m_{q_i}(x)$	distribuição de probabilidade do primeiro dígito *x* do coeficiente q_i
$\boldsymbol{M}$	matriz de características
$\boldsymbol{M_F}$	matriz de características após a eliminação de características
$\boldsymbol{M_{Te}}$	matriz de teste
$\boldsymbol{M'_{Te}}$	matriz de teste escalonada
$\boldsymbol{M'_{Tek}}$	matriz de teste escalonada e ordenada até a característica *k*
$\boldsymbol{M'_{TeBNF}}$	matriz de teste escalonada e ordenada com *BNF* características
$\boldsymbol{M_{Tr}}$	matriz de treinamento
$\boldsymbol{M'_{Tr}}$	matriz de treinamento escalonada
$\boldsymbol{M'_{Trk}}$	matriz de treinamento escalonada e ordenada até a característica *k*
$\boldsymbol{M'_{TrBNF}}$	matriz de treinamento escalonada e ordenada com *BNF* características
N	tamanho do *corpus*
n	quadro do sinal no tempo
NCF	*Number of Current Features* (número atual das características)
N_f	número total de quadros de voz num arquivo AMR
NFL	patamar de ruído
N_p	número total de parâmetros ACELP extraídos
N_{TE}	número de linhas da matriz de teste
N_{TR}	número de linhas da matriz de treinamento
PCM	*Pulse Code Modulation*
$\hat{p}$	parâmetro do codec AMR quantizado
q	pares de espectro de linha agregados
q_j	pares de espectro de linha individuais
r	coeficiente de correlação
RBF	*Radial Base Function*
RFE	*Recursive Feature Elimination*
SID	*Silence Descriptor*
SNR	*Signal to noise ratio* (relação sinal-ruído)
SVM	*Support Vector Machine*
T	período de *pitch*
TIMIT	*Acoustic-Phonetic Continuous Speech Corpus, Texas Instruments*

	& MIT
TNF	número total das características
UBM-GMM	*Universal background model - Gaussian mixture model*
$\boldsymbol{X} = [x_1\ x_2\ \dots x_k]$	vetor linha de parâmetros AMR extraídos, $x = a_i$, q_j,T, β, E, g_c ou *NFL*
$\boldsymbol{\chi} = [\ \chi_1\ \chi_2\ \cdots \chi_{NCF}]$	vetor linha de características
$\boldsymbol{\chi}_{\mathrm{k}} = [\chi_{1k}\ \chi_{2k}\ \chi_{3k} \cdots \chi_{NCFk}]$	vetor linha de características do evento *k*
$\boldsymbol{\chi}' = [\ \chi'_1\ \chi'_2 \cdots \chi'_{NCF}]$	vetor linha de características escalonadas
$\boldsymbol{\chi}'_{rk} = [\ \chi'_{r1}\ \chi'_{r2} \cdots \chi'_{rk}]$	vetor linha de características escalonadas e ordenadas da 1ª característica até a *k*-ésima
$\boldsymbol{y} = [\ y_1 y_2 \dots y_N]^T$	vetor coluna de rótulos

LISTA DE TERMOS TÉCNICOS TRADUZIDOS DO INGLÊS

Acurácia	*accuracy*
Alimentação progressiva	*feedforward*
Análise de áudio digital forense	*digital audio forensics*
Aprendizado de máquina	*machine learning*
Aprendizagem profunda	*deep learning*
Autocodificador empilhado	*stacked autoencoder*
Busca em grade	*grid search*
Características	*features*
Classificação das características	*feature rank*
Contêiner	*container*
Descida de gradiente	*gradient descent*
Desempacotamento	*unpacking*
Deslocamento de quadro	*frame offset*
Dicionário	*codebook*
Domínio da compressão	*compressed domain*
Fluxo de bits	*bitstream*
Frequência da rede elétrica	*electric network frequency*
Função de base radial	*radial base function*
Função de núcleo	*kernel function*
Grade de quadros	*frame grid*
Máquina de vetor suporte	*support vector machine*
Multimídia forense	*multimedia forensics*
Obliquidade	*skewness*
Retropropagação	*backpropagation*
Seleção de características	*feature Selection*
Sobreajuste	*overfitting*
Taxa de bits	*bitrate*
Validação cruzada	*cross-validation*
Valor atípico	*outlier*

1- INTRODUÇÃO

O número crescente de investigações criminais no Brasil e a facilidade de gravação pelo cidadão comum têm gerado grande volume de áudio digital capturado do ambiente ou de chamadas telefônicas que trafegam pela rede de telefonia móvel. À medida que aumentam o volume e a diversidade de investigados, muitas vezes envolvidos em vazamentos de dados ou crimes de colarinho branco, as contestações e questionamentos de autenticidade também aumentam, exigindo um exame altamente especializado para buscar indícios de adulteração. A literatura demonstra que a pesquisa é mais intensa na detecção de falsificações em imagens digitais, havendo lacunas significativas em relação ao áudio digital, dificultando o trabalho dos peritos forenses pela falta de técnicas efetivas e padronizadas para exames em larga escala. Além de ser matéria discutida nos tribunais, a demonstração de que um arquivo de áudio digital é autêntico surge como um problema complexo de engenharia em que, até os dias atuais, poucas técnicas são aplicáveis de forma padrão a um grande número de casos.

O termo autenticação é frequentemente usado, num contexto legal, para descrever o estabelecimento de um fundamento jurídico apropriado para a admissão de uma gravação como prova num processo judicial. Isso geralmente é realizado por uma parte envolvida nos eventos gravados ou envolvida no processo de gravação. A parte afirma que os eventos ouvidos durante a reprodução da gravação são consistentes com a sua lembrança sobre a forma como esses eventos aconteceram. Quando isso é contestado ou não puder ser realizado, uma análise científica pode ser conduzida para testar as alegações contestadas (SWGDE, 2017).

Autenticidade do áudio, em termos da prática forense, significa determinar se o arquivo digital examinado representa fielmente o fato acústico tal como foi armazenado originalmente na primeira mídia. Para ser admitida como prova, a gravação de áudio deve ser, dentro de alguma medida de aceitação, a primeira manifestação do som num formato armazenado e recuperável (SWGDE, 2017), ou seja, deve ser autêntica. Por outro lado, o arquivo digital pode ser formado por um único trecho gravado em operação ininterrupta ou por trechos contínuos concatenados por operações no gravador (como pausas ou paradas), ou ainda concatenados por interrupções próprias do sistema de transmissão de áudio (como na rede de telefonia móvel). Observa-se que a detecção de edições, ou seja, a verificação da autenticidade do áudio digital, incluindo a determinação temporal/espectral precisa das

adulterações realizadas, é uma tarefa não trivial e que se configura atualmente como um problema aberto.

A autenticação de áudio consiste em um conjunto de técnicas da ciência forense que visa a determinar se uma gravação de áudio é original e a revelar todos os artefatos de manipulação fraudulenta. A prova de áudio deve ser autêntica para ser admitida pelo juízo; portanto, o perito forense precisa de um procedimento específico para cada tipo de gravação de áudio. A tarefa de encontrar fraudes no áudio era difícil na era analógica e se tornou um problema de engenharia complexo e de solução morosa em tempos de áudio digital. A alta disponibilidade de programas de edição de áudio, muitas vezes gratuitos, e as incontáveis formas de adulterar o conteúdo de áudio demandam uma extensa coleção de algoritmos para contra-atacar os fraudadores.

Nos últimos vinte anos, alguns pesquisadores elaboraram propostas para tentar responder ao desafio da autenticação de áudio digital. As técnicas desenvolvidas podem ser classificadas em ativas, como a marca d'água, e passivas, como qualquer uma que use apenas propriedades intrínsecas do áudio para revelar os traços de adulteração. As técnicas passivas são geralmente mais úteis porque as gravações do mundo forense não têm normalmente qualquer dado de autenticação ativa embutido. Tais técnicas usam processamento digital de sinais para revelar adulterações imperceptíveis e podem ser consideradas uma parte da especialidade da multimídia forense.

1.1 - MULTIMIDIA FORENSE

A análise da prova material é um procedimento comum no processo penal e no processo civil na qual os peritos forenses devem responder à acusação ou à defesa baseados na ciência forense. Em alguns sistemas processuais penais, como no Brasil, a análise das provas também é feita na fase de investigação para embasar as diligências policiais e medidas cautelares judiciais. Dentre todos os campos existentes da ciência forense, a multimídia forense é uma área especializada e dedicada a examinar provas multimídia como fotografia digital, áudio e vídeo digital para admiti-las num processo legal (Battiato *et al*., 2016). A primeira vez em que a multimídia forense foi reconhecida como um campo da ciência forense foi em 2000 e seu crescimento exponencial pode ser explicado pela enorme quantidade de informação digital armazenada e gerada, por exemplo, por smartphones e câmeras de segurança. Especificamente, a análise de áudio digital forense tem por objetivo examinar a prova de áudio de várias formas, como verificar a autenticidade, melhorar a inteligibilidade da voz, esclarecer fatos baseados em

eventos acústicos (como análise de ruídos de disparos de arma de fogo) e para identificar locutores.

A análise de áudio forense tem uma longa história a ser contada que vai desde as fitas analógicas (como no icônico caso *Watergate*) até o enfrentamento da autenticidade de áudio digital. Devido à predominância esmagadora das técnicas digitais para armazenar áudio, o áudio digital se tornou praticamente o único tipo de prova acústica examinada nos últimos quinze anos. Tal fato, aliado ao nascimento de vários padrões de compressão de áudio, estimulou a pesquisa especializada para criar técnicas adequadas à autenticação de áudio. A verificação de integridade e autenticidade de áudio digital é tida atualmente como uma área complexa da ciência forense.

Uma revisão do estado da arte em multimídia forense pode ser apreciada no artigo de Zakariah *et al.* (2017) que aborda especificamente a análise passiva de áudio digital forense e a divide em dois tipos: análise baseada no contêiner e análise baseada no conteúdo. A primeira utiliza a estrutura do arquivo e metadados, enquanto a segunda se vale do conteúdo binário do arquivo em si, também chamado de do fluxo de bits.

A maioria das técnicas disponíveis para verificar a integridade e autenticidade dos registros de áudio digital é baseada no seu conteúdo. Esse conjunto de técnicas pode ser dividido em algumas categorias, dentre as quais podem ser destacadas o critério ENF (*Electric Network Frequency*) e a identificação do histórico de compressão. O critério da frequência da rede elétrica ENF é um dos métodos mais confiáveis para análise forense e é baseado na oscilação aleatória da frequência da rede elétrica. Pode ser empregado com um banco de dados da ENF, para comparar com o arquivo examinado, ou sem esse banco de dados. No primeiro caso, é possível determinar o instante da edição, a região geográfica e data da gravação, enquanto no segundo caso a continuidade da fase do sinal ENF é o principal elemento de análise.

Já a identificação do histórico de compressão desempenha papel importante na análise forense de áudio digital, pois revela a compressão prévia em formatos a princípio descomprimidos e se a qualidade do áudio foi manipulada pela transcodificação (quando o codec da segunda compressão é diferente da primeira) de baixa para alta taxa de bits. Uma variação dessa técnica é chamada de análise do nível de compressão (CLA, *compression level analysis*) e se baseia na baixa correlação de amostras não comprimidas, em contraste com uma maior correlação de amostras decodificadas de arquivos comprimidos com codificadores perceptuais (SWGDE, 2017). A detecção de compressão dupla é a técnica que revela se o áudio digital foi comprimido duas vezes, indicando que pode ter havido

algum tipo de manipulação, enquanto a detecção de edições pelo deslocamento de quadros é uma das ferramentas mais confiáveis para análise de arquivos de áudio gerados por codificadores baseados em MDCT (*Modified Discrete Cosine Transform*), como o MP3.

Dentre as categorias de técnicas de áudio digital forense, a identificação do histórico de compressão é o tema desta tese. Especificamente, o estudo da compressão dupla em arquivos de áudio digital comprimidos com o codec AMR é o foco do trabalho. Em linhas gerais, o interesse pelo codec AMR pode ser explicado pelo seu amplo emprego como padrão de codificador de voz do sistema móvel GSM (*Global System for Mobile Communications*) e estar presente em muitos gravadores digitais e smartphones para registro de áudio. Detectar compressão dupla em um arquivo AMR significa que o arquivo em questão é incompatível com um arquivo original comprimido uma única vez e que, provavelmente, uma modificação foi intencionalmente realizada no conteúdo antes de se comprimir pela segunda vez para gerar um novo arquivo AMR.

1.2 - OBJETIVOS

O objetivo geral desta tese consiste na elaboração de um novo método para detectar arquivos AMR submetidos à compressão dupla e que tenha desempenho superior aos métodos conhecidos.

1.2.1 - Objetivos específicos

Para alcançar o objetivo geral desta tese, os seguintes objetivos específicos foram propostos:

- Realizar experimentos exploratórios sobre o comportamento do codificador AMR quando é realizada a compressão dupla;
- Definir a forma de extração de características (*features*) adequadas ao problema;
- Definir e realizar experimentos com a rede neural a ser utilizada para a detecção, incluindo o melhor algoritmo de escalonamento das características;
- Definir e realizar experimentos com o método de seleção de características considerado mais adequado para o problema;
- Realizar a análise de robustez do método proposto.

1.3 - ORGANIZAÇÃO DA TESE

A presente tese é dividida em nove capítulos. O primeiro contém a introdução do trabalho, a contextualização e a justificativa do problema de pesquisa. Nele também são apresentados os objetivos do estudo.

O segundo capítulo é iniciado com a contextualização e revisão da literatura sobre técnicas de autenticação de áudio, dentre elas o critério ENF e técnicas de análise baseada na compressão. Os principais trabalhos na área são explicados de forma resumida, abordando a própria definição de compressão dupla e o estado da arte na sua detecção para o codec AMR. A detecção de transcodificação também é citada, assim como a determinação do histórico de compressão de arquivos tipo WAV.

O terceiro capítulo é iniciado com a descrição dos principais aspectos do codec AMR utilizado nas redes móveis de telefonia, abordando os detalhes técnicos necessários para o desenvolvimento do método de detecção proposto, como a codificação baseada em predição linear. O formato padrão de arquivos de áudio AMR é explicado nesse capítulo, evidenciando as diferenças fundamentais entre esse formato e o uso do codec nas redes móveis. O capítulo é encerrado com a delimitação do escopo do trabalho para a detecção de compressão dupla nos arquivos do formato AMR.

O quarto capítulo é dedicado à descrição completa do método proposto e exposição da sua fundamentação teórica. São explorados os fundamentos para a detecção da compressão dupla AMR baseados no domínio da compressão. A detecção pode ser feita pelos parâmetros extraídos do fluxo de bits AMR ou do algoritmo de codificação. Após uma visão geral do diagrama de blocos do método, o capítulo continua com um detalhamento do algoritmo elaborado. A implementação da extração dos parâmetros no domínio da compressão dos arquivos AMR é descrita, assim como o cálculo das características estatísticas é matematicamente definido. O escalonamento robusto, o classificador e a seleção de características são algoritmos integrantes do método proposto e são detalhadamente descritos ao longo desse capítulo.

O quinto e sexto capítulos tratam dos experimentos computacionais realizados para aferir o desempenho do método. Os detalhes de implementação e uma descrição mais aprofundada dos algoritmos são fornecidos no quinto capítulo. O *corpus* TIMIT, empregado para as simulações comparativas, é detalhado no seu tamanho, tipo de arquivos digitais e conteúdo. O fluxo de dados entre os módulos do algoritmo é estabelecido e são dados detalhes da compressão AMR, simples e dupla, cálculo das características e exclusão de características. Para a finalidade desta tese, um experimento é definido no capítulo quinto e sua forma de criação é mostrada a partir da extração de conjuntos de treinamento e teste. Os resultados do escalonamento robusto e da seleção de características, além da implementação do classificador e definição de métricas de desempenho, encerram o quinto capítulo. Os resultados consolidados dos experimentos computacionais utilizando o *corpus*

TIMIT, acompanhados por visualizações das diferenças entre características do áudio AMR com compressão simples e dupla, compreendem o sexto capítulo.

O sétimo capítulo trata da análise de robustez, que é um conjunto de experimentos realizados em condições adversas diferentes daquelas com o uso do *corpus* TIMIT original. Foram selecionadas algumas condições já exploradas na literatura, como arquivos com duração variável, ataque de deslocamento de quadros, áudio com ruído adicionado e troca do *corpus* nos experimentos. Os resultados da análise de robustez estão diretamente relacionados à capacidade do método de discriminar arquivos AMR com compressão dupla em situações diferentes daquelas dos experimentos iniciais.

No oitavo capítulo os resultados alcançados com o método proposto são comparados aos resultados dos métodos reportados na literatura. Uma discussão dos resultados também é desenvolvida nesse capítulo, abordando os efeitos do algoritmo sobre a detecção dos arquivos AMR e os motivos mais prováveis para o desempenho alcançado. As principais vantagens do método proposto também são analisadas nesse capítulo. Já o nono capítulo trata da conclusão desta tese, limitações do método e futuras linhas de pesquisa que poderiam expandir a utilidade do método para outras condições.

2 - REVISÃO DA LITERATURA

A revisão da literatura sobre análise de áudio forense demonstra que muitos métodos diferentes foram propostos com abordagens diversificadas, porém sem convergência para um método único que solucionasse o problema da autenticação forense. Neste capítulo são abordadas duas categorias de análise passiva de áudio forense baseadas no conteúdo: o critério ENF e a identificação do histórico de compressão. Inicialmente o critério ENF é explicado de forma mais superficial e a identificação do histórico de compressão é abordada de forma mais aprofundada.

O critério ENF, que utiliza o ruído da frequência da rede elétrica e seus harmônicos, é reconhecido pela comunidade forense como o mais confiável para detectar e localizar manipulações em áudio digital (SWGDE, 2017). O aperfeiçoamento dessa técnica tem rendido resultados notórios, mesmo para os arquivos digitais gravados em equipamentos alimentados a bateria em que o sinal ENF introduzido é muito fraco. Tal método usa as variações lentas de frequência e fase do sinal da rede elétrica como uma referência confiável para checar a continuidade, a data da gravação e para apontar emendas (Grigoras, 2003). Dentre os trabalhos publicados sobre a técnica ENF, alguns podem ser escolhidos para elaborar um panorama do estado da arte do método.

Quando existe uma base de dados confiável do sinal ENF e o áudio analisado apresenta esse sinal com intensidade suficiente, é possível determinar a data, a localidade geográfica da gravação e o local exato da edição no áudio, mesmo se o gravador for alimentado por baterias. Contudo, a detecção do sinal ENF é uma tarefa difícil se esse sinal for muito fraco no áudio gravado (Kajstura *et al.,* 2005). O uso do critério ENF é viável, mesmo sem uma base de dados, se a fase do sinal é estimada pelo uso de DFT (*Discrete Fourier Transform*) de alta precisão para detectar suas descontinuidades. Tais descontinuidades indicam as manipulações sofridas pelo áudio digital, uma vez que as manipulações mais comuns causam mudanças abruptas nos valores da fase (Rodriguez *et al.*, 2010), permitindo visualizar acréscimos ou supressões no áudio.

O rastreamento das variações do valor da ENF é uma abordagem interessante para detectar descontinuidades porque o áudio inautêntico apresenta variações anômalas de valores de ENF e que podem ser detectadas com um nível de liminar variável (Esquef *et al.*, 2014) ou com um detector de valores atípicos baseado na curtose das amostras, usando máquina de vetor suporte (SVM – *Support Vector Machine*) (Reis *et al*., 2017*).* Outra aplicação do critério ENF é a identificação de áudio recapturado, pois o tom ENF é

gravado duas vezes sempre que o áudio original é recapturado de alguma maneira, como numa conversão analógica-digital ou numa gravação durante reprodução em diferentes gravadores, o que permite identificar a manipulação (Su *et al.*, 2013).

2.1 - IDENTIFICAÇÃO DO HISTÓRICO DE COMPRESSÃO

A identificação do histórico de compressão nos arquivos de áudio tem papel importante no conjunto de ferramentas forenses para a autenticação de áudio. Ela pode apontar, por exemplo, se foi utilizado mais de um tipo de codificador por meio da investigação de medidas estatísticas do sinal ou de peculiaridades de compressão. O histórico de compressão também é um tópico importante na análise de áudio digital forense porque as operações de compressão verificadas no áudio e no suposto gravador devem ser compatíveis. A detecção de compressão dupla, que é um tópico da identificação do histórico de compressão, tem a vantagem de apontar com segurança se o arquivo de áudio foi manipulado, pois as modificações fraudulentas de conteúdo devem ser realizadas nas amostras decodificadas que, posteriormente, devem ser codificadas novamente.

Sempre que se deseja adulterar um arquivo de áudio digital comprimido, como fazer supressões ou emendas, é necessário decodificar o arquivo original para o domínio do tempo, fazer a adulteração e comprimir novamente para um formato que normalmente é o mesmo do arquivo original para manter a compatibilidade com o gravador. Essa operação, ilustrada na Figura 2-1, é chamada de compressão dupla.

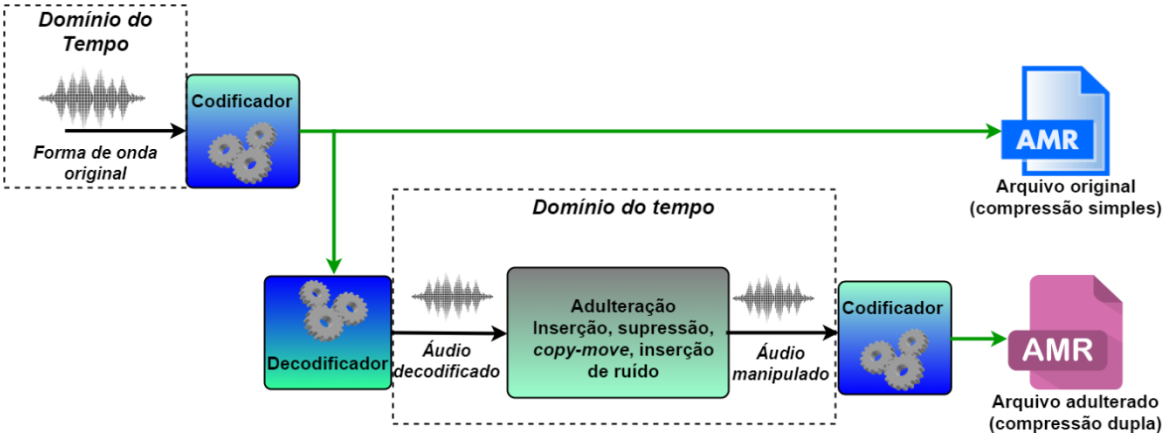


Figura 2-1 - Procedimento de adulteração do áudio digital com posterior compressão dupla.

A estrutura de quadros AMR é formada por blocos de bits codificados, sem acesso direto às amostras no domínio do tempo, pois o fluxo de bits apenas disponibiliza parâmetros AMR quantizados. Em outras palavras, é inviável manipular a voz diretamente no áudio codificado e, portanto, uma decodificação é necessária. Se o formato do arquivo

manipulado for o mesmo do arquivo original, ocorre a compressão dupla como um procedimento inevitável para o fraudador.

Os primeiros trabalhos sobre detecção de compressão dupla foram voltados para o codificador MP3, que é baseado na MDCT. A literatura traz algumas técnicas baseadas em propriedades da MDCT para detectar compressão dupla, como o método da distribuição global e distribuição por bandas dos primeiros dígitos MDCT (Yang *et al.*, 2010) e a diferença da estatística dos coeficientes MDCT (Liu *et al.*, 2010). O método baseado na distribuição global dos primeiros dígitos MDCT utiliza um classificador para discernir entre a estatística dos primeiros dígitos de um arquivo MP3 com compressão simples e com compressão dupla. Já o método da distribuição por bandas dos primeiros dígitos MDCT melhora a detecção da compressão dupla pela percepção de que os coeficientes MDCT de um quadro são quantizados de maneira diferente, dependendo da sua faixa de frequências, gerando estatísticas diferentes para cada banda (Yang *et al.*, 2010).

Outra forma de detecção de compressão dupla de arquivos MP3, comprimidos de uma taxa de bits menor para uma maior, é baseada na diferença da estatística dos coeficientes MDCT em si. O princípio é que, para cada sub-banda do banco de filtros do algoritmo MP3, os coeficientes MDCT são modificados de forma diferente para compressão simples e dupla, de modo que as razões entre os coeficientes de valor absoluto maior que certo limiar e o número total de coeficientes mudam. A partir dos arquivos MP3 com compressão dupla, as razões são calculadas e um algoritmo extrai as características estatísticas de interesse (Liu *et al.*, 2010).

Outra abordagem para a identificação do histórico de compressão pode ser útil quando se dispõe apenas do arquivo de áudio digital descomprimido (por exemplo, formato WAV). Esse problema é verificado com certa frequência no meio forense quando uma manipulação é feita no domínio PCM (*pulse code modulation*) e o áudio não é comprimido novamente, evitando assim o emprego de algoritmos de detecção de compressão dupla. O método, proposto por Luo *et al.* (2012), é capaz de identificar se houve compressão pelos codificadores MP3 e WMA e qual foi a taxa de bits usada. Ele é baseado no fato de que, ao se aplicar o algoritmo de compressão em um arquivo WAV já comprimido, o número de coeficientes MDCT quantizados para zero será maior do que para um arquivo WAV nunca antes comprimido.

2.2 - ESTADO DA ARTE PARA DETECÇÃO DE COMPRESSÃO DUPLA DO CODEC AMR

Foram encontrados na literatura três trabalhos que abordam o problema da detecção de compressão dupla AMR. O primeiro utiliza a extração de características acústicas do áudio decodificado e as insere num classificador do tipo SVM, alcançando uma acurácia média de 87% (Shen *et al.*, 2012). Dois anos depois, foi reportado um método que insere o áudio decodificado diretamente numa rede neural de aprendizagem profunda de três camadas para diferenciar os arquivos AMR com compressão dupla, oferecendo acurácias da ordem de 91% (Luo *et al.*, 2014). Num terceiro estudo, um método baseado na rede neural do tipo autocodificador empilhado alcançou acurácias comparativas da ordem de 98%, usando, mais uma vez, o áudio decodificado diretamente como entradas da rede neural (Luo *et al.*, 2017). A seguir as três técnicas são descritas em mais detalhes.

Na primeira técnica (Shen *et al.*, 2012), características estatísticas relacionadas à distribuição de energia e baseadas na DFT são extraídas dos arquivos AMR descomprimidos no domínio PCM. Após a extração dessas características do áudio, os vetores das características do *corpus* TIMIT (Garofolo *et al.*, 1993) são usados para o treinamento e teste do classificador SVM. Já a segunda técnica (Luo *et al.*, 2014) é baseada numa rede neural de aprendizagem profunda e também tem por objetivo discriminar arquivos de áudio AMR com compressão simples daqueles com compressão dupla. A metodologia para o uso das redes neurais consiste inicialmente em descomprimir os arquivos AMR, normalizar as amostras PCM e utilizá-las diretamente como entradas para as redes no treinamento e no teste. Os resultados dos experimentos foram alcançados com uso de estratégia de votação por maioria e com o *corpus* TIMIT.

Na terceira técnica (Luo *et al.*, 2017), os autores não se baseiam na forma tradicional de resolver problemas de reconhecimento de padrões que usa um estágio de extração manual de características e outro de classificação. A técnica adotada consiste na extração automática de características a partir dos dados por meio da arquitetura de aprendizagem profunda SAE (*Stacked Autoencoder*) e um classificador do tipo UBM-GMM (*universal background model - Gaussian mixture model*). Os experimentos realizados foram direcionados ao uso do *corpus* TIMIT, embora tenham sido usados outros bancos de dados para as avaliações. Uma análise de robustez foi proposta nesse trabalho para ataque de descolamento de quadros, áudio contaminado por ruído, *corpus* diferente, diminuição do conjunto de treinamento e áudio com duração variável. As acurácias dessa técnica são as maiores dentre aquelas até então conhecidas, alcançando 98%.

3 - O CODEC AMR

Os métodos encontrados na literatura para a detecção da compressão dupla AMR se baseiam no áudio decodificado e apresentam acurácias elevadas. Com o intuito de propor um algoritmo de melhor desempenho para o problema da compressão dupla, é necessário estudar em maior profundidade o codificador e o decodificador AMR, buscando formular uma estratégia diferente das até então existentes, o que é apresentado nas próximas seções.

3.3 - PRINCIPAIS ASPECTOS

O codec AMR, cuja primeira padronização 3GPP data de 1999 (3GPP - TS 26.090 v1.0.0), é um codificador/decodificador projetado para o sinal de voz a ser transmitido nos sistemas de telefonia móvel de comutação de circuitos, codificando o sinal de voz na banda entre 80 Hz até 3400 Hz (chamado também de AMR-NB, *narrow band).* O AMR foi originalmente desenvolvido e padronizado pela ETSI (*European Telecommunications Standards Institute*) para o sistema GSM, mas atualmente é um codec obrigatório para o sistema celular de 3ª geração. Trata-se de um codificador multitaxa com taxa controlada pela fonte (também chamada de transmissão descontínua, DTX - *Discontinuous Transmission*) e que inclui um detector de atividade de voz (VAD, *Voice Activity Detector*), um sistema de geração de ruído de conforto (CNG - *Confort Noise Generator*) e um mecanismo de cancelamento de erro para mitigar os efeitos da transmissão e perdas de quadros (3GPP-TS 26.071, 2015). O AMR é um codificador de voz integrado com oito taxas (chamados modos) de transmissão em banda estreita (de 4,75 kbit/s até 12,2 kbit/s) e um modo de codificação de ruído de fundo de baixa taxa, podendo chavear sua taxa a cada quadro de 20ms. Na Tabela 3-1 estão listados os modos do codificador AMR, em que o modo SID é o descritor de silêncio (*silence descriptor*) cuja taxa de bits ocorre quando os quadros de silêncio são transmitidos sequencialmente.

Tabela 3-1 - Taxas de bits de codificação de fonte AMR. (3GPP-TS 26.071, 2015)

Modo	Taxa de bits
AMR_12.20	12,20 kbit/s
AMR_10.20	10,20 kbit/s
AMR_7.95	7,95 kbit/s
AMR_7.40	7,40 kbit/s
AMR_6.70	6,70 kbit/s
AMR_5.90	5,90 kbit/s
AMR_5.15	5,15 kbit/s
AMR_4.75	4,75 kbit/s
AMR_SID	1,80 kbit/s

A entrada do codificador deve ser um sinal PCM linear de 13 bits e amostrado a

8kHz, processado em quadros de 160 amostras (20 ms), enquanto a saída é formada por blocos codificados por funções de transcodificação cujo número de bits depende do modo escolhido. Cada bloco codificado forma o fluxo de bits que contém as representações quantizadas dos parâmetros, cuja alocação de bits depende da taxa de bits AMR. Por exemplo, os coeficientes de predição linear não estão diretamente presentes no áudio codificado porque uma quantização vetorial e uma transformação não linear são aplicadas para tornar tais coeficientes menos sensíveis aos erros de transmissão. Dessa forma, um quadro de 2080 bits (160 amostras) será codificado em blocos de 95, 103, 118, 134, 148, 159, 204 ou 244 bits para os modos AMR_4.75, AMR_5.15, AMR_5.90, AMR_6.70, AMR_7.40, AMR_7.95, AMR_10.2 e AMR_12.2, respectivamente.

O algoritmo de codificação do sinal de voz é o MR-ACELP (*Multi-Rate Algebraic Code Excited Linear Prediction),* baseado no modelo CELP (*Code Excited Linear Prediction*) que utiliza um filtro de síntese de predição linear (LP), ou de curto termo, de 10ª ordem dado por:

$$H(z) = \frac{1}{\hat{A}(z)} = \frac{1}{1 + \sum_{i=1}^{m} \hat{a}_i z^{-i}}, \tag{3.1}$$

onde $\hat{a}_i$ são os parâmetros quantizados do filtro de LP e m=10 é a sua ordem. O filtro de síntese de longo termo, ou filtro de *pitch,* é dado por:

$$\frac{1}{B(z)} = \frac{1}{1 - \beta z^{-T}} \tag{3.2}$$

onde T (*pitch lag*) é o número de amostras num período de *pitch* e β é o ganho de *pitch*. O filtro de *pitch* é implementado usando o método de dicionário (*codebook*) adaptativo.

No modelo de síntese CELP o sinal de voz é codificado pelo algoritmo de análise por síntese. Inicialmente, o sinal de excitação é construído adicionando dois vetores dos dicionários fixo e adaptativo. A voz é sintetizada alimentando esses vetores na entrada do filtro de síntese de curto termo (LP). A sequência de excitação ótima é escolhida dos dicionários usando o procedimento de busca de análise por síntese em que o erro entre o sinal original e o sintetizado é minimizado de acordo com uma medida de distorção perceptiva implementada por um filtro. A cada 160 amostras o sinal de voz é analisado para extrair os parâmetros do modelo CELP (coeficientes do filtro LP, índices dos dicionários fixo e adaptativo, e ganhos).

A análise de predição linear é feita duas vezes por quadro no modo AMR_12.20 e

uma vez nos demais modos, seguida pela conversão dos parâmetros LP para pares de espectro de linha (LSP, *line spectrum pairs*), que têm propriedades úteis para a transmissão num canal de telecomunicações, como menor sensibilidade ao ruído de quantização. Resumidamente, os valores transmitidos dos LSP são as raízes dos polinômios *P(z)* e *Q(z)* derivados de $A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}$ da seguinte forma:

$$\begin{aligned} P(z) &= A(z) + z^{-m-1} A(z^{-1}) \\ Q(z) &= A(z) - z^{-m-1} A(z^{-1}) \end{aligned} \tag{3.3}$$

em que *m* é a ordem do preditor (*m*=10 no codec AMR).

As raízes de *P(z)* e *Q(z)* ocorrem em pares conjugados e estão no círculo unitário no plano complexo, alternando-se ao longo desse círculo. Dessa forma, a representação em LSP usa a localização das raízes de *P(z)* e *Q(z)*, isto é, as frequências ω das raízes $z = e^{j\omega}$, também chamadas de frequências de espectro de linhas (LSF, *line spectral frequencies*). Como elas ocorrem em pares, apenas metade precisa ser transmitida (entre 0 e π) na ordem $0<\omega_1<\omega_2<...<\omega_{10}<\pi$. O termo LSP se refere, no padrão AMR, às quantidades no domínio dos cossenos $q_i = \cos(\omega_i)$, em que q_i é o *i*-ésimo par de espectro de linha.

Após a análise LP, o algoritmo CELP realiza a busca de *pitch* pelo uso dos dicionários algébricos em que a excitação para o filtro LP é um conjunto limitado de pulsos, com a vantagem de que esse tipo de dicionário pode ser muito grande sem causar grandes problemas de complexidade do algoritmo. No AMR, a estrutura dos dicionários algébricos é baseada em ISPP (*interleaved single-pulse permutation*).

O quadro de voz a ser codificado é dividido em quatro subquadros de 5ms cada e os índices dos dicionários fixo e adaptativo são transmitidos a cada subquadro. O codificador, ao final, produz um fluxo de dados de saída num formato único, em que a distribuição dos bits em cada quadro varia com o modo do codificador. Os parâmetros transmitidos em todos os modos são os LSP quantizados, *pitch delay* e *algebraic code*, porém com número variável de bits. A escolha dos modos define a taxa de bits, resultando numa transmissão feita adaptativamente conforme as condições do canal (taxa de erro de bits), obedecendo a um algoritmo apropriado e que faz parte do padrão do codificador.

Por fim, o decodificador AMR recebe o fluxo de bits para obter os seguintes parâmetros de cada quadro transmitido: vetores LSP, *pitch lags*, índices dos dicionários e ganho de *pitch* (dependendo do modo de transmissão). Os vetores LSP são convertidos para os coeficientes LP e interpolados para obter os filtros LP em cada subquadro, para o

qual a excitação é construída pela adição dos vetores de índices multiplicados pelos respectivos ganhos, de modo que a voz é reconstruída filtrando essa excitação pelo filtro LP de síntese. O padrão AMR gera sinais decodificados na faixa de frequência entre 80Hz e 3400Hz pelo uso de filtros digitais, tanto no codificador quanto no decodificador.

3.2 - FORMATO DE ARQUIVOS AMR

Devido às propriedades de compressão do codec AMR e seu amplo uso nos sistemas de telefonia móvel, a criação de um formato de arquivo digital para áudio foi pensada para aplicações de internet, como o envio de anexo de e-mail ou armazenamento de notas de voz em aplicativos de smartphone. Esse formato rapidamente foi adotado por grande número de gravadores de áudio digital e mais recentemente por aplicativos de smartphones capazes de gravar diálogos no ambiente ou em conversas telefônicas. Os arquivos de áudio têm extensão AMR e apresentam a estrutura geral formada por um cabeçalho seguido pelos quadros AMR (Sjoberg *et al.*, 2007) com taxa de bits constante.

Para um arquivo AMR com apenas um canal, o cabeçalho do arquivo é a sequência específica de caracteres "#!AMR\n" (0x2321414D520A). Logo em seguida, os quadros codificados são dispostos consecutivamente no tempo, alinhados em bytes, e armazenados na ordem temporal. Cada quadro começa com um byte no seguinte formato: P-FT-FT-FT-FT-Q-P-P. Os bits FT (*frame type*) indicam o modo AMR usado (incluindo AMR_SID) e, , podem assumir os valores da Tabela 3-2. Já o bit Q, indica a qualidade do quadro (0 corresponde ao quadro danificado e 1 quadro com boa qualidade), enquanto os bits P são de preenchimento (*padding*) para completar 8 bits (assumem o valor zero).

Após o byte de cabeçalho, seguem os bits de parâmetros de voz, completados com zero para atingir um alinhamento de oito bits (bits P). Como exemplo, na Figura 3-1 é mostrado um diagrama de um quadro AMR no modo AMR_5.90 (118 bits) no formato de armazenamento. Os quadros de voz não recebidos ou os quadros entre as atualizações de descritor de silêncio (SID) durante os períodos sem voz devem ser armazenados como *No Data* (tipo 15). Os quadros de ruído de conforto diferentes do AMR_SID (tipo 8), ou seja, dos tipos 9, 10 e 11, não devem ser usados no formato de arquivos AMR.

Tabela 3-2 - Interpretação do tipo de quadro. Adaptado de 3GPP-TS 26.101 (2015)

Frame Type	Frame content (AMR mode, comfort noise, or other)
0	AMR 4,75 kbit/s
1	AMR 5,15 kbit/s
2	AMR 5,90 kbit/s
3	AMR 6,70 kbit/s (PDC-EFR)
4	AMR 7,40 kbit/s (TDMA-EFR)
5	AMR 7,95 kbit/s
6	AMR 10,2 kbit/s
7	AMR 12,2 kbit/s (GSM-EFR)
8	AMR SID
9	GSM-EFR SID
10	TDMA-EFR SID
11	PDC-EFR SID
12-14	For future use
15	No Data (No transmission/No reception)

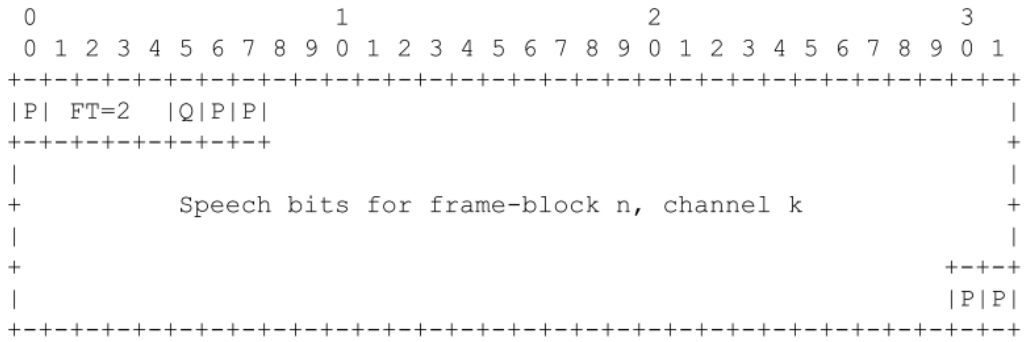


Figura 3-1 - Diagrama de um quadro AMR no modo AMR_5.90 (118 bits) no formato de armazenamento. Extraído de Sjoberg *et al.* (2007).

4 - MÉTODO PROPOSTO PARA DETECTAR COMPRESSÃO DUPLA AMR

A ideia principal empregada no método proposto é usar os arquivos AMR codificados, isto é, no *domínio da compressão*, substituindo a abordagem tradicional que usa o áudio PCM decodificado. A formulação de características passa a ser realizada no domínio da compressão, procedimento que, consoante será visto adiante, traz aumento de desempenho na detecção. A extração de parâmetros do arquivo AMR codificado é uma técnica que precisa ser melhor estudada e é fundamental para o algoritmo do método. Além da mudança de paradigma no modo de abordar o problema, é necessário o uso de alguns métodos específicos, como escalonamento robusto e seleção de características, imprescindíveis para alcançar o desempenho do método detalhado neste capítulo.

4.1 - FUNDAMENTOS PARA A DETECÇÃO DE COMPRESSÃO DUPLA DO CODEC AMR

Conforme se depreende da revisão da literatura no Capítulo 2, a detecção de compressão dupla em arquivos no formato AMR é um tópico interessante da multimídia forense. Assim como nos codificadores baseados em MDCT, o interesse forense pela identificação do histórico de compressão em arquivos AMR vem do questionamento sobre a autenticidade do áudio. Ao invés de considerar as propriedades da MDCT, as técnicas existentes de detecção de compressão dupla AMR focam nos algoritmos de aprendizado de máquina, haja vista que o algoritmo AMR não se vale de nenhum tipo de transformada.

Os trabalhos reportados na literatura até então usam os sinais temporais decodificados (ou seja, no domínio do tempo) de um arquivo AMR como ponto de partida para detectar a compressão dupla. Por outro lado, o método apresentado nesta tese aborda o problema *diretamente* no domínio da compressão. Essa escolha, que é diferente de todas as técnicas até então conhecidas, pode ser justificada pelo princípio da compressão de sinais. O padrão AMR apresenta um conjunto de parâmetros projetados e calculados para reduzir a redundância estatística do sinal de voz digitalizado e para preservar a assinatura espectral que caracteriza a informação e o locutor. Além disso, toda a informação contida no sinal decodificado no domínio do tempo já está presente no formato comprimido de uma forma mais eficiente. Por tudo isso, as características no domínio da compressão podem ser obtidas diretamente do formato codificado sem a necessidade de análise da forma de onda.

A extração de características no domínio da compressão já foi usada com sucesso na área de processamento de imagens. Por exemplo, a análise no domínio da compressão é

uma abordagem interessante nessa área porque os algoritmos de compressão realizam uma espécie de filtragem da informação e decomposição de conteúdo (Chang, 1995), mas ainda conservando informações essenciais para as análises (Yeo e Liu, 1995). Para algumas aplicações em imagens, foi provado que os métodos no domínio da compressão são mais acurados que os similares no domínio dos pixels, oferecendo avanços estatisticamente mais significativos (Delac *et al.*, 2009). Para os problemas de classificação de padrões com imagens comprimidas baseadas na DCT (*Discrete Cosine Transform*), o domínio da compressão é melhor do que o domínio dos pixels porque ele diminui a dependência entre os elementos das características e simplifica a escolha deles, se comparado ao cálculo diretamente com pixels (Luo e Eleftheradis, 2000).

A capacidade de classificação baseada no domínio da compressão para as aplicações de áudio é similar àquela usada para o processamento de imagens. A extração de características de áudio MPEG-1 no domínio da compressão permite a criação de índices para classificar o som como música, voz ou silêncio (Pfeiffer e Vincent, 2003). O método descrito em Chang *et al.*, (2007) calcula características baseadas nos coeficientes MDCT, extraídos do áudio MP3 por decodificação parcial, para remover silêncio e separar diferentes tipos de áudio (voz, ambiente e música). Para o codec AMR, a abordagem do domínio da compressão foi usada para o reconhecimento automático de locutores. Medidas estatísticas simples, como o coeficiente de variação e obliquidade, foram usadas e aplicadas aos parâmetros do fluxo de bits AMR, como os índices e ganhos de dicionário e índices de LSF (Petracca *et al.*, 2005). Tais parâmetros não são linearmente relacionados à formação da voz, mas têm informação relevante para comporem uma característica de voz (Petracca *et al.*, 2006). Além disso, a baixa dependência estatística dos parâmetros do codificador de áudio (Spanias, 1994) os torna adequados para formar características a serem usadas em técnicas de classificação.

Para os arquivos AMR tratados nesta tese, serão apresentados e comparados, nesta seção, dois métodos de extração de parâmetros para gerar características no domínio da compressão e, ao final, um deles será eleito para ser usado no método proposto. No primeiro, já tratado na literatura (Petracca *et al.*, 2005), os parâmetros são extraídos diretamente do arquivo codificado, ou seja, do fluxo de bits e são aqui chamados *parâmetros do fluxo de bits*; no segundo, os parâmetros são extraídos por quantização inversa (ou por "desempacotamento"), ou seja, o arquivo codificado é submetido a uma versão modificada do decodificador, porém o produto útil dessa operação não é o áudio decodificado, e sim os dados adicionais gerados (parâmetros específicos do algoritmo de

codificação, como coeficientes LP e LSP) (Sampaio, 2019). Esses parâmetros são chamados nesta tese de *parâmetros do codificador*. Nas seções a seguir, os dois métodos são discutidos em detalhes e com ilustração de resultados para justificar a escolha de um deles.

4.1.1. Extração a partir dos parâmetros no fluxo de bits

Cada quadro de 160 amostras de voz é codificado no formato AMR usando parâmetros do codificador CELP, como os LSP, períodos de *pitch* e ganhos de dicionário. Esses parâmetros são quantizados vetorialmente para formar o fluxo de bits do arquivo AMR, cuja alocação de bits varia com o modo AMR. Como cada parâmetro é representado por bits do fluxo, uma primeira abordagem seria o uso direto do fluxo de bits do arquivo AMR codificado para detectar algum efeito da compressão dupla. A análise estatística desses parâmetros, visando à descoberta dos efeitos da compressão dupla, é apresentada a seguir.

Tomando, por exemplo, a taxa de 4,75 kbits/s para os arquivos AMR, é necessário inicialmente separar no fluxo de bits os parâmetros de interesse para verificar seu comportamento estatístico ao longo de vários arquivos de um *corpus*. Cada quadro AMR na taxa de 4,75 kbits/s tem 95 bits, porém certos agrupamentos desses bits correspondem a parâmetros definidos pelo padrão do codificador. O significado desses bits pode ser observado na Tabela 4-1, da qual podem ser extraídos, por exemplo, os três índices dos subvetores LSF (8 bits, 8 bits e 7 bits). Da mesma forma, podem ser extraídos o índice do dicionário adaptativo do 1º subquadro (8 bits), o ganho de dicionário para o 1º subquadro (8 bits) e o ganho de dicionário para o 3º subquadro (8 bits).

Tabela 4-1 - Parâmetros de saída em ordem de ocorrência e alocação de bits para um quadro de 20 ms, taxa de 4,75 kbits/s. Extraído de 3GPP - TS 26.090 v13.0.0 (2015)

Bits (MSB-LSB)	Description
s1 – s8	index of 1st LSF subvector
s9 - s16	index of 2nd LSF subvector
s17 – s23	index of 3rd LSF subvector
s24 – s31	adaptive codebook index (subframe 1)
s32	position subset (subframe 1)
s33 – s35	position of 2nd pulse (subframe 1)
s36 – s38	position of 1st pulse (subframe 1)
s39	sign information for 2nd pulse (subframe 1)
s40	sign information for 1st pulse (subframe 1)
s41 – s48	codebook gains (subframe 1)
s49 – s52	adaptive codebook index (relative) (subframe 2)
s53 – s61	same description as s32 – s40 (subframe 2)
s62 - s65	same description as s49 – s52 (subframe 3)
s66 – s82	same description as s32– s48 (subframe 3)
s83 – s95	same description as s49 – s61 (subframe 4)

Para elaborar estatísticas úteis dos parâmetros do fluxo de bits, um *corpus* de áudio (conjunto de locuções gravadas nas mesmas condições) deve ser considerado com um número razoável de eventos. Foi utilizado o *corpus* TIMIT (Garofolo *et al*., 1993), o qual será descrito em detalhes na Seção 5.1, que é um conjunto de 6300 arquivos que foi usado em vários trabalhos de multimídia forense, como em Romero e Wilson (2010), Jenner e Kwasinski (2012), Shen *et. al* (2012), Luo *et. al* (2014), e Luo *et. al* (2017). Com o intuito de avaliar o comportamento dos parâmetros do fluxo de bits AMR na compressão dupla e concluir sobre a viabilidade do uso de medidas estatísticas para detectá-la, os arquivos do *corpus* TIMIT foram codificados em AMR na taxa constante de 4,75 kbits/s, isto é, foram obtidos 6300 arquivos com compressão simples. Logo em seguida, tais arquivos foram decodificados para o domínio do tempo e codificados novamente na mesma taxa de 4,75 kbits/s, gerando 6300 arquivos com compressão dupla, tudo com o uso do codec padrão AMR (3GPP AMR Codec - Release 10. 2017).

Dessa forma, de cada arquivo AMR com compressão simples e dupla, foram extraídos, a título de ilustração e mediante leitura binária dos arquivos codificados, os seguintes parâmetros: três índices dos subvetores LSF, índice do dicionário adaptativo do 1º subquadro, ganho de dicionário para o 1º subquadro e ganho de dicionário para o 3º subquadro. Assim, para cada arquivo AMR, foi gerada uma sequência para cada um desses parâmetros, cujo tamanho depende do número de quadros codificados do arquivo e, em seguida, foram calculadas médias estatísticas exemplificativas para cada sequência. As definições matemáticas dessas médias são mostradas a seguir, em que $\hat{p}$ são os parâmetros quantizados e N_p é o número total de parâmetros por arquivo:

Média do parâmetro $\hat{p}$:

$$\bar{\hat{p}} = \frac{1}{N_p}\sum_{j=1}^{N_p}\hat{p}_j \, . \tag{4.1}$$

Média do parâmetro $\hat{p}$ elevado ao quadrado:

$$\overline{\hat{p}^2} = \frac{1}{N_p}\sum_{j=1}^{N_p}\hat{p}_j^2 \, . \tag{4.2}$$

Após as extrações dos parâmetros e aplicação das médias estatísticas, é necessário visualizar os resultados. Foram selecionados, a título de exemplo, dois parâmetros para a confecção de histogramas e gráficos de espalhamento (*scatter plot*) que permitissem perceber diferenças entre arquivos com compressão simples e dupla.

No histograma da Figura 4-1, que corresponde às médias dos índices do 3º subvetor LSF, não se mostra significativa a diferença de distribuições dos 6300 arquivos com compressão simples (barras escuras) e dos 6300 arquivos com compressão dupla (barras brancas), fato confirmado pelo gráfico de espalhamento da Figura 4-2 em que a dispersão é aproximadamente simétrica em relação à linha identidade (em vermelho), isto é, a compressão dupla não causa nenhuma tendência de aumento ou diminuição desse índice.

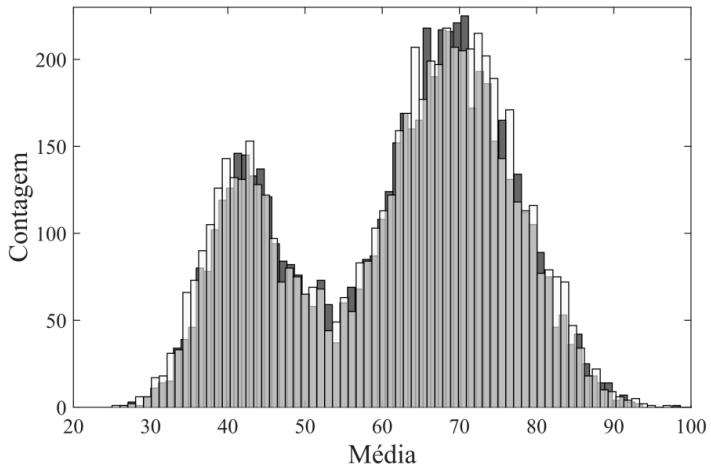


Figura 4-1 - Histograma da média do 3º subvetor LSF dos arquivos com compressão simples (barras escuras) e com compressão dupla (barras brancas).

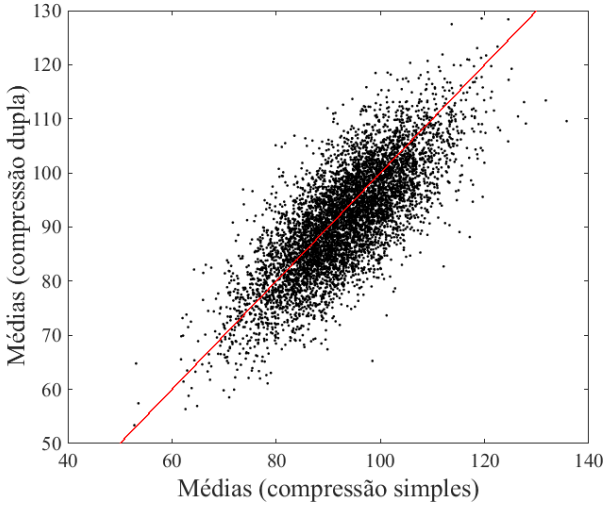


Figura 4-2 - Gráfico de espalhamento das médias do 3º subvetor LSF dos arquivos com compressão simples (eixo horizontal) e com compressão dupla (eixo vertical).

Já no histograma da Figura 4-3, que corresponde às médias dos ganhos de dicionário para o 1º subquadro elevados ao quadrado, também não é evidente a diferença de distribuições dos arquivos com compressão simples (barras escuras) e com compressão dupla (barras brancas). O gráfico de espalhamento da Figura 4-4 revela que a dispersão é

aproximadamente simétrica em relação à linha identidade, não permitindo atribuir diferenças significativas entre os arquivos com compressão simples e dupla.

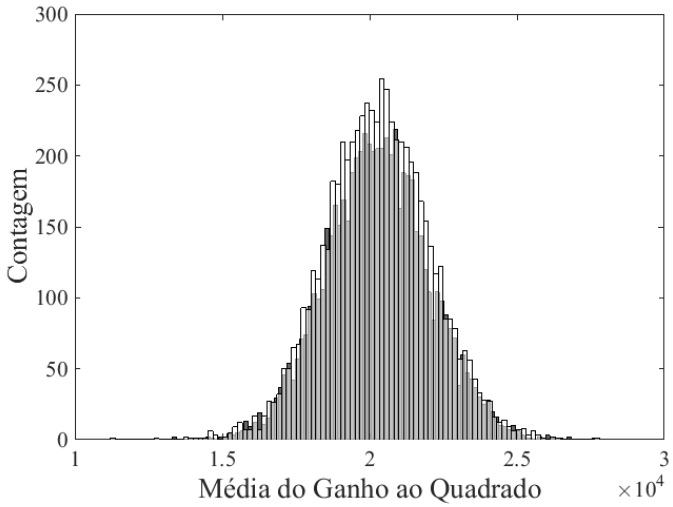


Figura 4-3 - Histograma da média dos ganhos de dicionário para o 1º subquadro elevados ao quadrado dos arquivos com compressão simples (barras escuras) e com compressão dupla (barras brancas).

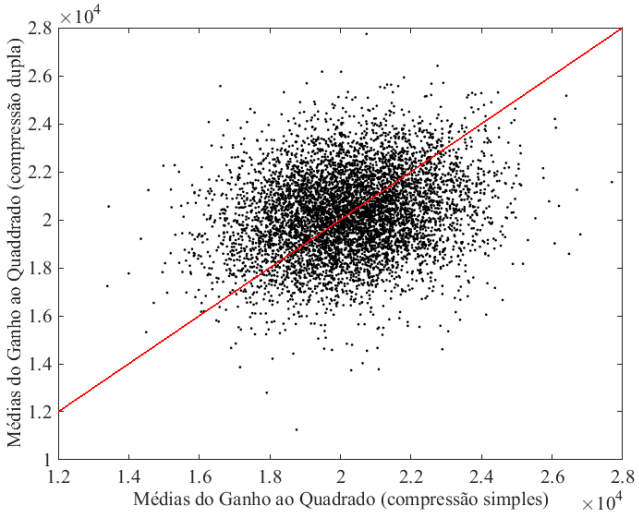


Figura 4-4 - Gráfico de espalhamento dos ganhos de dicionário para o 1º subquadro elevados ao quadrado dos arquivos com compressão simples (eixo horizontal) e com compressão dupla (eixo vertical).

Foi feita a mesma análise anterior para a maioria dos parâmetros do fluxo de bits dos arquivos AMR do *corpus* TIMIT e, ao final, todos os parâmetros analisados tiveram comportamento estatístico similar em relação à compressão dupla AMR, ou seja, sem tendências de discriminação entre arquivos com compressão simples ou dupla. Esse fato indica que o uso dos parâmetros do fluxo de bits AMR para a extração de características não é uma abordagem promissora para a detecção de compressão dupla AMR, uma vez que o processo de quantização pode modificar os valores e o comportamento estatístico

dos parâmetros do codificador. A detecção da compressão dupla AMR, entretanto, pode ser melhorada pela extração direta dos parâmetros do codificador mediante quantização inversa, conforme detalhado na próxima seção.

4.1.2. Extração a partir dos parâmetros do codificador

Levando em conta que a codificação de um arquivo AMR gera uma cadeia de parâmetros LP, LSP, *pitch lag* e ganhos que são dependentes do conteúdo de voz, uma comparação biunívoca de valores entre os respectivos parâmetros de arquivos com compressão simples e dupla provavelmente não ofereceria um resultado útil. Conforme a literatura mostra, a detecção da compressão dupla AMR é avaliada comparando um arquivo com compressão simples com sua versão com compressão dupla, em que a primeira e a segunda taxas de compressão podem ser diferentes (Shen *et al*., 2012). Portanto, uma abordagem mais eficaz é analisar o comportamento dos parâmetros do codificador AMR por meio da escolha de algumas medidas estatísticas para revelar possíveis diferenças entre arquivos com compressão dupla e simples.

A quantização inversa permite extrair parâmetros do codificador a partir do arquivo AMR codificado. Uma forma de se formular características para um arquivo AMR é calcular medidas estatísticas para esses parâmetros em cada arquivo AMR proveniente de um *corpus*. A estatística descritiva oferece uma série de medidas que podem ser usadas para a elaboração de características. Essas medidas descrevem o comportamento estatístico dos parâmetros do codificador ao longo dos arquivos e podem, a depender da natureza da medida, assumir valores diferentes para arquivos com compressão simples e dupla.

Para ilustrar algumas das características, considere-se a média aritmética. A média de todos os primeiros coeficientes LP (definido aqui como a_1) dos quadros de um arquivo AMR pode ser utilizada como uma característica desse arquivo. A mediana desses coeficientes também pode ser definida como uma característica. Assim sendo, se forem usados os dez coeficientes LP e LSP para medidas estatísticas diversas, o número de características pode ser aumentado. A quantidade e tipo de características no domínio da compressão necessárias para bem discriminar arquivos AMR com compressão simples e dupla são parâmetros difíceis de determinar, haja vista a complexidade envolvida no problema. Empiricamente é possível afirmar que, quanto maior a quantidade de características, mais provável é a melhoria da detecção, até certo limite nessa quantidade (dependendo, ainda, da rede neural usada). Conforme será explicado adiante nesta tese, na Seção 4.8, uma alta quantidade inicial de características é desejável para que seja possível

selecionar as melhores de acordo com a taxa AMR e o conjunto de treinamento do experimento em questão.

Com o intuito de avaliar o comportamento dos coeficientes LP na compressão dupla e concluir sobre a viabilidade do uso de medidas estatísticas para detectá-la, os arquivos do *corpus* TIMIT foram codificados para AMR na taxa constante de 4,75 kbits/s, isto é, foram obtidos 6300 arquivos com compressão simples. Logo após, esses arquivos foram decodificados e codificados novamente na mesma taxa de bits de 4,75 kbits/s, gerando 6300 arquivos com compressão dupla. Em seguida, um decodificador AMR modificado foi usado para extrair os coeficientes LP de cada arquivo (os detalhes de implementação desse decodificador serão discutidos na Seção 4.3), gerando 6300 arquivos binários com os coeficientes LP dos arquivos com compressão simples e 6300 arquivos com os de compressão dupla (Sampaio, 2019).

Cada arquivo binário de coeficientes LP deve, portanto, ser processado para computar medidas estatísticas para comparação entre arquivos com compressão simples e dupla. Por exemplo, para um arquivo do banco TIMIT, pode ser calculada a média dos coeficientes LP para sua versão AMR com compressão simples e a média para sua versão com compressão dupla. Calculando para todos os arquivos disponíveis, haverá 6300 médias de compressão simples e 6300 médias de compressão dupla que podem ser comparadas aos pares (cada par é originado do mesmo arquivo do banco TIMIT) ou todas juntas. Dessa forma, a depender da medida estatística utilizada, a discriminação entre compressão simples e dupla pode ser facilitada.

Como análise estatística preliminar, duas medidas foram selecionadas para caracterizar os coeficientes LP dos arquivos AMR (o conjunto completo de medidas estatísticas utilizadas nesta tese está descrito na Subseção 4.4.1). Tais medidas e definições matemáticas são mostradas a seguir, em que a_i são os coeficientes LP de i-ésima ordem no filtro LP (i=1...10) e N_p é o número total de coeficientes por arquivo:

Média do coeficiente a_i:

$$\bar{a}_i = \frac{1}{N_p}\sum_{j=1}^{N_p} a_{ij}\,. \tag{4.3}$$

Média do coeficiente a_i elevado ao quadrado:

$$\overline{a_i^2} = \frac{1}{N_p}\sum_{j=1}^{N_p} a_{ij}^2\,. \tag{4.4}$$

Após as extrações dos coeficientes LP e aplicação das medidas estatísticas, é necessário visualizar os resultados. Foram selecionados, a título de exemplo, dois coeficientes LP para a confecção de histogramas e gráficos de espalhamento que permitissem perceber diferenças entre arquivos com compressão simples e dupla.

No histograma da Figura 4-5, que corresponde à média do 7º coeficiente LP, pode ser observada a diferença de distribuições dos 6300 arquivos com compressão simples (barras escuras) e dos 6300 com compressão dupla (barras brancas), pois houve deslocamento à esquerda após a compressão dupla (as barras em cinza representam sobreposição de histogramas).

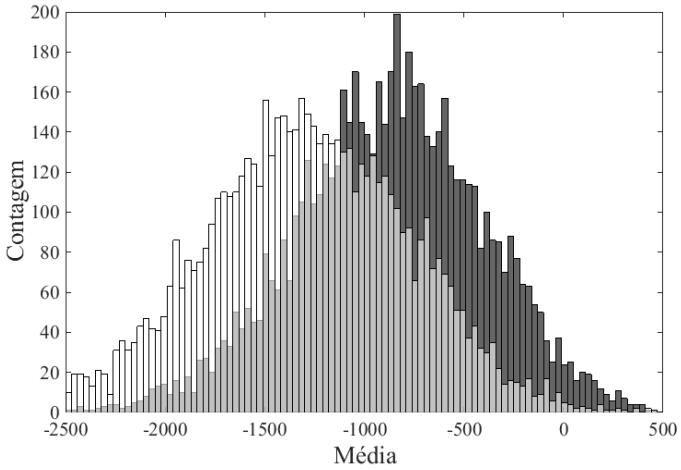


Figura 4-5 - Histograma da média do 7º coeficiente LP dos arquivos com compressão simples (barras escuras) e com compressão dupla (barras brancas).

Ainda para o mesmo coeficiente LP, a sua média em cada um dos 6300 arquivos com compressão simples e 6300 com compressão dupla pode ser vista no gráfico de espalhamento da Figura 4-6. Nessa figura cada ponto corresponde a um mesmo arquivo do banco TIMIT, cuja posição no gráfico é dada pela média do coeficiente LP do arquivo AMR com compressão simples e pela respectiva média para o arquivo com compressão dupla. É possível perceber com nitidez a tendência de diminuição dos valores das médias após a compressão dupla em pelo menos 96% dos arquivos, pois a mancha está visivelmente deslocada abaixo da linha identidade (em vermelho), ou seja, essa medida estatística tem um relevante poder discriminatório.

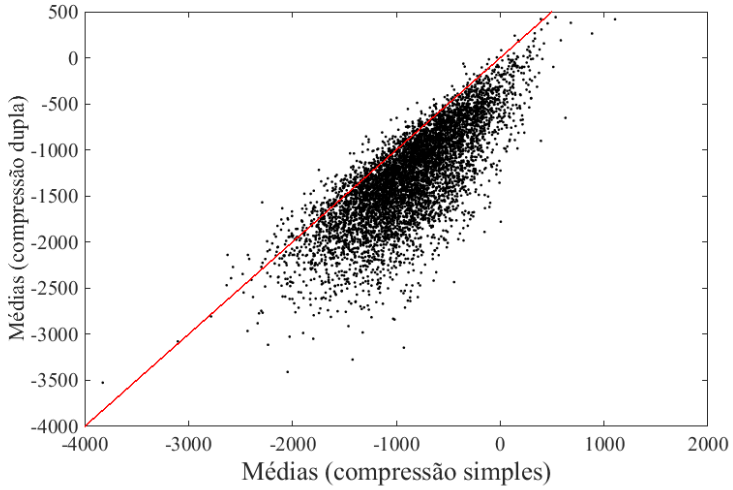


Figura 4-6 - Gráfico de espalhamento das médias do 7º coeficiente LP dos arquivos com compressão simples (eixo horizontal) e com compressão dupla (eixo vertical).

Já para o 6º coeficiente LP, a mediana dos desvios absolutos (MAD) revela maior diferença estatística entre os arquivos com compressão simples e dupla. No histograma da Figura 4-7 é possível observar que, além do deslocamento para a direita, a distribuição se torna mais alargada após a compressão dupla.

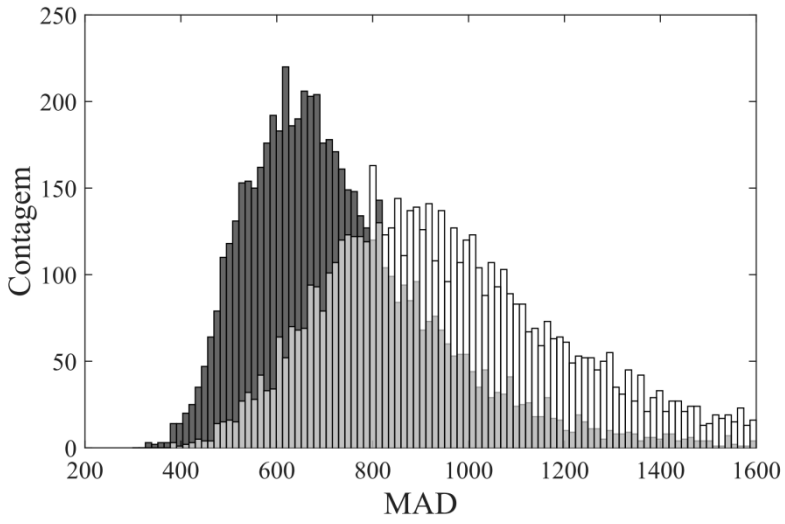


Figura 4-7 – Histograma da mediana dos desvios absolutos (MAD) do 6º coeficiente LP dos arquivos com compressão simples (barras escuras) e com compressão dupla (barras brancas).

Já o gráfico de espalhamento da Figura 4-8 mostra que a compressão dupla aumenta a MAD (em pelo menos 97% dos arquivos) para essa taxa de bits AMR, ou seja, essa medida estatística também tem um relevante poder discriminatório.

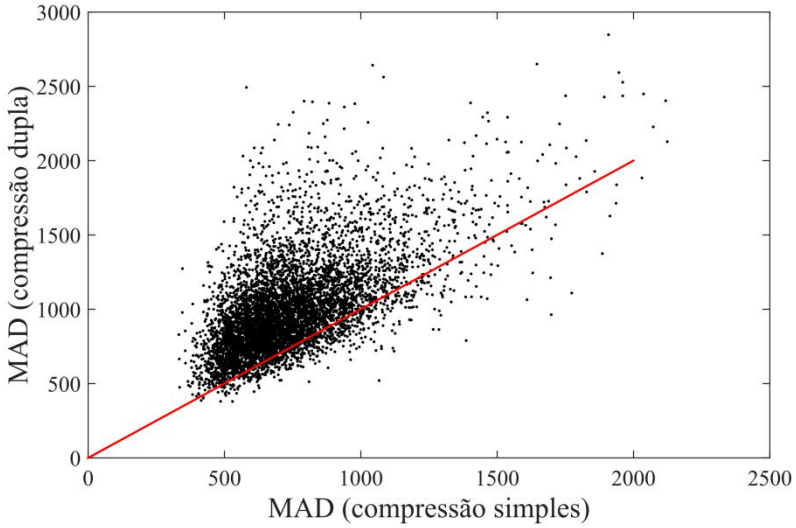


Figura 4-8 - Gráfico de espalhamento das médias do 4º coeficiente LP ao quadrado dos arquivos com compressão simples (eixo horizontal) e com compressão dupla (eixo vertical).

Embora tenham sido ilustrados apenas dois coeficientes LP com duas medidas estatísticas em que restou evidente a capacidade de discriminação entre compressão AMR simples e dupla, as análises dos demais coeficientes LP, além da análise de outros parâmetros, como LSP e *pitch*, com outras medidas estatísticas (detalhadas na Seção 4.4), demonstram que a quantização inversa é uma melhor abordagem para a construção de características dos arquivos AMR em comparação com o uso dos parâmetros do fluxo de bits. Por tal motivo, a quantização inversa para extração de parâmetros do codificador AMR foi adotada nesta tese para a elaboração de características que representam os arquivos AMR na entrada da rede neural (Sampaio, 2019).

4.2 - VISÃO GERAL DO MÉTODO PROPOSTO

Poucos trabalhos que propõem metodologias para identificar assinaturas espectrais diretamente no domínio da compressão são encontrados na literatura, conforme já comentado no capítulo anterior. Em termos gerais, a extração de características no domínio da compressão é baseada na DCT para o codec MP3. Já para o codec AMR, foram encontradas referências que investigaram características baseadas apenas nos parâmetros do fluxo de bits.

Uma nova metodologia baseada em SVM é proposta nesta tese para detectar a compressão dupla de sinais de voz diretamente nos arquivos codificados AMR. Os parâmetros do codificador AMR foram estudados e escolhidos para discriminar as duas classes de arquivos AMR, observando também as características do classificador SVM. A escolha da rede SVM como núcleo do método se deve aos bons resultados na detecção do

mesmo problema (Shen *et al*., 2012) e ao estudo realizado por Fathima e Kishnan (2018). Nesse estudo, o problema da detecção de compressão dupla AMR foi investigado usando os classificadores SVM, UBM-GMM e Bayesiano. Concluiu-se que a SVM é uma das melhores opções para o uso de características específicas do problema da compressão dupla AMR.

No desenvolvimento do método proposto com SVM, três etapas distintas podem ser mencionadas com progressivo aumento de desempenho: (1) introdução das características no domínio da compressão, (2) utilização do escalonamento robusto e (3) utilização da seleção de características.

Na etapa 1, uma modificação foi introduzida no método proposto em (Shen *et al.*, 2012) que provocou aumento no desempenho. A modificação foi a introdução de características no domínio da compressão no lugar das características acústicas baseadas nos sinais decodificados. As características foram baseadas em medidas estatísticas dos coeficientes LP e LSP extraídos dos arquivos AMR por quantização inversa (Sampaio, 2019) e a SVM foi ajustada para o uso de kernel RBF (*Radial Base Function*). Os testes com o *corpus* TIMIT mostraram que uma acurácia média de 94% foi atingida, superando o método inicial (84%). Para um maior detalhamento da etapa 1, incluindo experimentos e resultados, o trabalho descrito em Sampaio e Nascimento (2018) pode ser consultado.

Na etapa 2, um aumento significativo de desempenho foi adicionado ao método da etapa 1. As mesmas características no domínio da compressão foram utilizadas, mas um escalonamento robusto logístico generalizado (Cao *et al.*, 2016) foi usado (esse algoritmo é referenciado na Seção 4.6). O desempenho médio atingido com o *corpus* TIMIT e SVM usando kernel RBF foi de 98%, equivalente ao desempenho em Luo *et al.* (2017). Para um maior detalhamento da etapa 2, incluindo experimentos e resultados, o trabalho descrito em Sampaio e Nascimento (2019) pode ser consultado.

Na etapa 3, que corresponde ao método descrito nesta tese, houve um acréscimo de características em relação à etapa 2. Foram introduzidas características baseadas no *pitch lag*, ganhos de dicionário, patamares de ruído e energia de quadro, enquanto o método de escalonamento foi o mesmo da etapa 2. Foi introduzido também o algoritmo de seleção de características com eliminação recursiva de características e redução de polarização por correlação (SVM-RFE CBR), proposto por Yan e Zhang (2015), para calcular a classificação das características. Essa abordagem permite avaliar o número ótimo de características para cada conjunto de treinamento e para cada taxa de bits do padrão AMR. Um aumento de desempenho foi conseguido com o uso da seleção de características,

kernel RBF na SVM e com o *corpus* TIMIT, com acurácia média atingida de 99,16%, além de alto desempenho na análise de robustez (Sampaio e Nascimento, 2020).

O diagrama de blocos da Figura 4-9 mostra uma visão geral do método proposto, descrito em detalhes a seguir. Um *corpus* de voz é a entrada para o algoritmo e no diagrama é identificado como o bloco *Corpus*. O *corpus* TIMIT é inicialmente usado para a avaliação de desempenho, porém outro *corpus* pode ser usado em outras análises, como na análise de robustez.

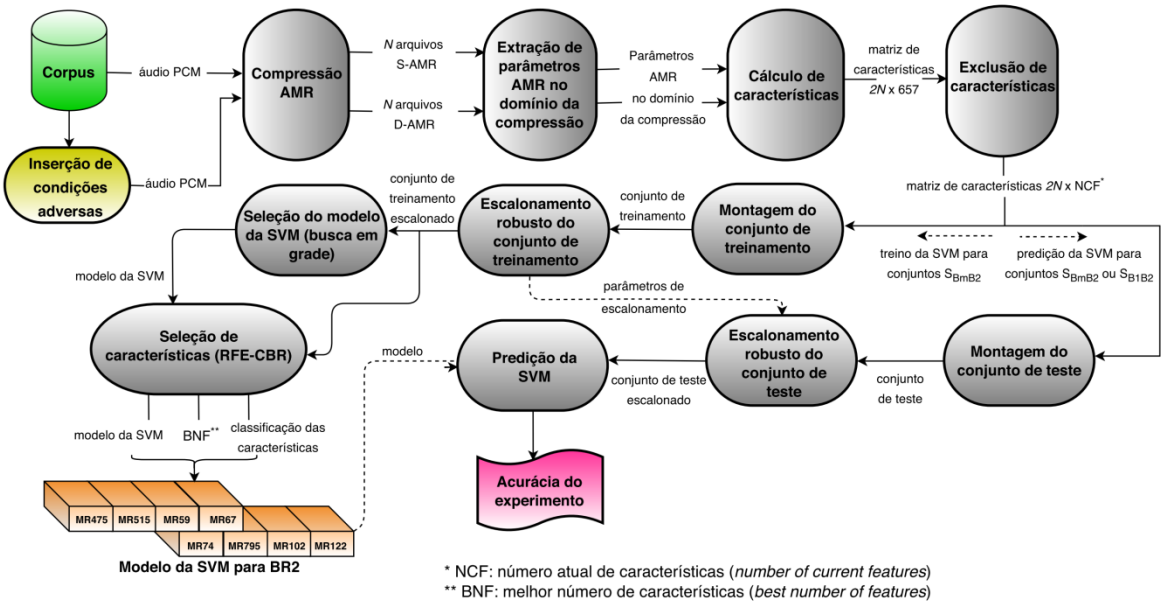


Figura 4-9- Diagrama de blocos do método proposto.

Logo à direita do bloco *Corpus* está o bloco *Compressão AMR*. Esse módulo é responsável pela construção de um banco de dados de arquivos AMR com compressão simples (chamados doravante de S-AMR) e com compressão dupla (chamados de D-AMR). Para os *N* arquivos do *corpus* que contêm *N* formas de onda distintas e que estão na entrada desse módulo, são produzidos 2*N* arquivos AMR na saída. O primeiro grupo de *N* arquivos é construído usando compressão AMR simples, enquanto o segundo grupo de *N* arquivos com compressão dupla. Esses 2*N* arquivos são codificados nas oito taxas padrão AMR (4,75 kbits/s até 12,2 kbits/s), respectivamente, e correspondem ao que é chamado um conjunto comprimido *S*.

Podem existir dois tipos de conjuntos comprimidos, de acordo com a proposta de Shen *et al.* (2012). No primeiro tipo, BR1 designa a primeira taxa de bits AMR quando o sinal é submetido à compressão dupla e BR2 indica a segunda taxa de bits quando o mesmo sinal é submetido à compressão dupla. Para o grupo dos *N* primeiros arquivos com compressão simples de um mesmo conjunto comprimido, BR2 também designa a

respectiva taxa de compressão (no caso, a taxa de compressão simples). Por exemplo, na Figura 4-10 é ilustrado um conjunto comprimido desse tipo, chamado de S_{B1B2}, em que BR2= 4,75 kbits/s e BR1=12,2 kbits/s.

No segundo tipo de conjunto comprimido, os *N* arquivos do *corpus* que devem ser submetidos à compressão dupla são subdivididos em 8 subgrupos (com aproximadamente *N*/8 arquivos por subgrupo). Cada subgrupo de arquivos é comprimido em AMR usando uma das 8 possíveis taxas de compressão. Logo após, os *N* arquivos (o total dos 8 subgrupos) são comprimidos pela segunda vez usando uma taxa BR2 conforme definida anteriormente. O grupo dos *N* arquivos com compressão simples é construído conforme descrito para os conjuntos comprimidos S_{B1B2}. Para designar o segundo tipo de conjunto comprimido, o acrônimo S_{BmB2} é usado neste trabalho, onde *Bm* se refere à característica multitaxa (as oito possíveis taxas de bits) da primeira taxa de compressão dos arquivos com compressão dupla. Um exemplo de conjunto S_{BmB2} é mostrado na Figura 4-10. Para um único *corpus*, considerando as oito taxas AMR, é possível criar 64 conjuntos S_{B1B2} e 8 conjuntos S_{BmB2}.

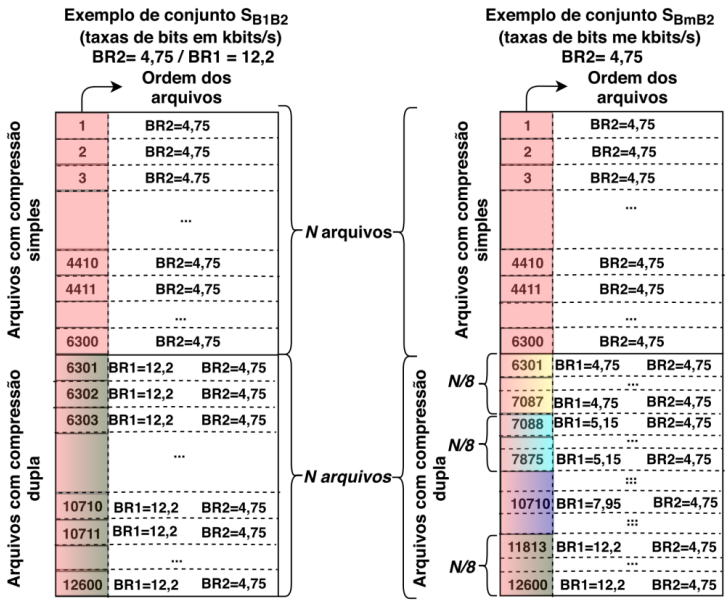


Figura 4-10 - Exemplos de conjuntos S_{B1B2} e S_{BmB2} de arquivos com BR2=4,75kbits/s e BR1=12,2kbits/s. Se for usado o *corpus* TIMIT, *N*=6300.

O outro módulo abaixo do bloco *Corpus* consiste em um algoritmo que insere condições adversas no sinal original do *corpus* e que podem mascarar o desempenho de detecção de compressão dupla AMR. As condições adversas tratadas pelo bloco *Inserção de condições adversas* são: mudança de *corpus*, duração fixa dos arquivos de áudio, ataque de deslocamento de quadros e contaminação do áudio por ruído. Tais condições são avaliadas no Capítulo 7. Neste trabalho, robustez é entendida como a habilidade do

algoritmo de manter um desempenho aceitável na detecção correta da compressão dupla AMR mesmo sob condições adversas, de acordo com a metodologia proposta em Luo *et al*. (2017).

A entrada do bloco *Extração de parâmetros AMR no domínio da compressão* são os *2N* arquivos do conjunto comprimido, formado por arquivos S-AMR e D-AMR. O fluxo de bits de cada arquivo é desempacotado sem a decodificação do sinal de áudio. Algumas mudanças introduzidas no código fonte do decodificador padrão AMR permitem a leitura do fluxo de bits dos arquivos AMR e permitem aplicar quantização inversa para obter os parâmetros do codificador úteis para a construção de características. Os coeficientes LP, LSP e os ganhos e períodos de *pitch* são alguns dos parâmetros do codificador utilizados para o cálculo das características. A maioria das variáveis usadas na construção das características são escalares, com exceção dos coeficientes LP e LSP, que são vetores de ordem 10. As variáveis extraídas podem ser conferidas na Tabela 4-2 da Seção 4.3 com os detalhes da extração.

A entrada do bloco *Cálculo de características* recebe um conjunto de parâmetros extraídos no bloco anterior e sua saída produz uma matriz de características. Essas características são obtidas pelo cálculo de medidas estatísticas sobre os parâmetros extraídos do codificador. Um total de 8 matrizes de características são construídas de acordo com os 8 possíveis conjuntos comprimidos S_{BmB2}. Já para os 64 possíveis conjuntos comprimidos S_{B1B2}, 64 matrizes de características podem ser construídas. Todas as matrizes de características têm *2N* linhas e número de colunas que corresponde ao número total de características. Os arquivos S-AMR são descritos pelas primeiras *N* linhas da matriz de características, enquanto os arquivos D-AMR, que são parte do mesmo conjunto comprimido, são descritos pelas *N* linhas restantes da matriz. O número máximo de características gerado pelo método proposto é 657. Esse assunto será discutido em mais detalhes na Seção 4.4.

As características produzidas no bloco *Cálculo de características* são apresentadas como entrada no bloco *Exclusão de características*. Esse módulo é responsável por analisar as matrizes de características de dimensão $2N \times 657$, construídas no módulo anterior, e por verificar possíveis inconsistências. Essas inconsistências são associadas ao comportamento estatístico dos parâmetros do codificador. Características com valores zero ou constantes ao longo dos arquivos AMR não produzem informação útil para os processos de treinamento e predição. Elas também causam inconsistências no algoritmo de escalonamento adotado no método proposto (esses módulos podem ser vistos no diagrama

de blocos como os blocos *Escalonamento robusto do conjunto de treinamento* e *Escalonamento robusto do conjunto de teste*). Por esses motivos, tais características são eliminadas, reduzindo também o grau de complexidade do algoritmo e ao mesmo tempo melhorando a eficiência do método. O número de características após a eliminação é definido como o número atual de características (NCF, do inglês *number of current features*), de tal forma que NCF $\leq$ 657. Portanto, a saída do bloco *Exclusão de características* é uma matriz de dimensões *2N* × *NCF*. As características eliminadas dependem da taxa de bits AMR, do parâmetro do codificador usado e da medida estatística empregada. Por exemplo, médias geométricas ou harmônicas e certas distribuições de probabilidade de primeiros dígitos geram características passíveis de eliminação. As características excluídas podem ser conferidas em detalhes na Tabela 5-6 da Seção 5.5.

O fluxo da Figura 4-9 contém uma bifurcação após o bloco de *Exclusão de características*. O fluxo à esquerda leva ao algoritmo de treinamento da SVM, enquanto o fluxo à direita diz respeito ao procedimento de predição do método. Apenas os conjuntos comprimidos S_{BmB2} são usados para o treinamento da SVM, permitindo um treinamento mais genérico e balanceado (Shen *et al.*, 2012). Os conjuntos S_{B1B2} e S_{BmB2} são usados para a predição e avaliação de desempenho.

O fluxo do treinamento da SVM contém o bloco chamado *Montagem do conjunto de treinamento* que é responsável por duas tarefas: (1) definir o experimento e (2) extrair o conjunto de treinamento. Neste trabalho, um experimento é definido como um processo completo de treinamento e teste usando o mesmo conjunto de treinamento. Em outras palavras, sempre que o conjunto de treinamento é modificado, um novo experimento se inicia. Em um experimento, as linhas da matriz de características são reservadas na razão de 70% para o treinamento e 30% para o teste. Cada arquivo do *corpus* origina dois arquivos, S-AMR e D-AMR, e as características criadas a partir desses arquivos são sempre incluídas no conjunto de treinamento (ou no conjunto de teste, por consequência). Em outras palavras, as características originadas dos arquivos S-AMR e D-AMR sempre estão na mesma quantidade num experimento. Para construir um novo experimento, a matriz de características é misturada de tal forma que conjuntos de treinamento e teste diferentes são selecionados, resultando em modelos SVM e acurácias diferentes. Para avaliar o desempenho do método, a acurácia média obtida de todos os experimentos é considerada. Uma explicação detalhada sobre a natureza dos experimentos é dada na Seção 5.6.

À esquerda do bloco *Montagem do conjunto de treinamento* está o bloco

Escalonamento robusto do conjunto de treinamento. O estudo do comportamento estatístico das características, derivadas dos arquivos AMR gerados a partir do *corpus* TIMIT, mostrou que cerca de 10% dos valores das características eram valores atípicos em alguns casos (esse estudo é mostrado em detalhes na Seção 5.8). Os valores atípicos das características têm uma faixa dinâmica significativamente maior do que os demais valores da característica. Os resultados preliminares da avaliação de desempenho do método mostraram que os valores atípicos tinham influência prejudicial na classificação. Essa propriedade estatística das características motivou a investigação e uso de um método de escalonamento mais elaborado. Um algoritmo de escalonamento robusto foi escolhido para reduzir os efeitos prejudiciais dos valores atípicos e para melhorar a acurácia da SVM na identificação do padrão da compressão dupla AMR. Como os parâmetros de escalonamento calculados para o conjunto de treinamento devem ser os mesmos usados pelo conjunto de teste, tais parâmetros são armazenados e usados pelo bloco *Escalonamento robusto do conjunto de teste*.

O conjunto de treinamento escalonado é a entrada para o bloco *Seleção do modelo da SVM*. O propósito desse bloco é determinar os parâmetros do kernel SVM para cada conjunto de treinamento por meio da técnica de busca em grade descrita em Chang e Lin (2011) e da máxima acurácia de validação cruzada. Esse método popular foi usado porque ele apresenta uma implementação computacional mais direta e oferece bons resultados para o uso da SVM (Chang e Lin., 2011). A determinação dos parâmetros da SVM define o seu modelo para um conjunto de treinamento.

Para melhorar o desempenho do método proposto, o bloco *Seleção de características* analisa e seleciona as características escalonadas do conjunto de treinamento. A eliminação de características reduz a complexidade do algoritmo na fase de treinamento e o uso de características mais significativas permite uma convergência mais rápida (Guyon e Elisseeff, 2003). Se as características são ordenadas conforme a importância e um subconjunto delas tem melhor desempenho do que todas juntas, o problema do sobreajuste é aliviado (Yan e Zhang, 2015). No método de seleção de características embutido, a eliminação recursiva de características diminui o problema do sobreajuste e aumenta o desempenho do modelo. O algoritmo SVM-RFE CBR, escolhido para melhorar o desempenho do método proposto, foi projetado para um grande espaço de características, que é o caso da detecção de compressão dupla em arquivos AMR (Yan e Zhang, 2015). O algoritmo CBR foi incluído porque um número significativo de características apresenta correlação, de acordo com a análise experimental detalhada na

Seção 5.10. As entradas desse bloco são o modelo SVM e as características escalonadas do conjunto de treinamento.

O algoritmo de seleção de características calcula o ordenamento das características em que a primeira característica é a mais importante, de acordo com o critério adotado. O algoritmo usa, como critério de ordenamento, a medida de dispersão das características calculada da função objetivo da SVM (chamada também de função de Lagrange). A função objetivo é calculada usando os parâmetros do modelo da SVM, como os vetores suporte. No início da iteração, a função objetivo da SVM é calculada incluindo uma dada característica em avaliação. Logo em seguida, a função objetivo é calculada sem a característica em avaliação. A medida de dispersão da característica é dada pela diferença entre esses dois valores calculados. Após todas as medidas das características serem calculadas, a característica com a menor medida é eliminada. Os mesmos cálculos com a função objetivo da SVM são feitos novamente com as características restantes. No final, a última característica que sobrou será a melhor característica e a iteração começa novamente. Em relação ao algoritmo de CBR, as características fortemente correlacionadas e removidas acidentalmente são inseridas de volta no conjunto das características sobreviventes (Yan e Zhang, 2015). Esse procedimento evita o descarte de características significativas durante o cálculo do SVM-RFE.

Em adição, o bloco *Seleção de características* pode determinar o número de características que permite a mais alta acurácia de validação cruzada, definida neste trabalho como o melhor número de características (BNF). Um grupo de características é definido como um subconjunto das características ordenadas que contém essas características desde a primeira (a mais importante) até a *k*-ésima. O parâmetro BNF é achado pelo cálculo da acurácia de validação cruzada de todos os possíveis grupos de características, de k=1 até k=NCF (número atual das características), usando os parâmetros C (constante de penalidade) e γ (constante do kernel RBF) do modelo da SVM. Após o cálculo da acurácia de validação cruzada de cada um dos NCF grupos, o algoritmo busca a maior acurácia e respectivo número de características do grupo. O melhor número de características BNF é, portanto, o número k de características do grupo que fornece a máxima acurácia de validação cruzada. A saída do bloco *Seleção de características* produz o ordenamento das características e o parâmetro BNF para um dado experimento. Essas grandezas, juntas com o modelo da SVM, formam o *Modelo da SVM para BR2* que é armazenado para cada taxa AMR e cada experimento.

Retornando ao diagrama de blocos da Figura 4-9, onde está a bifurcação, e

seguindo o fluxo à direita, o bloco chamado *Montagem do conjunto de teste* é encontrado. Nesse módulo, a matriz de características é misturada e o conjunto de teste também é extraído da matriz de características para um experimento. É o mesmo algoritmo usado no bloco *Montagem do conjunto de treinamento* descrito anteriormente. Esse procedimento garante que o conjunto de teste tem diferentes vetores em relação ao conjunto de treinamento, condição essencial para a validade dos experimentos.

O bloco *Escalonamento robusto do conjunto de teste* realiza o escalonamento robusto com o mesmo algoritmo e com os mesmos parâmetros de escalonamento usados no bloco *Escalonamento robusto do conjunto de treinamento* (a seta tracejada na Figura 4-9 mostra essa conexão), tratando-se de procedimento essencial para o bom desempenho da SVM (Cao *et al.*, 2016). O conjunto de teste escalonado é a entrada para o bloco *Predição da SVM* que implementa a predição usando o conjunto de teste e o *Modelo da SVM para BR2* já armazenado. A escolha desse modelo é baseada na taxa de bits AMR do arquivo em avaliação. Após o processamento de cada experimento, a acurácia é armazenada para cada taxa de bits AMR com o propósito de obter a média de todos os experimentos realizados.

Um detalhamento mais aprofundado do método proposto pode ser observado nas próximas seções deste capítulo.

4.3 - EXTRAÇÃO DE PARÂMETROS NO DOMÍNIO DA COMPRESSÃO

No bloco *Extração de parâmetros AMR no domínio da compressão* são extraídos parâmetros do codificador usando o decodificador AMR com modificações, ao invés de realizar a extração de parâmetros AMR quantizados no seu fluxo de bits. Isso permite acessar os parâmetros diretamente por meio de quantização inversa e, como já demonstrado na Subseção 4.1.2, permite também perceber melhor os artefatos da compressão dupla. Foram introduzidas modificações mínimas no código fonte do decodificador AMR (3GPP AMR Codec – Release 10, 2017) para extrair os parâmetros listados na Tabela 4-2, que também inclui os nomes das variáveis do código fonte, os símbolos adotados e o número de parâmetros por quadro.

A análise do código fonte do decodificador AMR, escrito em linguagem C, permite extrair os parâmetros para sete arquivos binários pela modificação dos arquivos *decoder.c* e *sp_dec.c*. Após compilação, uma versão modificada do decodificador é utilizada sobre os arquivos AMR para gerar os arquivos binários de parâmetros, cujos tamanhos dependem do número de quadros de voz do áudio codificado. Esse procedimento também é conhecido

como desempacotamento (*unpacking*), pois o algoritmo de decodificação é usado para processar o áudio codificado apenas para extrair parâmetros do codificador.

Tabela 4-2 – Parâmetros do codificador extraídos dos arquivos AMR. Todas as variáveis são declaradas como inteiro de 32 bits.

Parâmetro	Nome da variável no código fonte	Símbolo	Quantidade por quadro
Coeficientes LP	A_t[]	a_i, $i=1...10$	40
Pares espectrais de linha	lsp[]	q_j, $j=1...10$	40
Período de *pitch* (parte inteira)	T0	T	4
Ganho de *pitch*	gain (MR122 e MR795) gain_pit (demais modos)	β	4
Energia do quadro	currEnergy	E	1
Ganho do dicionário fixo	gain_code (MR22 e MR795) gain_cod (demais modos)	g_c	4
Patamar de ruído	noiseFloor	NFL	1

Cada um dos parâmetros guarda relações específicas com o áudio codificado, levando alguma informação que pode ser útil para detectar a compressão dupla AMR nas oito taxas possíveis. Embora seja possível admitir que o uso de menos parâmetros fosse suficiente para discriminar a compressão dupla, os experimentos demonstraram que as acurácias com menos parâmetros não atingem o mesmo nível de desempenho. A título de informação, foram realizados experimentos considerando apenas os coeficientes LP e escalonamento padrão tipo min-máx, mantendo os demais procedimentos idênticos ao método proposto (Sampaio e Nascimento, 2018). A quantização inversa oferece os valores dos coeficientes LP não quantizados, ou seja, antes da transformação e quantização, correspondendo aos coeficientes a_i do filtro de predição linear. Conforme se observa nos resultados, as acurácias obtidas com o *corpus* TIMIT indicam que o emprego apenas dos coeficientes LP gera bons resultados (média de 93,6%), porém inferiores ao estado da arte (média de 98,83%) em Luo *et al*. (2017). Esse fato, em si, justifica o uso de mais parâmetros AMR para uma melhor detecção da compressão dupla AMR.

4.4 - CARACTERÍSTICAS ESTATÍSTICAS PROPOSTAS

Após a extração dos parâmetros no domínio da compressão, é necessário projetar características significativas para detecção de compressão dupla AMR. A técnica de engenharia de características (Heaton, 2016) foi usada para projetar um conjunto de características adequadas para a detecção de compressão dupla AMR. O principal objetivo

dessa técnica é criar características com uma maior dispersão de faixa dinâmica, ou seja, maior variância para melhor perceber a assinatura espectral associada à compressão dupla AMR para todas as taxas de bits.

Apesar da complexidade do problema, um conjunto de características estatísticas simples, embora numerosas, pode ser utilizado para descrever a compressão dupla. Esse conjunto foi elaborado após a realização de uma série de simulações baseadas no *corpus* TIMIT destinada a pesquisar o comportamento estatístico dos parâmetros do codificador AMR. As características estatísticas podem ser divididas em dois grupos: no primeiro, são agrupadas as medidas estatísticas básicas derivadas da estatística descritiva; no segundo, são descritas as distribuições de probabilidade de primeiros dígitos para os coeficientes LP e LSP.

4.4.1 - Estatísticas básicas

Um resumo das medidas estatísticas básicas utilizadas no método proposto pode ser observado na Tabela 4-3, em que *x* é o parâmetro do qual se calculam as características. Medidas estatísticas similares aplicadas aos parâmetros do fluxo de bits AMR, como média, desvio padrão, coeficiente de variação e obliquidade, já foram reportadas em Petracca *et al.*(2005).

Tabela 4-3 – Medidas estatísticas básicas utilizadas no método proposto.

Símbolo	Medida	Símbolo	Medida
$\overline{x}$	média	σ_x^2	variância
σ_x	desvio padrão	$Mo(x)$	moda
$Kurt(x)$	curtose	$\gamma_1(x)$	obliquidade
$\tilde{x}$	mediana	$max(x)$	valor máximo
$min(x)$	valor mínimo	$CV(x)$	coeficiente de variação
$\overline{x}_{geom}$	média geométrica	$\overline{x}_{harm}$	media harmônica
$meanabs(x)$	média dos módulos dos elementos	$meansqr(x)$	média dos elementos ao quadrado
ADev(x)	desvio absoluto da média	MAD(x)	mediana dos desvios absolutos
$trimmean(x)$		média excluindo 5% de valores atípicos	

As fórmulas a seguir definem as medidas em termos de comandos de MATLAB® (versão 7.8.0.347), em que $\boldsymbol{X}$ é um vetor formado por um dos parâmetros *x* da Tabela 4-2, ou seja, $\boldsymbol{X} = [x_1, x_2, \ldots x_{N_p}]$, em que N_p é o número de parâmetros extraídos do arquivo AMR:

- Média:

$$\bar{x} = mean(\boldsymbol{X}) \tag{4.5}$$

- Variância:

$$\sigma_x^2 = var(\boldsymbol{X}) \tag{4.6}$$

- Desvio padrão:

$$\sigma_x = std(\boldsymbol{X}) \tag{4.7}$$

- Moda:

$$Mo(x) = mode(\boldsymbol{X}) \tag{4.8}$$

- Curtose:

$$Kurt(x) = \frac{moment(\boldsymbol{X}, 4)}{(std(\boldsymbol{X}))\text{^}4} - 3 \tag{4.9}$$

onde *moment* (***X***,4) calcula o momento central de 4ª ordem de ***X***.

- Obliquidade:

$$\gamma_1(x) = skewness(\boldsymbol{X}) \tag{4.10}$$

- Mediana:

$$\tilde{x} = median(\boldsymbol{X}) \tag{4.11}$$

- Valor máximo:

$$max(x) = max(\boldsymbol{X}) \tag{4.12}$$

- Valor mínimo:

$$min(x) = min(\boldsymbol{X}) \tag{4.13}$$

- Coeficiente de variação:

$$CV(x) = \frac{std(\boldsymbol{X})}{mean(\boldsymbol{X})} \tag{4.14}$$

- Média geométrica:

$$\bar{x}_{geom} = geomean(abs(\boldsymbol{X})) \tag{4.15}$$

onde *abs*(***X***) é o módulo dos valores no vetor ***X***.

- Média harmônica:

$$\bar{x}_{harm} = harmmean(\boldsymbol{X}) \tag{4.16}$$

- Média dos módulos dos elementos:

$$meanabs(x) = mean(abs(\boldsymbol{X})) \tag{4.17}$$

- Média dos elementos ao quadrado:

$$meansqr(x) = mean(\boldsymbol{X}.\text{^}2) \tag{4.18}$$

- Desvio absoluto da média:

$$ADev(x) = mad(\boldsymbol{X}) \tag{4.19}$$

- Mediana dos desvios absolutos:

$$MAD(x) = mad(\boldsymbol{X}, 1) \tag{4.20}$$

- Média excluindo 5% de valores atípicos:

$$trimmean(x) = trimmean(\boldsymbol{X}, 5) \tag{4.21}$$

Alguns experimentos realizados demonstram a utilidade das medidas propostas aplicadas a parâmetros diferentes dos coeficientes LP e LSP, ou seja, aos parâmetros *T*, β, *E, g_c e NFL*. Como exemplo, considere-se o gráfico de espalhamento da Figura 4-11 em que a característica calculada foi *meanabs*(***X***), $\boldsymbol{X} = [g_{c1}, g_{c2}, \ldots g_{cN_p}]$, ou seja, as médias dos módulos dos ganhos de dicionário fixo para os arquivos AMR derivados do banco TIMIT com BR1=BR2=4,75 kbits/s, BR1 constante. Pela análise da figura, é possível ver que as médias, para quase todos os arquivos considerados, diminuíram após a compressão dupla (quase todos os pontos estão abaixo da linha identidade, em vermelho).

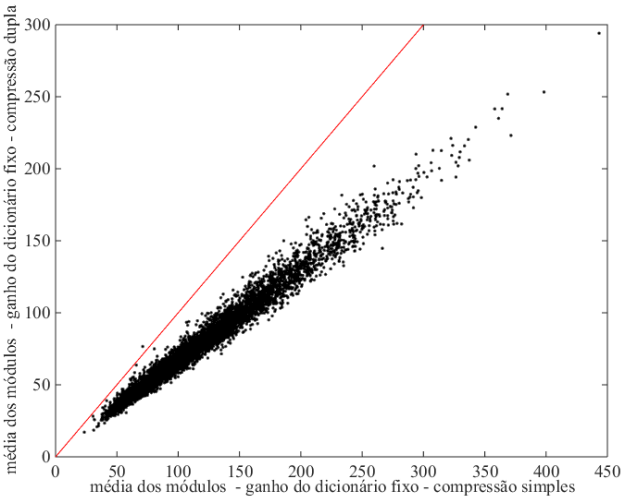


Figura 4-11 - Gráfico de espalhamento de *meanabs*($\boldsymbol{X}$), $\boldsymbol{X} = [\boldsymbol{g_{c1}}, \boldsymbol{g_{c2}}, \ldots \boldsymbol{g_{cN_p}}]$, para 6300 arquivos S-AMR e 6300 arquivos D-AMR.

4.4.2 - Distribuição de probabilidade dos primeiros dígitos

Além das estatísticas básicas da subseção anterior, as distribuições de probabilidade de primeiros dígitos dos coeficientes LP e LSP são também utilizadas. Os coeficientes LP e LSP produzidos pelo algoritmo do codificador AMR são representados em aritmética de ponto fixo, condição que permite a extração desses coeficientes para os arquivos binários em valores inteiros. Os primeiros dígitos ou dígitos significativos são os dígitos mais significativos de um inteiro sem considerar o dígito zero e, portanto, assumem valores naturais entre 1 e 9. Sua distribuição de probabilidade numa série numérica pode ser estimada respectivamente a partir da frequência normalizada do histograma da ocorrência desses primeiros dígitos e, de acordo com a Lei de Benford, numa série de números naturais, a distribuição de probabilidade dos primeiros dígitos apresenta um comportamento logarítmico (Fu *et al.*, 2007).

As primeiras aplicações da Lei de Benford para detectar compressão dupla foram estudadas para o codificador MP3. Observou-se que, para os coeficientes MDCT, a distribuição de probabilidade dos primeiros dígitos respeita a seguinte expressão (Yang *et al.*, 2010):

$$p(x) = \log_{10}\left(1 + \frac{1}{x}\right), x = 1,2...,9 \tag{4.22}$$

Os coeficientes MDCT seguem aproximadamente a Lei de Benford, porém, após a sua quantização, as distribuições de probabilidades dos primeiros dígitos (com compressão simples) variam com a taxa de bits e se apresentam como linhas aproximadamente retas numa escala bilogarítmica, cujas inclinações são função das taxas de bits. A compressão

dupla de arquivos de áudio MP3, por outro lado, implica a dupla quantização dos coeficientes da MDCT, o que leva a diferenças entre as propriedades dos coeficientes do áudio com compressão simples e do áudio com compressão dupla. Verifica-se que a compressão dupla afeta a distribuição de probabilidade dos primeiros dígitos dos coeficientes MDCT quantizados, de tal forma que a Lei de Benford não é mais válida. As distribuições resultantes da compressão dupla não mais se apresentam como linhas retas na escala bilogarítmica. Pelo contrário, tais distribuições se revelam como curvas irregulares em diferentes taxas de bits, de modo que, quanto maior a segunda taxa de bits da compressão dupla, maior é o desvio em relação à Lei de Benford (Yang *et al.*, 2010).

Esses resultados descrevem a possibilidade de discernir entre um arquivo MP3 comprimido uma única vez e outro arquivo submetido à compressão dupla, haja vista o potencial discriminador dessa propriedade. Embora o codificador AMR tenha sido desenvolvido com uma abordagem diferente do codificador MP3, uma analogia é proposta nesta tese para investigar a compressão dupla usando as distribuições de probabilidade de primeiros dígitos dos coeficientes LP e LSP extraídos dos arquivos AMR.

As distribuições de probabilidade dos primeiros dígitos dos coeficientes LP e LSP (a_i e q_j, respectivamente, i=1,2,...10, j =1,2,...10), estimadas pelas frequências normalizadas dos histogramas dos vetores $\boldsymbol{X}$, são, nesta tese, representadas por $m_x(k)$, onde k é o primeiro dígito, k=1,2,…9, e x pode ser a_i ou q_j, ou ainda a e q, para todos os coeficientes *agregados*. Por exemplo, $m_{a_2}(1)$ é calculado pela distribuição de probabilidade do dígito 1 quando $\boldsymbol{X} = [a_{21}, a_{22}, \ldots a_{2N_p}]$ (estimada pelo histograma calculado sobre o vetor $\boldsymbol{X}$), em que N_p assume aqui o valor da quantidade de subquadros do arquivo AMR. Já $m_a(1)$, que designa a distribuição de probabilidade do dígito 1 para os coeficientes LP agregados, é calculada para $\boldsymbol{X} = \left[a_{11}, a_{21}, a_{31}, \ldots, a_{101}, a_{12}, a_{22}, a_{32} \ldots a_{102} \ldots a_{1\,N_p} \ldots a_{10\,N_p}\right]$, em que a quantidade total de parâmetros agora é $10 \times N_p$ (cada subquadro do arquivo AMR gera 10 coeficientes LP para os modos AMR, exceto para os modos MR795 e MR122, que geram 20, conforme especificação do padrão AMR).

Inicialmente é interessante visualizar o comportamento das distribuições de probabilidades dos primeiros dígitos dos coeficientes LP e LSP de arquivos AMR com compressão simples em relação à Lei de Benford e se, após a compressão dupla, haveria alguma diferença nesse comportamento. Embora não seja escopo desta tese realizar uma análise matemática mais aprofundada desse comportamento, é possível experimentalmente fazer algumas observações tomando novamente o *corpus* TIMIT para concluir pelo uso ou

não das distribuições de probabilidades dos primeiros dígitos como características. Os experimentos computacionais mostraram que essas distribuições de probabilidades de primeiros dígitos não obedecem à Lei de Benford. No entanto, as distribuições dos arquivos S-AMR são diferentes daquelas dos arquivos D-AMR.

Por exemplo, considerando 6300 arquivos AMR com compressão simples e 6300 arquivos com compressão dupla, gerados a partir do *corpus* TIMIT, todos na taxa de 4,75kbits/s para BR1 e BR2 (conjunto comprimido S_{B1B2}), foi realizada a extração de todos os coeficientes LP. Após a estimativa das distribuições de probabilidades de primeiros dígitos de todos os coeficientes LP dos arquivos AMR com compressão simples (todos os coeficientes agregados para todos os 6300 arquivos) e a mesma estimativa para os 6300 arquivos com compressão dupla, foi gerada a Figura 4-12 em escala bilogarítmica (log-log).

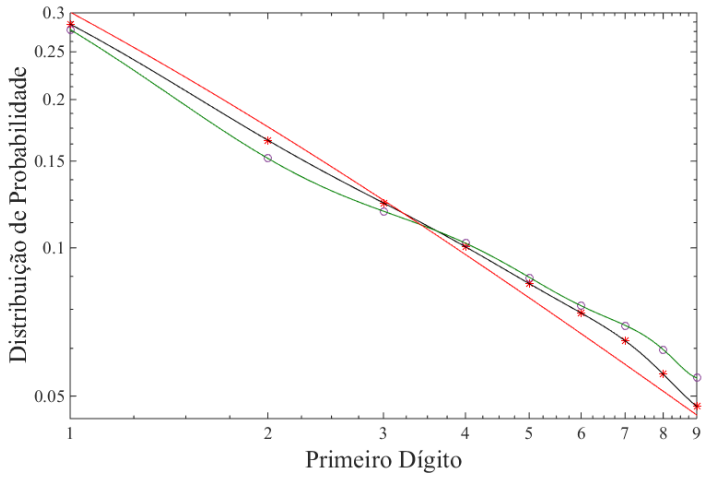


Figura 4-12 – Distribuições de probabilidade $m_a(k)$ dos primeiros dígitos k dos coeficientes LP de 6300 arquivos AMR com compressão simples (curva em preto e marca "*") e 6300 arquivos AMR com compressão dupla (curva em verde e marca "o"). A linha reta em vermelho corresponde à distribuição da Lei de Benford

Essa figura indica que nem os arquivos AMR com compressão simples, nem com compressão dupla, têm coeficientes LP que obedecem a Lei de Benford, pois as distribuições não se ajustam à linha reta na escala. Apesar disso, as distribuições dos arquivos com compressão simples (pontos com "*" e curva interpolada em preto) não são as mesmas dos arquivos com compressão dupla (pontos com "o" e curva interpolada em verde), o que indica uma diferença que pode ser útil para a detecção de compressão dupla.

Explorando os coeficientes LP separados, para o coeficiente a_{10} (10^{o} coeficiente LP) foi feito o mesmo experimento. Considerando os arquivos AMR com taxa BR1=4,75kbits/s e BR2=12,2kbits/s (conjunto comprimido S_{B1B2}), foi realizada a extração

de todos os coeficientes a_{10}. Após a estimativa das distribuições de probabilidade dos primeiros dígitos de todos os coeficientes a_{10}, foi gerada a Figura 4-13 em escala bilogarítmica.

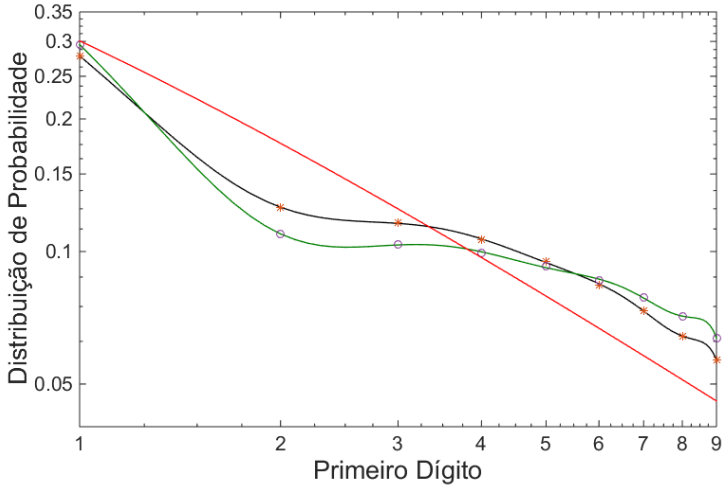


Figura 4-13– Distribuições de probabilidade $\boldsymbol{m_{a_{10}}(k)}$ dos primeiros dígitos *k* dos coeficientes a_{10} de 6300 arquivos AMR com compressão simples (curva em preto e marca "*") e 6300 arquivos AMR com compressão dupla (curva em verde e marca "o"). A linha reta em vermelho corresponde à distribuição da Lei de Benford

Essa figura indica que os arquivos AMR com compressão simples ou dupla têm coeficientes a_{10} que não obedecem a Lei de Benford. Observa-se, entretanto, que a diferença de distribuição de probabilidades entre os arquivos com compressão simples e dupla é mais nítida nesse caso. Esse fato indica novamente que as distribuições de probabilidade de primeiros dígitos podem ser usadas como características para a detecção da compressão dupla AMR.

Os resultados anteriores indicam que as distribuições de probabilidade de primeiros dígitos podem ser úteis para detectar compressão dupla AMR. Para visualizar os efeitos da compressão dupla de forma mais ilustrativa, outro experimento foi realizado com cada um dos arquivos TIMIT de forma individual, isto é, as distribuições de probabilidade de primeiros dígitos foram calculadas considerando apenas os coeficientes de cada um dos 6300 arquivos. Dessa forma, é possível observar como as distribuições de probabilidade dos primeiros dígitos se comportam no arquivo na versão com compressão simples e na versão com compressão dupla (mais uma vez, BR1=BR2=4,75kbits/s).

Na Figura 4-14 é mostrado o gráfico de espalhamento das distribuições $m_a(9)$ para todos os coeficientes LP em cada um dos 6300 arquivos AMR (*corpus* TIMIT). Observa-se que, para a maioria dos arquivos, essa distribuição aumenta após a compressão dupla (mancha deslocada acima da linha identidade, em vermelho).

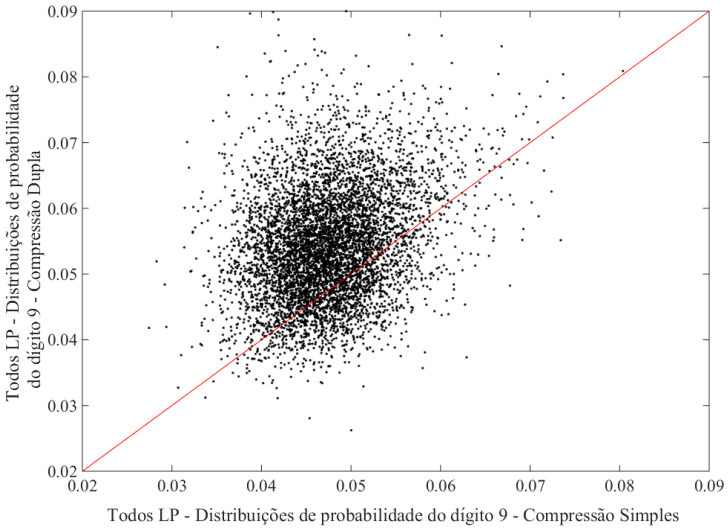


Figura 4-14 – Gráfico de espalhamento das distribuições de probabilidade $m_a(9)$ dos primeiros dígitos 9 dos coeficientes LP de 6300 arquivos S-AMR e D-AMR.

Algo similar ocorre para o gráfico de espalhamento das distribuições de probabilidade $m_{a_1}(9)$ em cada um dos 6300 arquivos AMR (*corpus* TIMIT), conforme a Figura 4-15 (BR1=BR2=4,75kbits/s). Mais uma vez se observa que a maioria dos arquivos apresentou aumento de $m_{a_1}(9)$ após a compressão dupla, o que demonstra a utilidade dessa medida.

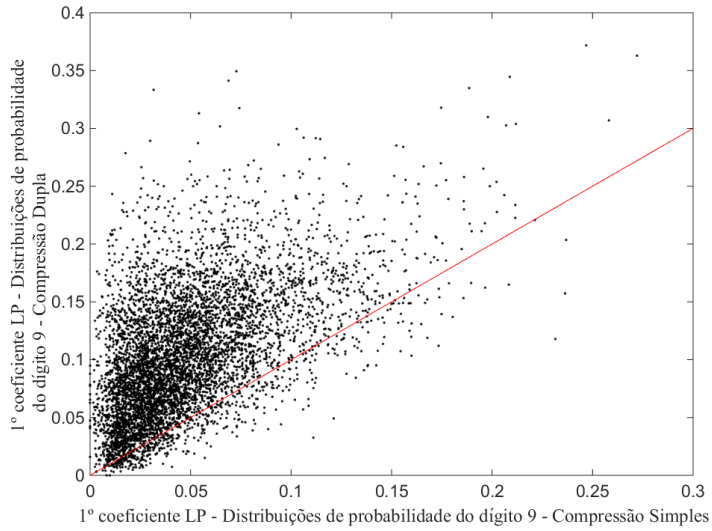


Figura 4-15 - Gráfico de espalhamento das distribuições de probabilidade $m_{a_1}(9)$ dos primeiros dígitos 9 do 1º coeficiente LP de 6300 arquivos S-AMR e D-AMR.

Na Figura 4-16 é mostrado o gráfico de espalhamento das distribuições de probabilidade $m_q(3)$ do dígito 3 para todos os coeficientes LSP em cada um dos 6300 arquivos AMR (*corpus* TIMIT, BR1=BR2=4,75kbits/s). Observa-se que, para a maioria dos arquivos, o valor da distribuição diminui após a compressão dupla (mancha deslocada

abaixo da linha identidade, em vermelho), demonstrando que essa medida é útil para ser usada como característica.

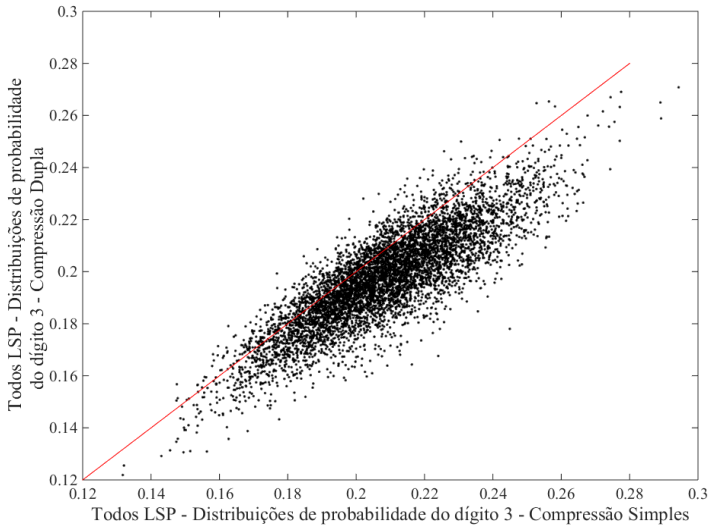


Figura 4-16 - Gráfico de espalhamento das distribuições de probabilidade $m_q(3)$ dos primeiros dígitos 3 de todos os coeficientes LSP de 6300 arquivos S-AMR e D-AMR.

Foi constatado, ainda, que determinados primeiros dígitos não ocorrem em certos coeficientes LSP nos arquivos AMR derivados do *corpus* TIMIT. Para q_1, não ocorrem os primeiros dígitos 1, 4, 5, 6, 7, 8 e 9; já para q_2 e q_9 não ocorrem os primeiros dígitos 4, 5, 6, 7, 8 e 9 (essa propriedade é descrita na Seção 5.5). Apesar desse fato, as distribuições de probabilidade de primeiros dígitos dos demais coeficientes LSP podem ser utilizadas para a detecção de compressão dupla AMR. Por exemplo, o gráfico de espalhamento da Figura 4-17 (*corpus* TIMIT, BR1=BR2=4,75kbits/s), correspondente a $m_{q_3}(3)$, demonstra que as distribuições de probabilidade do primeiro dígito 3 aumentam na maioria dos arquivos AMR após a compressão dupla (poucos pontos estão abaixo da linha identidade, em vermelho).

Importante ressaltar que os exemplos dados nesta subseção foram realizados com o *corpus* TIMIT na taxa BR1=BR2=4,75 kbits/s, com primeira taxa de compressão constante para todo o *corpus*. Entretanto, foi realizada a mesma análise para todas as sete taxas AMR restantes, sendo constatado que os comportamentos dos primeiros dígitos dos coeficientes LP e LSP eram muito semelhantes ao apresentado. Essas observações permitiram concluir que o uso das distribuições de probabilidade dos primeiros dígitos dos coeficientes LP e LSP são úteis para a composição de uma matriz de características.

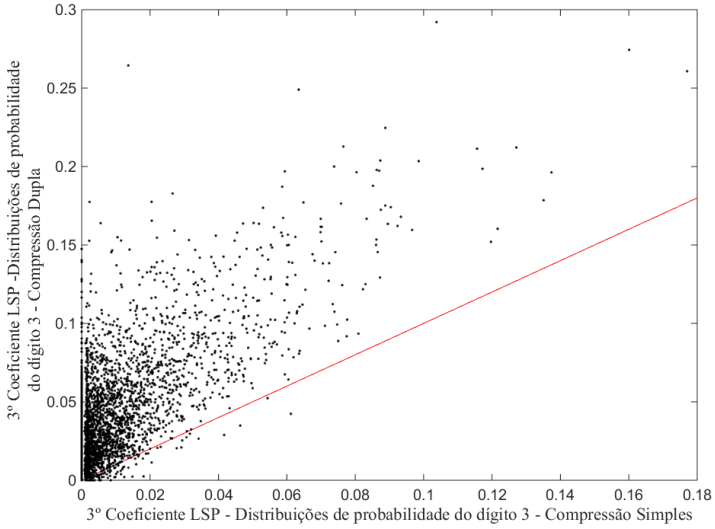


Figura 4-17 - Gráfico de espalhamento das distribuições de probabilidade $\boldsymbol{m_{q_3}(3)}$ dos primeiros dígitos 3 do 3º coeficiente LSP de 6300 arquivos S-AMR e D-AMR.

4.4.3 - Matriz de características

A extração de parâmetros no domínio da compressão e as medidas estatísticas propostas têm por finalidade elaborar uma forma eficiente de representar cada arquivo AMR, ou seja, elaborar um vetor de características para ser usado como entrada de uma SVM. Dessa forma, seja o vetor linha de características do arquivo *k* de um *corpus* considerado definido como:

$$\boldsymbol{\chi}_{\mathrm{k}} = [\chi_{1k}\, \chi_{2k}\, \chi_{3k} \cdots \chi_{TNFk}], \tag{4.23}$$

em que *TNF* é o número total de características. Considerando um *corpus* de tamanho *N*, define-se matriz de características $\boldsymbol{M}$ como a matriz formada pelos vetores de características $\boldsymbol{\chi}_{\mathrm{k}}$ dos *N* arquivos AMR com compressão simples, seguidos pelos vetores dos *N* arquivos AMR com compressão dupla, ou seja:

$$\boldsymbol{M} = \begin{bmatrix} \chi_{11} & \chi_{21} & \chi_{31} & \cdots & \chi_{TNF\,1} \\ \chi_{12} & \chi_{22} & \chi_{32} & \cdots & \chi_{TNF\,2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \chi_{1N} & \chi_{2N} & \chi_{3N} & \cdots & \chi_{TNF\,N} \\ \chi_{1(N+1)} & \chi_{2(N+1)} & \chi_{3(N+1)} & \cdots & \chi_{TNF(N+1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \chi_{1(2N)} & \chi_{2(2N)} & \chi_{3(2N)} & \cdots & \chi_{TNF(2N)} \end{bmatrix}. \tag{4.24}$$

Pela combinação da Tabela 4-3, das distribuições de probabilidade dos primeiros dígitos e da Tabela 4-2, são totalizadas as características na quantidade *TNF* = 657 da maneira a seguir descrita. Considerando os coeficientes LP e LSP, para cada a_i e q_j, i,j=1...10, e todos os dez coeficientes *a* e *q* agregados, é possível calcular as estatísticas de forma individual, contando 10 coeficientes a_i, 10 coeficientes q_i, um grupo de coeficientes

agregados *a* e um grupo de coeficientes agregados *q*, totalizando 22 parâmetros. Essa forma de cálculo é análoga ao método proposto para o caso dos coeficientes MDCT em que eles são considerados globalmente (agregados) ou por faixas de frequência (Yang *et al.*, 2010). As 17 medidas da Tabela 4-3 e as 9 distribuições de probabilidade de primeiros dígitos são aplicadas aos 22 coeficientes LP e LSP, totalizando 22 × (17 + 9) = 572 características. De volta à Tabela 4-2, os 5 parâmetros restantes (T, β, E, g_c *e* NFL) foram utilizados apenas para o cálculo das características da Tabela 4-3, resultando em mais 17 × 5 = 85 características adicionais. Dessa forma, o número total de características possíveis no método proposto é 572 + 85 = TNF = 657.

O número total de características é deliberadamente elevado para cumprir melhor o propósito de descrever o problema da compressão dupla AMR. Importante considerar que o conjunto de características proposto é usado para as oito taxas AMR e, a princípio, para qualquer *corpus* de voz em diferentes condições, com a finalidade de manter as acurácias elevadas. Conforme será visto no algoritmo de seleção de características e na análise de robustez, o número elevado de características é um dos fatores primordiais para a manutenção do desempenho do algoritmo proposto nas mais variadas condições.

Para melhor expressar a matriz $\boldsymbol{M}$, resta definir como estarão dispostas as 657 características nos vetores $\boldsymbol{\chi}$. Na Tabela 4-4 é realizado um mapeamento por números para as características e que coincidem com as colunas em que elas estão dispostas na matriz $\boldsymbol{M}$. Essa convenção será importante durante os experimentos tratados mais adiante nesta tese.

Tabela 4-4 – Mapeamento da matriz de características. Os números correspondem às colunas dos vetores de características.

	a	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	q	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	T	β	E	g_c	NFL
$\overline{x}$	1	27	53	79	105	131	157	183	209	235	261	287	313	339	365	391	417	443	469	495	521	547	573	590	607	624	641
σ_x^2	2	28	54	80	106	132	158	184	210	236	262	288	314	340	366	392	418	444	470	496	522	548	574	591	608	625	642
σ_x	3	29	55	81	107	133	159	185	211	237	263	289	315	341	367	393	419	445	471	497	523	549	575	592	609	626	643
Mo	4	30	56	82	108	134	160	186	212	238	264	290	316	342	368	394	420	446	472	498	524	550	576	593	610	627	644
Kurt	5	31	57	83	109	135	161	187	213	239	265	291	317	343	369	395	421	447	473	499	525	551	577	594	611	628	645
γ_1	6	32	58	84	110	136	162	188	214	240	266	292	318	344	370	396	422	448	474	500	526	552	578	595	612	629	646
$\widetilde{x}$	7	33	59	85	111	137	163	189	215	241	267	293	319	345	371	397	423	449	475	501	527	553	579	596	613	630	647
max	8	34	60	86	112	138	164	190	216	242	268	294	320	346	372	398	424	450	476	502	528	554	580	597	614	631	648
min	9	35	61	87	113	139	165	191	217	243	269	295	321	347	373	399	425	451	477	503	529	555	581	598	615	632	649
CV	10	36	62	88	114	140	166	192	218	244	270	296	322	348	374	400	426	452	478	504	530	556	582	599	616	633	650
$\overline{x}_{geom}$	11	37	63	89	115	141	167	193	219	245	271	297	323	349	375	401	427	453	479	505	531	557	583	600	617	634	651
$\overline{x}_{harm}$	12	38	64	90	116	142	168	194	220	246	272	298	324	350	376	402	428	454	480	506	532	558	584	601	618	635	652
meanabs	13	39	65	91	117	143	169	195	221	247	273	299	325	351	377	403	429	455	481	507	533	559	585	602	619	636	653
meansqr	14	40	66	92	118	144	170	196	222	248	274	300	326	352	378	404	430	456	482	508	534	560	586	603	620	637	654
ADev	15	41	67	93	119	145	171	197	223	249	275	301	327	353	379	405	431	457	483	509	535	561	587	604	621	638	655
MAD	16	42	68	94	120	146	172	198	224	250	276	302	328	354	380	406	432	458	484	510	536	562	588	605	622	639	656
trimmean	17	43	69	95	121	147	173	199	225	251	277	303	329	355	381	407	433	459	485	511	537	563	589	606	623	640	657
$m_x(1)$	18	44	70	96	122	148	174	200	226	252	278	304	330	356	382	408	434	460	486	512	538	564					
$m_x(2)$	19	45	71	97	123	149	175	201	227	253	279	305	331	357	383	409	435	461	487	513	539	565					
$m_x(3)$	20	46	72	98	124	150	176	202	228	254	280	306	332	358	384	410	436	462	488	514	540	566					
$m_x(4)$	21	47	73	99	125	151	177	203	229	255	281	307	333	359	385	411	437	463	489	515	541	567					
$m_x(5)$	22	48	74	100	126	152	178	204	230	256	282	308	334	360	386	412	438	464	490	516	542	568					
$m_x(6)$	23	49	75	101	127	153	179	205	231	257	283	309	335	361	387	413	439	465	491	517	543	569					
$m_x(7)$	24	50	76	102	128	154	180	206	232	258	284	310	336	362	388	414	440	466	492	518	544	570					
$m_x(8)$	25	51	77	103	129	155	181	207	233	259	285	311	337	363	389	415	441	467	493	519	545	571					
$m_x(9)$	26	52	78	104	130	156	182	208	234	260	286	312	338	364	390	416	442	468	494	520	546	572					

4.5 - EXCLUSÃO DE CARACTERÍSTICAS E MATRIZES DE TREINAMENTO

Devido às propriedades dos parâmetros AMR e das medidas estatísticas utilizadas, é necessário realizar um pré-processamento na matriz de características antes do treinamento da SVM. Por exemplo, conforme já mencionado na Subseção 4.4.2, alguns primeiros dígitos não acontecem em certos coeficientes LSP, resultando em distribuições de probabilidade iguais a zero. Como a matriz de características tem uma metade para arquivos com compressão simples e a outra para aqueles com compressão dupla, deve ser realizada inicialmente uma busca por características que tenham desvio padrão nulo em cada metade de cada vez, e, se encontradas em alguma delas, a característica inteira é eliminada. Após esse procedimento, uma nova matriz de características $\boldsymbol{M_F}$ é computada com menos colunas que a original, estabelecendo um novo número atual das características NCF para cada taxa AMR. Assim, define-se a matriz $\boldsymbol{M_F}$ como:

$$\boldsymbol{M_F} = \begin{bmatrix} \chi_{11} & \chi_{21} & \chi_{31} & \cdots & \chi_{NCF\,1} \\ \chi_{12} & \chi_{22} & \chi_{32} & \cdots & \chi_{NCF\,2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \chi_{1N} & \chi_{2N} & \chi_{3N} & \cdots & \chi_{NCF\,N} \\ \chi_{1(N+1)} & \chi_{2(N+1)} & \chi_{3(N+1)} & \cdots & \chi_{NCF(N+1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \chi_{1(2N)} & \chi_{2(2N)} & \chi_{3(2N)} & \cdots & \chi_{NCF(2N)} \end{bmatrix}, \tag{4.25}$$

em que $\boldsymbol{\chi}_{\mathrm{k}} = [\chi_{1k}\,\chi_{2k}\,\chi_{3k}\cdots\chi_{NCFk}]$ é o vetor de características do evento k após reordenamento e eliminação de características.

A matriz $\boldsymbol{M_F}$ tem características de arquivos AMR com compressão simples (primeira metade das linhas) e dupla (segunda metade); porém, algumas frações delas precisam ser extraídas para formar as matrizes de treinamento e teste. Antes, contudo, é preciso embaralhar os eventos (linhas) da metade superior de $\boldsymbol{M_F}$ (compressão simples) e da metade inferior (compressão dupla) para extrair matrizes de treinamento e teste diferentes, de forma que seja possível computar experimentos distintos. O embaralhamento é feito de modo que as matrizes de treinamento e teste sempre tenham, respectivamente, o mesmo evento (arquivo TIMIT) nas versões com compressão simples e dupla, ou seja, o embaralhamento das duas metades de $\boldsymbol{M_F}$ é feito com a mesma permutação e, posteriormente, as matrizes de treinamento e teste são extraídas. A montagem da matriz de treinamento é feita extraindo uma fração (por exemplo, 70%) das características de arquivos com compressão simples e a mesma fração de arquivos com compressão dupla. A matriz de teste é montada com as linhas remanescentes para incluir também características de compressão simples e dupla. Esse procedimento é necessário para que a SVM seja

treinada com as versões em compressão simples e dupla das características dos mesmos arquivos AMR, aplicando-se o mesmo raciocínio para o teste. A extração das matrizes de treinamento e teste é obrigatória antes do escalonamento, pois ambas as matrizes devem ser escalonadas com os mesmos parâmetros. Assumindo a fração de 70% para a extração da matriz de treinamento e a forma de extração proposta, definem-se as matrizes de treinamento $\boldsymbol{M_{Tr}}$ e a matriz de teste $\boldsymbol{M_{Te}}$ como:

$$\boldsymbol{M_{Tr}} = \begin{bmatrix} \chi_{11} & \chi_{21} & \chi_{31} & \cdots & \chi_{NCF\,1} \\ \chi_{12} & \chi_{22} & \chi_{32} & \cdots & \chi_{NCF\,2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \chi_{1floor(0{,}7*N)} & \chi_{2floor(0{,}7*N)} & \chi_{3floor(0{,}7*N)} & \cdots & \chi_{NCF\,floor(0{,}7*N)} \\ \chi_{1(N+1)} & \chi_{2(N+1)} & \chi_{3(N+1)} & \cdots & \chi_{NCF\,(N+1)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \chi_{1[N+floor(0{,}7*N)]} & \chi_{2[N+floor(0{,}7*N)]} & \chi_{3[N+floor(0{,}7*N)]} & \cdots & \chi_{NCF\,[N+floor(0{,}7*N)]} \end{bmatrix} \text{ e} \tag{4.26}$$

$$\boldsymbol{M_{Te}} = \begin{bmatrix} \chi_{1[floor(0{,}7*N)+1]} & \chi_{2[floor(0{,}7*N)+1]} & \chi_{3[floor(0{,}7*N)+1]} & \cdots & \chi_{NCF\,[floor(0{,}7*N)+1]} \\ \chi_{1[floor(0{,}7*N)+2]} & \chi_{2[floor(0{,}7*N)+2]} & \chi_{3[floor(0{,}7*N)+2]} & \cdots & \chi_{NCF\,[floor(0{,}7*N)+2]} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \chi_{1N} & \chi_{2N} & \chi_{3N} & \cdots & \chi_{NCF\,N} \\ \chi_{1[N+floor(0{,}7*N)+1]} & \chi_{2[N+floor(0{,}7*N)+1]} & \chi_{3[N+floor(0{,}7*N)+1]} & \cdots & \chi_{NCF\,[N+floor(0{,}7*N)+1]]} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \chi_{1\,2N} & \chi_{2\,2N} & \chi_{3\,2N} & \cdots & \chi_{NCF\,2N} \end{bmatrix}. \tag{4.27}$$

Na Figura 4-18 é esboçado, para maior clareza, o procedimento de extração das matrizes $\boldsymbol{M_{Tr}}$ e a $\boldsymbol{M_{Te}}$ a partir da matriz $\boldsymbol{M_F}$. Os números indicados são as linhas das matrizes desenhadas. Essa extração é realizada a cada novo experimento, conforme será detalhado na Seção 5.7.

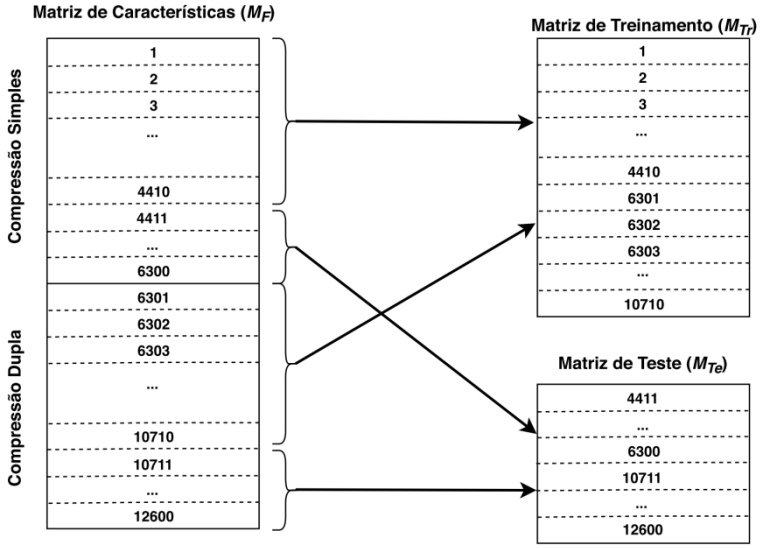


Figura 4-18 – Procedimento de extração das matrizes de treinamento e teste. Os números representam as linhas das matrizes para o *corpus* TIMIT.

4.6 - MÉTODO DE ESCALONAMENTO UTILIZADO

O escalonamento é uma etapa muito importante antes do uso das características no

treinamento da SVM e sua ausência pode potencialmente degradar os resultados dos experimentos. Existem algumas técnicas para escalonar dados, como os algoritmos min-max e *z-score*, mas foi verificado que o tipo de escalonamento conhecido como robusto (Cao *et al.*, 2016) é mais adequado para as matrizes de características AMR porque elas apresentam numerosos valores atípicos e valores concentrados em pequenas faixas (a análise de valores atípicos será apresentada na Seção 5.8). O número de valores atípicos observados nas características de arquivos AMR pode ser originado da variabilidade entre os locutores do banco TIMIT, provocando o surgimento de faixas de concentração de valores.

Foi escolhida uma ferramenta de escalonamento robusto proposta por Cao *et. al* (2016) para processar as características e que é baseada no algoritmo logístico generalizado (GL) que escalona os dados de forma uniforme numa dada faixa, mesmo se houver valores atípicos (um desenvolvimento matemático mais detalhado desse algoritmo pode ser encontrado no Apêndice A desta tese). A escolha do algoritmo GL se deve à sua capacidade de minimizar a influência dos valores atípicos nas características escalonadas, uma vez que, em se tratando de algoritmos de classificação, os valores atípicos são normalmente prejudiciais para o seu desempenho. Já a elevada concentração de valores também prejudica a classificação, pois esconde a verdadeira disposição das características. Pelo formato da função GL, conclui-se que o algoritmo permite, ao mesmo tempo, trazer os valores atípicos para valores mais próximos da maioria dos dados e desconcentrar valores para melhor uso na classificação.

Além do escalonamento robusto, outra constatação foi a existência de características esparsas (com poucos elementos não nulos) e que aumentam a quantidade de zeros nos vetores de características. As SVM são algoritmos bem adequados para problemas com eventos esparsos e espaços de entrada com altas dimensões (Joachims, 1998), aumentando, a princípio, seu desempenho nessas condições. Considerando essa propriedade, as características esparsas recebem um tratamento adicional no método proposto. As características que contêm quantidade considerável de elementos de valor zero são submetidas ao escalonamento robusto, porém mantendo esses valores zero, haja vista que o escalonamento substituiria esses zeros por um valor constante (em geral, pelo valor da metade da faixa de escalonamento). Para manter os valores zero, eles são localizados e extraídos de cada característica com uma quantidade mínima de zeros, sendo recuperados após o escalonamento robusto, conforme o diagrama explicativo da Figura 4-19, em que os valores x_k e x'_k são, respectivamente, os valores de uma dada

característica antes e após o escalonamento robusto.

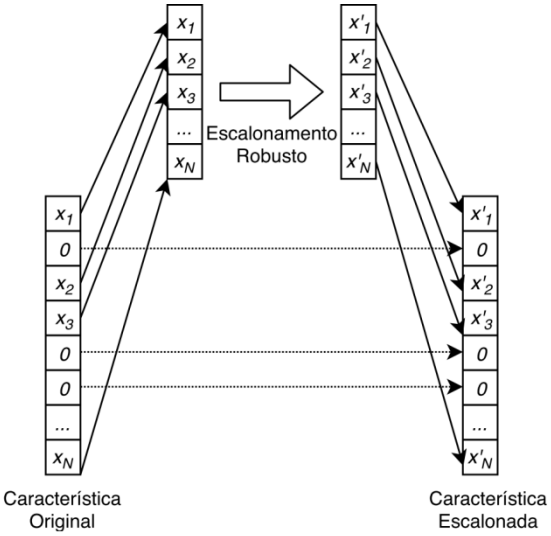


Figura 4-19 – Procedimento para o escalonamento dos elementos nulos das características esparsas no método proposto.

Após o escalonamento robusto, são geradas as matrizes de treinamento escalonada $\boldsymbol{M'_{Tr}}$ e a matriz de teste escalonada $\boldsymbol{M'_{Te}}$ que serão usadas para a seleção do modelo da SVM. Conforme já mencionado, $\boldsymbol{M'_{Tr}}$ e $\boldsymbol{M'_{Te}}$ são calculadas ao mesmo tempo para os conjuntos S_{BmB2}, enquanto para os conjuntos S_{B1B2} $\boldsymbol{M'_{Te}}$ é calculada posteriormente.

4.7 - SVM E SELEÇÃO DE MODELO

A formulação matemática da SVM é baseada em algoritmo de otimização com aproximações para a função de núcleo. Dependendo dessa função, existem alguns parâmetros que necessitam de ajuste para a predição da SVM, procedimento que é chamado de seleção de modelo da SVM. Um detalhamento teórico do algoritmo da SVM pode ser encontrado no Apêndice B desta tese.

Após o escalonamento das matrizes $\boldsymbol{M_{Tr}}$ e $\boldsymbol{M_{Te}}$, é necessário selecionar um modelo de SVM para cada BR2 e para cada experimento por meio de uma busca em grade 12 × 12 por parâmetros otimizados da SVM. Partindo de uma matriz de treinamento escalonada $\boldsymbol{M'_{Tr}}$, o núcleo RBF é usado para a SVM, o qual permite dois ajustes: o parâmetro de penalidade C e a constante γ. O núcleo RBF foi escolhido após uma série de experimentos preliminares com os núcleos linear, quadrático e polinomial. Nesses experimentos comparativos, realizados com os mesmos conjuntos de treinamento, foi observado que o núcleo RBF apresentou acurácia média superior aos demais núcleos. Esse resultado está de acordo com o trabalho prévio desenvolvido com SVM em que foi usado o núcleo RBF (Shen *et al.*, 2012).

Foi seguida a metodologia proposta por Chang e Lin (2011) e realizada uma busca

em grade que usa um procedimento de validação cruzada de *n* dobras para determinar C e γ que maximizam as acurácias. Inicialmente são fixados valores de C e γ cobrindo uma faixa larga de valores para iniciar uma busca solta (*loose*) numa grade 12 × 12 em que a maior acurácia é encontrada num par C e γ. Logo após, as vizinhanças de C e γ que proporcionam essa máxima acurácia são divididas em um passo 1/12. Então uma busca refinada (*fine*) é feita dentro dessas vizinhanças para achar a máxima acurácia de validação cruzada e, por fim, uma segunda busca refinada é feita da mesma forma.

No final, são selecionados C e γ que proporcionam a máxima acurácia de validação cruzada para um dado conjunto de treinamento. O procedimento de seleção do modelo da SVM usa apenas as matrizes $\boldsymbol{M}'_{Tr}$ dos conjuntos S_{BmB2}, pois essas matrizes têm todas as oito taxas de primeira compressão. Para o processamento dos conjuntos S_{B1B2}, não é necessário treinar a SVM porque já existem os modelos SVM para BR2, os quais carregam $\boldsymbol{M_{Tr}}$, apenas para o escalonamento, e a permutação usada na sua extração, de modo que não sejam testados os mesmos eventos que foram usados no treinamento. Dessa forma, os conjuntos S_{B1B2} são usados apenas para extrair a matriz de teste $\boldsymbol{M_{Te}}$ e para computar as acurácias.

4.8 - SELEÇÃO DE CARACTERÍSTICAS

O treinamento da SVM pode ser feito usando todas as características atuais disponíveis (NCF) na matriz $\boldsymbol{M}'_{Tr}$, assim como o teste pode ser realizado usando todas as características da matriz $\boldsymbol{M}'_{Te}$. No trabalho descrito em Sampaio e Nascimento (2019), o treinamento e predição da SVM foram realizados exatamente dessa forma, ou seja, as matrizes de treinamento e teste foram submetidas ao escalonamento robusto e foram mantidas todas as características disponíveis. Os resultados obtidos naquele estudo foram satisfatórios, uma vez que atingiram acurácias da ordem de 98%, aproximadamente iguais àquelas mais altas até então reportadas em Luo *et al.* (2017).

Apesar dos resultados utilizando todas as características, é provável que seja atingido um melhor desempenho usando menos características do que as atuais (NCF), haja vista que as matrizes de treinamento mudam a cada experimento e as taxas de bits AMR podem assumir oito valores diferentes. Conclui-se que, considerando o alto número de características existentes, usar todas elas para treinar a SVM, em todas as taxas AMR, tende a ser um procedimento subótimo.

Para melhorar o desempenho da detecção de compressão dupla AMR reportado na literatura, foram inseridos a mudança para o domínio da compressão e o escalonamento

robusto. Uma ideia para aperfeiçoar ainda mais o método surge do estudo das técnicas de seleção de características (FS, *feature selection*), que são algoritmos destinados à classificação das características em ordem de importância conforme um critério estabelecido. A técnica de seleção de características permite remover características irrelevantes, ruidosas e redundantes. O uso de menos características do que o total disponível para aumentar o desempenho de predição de uma rede neural, evitando o sobreajuste e simplificando os modelos calculados, é uma estratégia conhecida na disciplina de aprendizado de máquina (Guyon e Elisseeff, 2003). Como corolário importante do uso da FS, ela permite explorar mais profundamente o significado dos dados pela análise da importância das características, permitindo compreender quais delas melhor representam o problema em análise.

Após pesquisa dos algoritmos mais eficientes e adequados ao problema de escolher quais as melhores características num conjunto numeroso delas, a eliminação recursiva de características (RFE, *recursive feature elimination*) para SVM foi considerada uma boa opção para a tarefa. Essa técnica diminuiu o problema do sobreajuste e melhora o desempenho dos modelos específicos para o classificador SVM. Yan e Zhang (2015) propuseram um algoritmo de SVM-RFE CBR que também leva em conta a polarização causada por características altamente correlacionadas. Eles incorporaram a estratégia de redução de polarização por correlação (CBR – *correlation bias reduction*) no procedimento de eliminação de características, aperfeiçoando o método RFE original e superando as abordagens tradicionais, como a análise de componentes principais. O algoritmo SVM-RFE CBR está detalhado no Apêndice C desta tese.

A saída do algoritmo SVM-RFE CBR é um vetor F_{Rank} que pode ser usado para selecionar quais características serão de fato utilizadas na predição da SVM. O primeiro elemento de F_{Rank} é a característica *mais importante* (ou melhor) com base na métrica do algoritmo, seguida das demais com importância decrescente. Embora seja possível inicialmente propor o uso de apenas algumas das primeiras melhores características (as 50 primeiras, por exemplo), essa não foi a abordagem implementada no método proposto.

De posse de F_{Rank}, o algoritmo proposto para selecionar as características busca computar alguma medida com elas e escolher quais apresentam o melhor desempenho. Seja $\boldsymbol{M}'_{Trk}$ a matriz de treinamento escalonada em que as características (colunas) foram reordenadas conforme F_{Rank} e mantidas apenas as características desde a primeira até a *k*-ésima. Em outras palavras, enquanto a matriz $\boldsymbol{M}'_{Tr}$ tem linhas que correspondem a vetores com NCF (número atual das características) colunas na forma $\boldsymbol{\chi}' = [\chi'_1, \chi'_2, \chi'_3, \chi'_4, \ldots \chi'_{NCF}]$, a

matriz $\boldsymbol{M}'_{Trk}$ tem linhas com k colunas na forma $\boldsymbol{\chi}'_{rk} = [\chi'_{r1}, \chi'_{r2}, \chi'_{r3}, \chi'_{r4}, \ldots \chi'_{rk}]$, em que χ'_{ri} corresponde à i-ésima característica escalonada e ordenada conforme F_{Rank}. Para gerar um critério de desempenho simples e significativo para escolher o melhor k a ser usado ($1 \leq k \leq$ NCF) na matriz de treinamento $\boldsymbol{M}'_{Trk}$, a proposta desta tese é utilizar a acurácia de validação cruzada obtida com a SVM. Os parâmetros da função de núcleo Gaussiana C e γ são determinados na seleção do modelo SVM por busca em grade abordada na Seção 4.7. Com esses parâmetros, uma acurácia de validação cruzada é calculada pela SVM com as matrizes $\boldsymbol{M}'_{Trk}$ para cada k variando de 1 até NCF. Após o cálculo das NCF acurácias, a máxima acurácia é encontrada e o valor de k para essa acurácia é denominado BNF – melhor número de características (*best number of features*). Como a melhor acurácia de validação cruzada é um critério válido para predizer o desempenho do modelo com a matriz de teste, o melhor número de características BNF obtido com a matriz de treinamento é o melhor a ser usado para a predição da SVM com a matriz de teste ordenada com F_{Rank} e limitada a BNF características (Guyon e Elisseeff, 2003).

O fluxograma da Figura 4-20 resume o procedimento adotado para determinar o melhor número de características BNF de uma matriz de treinamento em uma determinada BR2, em que as linhas sólidas são do fluxo de processamento e as linhas tracejadas são de dados de entrada ou saída. No início, o algoritmo SVM-RFE CBR calcula F_{Rank} com a matriz de treinamento escalonada e os rótulos (+1 ou –1) dessa matriz como entradas. Em seguida, a matriz de treinamento é reordenada conforme F_{Rank}. Essa matriz é transformada em outra matriz com k características, em que as k primeiras são preservadas e as demais retiradas. Cada matriz com k características é usada para o cálculo da acurácia de validação cruzada com a SVM, a qual usa os parâmetros já calculados na busca em grade. Cada acurácia (Acc) é armazenada para cada matriz, num total de NCF (número atual das características) matrizes. Ao final de NCF iterações, haverá NCF acurácias, bastando determinar a máxima Acc e o número k de características usado no seu cálculo. Ao final, o melhor número de características BNF será igual ao número k de características que gerou a máxima Acc.

Para cada matriz de treinamento dos conjuntos S_{BmB2}, e para cada taxa de bits BR2, o melhor número de características BNF assumirá um valor diferente. Essa propriedade é determinante para o bom desempenho do método proposto, pois o número de características usadas se adapta às diferentes condições de treinamento e teste.

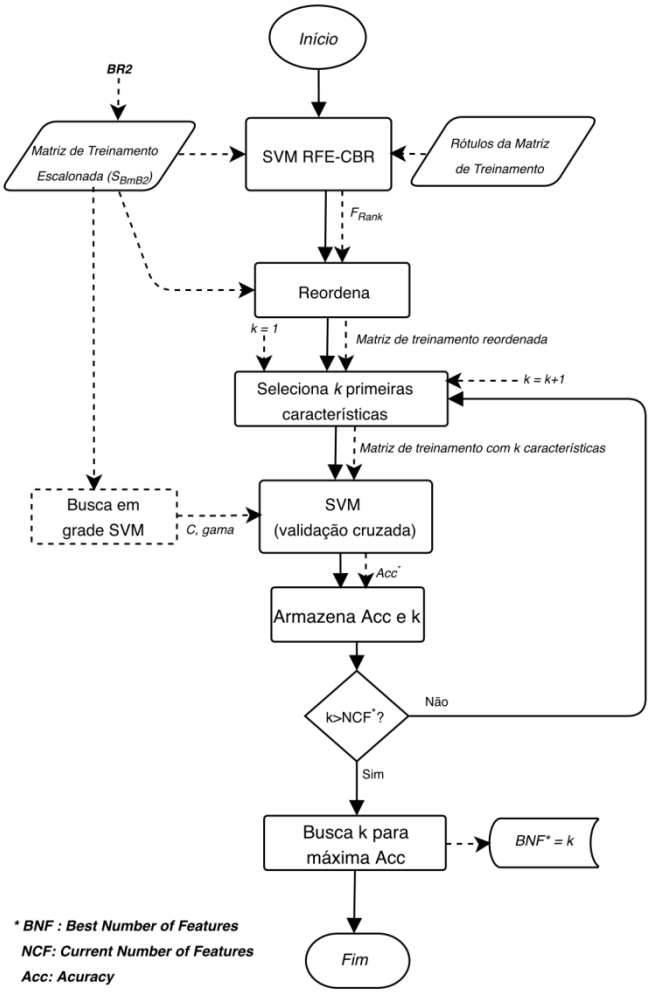


Figura 4-20 – Procedimento para a determinação do melhor número de características (BNF).

5- CONFIGURAÇÃO DOS EXPERIMENTOS

Após a concepção teórica do método proposto nesta tese, uma série de experimentos exaustivos foi realizada para constatar a sua eficácia. A fim de preservar a consistência e compatibilidade, foi seguida a configuração de experimentos proposta por Shen *et. al* (2012) e adotado o *corpus* TIMIT (Garofolo *et al.*, 1993) com durações dos arquivos inalteradas para todos os 20 experimentos realizados, exceto para a análise descrita no Capítulo 7. Nas seções seguintes, os principais detalhes de implementação dos experimentos serão explorados visando a uma melhor compreensão do método proposto.

5.1 - *CORPUS* TIMIT

O *corpus* TIMIT é formado por 6300 arquivos de áudio não comprimidos cujos conteúdos são 10 frases foneticamente compactas faladas por 630 locutores nativos em um dos 8 dialetos regionais do inglês dos Estados Unidos da América (*New England, Northern, North Midland, South Midland, Southern, New York City, Western* e *Army Brat*). A aplicação mais comum desse *corpus* é no teste de algoritmos de reconhecimento de fala, mas seu uso na área de autenticação de áudio forense se consolidou pela quantidade de arquivos e homogeneidade das gravações.

A duração dos arquivos do *corpus* TIMIT varia entre 915 ms até 7788 ms e eles foram codificados no formato SPHERE WAV, com taxa de amostragem de 16 kHz e amostras de 16 bits. Tal formato não é o mesmo WAV RIFF comumente encontrado, exigindo uma conversão de cabeçalho dos arquivos TIMIT originais. Para o uso do *corpus* TIMIT, foi necessário também fazer a conversão para a taxa de amostragem de 8 kHz por meio de filtragem *anti-aliasing*, preservando as amostras em 16 bits. Essa operação foi necessária para garantir a compatibilidade com o codificador AMR que exige em sua entrada áudio amostrado a 8 kHz e amostras de 13 bits.

Nesta tese, os arquivos originais do *corpus* TIMIT, que são disponibilizados ao longo de várias pastas numa estrutura de diretórios correspondente às frases e regiões dos dialetos, foram copiados de forma aleatória para uma única pasta, renomeados de 1 a 6300 e convertidos conforme já mencionado. A Tabela 5-1 traz alguns exemplos de arquivos do *corpus* TIMIT já convertidos para WAV RIFF 16 bits, incluindo também o conteúdo e demais informações dos arquivos.

Tabela 5-1 – Exemplos de arquivos do *corpus* TIMIT.

Informações (todos em formato WAV com taxa de amostragem 16 kHz, 16bits)		
Tamanho	Duração	Conteúdo (frase)
112 kB	3 s	*She had your dark suit in greasy wash water all year.*
96 kB	3 s	*Don't ask me to carry an oily rag like that.*
136 kB	4 s	*The revised procedure was acclaimed as a long-overdue reform.*
156 kB	5 s	*Severe myopia contributed to Ron's inferiority complex.*

5.2 - VISÃO GERAL E FLUXO DE DADOS

Os experimentos realizados com o *corpus* TIMIT convertido para 8 kHz estão resumidos na Figura 5-1, que é um diagrama de fluxo de dados com detalhes da implementação realizada em MATLAB® para os experimentos. As linhas tracejadas indicam entrada ou saída de apenas um dado, as linhas contínuas indicam transferência de dados e as linhas pontilhadas indicam laço de repetição. Inicialmente, ao observar o diagrama, percebe-se que, fixadas as entradas de dados *tipo conjunto*, *BR1* e *BR2*, os experimentos são repetidos 20 vezes para cada combinação de taxas de bits AMR e tipos de conjuntos comprimidos, que podem variar entre 8 e 2 valores, respectivamente. Dessa forma, para os conjuntos S_{BmB2}, são realizados 8×20 = 160 experimentos (20 experimentos para 8 taxas), enquanto para os conjuntos S_{B1B2}, são realizados 8×8×20=1280 experimentos (20 experimentos para 64 combinações de taxas). O *corpus* TIMIT com 6300 arquivos WAV é codificado para 6300 arquivos S-AMR e 6300 arquivos D-AMR, totalizando 12600 arquivos AMR. Tais arquivos são usados para o cálculo das características, resultando na matriz de características $\boldsymbol{M}$ de dimensões 12600 × 657. As 6300 primeiras linhas correspondem às características dos arquivos S-AMR, enquanto as 6300 últimas linhas correspondem às características dos arquivos D-AMR. Essa matriz é processada para eliminar as características nulas, originando uma matriz com NCF (número atual das características) colunas.

Para os conjuntos S_{BmB2}, os modelos SVM são calculados para cada BR2; já para os conjuntos S_{B1B2}, não há cálculo de modelos, pois são usados os mesmos calculados para os conjuntos S_{BmB2}. Dessa forma, para calcular os modelos, a matriz de treinamento é extraída da matriz de características de uma forma diferente para cada experimento. Nas seções seguintes, os módulos implementados serão descritos em mais detalhes.

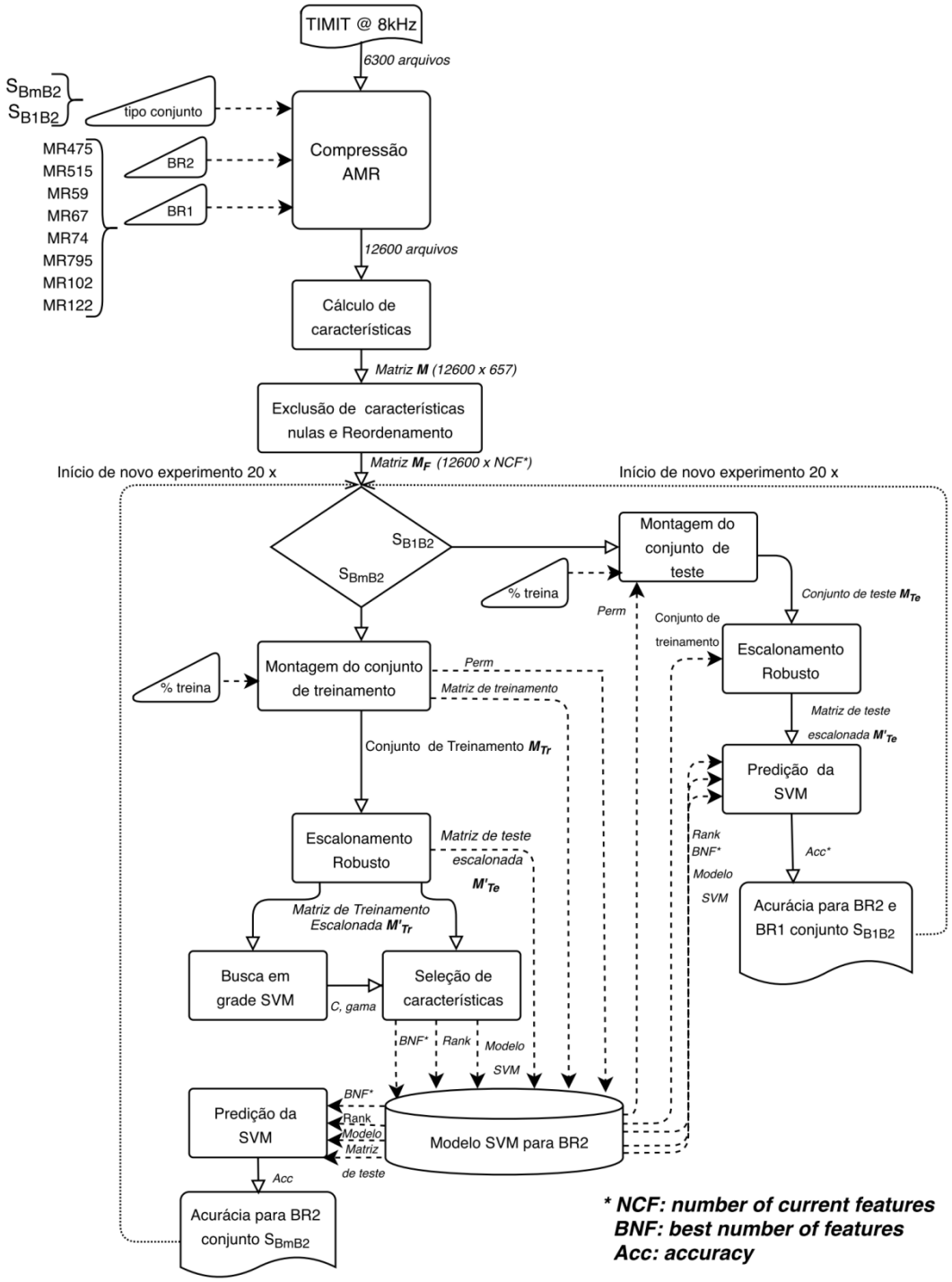


Figura 5-1 – Fluxo de dados detalhado para os experimentos.

5.3 - COMPRESSÃO AMR

O *corpus* TIMIT foi utilizado para gerar 6300 arquivos AMR com compressão simples e 6300 arquivos com compressão dupla, considerando que, para os conjuntos S_{BmB2}, BR1 assume todas as 8 taxas AMR nas seguintes frações de aproximadamente $N/8$, N=6300 para o *corpus* TIMIT (Shen *et. al*, 2012):

- 787 arquivos na taxa de 4,75 kbits/s
- 788 arquivos na taxa de 5,15 kbits/s

- 787 arquivos na taxa de 5,9 kbits/s
- 788 arquivos na taxa de 6,7 kbits/s
- 787 arquivos na taxa de 7,4 kbits/s
- 788 arquivos na taxa de 7,95kbits/s
- 787 arquivos na taxa de 10,2 kbits/s
- 788 arquivos na taxa de 12,2 kbits/s

Como exemplo, seja um arquivo do *corpus* TIMIT convertido para RIFF WAV com taxa de amostragem 8kHz e 16 bits por amostra. Esse arquivo é inicialmente lido e convertido para 13 bits (amostras divididas por 8 e truncados os 3 bits menos significativos) e convertido para AMR usando o codificador padrão na taxa BR2. Esse arquivo fará parte dos 6300 arquivos S-AMR. Para o conjunto comprimido S_{BmB2}, o arquivo RIFF WAV é codificado para AMR na taxa BR1 que varia conforme explicado anteriormente e, logo após, é descomprimido e comprimido novamente para BR2. Os 6300 arquivos resultantes terão compressão dupla. Já para o conjunto S_{B1B2}, BR1 é a mesma para todos os 6300 arquivos.

A Figura 5-2 mostra uma ilustração do algoritmo da geração de um conjunto comprimido S_{BmB2} em que BR2=4,75 kbits/s. O arquivo exemplo do *corpus* TIMIT (reamostrado a 8 kHz) é comprimido uma única vez para formar um arquivo S-AMR e esse mesmo arquivo é comprimido duas vezes para formar um arquivo D-AMR. Notar que BR1=12,2 kbits/s porque esse arquivo faz parte dos 788 últimos arquivos do *corpus*. Todos os conjuntos comprimidos de arquivos S-AMR e D-AMR são armazenados em disco para posterior utilização no cálculo das características.

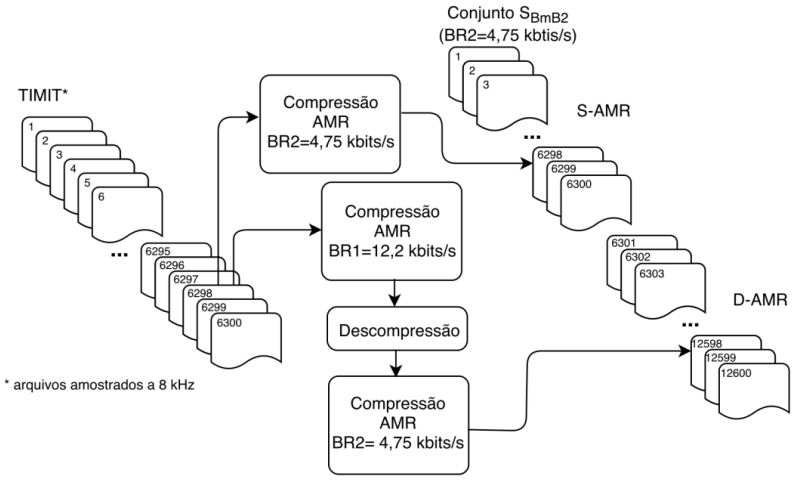


Figura 5-2 – Exemplo para o algoritmo de geração de um conjunto comprimido S_{BmB2}.

5.4 - CÁLCULO DE CARACTERÍSTICAS

O cálculo das características é um bloco essencial e muito importante do algoritmo

proposto. Ele transforma cada arquivo S-AMR ou D-AMR em um vetor de 657 características no domínio da compressão, independentemente do tamanho do arquivo AMR. Essas características, conforme já explicado, não são calculadas a partir das formas de onda dos arquivos AMR decodificados, e sim a partir de parâmetros do codificador. É possível dividir o cálculo das características em duas etapas: modificação do decodificador AMR e leitura dos parâmetros com montagem dos vetores.

5.4.1. Modificação do decodificador AMR

O código fonte do decodificador padrão AMR (3GPP-AMR Codec, 2017), escrito em linguagem C, realiza todas as operações matemáticas necessárias para transformar os parâmetros do fluxo de bits em parâmetros do codificador. Em outras palavras, se um arquivo AMR é a entrada, a saída do decodificador pode ser, mediante pequenas modificações do seu código fonte, uma sequência de parâmetros. A forma mais imediata de extrair os parâmetros do codificador da Tabela 4-2 é modificar o código fonte para que sejam gerados arquivos binários com tais parâmetros durante o processo de decodificação; essa operação é denominada nesta tese de *desempacotamento*.

Foram adotados alguns cuidados para que os valores extraídos no desempacotamento fossem condizentes com os parâmetros buscados. Os arquivos gerados com os parâmetros foram criados como binários, conforme o comando em C a seguir que cria, por exemplo, o arquivo LSP.bin para armazenar os parâmetros LP:

```
fopen("LPC.bin","wb");
```

Foi verificado que todas as variáveis do código fonte que correspondiam aos parâmetros do codificador eram do tipo *Word32* e definidas como *long*, isto é, inteiros de 32 bits com sinal. As variáveis foram armazenadas nos arquivos binários com comandos que escrevem os valores dos parâmetros no arquivo binário correspondente. Observa-se que o tamanho de cada arquivo binário é variável e depende do tamanho do arquivo AMR. A Tabela 5-2 resume as principais linhas de comando inseridas para a extração de cada variável dos parâmetros de interesse do codificador. Os ponteiros de arquivos foram definidos em outro trecho do código fonte.

Tabela 5-2 – Parâmetros do codificador extraídos dos arquivos AMR em que todas as variáveis são declaradas como inteiro de 32 bits.

Parâmetro	Variáveis	Linhas inseridas
Coeficientes LP	`A_t[]`	`fwrite(A_t, sizeof (Word32), AZ_SIZE, file_LPC)`
Pares espectrais de linha	`lsp[]`	`fwrite(lsp, sizeof (Word32), M, file_LSP)`
Período de *pitch*	`T0`	`fwrite(T0,sizeof(Word32),1,file_pitch)`
Ganho de *pitch*	`gain` `gain_pit`	`fwrite(&gain,sizeof(long),1,file_pitch_g)` `fwrite(gain_pit,sizeof(long),1,file_pitch_g)`
Energia do quadro	`currEnergy`	`fwrite(&currEnergy,sizeof(Word32),1,file_currenergy)`
Ganho do dicionário fixo	`gain_code` `gain_cod`	`fwrite(gain_code,sizeof(long),1,file_code_g)` `fwrite(gain_cod,sizeof(long),1,file_code_g)`
Patamar de ruído	`noiseFloor`	`fwrite(&noiseFloor,sizeof(Word32),1,file_noisefloor)`

Os 7 arquivos binários gerados têm valores dispostos na ordem *Little endian*. A Tabela 5-3 exemplifica quatro valores consecutivos de um arquivo binário visualizados num leitor hexadecimal (arquivo referente a um arquivo S-AMR do *corpus* TIMIT com BR2= 5,9 kbits/s). Os valores estão convertidos para decimal (faixa dinâmica possível entre −2.147.483.647 e +2.147.483.647) utilizando complemento a dois. Na linha da tabela correspondente aos coeficientes LP, o número em negrito 4096 indica um valor constante atribuído pelo codificador para o primeiro coeficiente LP do *array* `A_t[]`, ou seja, `A_t[0]` (o array tem tamanho 10 e tem 11 elementos). Tal valor está repetido a cada 10 coeficientes LP no arquivo binário gerado e caracteriza o elemento constante do denominador do filtro só de polos da Equação (3.1).

Tabela 5-3 – Quatro valores consecutivos dos arquivos binários extraídos referentes aos parâmetros de codificador AMR (arquivo S-AMR do *corpus* TIMIT com BR2= 5,9 kbits/s).

Parâmetros do arquivo binário	Valores hexadecimais				Valores decimais			
Coeficientes LP	**00 10 00 00**	DA F2 FF FF	60 12 00 00	A2 F1 FF FF	**4096**	-3366	4704	-3678
Pares espectrais de linha	D4 CB FF FF	EB B6 FF FF	A5 97 FF FF	CB 8A FF FF	-13356	-18709	-26715	-30005
Período de *pitch*	19 00 00 00	14 00 00 00	15 00 00 00	1D 00 00 00	25	20	21	29
Ganho de *pitch*	AE 17 00 00	70 2D 00 00	C2 05 00 00	B8 0E 00 00	6062	11632	1474	3768
Energia do quadro	2A 00 00 00	04 00 00 00	53 03 00 00	8A 03 00 00	42	4	851	906
Ganho do dicionário fixo	59 01 00 00	E7 00 00 00	05 01 00 00	8C 00 00 00	345	231	261	140
Patamar de ruído	30 01 00 00	C0 05 00 00	C0 05 00 00	C0 05 00 00	304	1472	1472	1472

5.4.2. Leitura de parâmetros e montagem de vetores

Após a geração dos arquivos binários, é necessária a leitura dos parâmetros e cálculo das características. Cada arquivo binário é lido com o comando MATLAB® `fread(fid,"*int32")`, em que `fid` é um identificador de arquivos fornecido pelo comando `fopen` e `*int32` é uma opção para a leitura dos dados em formato nativo de inteiro de 32 bits. Para os coeficientes LP e os parâmetros LSP, é realizada a leitura de cada parâmetro a_i e q_j de forma individual (para cada valor de i e j) e também a leitura de

todos os coeficientes *a* e *q* agregados para o cálculo das características. A Tabela 5-4 traz exemplos de 20 valores consecutivos lidos via MATLAB® para alguns parâmetros de um arquivo S-AMR codificados na taxa 12,2 kbits/s. Os valores em negrito indicam o valor constante atribuído pelo codificador para o primeiro coeficiente LP.

Tabela 5-4 – Vinte valores consecutivos dos arquivos binários extraídos referentes aos parâmetros de codificador AMR (arquivo S-AMR do *corpus* TIMIT com BR2= 12,2 kbits/s). Os valores em negrito indicam o valor constante atribuído pelo codificador para o primeiro coeficiente LP.

Parâmetros do arquivo binário	*a*	*q*	*T*	β	*E*	g_c	*NFL*
Valores dos parâmetros (lidos em MATLAB®)	**4096**	30506	74	18840	5082	260	304
	-1938	26908	77	6556	13173	232	304
	2932	21124	64	8192	13369	352	1472
	-1862	14338	67	13924	7283	210	1472
	2577	6391	109	6556	4704	376	1472
	-1634	-2386	112	15564	4240	246	1472
	1863	-10615	61	13924	5860	250	1472
	-1031	-18223	64	3276	14002	148	496
	1049	-24318	84	3276	17013	298	384
	-344	-28223	87	6556	20820	152	384
	262	31012	131	16384	23041	248	384
	4096	27817	126	19660	19060	126	384
	-126	21249	34	14744	9478	324	384
	336	13676	34	11468	1939	94	384
	-195	4783	21	11468	866	152	384
	484	-4771	27	9828	1279	78	384
	-255	-13230	110	6556	3125	158	384
	620	-21445	112	6556	7008	66	384
	-266	-27636	18	19660	5450	130	384

A partir dos valores lidos de cada arquivo S-AMR e D-AMR e exemplificados na Tabela 5-4, os vetores com 657 características são montados seguindo a ordem da Tabela 4-4 e usando as Equações (4.5) a (4.21), sendo gerado um vetor de características para cada arquivo AMR. Para as distribuições de probabilidades dos primeiros dígitos dos coeficientes LP e parâmetros LSP, foram usadas como estimativas os valores normalizados dos histogramas dos primeiros dígitos (comando MATLAB® *histogram* com a opção *normalization* definida como *pdf*). O cálculo do primeiro dígito de um número foi realizado pela divisão do número por uma potência de dez e considerando como resposta o menor número inteiro mais próximo desse resultado.

5.5 - EXCLUSÃO DE CARACTERÍSTICAS NULAS

Durante os experimentos realizados, foram verificadas empiricamente as propriedades numéricas dos parâmetros do codec AMR. Tais propriedades influenciam diretamente o comportamento das características, de forma que algumas apresentam desvio padrão nulo, quer seja por possuírem valores nulos ou por serem constantes para todos os arquivos AMR. Portanto, essas características não contêm informação útil para o algoritmo de detecção e são eliminadas para simplificar o treinamento e predição da SVM. A Tabela 5-5 resume as propriedades numéricas verificadas dos parâmetros do codificador AMR e suas consequências.

Tabela 5-5 – Propriedades numéricas de alguns parâmetros do codec AMR e suas consequências para as características calculadas.

Parâmetros do codificador AMR	Propriedades	Consequências
q_1	Ocorrem apenas os primeiros dígitos 2 e 3.	As distribuições de probabilidade dos demais primeiros dígitos são nulas.
q_2	Ocorrem apenas os primeiros dígitos 1, 2 e 3.	
q_3	Não ocorrem primeiros dígitos 4 e 5.	As distribuições de probabilidade desses primeiros dígitos são nulas.
q_9	Ocorrem apenas os primeiros dígitos 1, 2 e 3.	As distribuições de probabilidade dos demais primeiros dígitos são nulas.
q_{10}	Ocorrem apenas os primeiros dígitos 2 e 3.	
β	Os valores máximos (ou mínimos) do ganho de *pitch* são os mesmos em todos os arquivos AMR para algumas taxas de bits. O ganho de *pitch* assume valor zero em pelo menos um quadro de todos os arquivos AMR.	Valores máximos (ou mínimos) são constantes para todos os arquivos AMR. A média geométrica assume valor zero e à harmônica é atribuído valor zero.
E	Os valores zero de energia de quadro ocorrem no início e final dos arquivos, são os mais frequentes e ocorrem em todos os arquivos AMR.	A moda e o valor mínimo são sempre zero ($E \geq 0$), assim com as médias geométrica e harmônica a cujos valores são atribuídos zero.
g_c	O ganho de dicionário fixo assume valor zero em pelo menos um quadro de todos os arquivos AMR.	O valor mínimo e a média geométrica assumem valor zero e à harmônica é atribuído valor zero.
NFL	Os valores zero de patamar de ruído são os mais frequentes e ocorrem em todos os arquivos AMR.	A moda e o valor mínimo são sempre zero ($NFL \geq 0$), assim com as médias geométrica e harmônica a cujos valores são atribuídos zero.

Mediante as propriedades descritas na Tabela 5-5, as características eliminadas são detalhadas na Tabela 5-6, em que algumas delas são excluídas de todos os modos AMR, enquanto outras são excluídas apenas de alguns modos. Para todos os modos AMR são excluídas 38 características nulas, enquanto 7 características são excluídas somente de alguns modos.

Tabela 5-6 – Características eliminadas dos modos AMR (conjuntos comprimidos S_{BmB2} com *corpus* TIMIT). As características correspondem às combinações de linhas e colunas da tabela, como por exemplo *max(*β). "Todos" significa que a característica é eliminada de todos os modos AMR. Células vazias significam que a característica não é excluída. As exceções seguem os seguintes símbolos:

* excluída dos modos MR122 e MR795
$ excluída dos modos MR67, MR74 e MR122
\# excluída de todos os modos, exceto MR475
& excluída apenas no modo MR122

Medidas	Parâmetros do codificador AMR								
	q_1	q_2	q_3	q_9	q_{10}	β	*E*	g_c	*NFL*
Mo							todos		todos
$\tilde{x}$									todos
max						*			
min						#	todos	$	todos
$\bar{x}_{geom}$						*	todos	&	todos
$\bar{x}_{harm}$						*	todos	&	todos
MAD									todos
$m_x(1)$	todos				todos				
$m_x(4)$	todos	todos	todos	todos	todos				
$m_x(5)$	todos	todos	todos	todos	todos				
$m_x(6)$	todos	todos		todos	todos				
$m_x(7)$	todos	todos		todos	todos				
$m_x(8)$	todos	todos		todos	todos				
$m_x(9)$	todos	todos		todos	todos				

Após a contabilização das características nulas da Tabela 5-6, é possível determinar o número atual das características NCF para cada taxa AMR conforme listadas a seguir:

- 619 características na taxa de 4,75 kbits/s
- 618 características na taxa de 5,15 kbits/s
- 618 características na taxa de 5,9 kbits/s
- 617 características na taxa de 6,7 kbits/s
- 617 características na taxa de 7,4 kbits/s
- 615 características na taxa de 7,95kbits/s
- 618 características na taxa de 10,2 kbits/s
- 612 características na taxa de 12,2 kbits/s

5.6 - NÚMERO NECESSÁRIO DE EXPERIMENTOS

Um experimento é definido como um processo completo de treinamento e predição usando o mesmo conjunto de treinamento. Dessa forma, cada conjunto de treinamento dá origem a um experimento diferente cujas acurácias também são diferentes. A extração da matriz de treinamento $\boldsymbol{M_{Tr}}$ a partir da matriz de características $\boldsymbol{M_F}$ é realizada após o embaralhamento da metade superior de $\boldsymbol{M_F}$ e da metade inferior de $\boldsymbol{M_F}$ com a mesma permutação. Para o *corpus* TIMIT, das 6300 primeiras linhas de $\boldsymbol{M_F}$, são extraídas 4410

linhas e das 6300 últimas linhas de $\boldsymbol{M_F}$ mais 4410, ambas com a mesma permutação de tamanho 6300. Como das 6300 linhas de $\boldsymbol{M_F}$ são extraídas 4410 linhas, o número total de experimentos possíveis é a combinação de 6300 elementos de 4410 em 4410, um número que pode ser calculado por:

$$C_{np} = \frac{n!}{p!\,(n-p)!} = \frac{6300!}{4410!\,1890!} = 2{,}503690548.10^{1669} \rightarrow \infty \quad (5.1)$$

Mediante esse resultado, a definição do número de experimentos necessários para aferir o desempenho do método como uma fração do número de experimentos possíveis se torna inviável. Em termos heurísticos, quanto maior a quantidade de experimentos realizados, melhor a média das acurácias dos experimentos se aproxima do desempenho real do método. No entanto, realizar um elevado de número de experimentos pode ser desnecessário para estimar a acurácia do método. Haja vista o proposto em *Shen et. al* (2012) para os experimentos com SVM, neste trabalho foram realizados vinte experimentos para estimar a acurácia do método proposto com o *corpus* TIMIT. Dessa forma, acredita-se que um maior número de experimentos não modificaria de forma significativa a acurácia média do método.

5.7 - MONTAGEM DO CONJUNTO DE TREINAMENTO

Os conjuntos de treinamento e teste definem cada um dos experimentos realizados e são montados a partir das matrizes de características $\boldsymbol{M_F}$. Antes da extração desses conjuntos, para cada um dos 20 experimentos e em cada taxa de bits BR2, as 6300 primeiras linhas da matriz $\boldsymbol{M_F}$ (características dos arquivos S-AMR) e as 6300 últimas linhas (características dos arquivos D-AMR) são embaralhadas usando a mesma permutação aleatória. Em seguida, são extraídas matrizes de treinamento $\boldsymbol{M_{Tr}}$ com 70% dos arquivos do *corpus*, ou seja, essas matrizes são formadas pelas 4410 primeiras linhas da primeira metade da matriz $\boldsymbol{M_F}$ embaralhada (arquivos S-AMR) e 4410 primeiras linhas da segunda metade da matriz $\boldsymbol{M_F}$ embaralhada (arquivos D-AMR), totalizando matrizes de treinamento $\boldsymbol{M_{Tr}}$ com 8820 linhas. Por conseguinte, os conjuntos de teste são montados com as linhas restantes da matriz $\boldsymbol{M_F}$, formando a matriz $\boldsymbol{M_{Te}}$ com 3780 linhas (1890 linhas de características de arquivos S-AMR e 1890 linhas de características de arquivos D-AMR).

A Figura 5-3 descreve, por meio de uma versão simplificada, o procedimento de montagem dos conjuntos de treinamento e teste para dois experimentos. Para o primeiro experimento, a matriz simplificada $\boldsymbol{M_F}$ de 20 linhas é misturada pela permutação 1 (simples inversão da ordem das linhas, por exemplo) conforme descrito anteriormente, isto

é, por operações nas suas metades, e gera um conjunto de treinamento e teste 1 pela extração das matrizes simplificadas $\boldsymbol{M_{Tr}}$ e $\boldsymbol{M_{Te}}$. Para o segundo experimento, a matriz simplificada $\boldsymbol{M_F}$ é misturada por outra permutaçãoe novas matrizes são extraídas. Observa-se que na segunda permutação (simples separação entre linhas pares e ímpares, por exemplo), novos conjuntos de treinamento e teste 2 são extraídos. No método proposto, cada experimento usa uma permutação diferente de tamanho 6300 e novos conjuntos de treinamento são gerados para posterior escalonamento robusto.

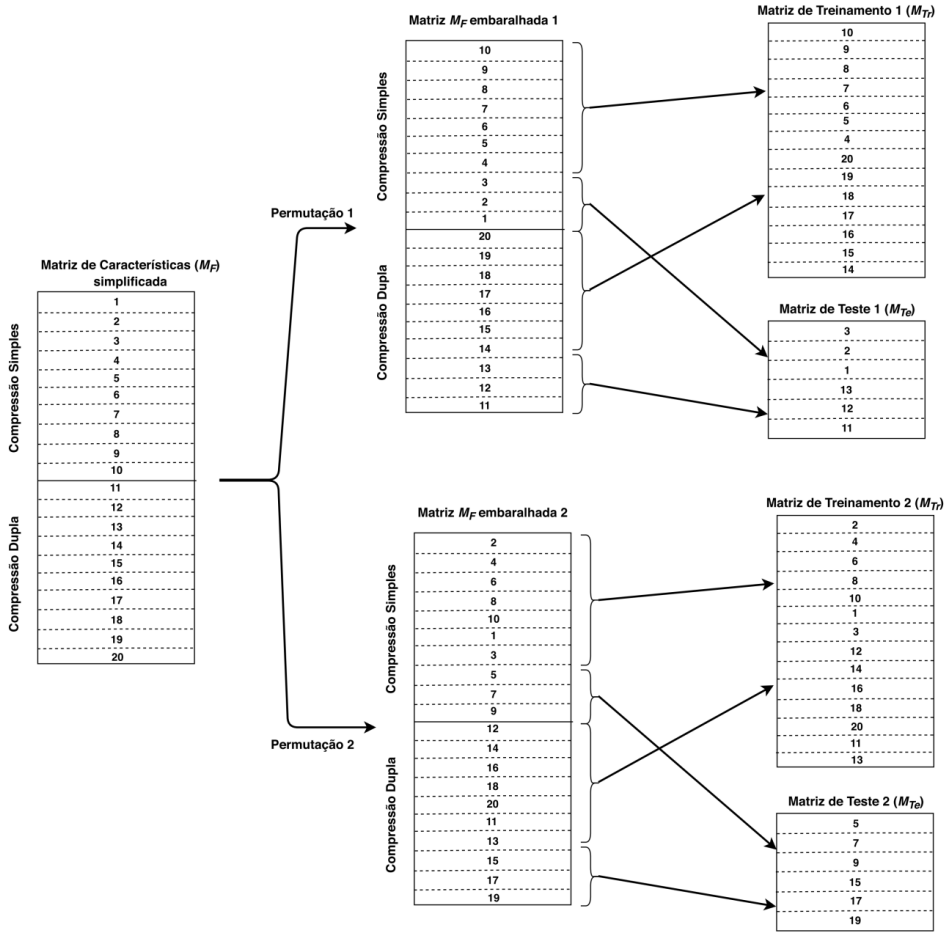


Figura 5-3 – Descrição simplificada do procedimento de montagem de conjuntos de treinamento e teste (dois experimentos).

Acompanhando a montagem das matrizes de treinamento $\boldsymbol{M_{Tr}}$, são gerados vetores de rótulos $\boldsymbol{y}$ com 4410 colunas com rótulos +1, representando os arquivos S-AMR, seguidas de 4410 colunas de rótulos -1, representando os arquivos D-AMR. Da mesma forma, são gerados os vetores de rótulos para as matrizes de teste $\boldsymbol{M_{Te}}$ com 1890 rótulos +1 seguidos de 1890 rótulos -1. Esses vetores de rótulos são usados em todos os experimentos, já que os vetores de características dos arquivos S-AMR e D-AMR não são misturados dentro das matrizes $\boldsymbol{M_{Tr}}$ e $\boldsymbol{M_{Te}}$ (a mistura para definição do experimento é feita antes da extração dos conjuntos de treinamento e teste).

5.8 - ESCALONAMENTO ROBUSTO

Conforme já mencionado na Seção 5.2, as características χ extraídas dos arquivos S-AMR e D-AMR apresentam considerável número de valores atípicos. A sua existência degrada o desempenho da SVM e o algoritmo de escalonamento deve contemplar a redução de tais valores (Cao *et. al*, 2016). Na Tabela 5-7 é apresentado um resumo da situação dos valores atípicos ao longo das características não nulas dos conjuntos S_{BmB2} nas diferentes taxas AMR (12600 valores por característica). As características atuais (NCF) foram analisadas e os valores atípicos foram calculados pelo método dos quartis, uma vez que as distribuições de probabilidade estimadas pelos histogramas das características não apontam para uma distribuição normal para todas elas. Nesse método, um valor é considerado atípico se sua distância acima do maior quartil ou abaixo do menor quartil for de pelo menos 1,5 vezes a distância interquartil. Como cada característica apresenta uma quantidade diferente de valores atípicos, o percentual dessa quantidade é calculado e quatro faixas são consideradas para melhor visualizar esses percentuais.

Tabela 5-7 – Quantidades de valores atípicos nas características extraídas dos arquivos S-AMR e D-AMR (método dos quartis).

Quantidade de valores atípicos (faixas)	Quantidade de características na faixa							
	Taxas AMR (em kbits/s)							
	4,75	5,15	5,90	6,70	7,40	7,95	10,2	12,2
menos de 1%	191	196	182	179	179	180	183	178
entre 1% e 5%	339	338	354	355	355	353	351	351
entre 5% e 10%	45	40	33	33	33	33	34	31
mais de 10%	44	44	49	50	50	49	50	52

A análise indica que todas as características têm valores atípicos em todas as taxas AMR e que a maioria delas tem entre 1% e 5% de valores atípicos, havendo, ainda, pelo menos 44 características que têm mais de 10% de valores atípicos. Esses resultados demonstram a necessidade de um algoritmo de escalonamento robusto para minimizar a influência dos valores atípicos no desempenho da SVM.

O escalonamento robusto é feito nos conjuntos de treinamento antes do seu uso na SVM de tal forma que as características passam a ter valores entre 0 e 1. O algoritmo logístico generalizado (GL) traz os valores atípicos para próximo de 0 ou 1 por meio da função de transferência logística não linear, diminuindo a influência desses valores no desempenho da SVM. Tal algoritmo também expande valores significativos e concentrados em uma faixa, geralmente obliterados pelos valores atípicos, balanceando as quantidades antes do treinamento e teste da SVM. Um exemplo de escalonamento robusto de uma característica de um conjunto de treinamento pode ser observado na Figura 5-4,

que é uma composição de boxplots e de um gráfico de espalhamento em que a linha tracejada mostra um exemplo de escalonamento de valor atípico. A característica usada é a moda dos coeficientes LP agregados para a taxa BR2= 7,4 kbits/s num conjunto S_{BmB2}. A distribuição original da característica é mostrada no Boxplot A e após o escalonamento robusto no Boxplot B, em que os valores escalonados não têm valores atípicos (os boxplots estão na mesma escala no gráfico de espalhamento). Nessa figura, as linhas tracejadas indicam que um valor atípico foi mapeado para um valor próximo de um. Os boxplots dessa figura também esclarecem que os dados centrais no Boxplot A, inicialmente concentrados próximos de zero, são mapeados para uma faixa maior no Boxplot B (entre 0,3 e 0,7). Observa-se que a informação levada pelos valores atípicos é diminuída e a do restante dos dados é enfatizada.

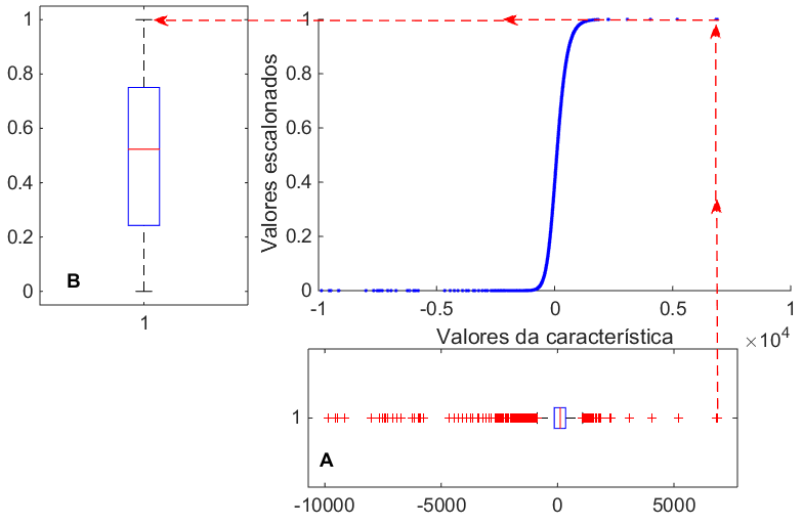


Figura 5-4 – Composição de boxplots e gráfico de espalhamento que mostra um exemplo de escalonamento robusto GL.

Uma vez escalonado o conjunto de treinamento, o conjunto de teste deve ser escalonado com os mesmos parâmetros, caso contrário o desempenho da SVM diminui. O algoritmo de escalonamento robusto GL gera esses parâmetros, os quais são usados no escalonamento do conjunto de teste antes da predição da SVM.

5.9 - LibSVM E BUSCA EM GRADE

Em todos os experimentos deste trabalho, foi utilizado o algoritmo descrito em Chang e Lin(2011), cujo pacote de rotinas é chamado LibSVM e pode ser baixado em LibSVM (2018). Embora tal pacote ofereça rotinas de escalonamento, foi usado o escalonamento robusto GL para treinamento e teste, como já mencionado. Também foi usado o núcleo RBF para a SVM, o qual permite o ajuste de dois parâmetros, C e γ, cuja escolha é implementada por meio da busca em grade, realizada para cada conjunto de

treinamento e considerando as acurácias de validação cruzada de 5 dobras.

Para cada experimento, o procedimento de busca em grade é usado para achar os melhores parâmetros C e γ para um dado conjunto de treinamento por meio da máxima acurácia de validação cruzada. Inicialmente é formada uma grade 12×12 com valores de C e γ em potências de 2 e feita uma busca frouxa pela máxima acurácia de validação cruzada (Chang e Lin, 2011), totalizando 144 acurácias calculadas. Logo em seguida, é elaborada uma nova grade 12×12 na vizinhança dos valores de C e γ achados na busca frouxa e, então, realizada uma busca refinada. Por último, uma nova grade 12×12 e uma segunda busca refinada é realizada. Finalmente, para cada taxa BR2, o modelo da SVM é determinado como aquele que ofereceu as melhores acurácias de validação cruzada e, posteriormente, é usado para o cálculo do melhor número de características BNF.

A Figura 5-5 ilustra uma busca frouxa em grade, seguida de uma busca refinada, em que os valores iniciais da busca frouxa são os utilizados no algoritmo. Notar que uma vizinhança é formada pelos valores adjacentes ao valor da grade que proporciona a máxima acurácia de validação cruzada (ponto vermelho na grade da busca frouxa). Nessa figura, os valores de máxima acurácia são $C=2^5$ e $\gamma=2^{-3}$ e a vizinhança para a busca refinada é $[2^3,2^7]$ para C e $[2^{-1}, 2^{-5}]$ para γ (em vermelho na figura). Essas vizinhanças são divididas em 11 intervalos iguais (Δ) para a busca refinada. Por exemplo, a vizinhança de C para a busca refinada tem $\Delta_C= (2^7-2^3)/11$ e os novos valores de grade são definidos pelos intervalos de mesmo comprimento Δ_C ($2^3+ \Delta_C$, $2^3+ 2\Delta_C$, $2^3+ 3\Delta_C$,...,2^7). No método proposto, é realizada uma busca frouxa e duas buscas refinadas, em que a segunda busca refinada é implementada da mesma forma a partir das vizinhanças da primeira busca refinada. Os valores finais para C e γ entregues pelo método são os valores que proporcionam a máxima acurácia de validação cruzada após a segunda busca refinada.

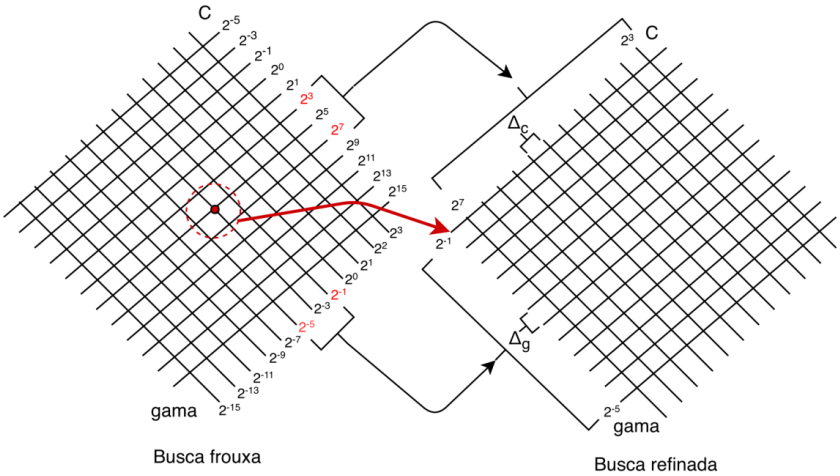


Figura 5-5 – Diagrama explicativo para uma busca em grade frouxa e uma refinada.

5.10 - SELEÇÃO DE CARACTERÍSTICAS

Além da eliminação recursiva de características, o algoritmo SVM-RFE CBR conta com a redução de polarização por correlação para evitar a eliminação desnecessária de características correlacionadas. Seu uso se justifica se as características apresentarem correlação significativa, evitando, assim, a polarização do algoritmo de eliminação de características. A análise de correlação entre as características dos arquivos AMR pode ser realizada pelo cálculo dos coeficientes de correlação entre todas elas, de forma a estimar a pertinência do uso do algoritmo CBR. Esse cálculo dos coeficientes fornece resultados importantes como, por exemplo, se existem características iguais entre si ou altamente correlacionadas num certo grau de estimativa.

O coeficiente de correlação *r* entre as características *i* e *j* para a análise realizada é definido como:

$$r_{i,j} = \frac{\sum_{k=1}^{2N}(\chi'_{i,k} - \overline{\chi'_i})(\chi'_{j,k} - \overline{\chi'_j})}{\sqrt{\sum_{k=1}^{2N}\left(\chi'_{i,k} - \overline{\chi'_i}\right)^2}\sqrt{\sum_{k=1}^{2N}\left(\chi'_{j,k} - \overline{\chi'_j}\right)^2}}, \tag{5.2}$$

em que:

N é o tamanho do *corpus* (6300 para o *corpus* TIMIT);

i,*j* assumem valores entre 1 e NCF (número atual das características); e

$\chi'_{i,k}$ é o *k*-ésimo valor da *i*-ésima característica escalonada.

A análise de correlação foi realizada sobre as matrizes de características não nulas dos conjuntos comprimidos. Foram encontradas 7 características idênticas ($r = 1$), conforme a Tabela 5-8, cujos motivos também estão descritos na tabela. Por serem idênticas, o algoritmo SVM-RFE CBR automaticamente realiza o tratamento delas para garantir resultados satisfatórios.

Tabela 5-8 – Características idênticas ($r = 1$) identificadas nos conjuntos comprimidos (os números se referem à Tabela 4-4).

Número	Característica	Número	Característica	Motivo
320	$max(q_1)$	294	$max(q)$	$q_1 = max(q_1, q_{2,\ldots}, q_{10})$
555	$min(q_{10})$	295	$min(q)$	$q_{10} = min(q_1, q_{2,\ldots}, q_{10})$
325	$meanabs(q_1)$	313	$mean(q_1)$	$q_1 \geq 0$
585	$meanabs(T)$	573	$mean(T)$	$T > 0$
619	$meanabs(E)$	607	$mean(E)$	$E > 0$
636	$meanabs(g_c)$	624	$mean(g_c)$	$g_c > 0$
641	$meanabs(NFL)$	653	$mean(NFL)$	NFL ≥ 0

Já a Tabela 5-9 informa a quantidade de correlações entre características não nulas para cada faixa de valores de *r*. Esse resultado indica que as características podem ser

consideradas correlacionadas em grau tal que o uso do algoritmo CBR se justifica.

Tabela 5-9 – Quantidade de correlações entre características não nulas para faixas de valores de *r* nos conjuntos comprimidos.

Faixas de valores de *r*	Quantidade de correlações na faixa
$0,9 \leq \|r\| \leq 0,9999$	607
$0,85 \leq \|r\| \leq 0,9999$	981
$0,8 \leq \|r\| \leq 0,9999$	1646
$0,7 \leq \|r\| \leq 0,9999$	3657

A saída do algoritmo SVM-RFE CBR é um vetor F_{rank} com a ordem das características, ou seja, os índices ordenados das características conforme estão dispostas no conjunto de treinamento original. Uma vez ordenadas, as características do conjunto de treinamento formam grupos para calcular as acurácias de validação cruzada usando os parâmetros C e γ da busca em grade anteriormente realizada, obtendo NCF (número atual das características) acurácias (existem NCF grupos possíveis). A máxima acurácia corresponde a um grupo de características cuja quantidade é o melhor número de características BNF, conforme citado na Seção 4.2.

À medida que o número de características do grupo aumenta, a acurácia de validação cruzada varia, chegando a um máximo. Esse resultado é demonstrado na Figura 5-6, que é um gráfico das acurácias de validação cruzada pelo número de características do grupo, para um experimento e para quatro taxas AMR. De acordo com as indicações dessa figura, o ponto de melhor número de características (BNF) para o modo MR475 é 22, que é menor do que 619 (número atual das características). Esse comportamento também é observado em todos os modos AMR, como MR67, MR74 e MR515 com os valores de BNF de 209, 172 e 68 características, respectivamente. A acurácia de validação cruzada diminui com o aumento do número de características após o ponto de BNF em todos os modos AMR.

Na Tabela 5-10 é possível observar os valores mínimos e máximos do melhor número de características BNF para cada taxa de bits AMR nos 20 experimentos, restando evidente que os valores se modificam desde poucas características até valores mais elevados, dependendo de BR2 e do conjunto de treinamento. Esses valores confirmam a estratégia adotada pelo método de inicialmente calcular um maior número de características para usar algumas delas na obtenção de acurácias maiores. Um subconjunto de todas as 657 características pode, portanto, ser capaz de atingir acurácias maiores em todas as taxas de bits e experimentos do que o conjunto de todas as características.

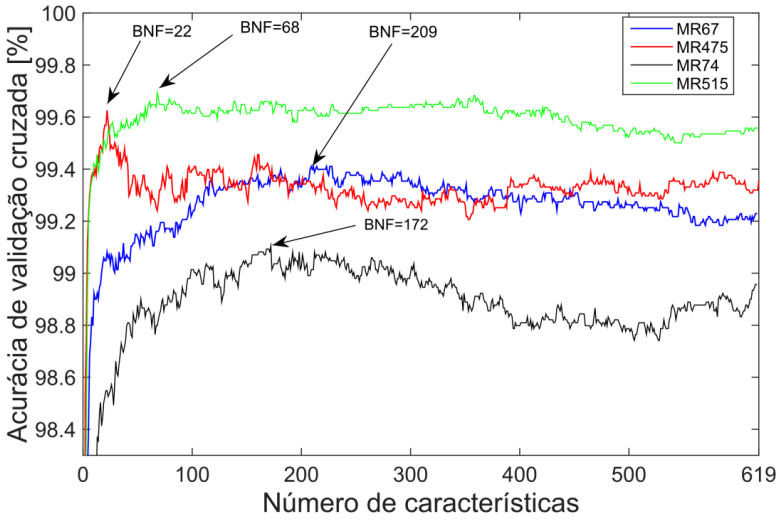


Figura 5-6 – Variação da acurácia de validação cruzada quando o número de características ordenadas aumenta para um experimento em um conjunto S_{BmB2} e quatro taxas AMR.

Para todos os experimentos e para todas as taxas de bits AMR, um ordenamento das características é realizado pelo algoritmo SVM-RFE CBR para buscar a máxima acurácia de validação cruzada. Considerando todos esses ordenamentos, é importante determinar se algumas características e quais delas são ordenadas nas primeiras posições, ou seja, quais são as características mais importantes para a maioria dos ordenamentos realizados. Uma forma de determinar essas características é aferir com qual frequência elas ocorrem nas primeiras posições de ordenamento (por exemplo, as 10 primeiras). Um resultado pode ser visto na Figura 5-7, que é um gráfico de barras empilhadas das 10 características mais importantes ordenadas pelo método SVM-RFE CBR, calculadas sobre 160 ordenamentos (20 experimentos, 8 taxas AMR cada) com conjuntos S_{BmB2}. As 10 características mais importantes estão ordenadas no eixo horizontal e o eixo vertical corresponde à taxa de ocorrência das características (em %). Assim, a soma das barras verticais empilhadas em uma ordem é sempre 100%. Cada barra colorida tem uma altura que corresponde à taxa de ocorrência da característica assinalada (números com 3 dígitos nas barras) sobre os 160 ordenamentos.

Tabela 5-10 – Valores mínimos e máximos do melhor número de características BNF nos 20 experimentos em cada taxa de bits AMR.

BNF	Taxa AMR (kbits/s)							
	4,75	5,15	5,90	6,70	7,40	7,95	10,2	12,2
Mínimo	22	54	90	48	96	107	110	93
Máximo	430	559	475	508	435	514	612	452

Por exemplo, a característica 390≡$m_{q_3}(9)$ aparece como a segunda característica

mais importante em 75% dos ordenamentos. A Figura 5-7 também mostra, por exemplo, a primeira e a segunda características mais importantes como as seguintes distribuições de probabilidade de primeiros dígitos dos coeficientes LSP: 514≡$m_{q_8}(3)$ com cerca de 75%, 515≡$m_{q_8}(4)$ com cerca de 25%, 390≡$m_{q_3}(9)$ com cerca de 74%, e 488≡$m_{q_7}(3)$ com cerca de 26%, respectivamente. As seguintes características são encontradas como a terceira mais importante: 488≡$m_{q_7}(3)$, 514≡$m_{q_8}(3)$, 131≡$\overline{a_5}$, 410≡$m_{q_4}(3)$, 489≡$m_{q_7}(4)$, 502≡ $max(q_8)$ e 558≡$\overline{q_{10}}_{harm}$. Conforme se observa, os coeficientes LSP formam grande parte das três primeiras melhores características dos ordenamentos, principalmente as estatísticas relativas às distribuições de probabilidades de primeiros dígitos. A correspondência entre os números encontrados na Figura 5-7 e as demais características pode ser verificada na Tabela 4-4.

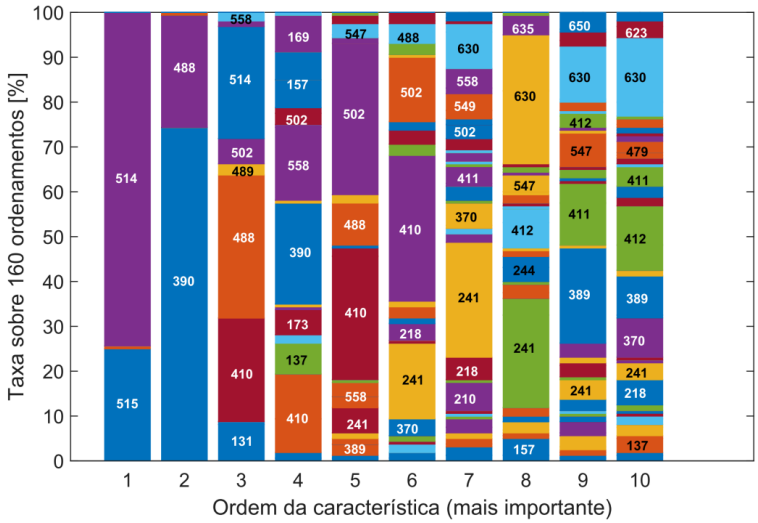


Figura 5-7 – Gráfico de barras empilhadas das 10 características mais importantes ordenadas sobre 20 experimentos com 8 taxas AMR com conjuntos S_{BmB2}.

5.11 - ANÁLISE DE DESEMPENHO

Para cada um dos 20 experimentos e para cada taxa AMR com o *corpus* TIMIT, são determinados os modelos SVM, a ordem das características e os respectivos melhores números de características BNF. Tais parâmetros são usados para a montagem dos conjuntos de teste, escalonamento desses conjuntos e predição da SVM para os conjuntos comprimidos S_{BmB2} e S_{B1B2}. Os conjuntos de teste são formados por vetores diferentes daqueles dos respectivos conjuntos de treinamento e com o mesmo número de vetores correspondentes a arquivos S-AMR e D-AMR. Como os rótulos são conhecidos (primeira metade do conjunto de teste com rótulo 1 e segunda metade com rótulo -1), a análise de desempenho neste trabalho consiste em comparar os rótulos preditos pela SVM e os

rótulos conhecidos, de modo que o desempenho será tão maior quanto mais rótulos forem coincidentes (acertos).

Dentre as maneiras de associar o número de rótulos coincidentes a uma acurácia, a análise de desempenho da SVM escolhida é baseada no cálculo da acurácia definido como (Luo *et al.*, 2017):

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \quad , \tag{5.3}$$

onde, considerando um conjunto de teste:

TP : é a soma de todos os casos de verdadeiro positivo, ou seja, em que o rótulo do conjunto de teste é -1 (compressão dupla) e o rótulo predito pela SVM é -1 (compressão dupla).

TN: é a soma de todos os casos de verdadeiro negativo, ou seja, em que o rótulo do conjunto de teste é 1 (compressão simples) e o rótulo predito pela SVM é 1 (compressão simples).

FP: é a soma de todos os casos de falso positivo, ou seja, em que o rótulo do conjunto de teste é 1 (compressão simples) e o rótulo predito pela SVM é -1 (compressão dupla).

FN: é a soma de todos os casos de falso negativo, ou seja, em que o rótulo do conjunto de teste é -1 (compressão dupla) e o rótulo predito pela SVM é 1 (compressão simples).

Pela análise dessa definição, se *FP* e *FN* forem nulas, a acurácia valerá 1 (100% de acerto) e se *FP* e *FN* valem 1, a acurácia será nula. A relação entre *TP* e *FN* é tal que *TP* + *FN* = $N_{TE}/2$, ou seja, os vetores com rótulos -1 (que são a metade dos vetores do conjunto de teste) podem ter predição 1 ou -1. A mesma relação se verifica para *TN* e *FP*, de modo que *TP*+*FN*+*TN*+*FP* = N_{TE}. Como exemplo de cálculo das somas anteriores e acurácia, seja um conjunto de teste hipotético em que foram obtidos os seguintes rótulos de predição: para a metade S-AMR do conjunto de teste (1890 vetores com rótulo 1), foram preditos 750 rótulos 1 e 1140 rótulos -1; logo, *TN* vale 750 e *FP* vale 1140: para a metade D-AMR (1890 vetores com rótulo -1), foram preditos 950 rótulos 1 e 940 rótulos -1; logo, *FN*= 950 e *TP*=940. A acurácia final, portanto, será (940 + 750) / (940 + 950 + 750 + 1140) = 1690 / 3780 = 0,4471 (44,71%).

Outras medidas de interesse para o problema da detecção da compressão dupla

AMR são a taxa de detecção de compressão dupla, chamada neste trabalho de $\overline{\text{TP}}$, e a taxa de detecção de compressão simples, chamada de $\overline{\text{TN}}$, ou seja, as taxas de acerto considerando apenas os conjuntos de arquivos com compressão dupla e simples, respectivamente (Luo *et al.*, 2017). Como metade do conjunto de teste é formada por eventos com compressão simples e a outra formada por eventos com compressão dupla, $\overline{\text{TP}}$ deve ser calculada como o número de acertos dos arquivos de compressão dupla (TP) em relação a todos os eventos com rótulo -1, ou seja, em relação à TP + FN, de modo a expressar a capacidade do algoritmo de acertar quais eventos são de compressão dupla: portanto, se FN=0, $\overline{\text{TP}}$=1. O mesmo raciocínio pode ser usado para conceber $\overline{\text{TN}}$, de modo que neste trabalho as taxas $\overline{\text{TP}}$ e $\overline{\text{TN}}$ são definidas como:

$$\overline{\text{TP}} = \frac{\text{TP}}{\text{TP} + \text{FN}} \text{ e} \tag{5.4}$$

$$\overline{\text{TN}} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{5.5}$$

6 - RESULTADOS COM *CORPUS* TIMIT

Os experimentos descritos no Capítulo 5 geraram resultados que são apresentados neste capítulo. A forma como os experimentos foram implementados para os conjuntos S_{BmB2} e o uso do mesmo *corpus* TIMIT permite uma comparação direta com os trabalhos publicados sobre o mesmo tema, conforme será apresentado no Capítulo 8.

6.1 - CONJUNTOS S_{BmB2}

Foram realizados 20 experimentos para cada taxa AMR com os conjuntos S_{BmB2}. É possível concluir que o método proposto é efetivo na detecção da compressão dupla AMR, pois a acurácia média ao longo das taxas de bits AMR foi de 99,16%. Já a taxa de detecção de compressão dupla $\overline{TP}$ média ao longo das taxas de bits AMR é de 98,88%, enquanto a taxa de detecção de compressão simples $\overline{TN}$ é de 99,44%. Ao longo de todos os experimentos, a acurácia mínima foi de 98,12% para BR2=7,4kbits/s e a máxima foi de 99,79% para BR2=4,75kbits/s. Todos os valores médios, mínimos e máximos das acurácias calculadas podem ser vistos na Tabela D-1 do Apêndice D.

6.2 - CONJUNTOS S_{B1B2}

Foram realizados 20 experimentos por taxa AMR com os conjuntos S_{B1B2}, todos baseados nos modelos SVM calculados para os conjuntos S_{BmB2}, adotando o procedimento de misturar as matrizes de características usando as mesmas permutações correspondentes aos experimentos com os conjuntos S_{BmB2}. Dessa forma, para cada experimento, as extrações dos conjuntos de treinamento e teste foram feitas da mesma forma que na geração dos modelos. O objetivo é analisar o desempenho do modelo quando BR1 é constante ao longo do conjunto de teste.

Os valores médios, mínimos e máximos das 64 acurácias calculadas ao longo dos 20 experimentos (1280 acurácias calculadas) estão dispostos na Tabela D-2 do Apêndice D. A acurácia média ao longo das taxas de bits AMR vale 99,17% (média de todas as colunas de médias), muito próxima da acurácia média dos conjuntos S_{BmB2} (99,16%). Já a taxa de detecção de compressão dupla $\overline{TP}$ média ao longo das taxas de bits AMR é de 99,08%, enquanto a taxa de detecção de compressão simples $\overline{TN}$ é de 99,26%. Ao longo de todos os experimentos, a acurácia mínima foi de 91,59% para BR1=6,7kbits/s e BR2=12,2kbits/s e a máxima foi de 99,95% para BR1=4,75kbits/s e BR2=4,75kbits/s. Dessa forma, é possível concluir que o método proposto também é efetivo na detecção da compressão dupla AMR quando BR1 é constante ao longo do conjunto de teste e que os modelos para BR1 variável também valem para BR1 fixa.

Pela análise dos resultados é possível também constatar uma tendência de aumento das acurácias na operação de transcodificação para cima (BR1<BR2), ou seja, as acurácias próximas à extremidade inferior esquerda da tabela são maiores que as demais. Analogamente, existe uma tendência de diminuição das acurácias na operação de transcodificação para baixo (BR1>BR2). O comportamento das acurácias é compatível com os resultados de Shen *et. al* (2012), ou seja, a relação das taxas de bits BR1 e BR2 influencia os resultados do método, algo esperado devido à perda de informação nas operações de transcodificação para baixo.

6.3 - VISUALIZAÇÃO DE CARACTERÍSTICAS

Embora os resultados demonstrem a efetividade do método proposto em termos de taxas de detecção com o *corpus* TIMIT, é necessário, ainda, utilizar formas de visualizar as estatísticas das características propostas, comparando aquelas correspondentes aos arquivos S-AMR com aquelas dos arquivos D-AMR. Pelas altas acurácias alcançadas, é esperado que existam diferenças significativas entre as características dos dois tipos de arquivos AMR. Duas formas de visualizar tais estatísticas são aqui propostas: por meio de histogramas e por meio de redução de dimensionalidade.

As estatísticas das características podem ser inspecionadas por meio de histogramas para ilustrar a efetividade do método proposto. A Figura 6-1 e a Figura 6-2 são uma composição de histogramas das características escalonadas, dispostas em barras verticais, lado a lado. O eixo horizontal indica a ordem da característica e o eixo vertical representa os valores das características escalonadas entre 0 e 1. Os valores dos histogramas são representados por uma escala de cor. Para computar essa figura, um experimento com o *corpus* TIMIT foi escolhido usando um conjunto comprimido S_{BmB2} com BR2=4,75 kbits/s. O conjunto de treinamento com características escalonadas, e ordenado conforme o algoritmo SVM RFE-CBR, é acessado apenas nas BNF (melhor número de características) primeiras características (no caso, BNF=81 características). O conjunto de treinamento extraído com as 81 características é dividido em duas matrizes: características dos arquivos S-AMR (primeiras 4410 linhas) e D-AMR (últimas 4410 linhas). A Figura 6-1 é computada a partir da matriz correspondente às características dos arquivos S-AMR e a Figura 6-2 da matriz correspondente aos arquivos D-AMR. Para cada matriz, um histograma de 40 intervalos é calculado para cada característica e os valores dos histogramas são representados usando uma escala de cor. No final, cada histograma é representado por uma barra colorida vertical, totalizando 81 barras, cujas cores variam de

valores baixos (azul escuro) a altos (vermelho escuro), de acordo com o mapa de cores à direita.

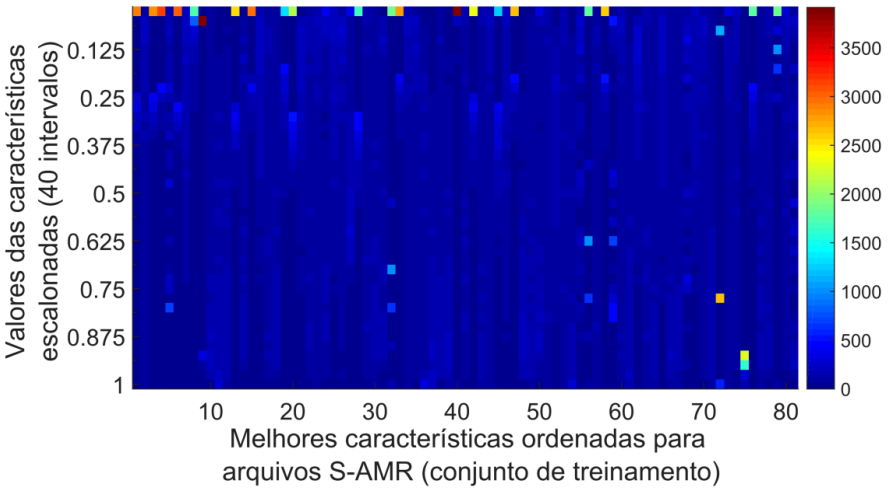


Figura 6-1 – Composição de histogramas de características escalonadas extraídas de um experimento com *corpus* TIMIT, na taxa de 4,75 kbits/s, para arquivos S-AMR.

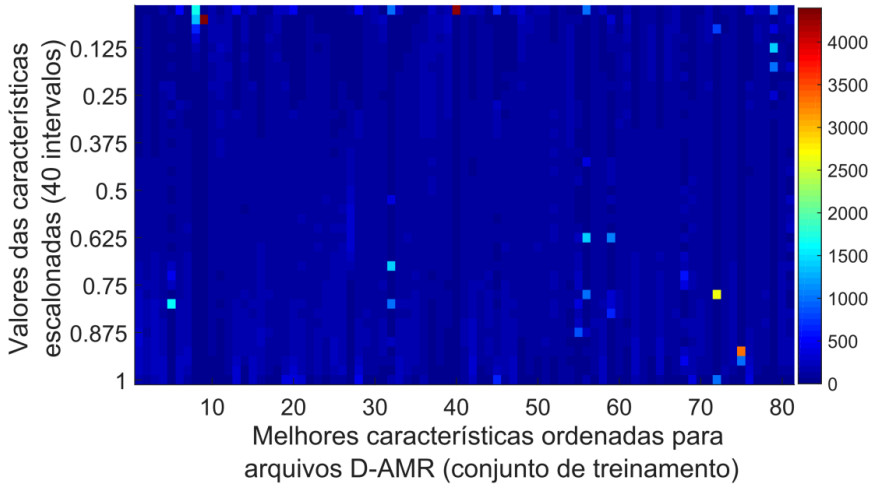
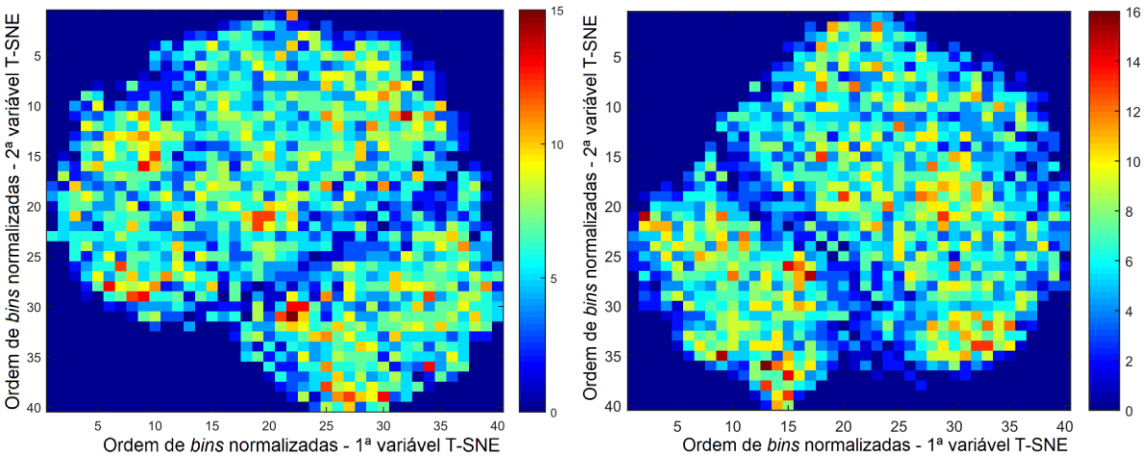


Figura 6-2 – Composição de histogramas de características escalonadas extraídas de um experimento com *corpus* TIMIT, na taxa de 4,75 kbits/s, para arquivos D-AMR.

É possível observar que a compressão dupla modifica a maioria das distribuições das características dos arquivos AMR da seguinte forma: ela espalha a distribuição de algumas características, pois a quantidade de pontos coloridos presentes na Figura 6-1 diminui na Figura 6-2; e ela suaviza a distribuição geral das características, provocando um aspecto mais uniforme visto na Figura 6-2.

Outra forma de visualizar as diferenças entre as características dos arquivos S-AMR e D-AMR é baseada na capacidade de redução de dimensionalidade do algoritmo t-SNE (Maaten e Hinton, 2008). Esse algoritmo recebe as características escalonadas como entrada e calcula, por meio de aprendizado não supervisionado, uma versão delas com dimensão reduzida. A Figura 6-3 é formada por histogramas bidimensionais representando

as características dos arquivos S-AMR (a) e D-AMR (b) (após a exclusão de características nulas) reduzidas para a dimensão 2 pelo algoritmo t-SNE (conjuntos S_{BmB2} na taxa de 4,75 kbits/s com número atual das características NCF=619). Por exemplo, um vetor de tamanho 619 que representa um arquivo AMR é reduzido a um vetor de tamanho 2 e o respectivo ponto é atribuído a um dos 40 × 40 intervalos do histograma bidimensional. Para construir a Figura 6-3, os 6300 vetores de tamanho 2 dos arquivos S-AMR e D-AMR são atribuídos aos intervalos e o número de pontos em cada intervalo define a intensidade da escala de cores à direita. As diferenças de formato, espaços ocupados e orientação entre as figuras mostram que o método proposto é capaz de produzir características discriminantes para detectar a compressão dupla AMR.



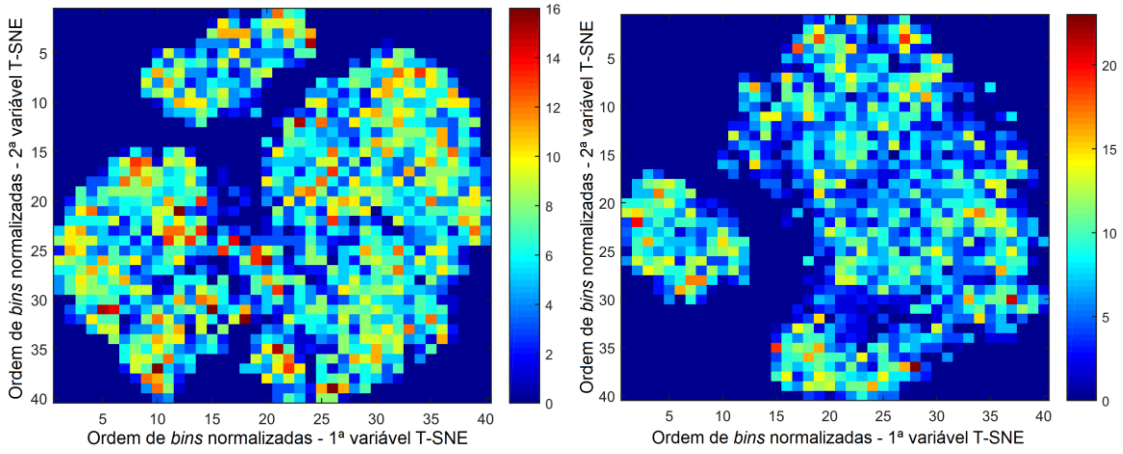
(a) (b)

Figura 6-3 – Histogramas bidimensionais para representação das características (após exclusão das nulas) usando o algoritmo de redução de dimensionalidade t-SNE para 6300 arquivos S-AMR (a) e D-AMR (b).

A comparação entre a Figura 6-3 (a) e (b) foi realizada com todas as NCF características atuais do conjunto comprimido (após a exclusão das nulas). Entretanto, se apenas o melhor número de características BNF for utilizado para a redução de dimensionalidade, uma melhor visualização das diferenças entre arquivos S-AMR e D-AMR poderia ser conseguida. Em outras palavras, os histogramas bidimensionais podem apresentar diferenças mais evidentes entre os dois tipos de arquivos AMR, haja vista as altas acurácias alcançadas.

Como cada experimento e cada taxa AMR tem um melhor número de características BNF, foi tomado como exemplo um experimento na taxa de 4,75 kbits/s para o uso do algoritmo t-SNE em que BNF=22, ou seja, os vetores de tamanho 22 que representam os arquivos AMR são convertidos para vetores com dimensão 2 e posterior visualização. A Figura 6-4 é formada por histogramas bidimensionais elaborados da

mesma forma que a Figura 6-3 e que representam as características dos arquivos S-AMR (a) e D-AMR (b) reduzidas para a dimensão 2 pelo algoritmo t-SNE (conjuntos S_{BmB2} na taxa de 4,75 kbits/s com BNF=22). Conforme previsto, os histogramas da Figura 6-4 evidenciam uma maior diferença estatística entre os arquivos S-AMR e D-AMR, ressaltada pelo uso apenas do melhor número de características BNF com o algoritmo t-SNE. Esse resultado confirma, portanto, a capacidade de discriminação do método proposto.



(a) (b)

Figura 6-4 – Histogramas bidimensionais para representação das 22 melhores características usando o algoritmo de redução de dimensionalidade t-SNE para 6300 arquivos S-AMR (a) e D-AMR (b).

7 - ANÁLISE DE ROBUSTEZ

Apesar dos resultados satisfatórios citados no Capítulo 6, o método proposto foi experimentado apenas com o *corpus* TIMIT com as durações e níveis de ruído originais. Em condições reais, é esperado que o conteúdo de áudio de interesse apresente algumas condições adversas, como contaminação por ruído e duração variável, além de alguns procedimentos específicos de adulteração que visam a evitar a detecção da compressão dupla. Nesta análise de robustez, é seguida a metodologia proposta por Luo *et. al* (2017) para computar o desempenho do método para arquivos de duração variável, ataque de deslocamento de quadro, áudio contaminado por ruído e um *corpus* diferente do TIMIT. Como o método proposto calcula novos modelos SVM para cada conjunto de treinamento dos conjuntos comprimidos S_{BmB2}, novos modelos são computados para todos os experimentos da análise de robustez, exceto para os experimentos de ataque de deslocamento de quadros. Foi realizado um experimento para cada condição adversa abordada a seguir.

7.1 - ARQUIVOS COM DURAÇÃO VARIÁVEL

Uma das condições que potencialmente afeta o desempenho do método proposto é a duração dos arquivos AMR ao longo do *corpus*, uma vez que a quantidade de parâmetros AMR extraídos é proporcional à quantidade dos quadros de voz codificados (que são diferentes dos quadros de ruído de fundo SID). Em outras palavras, quanto mais quadros de voz o arquivo tem, mais parâmetros podem ser extraídos e melhor se configuraria, em tese, a estatística desses parâmetros. Em casos extremos, o desempenho do método se tornaria estável a partir de certa duração, já que as estatísticas não mudariam a partir daí e, por outro lado, com uma duração muito pequena, o desempenho do método seria inferior, haja vista a pequena quantidade de parâmetros disponíveis. A proposta para este experimento é gerar, a partir do *corpus* TIMIT, novos *corpora*, um para cada duração fixa escolhida dos arquivos AMR, e em seguida obter os modelos da SVM para cada taxa BR2, da mesma forma realizada nos experimentos anteriores.

7.1.1 - Geração de novos *corpora*

Para confirmar as hipóteses de comportamento do método proposto, foram realizados experimentos em que os arquivos AMR tinham durações fixas dentro do *corpus* e gerados a partir do *corpus* TIMIT. Dessa forma, as durações escolhidas e a forma de geração estão descritas a seguir:

- 125 ms, 250 ms e 500 ms: os arquivos foram gerados a partir da extração de um trecho dos arquivos TIMIT definido a partir do máximo envelope de voz (comando *envelope* no MATLAB®), que é fixado como o meio do trecho extraído. Foram gerados novos *corpora* com 6300 arquivos.
- 1 s e 2 s: inicialmente foram descartados arquivos TIMIT com durações inferiores às pretendidas e depois os restantes foram cortados entre um ponto e o final do arquivo, de modo que a duração fosse a desejada. Dessa forma, foram gerados novos *corpora* com 6298 e 5809 arquivos, respectivamente.
- 3 s: para gerar um *corpus* de arquivos com essa duração e com um número de arquivos próximo a 6300, foi necessário unir arquivos TIMIT gravados pelos mesmos locutores (para manter o áudio nas mesmas condições de gravação) e extrair segmentos de 3 segundos a partir do início dessa união, descartando as últimas amostras. Essa extração de segmentos foi realizada para cada união correspondente a um locutor disponível (630 locutores). Dessa forma, foi gerado um novo *corpus* com 6128 arquivos.

Com o objetivo de gerar novos *corpora* com a máxima uniformidade de gravação possível em cada arquivo e com um número de arquivos mais próximo possível de 6300, a forma de gerar os *corpora* dependeu da duração pretendida. Observa-se que isso foi necessário porque os arquivos do *corpus* TIMIT não têm duração fixa e, na sua maioria, têm duração próxima a 1 segundo.

7.1.2 - Resultados

Para cada novo *corpus*, foram gerados novos conjuntos comprimidos S_{BmB2} e computados novos modelos de SVM para cada BR2, de modo a ser possível aferir as acurácias. Na Tabela D-3 do Apêndice D estão expostos os resultados detalhados das acurácias de teste para todas as taxas de bits AMR e durações, revelando que o método proposto é efetivo para discriminar arquivos AMR com compressão dupla mesmo com curtas durações. As acurácias médias se elevam progressivamente à medida que as durações dos arquivos AMR aumentam, partindo de 94% (125 ms) e alcançando uma estabilização a partir de 2 s de duração, com valores médios na faixa de 99%, confirmando o desempenho inicialmente esperado para o método.

7.2 - ATAQUE DE DESLOCAMENTO DE QUADRO

Esse tipo de ataque testa o método de detecção quando um deslocamento de quadro ocorre no domínio do tempo, isto é, quando a estrutura de quadros original é quebrada mediante a supressão de amostras. Tal teste é útil porque alguns métodos podem extrair características relacionadas à estrutura de quadros, havendo, pois, queda no desempenho

quando os quadros são deslocados.

O método proposto não analisa quadros no domínio do tempo para detectar a compressão dupla como os algoritmos reportados na literatura; portanto, o método é teoricamente imune ao ataque de deslocamento de quadro. Apesar de essa hipótese ser plausível, é fato que o codificador AMR analisa os quadros para computar os parâmetros no domínio da compressão e um deslocamento de quadro aplicado aos arquivos do *corpus* TIMIT poderia, potencialmente, perturbar o desempenho do método.

Devido às modificações mínimas no *corpus* TIMIT introduzidas pelo ataque de deslocamento de quadros, foram usados os mesmos modelos SVM computados para o *corpus* TIMIT original com os conjuntos S_{BmB2}, assumindo o mesmo procedimento para o teste. Dessa forma, foram calculadas novas matrizes de características para cada BR2 e para cada deslocamento de quadro, limitados ao tamanho do quadro de análise AMR (160 amostras), ou seja, 25, 50, 75, 100, 125 e 150 amostras, mediante a supressão das primeiras amostras de cada arquivo do *corpus* TIMIT antes do cálculo das características. Os resultados detalhados na Tabela D-4 confirmam que o método proposto é imune ao ataque de descolamento de quadro, pois as acurácias médias são aproximadamente as mesmas alcançadas com o *corpus* TIMIT original.

7.3 - ÁUDIO COM RUÍDO BRANCO ADICIONADO

O ruído diminuirá o desempenho do método proposto porque as diferenças sutis entre as características de arquivos AMR com compressão simples e dupla podem ficar mascaradas. A intensidade da influência do ruído nas acurácias do método depende da forma como o algoritmo é elaborado para a extração de características. Como o método proposto é baseado em parâmetros extraídos do áudio AMR codificado, não utilizando o áudio decodificado no domínio do tempo, provavelmente a adição de ruído degradará em menor grau as acurácias.

7.3.1 - Procedimento para adição de ruído ao *corpus* TIMIT

Como a adição de ruído aos arquivos do *corpus* TIMIT dá origem a novos *corpora*, foram calculados novos modelos SVM para cada BR2 e para cada relação sinal ruído (SNR) após adição de ruído branco gaussiano (AWGN). O valores de SNR selecionados foram 5dB, 10dB, 20dB e 30dB, calculados mediante a medição da potência do sinal (média das amostras ao quadrado, comando *awgn* do MATLAB®) antes de adicionar o ruído, todas as medidas em dB.

Para a SNR de 30 dB, o áudio resultante é percebido praticamente como o original sem ruído. Para a SNR de 20 dB, o ruído pode ser percebido de forma muito discreta quando há pausas nas falas gravadas. Já para SNR de 5 dB, o ruído é percebido ao fundo de maneira bem evidente, dificultando o entendimento das falas.

7.3.2 - Resultados

A Tabela D-5 no Apêndice D contém todas as acurácias dos experimentos. Nela é possível observar que o método proposto é praticamente imune ao ruído AWGN se a SNR vale 30 dB (acurácia média de 99,04%) e o desempenho é satisfatório mesmo em baixas SNR, como 5dB (acurácia média de 98,04%). Pela observação de todos os resultados, o desempenho se mantém aceitável após a adição de ruído AWGN.

7.4 - *CORPUS* CARIOCA1

Ao utilizar o método proposto apenas com o *corpus* TIMIT, as condições do problema ficam delimitadas a um nível de ruído de fundo de gravação e a um idioma, já que esse *corpus* é altamente homogêneo. Embora os experimentos com o *corpus* TIMIT sejam necessários para a comparação com o estado da arte e para controlar as possíveis interferências nos resultados, o método deve ser testado com outro *corpus*, um procedimento que é mais coerente com o uso forense pretendido.

7.4.1 - Formação de novo *corpus*

O *corpus* CARIOCA1 (Esquef *et al.*, 2014) foi escolhido para aferir o desempenho do método proposto porque ele é formado por falas em português do Brasil e gravadas de chamadas de telefone fixo, ou seja, sob condições mais ruidosas e conteúdo limitado à banda telefônica. Esse *corpus* foi originalmente montado para o teste de algoritmos de detecção de descontinuidades no sinal ENF, contendo 200 arquivos de áudio PCM sem compressão, com durações entre 19,86 segundos e 34,40 segundos, taxa de amostragem de 44 kHz e amostras de 16 bits.

O *corpus* CARIOCA1 foi modificado de forma a ser possível extrair arquivos com duração de 1 segundo para criar um novo *corpus* similar ao TIMIT. Inicialmente os arquivos foram filtrados, subamostrados para a taxa de 8 kHz e posteriormente picotados sequencialmente desde o começo, em quadros de 1 segundo e sem sobreposição. Ao final do picote sequencial de cada um dos 200 arquivos, a última parte desse arquivo é alcançada e apenas considerada como um novo arquivo se durar mais de 915 ms, que é a duração mínima dos arquivos do *corpus* TIMIT, sendo descartada se durar menos. Dessa forma, foi gerado um novo *corpus* com 5875 arquivos para computar novos modelos SVM

e aplicar o método proposto.

7.4.2 - Resultados

A Tabela D-6 no Apêndice D demonstra que o método proposto também é efetivo com outro *corpus*, pois a acurácia média atinge 99,35% e $\overline{\text{TP}}$ média atinge 99,20%, taxas inclusive mais altas do que com o *corpus* TIMIT. Os valores de NCF (número atual das características) e BNF (melhor número de características) são diferentes do *corpus* TIMIT e para cada taxa de bits AMR considerada, demonstrando que o método proposto adapta esses parâmetros ao conjunto de treinamento em questão.

7.5. ÁUDIO COM RUÍDO FORENSE ADICIONADO

A influência do ruído tipo AWGN no desempenho do método proposto foi explorada na Seção 7.3, contudo esse tipo de ruído é apenas um dentre vários que podem estar presentes no áudio analisado. O estudo da variação de desempenho do método quando o ruído está presente é de grande interesse da área forense, haja vista a inevitável adição de ruído ao conteúdo de voz registrado. Dessa forma, nesta análise de robustez, é explorada a adição de um tipo de ruído diferente do AWGN e mais próximo de casos reais. Para delimitar o escopo desta análise, será acrescentado ao *corpus* TIMIT um ruído gerado predominantemente pelo equipamento gravador e extraído de uma gravação ambiental real.

7.5.1 - Procedimento para Adição de Ruído Forense ao *Corpus* TIMIT

Inicialmente foi selecionado um trecho de arquivo de áudio forense sem conteúdo de voz e formado apenas por ruído de fundo típico de gravações ambientais. O trecho selecionado tem a duração do maior arquivo do *corpus* TIMIT (7788 ms) para que a adição de ruído seja sempre feita com trechos contínuos de ruído ao longo do *corpus*. A Figura 7-1 é um espectrograma do trecho selecionado na sua forma original, isto é, na taxa de amostragem 44,1 kHz e 16 bits por amostra (o espectrograma foi construído com FFT de tamanho 256 e janelamento Hanning). Já a Figura 7-2 é o espectrograma do trecho convertido para taxa de amostragem 8 kHz, mantidas as demais condições, para ser usado na adição de ruído ao *corpus* TIMIT (também reamostrado a 8 kHz). Observa-se que o ruído apresenta espectro variante no tempo com energia de maior intensidade entre 2500 Hz e 3000 Hz.

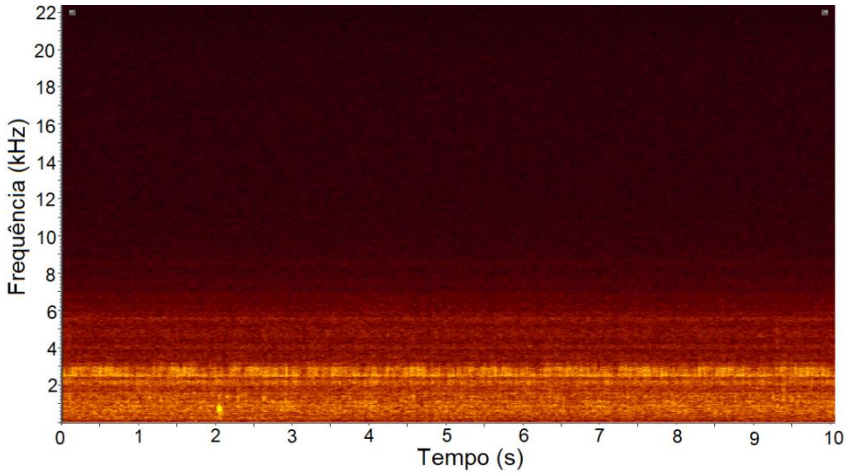


Figura 7-1 – Espectrograma do trecho de ruído forense selecionado na taxa de amostragem original de 44,1 kHz (FFT de tamanho 256).

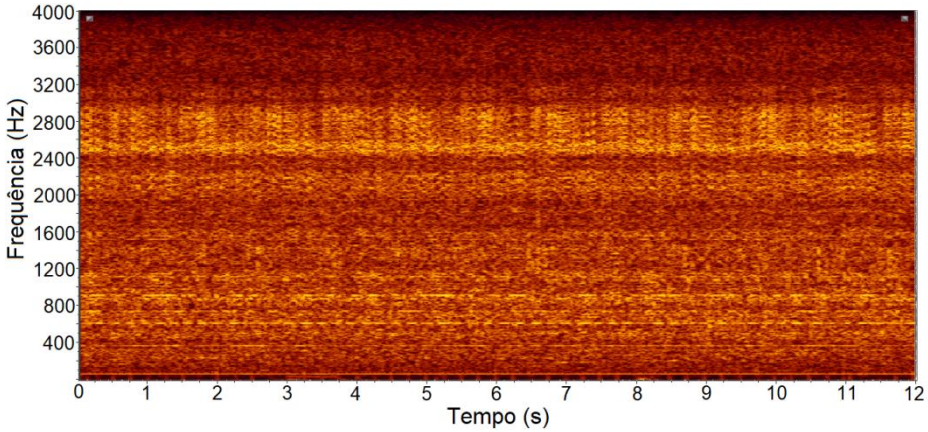
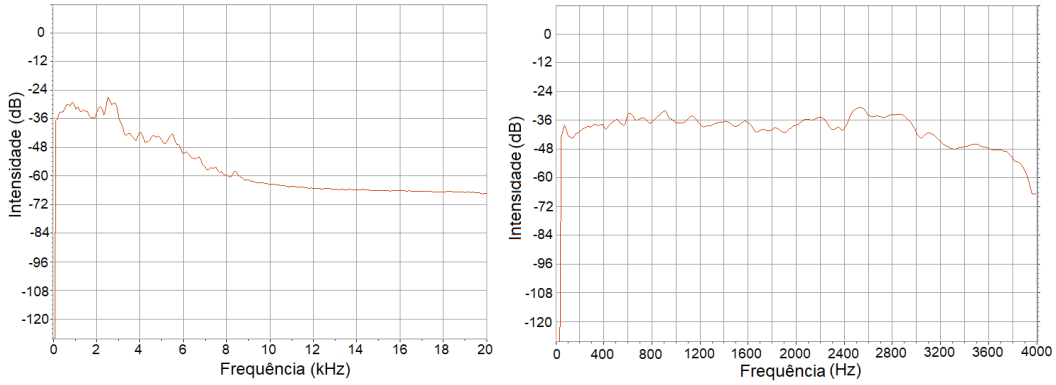


Figura 7-2 – Espectrograma do trecho de ruído forense selecionado na taxa de amostragem de 8 kHz (FFT de tamanho 256).

A Figura 7-3 (a) é o espectro de média de longo termo (LTA, *long term average*) do sinal original, calculado pela média das amplitudes de FFT com janelas de tempo de tamanho 256, enquanto a Figura 7-3 (b) é o espectro LTA do sinal amostrado a 8 kHz. As figuras evidenciam a variedade de componentes espectrais presentes no trecho de ruído.



(a) (b)

Figura 7-3 – Espectros LTA do trecho de ruído forense selecionado na taxa de amostragem original de 44,1 kHz (a) e 8kHz (b) (FFT de tamanho 256).

Para gerar novos *corpora* com diferentes relações sinal-ruído, foi utilizado o mesmo procedimento empregado com ruído tipo AWGN, ou seja, as potências dos arquivos do *corpus* TIMIT foram medidas (soma das amostras ao quadrado), assim como a potência do arquivo de ruído com a mesma duração do respectivo arquivo TIMIT, e calculado o ganho necessário no arquivo de ruído para uma desejada SNR, tudo em dB. Foram testadas as SNR de valores 5dB, 10dB, 20dB e 30dB. Para cada BR2 e SNR, foram calculados novos modelos SVM, totalizando 32 modelos e experimentos distintos.

Para a SNR de 30 dB, o áudio resultante é percebido praticamente como o original com ruído muito discreto ao fundo. Para a SNR de 10 dB e 20 dB, o ruído pode ser percebido de forma discreta quando há pausas nas falas gravadas. Já para SNR de 5 dB, o ruído é percebido ao fundo de maneira muito evidente, dificultando ou impossibilitando o entendimento das falas.

7.5.2 - Resultados

A Tabela D-7 mostra que o método proposto é praticamente imune ao ruído forense escolhido se a SNR vale 30 dB (acurácia média de 99,07%) e o desempenho é satisfatório mesmo em baixas SNR, como 5 dB (acurácia média de 96,48%). Pela observação de todos os resultados, o desempenho se mostra o mesmo daquele alcançado com ruído do tipo AWGN com SNR de 20 dB e um pouco inferior para 5 dB e 10 dB (média de 97,73% para AWNG e média de 96,66% para o ruído forense). Conclui-se que, mesmo após a adição do ruído forense selecionado, o método apresenta desempenho aceitável.

8- COMPARAÇÃO COM O ESTADO DA ARTE E DISCUSSÃO

Conforme mencionado na Seção 2.2, foram identificados três métodos na literatura para discriminar arquivos AMR com compressão dupla e, em todos eles, o algoritmo é baseado no áudio descomprimido. Além desses métodos, podem ser citados três métodos em que o algoritmo é baseado em características no domínio da compressão, que é uma abordagem que se mostra adequada para a detecção de compressão dupla de arquivos AMR (Sampaio, 2019). O método com maior desempenho é completamente descrito nesta tese (Sampaio e Nascimento, 2020) e os outros dois métodos (Sampaio e Nascimento, 2018) (Sampaio e Nascimento, 2019) são versões cujos blocos são também descritos nesta tese. No método proposto em Sampaio e Nascimento (2018), as características são baseadas apenas nos coeficientes LP e LSP e não são usados o escalonamento robusto e nem seleção de características. Já no método descrito em Sampaio e Nascimento (2019), é adicionado o escalonamento robusto no processamento das características usadas em Sampaio e Nascimento (2018).

Na avaliação comparativa do método proposto com os resultados publicados, conforme a Tabela 8-1, conclui-se que ele supera o estado da arte em todas as taxas de bits, mantidas as mesmas condições de teste. A coluna *média* denota a média aritmética das acurácias ao longo das taxas AMR e, conforme é possível observar, o método proposto alcança a maior acurácia média.

Tabela 8-1 – Comparação de desempenho entre os métodos existentes para a detecção de compressão dupla AMR (*corpus* TIMIT em conjuntos S_{BmB2}).

Métodos	Acurácias Médias em %								
	BR2 (kbits/s)								
	4,75	5,15	5,90	6,70	7,40	7,95	10,2	12,2	*Média*
(Shen *et al.*,2012)	79,76	83,73	87,26	86,17	80,14	82,36	85,31	88,16	84,11
(Luo *et al.*,2014)	91,14	91,32	91,18	91,23	91,39	91,28	91,27	91,33	91,27
(Sampaio e Nascimento, 2018)	92,62	92,53	94,03	93,80	93,72	94,00	93,99	94,58	93,66
(Luo *et al.*,2017)	98,78	98,88	98,74	98,77	98,95	98,87	98,84	98,82	98,83
(Sampaio e Nascimento, 2019)	99,13	99,08	98,90	98,77	98,83	99,00	98,84	98,48	98,88
Método Proposto (Sampaio e Nascimento, 2020)	**99,35**	**99,21**	**99,16**	**99,12**	**99,01**	**99,07**	**99,19**	**99,18**	**99,16**

Os resultados das simulações computacionais usando o *corpus* TIMIT demonstram um melhor desempenho do método proposto frente aos demais. A construção de características diretamente do domínio da compressão, associada ao escalonamento robusto e seleção de características, provou ser muito eficiente na detecção de arquivos S-AMR e

D-AMR. A combinação das características no domínio da compressão, escalonamento robusto e o algoritmo de seleção de características SVM-RFE CBR levam às altas acurácias nos conjuntos comprimidos S_{BmB2} e S_{B1B2} (Tabela D-1 e Tabela D-2).

O acréscimo do procedimento de escalonamento robusto trata a influência dos valores atípicos e concentrados nas características no domínio da compressão, conforme visto na Seção 5.8. Esse escalonamento minimiza os efeitos indesejados dos valores atípicos e aumenta a dispersão das características para facilitar a detecção da compressão dupla AMR.

O método de seleção de características SVM-RFE CBR realiza compensações para melhorar a acurácia: se as características não são no todo úteis, o algoritmo elimina algumas delas. O valor máximo do melhor número de características BNF é menor do que o número atual das características NCF para todas as taxas de bits, conforme visto na Tabela 5-10. A acurácia de validação cruzada diminui após o ponto de BNF, ao tempo que o número de características aumenta, de acordo com o exemplo da Figura 5-6, o que não é um fato intuitivamente esperado. A escolha de um subconjunto específico das características aumenta as acurácias provavelmente porque isso minimiza o problema do sobreajuste na SVM.

O disposto na Figura 5-7 indica que as características de maior dispersão para a detecção da compressão dupla AMR são relacionadas às distribuições de probabilidade de primeiros dígitos dos coeficientes LSP. Esse é um resultado interessante porque as distribuições de probabilidade de primeiros dígitos dos coeficientes MDCT também são características importantes para a detecção da compressão dupla MP3, conforme citado na Seção 2.1. Se a análise da Figura 5-7 for ampliada para mais características, a predominância das distribuições de probabilidade de primeiros dígitos dos coeficientes LSP pode também ser observada, conforme se verifica para $410 \equiv m_{q_4}(3)$, $488 \equiv m_{q_7}(3)$, $514 \equiv m_{q_8}(3)$, e assim por diante.

As composições de histogramas da Figura 6-1 e da Figura 6-2 mostram que as características no domínio da compressão, escalonamento robusto e seleção de características revelam a assinatura espectral do áudio AMR com compressão dupla para o classificador SVM. Da mesma forma, os histogramas da Figura 6-3 e da Figura 6-4 mostram como o método proposto expõe as diferenças entre compressão simples e dupla, haja vista que a redução de dimensionalidade gera histogramas com formatos similares, mas com orientações muito diferentes (observar o aspecto de inversão/espelhamento dos

histogramas nas figuras). Mesmo se um *corpus* diferente do TIMIT for usado nos experimentos, o emprego das três técnicas aborda o problema da compressão dupla AMR com eficiência.

Os resultados da Tabela D-2 mostram uma tendência de aumento das acurácias na transcodificação para cima (valores na extremidade inferior esquerda da tabela) e, portanto, uma tendência de diminuição das acurácias na transcodificação para baixo (valores na extremidade superior direita da tabela), similar aos resultados apresentados em Shen *et. al* (2012). A relação entre a primeira e a segunda taxa de compressão influencia o método, fato que é esperado devido à perda de informação nas operações de transcodificação para baixo. Essa condição também é observada em outros codecs, como o MP3 (Yang *et. al*, 2009) (Yang *et. al*, 2010) (Liu *et. al*, 2010) (Luo *et. al*, 2012).

A análise de robustez indica que o método proposto é estável sob algumas condições adversas. Se os clipes de áudio são muito pequenos, como 125 ms, mas o conteúdo de voz está presente, o método ainda oferece acurácias aceitáveis, como apresentado na Tabela D-3, possivelmente porque as medidas estatísticas são menos afetadas pelo comprimento do clipe acima de uma duração mínima. Essa deve também ser a razão pela qual o método é imune ao ataque de deslocamento de quadros como visto na Tabela D-4. A distorção inserida pelo ruído AWGN e por um ruído forense degrada as acurácias do método, como observado na Tabela D-5 e na Tabela D-7, mas os efeitos não são tão intensos, mesmo se a SNR é baixa. Esse resultado provavelmente ocorre porque as características mais importantes, baseadas em parâmetros no domínio da frequência, como os coeficientes LSP, são menos afetadas pelo ruído. Por outro lado, se o *corpus* é trocado, as acurácias se mantêm altas, como mostrado na Tabela D-6, indicando que o método adapta automaticamente os modelos SVM e os resultados do método SVM-RFE CBR de acordo com os conjuntos de treinamento e taxas de bits, ou seja, de acordo com *corpus* em questão. Essa propriedade é especialmente útil no contexto forense porque as condições de gravação variam caso a caso.

Algumas vantagens do método proposto podem ser citadas em relação aos métodos já publicados:

- As acurácias com o *corpus* TIMIT, para 20 experimentos, são mais altas para todas as taxas de bits, assumindo valores maiores que 99%. As altas taxas oferecem maior confiabilidade para a detecção da compressão dupla AMR num caso forense hipotético, haja vista que não foram encontrados métodos determinísticos para tal tarefa (todos os métodos são baseados em aprendizado de máquina).

- O cálculo das características é direto e rápido, pois a extração dos parâmetros AMR e as operações envolvidas são baseadas em procedimentos simplificados, como desempacotamento de parâmetros e estatísticas básicas.
- A decodificação de arquivos AMR é desnecessária para aplicar o método, pois o cálculo das características é completamente baseado em parâmetros do domínio da compressão.
- A extração de características é baseada num formato comprimido de menor redundância, bem diferente da forma de onda no domínio do tempo. O fluxo de bits AMR usa uma representação da voz com baixa redundância, a qual proporciona vetores de características mais eficientes para a SVM.
- A seleção de características se adapta à taxa de bits AMR e ao conjunto de treinamento, gerando um número variável de características utilizadas em cada experimento. Ao invés de um número fixo de características, o método proposto ordena as características disponíveis e determina o melhor subconjunto delas para maximizar as acurácias em cada taxa, propriedade que traz flexibilidade frente a condições adversas como aquelas abordadas na análise de robustez.
- As características extraídas permitem melhor visualizar e compreender o problema, inclusive apontar quais são as mais relevantes conforme o critério do algoritmo SVM-RFE CBR. Essa abordagem não é oferecida por outras metodologias, como na extração a automática pela rede SAE (Luo *et al*., 2017).
- De acordo com a análise de robustez, o método proposto oferece acurácias maiores do que o estado da arte com *corpus* diferente do TIMIT e com áudio contaminado por ruído AWGN, atingindo acurácias da ordem de 99%.

9 - CONCLUSÕES E RECOMENDAÇÕES

A multimídia forense e a análise de áudio digital forense são áreas relativamente novas da ciência forense que são dedicadas a processar provas multimídia, como áudio, imagens e vídeo. O áudio digital AMR deve ser autêntico para ser admitido como prova em processos judiciais. Uma técnica útil para verificar a autenticidade do áudio no formato AMR é determinar se ele foi comprimido uma única vez ou duas, pois, se tal áudio foi submetido à compressão dupla, ele provavelmente é inautêntico porque um arquivo AMR original deve apresentar apenas uma única compressão. Contudo, a detecção da compressão dupla AMR consiste em um problema de engenharia complexo cuja solução ainda está em andamento.

O estado da arte e trabalhos prévios sobre a detecção da compressão dupla AMR usam apenas áudio decodificado. Nesta tese, uma nova abordagem foi proposta para detectar a compressão dupla AMR utilizando características no domínio da compressão baseadas nos coeficientes de predição linear, pares de espectro de linhas, períodos de *pitch*, ganhos de *pitch*, energia do quadro, ganho do dicionário fixo e patamar de ruído do quadro. Após exaustivos experimentos computacionais com o *corpus* TIMIT, foi constatado que o método proposto oferece alto desempenho.

O método é inovador em vários aspectos, pois usa apenas características no domínio da compressão, escapando de transitórios complexos do áudio descomprimido. Ele também usa a engenharia de características para projetar características estatísticas simples e adaptadas ao problema em questão, superando os algoritmos de extração automática de características, como os autocodificadores empilhados, neste caso específico. Além disso, o método é o primeiro conhecido a usar escalonamento robusto e um algoritmo de seleção de características para discriminar áudio AMR com compressão dupla. Dessa forma, a metodologia proposta se configura como a principal contribuição científica da presente tese.

A análise de robustez mostrou que o método proposto é confiável e provê altas acurácias, mesmo usando um *corpus* diferente e sob condições adversas, como ruído adicionado, curta duração dos arquivos e ataque por deslocamento de quadro.

Apesar dos resultados apresentados, o método proposto tem limitações que demandam estudos continuados. Por exemplo, como o método é baseado principalmente nos coeficientes LP e LSP, além do período de *pitch*, a extração de características só é possível em quadros em que há voz. Se o áudio em questão tiver pouca ou nenhuma voz,

como nos casos de análise de eventos acústicos (um exemplo seria um sinal proveniente de um disparo de arma de fogo ou de explosões), o método proposto é ineficaz para a autenticação do áudio. Uma solução possível para tal limitação seria o uso dos quadros SID gerados pelo codificador AMR quando não há voz de intensidade mínima.

Outro aspecto limitador do uso do método é a necessidade de um *corpus* homogêneo para o aprendizado da SVM. Em muitos casos de multimídia forense, os arquivos a serem autenticados são únicos ou em pequeno número, não havendo outros arquivos gravados nas mesmas condições para a montagem de *corpus*. Se o gravador envolvido no caso estiver disponível, a geração de *corpus* em condições ambientais similares pode ser uma solução. Outra opção seria a criação prévia de modelos específicos para diversas condições de gravação, como ambientes diferentes e chamadas telefônicas, porém sem garantia de adequação satisfatória às condições de gravação dos arquivos recebidos para um exame forense. A criação de uma ferramenta forense para detectar compressão dupla AMR deve enfrentar esse problema.

REFERÊNCIAS BIBLIOGRÁFICAS

3GPP − 3rd Generation Partnership Project (2017). AMR Codec - Release 10. Disponível em: <http://www.3gpp.org/ftp/Specs/archive/26_series/26.104/>. Acesso em: 10 ago, 2018.

3GPP − 3rd Generation Partnership Project TS 26.071 v13.0.0 (2015): Technical Specification Group GSM/EDGE - Radio Access Network - Link Adaptation (Release 13).

3GPP − 3rd Generation Partnership Project TS 26.090 v13.0.0 (2015): Mandatory Speech Codec speech processing functions - Adaptive Multi-Rate (AMR) speech codec - Transcoding functions (Release 13).

3GPP − 3rd Generation Partnership Project TS 26.101 v13.0.0 (2015): Mandatory Speech Codec speech processing functions - Adaptive Multi-Rate (AMR) speech codec frame structure (Release 13).

Battiato, S.; Giudice, O.; Paratore, A. (2016). Multimedia Forensics: Discovering the History of Multimedia Contents. In: Proc. 17th International Conference on Computer Systems and Technologies (CompSysTech), Palermo, Italy, p. 5-16.

Cao, H., Stojkovic, I., Obradovic, Z. (2016). A robust Data Scaling Algorithm to Improve Classification Accuracies in Biomedical Data. BMC Bioinformatics, vol. 17:359.

Chang, S. (1995). Compressed-domain Techniques for Image/Video Indexing and Manipulation. In: Proc. Int. Conf. on Image Processing, Washington, USA, pp. 314-317.

Chang, CC., Lin, CJ., (2011). LIBSVM: a Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology, Vol. 2, Issue 3, Article No. 27.

Chang, L., Yu, X., Tan, H., Wan, W. (2007). Research and Application of Audio Feature in Compressed Domain. In: Proc. IET Conf. Wireless. Mobile and Sens. Networks, Shangai, China.

Delac, K., Grgic, M., Grgic, S. (2009). Face Recognition in JPEG and JPEG2000 Compressed Domain. Image and Vision Computing, vol. 27, no. 8, pp. 1108-1120.

Esquef, P. A. A., Apolinário Jr., J. A., Biscainho, L. W. P. (2014). Edit Detection in Speech Recordings Via Instantaneous Electric Network Frequency Variations. IEEE Transactions on Information Forensics and Security, vol. 9, no. 12, p. 2314-2326.

Fathima, N. P., Kishnan, C. V. V. (2018). Analysis of Different Classifier for the Detection of Double Compressed AMR Audio. International Journal of Advance Research, Ideas and Innovations in Technology, vol. 4-2, pp. 98–107.

Fu, D., Shi, Y., Su, Q. (2007). A generalized Benford's law for JPEG coefficients and its applications in image forensics. In: Proc. SPIE Electronic Imaging, Security and Watermarking of Multimedia Contents IX, vol. 6505, pp. 47-58.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., Dahlgren, N. L., Zue, V. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. Disponível em: <https://catalog.ldc.upenn.edu/LDC93S1>. Acesso em: 26 Feb. 2018.

Grigoras, C. (2003). Forensic Analysis of the Digital Audio Recordings - The Electric Network Frequency Criterion. Forensic Science International, vol. 136, Suppl. 1, p. 368-369.

Guyon, I., Elisseeff, A. (2003), An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, vol. 3, p. 1157-1182.

Haykin, S. Neural Networks and Learning Machines (2003). 3rd Edition, USA, Prentice Hall.

Heaton, J. (2016). An Empirical Analysis of Feature Engineering for Predictive Modeling. In: Proc. IEEE SoutheastCon 2016, Norfolk, USA, pp. 1–6.

Jenner, F. and Kwasinski, A. (2012). Highly Accurate Non-intrusive Speech Forensics for Codec Identifications from Observed Decoded Signals. In: Proc. IEEE Int. Conf. Acoust., Speech Signal Process., Kyoto, Japan, pp. 1737–1740.

Joachims, T. (1998). Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Proc. 10th European Conference on Machine Learning: ECML-98, Chemnitz, Germany, pp. 137-142.

Kajstura, M., Trawinska, A., Hebenstreit, J. (2005). Application of the Electrical Network Frequency (ENF) Criterion: a Case of a Digital Recording. Forensic Science International, no. 155, p. 165-171.

LIBSVM version 3.2.2. (2018). Disponível em: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. Accesso em: 10 jan. 2018.

Liu, Q., Sung, A. H., Qiao, M. (2010). Detection of Double MP3 Compression. Cognit. Comput., Vol.2, No. 4; pp. 291–296.

Luo, H., Luo, Eleftheriadis, A. (2000). On Face Detection in the Compressed Domain. In: Proc. 8th ACM Int. Conf. on Multimedia (Multimedia‘00), Marina Del Rey, USA, pp. 285-294.

Luo, D., Luo, W., Yang, R., Huang, J. (2012). Compression History Identification for Digital Audio Signal. In: Proc. IEEE Int. Conf. Acoust., Speech Signal Process., Kyoto, Japan, pp. 1732-1736.

Luo, D., Yang, R., Huang, J. (2014). Detecting Double Compressed AMR Audio Using Deep Learning. In: Proc. Int. Conf. Acoust., Speech, Signal Process, Appl., Florence, Italy, pp. 2669–2673.

Luo, D., Yang, R., Lin, B., Huang, J. (2017). Detection of Double Compressed AMR Audio Using Stacked Autoencoder. IEEE Trans. Inf. Forensics Security, Vol.12, No. 2, pp. 432–444.

Maaten, L.V., Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, vol. 9, p. 2579-2605.

Petracca, M., Servetti, A., De Martin, J. C. (2005). Low-complexity Automatic Speaker Recognition in the Compressed GSM AMR Domain. In: Proc. IEEE Int. Conf. Multimedia and Expo (ICME), Netherlands, pp. 1-4.

Petracca, M., Servetti, A., De Martin, J. C. (2006). Performance Analysis of Compressed-Domain Automatic Speaker Recognition as a Function of Speech Coding Technique and Bitrate. In: Proc. IEEE Int. Conf. Multimedia and Expo (ICME), Toronto, Canada, pp. 1396-1396.

Pfeiffer, S., Vincent, T. (2003). Survey of Compressed Domain Audio Features and Their Expressiveness. In: Proc. SPIE Conf. Electronic Imaging, San Francisco, USA, pp. 133-147.

Reis, P. M. G. I., Costa, J. P. C. L., Miranda, R. K., Del Galdo, G. (2017). ESPRIT-Hilbert-based Audio Tampering Detection with SVM Classifier for Forensic Analysis Via Electrical Network Frequency. IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, p. 853-864.

Rodríguez, D. P. N., Apolinário, J. A., Biscainho, L. W. P. (2010). Audio Authenticity: Detecting ENF Discontinuity with High Precision Phase Analysis. IEEE Transactions on Information Forensics and Security, vol. 5, no. 3, p. 534-543.

Romero, D., Wilson, C. Y. (2010). Automatic Acquisition Device Identification from Speech Recordings. In: Proc. 2010 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2010), 2010, Texas, USA, p. 1806-1809.

Sampaio, J. F. P., Nascimento, F. A. O. (2018). Double Compressed AMR Audio Detection Using Linear Prediction Coefficients and Support Vector Machine. In: Proc. The Brazilian Conf. on Automation (CBA2018), Joao Pessoa, Brazil. Disponível em: < www.doi.org/10.20906/CPS/CBA2018-0005>

Sampaio, J. F. P.(2019). AMR Compressed-Domain Analysis for Multimedia Forensics Double Compression Detection. The Brazilian Journal of Police Science, vol. 9, no. 2, pp. 43–70 <http://dx.doi.org/10.31412%2Frbcp.v9i2.534>

Sampaio, J. F. P., Nascimento, F. A. O. (2019). Forensic AMR Double Compression Detection Using Linear Prediction Coefficients and Robust Scaling. J. Audio Eng. Soc., vol. 67, issue 10, p. 795-806. < https://doi.org/10.17743/jaes.2019.0028>

Sampaio, J. F. P., Nascimento, F. A. O. (2020). Detection of AMR Double Compression using Compressed-domain Speech Features, Forensic Science Int.: Digital Investigation, vol. 33C, 200907. < https://doi.org/10.1016/j.fsidi.2020.200907>

Scientific Working Group on Digital Evidence – SWGDE (2017). SWGDE Best Practices for Digital Audio Authentication – Version 1.2 – Feb. 2017. Disponível em: < https://www.swgde.org/documents/Current%20Documents>. Acesso em: 09 Ago. 2018.

Shen, Y., Jia, J., Cai, L. (2012). Detecting Double Compressed AMR-format Audio Recordings. In: Proc. 10th Phonetics Conf. China (PCC), Shanghai, China, pp. 1–5.

Sjoberg, J., Westerlund, M., Lakaniemi, A., Xie, Q. (2007). RFC 4867 - RTP Payload Format and File Storage Format for the Adaptive Multi-Rate (AMR) and Adaptive Multi-Rate Wideband (AMR-WB) Audio Codecs . The Internet Engineering Task Force (IETF®).

Spanias, A. S. (1994). Speech Coding: a Tutorial Review. Proceedings of the IEEE, vol. 82, no. 10, pp. 1541–1582.

Su, H., Garg, R., Hajj-Ahmad, A., Wu, M. (2013). ENF Analysis on Recaptured Audio Recordings. In: Proc. 2013 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Canada, p. 3018-3022.

Yan, K., Zhang, D. (2015). Feature Selection and Analysis on Correlated Gas Sensor Data with Recursive Feature Elimination, Sensors and Actuators B: Chemical, vol. 212, pp. 353-363.

Yang, R., Shi, Q., Huang, J. (2009). Defeating Fake-quality MP3. In: Proc. ACM Workshop Multimedia Secur., Princeton, USA, pp.117–124.

Yang, R., Shi. Y., Huang, J. (2010). Detecting Double Compression of Audio Signal. In: Proc. SPIE Conference on Media Forensics and Security II, vol. 7541, USA, pp. 1-10.

Yeo, B. L., Liu, B. (1995). Rapid Scene Analysis on Compressed Video. IEEE Trans. on Circuits and Systems for Video Tech., vol. 5, no.6, pp. 533–544.

Zakariah, M., Khan, M. K,. Malik, H. (2017). Digital Multimedia Audio Forensics: Past, Present and Future. Multimedia Tools and Applications, vol. 77, p. 1009-1040.

APÊNDICES

A – ESCALONAMENTO ROBUSTO

O algoritmo GL (logístico generalizado) é derivado do método de equalização de histogramas e usa funções logísticas generalizadas para se ajustar às funções densidade de probabilidade cumulativas dos dados em questão e, dessa forma, pode mapear os dados originais em um intervalo desejado (Cao *et. al*, 2016).

Foram consideradas duas propriedades do algoritmo GL que resolvem os problemas das características no domínio da compressão dos arquivos AMR. Na primeira, o algoritmo mapeia os dados para um intervalo de valores uniformemente distribuídos, tornando mais discerníveis os pontos que antes eram densamente concentrados num intervalo, propriedade essa que permite uma maior representação das diferenças que existem entre eles. Na segunda propriedade, o algoritmo reduz as distâncias entre os pontos atípicos e os demais pontos, algo que torna o algoritmo GL robusto aos valores atípicos.

Considerando que os valores de uma dada característica é uma variável aleatória χ, o valor escalonado *v'* é definido como:

$$v' = P_{\chi}(v) \tag{A.1}$$

em que P_{χ} é a função densidade de probabilidade cumulativa (CDF) da variável aleatória χ.

Usar uma CDF para mapear valores é uma técnica já conhecida de equalização de histogramas no processamento de imagens digitais. A principal diferença entre o algoritmo GL e a equalização de histogramas é que a CDF não é usada apenas para escalonar os dados, mas também para aproximar a expressão funcional da CDF, de modo que ela pode ser usada para escalonar valores desconhecidos.

Observando apenas os dados, não é conhecida a forma exata da CDF de χ, sendo necessário aproximá-la pela função densidade de probabilidade cumulativa empírica (ECDF) no valor *v*, definida como:

$$\hat{P}_{\chi}(v) = \frac{número\ de\ amostras\ \leq v}{n} = \frac{1}{n}\sum_{i=1}^{n} 1_{\chi_i \leq v} \tag{A.2}$$

em que *n* é o número de amostras, χ_i é o valor de χ na *i*-ésima amostra.

Geralmente a ECDF não pode ser representada por uma expressão fechada e, devido às variações aleatórias dos dados, ela se apresenta muito irregular. Para superar isso, a função logística generalizada (GL) $L(\chi)$ é usada para aproximar a ECDF de

qualquer conjunto de dados, não importando sua distribuição, conforme a definição:

$$L(\chi) = \frac{1}{(1 + Qe^{-B(x-M)})^{\frac{1}{v}}} \tag{A.3}$$

Essa função GL é flexível para aproximar uma variedade de distribuições. Uma de suas propriedades é que ela mapeia valores no intervalo (∞,-∞) para intervalo (0,1), tornando o algoritmo GL robusto para valores atípicos. Já os parâmetros *Q*, *B*, *M* e *v* devem ser calculados com base nos dados para que a função GL se ajuste da melhor forma à ECDF. Seja η a soma das diferenças ao quadrado entre a função GL e a ECDF:

$$\eta = \sum_{i=1}^{n} \left\| L(\chi_i) - \hat{P}_X(\chi_i) \right\|^2 \tag{A.4}$$

Como o melhor conjunto de parâmetros para $L(\chi)$ deve minimizar η, a função GL mais apropriada para aproximar a ECDF pode ser encontrada resolvendo o seguinte problema de otimização:

$$\underset{B,M,Q,v}{minimizar}\ \eta(B, M, Q, v) \tag{A.5}$$

Como as Equações (A.3) e (A.4) são diferenciáveis, as derivadas parciais de η em relação aos parâmetros são imediatas e um mínimo local pode ser encontrado por meio de algoritmos de otimização de descida de gradiente. Para tanto, os valores iniciais dos parâmetros Q_0, B_0, M_0 e v_0 devem ser cuidadosamente escolhidos, conforme as seguintes expressões:

$$M_0 = \tilde{\chi} \tag{A.6}$$

$$v_0 = \log_2(1 + Q_0) \tag{A.7}$$

$$B_0 = \frac{\ln\left[(1 + Q_0)^{\log_2 10} - 1\right] - \ln Q_0}{\tilde{\chi} - \chi_{min}} \tag{A.8}$$

$$\frac{1}{1 + Q_0 e^{\{\ln\left[(1+Q_0)^{\log_2 10} - 1\right] - \ln Q_0\} \frac{\chi_{max} - \tilde{\chi}}{\chi_{min} - \tilde{\chi}}}} = 0{,}9^{\log_2(1+Q_0)} \tag{A.9}$$

Em que $\tilde{\chi}$ é a mediana dos dados, χ_{min} é o valor mínimo e χ_{max} é o valor máximo. O valor de Q_0 deve ser obtido pela resolução numérica da Equação (A.9), com o qual se calculam B_0 e v_0. Com essa inicialização de valores, é possível encontrar uma função GL que aproxime bem uma ECDF.

B – MÁQUINA DE VETOR SUPORTE (SVM)

Neste apêndice é realizada uma breve revisão da teoria e da formulação matemática do algoritmo SVM. A SVM é uma categoria de rede neural de alimentação progressiva (*feedforward*) que, em termos gerais, é uma máquina binária de aprendizado com algumas propriedades altamente elegantes (Haykin, 2003). Num contexto de padrões separáveis, a ideia principal por trás da SVM pode ser resumida da seguinte forma: considerando uma amostra de treinamento, a SVM constrói um *hiperplano* para ser uma superfície de decisão, de tal forma que a margem de separação entre os exemplos positivos e negativos é maximizada. Essa ideia básica é estendida como um princípio para lidar com o caso mais difícil de padrões não linearmente separáveis.

Uma noção central para a SVM é o kernel de produto interno entre um vetor suporte $\boldsymbol{\chi}^s$ e um vetor $\boldsymbol{\chi}$ retirado do espaço de dados de entrada. Um ponto importante é que os vetores suporte são um pequeno subconjunto dos dados de entrada, extraídos do próprio conjunto de treinamento, pelo algoritmo. Essa propriedade central explica a nomenclatura do método de aprendizado da SVM como um método de kernel. Tal método usado para projetar uma SVM é ótimo e baseado em otimização convexa. Dessa forma, o grande impacto do uso da SVM é na solução de problemas difíceis de classificação de padrões.

Um problema clássico para o estudo da SVM são os padrões linearmente separáveis. Considere as amostras de treinamento $\{\boldsymbol{\chi}_i, y_i\}_{i=1}^{N_{TR}}$, em que $\boldsymbol{\chi}_i$ é a i-ésima amostra e y_i é a sua classe correspondente. Como as classes representadas por $y_i = +1$ e $y_i = -1$ são linearmente separáveis, a equação de uma superfície de decisão na forma de um hiperplano de separação é dada por:

$$\boldsymbol{w}^T\boldsymbol{\chi} + b = 0 \tag{B.1}$$

em que $\boldsymbol{w}$ é um vetor de pesos ajustável e b é um *bias* (valor de deslocamento). Para os pontos que não pertencem ao hiperplano, é possível escrever:

$$\begin{aligned} \boldsymbol{w}^T\boldsymbol{\chi}_i + b \geq 0 \ \text{ para } \ y_i = +1 \\ \boldsymbol{w}^T\boldsymbol{\chi}_i + b < 0 \ \text{ para } \ y_i = -1 \end{aligned} \tag{B.2}$$

A condição de padrões linearmente separáveis é assumida apenas para explicar a ideia básica por trás da SVM, porém essa condição pode ser relaxada.

Para um vetor $\boldsymbol{w}$ e um *bias* b, a separação entre o hiperplano definido na Equação (B.1**)** e o ponto mais próximo é chamada de margem de separação, denominada por ρ. O objetivo da SVM é achar um hiperplano tal que ρ seja maximizada, o qual é denominado

hiperplano ótimo. Sejam $\boldsymbol{w_o}$ e um *bias* b_o os valores de $\boldsymbol{w}$ e b para esse hiperplano ótimo que se deseja encontrar. O par ($\boldsymbol{w_o}$, b_o) deve satisfazer a seguinte condição:

$$\begin{aligned} \boldsymbol{w}_o^T \boldsymbol{\chi}_i + b_o \geq 1 \ \text{ para } \ y_i = +1 \\ \boldsymbol{w}_o^T \boldsymbol{\chi}_i + b_o \leq 1 \ \text{ para } \ y_i = -1 \end{aligned} \tag{B.3}$$

Os pontos $(\boldsymbol{\chi}_i, y_i)$ que satisfazem a Equação (B.3) com o sinal de igualdade são chamados de vetores suporte – daí o nome máquina de vetor suporte. Os demais exemplos no conjunto de treinamento são completamente irrelevantes. Os vetores suporte são os pontos mais próximos do hiperplano ótimo e são os mais difíceis de classificar. A margem se separação ρ é dada por:

$$\rho = \frac{2}{\|w_o\|} \tag{B.4}$$

Dessa forma, maximizar a margem de separação entre duas classes equivale a minimizar a norma euclidiana do vetor de pesos, o que torna único o hiperplano ótimo.

Para se encontrar o hiperplano ótimo, pode ser usada a otimização quadrática sob a teoria da otimização convexa. Pela combinação da Equação (B.3) e a definição da função custo convexa $\boldsymbol{\Phi(w)}$, para um conjunto de treinamento $\{\boldsymbol{\chi}_i, y_i\}_{i=1}^{N_{TR}}$ é possível definir um problema de otimização restrita chamado de problema *primal* da seguinte forma:

Achar os valores ótimos do vetor de pesos $\boldsymbol{w}$ *e do bias b tais que satisfaçam a restrição* $y_i(\boldsymbol{w}^T \boldsymbol{\chi}_i + b) \geq 1$ *e que o* $\boldsymbol{w}$ *minimize* $\boldsymbol{\Phi(w)} = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$

Tal problema pode ser resolvido pelo método dos multiplicadores de Lagrange, definidos como α_i. É possível, partindo do problema primal, construir um problema *dual* com os multiplicadores de Lagrange fornecendo a solução ótima, conforme a seguir:

Achar os multiplicadores de Lagrange $\{\alpha_i\}_{i=1}^{N_{TR}}$ *que maximizam a função objetivo*

$$Q(\alpha) = \sum_{i=1}^{N_{TR}} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_{TR}} \sum_{j=1}^{N_{TR}} \alpha_i \alpha_j y_i y_j \boldsymbol{\chi}_i^T \boldsymbol{\chi}_j \tag{B.5}$$

sujeita às seguintes restrições: (1) $\sum_{i=1}^{N_{TR}} \alpha_i \, y_i = 0$ *e (2)* $\alpha_i \geq 0$

A função objetivo $Q(\alpha)$ depende apenas das amostras de entrada na forma de produtos internos $\boldsymbol{\chi}_i^T \boldsymbol{\chi}_j$. Os vetores suporte, como um subconjunto do conjunto de treinamento, satisfazem a desigualdade na restrição (2) e os demais vetores satisfazem a igualdade (α_i=0). Após determinar os multiplicadores ótimos $\alpha_{o,i}$, o vetor $\boldsymbol{w_o}$ e o *bias* b_o ótimos podem ser calculados pelas seguintes expressões, definindo o hiperplano ótimo para

os padrões linearmente separáveis com N_s vetores suporte:

$$\boldsymbol{w}_o = \sum_{i=1}^{N_s} \alpha_{o,i} y_i \boldsymbol{\chi}_i \tag{B.6}$$

$$b_o = 1 - \sum_{i=1}^{N_s} \alpha_{o,i} y_i \boldsymbol{\chi}_i^T \boldsymbol{\chi}^s \tag{B.7}$$

Quando os padrões não são linearmente separáveis, não é possível construir um hiperplano de separação sem haver erros de classificação. No entanto, é útil achar um hiperplano ótimo tal que a probabilidade de erros de classificação seja minimizada. Nesse caso, a margem de separação é chamada de suave (*soft*) quando a amostra cai dentro da região de separação, mas no lado correto, ou cai no lado errado do hiperplano. Para definir melhor essa situação, as variáveis de folga (*slack*) $\{\xi_i\}_{i=1}^{N_{TR}}$ medem o desvio das amostras da posição ideal para separação. Da mesma maneira que no caso de padrões linearmente separáveis, uma função custo $\boldsymbol{\Phi}(\mathbf{w})$ deve ser minimizada, porém considerando as folgas ξ_i. Essa nova função é expressa por:

$$\boldsymbol{\Phi}(\mathbf{w}, \xi) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N_{TR}} \xi_i \tag{B.8}$$

O parâmetro *C* controla o equilíbrio entre a complexidade do algoritmo e o número de pontos não separáveis e pode ser visto como o recíproco de um parâmetro de regularização. Quando um grande valor é estipulado para *C*, o conjunto de treinamento é considerado altamente confiável, porém, se é estipulado um valor pequeno, esse conjunto é considerado ruidoso e impreciso. Como *C* deve ser escolhido pelo usuário, ele pode ser determinado por alguns métodos, como a validação cruzada para seleção ótima do parâmetro de regularização 1/*C*. A seleção de parâmetros da SVM é tratada na próxima seção deste trabalho.

Usando o método dos multiplicadores de Lagrange e procedendo de maneira similar ao caso de padrões linearmente separáveis, o problema dual para padrões não separáveis pode ser enunciado como:

Achar os multiplicadores de Lagrange $\{\boldsymbol{\alpha}_i\}_{i=1}^{N_{TR}}$ *que maximizam a função objetivo*

$$Q(\alpha) = \sum_{i=1}^{N_{TR}} \alpha_i - \frac{1}{2}\sum_{i=1}^{N_{TR}}\sum_{j=1}^{N_{TR}} \alpha_i \alpha_j y_i y_j \boldsymbol{\chi}_i^T \boldsymbol{\chi}_j \tag{B.9}$$

sujeita às seguintes restrições: (1) $\sum_{i=1}^{N_{TR}} \alpha_i y_i = 0$ *e (2)* $0 \leq \alpha_i \leq C$

em que C é um parâmetro positivo especificado pelo usuário.

A função objetivo $Q(\alpha)$ é a mesma para o caso de padrões linearmente separáveis, mas a restrição para os valores de α_i é maior, enquanto o cálculo de $\boldsymbol{w_o}$, b_o e dos vetores suporte é o mesmo.

A construção de uma SVM apta para a tarefa de reconhecimento de padrões é baseada em duas operações principais: mapeamento não linear dos vetores de entrada para um espaço de características de maior dimensão que é escondido da entrada e da saída; e a construção de hiperplano ótimo para separar tais características. Como o número de características do espaço escondido é determinado pelo número de vetores suporte, a teoria da SVM permite determinar o tamanho ótimo do espaço de características.

A SVM também é considerada uma máquina de *kernel* de produto interno. Seja a função não linear $\boldsymbol{\phi(\chi)}$ que transforma o vetor $\boldsymbol{\chi}$ do espaço de entrada em um vetor de maior dimensão no espaço das características. Dessa forma, $\boldsymbol{\phi(\chi) = [\varphi_1(\chi), \varphi_2(\chi), \ldots]^T}$, em que φ_i são funções escalares. O hiperplano de decisão no espaço de saída é dado por:

$$\sum_{i=1}^{N_s} \alpha_i y_i \boldsymbol{\phi}^T(\boldsymbol{\chi_i})\boldsymbol{\phi(\chi)} = 0 \tag{B.10}$$

em que $\boldsymbol{\chi_i}$ são os vetores suporte. O termo $\boldsymbol{\phi}^T(\boldsymbol{\chi_i})\boldsymbol{\phi(\chi)}$ representa um produto interno, representado doravante por $K(\boldsymbol{\chi}, \boldsymbol{\chi_i})$ e chamado de *kernel* de produto interno. Logo é possível afirmar que:

$$K(\boldsymbol{\chi}, \boldsymbol{\chi_i}) = \boldsymbol{\phi}^T(\boldsymbol{\chi_i})\boldsymbol{\phi(\chi)} = \sum_{j=1}^{\infty} \varphi_j(\boldsymbol{\chi_i})\varphi_j(\boldsymbol{\chi}), \;\; i = 1,2..,N_s \tag{B.11}$$

É suficiente especificar o kernel para a classificação de padrões, não sendo necessário calcular o vetor de pesos $\boldsymbol{w_o}$ nem os valores de *bias*. Por isso a aplicação da Equação (B.11) é chamada de *truque de kernel* e a SVM é chamada também de máquina de kernel. A expansão dessa equação permite conceber uma superfície de decisão não linear no espaço da entrada cuja imagem no espaço das características é linear. Observa-se que não é necessário a priori conhecer a função não linear $\boldsymbol{\phi(\chi)}$, o que seria uma tarefa muito difícil, sendo suficiente utilizar uma função de kernel conhecida que obedeça a determinadas condições. Essas propriedades permitem estabelecer o problema dual da otimização restrita da SVM da seguinte forma:

Dadas as amostras de treinamento $\{\boldsymbol{\chi}_i, y_i\}_{i=1}^{N_{TR}}$, a*char os multiplicadores de Lagrange* $\{\boldsymbol{\alpha}_i\}_{i=1}^{N_{TR}}$ *que maximizam a função objetivo*

$$Q(\alpha) = \sum_{i=1}^{N_{TR}} \alpha_i - \frac{1}{2} \sum_{i=1}^{N_{TR}} \sum_{j=1}^{N_{TR}} \alpha_i \alpha_j y_i y_j \, K(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) \tag{B.12}$$

sujeita às seguintes restrições: (1) $\sum_{i=1}^{N_{TR}} \alpha_i \, y_i = 0$ *e (2)* $0 \leq \alpha_i \leq C$

em que C é um parâmetro positivo especificado pelo usuário.

A escolha da função de kernel é relativamente livre e depende da natureza do conjunto de treinamento e do problema específico. Os tipos de kernel $K(\boldsymbol{\chi}, \boldsymbol{\chi}_i)$ para SVM mais comuns são:

- Linear: $\boldsymbol{\chi}^T \boldsymbol{\chi}_i$
- Polinominal: $(\boldsymbol{\chi}^T \boldsymbol{\chi}_i + 1)^p$, p definido pelo usuário.
- RBF: $e^{\left(-\gamma \|\boldsymbol{\chi} - \boldsymbol{\chi}_i\|^2\right)}$, γ definido pelo usuário.
- Sigmoide: tanh $(\gamma \boldsymbol{\chi}^T \boldsymbol{\chi}_i + r)$, γ e r definidos pelo usuário.

Na SVM com kernel RBF, o número de funções RBF e seus centros são determinados automaticamente pelo número de vetores suporte e seus valores, respectivamente. Já para todas as funções de kernel, a dimensionalidade do espaço das características é determinada pelo número de vetores suporte extraídos do conjunto de treinamento, dado pela solução do problema dual. Essa solução evita o uso de heurística no projeto da SVM, situação comum na concepção de redes RBF e perceptrons multicamada.

A arquitetura de uma SVM em geral possui uma camada escondida de dimensão N_s. A dimensionalidade de camada escondida é propositalmente mantida alta para permitir a construção de um hiperplano ótimo no espaço das características. Para conseguir um bom desempenho de generalização, a complexidade do modelo é controlada pela imposição de certas restrições na construção do hiperplano com margem de separação suave, o que resulta na extração de uma fração das amostras de treinamento para serem vetores suporte.

C – ELIMINAÇÃO RECURSIVA DE CARACTERÍSTICAS COM REDUÇÃO DE POLARIZAÇÃO POR CORRELAÇÃO

O método SVM-RFE é um algoritmo do tipo embutido, ou seja, ele usa um critério formulado a partir dos coeficientes e vetores suporte dos modelos SVM para avaliar as características e remover recursivamente aquelas com menores valores do critério. Ele também é um método de eliminação regressivo, pois é capaz de modelar as dependências entre as características, além de não usar a acurácia de validação cruzada sobre a matriz de treinamento como um critério de seleção. Essas propriedades trazem algumas vantagens, como menor propensão ao sobreajuste, uso total do conjunto de treinamento e maior velocidade, especialmente quanto o número de características é elevado. O método SVM-RFE tem sido usado com sucesso em muitos problemas, como seleção de genes e sensores de gás (Yan e Zhang, 2015).

Contudo, quando algumas das características são altamente correlacionadas, o seu critério de avaliação é influenciado e as importâncias das características serão subestimadas mesmo se tiverem alta relevância. Esse fenômeno afeta vários algoritmos de seleção de características, dentre eles o SVM-RFE, trazendo estimativas erradas da importância das características. Ocorre o que Yan e Zhang (2015) chamaram de polarização por correlação, que, por sua vez, pode ser corrigida pela CBR, aumentando as acurácias na seleção de características. O algoritmo de CBR, portanto, pode ser usado para melhorar a classificação das características quando elas tiverem alta correlação.

O método SVM-RFE CBR foi avaliado para aplicações de sensores de gás para fins biomédicos em que o número de características era elevado (mais de 1000) e as características em si eram altamente correlacionadas (Yan e Zhang, 2015). Esse cenário se mostra compatível com a detecção de compressão dupla AMR em que existem 657 características, algumas correlacionadas conforme exemplificado anteriormente.

Existem dois tipos de algoritmos SVM-RFE, dependendo do modelo usado de SVM: linear e não linear. Quando o número de características é muito maior do que o número de amostras, o algoritmo de SVM-RFE linear é mais adequado para evitar o sobreajuste. Contudo, em outras aplicações em que o número de amostras é maior, é esperado que o algoritmo SVM-RFE não linear supere o SVM-RFE linear porque ele pode se ajustar às amostras com menos polarização na predição (Yan e Zhang, 2015). Dessa forma, para a detecção de compressão dupla AMR com 6300 amostras (*corpus* TIMIT) e 657 características, o algoritmo SVM-RFE não linear é o mais adequado.

Seja um conjunto de treinamento de características AMR $\{\boldsymbol{\chi}_i, y_i\}_{i=1}^{N_{TR}}$, $\boldsymbol{\chi}_i \in \mathbb{R}^{NCF}$, $y_i \in \{-1,1\}$, em que N_{TR} é o número de amostras de treinamento. Uma SVM não linear mapeia as características num novo espaço de maior dimensão *h*:

$$\boldsymbol{\chi} \in \mathbb{R}^{NCF} \mapsto \boldsymbol{\phi}(\boldsymbol{\chi}) \in \mathbb{R}^{h}, \tag{C.1}$$

em que $\boldsymbol{\phi}(.)$ é a função não linear que mapeia as características no novo espaço. Nesse novo espaço, as amostras devem ser linearmente separáveis. A forma dual da função objetivo (formulação de Lagrange) para o problema de separação de duas classes pode ser escrita como:

$$Q(\alpha) = \sum_{i=1}^{N_{TR}} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{N_{TR}} \alpha_i \alpha_j y_i y_j \, \boldsymbol{\phi}(\boldsymbol{\chi}_i) \cdot \boldsymbol{\phi}(\boldsymbol{\chi}_j), \tag{C.2}$$

onde α_i são os multiplicadores de Lagrange e $\boldsymbol{\phi}(\boldsymbol{\chi}_i) \cdot \boldsymbol{\phi}(\boldsymbol{\chi}_j)$ denomina o produto interno entre os vetores $\boldsymbol{\phi}(.)$. Notar que a única forma em que $\boldsymbol{\phi}(.)$ é envolvida no algoritmo de treinamento é pelo produto interno. Sendo assim, é possível substituir $\boldsymbol{\phi}(\boldsymbol{\chi}_i) \cdot \boldsymbol{\phi}(\boldsymbol{\chi}_j)$ por uma função de núcleo $K(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)$ sem saber explicitamente a forma de $\boldsymbol{\phi}(.)$. Essa ideia é muito útil porque é difícil determinar a forma de $\boldsymbol{\phi}(.)$ em aplicações reais. Uma escolha muito comum para a função $K(.)$ é o núcleo gaussiano.

O critério de seleção de características do algoritmo SVM-RFE pode ser formulado da seguinte forma: se a eliminação de uma característica causa apenas pequenas mudanças na função objetivo (C.2), a característica deve ser removida. Essa estratégia leva ao seguinte critério de classificação da característica *k*:

$$J(k) = \frac{1}{2}\sum_{i,j=1}^{N_{TR}} \alpha_i \, \alpha_j y_i y_j K(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) - \frac{1}{2}\sum_{i,j=1}^{N_{TR}} \alpha_i \alpha_j y_i y_j \, K\left(\boldsymbol{\chi}_i^{(-k)}, \boldsymbol{\chi}_j^{(-k)}\right) \tag{C.3}$$

em que $(-k)$ significa que a característica *k* foi removida. As características com os menores *J* serão eliminadas a cada iteração do algoritmo. Ao final da iteração, a característica que sobra será a primeira mais importante conforme o critério *J*. Essa característica é retirada do conjunto para a próxima iteração e o procedimento se repete até ser encontrada a segunda característica mais importante, e assim sucessivamente.

Para que haja uma eficiência razoável, a implementação do algoritmo SVM-RFE não remove uma característica por vez quando a dimensão das características é muito grande, e sim processa as características em lotes. Inicialmente a metade das características é removida em cada iteração. Quando número das características restantes é menor do que um limiar, elas são removidas uma a uma para uma melhor precisão.

No algoritmo SVM RFE-CBR, conforme se observa na Equação (C.3), o cálculo do critério *J* depende dos parâmetros α_i, y_i e dos valores das funções de núcleo $K(\chi_i, \chi_j)$, considerada neste trabalho como o núcleo Gaussiano $K(\chi_i, \chi_j) = e^{-\gamma\left\|\chi_i - \chi_j\right\|^2}$. No início do algoritmo, a SVM é treinada com todas as características e um modelo é gerado. Os valores do parâmetro de penalidade C e a constante γ da SVM são ajustáveis pelo usuário no algoritmo SVM-RFE (são usados como padrão C=1 e $\gamma=2^{-6}$). Do modelo gerado são extraídos os valores de α_i não nulos e os vetores suporte: dessa forma, é possível calcular os valores de *J*(*k*) para cada uma das características, pois os valores de $K(\chi_i, \chi_j)$ são calculados usando os vetores suporte. Após ordenar as características de acordo com os *J*(*k*), a característica de menor *J* é retirada e colocada no conjunto das eliminadas. Na próxima iteração, a SVM é novamente treinada com o conjunto das características que ainda não foram eliminadas e que têm os maiores valores de *J*. Ao final, a última característica que sobrar será a primeira da classificação de características. Essa classificação, expressa por um vetor de ordenamento, é definida neste trabalho como F_{Rank}.

Um método bem conhecido para lidar com características correlacionadas é substituir cada grupo delas por um representante antes da seleção. Outra forma bastante utilizada é fazer a seleção por grupos de características ao invés de características isoladas. O algoritmo de CBR proposto por Yan e Zhang (2015), por sua vez, faz uso do procedimento de SVM-RFE para reduzir a influência trazida pela polarização de correlação. Quando um lote de características é removido numa iteração, um grupo de características correlacionadas pode ser removido por inteiro. Isso pode acontecer porque as características são realmente irrelevantes ou porque o critério de classificação delas foi incorretamente subestimado. Em ambas as situações, é possível mover um representante do grupo de volta para a lista de características não eliminadas. Então, dessa forma, esse representante pode ser avaliado novamente na próxima iteração sem a influência da polarização por correlação. O representante pode ser escolhido, inclusive, como a característica com o maior critério de seleção nessa iteração. Essa estratégia não muda o conjunto de características candidatas ou o critério da classificação, mas monitora e corrige as decisões potencialmente erradas causadas pela polarização por correlação. O propósito do algoritmo CBR é mover características provavelmente úteis do conjunto daquelas eliminadas para o conjunto daquelas ainda em avaliação. Foi comprovado que o algoritmo CBR melhora o desempenho do método SVM-RFE (Yan e Zhang, 2015).

D – RESULTADOS DAS SIMULAÇÕES COMPUTACIONAIS

Este apêndice contem os resultados completos das simulações computacionais realizadas nos experimentos desta tese, acompanhados das condições desses experimentos. São apresentados os resultados para o *corpus* TIMIT original, sem qualquer inserção de ruído ou outra condição adversa, e os resultados da análise de robustez.

D.1 – *CORPUS* TIMIT

As acurácias mínimas, medias e máximas obtidas nos 20 experimentos para os modelos SVM, para todas as oito taxas de bits BR2, para os conjuntos comprimidos S_{BmB2} e com *corpus* TIMIT original pode ser verificada na Tabela D-1.

Tabela D-1 – Acurácias mínimas, medias e máximas de 20 experimentos (conjuntos S_{BmB2} com *corpus* TIMIT).

BR2 (kbits/s)	Acurácias								
	Acc			$\overline{TP}$			$\overline{TN}$		
	Min	Méd	Max	Min	Méd	Max	Min	Méd	Max
4,75	98,84	**99,35**	99,79	97,67	**99,35**	99,79	97,88	**99,35**	100,00
5,15	98,68	**99,21**	99,66	98,20	**99,33**	99,84	98,04	**99,10**	99,95
5,90	98,41	**99,16**	99,50	97,30	**98,78**	99,42	98,84	**99,54**	100,00
6,70	98,65	**99,12**	99,63	97,72	**98,74**	99,58	98,25	**99,49**	99,95
7,40	98,12	**99,01**	99,76	96,30	**98,53**	99,68	98,41	**99,48**	99,95
7,95	98,49	**99,07**	99,58	97,09	**98,49**	99,63	99,10	**99,64**	99,89
10,2	98,81	**99,19**	99,50	97,72	**98,88**	99,74	98,94	**99,50**	99,89
12,2	98,89	**99,18**	99,47	98,47	**98,98**	99,47	98,78	**99,38**	99,79

Os valores médios, mínimos e máximos das 64 acurácias calculadas para 20 experimentos dos conjuntos comprimidos S_{B1B2} com o *corpus* TIMIT estão dispostos na Tabela D-2. Nesses experimentos, os modelos SVM para BR2 utilizados foram os mesmos calculados para os conjuntos S_{BmB2}.

Tabela D-2 – Acurácias mínimas, medias e máximas de 20 experimentos para os modelos SVM para BR2 (conjuntos S_{B1B2} com *corpus* TIMIT)

BR2 (kbits/s)		BR1 (kbits/s)																							
		4,75			5,15			5,90			6,70			7,40			7,95			10,2			12,2		
		min	méd	max	min	méd	max	min	méd	max	min	méd	max	min	méd	max	min	méd	max	min	méd	max	min	méd	max
4,75	Acc	99,02	**99,56**	99,95	98,39	**99,50**	99,87	98,68	**99,33**	99,79	99,02	**99,45**	99,81	99,10	**99,55**	99,81	98,62	**99,48**	99,89	96,93	**98,35**	99,10	98,33	**98,87**	99,31
	$\overline{TP}$	98,25	**99,27**	99,89	96,83	**99,13**	99,89	98,89	**99,72**	99,95	98,52	**99,54**	99,95	98,25	**99,31**	99,79	97,30	**99,10**	99,79	98,84	**99,41**	99,68	98,36	**99,20**	99,84
	$\overline{TN}$	99,58	**99,85**	100,00	99,58	**99,86**	100,00	97,57	**98,94**	99,89	98,31	**99,36**	99,95	99,21	**99,80**	100,00	99,58	**99,86**	100,00	94,29	**97,29**	99,10	96,83	**98,54**	99,89
5,15	Acc	99,39	**99,69**	99,89	98,62	**99,49**	99,81	98,44	**99,12**	99,66	98,73	**99,34**	99,79	98,65	**99,49**	99,81	98,81	**99,43**	99,68	96,96	**98,19**	99,34	97,96	**98,75**	99,39
	$\overline{TP}$	98,89	**99,53**	99,84	97,35	**99,16**	99,84	98,89	**99,65**	99,89	98,25	**99,55**	99,79	97,41	**99,25**	99,79	97,72	**99,13**	99,74	98,94	**99,36**	99,79	97,51	**99,20**	99,74
	$\overline{TN}$	99,63	**99,84**	100,00	99,47	**99,83**	100,00	97,35	**98,58**	99,89	97,99	**99,12**	99,95	99,21	**99,72**	100,00	99,21	**99,73**	99,95	94,39	**97,01**	99,74	96,46	**98,30**	99,89
5,90	Acc	99,02	**99,45**	99,84	98,54	**99,38**	99,87	99,02	**99,40**	99,68	98,39	**99,25**	99,63	98,33	**99,14**	99,60	97,94	**99,00**	99,50	98,23	**98,74**	99,52	98,25	**99,05**	99,71
	$\overline{TP}$	98,04	**99,34**	99,95	97,09	**99,18**	99,89	98,57	**99,53**	99,89	96,77	**98,96**	99,79	96,72	**98,71**	99,58	96,35	**98,42**	99,26	98,04	**98,97**	99,52	97,09	**98,87**	99,58
	$\overline{TN}$	98,84	**99,56**	100,00	98,84	**99,57**	100,00	98,36	**99,28**	100,00	98,84	**99,55**	100,00	98,89	**99,58**	100,00	98,89	**99,59**	100,00	97,30	**98,52**	99,58	98,15	**99,23**	100,00
6,70	Acc	98,89	**99,40**	99,71	98,76	**99,34**	99,76	98,65	**99,34**	99,81	98,76	**99,22**	99,63	98,52	**99,09**	99,60	98,04	**98,94**	99,55	98,02	**98,67**	99,15	98,47	**99,06**	99,52
	$\overline{TP}$	97,83	**99,30**	99,84	97,57	**99,16**	99,84	98,41	**99,46**	99,89	97,57	**98,94**	99,68	97,14	**98,63**	99,68	96,14	**98,33**	99,47	98,25	**98,93**	99,47	97,83	**98,92**	99,31
	$\overline{TN}$	98,41	**99,51**	99,95	98,41	**99,51**	99,95	97,62	**99,22**	99,95	98,47	**99,51**	99,95	98,47	**99,54**	99,95	98,47	**99,56**	99,95	97,20	**98,42**	99,21	97,78	**99,20**	99,89
7,40	Acc	98,20	**99,37**	99,71	97,78	**99,24**	99,76	98,81	**99,39**	99,74	97,94	**99,09**	99,58	97,75	**99,04**	99,55	97,35	**98,89**	99,47	98,23	**98,77**	99,47	98,49	**99,05**	99,58
	$\overline{TP}$	96,46	**99,25**	99,84	95,61	**98,99**	99,79	97,67	**99,45**	99,89	95,93	**98,68**	99,63	95,56	**98,57**	99,52	94,76	**98,24**	99,37	98,36	**99,02**	99,47	97,14	**98,83**	99,68
	$\overline{TN}$	98,36	**99,49**	99,95	98,41	**99,49**	99,95	98,36	**99,33**	99,95	98,47	**99,51**	99,95	98,47	**99,51**	99,95	98,52	**99,53**	99,95	97,09	**98,52**	99,95	97,72	**99,27**	99,95
7,95	Acc	98,68	**99,34**	99,81	98,44	**99,30**	99,66	99,07	**99,46**	99,68	98,17	**99,03**	99,58	97,91	**98,86**	99,50	97,72	**98,72**	99,44	96,96	**98,21**	99,15	98,39	**99,24**	99,71
	$\overline{TP}$	97,46	**99,05**	99,95	96,98	**98,93**	99,74	98,31	**99,32**	99,84	96,46	**98,45**	99,58	95,93	**98,11**	99,52	95,56	**97,82**	99,52	99,10	**99,49**	99,84	99,15	**99,47**	99,89
	$\overline{TN}$	98,99	**99,63**	99,89	98,99	**99,67**	99,95	98,68	**99,60**	99,95	98,84	**99,61**	99,89	98,94	**99,61**	99,89	98,94	**99,61**	99,89	94,81	**96,93**	98,57	97,57	**99,01**	99,89
10,2	Acc	98,70	**99,40**	99,71	98,92	**99,41**	99,74	99,15	**99,50**	99,81	98,60	**99,18**	99,58	98,39	**99,04**	99,66	98,25	**98,91**	99,47	98,17	**98,95**	99,68	98,65	**99,15**	99,68
	$\overline{TP}$	97,62	**99,27**	99,84	98,04	**99,30**	99,95	98,78	**99,57**	99,89	97,30	**98,83**	99,89	96,98	**98,51**	99,74	96,56	**98,24**	99,37	98,84	**99,22**	99,63	97,51	**98,98**	99,68
	$\overline{TN}$	98,99	**99,52**	99,89	98,99	**99,52**	99,89	98,47	**99,42**	99,89	98,99	**99,53**	99,89	99,05	**99,56**	99,95	99,05	**99,58**	99,95	96,83	**98,68**	99,84	97,94	**99,32**	99,95
12,2	Acc	96,75	**99,51**	99,84	99,42	**99,68**	99,89	99,10	**99,38**	99,74	91,59	**99,06**	99,71	99,21	**99,40**	99,66	99,07	**99,42**	99,68	98,02	**98,84**	99,23	98,62	**99,06**	99,60
	$\overline{TP}$	93,70	**99,56**	100,00	99,52	**99,85**	100,00	99,21	**99,55**	99,89	83,65	**98,69**	99,84	98,78	**99,30**	99,58	98,31	**99,30**	99,68	98,47	**99,30**	99,63	97,83	**98,87**	99,63
	$\overline{TN}$	98,78	**99,45**	99,84	98,94	**99,51**	99,84	98,47	**99,20**	99,74	98,73	**99,43**	99,79	98,99	**99,51**	99,84	98,94	**99,54**	99,84	96,77	**98,37**	99,58	98,47	**99,26**	99,68

D.2 – ANÁLISE DE ROBUSTEZ

Na Tabela D-3 podem ser vistos os resultados detalhados das acurácias obtidas de seis *corpora* formados por arquivos de mesma duração elaborados a partir do *corpus* TIMIT.

Tabela D-3 – Acurácias em % para os modelos SVM para cada BR2 usando os *corpora* de arquivos com durações variáveis (conjuntos S_{BmB2} gerados a partir do *corpus* TIMIT, durações em segundos)

BR2 (kbits/s)	Durações dos Arquivos					
	0,125	0,250	0,500	1	2	3
4,75	92,25	97,22	97,30	98,68	99,57	99,24
5,15	90,26	96,43	96,72	97,64	99,60	98,99
5,90	92,62	98,20	98,52	98,46	99,28	99,35
6,70	91,35	94,36	98,44	98,91	99,34	99,48
7,40	90,63	97,33	98,60	98,54	98,71	99,08
7,95	99,23	97,72	98,62	98,81	99,28	99,32
10,2	99,04	97,62	98,57	98,86	99,40	99,18
12,2	99,23	97,46	97,86	98,57	98,57	98,75
Média	**94,32**	**97,04**	**98,08**	**98,56**	**99,22**	**99,17**

Na Tabela D-4 podem ser vistos os resultados detalhados das acurácias obtidas de seis *corpora* formados por arquivos em que foram inseridos deslocamentos de quadro nos arquivos do *corpus* TIMIT (conjuntos S_{BmB2}).

Tabela D-4 – Acurácias em % para os modelos SVM para cada BR2 após o ataque de descolamento de quadro (conjuntos S_{BmB2} gerados pela supressão das primeiras amostras dos arquivos do *corpus* TIMIT, descolamentos em amostras).

BR2 (kbits/s)	Deslocamento de Quadro						
	25	50	75	100	125	150	Média
4,75	99,42	99,73	99,26	99,21	99,29	99,31	**99,37**
5,15	99,50	99,60	99,58	99,34	99,66	99,39	**99,51**
5,90	99,42	99,52	99,18	99,36	99,44	99,10	**99,34**
6,70	99,07	99,15	98,99	99,26	99,31	98,94	**99,12**
7,40	99,07	98,84	98,99	99,29	98,68	98,86	**98,95**
7,95	98,76	98,91	99,15	98,99	99,23	98,97	**99,00**
10,2	99,13	99,31	99,18	99,31	99,47	99,26	**99,28**
12,2	99,36	99,31	99,26	99,29	99,58	99,26	**99,34**

Na Tabela D-5 podem ser vistos os resultados detalhados das acurácias obtidas de quatro *corpora* (conjuntos S_{BmB2}) formados por arquivos do *corpus* TIMIT em que foi adicionado ruído do tipo AWGN.

Tabela D-5 – Acurácias em % para os modelos SVM para cada BR2 usando áudio com ruído adicionado (conjuntos S_{BmB2} gerados a partir do *corpus* TIMIT, SNR em decibels).

BR2 (kbits/s)	SNR			
	5	10	20	30
4,75	98,36	98,28	98,15	99,26
5,15	97,70	98,49	97,33	98,99
5,90	98,12	96,96	98,86	99,28
6,70	97,64	96,64	98,89	98,73
7,40	98,23	97,25	98,94	99,15
7,95	97,86	97,51	99,10	99,29
10,2	98,20	96,69	98,60	99,34
12,2	98,20	97,51	98,84	98,26
Média	**98,04**	**97,42**	**98,59**	**99,04**

Na Tabela D-6 podem ser vistos os resultados detalhados das acurácias obtidas com o *corpus* CARIOCA1 modificado (conjuntos S_{BmB2}) formado por arquivos gravados de chamadas de telefone fixo.

Tabela D-6 – Acurácias médias de teste (em %) e valores de NCF (número atual das características) e BNF (melhor número de características) para novos modelos SVM e para cada BR2 (conjuntos S_{BmB2} gerados a partir do *corpus* CARIOCA1 modificado, taxas em kbits/s).

BR2 (kbits/s)	Acurácias			Características	
	Acc	$\overline{TP}$	$\overline{TN}$	NCF	BNF
4,75	99,66	99,72	99,61	603	87
5,15	99,26	99,32	99,20	602	119
5,90	99,52	99,43	99,60	603	193
6,70	99,15	98,69	99,60	601	489
7,40	99,09	98,69	99,49	601	200
7,95	99,06	98,92	99,20	599	120
10,2	99,35	99,20	99,49	601	448
12,2	99,72	99,66	99,77	598	222
Média	**99,35%**	**99,20%**	**99,50%**		

Na Tabela D-7 podem ser vistos os resultados detalhados das acurácias obtidas usando áudio com ruído forense adicionado aos arquivos do *corpus* TIMIT (conjuntos S_{BmB2}). O procedimento para adição do ruído forense foi o mesmo usado para a geração dos arquivos contaminados por ruído AWGN cujos resultados estão na Tabela D-5.

Tabela D-7 – Acurácias em % para os modelos SVM para cada BR2 usando áudio com ruído forense adicionado (conjuntos S_{BmB2} gerados a partir do *corpus* TIMIT, SNR em decibels).

BR2 (kbits/s)	SNR			
	5	10	20	30
4,75	97,32	96,53	98,78	99,58
5,15	96,38	96,83	98,84	99,02
5,90	95,69	96,08	98,60	99,36
6,70	96,64	97,62	98,86	99,47
7,40	95,69	96,82	98,33	98,73
7,95	96,98	96,72	98,78	99,38
10,2	96,75	97,38	97,88	98,52
12,2	96,40	96,96	98,47	98,48
Média	**96,48**	**96,84**	**98,57**	**99,07**