BovChewing - Segmentação e classificação de eventos bioacústico do comportamento ingestivo de bovinos por meio de aprendizado de máquina

Campo Grande, abril de 2017.

Rodrigo Sanches Devigo

Dissertação apresentada à Faculdade de Computação, UFMS, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Fundação Universidade Federal de Mato Grosso do Sul
Faculdade de Computação

BovChewing - Segmentação e classificação de eventos bioacústico do comportamento ingestivo de bovinos por meio de aprendizado de máquina

Rodrigo Sanches Devigo

abril de 2017

Comissão Avaliadora:

- Cláudio Leonardo Lucchesi (Orientador)
- Marcelo Henriques de Carvalho, Faculdade de Computação, Universidade Federal de Mato Grosso do Sul (UFMS)
- Jonathan Andrade Silva, Universidade Federal de Mato Grosso do Sul (UFMS), Campus Ponta Porã
- Julio Kuhn da Trindade, Departamento de Diagnóstico e Pesquisa Agropecuária
- Fabiana Villa Alves, Embrapa Gado de Corte

Abstract

The use of bioacoustic methods for behavioral analysis has been developed in recent years, considering these are non-invasive methods for seeking an improvement in the cattle accuracy monitoring. However, the analysis of these data usually requires specialists and a lot of time, becoming a generally difficult task, in this way, it is necessary to construct ways to automate the phase of data analysis. There are studies that focus on the automation of data analysis with satisfactory results, but they use databases with recordings made under controlled conditions and some classify rumination events. This dissertation proposes the development of the BovChewing, a tool capable of detecting and classifying digestive events, based on semi-supervised methods for detection and supervised for classification, thus, performing data analysis. Experiments showed results that the tool is sufficient and satisfactory for the task, obtaining an accuracy of 63% in the segmentation task and 91% in the classification task. Future work points the improvement of the segment, integration with other bovine behavior analysis tools, and data processing in the cloud as a way to build a complete system.

Resumo

O uso de métodos da bioacústica para análises comportamentais têm se desenvolvido com intensidade nos últimos anos, por se tratar de métodos não invasivos buscando uma melhora no monitoramento de precisão de rebanhos. Entretanto, a análise desses dados geralmente requer especialistas e muito tempo tornando-se assim uma tarefa geralmente difícil, tendo a necessidade de construir maneiras de automatizar a fase da análise dos dados. Existem trabalhos que focam na automatização da análise dos dados com resultados satisfatórios, mas utilizam de base dados com gravações feitas em condições controladas e poucos classificam eventos de ruminação. Essa dissertação propõe o BovChewing, uma ferramenta capaz de detectar e classificar eventos ingestivos, baseados em métodos semi-supervisionado para a detecção e supervisionado para a classificação, realizando assim a análise dos dados. Experimentos mostraram resultados que a ferramenta é suficiente e satisfatória para com a tarefa obtendo uma acurácia de 63% na tarefa de segmentação e 91% na tarefa de classificação. Trabalhos futuros apontam para a melhora do segmentador, integração com outras ferramentas de análises de comportamentos bovinos e processamento dos dados na nuvem como uma forma de construir um sistema completo.

Dedicatória

Ao meus pais,
Antônio Anacleto Devigo e Célia Cristina Sanches Devigo,

Ao meu irmão,
Maurício Sanches Devigo,

À minha namorada,
Atemizia Janaina Costa Bazan,

Ao meus orientadores,
Prof. Dr. Cláudio Leonardo Lucchesi e Prof. Dr. Marcelo Henriques de Carvalho,

À minha família e aos sinceros e bons amigos.

Agradecimentos

Agradeço primeiramente aos meus pais Célia Cristina Sanches Devigo e Antônio Anacleto Devigo e ao meu irmão Maurício Sanches Devigo que estiveram distantes, mas sempre em minha vida fortalecendo o meu sonho de ser mestre. Nunca mediram palavras ou até mesmo gestos que só me fizeram acreditar e seguir em frente.

Quero também agradecer à minha namorada Atemizia Janaina Costa Bazan que viu o projeto nascer, e apesar dos altos e baixos do mestrado, sempre me proporcionou amizade, companheirismo, ideias, apoio e compreensão. Agradeço, imensamente, o apoio que me deste nesse processo. Você é parte integrante e essencial nesta conquista.

Agradeço à Dra. Fabiana Villa Alves, que em uma conversa, deu a ideia que fez nascer esse trabalho. Sempre entusiasta em criar softwares e aplicativos em prol da agropecuária de precisão. Nunca mediu esforços para a realização desse projeto.

Agradeço também ao Prof. Dr. Jonathan Andrade Silva, que sem o seu conhecimento e a sua disposição essa dissertação não teria sido concluída. Com seus ensinamentos sobre aprendizado de máquina e seus incentivos a realização de um sonho não seria possível.

Agradeço ao Prof. Dr. Júlio Kuhn da Trindade que se disponibilizou em ensinar os conceitos relacionados à atividades de pastejo e que sem os seus dados a construção do BovChewing não seria possível.

Agradeço ao Prof. Dr. Marcelo Henriques Carvalho, que durante a minha graduação sempre foi um exemplo de professor e sabedoria, foi pelo exemplo que decidi seguir a carreira de professor e fazer o mestrado. Ele foi responsável pela construção da ponte entre eu e o meu orientador.

Agradeço ao Prof. Dr. Cláudio Leonardo Lucchesi, meu orientador, que posso considerar um avô, ensinou-me e mostrou-me muito sobre a ciência. Mesmo com contratempos se mostrou amigo e mestre em diversos assuntos, e mesmo em meus momentos de perca de foco me trouxe de volta para o caminho. Como o professor Marcelo, você também é minha inspiração como docente hoje. Lembro-me de nossas reuniões repletas de histórias e causos de épocas pioneiras da computação, guardo elas em meu coração e em minha memória.

Agradeço ao Prof. Dr. Edson Takashi Matsubara, que me ajudou em momentos de

dúvidas sobre Inteligência Artificial (IA) e mostrou como é apaixonante essa área. E agradeço a todos os meus amigos e companheiros, que sempre estiveram dando apoio, seria preciso um livro apenas para nomeá-los.

Sumário

Lista de Tabelas

Lista de Figuras

BovChewing - Segmentação e classificação de eventos bioacústico do comportamento ingestivo de bovinos por meio de aprendizado de máquina

Rodrigo Sanches Devigo

abril de 2017

Capítulo 1

Introdução

1.1 Contextualização

Para serem eficientes em seu forrageamento, os herbívoros desenvolveram, ao longo da evolução, estratégias visando a otimizar o uso do tempo na busca e na colheita do alimento. Essas estratégias lhes permitem selecionar e colher, de forma geral, dietas de qualidade superior à média existente no pasto. Para tanto, os animais são capazes de alterar os padrões de deslocamento, preferência e ingestão em resposta às variações na estrutura das plantas, composição química e oferta de forragem (OF)[40]. Pode-se construir um modelo conceitual sobre processo de pastejo segundo Senft et al. (1987)[43] em uma maneira hierarquia em uma escala espaçotemporal como na Figura 1.1 determinando o desempenho dos animais. Dele depreende-se que os estudos de consumo de forragem pelos animais em pastejo devam levar em consideração a escala de tempo e de espaço, pois os mecanismos que o regulam são diferentes, dependendo da escala em que se situa o trabalho[40].

Este trabalho tem como foco as menores escalas do comportamento ingestivo, que variam de segundos a horas, mais especificamente os bocados e atividades similares. Nesta escala, a variável resposta de maior importância é a massa do bocado (MB)[6], sendo a profundidade de bocado seu parâmetro mais determinante[9]. Na Figura 1.2, é possível ver que o bocado é geometricamente determinado pelo produto de sua área e de sua profundidade. Segundo Hodgson et al.[26] a altura das plantas tem uma relação proporcional com a profundidade do bocado.

O bocado e outras atividades estão relacionados a habilidade do animal em escolher que partes da planta colher, tendo como objetivo uma seleção com partes mais nutritivas do pasto elevando a qualidade de sua dieta[10]. O aumento da taxa de bocados por exemplo, pode significar alguma situação de limitação de forragem (baixa altura do pasto e/ou massa de forragem) pois os animais tem uma profundidade de bocado limitado.

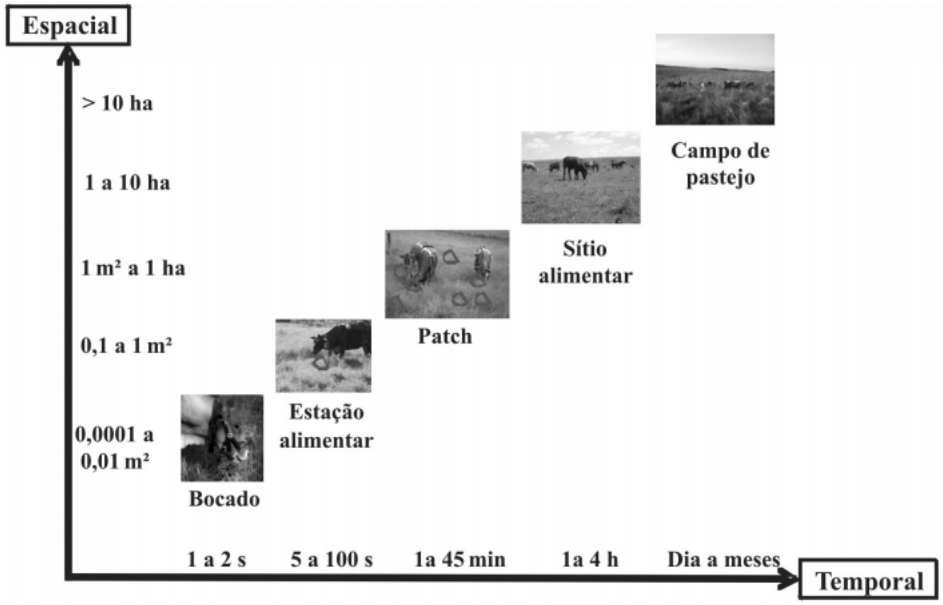


Figura 1.1: Escalas espaçotemporais que resultam do comportamento ingestivo em pastejo abordado como processo hierárquico (adaptado de Laca; Ortega, 1995; Bailey et al., 1996; Carvalho; Moraes, 2005; Bailey; Provenza, 2008)[32][5][10][6]. A resolução das escalas espaciais varia entre espécies de animais. Os intervalos aproximados são ilustrados para auxiliar o leitor na percepção da diferença entre os níveis. Os intervalos temporais são usados para definir as unidades de seleção.Fonte: Forragicultura: ciência, tecnologia e gestão dos recursos forrageiros. pg. 526[40]

O monitoramento de precisão do comportamento de gado é necessário para garantir que a maioria dos requisitos básicos de saúde e bem-estar animal serão atendidos e consistentes com as práticas que assegurem o uso sustentável e eficiente dos recursos naturais. Assim, diferentes esforços têm sido postos em encontrar técnicas para medir e monitorar a dieta e o comportamento alimentar de animais[17][27]. Entretanto, estudos de comportamento ingestivo de animais em pastejo carecem ainda de métodos acurados e de fácil utilização. Nos últimos anos, houve algum avanço, mas ainda há uma demanda de inovação em métodos precisos e automatizados que facilmente quantifique o tempo dedicado pelos ruminantes domésticos na realização de suas atividades diárias[47]. Uma maneira de alcançar esse objetivo é usar da bioacústica e seus métodos.

Os métodos da bioacústica tem sidos propostos por não serem invasivos, com baixa custo e que possibilita identificar as atividades dos ruminantes de forma contínua, sem afetar o comportamento ingestivo do animal[29]. Bioacústica para avaliar o comportamento de animais foi primeiramente proposta por Alkon et al.[2], Laca et al.[33]. Eles

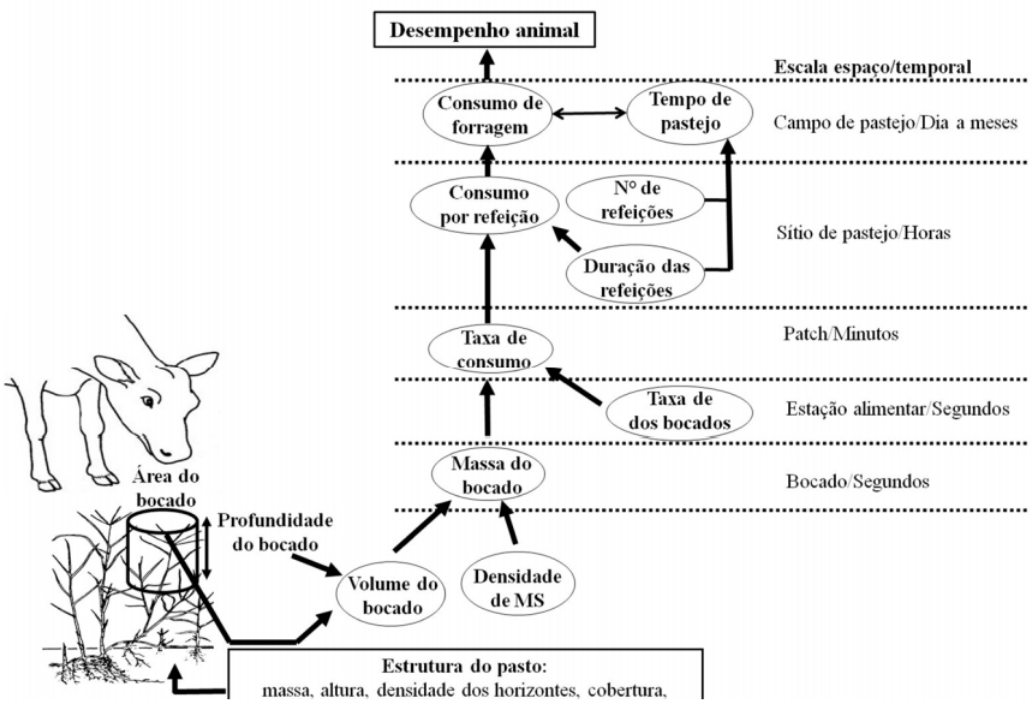


Figura 1.2: Constituição do consumo em pastejo a partir do bocado e principais componentes envolvidos no comportamento ingestivo em diferentes escalas espaçotemporais do processo de pastejo (adaptado de Allden; Whittaker, 1970; Cangiano et al., 1999; Bailey; Provenza, 2008; Carvalho et al., 2009)[3][8][6][11]. Fonte: Forragicultura: ciência, tecnologia e gestão dos recursos forrageiros. pg. 527[40]

constataram que bocados e mastigações poderiam ser facilmente identificados e contados pela inspeção de registros sonoros ao invés da observação direta. Segundo Laca, E; Wallis DeVries[31], a gravação de som contém uma grande riqueza de informações que podem ser reunidas de maneira que não interfira no comportamento de pastejo e que pode prestar-se a análise automatizada. Além disso, uma vantagem importante da abordagem acústica é que permite a contagem de mastigações e bocados, assim como de movimentos compostos de bocado e mastigação. A maioria das técnicas utilizadas para avaliar o comportamento ingestivo não consegue discriminar movimentos compostos de mastigação, manipulação e bocados[39]. Entretanto, alguns estudos promissores indicam a possibilidade de utilizá-los para quantificar parâmetros do consumo de forragem[31][21].

Diante deste panorama, surge uma oportunidade para propor uma ferramenta capaz de detectar e classificar eventos de mastigação, bocados, eventos compostos de bocado e mastigação e eventos de ruminação de bovinos utilizando aprendizado de máquina e a base de dados obtida por Da Trindade et al.(2012)[16] que contém gravações de 48 horas

e em ambientes com condições menos controladas. Levando isso em consideração a seguir é apresentado o objetivo geral e os objetivos específicos.

1.2 Objetivo Geral

Desenvolver uma ferramenta para segmentar, identificar e classificar eventos de mastigação, bocados, eventos compostos de bocado e mastigação e eventos de ruminação de bovinos utilizando dados coletados em ambientes não controlados.

1.2.1 Objetivos Específicos

- Desenvolver um algoritmo capaz de segmentar gravações de atividades ingestivas bovinos em eventos;
- Desenvolver um algoritmo capaz de classificar segmentos de possíveis eventos; e
- Desenvolver uma ferramenta capaz de segmentar, identificar e classificar eventos ingestivos bovinos.

1.3 Organização do texto

Essa dissertação de mestrado está organizada nos seguintes capítulos e apêndice:

- Capítulo 2: É apresentado uma revisão bibliográfica de diversos trabalhos referentes à automatização do processo de análise de dados bioacústico de atividades ingestivas de bovinos.
- Capítulo 3: É apresentado um conjunto de conceitos teóricos importantes para o entendimento do trabalho.
- Capítulo 4: É apresentado a metodologia aplicada nos experimentos para a escolha dos melhores métodos para as fases de segmentação (detecção) e classificação dos eventos ingestivos.
- Capítulo 5: É apresentado os resultados dos experimentos e são discutidos.
- Capítulo 6: É descrito a ferramenta BovChewing criada por meio dos métodos escolhidos no Capítulo 5.
- Capítulo 7: Apresenta-se a conclusão da dissertação, as devidas contribuições, limitações identificadas e propostas de trabalhos futuros.

- Apêndice A: Lista todos os relatórios dos experimentos dos classificadores.

Capítulo 2

Revisão bibliográfica

Monitorar de forma precisa o comportamento alimentar é essencial para uma gestão da atividade de pastejo. Uma indicação da saúde e do bem-estar do animal pode ser obtida por meio do monitoramento das atividades da pastagem[34]. A ingestão de forragem é a chave para a compreensão da dinâmica do sistema de pastejo de forragem[13].

Assim, vários dispositivos mecânicos e eletrônicos foram desenvolvidos para registrar automaticamente o comportamento de pastoreio. No entanto, estes métodos são em geral imprecisos e invasivos, porque exigem calibrações individuais e equipamentos volumosos sobre o animal[35]. Métodos acústicos foram apresentados de forma pioneira por Laca et al[33]., e eles permitem medir com precisão os movimentos mandibulares que podem ajudar a calcular o consumo de matéria seca ingerido a curto prazo. Entretanto esses métodos em alguns períodos com duração maior que minutos são impraticáveis, pois precisaria de um especialista para ouvir e marcar as atividades de pastejo manualmente assim consumindo muito tempo[35].

Como pioneiro na automatização da análise dos dados, em Milone et al.(2009)[34] descreve um modelo de Cadeia Oculta de Markov capaz de classificar as atividades ingestivas de ovelhas obtendo um resultado de 89% em eventos de mastigação, 58% em eventos de bocados e 56% em eventos compostos de bocados e mastigação além de identificar a espécie da forragem utilizada nos experimentos. Esse trabalho contou com as gravações realizadas em sessões que uma bandeja com a forragem plantada em vasos e dispostos ao alcance do animal equipado com um microfone sem fio, e apesar de ser um ambiente controlado as gravações continham ruídos externos.

Clapham et al.[13] realizou gravações e sua análise manual juntamente com uma análise automática utilizando o software SIGNAL que analisa baseado em características sonoras, tais como: frequência, intensidade, duração e tempo entre eventos. Essa análise automática é atrelada à parâmetros que foram definidos por meio tentativa e erro comparando os resultados preliminares do método automático com contagem manual dos

bocados. Em seu trabalho os autores buscaram contar os eventos de bocado apenas e não foi projetado para detectar a duração do evento. Seus experimentos obtiveram uma taxa de acerto de 95%.

Em Milone et al.(2012)[35] utilizando a técnica da Cadeia Oculta de Markov de seu trabalho de 2011, mostrou a possibilidade de identificar e classificar eventos ingestivos de bocado, mastigação e compostos por bocados mais mastigação, diferentemente do Clapham et al.[13] que era apenas bocados. Seus experimentos foram realizados em sessões onde a forragem ficava em vasos fixo em uma placa de madeira que o animal equipado com o microfone tinha acesso. Os resultados obtidos foram diferentes baseado na espécie de forragem e em sua altura, mas em média 82% para eventos de bocado, 93% para eventos de mastigação e 85% para eventos compostos de bocado e mastigação.

Navon et al.[37] implementou um algoritmo dividido em seis estágios, que utilizam conceitos de aprendizado de máquina e análise de características das gravações (duração, intensidade, formato, sequência de eventos), e utilizando gravações de gado, ovelhas e cabras foi capaz de identificar e classificar eventos ingestivos - bocado, mastigação e compostos de bocado mais mastigação - e com gravações com pouco ruído obteve uma taxa de acerto média de 94% com gado, 96% com gravações de cabras e 84% com gravações de ovelhas.

Em Tani et al.[45] foi aplicado técnicas de reconhecimento de padrão para medir atividades de alimentação e ruminação, sem a necessidade de ajustes em parâmetros. Usando acelerômetros junto com os microfones, busca por meio de análise dos espectrogramas gerados pelas gravações reconhecer um padrão, ou um espectrograma "médio"de cada tipo de evento ingestivo. Com o espectrograma médio é possível comparar com gravações novas e assim identificar e classificar eventos ingestivos. A metodologia obteve resultados parecidos com o de Navon et al.[37]. Em ambos os trabalhos constatou-se que ambientes com muito ruído podem comprometer os resultados.

Mais recentemente em Chelotti et al.[12] foi desenvolvido um algoritmo que em tempo real consegue detectar e classificar eventos de mastigação, bocado e eventos compostos de bocado e mastigação em pastejo de bovinos com pouco uso de recursos computacionais que em seus experimentos obteve uma taxa de acerto de 97,4% em média.

Apesar de ter diferentes procedimentos e resultados com a detecção e classificação de sinais associados à atividades de bovinos, inclusive alguns deles ofereceram possibilidades de classificação de eventos de mastigação, bocados e eventos compostos de bocado e mastigação, poucos deles detectam e classificam a atividade de ruminação. Aliado a isso, as bases de dados utilizada por Chelloti et al.[12] e Milone et al. [35] foram as mesmas que tem como característica gravações de curta duração e feitas em condições controladas.

Capítulo 3

Fundamentação teórica

Neste capítulo serão abordadas as definições dos algoritmos e técnicas para o entendimento do texto e sua aplicações nos experimentos e na construção da ferramenta BovChewing.

3.1 Comportamento ingestivo de animais em pastejo

Na escala de segundos do comportamento ingestivo dos bovinos as atividades avaliadas estão o bocado, a mastigação, o bocado com mastigação e mastigação de ruminação, nas próxima subseções cada uma dessas atividades será descrita.

3.1.1 Bocado

Considerado o átomo do processo de pastejo, compreende o processo de apreensão e arranquio da forragem para dentro da boca. A representação gráfica do som de um bocado é mostrado na Figura 3.1, e tendo em vista o objetivo este trabalho usa como variável de estudo o tempo do bocado (B), é possível ver neste exemplo que o tempo foi aproximadamente de 200ms.

O intervalo entre bocados também é um dado importante, e é representado pelo tempo do fim de um bocado, ou bocado com mastigação, e o início do próximo. Na Figura 3.2 está demarcado por porções hachuradas.

Com o intervalo entre bocados (IB) e a soma do tempo dos bocados (SB) e o número de bocados pode-se calcular a Taxa de bocados por minuto que é dado pela Equação 3.1.

$$Tx_B = \frac{B}{IB + SB} * 60 \tag{3.1}$$

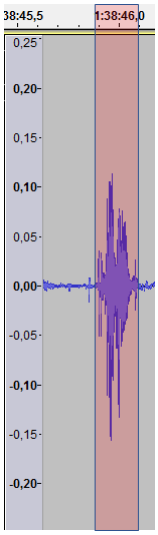


Figura 3.1: Bocado com uma duração aproximada de 200ms

3.1.2 Mastigação

Compreende o processo de mastigar a forragem desfolhada durante o pastejo. A Figura 3.3 mostra uma sequência de mastigação, onde é demarcado o tempo da mastigação (M), uma variável de estudo deste trabalho. Vale ressaltar que outra informação importante é a Taxa de mastigação por minuto dada como a relação entre o número de Mastigação e a soma do tempo de mastigação que é dado pela Equação 3.2.

$$Tx_M = \frac{M}{SM} * 60 \tag{3.2}$$

3.1.3 Bocado com Mastigação

Movimento composto de bocado mais mastigação, é uma estratégia dos animais para otimizar o tempo na colheita da forragem. Estratégia muito utilizada pelos animais em pastos com elevada abundância de forragem. A variável importante é o tempo do bocado mais mastigação (BM), a Figura 3.4 mostra nas marcações o movimento de bocado mais mastigação.

A relação entre o número de bocado mais mastigação e o intervalo entre os bocados mais mastigação em segundos e a soma do tempo de bocado mais mastigação em segundos é chamado de Taxa de bocado mais mastigação por minuto e é representado pela Equação 3.3.

$$Tx_{BM} = \frac{BM}{IBM + SBM} * 60 \tag{3.3}$$

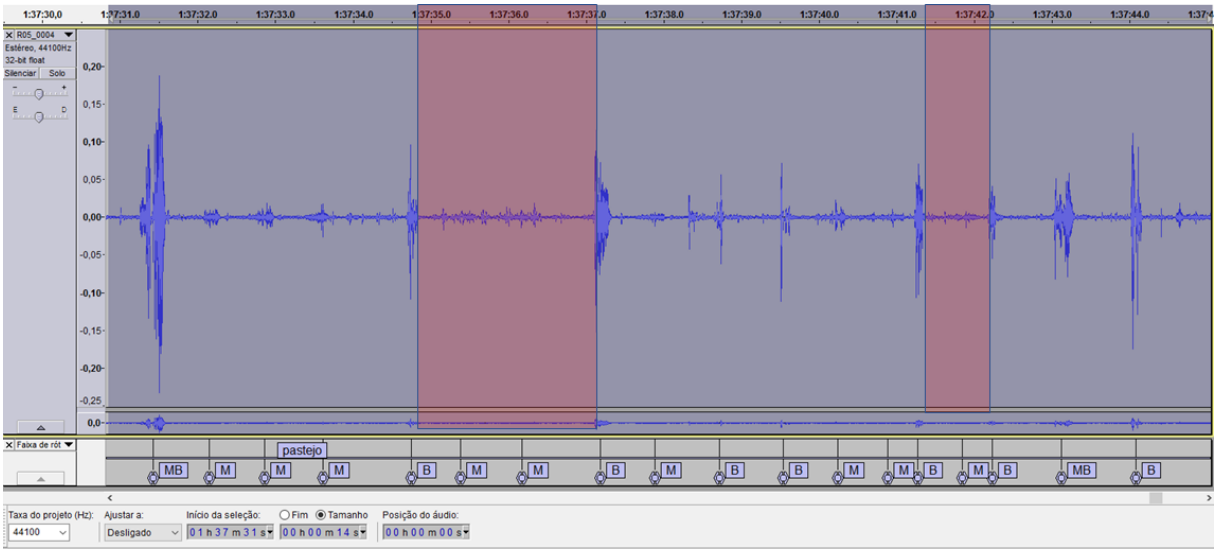


Figura 3.2: Intervalos entre bocados hachurados.

3.1.4 Mastigação de ruminação

Compreende o processo de mastigação do bolo ruminal, a Figura 3.5 mostra eventos de mastigação de ruminação (MR ou R) com um tempo aproximado de 262ms. É possível medir a Taxa de mastigação por minuto de ruminação dado pela relação entre o número de MR pela soma da duração dos MR em segundos como na Equação 3.4:

$$Tx_{MR} = \frac{MR}{SMR} * 60 \tag{3.4}$$

3.2 Aprendizado de máquina

As técnicas de aprendizado de máquina pregam o princípio da indução, que a partir de exemplos obtém-se conclusões genéricas sobre objetos de certos domínios, por exemplo, classificar se um email é spam ou não. As técnicas de aprendizado de máquina podem-se dividir em dois tipos: o supervisionado e o não-supervisionado.

No aprendizado supervisionado tem-se a figura de um professor externo, o qual apresenta o conhecimento do ambiente por conjuntos de exemplos na forma: entrada, saída desejada[25]. O algoritmo induz a representação do conhecimento representado pelos exemplos apresentados. O objetivo é de que com o conjunto de treinamento o algoritmo seja capaz de reproduzir saídas corretas para exemplos novos e desconhecidos.

No aprendizado não-supervisionado não há a presença de um professor, ou seja, não existem exemplos rotulados. O algoritmo aprende a representar (ou agrupar) as entradas

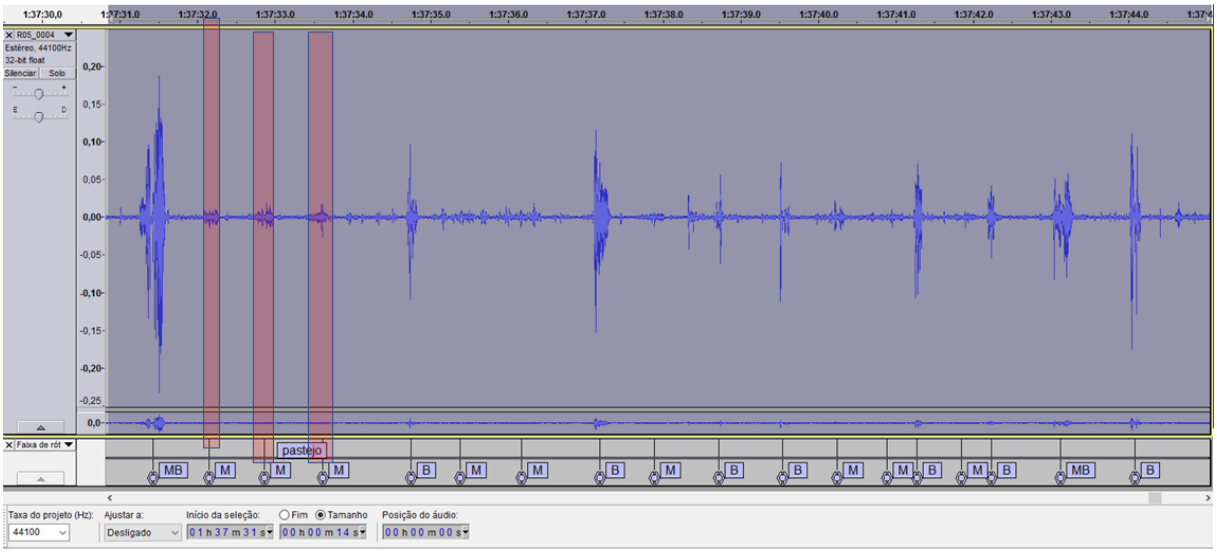


Figura 3.3: Evento de mastigação demarcado.

submetidas segundo uma medida de qualidade. Essas técnicas são utilizadas principalmente quando o objetivo for encontrar padrões ou tendências que auxiliem no entendimento dos dados.

O tipo de aprendizado de máquina abordado neste trabalho é o supervisionado. Na Figura 3.6 é apresentada uma visão geral simplificada de um processo de aprendizado de máquina supervisionado, um conjunto de treinamento rotulado na forma $(X_i,\ y_i)$ é oferecido, onde X_i representa o exemplo e y_i denota seu rótulo e então uma técnica de aprendizado de máquina é utilizada para gerar um classificador capaz de rotular novos exemplos desconhecidos.

Normalmente os exemplos X_i são representados por vetores de características, tais como, $X_i = (x_0, x_1, ..., x_n)$. Então o conjunto de treinamento é apresentado como uma matriz $m \times n$, onde m representa o número de exemplos e n o número de característica que representa aquele exemplo no nível de abstração desejado.

O rótulo ou classe pode-se entender como o fenômeno de interesse sobre o qual se deseja fazer previsões. Neste trabalho o fenômeno de interesse é a classificação de segmentos de áudio em eventos ingestivos bovinos: Bocado, Mastigação, compostos por Bocado e Mastigação e Mastigação de Ruminação.

Na Figura 3.6 é possível ver uma propriedade marcante das técnicas de aprendizado de máquina supervisionado, onde existem duas fases: a fase de treinamento e a fase de classificação. Na primeira fase o classificador aprende como é possível classificar novos exemplos a partir da análise dos exemplos rotulados oferecidos e na segunda fase o classificador construído é usado para classificar novos exemplos.

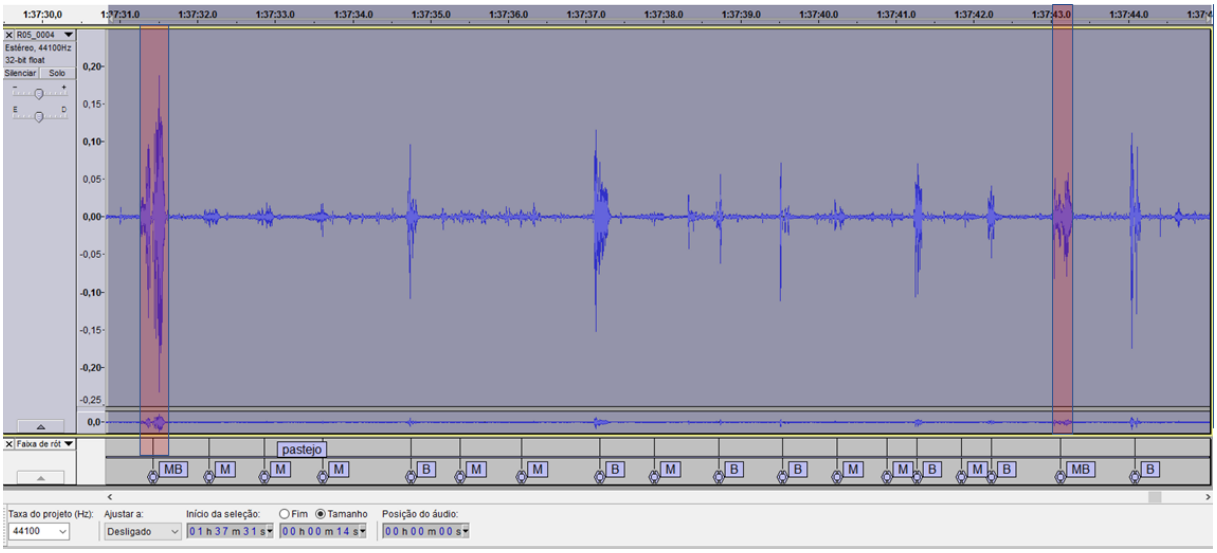


Figura 3.4: Evento de bocado mais mastigação.

As técnicas (algoritmos) utilizadas no trabalho para os experimentos e construção da ferramenta BovChewing são descritas nas próximas subseções.

3.2.1 K-Vizinhos mais próximos (KNN)

Considerado por Mitchell[36] um dos algoritmos mais simples de aprendizado de máquina pertencente ao grupo de métodos baseado em exemplos (ou instâncias), o KNN assume que todas as instâncias correspondem a pontos em um espaço n-dimensional em R^n. Os vizinhos próximos são definidos em termos da distância euclidiana. Mais precisamente, seja um exemplo aleatório x descrito pelo conjunto de características:

$$(a_1(x), a_2(x), ..., a_n(x))$$

Onde $a_t(x)\ \forall\ t \in [1, n]$ é o valor da característica a_t da instância x. Sendo assim a distância entre duas instâncias x_i e x_j se dá por:

$$d(x_i, x_j) \equiv \sqrt{\sum_{t=1}^{n} (a_t(x_i) - a_t(x_j))^2}$$

Uma vez que o conjunto de treinamento é composto por instâncias em um espaço n-dimensional, a tarefa de classificar um novo exemplo h é resumida em achar os k exemplos vizinhos mais próximos a h. Uma vez encontrados os k exemplos vizinhos mais próximos, se k for igual a 1, h é classificado com a classe do exemplo encontrado, já para valores de

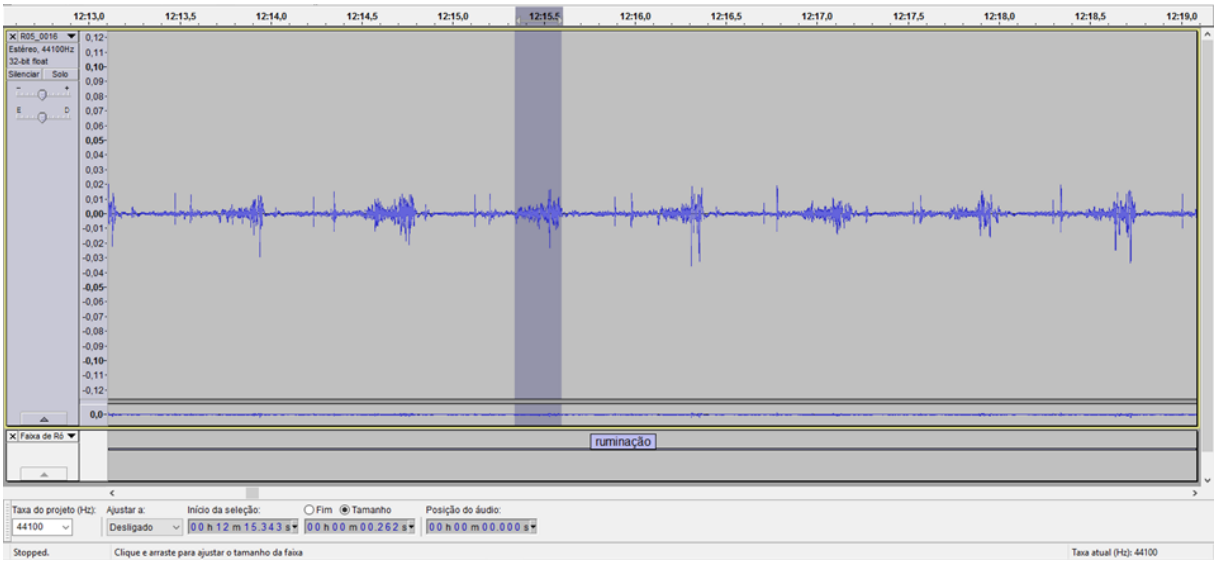


Figura 3.5: Evento de mastigação de ruminação selecionado.

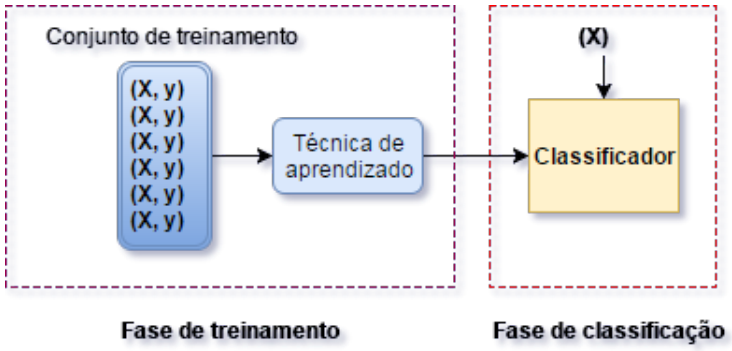


Figura 3.6: Uma visão geral simplifica de um processo de aprendizado de máquina.

k maiores que 1, h é classificado com a classe mais comum entre os k exemplos vizinhos mais próximos de h.

A Figura 3.7 representa a operação do algoritmo KNN para o caso que as instâncias são pontos em um espaço bidimensional e as classes possíveis são $+$ ou $-$. Um exemplo x_q a ser classificado é mostrado também. Note que para $k = 1$ o algoritmo classifica x_q como $+$, já quando $k = 5$ o algoritmo classifica x_q como $-$.

Durante a fase de treinamento o algoritmo apenas guarda o conjunto de treinamento sem realizar nenhum cálculo. Este algoritmo já foi utilizado nas mais variadas aplicações, por exemplo em reconhecimento de espécies de anuros e pássaros[48].

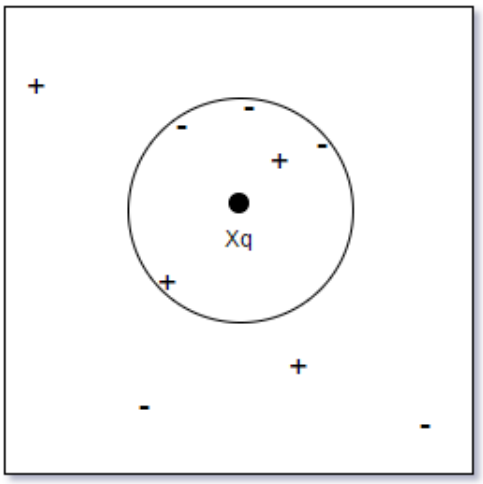


Figura 3.7: Na figura um conjunto de exemplos positivos e negativos, com um exemplo x_q a ser classificado. Para $k = 1$, x_q será classificado com $+$, entretanto para $k = 5$, x_q será classificado como $-$.

3.2.2 Máquina de vetores de suporte (SVM)

As máquinas de vetores de suporte acha um hiperplano ótimo separando os exemplos, por exemplo, o hiperplano que representa a maior margem de separação entre os exemplos de diferentes classes. Os pontos definidos pelo conjunto de treinamento que estão na margem (Hiperplano) são chamados de vetores de suportes.

A separação de forma linear entre classes com dados não lineares é algo impossível, o que é corrigido transformando o vetor de características dos exemplos que possuem n dimensões em um vetor de N dimensões, assim a separação pode ser encontrada em uma dimensão maior. Uma função: $\phi = \mathbb{R}^n \rightarrow \mathbb{R}^N$ é usada para transformar o vetor de característica, permitindo a definição do hiperplano para um conjunto de dado não-linear[15]. O *kernel* é uma função usada para encontrar a posição relativa de um novo exemplo no hiperplano e assim definir sua classe.

Durante a fase de treinamento esse hiperplano é encontrado por meio do conjunto de treinamento e a fase de classificação consiste em encontrar a posição relativa de um exemplo desconhecido e assim decidir a sua classe. Esse classificador já obteve sucesso por exemplo em classificação da espécie utilizando o coaxar de anuros[4].

3.2.3 Perceptron de Múltiplas Camadas

Uma rede neural artificial (RNA) é um modelo computacional inspirado no comportamento biológico do cérebro. As RNA são compostas por neurônios, que são unidades conectadas por ligações diretas que processam a entrada. Uma RNA pode ser usada para

problemas de classificação.

Perceptron de Múltiplas Camadas (PMC) é o tipo mais popular de RNA, onde várias camadas dos neurônios são treinados com o método de retro-propagação [7]. Nesse tipo de RNA os sinais são propagados da camada de entrada para a camada de saída por meio de uma camada oculta, e cada neurônio na camada oculta associa um peso para cada entrada. Após isso a camada passa o valor processado para a próxima camada de neurônios, até que a camada de saída processa os valores como as camadas escondida e estima a classe do exemplo. Os pesos de cada neurônio são aprendidos durante a fase de treinamento. Esse classificador já foi usado por exemplo em predição da saúde do gado leiteiro por meio de PMC e dados de sua respiração[22].

3.2.4 Cadeias Ocultas de Markov (HMM)

Este trabalho foca em classificar as atividades ingestivas de bovinos, mas para isso é necessário segmentar as gravações em segmentos de possíveis atividades e ruídos ou não importantes, e uma maneira de alcançar isso é utilizando Cadeias Ocultas de Markov. Até então, foram apresentadas técnicas em que o seu conjunto de treinamento é constituído de exemplos independentes, mas quando se trata, por exemplo de reconhecimento de voz, onde uma palavra é uma sequência de letras, as técnicas convencionais não são o bastante.

As Cadeias Ocultas de Markov (HMMs) são autômatos estocásticos que seguem as transições entre os estados com base em regras probabilísticas. Quando o HMM chega a um estado, ele emite uma observação, que, no caso de análise de sinal é geralmente um vetor de característica contínua[23].

3.3 Características

Todo exemplo apresentado para o algoritmo de aprendizado de máquina é descrito como um vetor de característica, que segundo Flach P.[20] as características determinam grande parte do sucesso de um aplicativo de aprendizado de máquina, porque um modelo é tão bom quanto suas características. Matematicamente o algoritmo deverá induzir a representação do conhecimento e assim construir um classificador. A seguir estão descritos as características utilizadas para descrever os exemplos de áudio de bocado, mastigação, bocado mais mastigação e mastigação de ruminação.

3.3.1 Coeficientes Mel-Cepstrais (MFCC)

Componentes de frequência mais baixa contêm mais informação, e uma forma de obter essas frequências mais baixas é utilizando o MFCC, onde ele extrai componentes de

frequências baixas e consegue compactar a representação do som de uma maneira ótima. O MFCC é uma representação do espectro de potência de curto prazo usando transformada de cosseno linear de uma densidade espectral para uma frequência mel[30].

3.3.2 Taxa de Cruzamento por Zero

A Taxa de Cruzamento por Zero, Zcr_m, consiste em contar, em uma janela s do sinal, a quantidade de vezes que o sinal cruza a mediana na escala da amplitude, normalizada pelo tamanho da janela, L, e é representado matematicamente pela Equação 3.5.

$$Zcr_m = \frac{1}{L}\sum_{n=0}^{L-1}\frac{|sgn\{s(n)\} - sgn\{s(n-1)\}|}{2}. \tag{3.5}$$

em que m representa um índice associado a uma janela, do sinal de voz, e

$$sgn\{s(n)\} = \begin{cases} +1, & \text{se } s(n) > 0 \\ -1, & \text{se } s(n) \leq 0 \end{cases} \tag{3.6}$$

3.3.3 Energia

A energia de uma janela s de tamanho L de um sinal S é dado pela Equação 3.7.

$$E_s = \sum_{i=0}^{L-1} s(i)^2 \tag{3.7}$$

Fisicamente a energia representa a capacidade de realizar o trabalho[42]. Particularmente, se o sinal analisado for por exemplo, um comando de voz, a energia representa o trabalho que os pulmões e as cordas vocais realizaram, em função do tempo, para produzir o som [18].

3.3.4 Entropia da energia

A entropia como variável de estado termodinâmica foi introduzida na física pelo físico alemão Rudolf Clausius na segunda metade do século XVIII, e então começou difundir uma ideia de que a entropia poderia ser usada em outras áreas, pois a mesma representa uma medida da desorganização de um sistema (termodinâmico por exemplo), tendo em vista que a entropia maximal de um sistema é quando o sistema atinge estabilidade e equilíbrio. Dessa ideia, foi promovido uma ideia de medida genérica de desorganização de um sistema[19].

A relação entre entropia e análise de sinal é baseado na hipotese que um ruído é uma projeção de um equilíbrio de um sistema termodinâmico em um sinal. Como resultado, um ruído possui um valor de entropia alto enquanto um som que representa por exemplo uma voz possui uma entropia baixo mostrando que é preciso ter uma energia e organização para produzir o som[19].

A entropia de uma janela s com tamanho L de um sinal S é dado pela Equação 3.8.

$$En_s = \sum_{i=0}^{L-1} p(s(i)) \log_2 p(s(i)) \tag{3.8}$$

Onde $p(s(i))$ é dado pela Equação 3.9.

$$p(s(i)) = \frac{s(i)^2}{\sum_{k=0}^{L-1} s(k)^2} \tag{3.9}$$

Capítulo 4

Metodologia

Para alcançar os objetivos vistos no Capítulo 1 foi preciso estabelecer uma metodologia para a realização dos experimentos e criação da ferramenta proposta por este trabalho. A seguir é apresentado a base de dados que foi utilizado para a construção da ferramenta.

4.1 Base de dados

Para a realização dos experimentos e desenvolvimento da ferramenta, uma base de dados de gravações do comportamentos bovinos se fez necessário.

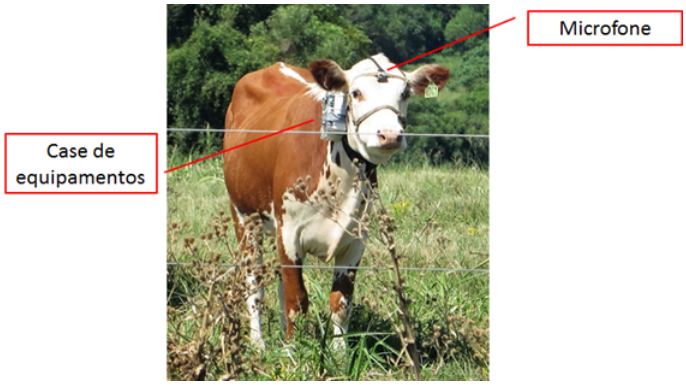
Os registros acústicos foram obtidos de novilhas da raça Braford, em média com 12 meses de idade e 170 kg de peso vivo manejadas em uma área experimental de pastagem natural, pertencente à Secretaria de Agricultura, Pecuária e Irrigação (SEAPI) do Governo do Estado do Rio Grande do Sul (302019S; 541502W; 125 m acima do nível do mar no município de São Gabriel (RS).

Nesta área experimental são realizados estudos onde o objetivo principal é identificar e compreender os impactos de níveis de intensificação da produção de forragem para a produção animal em uma pastagem natural do Bioma Pampa. Para tanto, é considerado para este estudo que o conhecimento relativo aos aspectos comportamentais e de ingestão de forragem pelos animais em pastejo é importante, porque esse conhecimento é fundamental para estabelecer metas de manejo do pastejo em ambientes pastoris.

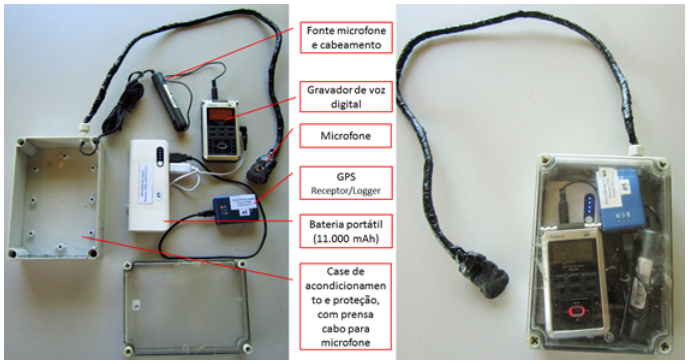
A vegetação da área experimental, e da qual os animais realizam o pastejo, é típica de campos mistos do Bioma Pampa, subarbustiva e campestre. O método de pastoreio adotado foi o de lotação contínua e taxa de lotação variável conforme condição de oferta de forram e estrutura do pasto preconizadas por Da Trindade et al.(2012)[16]. Em novembro de 2015 foram obtidos os registros acústicos, quando a oferta de forragem e os pastos apresentavam as seguintes características médias: 18.4 kg MS-100 kg de PV-1, massa de forragem 2,500 kg-ha-1 de matéria seca, altura do pasto de 12.9 cm e cobertura de área

de pastagem ocupada por espécies na forma de touceiras de 12

Para gravar o som e, posteriormente, analisar o comportamento ingestivo das novilhas em condição de pastejo, foi adotado como base a metodologia bioacústica proposta Da Trindade et al. (2011)[47]. Para gravar o som, os animais foram equipados com buçal permitia que um microfone de lapela, protegido por uma cápsula de isopor, fosse ajustado a fronte (Figura 4.1(a)). Os equipamentos para a realização das gravações acústicas, GPS e bateria ficavam acondicionados em um case de policarbonato e fibra de vidro (175 x 125 x 76 mm, Figura 4.1(b)), capaz de impedir a entrada de umidade e sujeiras e com elevada resistência impactos (índice de proteção - IP 67).



(a) Novilha equipada com buçal com microfone de lapela



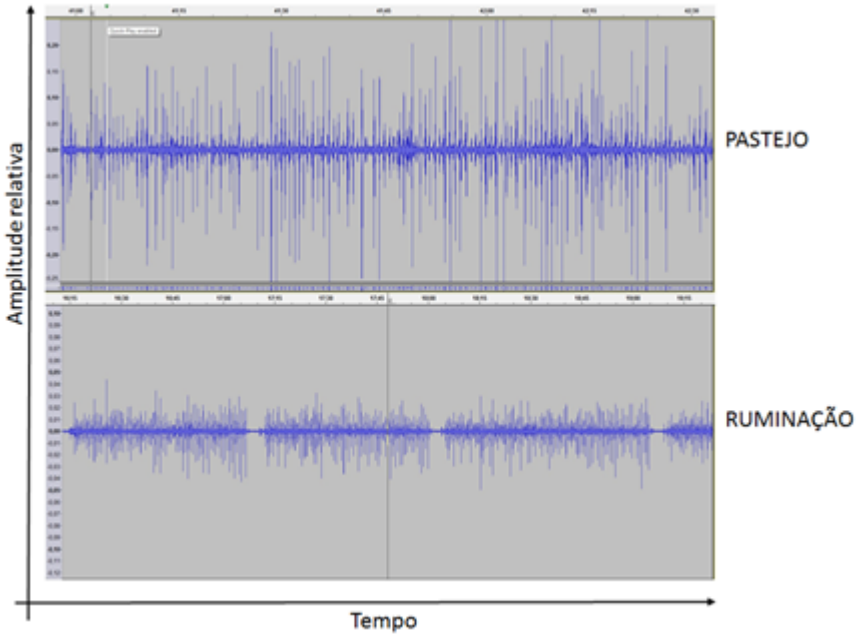
(b) Caixa do equipamento utilizado para a coleta de dados acústicos e dados do gps da novilha

Figura 4.1: Metodologia aplicada para a coleta dos dados

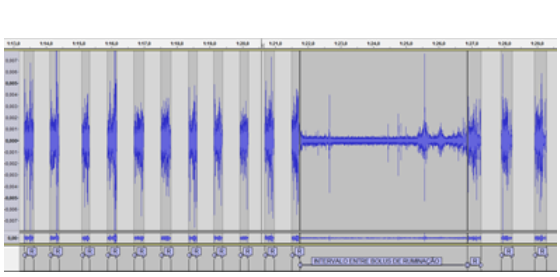
O sistema (Figura 4.1(b)) consistiu de um gravador digital (Roland®, R-05) equipado com cartão de memória SDHC de 32GB, microfone de lapela (Leson®, ML-70D), receptor GPS (Holux®, RCV-3000) e uma bateria portátil com 11.000mAH (El Shaddai Technology). O sistema é também alimentado também com pilhas e baterias próprias dos equipamentos, permite gravações acústicas contínuas por mais de 48h. Os estados

comportamentais, como pastejo e ruminação são discriminados, ademais o registro da localização e rota dos animais ao longo d o período de monitoramento acústico. O gravador digital foi configurado para gravar em modo de elevada resolução 16-bit/44.1 kHz, formato wav, mono e alta sensibilidade do microfone (interruptor do gravador alta/baixa).

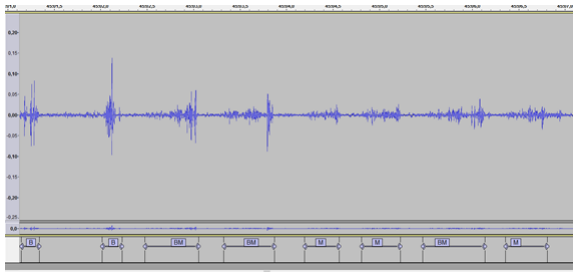
Os registros foram analisados usando o Audacity®, um software de áudio livre, de código aberto e multi-plataforma para gravação e edição multipista (http://www.audacityteam.org/). As atividades de pastejo e ruminação podem ser facilmente discriminadas de forma manual porque estas atividades tem características acústicas bem distintas (Figura 4.2).



(a) Diferenças acústicas entre atividades de pastejo e ruminação



(b) Marcação de eventos de ruminação



(c) Marcação de eventos de pastejo

Figura 4.2: Representação gráfica dos dados coletados.

Ao final das avaliações, foi gerada uma base de dados contendo marcação e rotulagem de 1330 eventos ingestivos selecionados e gravados de uma novilha que permaneceu equipada durante 48 horas consecutivas (Figura 4.1(a)):

- 238 sons representando bocados (B).
- 329 sons representando mastigação (M).
- 121 sons representando eventos compostos de mastigação e bocado (BM).
- 642 sons representado mastigação de ruminação (R).

4.2 Experimentos

As gravações são compostas por arquivos de áudio no formato WAV que contém toda a atividade do animal em um período de 48 horas e para a identificação dos eventos ingestivos é preciso segmentar esses arquivos em eventos importantes e ruídos e então classificá-los.

Tendo isso como parte do objetivo foi necessário dividir os experimentos em duas partes: Do segmentador automático e do classificador.

4.2.1 Segmentador automático

Tendo como objetivo a segmentação do sinal, a fim de separar ruído de possíveis eventos ingestivos foi necessário realizar experimentos com o intuito de garantir que novas gravações não precisem de uma segmentação manual tal como foi feito no conjunto de dados apresentado na Seção 4.1.

A Figura 4.3 apresenta o processo de experimentação. O segmentador recebe como entrada os parâmetros de execução e o arquivo da gravação (com os segmentos demarcados por um especialista apenas para fins experimentais). Em seguida o segmentador separa o arquivo em pequenos segmentos de possíveis eventos ingestivos removendo os ruídos. Ao final do processo é colhido algumas métricas do segmentador comparando os segmentos criados por ele com os segmentos demarcados manualmente por Da Trindade et al.[47].

A seguir é detalhado cada componente do processo e os métodos de segmentação que são avaliados pelo experimento.

Métodos de segmentação

Para fins da escolha do melhor processo de segmentação para no uso do *BovChewing*, foi realizado experimentos com dois métodos: Um método semi-supervisionado utilizando um modelo de classificação construído com um método supervisionado, uma máquina de vetores de suporte (SVM) para detectar possíveis eventos[23] e um método supervisionado de segmentação baseado em Cadeias Ocultas de Markov[23].

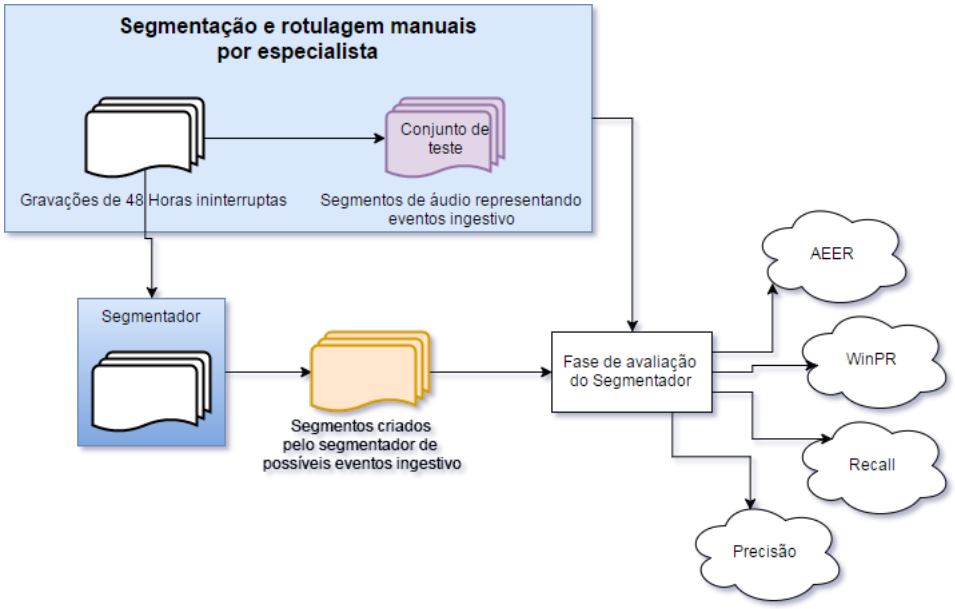


Figura 4.3: Modelo conceitual do processo de experimentos do segmentador.

Semi-supervisionado

Utilizando a biblioteca pyAudioAnalysis[23] dado um sinal x é possível segmentar o aúdio utilizando uma técnica de aprendizado supervisionada de uma maneira semi-supervisionada onde nenhum conjunto de treinamento externo é necessário.

O método tem como entrada o arquivo de áudio a ser segmentado e quatro parâmetros: *Window Size*, *Window Step*, *Smooth Window* e *Weight*, que são o tamanho da janela de amostragem do áudio, a sobreposição da janela, um tamanho de janela para suavizar as probabilidades do modelo SVM e um limiar para a detecção dos eventos respectivamente.

Segundo Giannakopoulos[23] o método segue os seguintes passos:

1. Um conjunto de características é retirado do sinal.
    - O sinal é dividido em janelas de tamanho *Window Size* com sobreposição de janela de tamanho *Window Step*.
    - De cada janela é calculado as seguintes características: 13 frequência mel, energia, taxa de cruzamento por zero, entropia da energia.
2. Um modelo SVM é treinado distinguindo entre janelas do sinal com altos picos de energia e janelas do sinal com baixos picos de energia.
    - É selecionado 10% das janelas com os menores valores de energia e 10% das janelas com os maiores valores de energia e se transformam em um conjunto

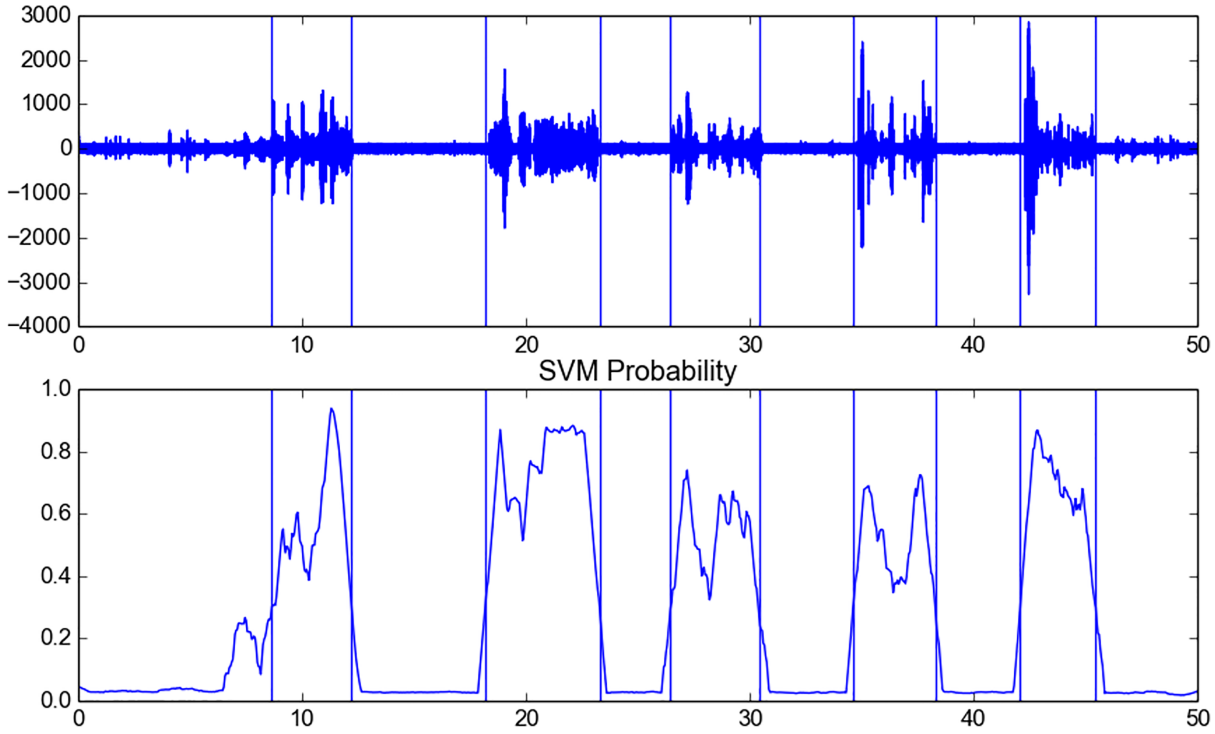


Figura 4.4: Aplicação do método Silence Removal sobre uma gravação de 50 segundos. Fonte: Giannakopoulos[23]

de treinamento para o modelo SVM.

- Um modelo SVM é treinado, onde as janelas com baixa energia são rotuladas como silêncio e as janelas com alta energia são rotuladas como evento.

3. O classificador SVM é aplicado (com uma saída probabilística) ao sinal, resultando em uma sequência de probabilidades que corresponde ao nível de confiança que alguma janela pertence a um evento ou não. O sinal passa por uma filtro de suavização da probabilidade utilizando o valor de *Smooth Window*.

4. Um limiar dinâmico é utilizado para detectar segmentos que representam eventos com o parâmetro *Weight* calcula-se a média ponderada dos 10% das janelas com maiores probabilidades de serem eventos com os 10% das janelas com menores probabilidades de serem eventos e assim se o valor da probabilidade ultrapassar esse limiar, a janela é considerada um evento.

É dado o nome de *Silence Removal* a este método da biblioteca pyAudioAnalysis.

A Figura 4.4 é mostrado de forma gráfica o resultado do *Silence Removal*, onde é possível ver o gráfico dos valores das probabilidades dos evento ser um evento ou não.

A lista abaixo mostra os argumentos utilizados nos experimentos do método semi-supervisionado, de forma que foi realizado um produto cartesiano dos argumentos gerando assim as possíveis configurações do experimento.

- Window Size: 0.02, 0.04, 0.05, 0.02, 0.01, 0.01
- Window Step: 0.01, 0.02, 0.025, 0.02, 0.005, 0.01
- Smooth Window: 0.02, 0.04, 0.05, 0.02, 0.01, 0.01, 0.1, 0.5, 1.0, 0.2
- Weight: 0.5, 0.1, 0.2, 0.3, 0.4, 0.6, 0.7, 0.8, 0.9, 1.0, 0.0

Para maiores informações sobre o método Silence Removal os leitores são referidos ao trabalho de Giannakopoulos[23].

Supervisionado com Cadeias Ocultas de Markov (HMM)

Utilizando a biblioteca pyAudioAnalysis[23] e o sinal com os segmentos (evento e não evento) anotados manualmente é possível treinar um modelo HMM que é usado para segmentar novos sinais em evento e não evento.

O método tem como entrada o arquivo de áudio a ser segmentado e dois parâmetros: Window Size e Window Step, que são o tamanho da janela de amostragem do áudio e a sobreposição da janela para a execução do método. A Tabela 4.1 mostra os argumentos utilizados nos experimentos do método supervisionado.

O método segue os seguintes passos:

- O sinal é dividido em janelas de tamanho *Window Size* com sobreposições de *Window Step* e para cada janela é calculado um conjunto de características. São elas: 13 frequência mel, energia, entropia da energia e a taxa de cruzamento com zero. Esse conjunto de janelas é chamado de conjunto de treinamento.
- Um modelo de Cadeia Oculta de Markov (HMM) é treinado utilizando o conjunto de treinamento gerado no passo anterior.
- O modelo é salvo em um arquivo e então é usado para segmentar novos áudios.

É dado o nome de HMMTrain a este método da biblioteca pyAudioAnalysis.

Para maiores informações sobre o método HMMTrain os leitores são referidos ao trabalho de Giannakopoulos[23].

Tabela 4.1: Parâmetros utilizados nos experimentos de segmentação do método supervisionado

Window Size	Window Step
0.5	0.5
0.4	0.4
0.3	0.3
0.2	0.2
0.08	0.04
1	0.5
0.5	0.25
0.4	0.2
0.3	0.15
0.2	0.1
0.1	0.05
0.02	0.01
0.05	0.05

Métricas de avaliação

Para avaliar os segmentos gerados pelos segmentadores automáticos, de possíveis eventos, estes foram comparados com os segmentos demarcados manualmente por Da Trindade et al.[47]. Dessa comparação as métricas são extraídas observando três tipos de erros: Ponto a ponto, borda a borda e evento a evento.

Para evento a evento a métrica utilizada foi a Taxa de Erro de Evento Acústico (AEER - Acoustic Event Error Rate), normalmente usado em detecção de contexto [24]. Definido por:

$$AEER = \frac{D + I + S}{N}$$

Onde N é o número de segmentos de eventos em cada arquivo de áudio, D é o número de segmentos de eventos perdidos, I é o número de segmentos de eventos extras e S é o número de eventos substítuidos calculado como sendo o $min(D, I)$. Foi considerado como um evento corretamente segmentado se ele começa ou termina com $\pm 100ms$ de diferença com a borda do evento real e se ele tem pelo menos 50% da duração do evento real. Por

isso, o melhor caso é o $AEER = 0$.

Para a análise borda a borda a métrica utilizada é baseada na abordagem desenvolvida por Scaiano e Inkpen[41]: O WinPR, onde é contado os erros borda a borda, mostrando os erros mascarados pelo AEER e seu limiar de $\pm 100ms$. O WinPR calcula os Positivos verdadeiros (PV), o negativos verdadeiros (NV), falsos positivos (FP), falsos negativos (FN), e com isso é possível calcular a precisão, revocação e F-Medida. Quanto maior o valor da precisão, revocação e F-Medida melhor o resultado da análise borda a borda.

Durante a análise ponto a ponto, assume-se que cada ponto do sinal real segmentado manualmente é comparado com cada ponto do sinal segmentado pelo segmentador tendo como resultado uma classificação binária (é sinal ou não é sinal), gerando assim uma tabela ou matriz de confusão (utilizada em detecção de som ou segmentação), de onde é extraído, além do VP, VN, FP e FN, a taxa de verdadeiros positivos (TVP), taxa de falsos negavitos (TFN) e o Coeficiente de Correlação Mathews (CCM). Valores altos para o TVP e valores baixos para o TFN indicam uma pequena perda de sinal.

4.2.2 Classificador

Observando o conjunto de dados apresentado na Seção 4.1 nota-se que o número de tipos de evento são quatro, que em aprendizado de máquina é dado o nome de classe, logo são quatro classes.

Levando em consideração o objetivo, pode-se dizer que o modelo que o trabalho busca montar é um modelo de aprendizado de máquina supervisionado multiclasse. Supervisionado pois sabe-se as classes do conjunto de treinamento e multiclasse pois o modelo deve classificar um novo segmento em mais de duas classes diferentes.

Tendo isso em vista, os experimentos de modelos de aprendizado de máquina supervisionado são compostos por quatro ingredientes básicos:

1. O método de validação do modelo.
2. O algoritmo e seus parâmetros.
3. O conjunto de treinamento e teste.
4. As métricas a serem avaliadas.

Na Figura 4.5 é possível visualizar o processo de experimentação de uma forma simplificada. O processo se inicia com a segmentação e rotulagem das gravações de áudio de forma manual por Da Trindade et al.[16], com os segmentos rotulados. O próximo passo é a extração de características para a geração de um arquivo contendo descrição de todos os segmentos, e então o Método de Validação Cruzada é executado para a avaliação do modelo utilizando essa base de dados.

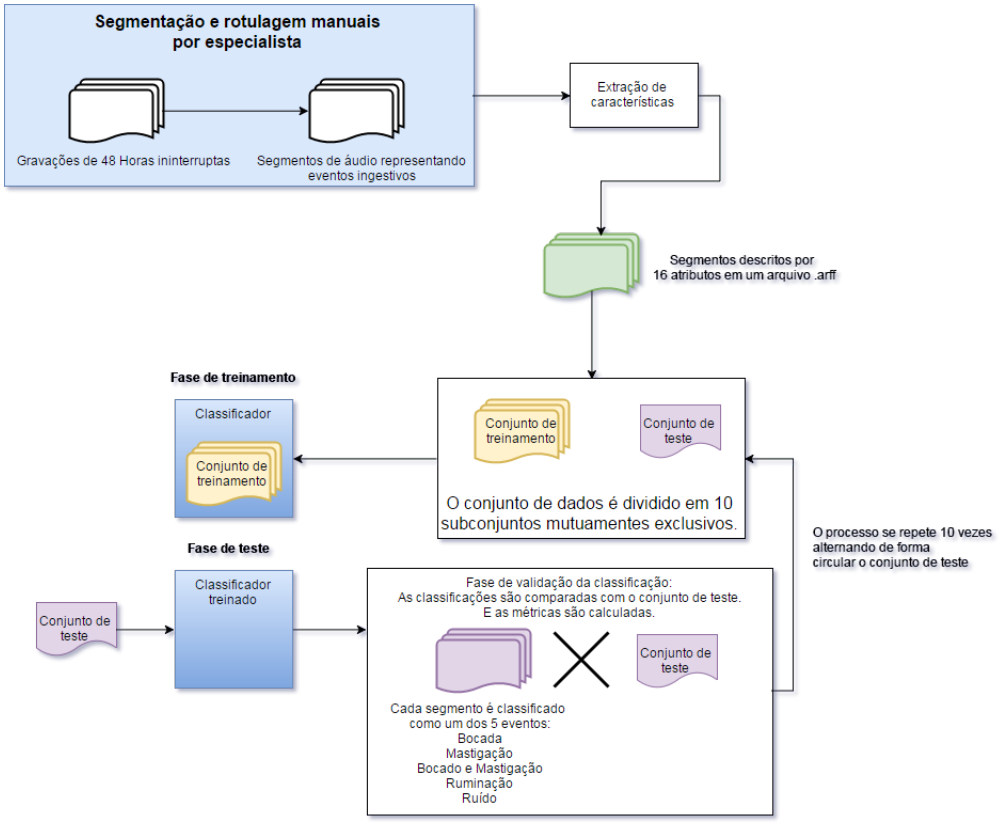


Figura 4.5: Modelo conceitual do processo de experimentos do classificador.

O método de validação do modelo

O método de validação escolhido foi o Método de Validação Cruzada de 10 partições, tendo em vista sua simplicidade. O Método de Validação Cruzada tem como objetivo avaliar a capacidade de generalização do modelo e assim evitando sobreajuste do conjunto de dados. Nesse método o conjunto de dados é dividido normalmente em 10 subconjuntos mutuamente exclusivos de mesmo tamanho, onde um subconjunto fica para teste e o restante para treinamento do classificador empregado no modelo e então após o treinamento e teste calcula-se as métricas do modelo. Esse processo é realizado 10 vezes alternando de forma circular o subconjunto de teste. Ao final é calculado uma acurácia do modelo baseada nas métricas de cada iteração.

O algoritmo e seus parâmetros

Neste trabalho foram escolhidos três algoritmos de classificação para os experimentos: SVM, KNN e Perceptron multicamadas. Para maiores informações sobre os algoritmos e técnicas os leitores são referidos ao livro de Mitchell[36]. Foram utilizadas as imple-

mentações dos algoritmos da ferramenta WEKA.

Na Tabela 4.2 são apresentados os parâmetros de cada algoritmo. A escolha dos parâmetros é baseada em diversos trabalhos sobre classificação de sinais bioacústicos[14][38][49][46].

Tabela 4.2: Parâmetros dos algoritmos utilizados nos experimentos

Classificador	Parâmetro
KNN	K = 1
	K = 3
	K = 5
	K = 7
LibSVM	Kernel Linear e C = 0.001
	Kernel Linear e C = 0.01
	Kernel Linear e C = 0.5
	Kernel Linear e C = 1
	Kernel Linear e C = 3
	Kernel Linear e C = 5
	Kernel Linear e C = 10
	Kernel RBF, C = 1 e G = 1
	Kernel RBF, C = 1 e G = 2
	Kernel RBF, C = 0.5 e G = 2
MultilayerPerceptron	L = 0.01 M = 0.1
	L = 0.1 M = 0.1
	L = 0.05 M = 0.1
	L = 0.08 M = 0.1

O conjunto de treinamento e teste

O conjunto de dados do experimento é composto por 1330 exemplos de eventos ingestivos tal como descritos na Seção 4.1. Mas para que o modelo possa aprender com esses dados normalmente é extraído um conjunto de características dos exemplos.

Esse conjunto de características é extraído utilizando a biblioteca pyAudioAnalysis[23]. Neste trabalho é proposto extrair:

- MFCC
- Zero-Crossing Rate
- Energia
- Entropia da energia

A escolha desse conjunto de características é baseado em diversos trabalhos envolvendo identificação automática de sinais bioacústicos[14][38][49][46]. O procedimento da extração segue como descrito no trabalho de Giannakopoulos[23].

E então após a extração de características todos os exemplos, que são segmentos de arquivos de áudio, transformam-se em um arquivo do tipo .arff[1] em que cada segmento é representado por 1 linha com 17 colunas, onde 16 representam características (13 valores representando as frequências mel, energia, entropia da energia e a taxa de cruzamento por zero) e 1 representa a classe do segmento (qual evento ingestivo aquele segmento representa).

A extração de características tem como entrada o tamanho da janela que o segmento será dividido e o tamanho da sobreposição da janela, sendo que para os experimentos foi adotado o tamanho da janela igual ao tamanho do segmento do evento e a sobreposição da janela a metade de seu tamanho.

As métricas a serem avaliadas

Ao final do processo de Validação Cruzada a ferramenta WEKA emite um relatório contendo algumas métricas associada ao desempenho do modelo. São elas:

- Taxa de verdadeiros positivos: Eficácia de um classificador para identificar exemplos de uma classe C_i.
- Taxa de falsos positivos: Mais conhecido como taxa de falso alarme na identificação de exemplos que não são da classe C_i mas foram classificados como.
- Precisão: Dada uma classe C_i, a precisão do modelo é dado por $\frac{VP}{VP+FP}$, onde VP é o número de verdadeiros positivos e FP é número de falsos positivos.
- F-Medida: A média harmônica entre a taxa de positivos verdadeiros e precisão. Representa a relação entre os eventos encontrados que são importantes e a fração de eventos importantes que são encontrados.

- Coeficiente de Correlação Matthews: O coeficiente de correlação de Matthews é uma medida de qualidade. Retorna um valor entre (-1) e (+1), em que um coeficiente de (+1) representa uma predição perfeita, (0) representa uma predição aleatoria média, e (-1) uma predição inversa.

- Curva ROC: É uma representação gráfica bi-dimensional que a Taxa de Verdadeiro Positivo (TVP) é projetada no eixo Y e a Taxa de Falso Positivo (TFP) é projetada no eixo X. O gráfico ROC descreve o *tradeoff* entre os exemplos classificados corretamente (VP) e os falsos alarmes (FP). Dessa curva é retirada a área, onde quanto mais próximo de 1.0 melhor, mostrando uma predição perfeita.

- Acurácia: Dada uma classe C_i, a acurácia do modelo é dado por $\frac{VP+VN}{N}$, onde VP é o número de verdadeiros positivos e VN é número de verdadeiros negativos e N é o número total de exemplos da classe C_i, ela representa o quão bom o modelo classifica corretamente os exemplos.

Todas as definições das métricas foram retiradas de Sokolova et al.[44] e Jurman et al.[28] e como a base de dados não é balanceada, para todas as métricas (do segmentador e do classificador) foram calculadas para cada tipo de evento e então é calculado a média desses valores, isso é conhecido como *macro-averaging* recomendado por Sokolova et al.[44].

Capítulo 5

Resultados

5.1 Segmentador

Os experimentos dos segmentadores foram executados com o propósito de escolher os melhores parâmetros dos métodos e o melhor método para a criação da ferramenta BovChewing.

Nessa seção é mostrado os resultados das escolhas dos parâmetros de cada método de segmentação e a comparação entre os resultados dos dois métodos de segmentação aplicando os melhores parâmetros.

5.1.1 Escolha dos parâmetros para o método Silence Removal

Na Tabela 5.1 é mostrado os 10 melhores resultados utilizando o método Silence Removal (Semi-supervisionado) da Seção 4.2.1 obtidos utilizando a metodologia descrita no Capítulo 4 e levando em consideração os melhores resultados na avaliação evento a evento ordenando de forma crescente pelo AEER.

Pode-se ver que o melhor resultado do AEER possui uma taxa de erro (TFN) muito alta onde pode ocorrer uma grande perda de sinal. Isso é devido ao fato do AEER avaliar se o segmento tem pelo menos 50% do segmento real que pode ter pouca informação importante. Adicionalmente essa combinação de parâmetros mostra um CCM negativo indicando um número muito grande de segmentos que não existem na segmentação manual.

Mas se caso for avaliar os melhores parâmetros pelo maior valor de F-Medida, tem-se uma avaliação do tipo borda à borda que foca em saber se os segmentos começam ou terminam no mesmo instante que os segmentos reais. Na Tabela 5.2 é mostrado os 10 melhores resultados ordenados de forma decrescente pelo valor de F-Medida (F1).

O melhor resultado possui um F-Medida (F1) alto e um AEER não tão diferente do

Tabela 5.1: Os melhores resultados do erro evento a evento (AEER) do método Silence Removal (os valores de parâmetros estão na ordem: Window Size, Window Step, Smooth Window, Weight).

Parâmetros	TVP	TFN	CCM	Precisão	F1	AEER
0.02 0.02 0.05 1.0	0.585307	0.414693	-0.01018	0.458119	0.513961	**0.997783**
0.02 0.01 0.02 0.9	0.775435	0.224565	**0.090634**	**0.483077**	0.595298	0.998505
0.01 0.005 0.01 0.9	**0.892418**	**0.107582**	0.02482	0.460216	**0.607267**	1.00299
0.05 0.025 0.04 1.0	0.79492	0.20508	0.049952	0.469252	0.590137	1.00299
0.05 0.025 0.05 1.0	0.79492	0.20508	0.049952	0.469252	0.590137	1.00299
0.05 0.025 0.1 1.0	0.79492	0.20508	0.05135	0.469656	0.590457	1.00299
0.05 0.025 0.2 1.0	0.79492	0.20508	0.051699	0.469757	0.590537	1.00299
0.05 0.025 1.0 1.0	0.79492	0.20508	0.054485	0.470569	0.591178	1.00299
0.01 0.01 1.0 1.0	0.750187	0.249813	0.077632	0.480337	0.585673	1.00299
0.04 0.02 1.0 1.0	0.597034	0.402966	-0.01295	0.457904	0.463279	1.00315

mostrado na Tabela 5.1 e a sua taxa de erro (TFN) é baixa e com valores altos para a Revocação mostrando que a fração de segmentos relevantes encontrados é alta.

Na Tabela 5.3 é mostrado os 10 melhores resultados ordenados de forma decrescente pelo Coeficiente de Correlação de Matthews (CCM), depois de forma decrescente pela Taxa de Verdadeiros Positivos (TVP) e por fim de forma crescente pela taxa de erro (TFN). É possível notar que o melhor valor de CCM indica uma relação positiva com a avaliação ponto a ponto, sem contar que seu TVP indica que a fração de segmentos relevantes encontrados é alta.

Mas observa-se que o segundo resultado aparece como o primeiro na Tabela 5.2 e que possui um TVP maior, uma taxa de erro (TFN) menor e uma leve mudança no CCM, indicando que o segundo resultado é melhor, uma vez que, é mais importante ter um valor alto de segmentos relevantes em detrimento de diminuição do CCM.

Após os resultados o conjunto de parâmetros escolhido para a segmentação utilizando o método Silence Removal (semi-supervisionado) foi:

- Window Size: 0.01
- Window Step: 0.005
- Smooth Window: 0.1

Tabela 5.2: Os melhores resultados do erro borda a borda (F1) do método Silence Removal (os valores de parâmetros estão na ordem: Window Size, Window Step, Smooth Window, Weight).

Parâmetros	TFN	CCM	Precisão	Revocação	F1	AEER
0.01 0.005 0.1 0.2	0.171677	**0.307047**	0.567921	0.828323	**0.673659**	1.597844
0.01 0.01 0.1 0.2	0.164796	0.288021	0.556359	0.835204333	0.667638	1.734345
0.01 0.005 0.2 0.2	0.144742	0.272706	0.546182	0.855257667	0.666461	1.569562
0.01 0.005 0.04 0.1	0.151571	0.272869	0.547814	0.848429	0.665581	1.888452
0.01 0.005 0.05 0.1	**0.137493**	0.263673	0.54043	**0.862506667**	0.664305	1.920915
0.01 0.01 0.2 0.2	0.138424	0.262917	0.540212	0.861575667	0.663912	1.725604
0.01 0.01 0.2 0.3	0.215109	0.296524	0.574366	0.784890667	0.663129	1.44012
0.01 0.005 0.02 0.1	0.198784	0.284428	0.563717	0.801216	0.661622	1.819667
0.02 0.01 0.2 0.2	0.161888	0.261182	0.546265	0.838112	0.661222	1.8369
0.01 0.005 0.2 0.3	0.222956	0.292826	**0.574873**	0.777044	0.660568	**1.270675**

- Weight: 0.2

5.1.2 Escolha dos parâmetros para o método HMMTrain

Na Tabela 5.4 é mostrado os 10 melhores resultados utilizando o método HMMTrain (Supervisionado) ordenado pelo AEER de forma crescente.

O melhor resultado possui um AEER baixo, o que é bom, mas apesar disso o seu CCM está baixo também indicando uma relação quase aleatória entre os segmentos encontrados e os reais. Isso ocorre por que o AEER é baseado no número de segmentos que possuem pelo menos 50% dos segmentos reais, mas o que o CCM mostra é que os segmentos encontrados não batem com os segmentos reais de forma ponto a ponto.

E durante a avaliação ponto a ponto tem-se os 10 melhores resultados mostrados na Tabela 5.5 ordenados por Coeficiente de Correlação de Matthews (CCM) de forma decrescente, depois por Taxa de Verdadeiros Positivos (TVP) também de forma decrescente e finalmente por taxa de erro (TFN) de forma crescente.

O valor do AEER do melhor resultado da avaliação ponto a ponto possui pouca diferença entre o melhor resultado da Tabela 5.4, mas em contra partida possui uma Taxa de Verdadeiros Positivos (TVP) baixa se comparada com outros resultados da Tabela 5.5 e

Tabela 5.3: Os melhores resultados do erro ponto a ponto (CCM, TVP, TFN) do método Silence Removal (os valores de parâmetros estão na ordem: Window Size, Window Step, Smooth Window, Weight).

Parâmetros	TVP	TFN	CCM	Precisão	F1	AEER
0.01 0.01 0.1 0.3	0.731444	0.268556	**0.315065**	**0.597735**	0.657485	1.505193
0.01 0.005 0.1 0.2	0.828323	0.171677	0.307047	0.567921	**0.673659**	1.597844
0.01 0.01 0.2 0.3	0.784891	0.215109	0.296524	0.574366	0.663129	1.44012
0.01 0.005 0.2 0.3	0.777044	0.222956	0.292826	0.574873	0.660568	**1.270675**
0.01 0.01 0.1 0.2	0.835204	0.164796	0.288021	0.556359	0.667638	1.734345
0.01 0.005 0.02 0.1	0.801216	0.198784	0.284428	0.563717	0.661622	1.819667
0.01 0.01 0.05 0.2	0.781988	0.218012	0.279096	0.565694	0.656187	1.746036
0.01 0.01 0.04 0.2	0.760851	0.239149	0.272877	0.568055	0.650245	1.73995
0.01 0.005 0.04 0.1	0.848429	0.151571	0.272869	0.547814	0.665581	1.888452
0.01 0.005 0.2 0.2	**0.855258**	**0.144742**	0.272706	0.546182	0.666461	1.569562

uma taxa de erro (TFN) alta mostrando que há uma grande perda de sinal.

Na avaliação borda a borda foram coletados os 10 melhores resultados mostrados na Tabela 5.6 ordenados pela F-Medida (F1) de forma decrescente.

O melhor resultao possui F-Medida (F1) alto e uma Revocação alta também, mostrando que a porção de segmentos encontrados relevantes é alta, sem mencionar que a taxa de erro (TFN) é baixa mas em contrapartida perde-se em CCM e apesar dessa perda a troca é boa pois há pouca perda de sinal mesmo que os segmentos encontrados não possuem o mesmo tamanho do segmento real.

Após os resultados o conjunto de parâmetros escolhido para a segmentação utilizando o método HMMTrain (supervisionado) foi:

- Window Size: 0.5
- Window Step: 0.5

5.1.3 Escolha do melhor método para segmentação

A Tabela 5.7 compara os dois melhores resultados de cada método de segmentação. Os valores em negrito são os melhores para aquela métrica. A partir dessa tabela fica claro

Tabela 5.4: Os melhores resultados do erro evento a evento (AEER) do método HMMTrain (os valores de parâmetros estão na ordem: Window Size, Window Step).

Parâmetros	TVP	TFN	CCM	Precisão	F1	AEER
0.08 0.04	0.930562	0.069439	0.094215	0.3229645	0.478842	**1.023816**
1.0 0.5	**0.996661**	**0.00334**	0.04752	0.255649	0.406911	1.030227
0.5 0.5	0.968071	0.031929	0.170226	0.494104333	**0.653398**	1.176593
0.2 0.2	0.447341	0.552659	**0.220613**	**0.638079**	0.480579	1.366133
0.3 0.3	0.546845	0.453155	0.215393	0.602089	0.551441	1.693908
0.4 0.4	0.681504	0.318496	0.203809	0.559099	0.606979	1.710854
0.3 0.15	0.266149	0.733851	0.050475	0.281822333	0.240749	1.851079
0.2 0.1	0.249089	0.750911	0.028277	0.259523667	0.227422	1.928888
0.4 0.2	0.37478	0.62522	0.050018	0.257971	0.281043	1.931664
0.5 0.25	0.596247	0.403753	0.037616	0.239211333	0.331621	2.09979

que o método supervisionado HMMTrain possui a menor taxa de erro e a maior taxa de revocação indicando um ótimo resultado na avaliação de ponto a ponto em comparação ao método Silence Removal, ou seja, o método HMMTrain encontra mais segmentos inteiros iguais ao segmentos reais do que o Silence Removal.

Apesar do método HMMTrain possuir essa vantagem sobre o método Silence Removal, o seu F1 e CCM é menor comparado com o do método Silence Removal indicando um pior resultado na avaliação borda a borda, ou seja, o método Silence Removal tem uma melhor habilidade de encontrar segmentos parciais mas com informações relevantes devido ao seu F1 em detrimento de um aumento da taxa de erro (TFN).

Considerando que o foco da ferramenta BovChewing é o seu uso para análise do comportamento bovino e suas atividade diárias e por tratar de dados que podem sofrer diversas interferências, a escolha do método de segmentação não pode partir apenas dos melhores resultados mas também do fato que o método HMMTrain é supervisionado e requer um conjunto de treinamento e o Silence Removal utiliza o próprio áudio para aprender e segmentar removendo os segmentos não importantes. Levando isso em consideração o método Silence Removal foi escolhido para compor o BovChewing por não ter a necessidade de um conjunto de treinamento que pode não ser o suficiente para segmentar novas gravações.

Tabela 5.5: Os melhores resultados do erro ponto a ponto (CCM, TVP, TFN) do método HMMTrain (os valores de parâmetros estão na ordem: Window Size, Window Step).

Parâmetros	TVP	TFN	CCM	Precisão	F1	AEER
0.2 0.2	0.447341	0.552659	**0.220613**	**0.638079**	0.480579	1.366133
0.3 0.3	0.546845	0.453155	0.215393	0.602089	0.551441	1.693908
0.4 0.4	0.681504	0.318496	0.203809	0.559099	0.606979	1.710854
0.5 0.5	0.968071	0.031929	0.170226	0.494104	**0.653398**	1.176593
0.08 0.04	0.930562	0.069439	0.094215	0.322965	0.478842	**1.023816**
0.3 0.15	0.266149	0.733851	0.050475	0.281822	0.240749	1.851079
0.4 0.2	0.37478	0.62522	0.050018	0.257971	0.281043	1.931664
1.0 0.5	**0.996661**	**0.00334**	0.04752	0.255649	0.406911	1.030227
0.5 0.25	0.596247	0.403753	0.037616	0.239211	0.331621	2.09979
0.2 0.1	0.249089	0.750911	0.028277	0.259524	0.227422	1.928888

5.2 Classificador

A Figura 5.1 mostra o efeito do parâmetro K sobre a performance (acurácia e taxa de erro) do modelo KNN sobre a base de dados descrita na Seção 4.1. A Acurácia se manteve em 88% com o K variando de 3 a 7, mas no último resultado a acurácia diminui um pouco mostrando que o aumento do parâmetro pode aumentar os falsos positivos diminuindo a eficácia do modelo mesmo que isso signifique uma diminuição da taxa de erro.

A Figura 5.2 mostra os resultados entre os dois Kernel(Linear e RBF) e os diversos valores dos seus parâmetros (C e G) e o seu efeito sobre a performance (acurácia e taxa de erro) usando o algoritmo LibSVM. Na Figura 5.2(a) a acurácia manteve-se na faixa de 86% 87% mas com uma taxa de erro alta comparada com o algoritmo KNN. Na Figura 5.2(b) a acurácia começou baixa com o $C = 0.001$ mas conforme foi aumentando o valor de C a variação foi pequena e isso também se refletiu na taxa de erro não tendo muita diferença entre $C = 0.01$ e $C = 10$, indicando que talvez pela natureza dos dados o método LibSVM não demonstra eficácia na tarefa de classificação.

Na Figura 5.3 é possível ver o efeito do parâmetro L (Learning rate) do algoritmo Perceptron de Multicamadas sobre a performance da classificação. Com pequenas mudanças no parâmetro L não há uma variação grande na acurácia mas há uma melhora na taxa de erro e apesar da mudança ser pequena, todas as taxas de erro são menores que os outros

Tabela 5.6: Os melhores resultados do erro borda a borda (F1) do método HMMTrain (os valores de parâmetros estão na ordem: Window Size, Window Step).

Parâmetros	TFN	CCM	Precisão	Revocação	F1	AEER
0.5 0.5	**0.031929**	0.170226	0.494104333	0.968071	**0.653398**	1.176593
0.4 0.4	0.318496	0.203809	0.559099	0.681503667	0.606979	1.710854
0.3 0.3	0.453155	0.215393	0.602089	0.546845	0.551441	1.693908
0.2 0.2	0.552659	**0.220613**	**0.638079**	0.447340667	0.480579	1.366133
0.08 0.04	0.069439	0.094215	0.3229645	0.9305615	0.478842	**1.023816**
1.0 0.5	0.00334	0.04752	0.255649	**0.9966605**	0.406911	1.030227
0.5 0.25	0.403753	0.037616	0.239211333	0.596246667	0.331621	2.09979
0.4 0.2	0.62522	0.050018	0.257971	0.374780333	0.281043	1.931664
0.3 0.15	0.733851	0.050475	0.281822333	0.266148667	0.240749	1.851079
0.2 0.1	0.750911	0.028277	0.259523667	0.249089333	0.227422	1.928888

Tabela 5.7: Comparação dos métodos de segmentação.

Método	Parâmetros	TFN	CCM	Precisão	Revocação	F1	AEER
HMMTrain	0.5 0.5	**0.031929**	0.1702263	0.494104333	**0.968071**	0.653398	**1.176593**
HMMTrain	0.4 0.4	0.318496	0.2038093	0.559099	0.681504	0.606979	1.710854
Silence Removal	0.01 0.005 0.1 0.2	0.171677	0.307047	0.567921333	0.828323	**0.673659**	1.597844
Silence Removal	0.01 0.01 0.1 0.3	0.268556	**0.3150653**	**0.597735**	0.731444	0.657485	1.505193

algoritmos.

A Tabela 5.8 mostra uma comparação dos resultados entre os algoritmos e seus parâmetros, os textos em negritos representam os melhores resultados para aquela métrica. O algoritmo que obteve o melhor resultado foi o Perceptron de Multicamadas com L = 0.05 e M = 0.1 com 91,5038% de acurácia e 17.61% de taxa de erro. Apesar de só a acurácia não ser a melhor métrica para avaliar um modelo de aprendizado de máquina, mas basta observar que o seu F-Medida é o maior entre todos os outros algoritmos, mostrando que sua Revocação e Precisão estão altas indicando que a fração de segmentos classificados corretamente e relevantes é alta. Quando observado o valor Coeficiente de Correlação de Mathews(CCM) com um valor de 0.9 indica uma predição quase perfeita e isso é refletido também da Área ROC com um valor próximo à 1.0.

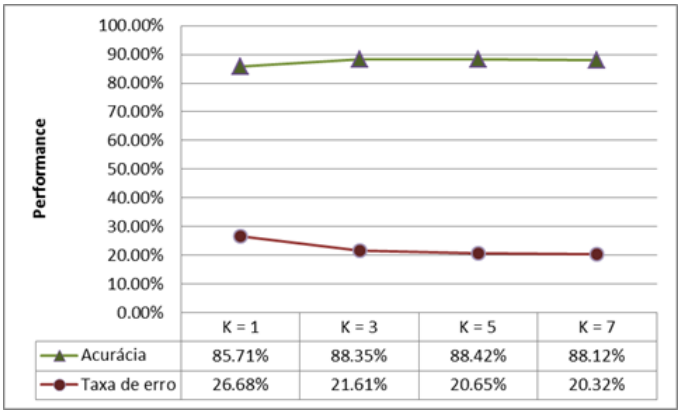
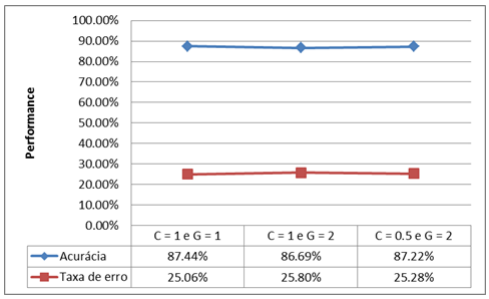
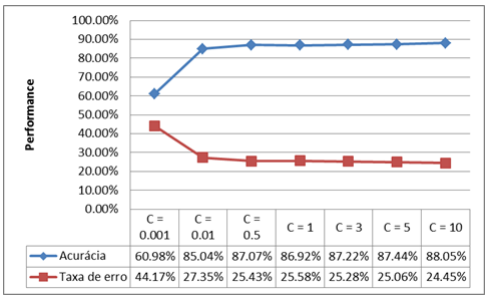


Figura 5.1: Efeito de K sobre as métricas de performance do modelo KNN.



(a) Efeito dos parâmetros C e G sobre a performance do modelo LibSVM com o Kernel RBF



(b) Efeito do parâmetro C sobre a performance do modelo LibSVM com o Kernel Linear

Figura 5.2: Efeito dos tipos de Kernels sobre a performance do modelo.

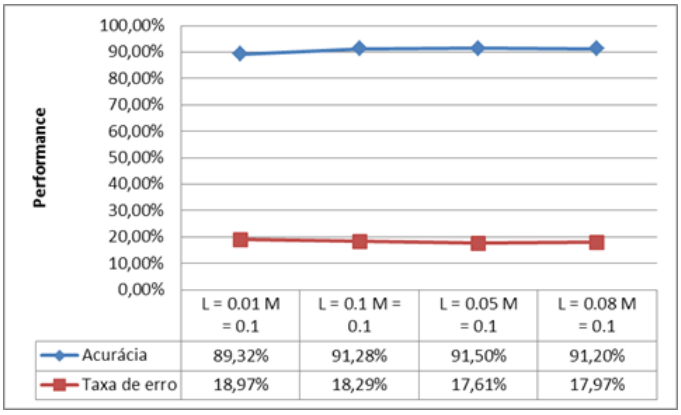


Figura 5.3: Efeito de L sobre as métricas de performance do modelo Perceptron de Multicamadas.

Tabela 5.8: Resultados dos experimentos dos classificadores

Classificador	**Parâmetro**	**Taxa de VP**	**Taxa de FP**	**Precisão**	**Revocação**	**F-Medida**	**CCM**	**Área ROC**
KNN	K = 1	0,857	0,029	0,856	0,857	0,857	0,83	0,914
	K = 3	0,883	0,024	0,88	0,883	0,881	0,86	0,957
	K = 5	0,884	0,024	0,878	0,884	0,879	0,86	0,968
	K = 7	0,881	0,025	0,875	0,881	0,877	0,85	0,972
LibSVM	Kernel Linear e C = 0.001	0,61	0,247	0,537	0,61	0,548	0,41	0,682
	Kernel Linear e C = 0.01	0,85	0,043	0,788	0,85	0,816	0,79	0,904
	Kernel Linear e C = 0.5	0,871	0,032	0,856	0,871	0,854	0,83	0,919
	Kernel Linear e C = 1	0,869	0,031	0,854	0,869	0,855	0,83	0,919
	Kernel Linear e C = 3	0,872	0,03	0,863	0,872	0,864	0,84	0,921
	Kernel Linear e C = 5	0,874	0,029	0,866	0,874	0,868	0,84	0,923
	Kernel Linear e C = 10	0,88	0,028	0,873	0,88	0,874	0,85	0,926
	Kernel RBF, C = 1 e G = 1	0,874	0,026	0,867	0,874	0,867	0,85	0,924
	Kernel RBF, C = 1 e G = 2	0,867	0,028	0,856	0,867	0,857	0,84	0,92
	Kernel RBF, C = 0.5 e G = 2	0,872	0,03	0,85	0,872	0,841	0,83	0,921
MultilayerPerceptron	L = 0.01 M = 0.1	0,893	0,023	0,89	0,893	0,891	0,87	0,985
	L = 0.1 M = 0.1	0,913	0,018	0,912	0,913	0,912	0,9	0,987
	L = 0.05 M = 0.1	**0,915**	**0,017**	**0,915**	**0,915**	**0,915**	**0,9**	**0,988**
	L = 0.08 M = 0.1	0,912	0,018	0,912	0,912	0,912	0,89	0,988

Como foi dito no Capítulo 4, foi aplicado o *macro-averaging* sobre as métricas. Na Tabela 5.9 é possível ver os valores das métricas por classes do algoritmo Perceptron de Multicamadas com L = 0.05 e M = 0.1. A classe que obteve pior resultado foi o evento de Bocada com Mastigação obtendo um valor de CCM baixo comparado à outra classe, mas mesmo com esse valor esse algoritmo obteve os melhores resultados comparados à outros algoritmos do experimento como pode ser visto nos relatórios no Apêndice A.

Como objetivo foi encontrar um conjunto de parâmetros e algoritmo para ser utilizado na ferramenta BovChewing a escolha dos parâmetros e a realização dos experimentos foram baseados em diversos trabalhos e foi constatado que o melhor algoritmo e conjunto de parâmetros que obteve os melhores resultado foi o Pecerptron de Multicamadas com L = 0.05 e M = 0.1, sendo assim, ele foi escolhido para compor a ferramenta BovChewing.

Tabela 5.9: Métricas por classe do algoritmo Perceptron de Multicamadas

Classe	TVP	TFP	Precisão	Revocação	F-Medida	CCM	ROC Area
Bocada	0.819	0.046	0.796	0.819	0.807	0.765	0.972
Mastigação	0.924	0.023	0.93	0.924	0.927	0.903	0.987
Bocada com Mastigação	0.628	0.033	0.655	0.628	0.641	0.606	0.962
Ruminação	1	0	1	1	1	1	1
Média das métricas	0.915	0.017	0.915	0.915	0.915	0.9	0.9

Capítulo 6

BovChewing

O objetivo do BovChewing é fornecer um leque de funcionalidades para ajudar em pesquisas e análises de dados acústicos envolvendo bovinos com ênfase em detecção de sons ingestivos (bocados, ruminação, mastigação e eventos compostos de apreensão com mastigação) e sua classificação. O BovChewing tem como príncipio o fácil uso e projeto de código extensivo.

A ferramenta é escrita em Python, uma linguagem de alto nível que atrai novos adeptos principalmente no meio científico e acadêmico. Uma das vantagens da linguagem que fez com que ela fosse escolhida para o uso na ferramenta foi seu ótimo balanço entre programação de alto nível e baixo nível e com poucas linhas de código tem-se um programa com desempenho otimizado.

BovChewing é uma ferramenta de código aberto e hospedado no GitBucket, inicialmente em um repositório privado mas com planos futuros de torná-lo público.

A Figura 6.1 mostra o modelo conceitual da ferramenta, com seus módulos responsáveis por executar suas funcionalidades.

Este Capítulo é dividido em mostrar as funcionalidades, como usar a ferramenta e a estrutura do código.

6.1 Funcionalidades

O BovChewing possui um série de funcionalidades tais como:

- Extração de características e criação de base de dados para treinamento utilizando segmentos de áudio de eventos digistivos.
- Testar base de dados novas e gerar relatórios de desempenho da ferramenta na tarefa de classificação.

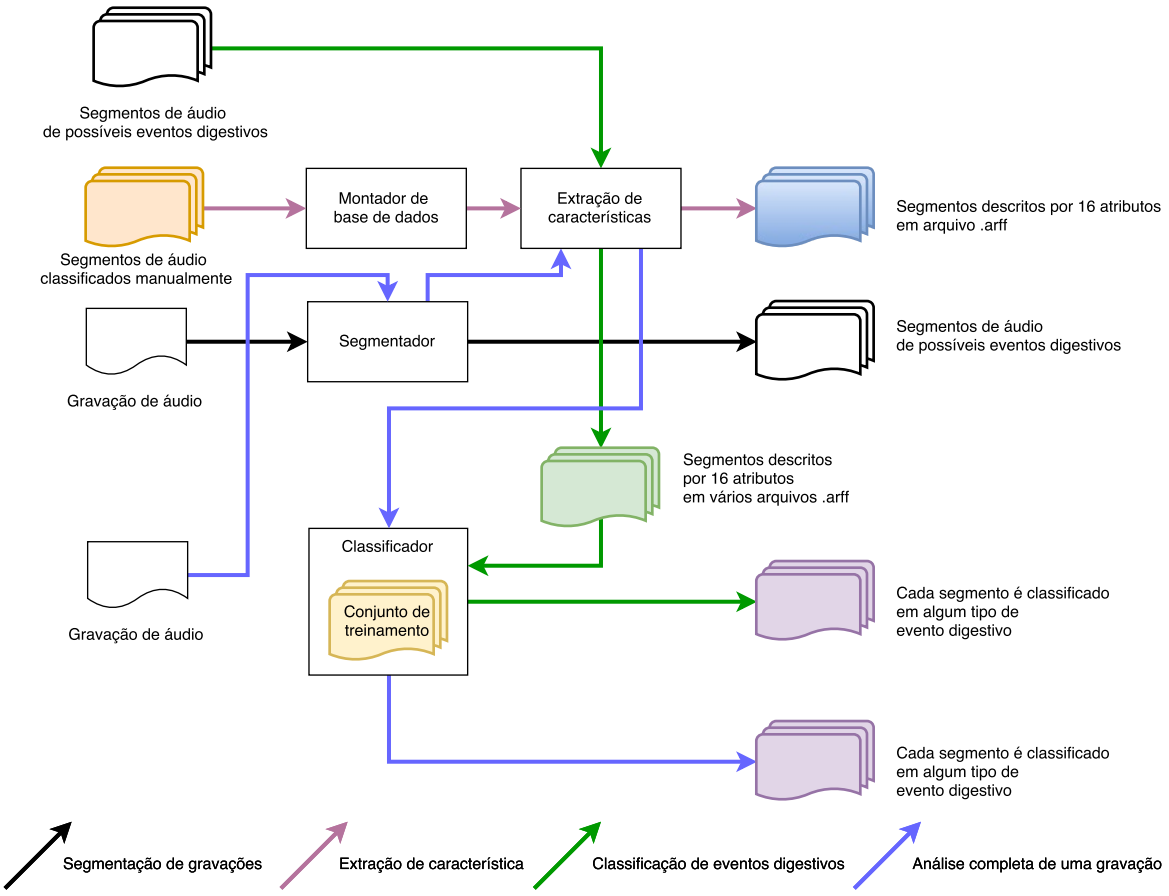


Figura 6.1: Modelo conceitual do BovChewing, onde vê-se as relações entre os módulos para realizar todas as funções da ferramenta.

- Segmentar gravações de áudio em possíveis eventos ingestivos utilizando o algorimto escolhido pelos experimentos do Capítulo 5.
- Classificar um conjunto de segmentos em eventos ingestivos usando o modelo classificador escolhido pelos experimentos do Capítulo 5.
- Calcular de uma gravação de áudio o número de Bocados, de mastigação, de ruminação e de eventos composto de apreensão e mastigação.
- Duração total de cada tipo de evento ingestivo em uma gravação e um gráfico para representar isso.

A interface do programa é por linha de comando, utilizando o bash do Linux ou o prompt de comandos do Windows.

Pré-requisitos

Para o seu completo funcionamento o ambiente onde o BovChewing será instalado deve possuir algumas ferramentas previamente instaladas.

- Python 2.7.
- As bibliotecas Numpy, Matplotlib, Scipy, Sklearn, Hmmlearn, Simplejson, Eyed3 instaladas.

Como cada sistema operacional possui uma maneira distinta de instalar o núcleo do Python e as bibliotecas fica a cargo do usuário preparar esse ambiente.

6.2 Modo de uso

A ferramenta possui uma interface amigável com comandos simples e alguns argumentos. A seguir a lista de comandos para cada funcionalidades e exemplo de uso.

6.2.1 Extração de característica

Caso o usuário possuir uma coleção de segmentos de áudio classificados manualmente e queira extrair características gerando assim um arquivo .arff para estudos e validação da ferramenta o Código 6.1 mostra o comando que deve ser usado:

```
python bovChewing.py mountDataset --folders [folder1] [folder2] --labels [
    class-of-folder1] [class-of-folder2] --model [name-of-dataset]
```

Código 6.1: Montagem de base de dados com segmentos classificados manualmente

O script espera como argumentos uma lista de pastas contendo os segmentos, onde cada pasta contém apenas segmentos do mesmo tipo de evento ingestivo, espera-se também uma lista de rótulos para cada pasta e por fim o nome do arquivo gerado por ele. O Código 6.2 mostra um exemplo de uso, onde temos 4 pastas - B, M, BM e R - contendo segmentos das classes Bocado, Mastigação, Bocado composto com Mastigação e Ruminação respectivamente. De forma abreviada optou-se por rotular as classes em B, M, BM e R respectivamente.

```
python bovChewing.py mountDataset --folders train/B train/M train/BM train/
    R --labels B M BM R --model BovineBehavior
```

Código 6.2: Exemplo de uso do mountDataset

A Figura 6.2 mostra um fragmento do documento gerado pelo script. No cabeçalho é possível ver quais características foram retiradas dos segmentos e quais sãos classes que aquele dataset possui. Todo o documento segue as especificações dos arquivos ARFF[1].

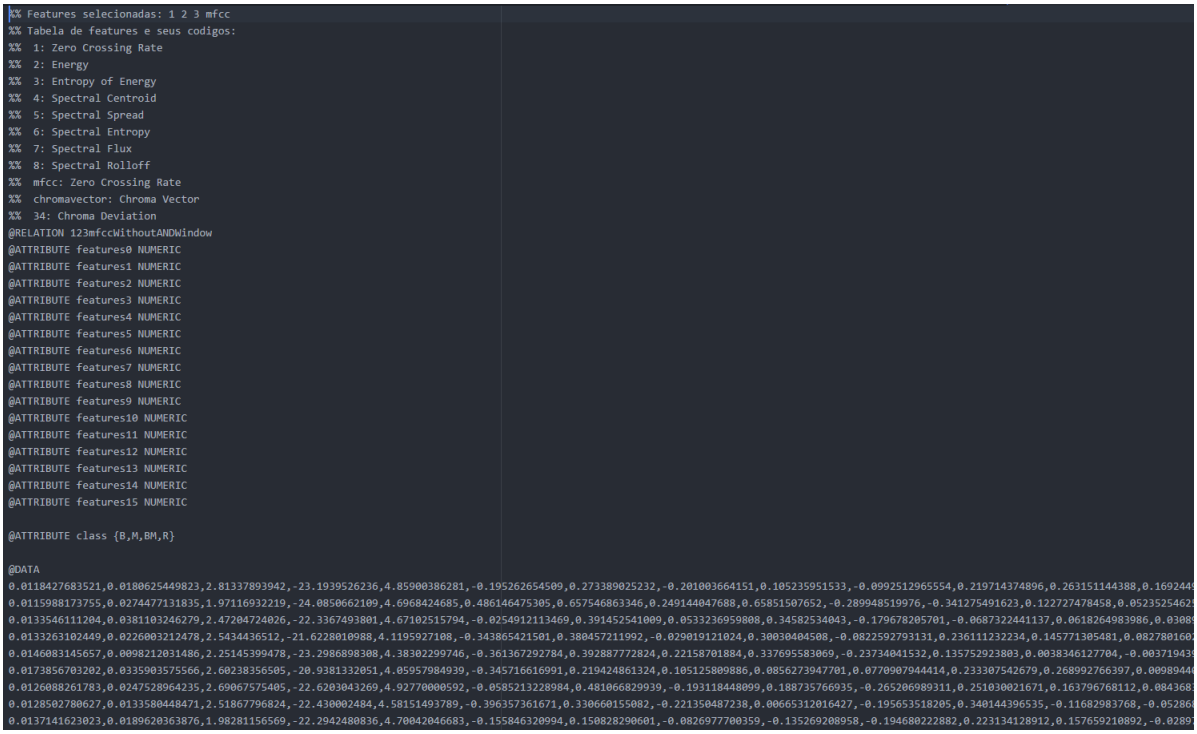


Figura 6.2: Saída do comando mountDataset

6.2.2 Avaliação de base de dados

No momento que a ferramenta permite a geração de base dados por meio de segmentos classificados manualmente fica claro a necessidade de uma funcionalidade para a avaliação da base de dados gerada. O BovChewing é capaz de realizar os mesmos testes do Capítulo 5 mostrando todas as métricas para melhoramento continuo da ferramenta e da generalização do modelo. O Código 6.3 mostra como avaliar um modelo.

```
python bovChewing.py evaluateDataset --arrf [file.arff]
```

Código 6.3: Avaliação de uma base de dados

A Figura 6.3 mostra um fragmento do relatório da avaliação de uma base de dados.

Para a avaliação da base de dados é montado três modelos de classificadores (Knn, SVM e Perceptron de Multicamadas) usando a Tabela 4.2 para configurar os modelos.

6.2.3 Segmentação de gravações

A fase da coleta de dados é realizada de maneira ininterrupta, ou seja, a gravação possui segmentos importantes e outros é apenas ruídos, tendo isso em vista é necessário dividir essa gravação em segmentos de possíveis eventos ingestivos.

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

Correctly Classified Instances        1213               91.203  %
Incorrectly Classified Instances       117                8.797  %
Kappa statistic                          0.8677
Mean absolute error                      0.0592
Root mean squared error                  0.1797
Relative absolute error                 17.7698 %
Root relative squared error             44.0457 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,811    0,051    0,775      0,811    0,793      0,746    0,970     0,889     B
               0,918    0,024    0,926      0,918    0,922      0,897    0,987     0,961     M
               0,628    0,031    0,673      0,628    0,650      0,616    0,960     0,698     BM
               1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     R
Weighted Avg.  0,912    0,018    0,912      0,912    0,912      0,894    0,988     0,943

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 193  14  31   0 |   a = B
  21 302   6   0 |   b = M
  35  10  76   0 |   c = BM
   0   0   0 642 |   d = R
```

Figura 6.3: Exemplo de um relatório do comando evaluateDataset

O BovChewing possui uma funcionalidade capaz de segmentar a gravação em segmentos de possíveis eventos ingestivos, levando em consideração os resultados dos experimentos do Capítulo 5. O Código 6.4 mostra o comando para segmentar um arquivo de áudio.

```
python bovChewing.py segment --file [recording.wav] --folder [name-folder-output]
```

Código 6.4: Segmentação de um arquivo de áudio

O resultado são vários arquivos que representam os segmentos de possíveis eventos ingestivos.

6.2.4 Classificação de eventos ingestivos

Caso o usuário possua segmentos de possíveis eventos ingestivos e queira classificar usando a base de dados do próprio BovChewing ou uma base de dados criada a partir de segmentos classificados manualmente como conjunto de treinamento, o BovChewing oferece um comando para isso, o Código 6.5 mostra como realizar essa tarefa.

```
python bovChewing.py classify --folder [folder-with-segments] [--dataset [file.arff]]
```

Código 6.5: Classificação de segmentos utilizando uma base de dados

O script recebe como parâmetro a pasta contendo os segmentos a serem classificados, todos devem estar em formato .wav e o parâmetro –dataset é opcional que quando há um valor atribuído, este valor é nome do arquivo do tipo .arff[1] que é usado como base de dados para a construção do conjunto de treinamento do modelo de classificação, caso não tenha nenhum valor atribuído para o parâmetro a base de dados padrão do BovChewing é usado para a construção do conjunto de treinamento. A saída é um arquivo que para cada segmento é informado o tipo de evento que ele pertence.

6.2.5 Análise completa de uma gravação

A funcionalidade mais completa do BovChewing, é que a partir de um arquivo de áudio de uma gravação das atividades de um bovino é capaz de segmentá-lo em eventos importantes e então classificá-los em eventos ingestivos e ao final gerar um relatório contendo a quantidade de cada tipo de evento detectado e a sua duração total, com a possibilidade de personalizar a base de dados utilizada para o conjunto de treinamento do modelo classificador. O Código 6.6 mostra o comando para realizar tal tarefa.

```
1 python bovChewing.py segmentAndClassify --file [recording.wav] [--dataset [
    file.arff]]
```

Código 6.6: Classificação de segmentos utilizando uma base de dados avaliada manualmente

A Figura 6.4 mostra a saída da avaliação de uma gravação, na Figura 6.4(a) há um sumário mostrando algumas informações da avaliação, como a duração dos segmentos de eventos ingestivos encontrados, o número total de segmentos e a distribuição dos segmentos entre cada tipo de evento ingestivo após a classificação. Já na Figura 6.4(b) é mostrado um gráfico com a distribuição dos tipos de eventos dentro da gravação.

6.2.6 Estrutura do código

BovChewing é composto por bibliotecas externas e algumas classes auxiliares, mas o núcleo da ferramenta é composta por cinco classes descritas a seguir:

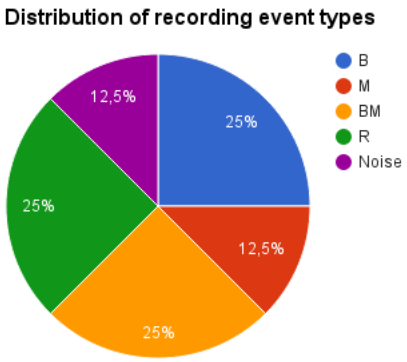
- Classifier.py: Implementa todos os métodos para classificação de segmentos.
- Segmenter.py: Implementa todos os métodos para a segmentação de arquivos de áudio.
- Metric.py: Implementa todos os métodos que avalia uma base de dados.
- Dataset.py: Implementa todos os métodos para a criação e gerenciamento de uma base de dados.

```
=== Summary ===

Record length               00:04:00
Number of segments          60
Sum of segments duration    00:03:30

=== Info by class ===
Class   Sum of segments duration    Number of segments Recording percentage
B       00:01:00                    10                 25%
M       00:00:30                    5                  12.5%
BM      00:01:00                    20                 25%
R       00:01:00                    25                 25%
Noise   00:00:30                    0                  12.5%
```

(a) Relatório mostrando o resultado da avaliação completa de uma gravação



(b) Gráfico mostrando a distribuição de eventos dentro de uma gravação

Figura 6.4: Saída do comando segmentAndClassify

- BovChewing.py: Implementa a classe principal responsável por preparar os argumentos dos comandos e instanciar as classes necessárias.

Capítulo 7

Conclusão e trabalhos futuros

Este trabalho apresenta o BovChewing - uma ferramenta que tem como objetivo segmentar e classificar dados bioacústicos de bovinos em eventos digestivos de forma automática - que tem como foco o uso de uma linguagem eficiente e de alto desepenho.

O BovChewing é capaz de segmentar e classificar com uma alta acurácia (63% na atividade de segmentação e 91.5% na atividade de classificação), calculando a duração de cada atividade (Bocado, Mastigação, Ruminação e evento composto de Bocado e Mastigação).

Apesar da acurácia do BovChewing ser baixa comparada a outros métodos de trabalhos anteriores como descrito no Capítulo 3, as análises dos resultados mostraram uma alta confiabilidade mesmo tendo uma base de dados com gravações obtidas em ambientes com condições menos controladas diferentemente das base de dados de outros trabalhos, adicionalmente a ferramenta é capaz de segmentar e classificar eventos de ruminação além das outras: bocado, mastigação e eventos composto de bocado e mastigação.

A ferramenta tem como um de seus objetivos ser de fácil uso, ou seja, sem a necessidade de ajustes de parâmetros. Para alcançar esse objetivo foi realizado vários experimentos a fim de selecionar e validar o conjunto de método e parâmetros do BovChewing. Outra contribuição importante é a possibilidade de treinamento de maneira dinâmica conforme o uso da ferramenta aumenta pela comunidade. Contudo a ferramenta comparada à CBRTA[12] possui algumas limitações tais como:

- BovChewing não processa os dados coletados em tempo real.
- O custo computacional do BovChewing não foi avaliado como foi por Chelotti et al.[12].

Mas em contrapartida o BovChewing é capaz de identificar e classificar eventos de Ruminação.

Sugerimos os seguintes tópicos como trabalhos futuros:

- Melhora do método de segmentação com avaliações do modelo de classificação usando os segmentos automáticos.
- Um estudo sobre o custo computacional da ferramenta.
- Um estudo sobre a possibilidade da inferência da massa consumida pelo animal por meio de aprendizado de máquina utilizando dados bioacústicos.
- Implementação do envio dos dados coletados por meio de redes sem fio e seu processamento na nuvem para garantir uma análise em tempo real.
- Integração do BovChewing com outras ferramentas de análise de saúde animal, tais como: leitura da temperatura e ganho de massa.

Referências Bibliográficas

[1] Attribute-relation file format (arff). `http://www.cs.waikato.ac.nz/ml/weka/arff.html`. Accessed: 2016-09-21.

[2] P. U. Alkon, Y. Cohen, and P. A. Jordan. Towards an acoustic biotelemetry system for animal behavior studies. *Journal of Wildlife Management*, 53(3):658–662, 1989.

[3] WG Allden and IA McDWhittaker. The determinants of herbage intake by grazing sheep: the interrelationship of factors influencing herbage intake and availability. *Crop and Pasture Science*, 21(5):755–766, 1970.

[4] Juan J. Noda Arencibia, Carlos M. Travieso, David Sánchez-Rodríguez, Malay Kishore Dutta, and Garima Vyas. Automatic classification of frogs calls based on fusion of features and svm. In Manish Parashar, Tirumale Ramesh, Jaric Zola, Nanjangud C. Narendra, Kishore Kothapalli, J. Amudha, Purushotham Bangalore, Deepa Gupta, Animesh Pathak, Sanjay Chaudhary, K. V. Dinesha, and Sushil K. Prasad, editors, *IC3*, pages 59–63. IEEE Computer Society, 2015.

[5] Derek W Bailey, John E Gross, Emilio A Laca, Larry R Rittenhouse, Michael B Coughenour, David M Swift, and Phillip L Sims. Mechanisms that result in large herbivore grazing distribution patterns. *Journal of Range Management*, 49(5):386–400, 1996.

[6] Derek W Bailey and Frederick D Provenza. Mechanisms determining large-herbivore distribution. *Resource ecology*, pages 7–28, 2008.

[7] IA Basheer and M Hajmeer. Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1):3–31, 2000.

[8] CA Cangiano, HH Fernández, and JR Galli. Conpast 3.0: programa de computación para la estimación del consumo de bovinos en pastoreo. *La Barrosa, Buenos Aires*, 1999.

[9] PC de F Carvalho, HL Gonda, MH Wade, JC Mezzalira, MF Do Amaral, EN Gonçalves, DT do Santos, L Nadin, and CHEC Poli. Características estruturais do pasto eo consumo de forragem: o quê pastar, quanto pastar e como se mover para encontrar o pasto. *Manejo estratégico da pastagem*, 4(2008):101–130, 2008.

[10] PCF Carvalho and A de MORAES. Comportamento ingestivo de ruminantes: bases para o manejo sustentável do pasto. *Manejo sustentável em pastagem*, 1:1–20, 2005.

[11] PCF Carvalho, JK Trindade, SC Da Silva, C Bremm, JC Mezzalira, C Nabinger, MF Amaral, IJ Carassai, RS Martins, TCM Genro, et al. Consumo de forragem por animais em pastejo: analogias e simulações em pastoreio rotativo. In *Proceedings of the XXV symposium on manejo de pastagens'.(Eds SC Da Silva, CGS Pedreira, JC Moura, VP Faria) pp*, pages 61–93, 2009.

[12] José O Chelotti, Sebastián R Vanrell, Diego H Milone, Santiago A Utsumi, Julio R Galli, H Leonardo Rufiner, and Leonardo L Giovanini. A real-time algorithm for acoustic monitoring of ingestive behavior of grazing cattle. *Computers and Electronics in Agriculture*, 127:64–75, 2016.

[13] William M. Clapham, James M. Fedders, Kim Beeman, and James P.S. Neel. Acoustic monitoring system to quantify ingestive behavior of free-grazing cattle. *Computers and Electronics in Agriculture*, 76(1):96–104, 2011.

[14] Juan Gabriel Colonna, Marco Cristo, Mario Salvatierra Júnior, and Eduardo Freire Nakamura. An incremental technique for real-time bioacoustic signal segmentation. *Expert Systems with Applications*, 42(21):7367 – 7374, 2015.

[15] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[16] Júlio K Da Trindade, Cassiano E Pinto, Fabio P Neves, Jean C Mezzalira, Carolina Bremm, Teresa CM Genro, Marcelo R Tischler, Carlos Nabinger, Horacio L Gonda, and Paulo CF Carvalho. Forage allowance as a target of grazing management: implications on grazing time and forage searching. *Rangeland Ecology & Management*, 65(4):382–393, 2012.

[17] Delagarde, Rémy, Caudal, Jean-Pierre, and Peyraud, Jean-Louis. Development of an automatic bitemeter for grazing cattle. *Ann. Zootech.*, 48(5):329–339, 1999.

[18] Li Deng and Douglas O'Shaughnessy. *Speech processing: a dynamic and optimization-oriented approach*. CRC Press, 2003.

[19] K Ekštein and T Pavelka. Entropy and entropy-based features in signal processing. In *Proceedings of PhD workshop systems & control*, 2004.

[20] Peter Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Cambridge University Press, New York, NY, USA, 2012.

[21] J.R. Galli, C.A. Cangiano, M.W. Demment, and E.A. Laca. Acoustic monitoring of chewing and intake of fresh and dry forages in steers. *Animal Feed Science and Technology*, 128(1-2):14 – 30, 2006.

[22] J. W. Gardner, E. L. Hines, F. Molinier, P. N. Bartlett, and T. T. Mottram. Prediction of health of dairy cattle from breath samples using neural network with parametric model of dynamic response of array of semiconducting gas sensors. *IEE Proceedings - Science, Measurement and Technology*, 146(2):102–106, Mar 1999.

[23] Theodoros Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. *PLoS ONE*, 10(12):1–17, 12 2015.

[24] Dimitrios Giannoulis, Dan Stowell, Emmanouil Benetos, Mathias Rossignol, Mathieu Lagrange, and Mark D Plumbley. A database and challenge for acoustic scene classification and event detection. In *21st European Signal Processing Conference (EUSIPCO 2013)*, pages 1–5. IEEE, 2013.

[25] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 1998.

[26] Jc Hodgson, DA Clark, and RJ Mitchell. Foraging behavior in grazing animals and its impact on plant communities. *Forage quality, evaluation, and utilization*, (foragequalityev):796–827, 1994.

[27] John Hodgson, Andrew W Illius, et al. *The ecology and management of grazing systems.* CAB international, 1996.

[28] Giuseppe Jurman, Samantha Riccadonna, and Cesare Furlanello. A comparison of mcc and cen error measures in multi-class prediction. *PLOS ONE*, 7(8):1–8, 08 2012.

[29] L. Klein, S.K. Baker, D.B. Purser, T. Zaknich, and A.C. Bray. *Telemetry to monitor sounds of chews during eating and rumination by grazing sheep.*, volume 20, page 423. Australian Society of Animal Production, 1994.

[30] Shashidhar G Koolagudi, Deepika Rastogi, and K Sreenivasa Rao. Identification of language using mel-frequency cepstral coefficients (mfcc). *Procedia Engineering*, 38:3391–3398, 2012.

[31] Laca and WallisDeVries. Acoustic measurement of intake and grazing behaviour of cattle. *Grass and Forage Science*, 55(2):97–104, 2000.

[32] EA Laca and IM Ortega. Integrating foraging mechanisms across spatial and temporal scales. In *International rangeland congress*, volume 5, pages 129–132, 1995.

[33] E.A. Laca, E.D. Ungar, and M.W. Demment. Mechanisms of handling time and intake rate of a large mammalian grazer. *Applied Animal Behaviour Science*, 39(1):3 – 19, 1994.

[34] D.H. Milone, H.L. Rufiner, J.R. Galli, E.A. Laca, and C.A. Cangiano. Computational method for segmentation and classification of ingestive sounds in sheep. *Computers and Electronics in Agriculture*, 65(2):228 – 237, 2009.

[35] Diego H. Milone, Julio R. Galli, Carlos A. Cangiano, Hugo L. Rufiner, and Emilio A. Laca. Automatic recognition of ingestive sounds of cattle based on hidden markov models. *Computers and Electronics in Agriculture*, 87:51 – 55, 2012.

[36] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[37] Shilo Navon, Amos Mizrach, Amots Hetzroni, and Eugene David Ungar. Automatic recognition of jaw movements in free-ranging cattle, goats and sheep, using acoustic monitoring. *Biosystems Engineering*, 114(4):474–483, 2013.

[38] Juan J. Noda, Carlos M. Travieso, and David SÃ¡nchez-RodrÃguez. Methodology for automatic bioacoustic classification of anurans based on feature fusion. *Expert Systems with Applications*, 50:100 – 106, 2016.

[39] P.D. Penning, A.J. Rook, and R.J. Orr. Patterns of ingestive behaviour of sheep continuously stocked on monocultures of ryegrass or white clover. *Applied Animal Behaviour Science*, 31(3):237–250, 1991.

[40] RA Reis, TF Bernardes, and GR Siqueira. Forragicultura: ciência, tecnologia e gestão dos recursos forrageiros. *Jaboticabal, 714p*, 2013.

[41] Martin Scaiano and Diana Inkpen. Getting more from segmentation evaluation. In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 362–366. Association for Computational Linguistics, 2012.

[42] Ervin Sejdić, Igor Djurović, and Jin Jiang. Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing*, 19(1):153–183, 2009.

[43] RL Senft, MB Coughenour, DW Bailey, LR Rittenhouse, OE Sala, and DM Swift. Large herbivore foraging and ecological hierarchies. *BioScience*, 37(11):789–799, 1987.

[44] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing Management*, 45(4):427 – 437, 2009.

[45] Yukinori Tani, Yasunari Yokota, Masato Yayota, and Shigeru Ohtani. Automatic recognition and classification of cattle chewing activity by an acoustic monitoring method with a single-axis acceleration sensor. *Computers and Electronics in Agriculture*, 92:54 – 65, 2013.

[46] Clifford Loh Ting Yuan and Dzati Athiar Ramli. *Frog Sound Identification System for Frog Species Recognition*, pages 41–50. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[47] Júlio Kuhn Da Trindade, Paulo César de Faccio Carvalho, Fabio Pereira Neves, Cassiano Eduardo Pinto, Horacio Leandro Gonda, Laura Beatriz Nadin, and Luis Henrique Silva Correia. Potencial de um método acústico em quantificar as atividades de bovinos em pastejo. *Pesquisa Agropecuária Brasileira*, 46:965–968, 08 2011.

[48] G. Vaca-Castaño and D. Rodriguez. Using syllabic mel cepstrum features and k-nearest neighbors to identify anurans and birds species. In *2010 IEEE Workshop On Signal Processing Systems*, pages 466–471, Oct 2010.

[49] Jie Xie, M. Towsey, A. Truskinger, P. Eichinski, Jinglan Zhang, and P. Roe. Acoustic classification of australian anurans using syllable features. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*, pages 1–6, April 2015.

Apêndice A

Relatórios dos experimentos dos classificadores

A.1 Algoritmo KNN com K = 1

123mfcc-knn-k1.dat

```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

IB1 instance-based classifier
using 1 nearest neighbours for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        1140               85.7143 %
Incorrectly Classified Instances       190               14.2857 %
Kappa statistic                          0.7852
Mean absolute error                      0.0724
Root mean squared error                  0.2668
Relative absolute error                 21.7606 %
```

```
Root relative squared error             65.4139 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,672    0,060    0,708      0,672    0,690      0,624    0,806     0,535     B
                 0,881    0,050    0,853      0,881    0,867      0,822    0,916     0,781     M
                 0,397    0,061    0,393      0,397    0,395      0,334    0,668     0,211     BM
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     R
Weighted Avg.    0,857    0,029    0,856      0,857    0,857      0,828    0,914     0,791

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 160  24  54   0    a = B
  19 290  20   0    b = M
  47  26  48   0    c = BM
   0   0   0 642    d = R
```

A.2 Algoritmo KNN com K = 3

123mfcc-knn-k3.dat

```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 3 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

IB1 instance-based classifier
using 3 nearest neighbours for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        1175               88.3459 %
Incorrectly Classified Instances       155               11.6541 %
Kappa statistic                          0.8245
Mean absolute error                      0.0735
Root mean squared error                  0.2161
Relative absolute error                 22.0847 %
Root relative squared error             52.9672 %
Total Number of Instances             1330
```

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

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0,773    0,065    0,722      0,773    0,746      0,689    0,918     0,705     B
                0,903    0,034    0,897      0,903    0,900      0,867    0,960     0,895     M
                0,430    0,041    0,510      0,430    0,466      0,420    0,801     0,353     BM
                1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     R
Weighted Avg.   0,883    0,024    0,880      0,883    0,881      0,859    0,957     0,862

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 184  17  37   0    a = B
  19 297  13   0    b = M
  52  17  52   0    c = BM
   0   0   0 642    d = R
```

A.3 Algoritmo KNN com K = 5

123mfcc-knn-k5.dat

```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 5 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

IB1 instance-based classifier
using 5 nearest neighbours for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        1176               88.4211 %
Incorrectly Classified Instances       154               11.5789 %
Kappa statistic                          0.8253
Mean absolute error                      0.0744
Root mean squared error                  0.2065
Relative absolute error                 22.3529 %
Root relative squared error             50.6241 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
```

```
                 0,798    0,069    0,717      0,798    0,755      0,700    0,940     0,760     B
                 0,915    0,035    0,896      0,915    0,905      0,874    0,971     0,917     M
                 0,355    0,036    0,494      0,355    0,413      0,371    0,847     0,384     BM
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     R
Weighted Avg.    0,884    0,024    0,878      0,884    0,879      0,858    0,968     0,880

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 190  16  32   0    a = B
  16 301  12   0    b = M
  59  19  43   0    c = BM
   0   0   0 642    d = R
```

A.4 Algoritmo KNN com K = 7

123mfcc-knn-k7.dat

```
=== Run information ===

Scheme:       weka.classifiers.lazy.IBk -K 7 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

IB1 instance-based classifier
using 7 nearest neighbours for classification


Time taken to build model: 0 seconds

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

Correctly Classified Instances        1172               88.1203 %
Incorrectly Classified Instances       158               11.8797 %
Kappa statistic                          0.8208
Mean absolute error                      0.0755
Root mean squared error                  0.2032
Relative absolute error                 22.6775 %
Root relative squared error             49.8066 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,773    0,071    0,705      0,773    0,737      0,678    0,945     0,783     B
                 0,921    0,035    0,896      0,921    0,909      0,878    0,975     0,926     M
                 0,355    0,038    0,483      0,355    0,410      0,365    0,871     0,405     BM
```

```
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     R
Weighted Avg.    0,881    0,025    0,875      0,881    0,877      0,854    0,972     0,889

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 184  18  36   0    a = B
  16 303  10   0    b = M
  61  17  43   0    c = BM
   0   0   0 642    d = R
```

A.5 Algoritmo LibSvm Linear e C = 0.001

123mfcc-libsvm-linear-c0001.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 0.001 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.49 seconds

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

Correctly Classified Instances         811               60.9774 %
Incorrectly Classified Instances       519               39.0226 %
Kappa statistic                          0.3465
Mean absolute error                      0.1951
Root mean squared error                  0.4417
Relative absolute error                 58.611  %
Root relative squared error            108.2917 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,282    0,033    0,650      0,282    0,393      0,356    0,624     0,312     B
                 0,310    0,216    0,321      0,310    0,315      0,095    0,547     0,270     M
                 0,000    0,000    0,000      0,000    0,000      0,000    0,500     0,091     BM
                 1,000    0,388    0,706      1,000    0,828      0,657    0,806     0,706     R
Weighted Avg.    0,610    0,247    0,537      0,610    0,548      0,405    0,682     0,472

=== Confusion Matrix ===
```

```
  a   b   c   d   <-- classified as
 67 146   0  25    a = B
  7 102   0 220    b = M
 29  70   0  22    c = BM
  0   0   0 642    d = R
```

A.6 Algoritmo LibSvm Linear e C = 0.01

123mfcc-libsvm-linear-c001.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 0.01 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.25 seconds

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

Correctly Classified Instances        1131               85.0376 %
Incorrectly Classified Instances       199               14.9624 %
Kappa statistic                          0.7713
Mean absolute error                      0.0748
Root mean squared error                  0.2735
Relative absolute error                 22.4732 %
Root relative squared error             67.056  %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,849    0,112    0,623      0,849    0,719      0,658    0,869     0,556     B
                 0,875    0,068    0,809      0,875    0,841      0,787    0,904     0,739     M
                 0,000    0,000    0,000      0,000    0,000      0,000    0,500     0,091     BM
                 0,998    0,013    0,986      0,998    0,992      0,985    0,993     0,985     R
Weighted Avg.    0,850    0,043    0,788      0,850    0,816      0,788    0,904     0,766

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 202  35   0   1    a = B
  34 288   0   7    b = M
  88  32   0   1    c = BM
```

```
   0   1   0 641    d = R
```

A.7 Algoritmo LibSvm Linear e C = 0.5

123mfcc-libsvm-linear-c05.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 0.5 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.27 seconds

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

Correctly Classified Instances        1158               87.0677 %
Incorrectly Classified Instances       172               12.9323 %
Kappa statistic                          0.8036
Mean absolute error                      0.0647
Root mean squared error                  0.2543
Relative absolute error                 19.4241 %
Root relative squared error             62.3413 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,828    0,099    0,646      0,828    0,726      0,665    0,864     0,565     B
                 0,915    0,037    0,891      0,915    0,903      0,870    0,939     0,836     M
                 0,149    0,018    0,450      0,149    0,224      0,220    0,565     0,144     BM
                 1,000    0,007    0,992      1,000    0,996      0,993    0,996     0,992     R
Weighted Avg.    0,871    0,032    0,856      0,871    0,854      0,833    0,919     0,800

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 197  21  19   1    a = B
  22 301   3   3    b = M
  86  16  18   1    c = BM
   0   0   0 642    d = R
```

A.8 Algoritmo LibSvm Linear e C = 1

123mfcc-libsvm-linear-c1.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.25 seconds

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

Correctly Classified Instances        1156               86.9173 %
Incorrectly Classified Instances       174               13.0827 %
Kappa statistic                          0.8016
Mean absolute error                      0.0654
Root mean squared error                  0.2558
Relative absolute error                 19.6499 %
Root relative squared error             62.7027 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,807    0,094    0,651      0,807    0,720      0,657    0,856     0,560     B
                 0,915    0,038    0,888      0,915    0,901      0,868    0,938     0,833     M
                 0,174    0,024    0,420      0,174    0,246      0,226    0,575     0,148     BM
                 1,000    0,006    0,994      1,000    0,997      0,994    0,997     0,994     R
Weighted Avg.    0,869    0,031    0,854      0,869    0,855      0,833    0,919     0,799

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 192  21  24   1    a = B
  20 301   5   3    b = M
  83  17  21   0    c = BM
   0   0   0 642    d = R
```

A.9 Algoritmo LibSvm Linear e C = 3

123mfcc-libsvm-linear-c3.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 3.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.3 seconds

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

Correctly Classified Instances        1160               87.218  %
Incorrectly Classified Instances       170               12.782  %
Kappa statistic                          0.8066
Mean absolute error                      0.0639
Root mean squared error                  0.2528
Relative absolute error                 19.1982 %
Root relative squared error             61.9778 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,782    0,084    0,669      0,782    0,721      0,657    0,849     0,562     B
                 0,909    0,036    0,893      0,909    0,901      0,868    0,936     0,834     M
                 0,273    0,031    0,465      0,273    0,344      0,309    0,621     0,193     BM
                 1,000    0,006    0,994      1,000    0,997      0,994    0,997     0,994     R
Weighted Avg.    0,872    0,030    0,863      0,872    0,864      0,840    0,921     0,804

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 186  19  32   1    a = B
  21 299   6   3    b = M
  71  17  33   0    c = BM
   0   0   0 642    d = R
```

A.10 Algoritmo LibSvm Linear e C = 5

123mfcc-libsvm-linear-c5.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 5.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.34 seconds

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

Correctly Classified Instances        1163               87.4436 %
Incorrectly Classified Instances       167               12.5564 %
Kappa statistic                          0.8102
Mean absolute error                      0.0628
Root mean squared error                  0.2506
Relative absolute error                 18.8594 %
Root relative squared error             61.4285 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,782    0,082    0,676      0,782    0,725      0,663    0,850     0,568     B
                 0,909    0,034    0,898      0,909    0,903      0,871    0,937     0,839     M
                 0,298    0,033    0,474      0,298    0,365      0,328    0,632     0,205     BM
                 1,000    0,006    0,994      1,000    0,997      0,994    0,997     0,994     R
Weighted Avg.    0,874    0,029    0,866      0,874    0,868      0,844    0,923     0,807

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 186  17  34   1    a = B
  21 299   6   3    b = M
  68  17  36   0    c = BM
   0   0   0 642    d = R
```

A.11 Algoritmo LibSvm Linear e C = 10

123mfcc-libsvm-linear-c10.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 0 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 10.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
```

```
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.48 seconds

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

Correctly Classified Instances        1171               88.0451 %
Incorrectly Classified Instances       159               11.9549 %
Kappa statistic                          0.8193
Mean absolute error                      0.0598
Root mean squared error                  0.2445
Relative absolute error                 17.956  %
Root relative squared error             59.9391 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,798    0,078    0,691      0,798    0,741      0,682    0,860     0,588     B
                 0,909    0,034    0,898      0,909    0,903      0,871    0,937     0,839     M
                 0,331    0,030    0,526      0,331    0,406      0,373    0,650     0,235     BM
                 1,000    0,006    0,994      1,000    0,997      0,994    0,997     0,994     R
Weighted Avg.    0,880    0,028    0,873      0,880    0,874      0,851    0,926     0,814

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 190  17  30   1    a = B
  21 299   6   3    b = M
  64  17  40   0    c = BM
   0   0   0 642    d = R
```

A.12 Algoritmo LibSvm RBF e C = 0.5 G = 2

123mfcc-libsvm-rbf-c05-g2.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 2.0 -R 0.0 -N 0.5 -M 40.0 -C 0.5 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
```

```
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.4 seconds

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

Correctly Classified Instances        1160               87.218  %
Incorrectly Classified Instances       170               12.782  %
Kappa statistic                          0.8059
Mean absolute error                      0.0639
Root mean squared error                  0.2528
Relative absolute error                 19.1982 %
Root relative squared error             61.9778 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,937    0,131    0,609      0,937    0,738      0,692    0,903     0,582     B
                 0,900    0,025    0,922      0,900    0,911      0,882    0,937     0,854     M
                 0,008    0,002    0,333      0,008    0,016      0,040    0,503     0,093     BM
                 0,997    0,000    1,000      0,997    0,998      0,997    0,998     0,998     R
Weighted Avg.    0,872    0,030    0,850      0,872    0,841      0,827    0,921     0,806

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 223  13   2   0    a = B
  33 296   0   0    b = M
 108  12   1   0    c = BM
   2   0   0 640    d = R
```

A.13 Algoritmo LibSvm RBF e C = 1 G = 1

123mfcc-libsvm-rbf-c1-g1.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 1.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
```

```
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.32 seconds

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

Correctly Classified Instances        1163               87.4436 %
Incorrectly Classified Instances       167               12.5564 %
Kappa statistic                          0.8104
Mean absolute error                      0.0628
Root mean squared error                  0.2506
Relative absolute error                 18.8594 %
Root relative squared error             61.4285 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,807    0,093    0,653      0,807    0,722      0,659    0,857     0,561     B
                 0,909    0,027    0,917      0,909    0,913      0,885    0,941     0,856     M
                 0,248    0,031    0,441      0,248    0,317      0,283    0,608     0,178     BM
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     R
Weighted Avg.    0,874    0,026    0,867      0,874    0,867      0,845    0,924     0,811

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 192  15  31   0    a = B
  23 299   7   0    b = M
  79  12  30   0    c = BM
   0   0   0 642    d = R
```

A.14 Algoritmo LibSvm RBF e C = 1 G = 2

123mfcc-libsvm-rbf-c1-g2.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.LibSVM -S 0 -K 2 -D 3 -G 2.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -model "C:\\Program Files\\Weka-3-8" -seed 1
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
```

```
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

LibSVM wrapper, original code by Yasser EL-Manzalawy = WLSVM

Time taken to build model: 0.34 seconds

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

Correctly Classified Instances        1153               86.6917 %
Incorrectly Classified Instances       177               13.3083 %
Kappa statistic                          0.7989
Mean absolute error                      0.0665
Root mean squared error                  0.258
Relative absolute error                 19.9887 %
Root relative squared error             63.2409 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,798    0,102    0,631      0,798    0,705      0,638    0,848     0,540     B
                 0,909    0,026    0,920      0,909    0,914      0,886    0,941     0,859     M
                 0,182    0,033    0,355      0,182    0,240      0,203    0,574     0,139     BM
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     R
Weighted Avg.    0,867    0,028    0,856      0,867    0,857      0,835    0,920     0,804

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 190  13  35   0    a = B
  25 299   5   0    b = M
  86  13  22   0    c = BM
   0   0   0 642    d = R
```

A.15 Algoritmo MultiLayer Perceptron L = 0.01 M = 0.1

123mfcc-multiperceptron-l001-m01.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.01 -M 0.1 -N 500 -V 0 -S 0 -E 20 -H a
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
```

```
             class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    -0.48337218486126543
    Node 4    -0.768572148710744
    Node 5    -1.6871995535489104
    Node 6    0.814312212419895
    Node 7    -0.10244447588730288
    Node 8    -1.327292224604074
    Node 9    -1.3802939802068097
    Node 10    -3.940734729823996
    Node 11    4.0305256045859466
    Node 12    -0.9362292522514863
    Node 13    -0.4324262591410016
Sigmoid Node 1
    Inputs    Weights
    Threshold    -2.162480660429885
    Node 4    -2.089301326557894
    Node 5    -0.055123397124686745
    Node 6    -0.8647777803050164
    Node 7    -3.197244361432438
    Node 8    1.513191676790789
    Node 9    2.584147414407476
    Node 10    5.420937864888395
    Node 11    -0.6765145297014854
    Node 12    -2.613100330878836
    Node 13    -3.030814556193566
Sigmoid Node 2
    Inputs    Weights
    Threshold    1.310119272700127
    Node 4    -0.9460317382296042
    Node 5    0.9852891124650244
    Node 6    0.0840916009092569
    Node 7    -0.7081227445839523
    Node 8    0.35693714623887457
    Node 9    -1.3316574804647043
    Node 10    -2.5931434589945264
    Node 11    -4.935214721151271
    Node 12    -1.1315387640095622
    Node 13    -0.7972761923786496
Sigmoid Node 3
    Inputs    Weights
    Threshold    -2.4573810820383613
    Node 4    2.3928637009984044
    Node 5    0.4465408344652045
    Node 6    -2.699389217770066
    Node 7    2.5565024609198232
    Node 8    -0.4183310957869252
    Node 9    -0.15933241338850243
    Node 10    0.468498931650024
    Node 11    -1.4629229884026582
    Node 12    3.1587326521086094
    Node 13    2.718238890043373
Sigmoid Node 4
    Inputs    Weights
    Threshold    -0.10290364271352499
    Attrib features0    2.2748312093381844
    Attrib features1    0.1921859733155296
    Attrib features2    -0.12345115159452298
    Attrib features3    1.3163169260938286
    Attrib features4    -1.1013201950932632
    Attrib features5    -1.1452338278064915
    Attrib features6    -0.5830415136426318
    Attrib features7    0.713270830556371
    Attrib features8    -0.1461363172082988
    Attrib features9    -0.026723513745234732
    Attrib features10    -0.3335657016397348
    Attrib features11    -0.9391519901599188
    Attrib features12    0.8673157841672288
    Attrib features13    -0.8481417255291318
    Attrib features14    -0.44939919045166327
    Attrib features15    0.45953911322572366
Sigmoid Node 5
    Inputs    Weights
```

```
    Threshold    0.14007873042173036
    Attrib features0    0.5507010631367487
    Attrib features1    -1.2893930103017845
    Attrib features2    -0.19213810614588428
    Attrib features3    0.7118468081152339
    Attrib features4    0.19301028043787952
    Attrib features5    0.1945347950922069
    Attrib features6    0.572666702007456
    Attrib features7    0.42438833067059056
    Attrib features8    -0.0579085323381227
    Attrib features9    0.4420969515341885
    Attrib features10    0.07866711778338802
    Attrib features11    -0.18228808799468774
    Attrib features12    0.1705786072718274
    Attrib features13    0.26840301957541074
    Attrib features14    -0.06016702729474558
    Attrib features15    -0.05278764041780751 5
Sigmoid Node 6
    Inputs    Weights
    Threshold    -0.15107654158100128
    Attrib features0    -1.0458449015634472
    Attrib features1    1.0257472922119188
    Attrib features2    0.6626246985365254
    Attrib features3    -0.38164391368211165
    Attrib features4    0.27429367807657834
    Attrib features5    0.2064556077162608
    Attrib features6    0.02606871282278718
    Attrib features7    -0.7135775045092565
    Attrib features8    0.0986868520783995
    Attrib features9    -0.12146645858963175
    Attrib features10    -0.06510859999637458
    Attrib features11    0.5249315454000955
    Attrib features12    -0.5020172797751314
    Attrib features13    0.2136681733357947
    Attrib features14    0.19746879410367557
    Attrib features15    -0.28838297194022977
Sigmoid Node 7
    Inputs    Weights
    Threshold    0.29362464324846865
    Attrib features0    2.465450569002424
    Attrib features1    0.042734835685208125
    Attrib features2    -0.4233134907668604
    Attrib features3    1.387853811792485
    Attrib features4    -0.9744634481574864
    Attrib features5    -1.0637027400522148
    Attrib features6    -0.6593500064373508
    Attrib features7    0.6565202116529063
    Attrib features8    0.24383740972823129
    Attrib features9    -0.4559021439578811
    Attrib features10    -0.9202281401530477
    Attrib features11    -0.5976069214202557
    Attrib features12    0.5889790664517642
    Attrib features13    -1.2127485333533317
    Attrib features14    -0.7595967430938332
    Attrib features15    0.7609771496946628
Sigmoid Node 8
    Inputs    Weights
    Threshold    -0.27234233515606554
    Attrib features0    0.0687291681110421
    Attrib features1    -0.7669445267146668
    Attrib features2    0.522529440755485
    Attrib features3    0.17411137431935575
    Attrib features4    -0.4860988620097895 5
    Attrib features5    0.45807062415243766
    Attrib features6    0.5603187928144912
    Attrib features7    0.7650698341832111
    Attrib features8    -0.3106978220568397
    Attrib features9    0.987963221882273
    Attrib features10    0.19966846781453362
    Attrib features11    -0.6822281263699207
    Attrib features12    0.5806409853164177
    Attrib features13    0.36124873659362644
    Attrib features14    0.4245101550769262
    Attrib features15    -0.33573923310759596
Sigmoid Node 9
    Inputs    Weights
    Threshold    -0.6498128200310435
    Attrib features0    0.04926016382792394
```

```
    Attrib features1    0.8337820887676474
    Attrib features2    1.5259094611998494
    Attrib features3    -0.31322175464452073
    Attrib features4    -1.2579062114461972
    Attrib features5    0.11336428242935337
    Attrib features6    -0.11066364983234851
    Attrib features7    0.7817992831313683
    Attrib features8    -0.6563323945278714
    Attrib features9    1.0441127624383784
    Attrib features10    0.9283597092048186
    Attrib features11    -1.3301895610655798
    Attrib features12    1.3385117673708067
    Attrib features13    0.2398965826835464
    Attrib features14    1.0634557452763778
    Attrib features15    -0.7195300907970964
Sigmoid Node 10
    Inputs    Weights
    Threshold    -0.7775864261495763
    Attrib features0    0.48270043951735847
    Attrib features1    1.3345351216061152
    Attrib features2    3.7315229526038363
    Attrib features3    -0.27888711286183276
    Attrib features4    -2.177844737917124
    Attrib features5    -0.1646629054900622
    Attrib features6    -0.7044947429569777
    Attrib features7    1.7463202319120634
    Attrib features8    -0.8748268280366707
    Attrib features9    1.4178948866356094
    Attrib features10    2.37880836980165
    Attrib features11    -1.8417498603294813
    Attrib features12    2.6442446934330612
    Attrib features13    -0.009493207688559747
    Attrib features14    1.7375066895505096
    Attrib features15    -0.8055902433592358
Sigmoid Node 11
    Inputs    Weights
    Threshold    1.3108430629204118
    Attrib features0    -1.8165805282310012
    Attrib features1    4.37303211766966
    Attrib features2    1.0943130709067563
    Attrib features3    -2.5655714610506903
    Attrib features4    -0.7852254678632212
    Attrib features5    -0.09121782247831489
    Attrib features6    -1.2030514811214057
    Attrib features7    -0.6230723709659719
    Attrib features8    -0.3750676584492823
    Attrib features9    -0.6060220051478485
    Attrib features10    -0.6587424698566056
    Attrib features11    -0.18086660000932125
    Attrib features12    0.23026082759345162
    Attrib features13    -0.731307831614465
    Attrib features14    0.17628804939291934
    Attrib features15    0.012863344922902697
Sigmoid Node 12
    Inputs    Weights
    Threshold    -0.045430440716434886
    Attrib features0    2.655325803616282
    Attrib features1    0.18868956033424722
    Attrib features2    -0.01071662431542678
    Attrib features3    1.5388208192578499
    Attrib features4    -1.351146016813293
    Attrib features5    -1.4162541812841818
    Attrib features6    -0.7964107452899329
    Attrib features7    0.8711149896486188
    Attrib features8    -0.20981615679276724
    Attrib features9    -0.08644643649749766
    Attrib features10    -0.3671848862917787
    Attrib features11    -1.0924500694646417
    Attrib features12    1.111590214459304
    Attrib features13    -1.0934875080141557
    Attrib features14    -0.5339452065591425
    Attrib features15    0.5891581730066165
Sigmoid Node 13
    Inputs    Weights
    Threshold    0.09750728720796735
    Attrib features0    2.545351235534443
    Attrib features1    -4.4028052684846967E-4
    Attrib features2    -0.31726259703814863
```

```
    Attrib features3    1.4955162805938653
    Attrib features4    -1.1170569609512182
    Attrib features5    -1.2796999431978096
    Attrib features6    -0.7491023096313476
    Attrib features7    0.8021901073375222
    Attrib features8    0.07782565964281368
    Attrib features9    -0.2776562934707276
    Attrib features10    -0.6745321800665504
    Attrib features11    -0.8105177606668912
    Attrib features12    0.8404684063747302
    Attrib features13    -1.1572026573841583
    Attrib features14    -0.6327477257643291
    Attrib features15    0.6543215424159197
Class B
    Input
    Node 0
Class M
    Input
    Node 1
Class BM
    Input
    Node 2
Class R
    Input
    Node 3


Time taken to build model: 3.84 seconds

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

Correctly Classified Instances        1188               89.3233 %
Incorrectly Classified Instances       142               10.6767 %
Kappa statistic                          0.8391
Mean absolute error                      0.0844
Root mean squared error                  0.1897
Relative absolute error                 25.3511 %
Root relative squared error             46.4961 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,794    0,060    0,744      0,794    0,768      0,716    0,969     0,875     B
                 0,912    0,037    0,890      0,912    0,901      0,868    0,984     0,964     M
                 0,479    0,032    0,598      0,479    0,532      0,494    0,945     0,579     BM
                 0,998    0,001    0,998      0,998    0,998      0,997    1,000     1,000     R
Weighted Avg.    0,893    0,023    0,890      0,893    0,891      0,869    0,985     0,930

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 189  20  29   0    a = B
  18 300  10   1    b = M
  47  16  58   0    c = BM
   0   1   0 641    d = R
```

A.16 Algoritmo MultiLayer Perceptron L = 0.05 M = 0.1

123mfcc-multiperceptron-l005-m01.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.05 -M 0.1 -N 500 -V 0 -S 0 -E 20 -H a
Relation:     123mfccWithoutANDWindow
```

```
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    -1.402922139229259
    Node 4    -0.7293418198431422
    Node 5    -4.667999504149982
    Node 6    3.4845817830336014
    Node 7    1.1527411621251857
    Node 8    -1.765497394148079
    Node 9    -2.4402634501707197
    Node 10    -6.770887423723096
    Node 11    6.154539881612841
    Node 12    -1.0062093332509074
    Node 13    -0.10325353659410928
Sigmoid Node 1
    Inputs    Weights
    Threshold    -4.0242013330931306
    Node 4    -2.900833592787341
    Node 5    1.1226457675888777
    Node 6    -2.6474914688229583
    Node 7    -4.388683507020738
    Node 8    3.604002118617197
    Node 9    5.687832963901874
    Node 10    7.034622474912413
    Node 11    1.6561329359834005
    Node 12    -3.5209043542783536
    Node 13    -4.085347496211682
Sigmoid Node 2
    Inputs    Weights
    Threshold    2.259762029818612
    Node 4    -1.5918629881464172
    Node 5    4.155838029844004
    Node 6    -2.231752779896461
    Node 7    -1.8186786191712152
    Node 8    -0.30330877238916565
    Node 9    -3.175277090265012
    Node 10    -0.18799407435318247
    Node 11    -7.440635802395568
    Node 12    -1.9001450637158972
    Node 13    -1.5179314308275167
Sigmoid Node 3
    Inputs    Weights
    Threshold    -3.3917704639005937
    Node 4    3.1987803458342743
    Node 5    0.21374390879211397
    Node 6    -2.5372883839770695
    Node 7    3.1002392692316345
    Node 8    -0.6896687240523736
    Node 9    0.09501095844056104
    Node 10    0.8686993064505629
    Node 11    -1.3168459397180559
    Node 12    4.104335551782031
    Node 13    3.5246087272086792
Sigmoid Node 4
    Inputs    Weights
    Threshold    0.1234483409963166
```

```
    Attrib features0    3.354993495311507
    Attrib features1    0.09563814948834716
    Attrib features2    -0.32608852174510644
    Attrib features3    1.636284035689394
    Attrib features4    -1.432268598204411
    Attrib features5    -1.3117509198821455
    Attrib features6    -0.6703494491389836
    Attrib features7    0.9132828891178331
    Attrib features8    -0.21337961367450559
    Attrib features9    0.1834889823525815
    Attrib features10    -0.3637513944483682
    Attrib features11    -1.5624740905830818
    Attrib features12    1.133961279505552
    Attrib features13    -1.2290474656817902
    Attrib features14    -0.44053592959208693
    Attrib features15    0.4011170006384916
Sigmoid Node 5
    Inputs    Weights
    Threshold    0.715125687342598
    Attrib features0    -0.3767770841626467
    Attrib features1    -3.9762708658457577
    Attrib features2    1.1390901280508472
    Attrib features3    2.6766825754787793
    Attrib features4    1.500217633546732
    Attrib features5    -0.9585752030106897
    Attrib features6    -0.06499879711476944
    Attrib features7    1.0531383069061182
    Attrib features8    -1.5459032366912788
    Attrib features9    0.04091956634447769
    Attrib features10    1.9282963022845607
    Attrib features11    -1.4263287313853314
    Attrib features12    -1.574050335328585
    Attrib features13    2.2576986270168935
    Attrib features14    0.4490797600208386
    Attrib features15    -3.0044583765757413
Sigmoid Node 6
    Inputs    Weights
    Threshold    1.6548514807696608
    Attrib features0    -2.542481063685866
    Attrib features1    3.4770992197312327
    Attrib features2    2.628175489453044
    Attrib features3    1.4976591852804497
    Attrib features4    0.46788319085718577
    Attrib features5    -0.2757224162022475
    Attrib features6    -0.2908626437387872
    Attrib features7    -0.5876594148117196
    Attrib features8    0.4580202068694466
    Attrib features9    -0.0019110856919282665
    Attrib features10    1.1161267567461421
    Attrib features11    -0.4101032305353316
    Attrib features12    -0.5220213635769096
    Attrib features13    -0.05970124312755391
    Attrib features14    0.9334343746826604
    Attrib features15    -2.7234742218715726
Sigmoid Node 7
    Inputs    Weights
    Threshold    1.157642467184399
    Attrib features0    3.526850303401837
    Attrib features1    -0.042474469155031405
    Attrib features2    -0.6613470762581924
    Attrib features3    2.056717358946679
    Attrib features4    -1.4916740747272443
    Attrib features5    -0.4944791879776878
    Attrib features6    -0.37622612343895345
    Attrib features7    0.2035781547799243
    Attrib features8    0.41977699254899026
    Attrib features9    -0.8366296774250301
    Attrib features10    -1.8029413457218368
    Attrib features11    -0.7656186495457941
    Attrib features12    0.7404644882216798
    Attrib features13    -1.9617041355378428
    Attrib features14    -0.2832620736750658
    Attrib features15    0.9827523210836616
Sigmoid Node 8
    Inputs    Weights
    Threshold    -1.199557742446311
    Attrib features0    0.19610883109449673
    Attrib features1    -0.8850511158813683
```

```
    Attrib features2    0.01516666060408407
    Attrib features3    -1.4006233076594312
    Attrib features4    -2.3281684091396384
    Attrib features5    0.074762592359003
    Attrib features6    0.740732779927147
    Attrib features7    1.3684463164075182
    Attrib features8    0.5108289656968857
    Attrib features9    2.3453392590874604
    Attrib features10    -0.7424841042204332
    Attrib features11    -1.5454566778692567
    Attrib features12    0.9870266446021726
    Attrib features13    0.6725074497900883
    Attrib features14    -0.2648457136111078
    Attrib features15    -1.5529534647626477
Sigmoid Node 9
    Inputs    Weights
    Threshold    -0.30614567169710516
    Attrib features0    -0.22326447439573133
    Attrib features1    -0.02536570852589102
    Attrib features2    1.0326682940048086
    Attrib features3    -3.1616009352781584
    Attrib features4    -4.982379242102734
    Attrib features5    -0.21045939880433345
    Attrib features6    -0.5527309310222213
    Attrib features7    1.6084134825009286
    Attrib features8    -0.655066239073241
    Attrib features9    1.5900900595641727
    Attrib features10    1.0944844566553342
    Attrib features11    -4.1468187912419925
    Attrib features12    2.429981189795824
    Attrib features13    1.8906149426220047
    Attrib features14    2.629328676135554
    Attrib features15    -1.9435438577922137
Sigmoid Node 10
    Inputs    Weights
    Threshold    -0.8829747561840455
    Attrib features0    0.17661810233706593
    Attrib features1    2.475018978554519
    Attrib features2    7.6335240826770825
    Attrib features3    -1.2265383756896608
    Attrib features4    -4.677104709482017
    Attrib features5    -0.1303575202682495
    Attrib features6    -2.016028038912988
    Attrib features7    2.0717174345721268
    Attrib features8    0.2874647443903272
    Attrib features9    0.9959924315702141
    Attrib features10    3.8388167101571096
    Attrib features11    -1.254026780509157
    Attrib features12    3.999146728274595
    Attrib features13    -1.5601002817448413
    Attrib features14    1.3670459467309772
    Attrib features15    0.33803883390297984
Sigmoid Node 11
    Inputs    Weights
    Threshold    5.204523194750406
    Attrib features0    -3.7452951400197567
    Attrib features1    8.065038482673902
    Attrib features2    0.4089384341327486
    Attrib features3    -5.626354249140449
    Attrib features4    -5.0284003708093055
    Attrib features5    -1.5907094791394285
    Attrib features6    -2.058198007309166
    Attrib features7    -0.6905633694961493
    Attrib features8    -1.4507126445408478
    Attrib features9    -0.8118336017755718
    Attrib features10    -0.3641557697767015
    Attrib features11    -0.15558011050914944
    Attrib features12    -0.04634147916639914
    Attrib features13    -0.3551467465868736
    Attrib features14    -0.20585034583071415
    Attrib features15    0.09476163748953652
Sigmoid Node 12
    Inputs    Weights
    Threshold    0.23467533475606456
    Attrib features0    3.8194579425843442
    Attrib features1    0.024418148458461506
    Attrib features2    -0.2939039072013402
    Attrib features3    1.8339907422319912
```

```
    Attrib features4    -1.6963541387256398
    Attrib features5    -1.551017416279434
    Attrib features6    -0.925001219284402
    Attrib features7    1.1295902459195084
    Attrib features8    -0.3121965788778997
    Attrib features9    0.2542796374273601 5
    Attrib features10    -0.34633339007576686
    Attrib features11    -1.7710319504818284
    Attrib features12    1.3848995830166693
    Attrib features13    -1.4977019325000622
    Attrib features14    -0.5373295896402481
    Attrib features15    0.5124586455811916
Sigmoid Node 13
    Inputs    Weights
    Threshold    0.48505579066808086
    Attrib features0    3.690479224255624
    Attrib features1    -0.17117036766653349
    Attrib features2    -0.4825653927739136
    Attrib features3    2.108064264040083
    Attrib features4    -1.5934425498072122
    Attrib features5    -1.110711199973343
    Attrib features6    -0.7595933259321725
    Attrib features7    0.8329298321308544
    Attrib features8    -0.03597111108956971
    Attrib features9    -0.10165560896978998
    Attrib features10    -0.7229147420371389
    Attrib features11    -1.4367465692482582
    Attrib features12    1.016935512931428
    Attrib features13    -1.7337301505049534
    Attrib features14    -0.4300491652274855
    Attrib features15    0.38375037920950084
Class B
    Input
    Node 0
Class M
    Input
    Node 1
Class BM
    Input
    Node 2
Class R
    Input
    Node 3


Time taken to build model: 3.85 seconds

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

Correctly Classified Instances        1217               91.5038 %
Incorrectly Classified Instances       113                8.4962 %
Kappa statistic                          0.8723
Mean absolute error                      0.0616
Root mean squared error                  0.1761
Relative absolute error                 18.5064 %
Root relative squared error             43.1769 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,819    0,046    0,796      0,819    0,807      0,765    0,972     0,899     B
                 0,924    0,023    0,930      0,924    0,927      0,903    0,987     0,966     M
                 0,628    0,033    0,655      0,628    0,641      0,606    0,962     0,717     BM
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     R
Weighted Avg.    0,915    0,017    0,915      0,915    0,915      0,898    0,988     0,948

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 195  12  31   0    a = B
  16 304   9   0    b = M
  34  11  76   0    c = BM
   0   0   0 642    d = R
```

A.17 Algoritmo MultiLayer Perceptron L = 0.08 M = 0.1

123mfcc-multiperceptron-l008-m01.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.08 -M 0.1 -N 500 -V 0 -S 0 -E 20 -H a
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    -2.7750662306942413
    Node 4    -1.1017504157460554
    Node 5    -5.214336006726594
    Node 6    4.92928078312533
    Node 7    3.450168516262616
    Node 8    -3.9060465155780246
    Node 9    -1.6370116196652473
    Node 10    -6.863257730604541
    Node 11    5.606978340326569
    Node 12    -1.4566303982045046
    Node 13    -0.31290588596519375
Sigmoid Node 1
    Inputs    Weights
    Threshold    -3.6090223567589153
    Node 4    -3.3658837377098223
    Node 5    0.9015906913402448
    Node 6    -2.6434400456422225
    Node 7    -5.02597417031671
    Node 8    4.651701014665398
    Node 9    6.995192359464152
    Node 10    7.091239623314837
    Node 11    1.3004996707954504
    Node 12    -4.011606803313098
    Node 13    -4.725669279119384
Sigmoid Node 2
    Inputs    Weights
    Threshold    3.5815965707395065
    Node 4    -2.05341056176634
    Node 5    5.81537305206439
    Node 6    -4.500302360025596
    Node 7    -3.004441327812628
    Node 8    1.0586948750419751
    Node 9    -5.53257129663404
    Node 10    -1.1884949110866831
    Node 11    -8.09256057578556
```

```
    Node 12    -2.420446527018864
    Node 13    -1.8095124157226155
Sigmoid Node 3
    Inputs    Weights
    Threshold    -4.106798053546199
    Node 4    3.497230558572497
    Node 5    0.20585560859676505
    Node 6    -2.315479821156115
    Node 7    2.964030900992128
    Node 8    -0.6307718773910628
    Node 9    0.03848466708750948
    Node 10    0.8231527201238003
    Node 11    -1.8639011572540853
    Node 12    4.4508616270354855
    Node 13    3.8128229391505193
Sigmoid Node 4
    Inputs    Weights
    Threshold    0.2855440434113337
    Attrib features0    3.6954445073960174
    Attrib features1    -0.0330801413810259
    Attrib features2    -0.37670439442942794
    Attrib features3    1.8154344640049394
    Attrib features4    -1.6742111682336445
    Attrib features5    -1.6124703131700715
    Attrib features6    -0.8607300231520632
    Attrib features7    1.0018267129879426
    Attrib features8    -0.35511278362326143
    Attrib features9    0.06373925810418278
    Attrib features10    -0.1881418541694407
    Attrib features11    -1.895047827324048
    Attrib features12    1.2770990347440152
    Attrib features13    -1.3282033634430495
    Attrib features14    -0.36735964134325105
    Attrib features15    0.39246776269290234
Sigmoid Node 5
    Inputs    Weights
    Threshold    -2.589461751161195
    Attrib features0    1.227195652096895
    Attrib features1    -1.051746176471479
    Attrib features2    2.5213933180991654
    Attrib features3    6.906459419829342
    Attrib features4    1.5170988899917184
    Attrib features5    -2.4313514159533343
    Attrib features6    3.639284611585844
    Attrib features7    -0.5505892102635052
    Attrib features8    -0.250719349537011
    Attrib features9    -0.3504641399136408
    Attrib features10    -0.8082458842683093
    Attrib features11    1.949523499955754
    Attrib features12    0.18682896921604375
    Attrib features13    1.292831132710956
    Attrib features14    -4.060224631221437
    Attrib features15    2.6052570100232795
Sigmoid Node 6
    Inputs    Weights
    Threshold    1.9837963673716676
    Attrib features0    -4.179621456479103
    Attrib features1    6.119992539298718
    Attrib features2    4.1539102700698525
    Attrib features3    0.5854157971836238
    Attrib features4    -0.06523102880696299
    Attrib features5    -1.7121197518104974
    Attrib features6    -0.883522876494671
    Attrib features7    -0.9893561933010688
    Attrib features8    -0.038030756539444044
    Attrib features9    -1.2040101238306917
    Attrib features10    0.4988902208817452
    Attrib features11    -1.5640484055603667
    Attrib features12    -1.3892802798776196
    Attrib features13    0.30768018359803706
    Attrib features14    1.2085464303737719
    Attrib features15    -2.7631460314975613
Sigmoid Node 7
    Inputs    Weights
    Threshold    1.4730414009203616
    Attrib features0    4.160089443301884
    Attrib features1    0.9424340427920924
    Attrib features2    -0.7975401629287017
```

```
    Attrib features3    1.3440838307001166
    Attrib features4    -1.944331196543108
    Attrib features5    -0.014113240327091046
    Attrib features6    0.033110844161852455
    Attrib features7    -0.5090687967056631
    Attrib features8    1.7612696509975843
    Attrib features9    -1.380504859487568
    Attrib features10    -3.253825387331728
    Attrib features11    -0.0522628411704349
    Attrib features12    0.7229090946564944
    Attrib features13    -3.1117866997257697
    Attrib features14    0.35720710775440484
    Attrib features15    2.7158670644729996
Sigmoid Node 8
    Inputs    Weights
    Threshold    -1.1230665359794976
    Attrib features0    -0.3576168343614527
    Attrib features1    -1.0508124596191828
    Attrib features2    -0.44392493253887594
    Attrib features3    -2.214657981055932
    Attrib features4    -2.575394043744259
    Attrib features5    0.05064358844973717
    Attrib features6    0.5733560957438643
    Attrib features7    2.796072423923676
    Attrib features8    0.9227443446940818
    Attrib features9    3.1804021053876257
    Attrib features10    -0.5508994878824711
    Attrib features11    -2.6734338994605915
    Attrib features12    0.07103644826305822
    Attrib features13    0.3681140596700052
    Attrib features14    1.0925603734531282
    Attrib features15    -3.370321928264068
Sigmoid Node 9
    Inputs    Weights
    Threshold    -0.38385910994595185
    Attrib features0    -0.7152080317524998
    Attrib features1    -0.011069648924675012
    Attrib features2    1.7977938444349777
    Attrib features3    -4.631845318751307
    Attrib features4    -7.4461577288392125
    Attrib features5    -0.7849457982228828
    Attrib features6    -1.7678947657493513
    Attrib features7    1.2721279699233325
    Attrib features8    -1.423440260482945
    Attrib features9    0.02700478761728683
    Attrib features10    1.7398610835080803
    Attrib features11    -4.56530493519337
    Attrib features12    2.33566593877268
    Attrib features13    2.8595992648886
    Attrib features14    3.0424119024234093
    Attrib features15    -2.2926220279661305
Sigmoid Node 10
    Inputs    Weights
    Threshold    -1.574679438762797
    Attrib features0    -0.46494349067105123
    Attrib features1    4.171896119201032
    Attrib features2    9.176972680005552
    Attrib features3    -2.579326030714805
    Attrib features4    -5.187497230877175
    Attrib features5    0.9707617530654093
    Attrib features6    -1.9511607382976437
    Attrib features7    1.7244331614490522
    Attrib features8    -0.40228983526951734
    Attrib features9    2.021988929043075
    Attrib features10    4.8577652181715445
    Attrib features11    -1.259746762712765
    Attrib features12    4.97193755855283
    Attrib features13    -1.877224252216168
    Attrib features14    1.0671774616112115
    Attrib features15    1.2998169553848016
Sigmoid Node 11
    Inputs    Weights
    Threshold    5.724529532468564
    Attrib features0    -4.111257573118133
    Attrib features1    11.217427544080378
    Attrib features2    -0.17828607562364102
    Attrib features3    -5.117159831102055
    Attrib features4    -5.613979976977862
```

```
    Attrib features5    -2.03819362291579
    Attrib features6    -1.3320462453341078
    Attrib features7    -0.7939359726898885
    Attrib features8    -1.4133483777678573
    Attrib features9    0.2586644615931105
    Attrib features10    -1.0164487888449643
    Attrib features11    1.858812180114834
    Attrib features12    0.7950988141979813
    Attrib features13    -0.6046782014255679
    Attrib features14    -2.6700915503209255
    Attrib features15    1.5275906067225076
Sigmoid Node 12
    Inputs    Weights
    Threshold    0.42632803599084534
    Attrib features0    4.145164556034941
    Attrib features1    -0.088422744306164
    Attrib features2    -0.3533394587147479
    Attrib features3    2.018466335716629
    Attrib features4    -1.9227228921159376
    Attrib features5    -1.8253471788779518
    Attrib features6    -1.1164927992097637
    Attrib features7    1.2077571018564937
    Attrib features8    -0.46214380524089227
    Attrib features9    0.1837294123282359
    Attrib features10    -0.16362619253111962
    Attrib features11    -2.097870044328639
    Attrib features12    1.5313364683474697
    Attrib features13    -1.59004628282174
    Attrib features14    -0.4571969779502779
    Attrib features15    0.5211293753237016
Sigmoid Node 13
    Inputs    Weights
    Threshold    0.7312999657517659
    Attrib features0    3.9612862746118673
    Attrib features1    -0.4000758770263451
    Attrib features2    -0.6046409377525939
    Attrib features3    2.39269919514414
    Attrib features4    -2.0797926250163146
    Attrib features5    -1.320309192346471
    Attrib features6    -0.8685276565626711
    Attrib features7    0.730240586012444
    Attrib features8    -0.1642477502884186
    Attrib features9    -0.2982576016917533
    Attrib features10    -0.5010562726266187
    Attrib features11    -1.763784720220773 7
    Attrib features12    1.04502999655202
    Attrib features13    -1.9035216259854952
    Attrib features14    -0.24658826002162018
    Attrib features15    0.19494676907083883
Class B
    Input
    Node 0
Class M
    Input
    Node 1
Class BM
    Input
    Node 2
Class R
    Input
    Node 3


Time taken to build model: 3.85 seconds

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

Correctly Classified Instances        1213               91.203  %
Incorrectly Classified Instances       117                8.797  %
Kappa statistic                          0.8677
Mean absolute error                      0.0592
Root mean squared error                  0.1797
Relative absolute error                 17.7698 %
Root relative squared error             44.0457 %
Total Number of Instances             1330

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

```
                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,811    0,051    0,775      0,811    0,793      0,746    0,970     0,889     B
                 0,918    0,024    0,926      0,918    0,922      0,897    0,987     0,961     M
                 0,628    0,031    0,673      0,628    0,650      0,616    0,960     0,698     BM
                 1,000    0,000    1,000      1,000    1,000      1,000    1,000     1,000     R
Weighted Avg.    0,912    0,018    0,912      0,912    0,912      0,894    0,988     0,943

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 193  14  31   0    a = B
  21 302   6   0    b = M
  35  10  76   0    c = BM
   0   0   0 642    d = R
```

A.18 Algoritmo MultiLayer Perceptron L = 0.1 M = 0.1

123mfcc-multiperceptron-l01-m01.dat

```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.1 -M 0.1 -N 500 -V 0 -S 0 -E 20 -H a
Relation:     123mfccWithoutANDWindow
Instances:    1330
Attributes:   17
              features0
              features1
              features2
              features3
              features4
              features5
              features6
              features7
              features8
              features9
              features10
              features11
              features12
              features13
              features14
              features15
              class
Test mode:    10-fold cross-validation

=== Classifier model full training set ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    -2.8282766300967994
    Node 4    -1.2345566374774455
    Node 5    -5.537323177743142
    Node 6    5.4115223195304445
    Node 7    3.951012392053634
    Node 8    -5.090175977887902
    Node 9    -2.0636522622717437
    Node 10    -7.484478762685109
    Node 11    5.817255662534512
    Node 12    -1.603026280658293
    Node 13    -0.4621243578847075
Sigmoid Node 1
    Inputs    Weights
    Threshold    -3.492417182775155
    Node 4    -3.847244438554354
    Node 5    0.4001634178495558
    Node 6    -2.5245086336545213
```

```
    Node 7    -5.552190444493765
    Node 8    4.626035035284231
    Node 9    7.713111238491402
    Node 10    8.071647028217498
    Node 11    1.0174705212692094
    Node 12    -4.495538589296348
    Node 13    -5.201296355275939
Sigmoid Node 2
    Inputs    Weights
    Threshold    3.5486240353125424
    Node 4    -2.4933702761695256
    Node 5    6.431555752467756
    Node 6    -5.050717668345339
    Node 7    -3.398940610946952
    Node 8    1.909675860240762
    Node 9    -6.483740450751399
    Node 10    -1.7529666069823124
    Node 11    -8.42134916294148
    Node 12    -2.9077861200611728
    Node 13    -2.2241781323456387
Sigmoid Node 3
    Inputs    Weights
    Threshold    -4.3758582788670575
    Node 4    3.609442980100335
    Node 5    0.41230295875095835
    Node 6    -2.295051092417571
    Node 7    2.9974019252418564
    Node 8    -0.49004339620048987
    Node 9    0.021856774480000944
    Node 10    0.7665732757399545
    Node 11    -2.0226518302614647
    Node 12    4.56670891713552
    Node 13    3.953041520219152
Sigmoid Node 4
    Inputs    Weights
    Threshold    0.35121993473929447
    Attrib features0    3.8730920595374605
    Attrib features1    -0.05596384687333792
    Attrib features2    -0.43171628322835204
    Attrib features3    1.8679912350354924
    Attrib features4    -1.7824695473986774
    Attrib features5    -1.6061618349572468
    Attrib features6    -0.9467741736178185
    Attrib features7    1.035467197467855
    Attrib features8    -0.5453467423342845
    Attrib features9    0.076589072169943
    Attrib features10    -0.16709508754457006
    Attrib features11    -2.0888669376627487
    Attrib features12    1.4520219339202238
    Attrib features13    -1.3884190333746351
    Attrib features14    -0.4715836521305382
    Attrib features15    0.42073955337430385
Sigmoid Node 5
    Inputs    Weights
    Threshold    -3.284463735706202
    Attrib features0    1.7438886978189185
    Attrib features1    -1.4502377163951135
    Attrib features2    3.0327890329857237
    Attrib features3    7.880738798322279
    Attrib features4    1.4563415029900824
    Attrib features5    -2.943415788876658
    Attrib features6    4.452505474371843
    Attrib features7    -0.8977370438717827
    Attrib features8    -0.5884481920952057
    Attrib features9    -0.20949569839907453
    Attrib features10    -0.40979002863772435
    Attrib features11    2.993119039955488
    Attrib features12    -0.44729525937695735
    Attrib features13    1.7405690159127738
    Attrib features14    -4.82399319121585
    Attrib features15    3.4875318633608305
Sigmoid Node 6
    Inputs    Weights
    Threshold    1.8553652547175394
    Attrib features0    -4.975857740600458
    Attrib features1    7.408731083862627
    Attrib features2    3.6774411352155503
    Attrib features3    -1.1320787652727178
```

```
    Attrib features4    -1.4182222640827216
    Attrib features5    -1.574751771315486
    Attrib features6    0.17272478720696818
    Attrib features7    -1.2225717408065309
    Attrib features8    -1.4271362836389605
    Attrib features9    -1.1725703514304313
    Attrib features10    0.24187357098059759
    Attrib features11    -1.4967270613317867
    Attrib features12    -2.0607306766549445
    Attrib features13    0.9854646736138746
    Attrib features14    0.6693854288200785
    Attrib features15    -3.1267563338175965
Sigmoid Node 7
    Inputs    Weights
    Threshold    1.7994875593599493
    Attrib features0    4.443489558771987
    Attrib features1    1.336723844852226
    Attrib features2    -0.7351282229805102
    Attrib features3    0.875662013673606
    Attrib features4    -2.277390804608974
    Attrib features5    0.4859306869185073
    Attrib features6    -0.2624002459444291
    Attrib features7    -0.7414216194669073
    Attrib features8    2.002461069913554
    Attrib features9    -1.1737454932409102
    Attrib features10    -3.4077740864615094
    Attrib features11    0.10918874404302435
    Attrib features12    0.5961632674010932
    Attrib features13    -3.500791854345229
    Attrib features14    0.9435372845787363
    Attrib features15    3.0327248045161372
Sigmoid Node 8
    Inputs    Weights
    Threshold    -1.1560773272137819
    Attrib features0    -0.6492195486420748
    Attrib features1    -0.7722823726223087
    Attrib features2    -0.42514813357423714
    Attrib features3    -2.498889711116467
    Attrib features4    -2.506972576876189
    Attrib features5    -0.033672895745621705
    Attrib features6    0.9882905234588155
    Attrib features7    3.565998054479026
    Attrib features8    0.9206540509482638
    Attrib features9    3.8603471389629007
    Attrib features10    -0.8405694092636992
    Attrib features11    -3.109378264736933
    Attrib features12    -0.008338447731633872
    Attrib features13    0.13516039886731904
    Attrib features14    1.5853138797649093
    Attrib features15    -4.082748225387493
Sigmoid Node 9
    Inputs    Weights
    Threshold    -0.5770718120915339
    Attrib features0    -1.1986548033530025
    Attrib features1    -0.12033470852892722
    Attrib features2    2.159038991809338
    Attrib features3    -5.67833816846797
    Attrib features4    -9.608891797225747
    Attrib features5    -1.012053869750641
    Attrib features6    -1.9151611006490463
    Attrib features7    0.8835948250791665
    Attrib features8    -1.494756077812804
    Attrib features9    -0.514939129771939
    Attrib features10    1.6336498321102106
    Attrib features11    -4.509617126693086
    Attrib features12    2.501188769711495
    Attrib features13    3.122814795640237
    Attrib features14    3.4907199730148792
    Attrib features15    -2.6022954990002156
Sigmoid Node 10
    Inputs    Weights
    Threshold    -1.9410464651502963
    Attrib features0    -0.7253740180498522
    Attrib features1    6.079539269534003
    Attrib features2    9.994150004643629
    Attrib features3    -3.2138039573341954
    Attrib features4    -5.285710207888153
    Attrib features5    1.5437986014284468
```

```
    Attrib features6    -1.5301488685738687
    Attrib features7    1.7739967758618662
    Attrib features8    -1.7446238159211223
    Attrib features9    3.1135310211806595
    Attrib features10    5.231080907326004
    Attrib features11    -1.0733452875325504
    Attrib features12    4.956476681929756
    Attrib features13    -1.4713933558762171
    Attrib features14    0.2979236939299724
    Attrib features15    2.3320947780684973
Sigmoid Node 11
    Inputs    Weights
    Threshold    5.763896217175352
    Attrib features0    -4.793873015423691
    Attrib features1    12.11630907458258
    Attrib features2    -0.1401158508608808
    Attrib features3    -5.245883447461915
    Attrib features4    -5.5243079907432415
    Attrib features5    -2.4400912440462954
    Attrib features6    -1.6875978138525056
    Attrib features7    -0.5362610933262213
    Attrib features8    -1.1749616017871884
    Attrib features9    0.5804331805545292
    Attrib features10    -0.8023513103254067
    Attrib features11    2.4617196906126533
    Attrib features12    0.9619474387463028
    Attrib features13    -0.8217105345836044
    Attrib features14    -3.1166008669175067
    Attrib features15    2.0374989274717903
Sigmoid Node 12
    Inputs    Weights
    Threshold    0.4680706048289333
    Attrib features0    4.311196603396243
    Attrib features1    -0.07088559615079842
    Attrib features2    -0.4090267791774398
    Attrib features3    2.087997473039944
    Attrib features4    -2.0378935412365515
    Attrib features5    -1.7886946529058025
    Attrib features6    -1.2127687830722893
    Attrib features7    1.2583206676840615
    Attrib features8    -0.6066887440350914
    Attrib features9    0.1972674491089951
    Attrib features10    -0.13848107804235732
    Attrib features11    -2.2264130345818245
    Attrib features12    1.684483207038758
    Attrib features13    -1.653983852728403
    Attrib features14    -0.5297887716649073
    Attrib features15    0.5577136538141412
Sigmoid Node 13
    Inputs    Weights
    Threshold    0.7567365061638702
    Attrib features0    4.161011001840818
    Attrib features1    -0.35885162071482846
    Attrib features2    -0.7429860489604626
    Attrib features3    2.3601661960814355
    Attrib features4    -2.206991480220633
    Attrib features5    -1.2759141180941587
    Attrib features6    -1.0109736954123252
    Attrib features7    0.7840529924391794
    Attrib features8    -0.38123573321508236
    Attrib features9    -0.2112327732508408
    Attrib features10    -0.5013749922127898
    Attrib features11    -1.9968869467774446
    Attrib features12    1.2277550921760267
    Attrib features13    -1.9421152622441584
    Attrib features14    -0.35638570857234403
    Attrib features15    0.29083534984209314
Class B
    Input
    Node 0
Class M
    Input
    Node 1
Class BM
    Input
    Node 2
Class R
    Input
```

```
    Node 3


Time taken to build model: 4.15 seconds

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

Correctly Classified Instances        1214               91.2782 %
Incorrectly Classified Instances       116                8.7218 %
Kappa statistic                          0.8688
Mean absolute error                      0.0587
Root mean squared error                  0.1829
Relative absolute error                 17.6259 %
Root relative squared error             44.8455 %
Total Number of Instances             1330

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,824    0,052    0,775      0,824    0,798      0,753    0,967     0,882     B
                 0,915    0,023    0,929      0,915    0,922      0,897    0,986     0,954     M
                 0,620    0,029    0,682      0,620    0,649      0,617    0,955     0,681     BM
                 1,000    0,001    0,998      1,000    0,999      0,998    1,000     1,000     R
Weighted Avg.    0,913    0,018    0,912      0,913    0,912      0,895    0,987     0,938

=== Confusion Matrix ===

   a   b   c   d   <-- classified as
 196  13  29   0    a = B
  21 301   6   1    b = M
  36  10  75   0    c = BM
   0   0   0 642    d = R
```