



**LUIZ CARLOS BRANDÃO JUNIOR**

**DETERMINAÇÃO DE UM MODELO NÃO INTRUSIVO DE  
QUALIDADE DE VOZ FUNDAMENTADO NA ANÁLISE DO  
SINAL NO DOMÍNIO DO TEMPO USANDO APRENDIZADO  
DE MÁQUINA**

**LAVRAS – MG**

**2019**

**LUIZ CARLOS BRANDÃO JUNIOR**

**DETERMINAÇÃO DE UM MODELO NÃO INTRUSIVO DE QUALIDADE DE VOZ  
FUNDAMENTADO NA ANÁLISE DO SINAL NO DOMÍNIO DO TEMPO USANDO  
APRENDIZADO DE MÁQUINA**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação para a obtenção do título de Mestre.

Prof. DSc. Demóstenes Zegarra Rodríguez

Orientador

Prof. DSc. Gabriel Fernando Pivaro Leite da Conceição

Coorientador

**LAVRAS – MG**

**2019**

**Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca  
Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).**

Junior, Luiz Carlos Brandão.

Determinação de um modelo não intrusivo de qualidade de voz fundamentado na análise do sinal no domínio do tempo usando aprendizado de máquina / Luiz Carlos Brandão Junior. - 2019.  
115 p. : il.

Orientador(a): Demóstenes Zegarra Rodríguez.

Coorientador(a): Gabriel Fernando Pivaro Leite da Conceição.  
Dissertação (mestrado acadêmico) - Universidade Federal de Lavras, 2019.  
Bibliografia.

1. Métricas de Qualidade de Voz. 2. Redes Sem Fio. 3. Aprendizado de Máquina. I. Rodríguez, Demóstenes Zegarra. II. Pivaro Leite da Conceição, Gabriel Fernando. III. Título.

**LUIZ CARLOS BRANDÃO JUNIOR**

**DETERMINAÇÃO DE UM MODELO NÃO INTRUSIVO DE QUALIDADE DE VOZ  
FUNDAMENTADO NA ANÁLISE DO SINAL NO DOMÍNIO DO TEMPO USANDO  
APRENDIZADO DE MÁQUINA**

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Engenharia de Sistemas e Automação para a obtenção do título de Mestre.

APROVADA em 14 de Junho de 2019.

Prof. DSc. Demóstenes Zegarra Rodríguez DCC/UFLA  
Prof. DSc. Renata Lopes Rosa DCC/UFLA  
Prof. DSc. Dante Coaquira Begazo PSI-EPUSP

Prof. DSc. Demóstenes Zegarra Rodríguez  
Orientador

Prof. DSc. Gabriel Fernando Pivaro Leite da Conceição  
Co-Orientador

**LAVRAS – MG  
2019**

*Dedico o presente trabalho a minha mãe, Aláide, que sempre foi e continua sendo o meu maior triunfo e apoio, seja nos momentos de angústia ou não e que fez de tudo para que a faculdade se tornasse um sonho possível.*

## AGRADECIMENTOS

Agradeço primeiramente a Deus por cada dia me mostrar o quanto somos realmente pequenos diante da grandiosidade de suas ações, não há vitória sem crescimento espiritual. Sou imensamente grato a minha mãe Alaíde Natividade de Souza, por sempre me apoiar e incentivar nesta luta, me ajudando em todos os momentos da minha vida, é nesse Forte que, todas às vezes, busco me apoiar. À Pró-Reitoria de Assuntos Estudantis e Comunitários (PRAEC) pela concessão da bolsa de estudos por um ano. Agradeço ao meu orientador Demóstenes Zegarra Rodríguez e coorientador Gabriel Fernando Pivaro Leite da Conceição pela dedicação, paciência e ensinamentos. Aos meus amigos e colegas de profissão que nunca saíram do meu lado e que estiveram presentes nesta caminhada, me encorajando e entendendo. Sou imensamente grato à Universidade Federal de Lavras (UFLA), em especial ao Programa de Pós-Graduação em Engenharia de Sistemas e Automação (PPGESISA), e ao Departamento de Engenharia (DEG) pela oportunidade de crescimento acadêmico. Aos demais professores pelo tempo dedicado ao ensino, ser professor não é tarefa nada fácil. E é lógico que, eu não poderia deixar de agradecer a você, caro(a) leitor(a), vindo(a) de um futuro que eu desconheça qual seja, se chegou aqui motivado(a) pela primeira história que escrevi, provavelmente ela, em seu tempo, já deva ter virado um livro. Fico muito feliz de tê-lo(a) por aqui também, verá que alguns momentos ou números contidos nesse primeiro volume das crônicas foram retirados daqui, dessa pesquisa, como por exemplo, o número de MOS ecoando em um dos capítulos ou a parte que os personagens têm alguns *Déjà vu*. A você e a todos os demais, que mesmo de maneira singela contribuíram de algum modo para o meu crescimento espiritual, sou imensamente grato. Muito obrigado mesmo, de coração!  
:)

*Agir, eis a inteligência verdadeira. Serei o que quiser. Mas tenho que querer o que for. O êxito está em ter êxito, e não em ter condições de êxito. Condições de palácio tem qualquer terra larga, mas onde estará o palácio se não o fizerem ali?*

*(Fernando Pessoa)*

## RESUMO

A Voz sobre o Protocolo de *Internet* (VoIP) é um dos serviços de comunicação que surgiu no início da década de 1990. Nos últimos anos, a capacidade das redes IP foi incrementada, e a tecnologia passou a ganhar mais espaço realizando-se investimentos em qualidade de serviço. Nesse trabalho, é proposta uma solução para estimar a qualidade de um sinal de voz utilizando a informação do sinal no domínio do tempo e com o suporte de algoritmos de aprendizado de máquina. A metodologia foi dividida em três etapas, sendo que na primeira, foram aplicadas degradações em ambientes que simulavam redes wireless fazendo alterações em dois parâmetros que foram, a relação sinal-ruído (SNR) e o tipo de esquema de modulação. Nos testes foram utilizados seis sinais sonoros originais distintos. Para realizar estas degradações, algoritmos implementados em MATLAB foram utilizados para simular o efeito do *fading* em ambientes *wireless*. Na segunda etapa plotou-se gráficos dos sinais sonoros (de áudio) degradados, no domínio do tempo que foram salvos, usou-se 272 imagens para fazer o treinamento em 12 algoritmos distintos de aprendizado de máquina implementados na ferramenta Weka. Na última etapa, os algoritmos treinados foram colocados em um *software* feito em Java denominado de PredictorFX a fim de prever o valor do MOS mediante uma imagem de áudio no domínio do tempo. Os resultados se mostraram satisfatórios, os melhores Algoritmos Treinados de Regressão (ATR), foram RandomTree, RandomForest e IBk com seus coeficientes de correlação variando de 0,9886 até 0,9989 na fase de validação para os dados que resultou o MOS, denominado de algoritmo treinado de regressão (ATR1). Em relação aos ATR2, que contém as informações extraídas das imagens, os melhores algoritmos foram RandomTree, RandomForest, M5P e MLP, com coeficiente de correlação variando entre 0,8638 até 0,9896, na fase de validação. E por fim, para os Algoritmos de Treinamento para a Classificação (ATC) denominado de ATC1 os melhores algoritmos foram OneR, J48, MLP e RandomForest com 58,82% a 96,32% das instâncias corretamente classificadas. Tais resultados demonstram que é possível realizar testes de qualidade de voz de maneira não intrusiva usando modelos fundamentados na recomendação ITU-T P.862.

**Palavras-chave:** Métricas de qualidade, Voz sobre IP (VoIP), Qualidade de Voz, Degradação, Desvanecimento, Wireless, Recomendação ITU-T P.862, Weka, Aprendizado de Máquina.

## ABSTRACT

Voice over Internet Protocol (VoIP) is one of the communication services that emerged in the early 1990s. In recent years, the capacity of IP networks has increased, and technology has gained more space by performing investment in quality of service. In this work, a solution is proposed to estimate the quality of a voice signal using signal information in the time domain and with the support of machine learning algorithms. The methodology was divided in three stages. In the first one, degradations were applied in environments that simulated wireless networks, making changes in two parameters that were, the signal-to-noise ratio (SNR) and the type of modulation scheme. In the tests, six different original sound signals were used. To perform these degradations, algorithms implemented in MATLAB were used to simulate the effect of fading in wireless environments. In the second step, graphs of the degraded audio signals were written, in the time domain that were saved, 272 images were used to train in 12 different machine learning algorithms implemented in the Weka tool. In the last step, the trained algorithms were placed in a Java-based software called PredictorFX in order to predict the value of MOS using an audio image in the time domain. The results were satisfactory, the best Regression Algorithms (ATR) were RandomTree, RandomForest and IBk with their correlation coefficients varying from 0.9886 to 0.9989 in the validation phase for the data that resulted in the MOS, called trained regression algorithm (ATR1). In relation to ATR2, which contains the information extracted from the images, the best algorithms were RandomTree, RandomForest, M5P and MLP, with correlation coefficient varying from 0.8638 to 0.9896, in the validation phase. Finally, for the Classification Training Algorithms (ATC) called ATC1, the best algorithms were OneR, J48, MLP and RandomForest with 58.82 % to 96.32 % of the correctly sorted instances. These results demonstrate that it is possible to conduct non-intrusive voice quality tests using models based on the ITU-T Recommendation P.862.

**Keywords:** Quality Metrics, Voice over IP (VoIP), Speech Quality, Impairment, Fading, Wireless, ITU-T Recommendation P.862, Weka, Machine Learning.

## LISTA DE FIGURAS

Figura 2.1 – Representação da modulação QPSK . . . . .	26
Figura 2.2 – Representações das modulações 16-QAM e 64-QAM . . . . .	26
Figura 2.3 – Visão geral da filosofia básica usada no PESQ . . . . .	28
Figura 2.4 – Representação da recomendação ITU-T P.862 . . . . .	29
Figura 2.5 – Representação do K-FOLD com K=10 . . . . .	37
Figura 2.6 – Representação de 10 Validações . . . . .	38
Figura 3.1 – Esquema adotado . . . . .	42
Figura 3.2 – Cenário Geral que simula uma comunicação semelhante a um ambiente real	43
Figura 3.3 – Esquema detalhado . . . . .	44
Figura 3.4 – Imagens dos áudios originais or105, or109, or114, or129, or134 e or137 . .	45
Figura 3.5 – Imagens do áudio desvanecido or114 . . . . .	46
Figura 3.6 – Escala de conversão do MOS . . . . .	48
Figura 3.7 – Classificação dos pixels das imagens que representam o sinal de áudio . . .	50
Figura 3.8 – Processo de escaneamento de imagens . . . . .	51
Figura 3.9 – Processo de escaneamento e composição do Vetor <sub>(Campo)</sub> . . . . .	52
Figura 3.10 – Gráfico contendo 50 pontos da função diferença . . . . .	53
Figura 3.11 – 272 Funções Discretas . . . . .	55
Figura 3.12 – Configuração r1, r2 e c1 . . . . .	56
Figura 3.13 – Esquema usado nos algoritmos de aprendizado de regressão e classificação	57
Figura 4.1 – Esquema da formulação dos algoritmos treinados, ATR1 (Regressão 1), ATR2 (Regressão 2) e ATC1 (Classificação 1) . . . . .	58
Figura 4.2 – Teste de hipótese para ATR1 . . . . .	59
Figura 4.3 – Comportamento dos CC para ATR1 . . . . .	63
Figura 4.4 – Comportamento dos EAM para ATR1 . . . . .	64
Figura 4.5 – Comportamento dos EQM para ATR1 . . . . .	64
Figura 4.6 – <i>Software</i> PredictorFX para ATR1 . . . . .	65
Figura 4.7 – Teste de Friedman e Nemenyi ATR2 . . . . .	67
Figura 4.8 – Predição da imagem or105 com SNR de 28dB e modulação 64-QAM para ATR2 . . . . .	68
Figura 4.9 – Cálculo da secante no triângulo retângulo . . . . .	69
Figura 4.10 – Predição da imagem or109 com SNR de 33dB e modulação QPSK para ATR2	71

Figura 4.11 – Predição da imagem or114 com SNR de 32dB e modulação 2-QAM para ATR2 . . . . .	72
Figura 4.12 – Predição da imagem or129 com SNR de 21dB e modulação 16-QAM para ATR2 . . . . .	74
Figura 4.13 – Predição da imagem or134 com SNR de 23dB e modulação 32-QAM para ATR2 . . . . .	75
Figura 4.14 – Predição da imagem or137 com SNR de 30dB e modulação 256-QAM para ATR2 . . . . .	77
Figura 4.15 – Comportamento dos CC para ATR2 . . . . .	78
Figura 4.16 – Comportamento dos EAM para ATR2 . . . . .	79
Figura 4.17 – Comportamento dos EQM para ATR2 . . . . .	80
Figura 4.18 – Teste de Friedman e Nemenyi para ATC1 . . . . .	81
Figura 4.19 – Predição da imagem or105 com SNR de 32dB e modulação QPSK para ATC1	84
Figura 4.20 – Predição da imagem or109 com SNR de 16dB e modulação 4-QAM para ATC1 . . . . .	87
Figura 4.21 – Predição da imagem or114 com SNR de 23dB e modulação 32-QAM para ATC1 . . . . .	90
Figura 4.22 – Predição da imagem or129 com SNR de 30dB e modulação 64-QAM para ATC1 . . . . .	93
Figura 4.23 – Predição da imagem or134 com SNR de 31dB e modulação 128-QAM para ATC1 . . . . .	96
Figura 4.24 – Predição da imagem or137 com SNR de 29dB e modulação 256-QAM para ATC1 . . . . .	99
Figura 4.25 – Comportamento das ICC para ATC1 . . . . .	100
Figura 4.26 – Comportamento das EKP para ATC1 . . . . .	101
Figura 4.27 – Comportamento dos EAM para ATC1 . . . . .	102
Figura 4.28 – Comportamento dos EQM para ATC1 . . . . .	102
Figura 4.29 – Modelos KazaR, RazaR, KazaM e MazaK . . . . .	103
Figura 1 – Funções Discretas(FD) para o MOS Péssimo . . . . .	112
Figura 2 – Funções Discretas(FD) para o MOS Ruim . . . . .	113
Figura 3 – Funções Discretas(FD) para o MOS Razoável . . . . .	114
Figura 4 – Funções Discretas(FD) para o MOS Bom . . . . .	115

## LISTA DE TABELAS

Tabela 3.1 – Informações dos áudios . . . . .	44
Tabela 3.2 – Valores do MOS para os áudios desvanecidos do arquivo original or105 . . .	47
Tabela 3.3 – Intervalos do $MOS_{Numérico}$ e $MOS_{Literal}$ . . . . .	48
Tabela 3.4 – Quantidade de imagens por $MOS_{Literal}$ . . . . .	49
Tabela 4.1 – Médias dos coeficientes de correlação para ATR1. . . . .	59
Tabela 4.2 – <i>Ranking</i> para ATR1. . . . .	59
Tabela 4.3 – Sumário dos Algoritmos Treinados para or105_r1. . . . .	60
Tabela 4.4 – Validação do modelo para os arquivos degradados or109_r1 . . . . .	61
Tabela 4.5 – Validação do modelo para os arquivos degradados or114_r1 . . . . .	62
Tabela 4.6 – Validação do modelo para os arquivos degradados or129_r1 . . . . .	62
Tabela 4.7 – Validação do modelo para os arquivos degradados or134_r1 . . . . .	62
Tabela 4.8 – Validação do modelo para os arquivos degradados or137_r1 . . . . .	63
Tabela 4.9 – Médias dos coeficientes de correlação para ATR2. . . . .	66
Tabela 4.10 – Ranqueamento para ATR2. . . . .	66
Tabela 4.11 – Sumário dos Algoritmos Treinados para or105_r2. . . . .	67
Tabela 4.12 – Validação do modelo para os arquivos degradados or109_r2 . . . . .	70
Tabela 4.13 – Validação do modelo para os arquivos degradados or114_r2 . . . . .	72
Tabela 4.14 – Validação do modelo para os arquivos degradados or129_r2 . . . . .	73
Tabela 4.15 – Validação do modelo para os arquivos degradados or134_r2 . . . . .	75
Tabela 4.16 – Validação do modelo para os arquivos degradados or137_r2 . . . . .	76
Tabela 4.17 – Médias dos coeficientes de correlação para ATC1. . . . .	80
Tabela 4.18 – Ranqueamento para ATC1. . . . .	81
Tabela 4.19 – Sumário dos Algoritmos Treinados para or105_c1. . . . .	82
Tabela 4.20 – Matrizes de Confusão dos algoritmos treinados para ATC1. . . . .	82
Tabela 4.21 – Validação do modelo para os arquivos degradados or109_c1 . . . . .	85
Tabela 4.22 – Matrizes de Confusão da validação da base de dados or109. . . . .	85
Tabela 4.23 – Validação do modelo para os arquivos degradados or114_c1 . . . . .	88
Tabela 4.24 – Matrizes de Confusão da validação da base de dados or114. . . . .	88
Tabela 4.25 – Validação do modelo para os arquivos degradados or129_c1 . . . . .	91
Tabela 4.26 – Matrizes de Confusão da validação da base de dados or129. . . . .	91
Tabela 4.27 – Validação do modelo para os arquivos degradados or134_c1 . . . . .	94

Tabela 4.28 – Matrizes de Confusão da validação da base de dados or134. . . . .	94
Tabela 4.29 – Validação do modelo para os arquivos degradados or137_c1 . . . . .	97
Tabela 4.30 – Matrizes de Confusão da validação da base de dados or137. . . . .	97
Tabela 4.31 – Quantidade de imagens por intervalo. . . . .	100
Tabela 6.1 – Comparação das informações dos algoritmos para or105_r1. . . . .	106

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
<b>1.1</b>	<b>Objetivo principal</b>	<b>18</b>
<b>1.1.1</b>	<b>Objetivos específicos</b>	<b>18</b>
<b>1.1.2</b>	<b>Organização do Trabalho</b>	<b>19</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>20</b>
<b>2.1</b>	<b>Sistemas de comunicação</b>	<b>20</b>
<b>2.1.1</b>	<b>Redes sem fio</b>	<b>20</b>
<b>2.1.2</b>	<b>Diferenças entre redes sem fio e de telefonia fixa</b>	<b>21</b>
<b>2.2</b>	<b>Radiofrequência (RF)</b>	<b>22</b>
<b>2.2.1</b>	<b>Canais no espectro de radiofrequência</b>	<b>22</b>
<b>2.3</b>	<b>Fenômenos que acontecem em um canal RF</b>	<b>22</b>
<b>2.3.1</b>	<b>Atenuação</b>	<b>22</b>
<b>2.3.2</b>	<b>Ganho</b>	<b>22</b>
<b>2.3.3</b>	<b>Desvanecimento</b>	<b>23</b>
<b>2.3.4</b>	<b>Modelos de Desvanecimento (<i>Fading</i>)</b>	<b>23</b>
<b>2.4</b>	<b>Parâmetros de avaliação de desempenho</b>	<b>24</b>
<b>2.5</b>	<b>Esquema de modulação</b>	<b>25</b>
<b>2.5.1</b>	<b>Modulação QPSK</b>	<b>25</b>
<b>2.5.2</b>	<b>Modulação QAM</b>	<b>26</b>
<b>2.6</b>	<b>Qualidade de Serviço e Qualidade de Experiência</b>	<b>27</b>
<b>2.6.1</b>	<b>Qualidade de Serviço (<i>QoS</i>)</b>	<b>27</b>
<b>2.6.2</b>	<b>Qualidade de Experiência (<i>QoE</i>)</b>	<b>27</b>
<b>2.7</b>	<b>Método de avaliação de qualidade de voz</b>	<b>27</b>
<b>2.7.1</b>	<b>Recomendação ITU-T P.862</b>	<b>28</b>
<b>2.8</b>	<b>Teste de Friedman-Nemenyi</b>	<b>29</b>
<b>2.9</b>	<b><i>Waikato Environment for Knowledge Analysis (Weka)</i></b>	<b>30</b>
<b>2.9.1</b>	<b>Dataset</b>	<b>30</b>
<b>2.9.2</b>	<b>Formato ARFF</b>	<b>31</b>
<b>2.9.3</b>	<b>Classifier</b>	<b>31</b>
<b>2.10</b>	<b>Validação Cruzada (Cross-Validation)</b>	<b>35</b>
<b>2.11</b>	<b>Medidas de desempenho</b>	<b>38</b>

<b>3</b>	<b>METODOLOGIA</b>	<b>42</b>
<b>3.1</b>	<b>Cenário geral</b>	<b>43</b>
<b>3.2</b>	<b>Mineração de Dados</b>	<b>46</b>
<b>3.3</b>	<b>Extração de Características de uma Imagem</b>	<b>49</b>
<b>3.3.1</b>	<b>Escaneando Imagens</b>	<b>51</b>
<b>3.3.2</b>	<b>Execução dos algoritmos de treinamento do Weka</b>	<b>55</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>58</b>
<b>4.1</b>	<b>Resultados da Regressão</b>	<b>58</b>
<b>4.1.1</b>	<b>Resultado da Regressão para ATR1</b>	<b>58</b>
<b>4.1.2</b>	<b>Resultado da Regressão para ATR2</b>	<b>66</b>
<b>4.2</b>	<b>Resultado da Classificação para ATC1</b>	<b>80</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>104</b>
<b>6</b>	<b>TRABALHOS FUTUROS</b>	<b>106</b>
	<b>REFERÊNCIAS</b>	<b>108</b>
	<b>APENDICE A – Funções Discretas para o MOS Péssimo</b>	<b>112</b>
	<b>APENDICE B – Funções Discretas para o MOS Ruim</b>	<b>113</b>
	<b>APENDICE C – Funções Discretas para o MOS Razoável</b>	<b>114</b>
	<b>APENDICE D – Funções Discretas para o MOS Bom</b>	<b>115</b>

## 1 INTRODUÇÃO

Mobilidade e acesso às redes sem fio são conceitos chaves para o mercado de telecomunicações, por oferecer um maior conforto aos seus usuários e por sua praticidade tornam-se tendências irreversíveis. Uma dessas praticidades é a utilização de serviços de comunicações de voz cujo sistema não se trata de uma tecnologia recente. Porém, até pouco tempo atrás, esse serviço era executado por redes tradicionais de telefonia, a qual não permite a mobilidade. A fim de reduzir custos e dar mobilidade aos seus usuários, o serviço de transmissão de voz sobre IP (VoIP) torna-se uma solução viável, embora se tenha uma quantidade maior de variáveis que degradam o sinal.

Segundo (CISCO, 2019) o tráfego global de dados móveis cresceu 71% em 2017, atingindo 11,5 *exabytes*<sup>1</sup> por mês no final de 2017, acima de 6,7 *exabytes* por mês no final de 2016. Além disso, o tráfego desses dados cresceu 17 vezes nos últimos 5 anos com 650 milhões de dispositivos móveis e conexões que foram adicionados em 2017, sendo que os *smartphones* representaram a maior parte desse crescimento, seguidos pelos módulos M2M<sup>2</sup>. Os dispositivos e conexões móveis globais em 2017 cresceram para 8,6 bilhões, ante 7,9 bilhões em 2016 (CISCO, 2019).

A telefonia tradicional ou fixa para funcionar demanda de uma infraestrutura de rede que envolve equipamentos de comutação por circuitos. Embora se tenha uma boa qualidade das ligações o serviço cobrado por essa qualidade é relativamente alto e há uma complexidade da estrutura para que esse sistema possa operar. A Anatel monitora a qualidade do Serviço Telefônico Fixo Comutado (STFC), a telefonia fixa, por meio de 16 indicadores operacionais definidos no Regulamento de Gestão da Qualidade (RGQ-STFC), aprovado pela Resolução nº 605/2012. Considerando os indicadores válidos de todas as prestadoras monitoradas durante o ano de 2017, o percentual de cumprimento de metas do serviço alcançou 78,6%. Esse resultado representa uma melhora ao verificado nos anos de 2013 (74,7%), 2014 (73,2%), 2015 (72,8%) e 2016 (77,5%)(ANATEL, 2018a). O Serviço Telefônico Fixo Comutado (STFC), possui um bom desempenho, se restringindo a um ponto de acesso onde o aparelho ou terminal ficará imóvel, deste modo, não permitindo a mobilidade de seus usuários. Tal serviço permite também a transferência de dados via *Asymmetric Digital Subscriber Line* (ADSL), por exemplo. Além disso, faz diferenças entre chamadas locais e de longa distância, com tarifas mais caras para

<sup>1</sup> Um *exabyte* é equivalente a um bilhão de *gigabytes* e mil *petabytes*

<sup>2</sup> M2M, máquina para máquina - do inglês *machine to machine*, é uma nomenclatura bem ampla que pode definir e descrever várias tecnologias de rede que fazem a comunicação entre dispositivos.

discagem internacionais e para celulares. Já na telefonia VoIP, o cenário muda, se o usuário estiver com um telefone celular, permitindo assim a mobilidade. Porém, se a pessoa estiver falando de um telefone IP, ou de um discador/*softphone*, ou Adaptador telefônico Analógico (ATA) com acesso a *Internet*, a transmissão poderá ser feita, parte em rede cabeada e parte em *Wireless*, não havendo diferenças entre chamadas locais, interurbanas ou internacionais.

Nas redes *Wireless* ocorrem diversos fenômenos físicos, como: reflexão de sinais, interferências, instabilidade física da antena, ruído, potência, entre outros. A relação entre o sinal e o ruído é descrita pela nomenclatura SNR(*Signal-to-noise ratio*) se referindo que no sinal transmitido pode ter ruído inserido (ENTELCO, 2015). Em relação ao desvanecimento, as ondas de rádio estão sujeitas a reflexões no solo que provocam alterações na sua amplitude e caminho percorrido ocasionando variações na potência do sinal recebido. Estas variações são chamadas de desvanecimento ou *fading*. Sendo causado também por obstáculos na linha de visada direta, ou por atenuação devido a chuvas.

A escala de qualidade de voz mais usada é o *Mean Opinion Score (MOS)* que é definido pela recomendação ITU-T P.800 (P.800, 1998). Este método se baseia na análise objetiva, onde o índice MOS é dado por algum algoritmo de medição. Este índice varia de 1 (péssimo) a 5 (excelente), sendo que uma pontuação 4 ou maior (HU; LOUIZOU, 2008), indica que a qualidade do sinal de voz está adequada ao serviço de telefonia. Em relação aos métodos subjetivos de um grupo de avaliadores sobre a qualidade de uma conversa, os ouvintes escutam uma amostra de voz, atribuindo uma pontuação em seguida. Sendo baseados em testes de audição conduzidos em um ambiente controlado de laboratório onde voluntários seguem os procedimentos que o supervisor do teste estabelece, ou usando uma ferramenta de *Crowdsourcing*. Por outro lado, os métodos objetivos são baseados em algoritmos que tentam prever a avaliação humana sobre um sinal de voz. Sendo subdivididos em dois tipos: Intrusivos e Não intrusivos. O primeiro método precisa de um sinal de referência para comparar com o sinal no receptor e garantir a qualidade da pontuação. Por esta razão os métodos intrusivos são mais confiáveis e são usados como referência para a avaliação objetiva. Por outro lado, os métodos não intrusivos são os que precisam apenas do sinal no receptor, ou em um dado ponto onde o sinal deve ser avaliado e assim são mais rápidos, o que possibilita seu uso em serviços de tempo real. No método objetivo intrusivo existem diversos algoritmos que avaliam a qualidade do sinal de voz. A recomendação ITU-T P.862 (P.862, 2007), popularmente conhecida como PESQ (*Perceptual Evaluation of Speech Quality*), possui o algoritmo mais empregado para este tipo de avaliação. Em relação ao

método não intrusivo, a recomendação ITU-T P.563 (P.563, 2004) utiliza um algoritmo padrão para avaliar a qualidade da voz que é aplicável para as previsões da qualidade de voz sem um sinal de referência.

A proposta deste trabalho é simular diferentes tipos de degradação em redes *Wireless* variando parâmetros como nível de ruído, SNR (dB) e tipo de modulação. Utilizando métricas de qualidade se determinará o índice MOS dos arquivos resultantes dos testes experimentais. Por ter uma maior acurácia o algoritmo intrusivo descrito na recomendação ITU-T P.862 será utilizado. Deste modo, obtém-se informações de parâmetros simulados com o índice de qualidade obtido para cada cenário simulado. Assim, poderá estabelecer a relação entre os parâmetros de rede e a qualidade, seja por meio de modelos matemáticos ou utilizando algoritmos de redes neurais e técnicas de aprendizado de máquinas (*machine learning*).

É importante destacar que na literatura atual existem poucos trabalhos que avaliam o desempenho de redes sem fio usando o índice de qualidade de voz, pois outros parâmetros como taxa de erros de bits são geralmente utilizados. Nesse sentido, o presente trabalho apresenta uma contribuição, pois o índice MOS é utilizado como parâmetro de desempenho.

## **1.1 Objetivo principal**

O objetivo principal é determinar um modelo não intrusivo de qualidade de voz fundamentado na análise do sinal no domínio do tempo. Para cumprir esse objetivo, diferentes níveis de degradação foram inseridos em um canal de transmissão *wireless* obtendo-se diversos áudios degradados. A determinação do modelo de avaliação de qualidade de voz é realizada usando algoritmos de aprendizado de máquina, os quais utilizam como entrada as características temporais do sinal de voz.

### **1.1.1 Objetivos específicos**

Os objetivos específicos são:

1. Examinar metodologias e propostas de métricas de qualidade de voz;
2. Analisar as degradações ocorridas no cenário simulado, com a ferramenta da recomendação ITU-T P.862, considerando os índices MOS obtidos;
3. Extrair características das imagens com extensão jpg e por meio de algoritmos de regressão e classificação analisar tais imagens dos áudios desvanecidos.

4. Usar as imagens dos áudios desvanecidos para treinar algoritmos do *software* WEKA e construir modelos de modo que possa prever o MOS usando informação da imagem.
5. Desenvolver um *software* em Java que usa os modelos construídos no *Weka* para prever os valores do MOS.

### 1.1.2 Organização do Trabalho

**Capítulo 1:** Demonstra a parte introdutória, sendo apresentado uma visão geral deste trabalho, o contexto em que ele está inserido, destacando o objetivo principal.

**Capítulo 2:** Apresenta uma Revisão Teórica, sendo dividido em sistemas de comunicação, Radiofrequência (RF), fenômenos que acontecem em um canal RF, parâmetros de avaliação de desempenho, modulações, qualidade de serviço e experiência, métodos de avaliação de qualidade de voz, rede neural artificial (RNA) e aprendizado de máquina (machine learning).

**Capítulo 3:** Apresenta a metodologia utilizada, sendo subdividida em duas etapas. Na primeira, chamado de cenário geral, foram aplicadas degradações em ambientes que simulam redes *Wireless*. Na segunda, foram usados os arquivos de imagens dos áudios desvanecidos para fazer modelos utilizando diversos algoritmos do WEKA, por meio de um software desenvolvido em Java que possa carregar outras imagens, realizando a predição do MOS em arquivos não usados para treinamento e teste.

**Capítulo 4:** Demonstra os resultados encontrados e a construção dos modelos para o uso no *software* PredictorFX desenvolvido em Java com a finalidade de prever o valor do MOS.

**Capítulo 5:** Conclusão a respeito de tudo que foi desenvolvido e projetado para a elaboração desse trabalho.

## 2 REFERENCIAL TEÓRICO

Neste capítulo serão abordados sistemas de comunicação, redes sem fio, conceitos de modulação e seus tipos, rádio frequência (RF), canais, propriedades de radiofrequência, redes neurais artificiais e aprendizado de máquina.

### 2.1 Sistemas de comunicação

De acordo com (COMER, 2016) o termo comunicação engloba uma área muito ampla e abarca muitos campos de estudo, é nesse sentido que ela entra na vida dos indivíduos de formas tão diferentes, seja em telefones, rádios, televisores e terminais de computador. A comunicação fornece informação aos navios em alto-mar, de aeronaves a foguetes e satélites no espaço, além de serviço de previsão do tempo. Em (HAYKIN; MOHER, 2011) os autores referem que a comunicação nada mais é do que o processo de transferência de informação gerada em um ponto no tempo e no espaço por uma fonte para um outro ponto de destino, se resumindo a esses itens.

1. **Transferência de informação:** é a transmissão de dados em banda base ou seja, na frequência que originalmente foi gerada.
2. **Meio físico:** onde se transita a informação, seja ele o ar, fios ou qualquer meio que possa transmitir uma informação.
3. **Onda portadora:** é a onda que transporta a informação em uma frequência diferente e maior que a informação base. A portadora pode ser a luz, sinal de micro-ondas, sinais elétricos e etc.
4. **Modulação:** Capta o sinal que se quer transmitir e modifica uma onda portadora de acordo com as variações de estados.
5. **Sinal modulado:** é o sinal resultante da modificação da onda portadora através da variação de estado da onda de banda base (sinal modulante) feita através da modulação, é nele que o receptor irá tirar a informação que deseja.

#### 2.1.1 Redes sem fio

O nome rede é comumente usado para nomear equipamentos interligados que compartilham recursos, informações e serviços. Em contrapartida as redes sem fio são sistemas de

comunicação não cabeadas, ou seja, não necessita de cabos. Dependendo do tipo de cobertura, a rede sem fio pode ser classificada como rede sem fio de área pessoal WPAN (*Wireless Personal Area Network*), rede de área local sem fio WLAN (*Wireless Local Area Network*), rede sem fio de área metropolitana WMAN (*Wireless Metropolitan Area Network*) (RAPPAPORT, 2009).

De acordo com (DIGERATI, 2009) a WPAN é a rede onde os nós ficam muito próximos uns dos outros, cuja finalidade é prover conectividade entre dispositivos pessoais separados por até 10 metros, aproximadamente. As WLAN são redes locais que usam ondas de rádio para transmissão de dados e para conexão à Internet, sem necessidade de usar os tradicionais cabos para conectar dispositivos. A WMAN são as redes sem fios que ocupam áreas mais abrangentes, como exemplo, na ligação de vários edifícios em uma cidade.

A WAN (*Wide Area Network*) é a rede de longa distância sendo normalmente o maior tipo de rede, podendo representar um país inteiro ou até o mundo inteiro. As WANs podem ser construídas para reunir várias MAN. A internet é o melhor exemplo de uma WAN. Escritórios corporativos em diferentes países podem ser conectados para criar uma WAN. As WANs podem usar cabos de fibra ótica ou podem ser sem fio usando tecnologia de micro-ondas ou linhas alugadas (HAYKIN; MOHER, 2011).

### **2.1.2 Diferenças entre redes sem fio e de telefonia fixa**

De acordo com (RAPPAPORT, 2009) a rede pública de telefonia comutada PSTN (*Public Switched Telephone Network*) ocorre por cabos de cobre, fibras ópticas, enlaces de micro-ondas e de satélites. As configurações de rede na PSTN são praticamente estáticas, pois as conexões só podem ser mudadas quando um assinante muda de localidade mediante requerimento e para isso somente com uma reprogramação. Já em relação as redes sem fio, elas são dinâmicas, com a configuração da rede sendo modificada sempre que um assinante muda de região com cobertura diferente da qual estava. A largura de banda do canal disponível para redes fixas pode ser aumentada instalando-se cabos de alta capacidade (cabo de fibra ótica ou coaxial), enquanto as redes sem fio são restritas pela escassa largura de banda de Rádio Frequência (RF) fornecida para cada usuário.

## 2.2 Radiofrequência (RF)

De acordo com (ANATEL, 2018b) a Radiofrequência é a faixa do espectro eletromagnético de 8,3 kHz a 3000 GHz, onde é possível a radiocomunicação. Os sinais alternados de alta frequência que percorrem cabos condutores e chegam até as antenas, onde são convertidas em ondas eletromagnéticas (OEM) e posteriormente irradiadas para o ar, essas ondas recebem o nome de Radiofrequência (RF). No receptor ocorre justamente o contrário, captando as OEM e transformando-as em sinais elétricos.

### 2.2.1 Canais no espectro de radiofrequência

O espectro de radiofrequência é dividido em faixas que são intervalos reservados, que por sua vez, é dividido em frequências menores para permitir a transmissão em paralelo de sinais diferentes em cada uma delas, essas sub frequências recebem o nome de canais (RUFINO, 2005). Como exemplos tem-se os canais de rádio (AM/FM) e canais usados nos sistemas de comunicações moveis.

## 2.3 Fenômenos que acontecem em um canal RF

Dentre os principais fenômenos que acontecem em um canal RF temos a atenuação, ganho, desvanecimentos e *fading*, que são descritos a seguir.

### 2.3.1 Atenuação

A atenuação pode ser definida como a diminuição da intensidade de energia de um sinal ao propagar-se através de um meio. Em ambientes abertos a dispersão de energia pelo meio pode ser vista como uma forma de atenuação, se a potência do sinal que atinge o receptor for menor que a emitida, então houve uma atenuação do meio em que o sinal se propagou. Em ambientes fechados, cabos por exemplo, quanto maior o comprimento do cabo maior será a atenuação até o ponto de o sinal tornar-se fraco a ponto de a mensagem enviada não ser mais entendida pelo destinatário (PINHEIRO, 2017).

### 2.3.2 Ganho

O ganho é o aumento da amplitude de um sinal de radiofrequência (RF) e é dado em referência a uma antena padrão, normalmente, uma antena isotrópica, onde o ganho é expresso em

(dBi). Uma antena isotrópica irradia o sinal igualmente bem em todas as direções, contudo ela é ideal fornecendo apenas padrões teóricos com os quais as antenas reais podem ser comparadas (PINHEIRO, 2017). Além desse ganho há outros tipos de ganho no sinal como é o apresentado por um dispositivo amplificador ou atenuador, que consiste em modificar a amplitude de um sinal aplicado à sua entrada. Quando se trata de sinal sonoro, geralmente expressa-se em decibel (dB).

### 2.3.3 Desvanecimento

Desvanecimento é a variação na amplitude do sinal no receptor devido às características do percurso do sinal e as mudanças nele, fazendo o sinal variar em amplitude, normalmente, tornando-se menor. Ele é causado por quatro fatores, sendo eles: variação na distância entre o transmissor e o receptor, mudanças nas características ambientais do percurso do sinal, presença de múltiplos percursos de sinal e movimento relativo entre o transmissor e o receptor (FRENZEL, 2009).

### 2.3.4 Modelos de Desvanecimento (*Fading*)

Grande escala é um tipo de *fading* que acontece quando ocorre atenuação do sinal emitido ao transpor obstáculos como elevações do terreno, construções, vegetação e montanhas dentre outros, esse tipo de desvanecimento recebe o nome de sombreamento e se encontram no trajeto entre o transmissor e o receptor. Os fenômenos atmosféricos também contribuem para esse desvanecimento como por exemplo, tem-se o caso da chuva, a neve e granizo, seja como for em todos os casos, tem-se uma perda de percurso (*path loss*) (WANG; CHEN; PEDRAM, 2017).

As variações de larga escala se referem às perdas no espaço livre que ocorre quando apenas parte da energia transmitida através de ondas eletromagnéticas é captada pela antena receptora. Pela reflexão quando uma onda de rádio encontra um obstáculo, parte dela ou sua totalidade é refletida, causando uma perda de potência para a onda de rádio. Difração se refere às ondas que refletem para a mesma pessoa (reflexão) ou quando passa, pelo muro (refração), com a intensidade bastante menor. Dispersão ou espelhamento, ocorre pelo fato que, quando uma onda de rádio se colide com uma superfície áspera, a energia refletida é difundida em todas as direções, devido à dispersão. Absorção ocorre quando um sinal atinge um objeto e ele é totalmente absorvido pelo seu material, a onda não atravessa, reflete ou contorna o

material. E por fim as interferências ocorrem quando no ambiente há algum outro sinal inibindo ou atrapalhando a onda chegar ao seu destino.

Por outro lado, as variações de pequena escala são causadas pela multiplicidade de percursos, por reflexão, difração e espalhamento da energia entre o transmissor e o receptor (desvanecimento multipercursos) e à velocidade de terminal móvel (desvio Doppler). Esse desvanecimento representa rápidas flutuações da amplitude do sinal em pequenas distancias de deslocamento, nesse caso é causado pela interferência de várias versões do sinal enviado. Alcançando ao usuário em tempos diferentes, por percorrer trajetos distintos entre a estação e o local de destino (HU; LOUIZOU, 2008).

As causas dos desvanecimentos podem ser das mais diversas, as ambientais por exemplo, fica por conta de obstáculos na linha de visada das antenas, os efeitos físicos no feixe de onda incluem obstrução, multipercursos, divergência, convergência e absorção, esses tipos de desvanecimento são classificados de planos. Já as reflexões em superfícies e as inversões térmicas, efeitos físicos de interferência e degradação, esses tipos de desvanecimentos são chamados de seletivos (ZANATELI, 2007).

Outro tipo de desvanecimento é o causado pelo superdimensionamento, ou seja, altura maior que a necessária das torres que ocasiona a diminuição de recepção. Esse superdimensionamento acarreta desvanecimento por causa do aumento do comprimento de metragem do guia de onda ou cabo coaxial, entre antena e Estação Rádio Base (ERB). Tais antenas estarão mais propensas às interferências e reflexões múltiplas de acordo com (MIYOSHI; SANCHEZ, 2002).

Em relação às cidades, às antenas e aos receptores costumam ficar em lugares mais baixos do que a vizinhança ao entorno, de acordo com (SILVA, 2004) (RODRÍGUEZ et al., 2012) (RODRÍGUEZ; ROSA; BRESSAN, 2014) no que diz respeito ao local no qual o sinal é transmitido, o desvanecimento (*fading*) pode ser de dois tipos, grande e pequena escala.

## **2.4 Parâmetros de avaliação de desempenho**

Os parâmetros de avaliação de desempenho são basicamente o BER e o MOS. A taxa de erro de bit (BER) é um parâmetro usado na avaliação de sistemas que transmitem dados digitais de um local para outro. Sistemas para os quais a taxa de erro de bit é aplicável, incluem *links* de dados de rádio, bem como sistemas de dados de fibra ótica, *Ethernet* ou qualquer sistema que transmita dados através de uma rede, onde ruído, interferência e *jitter* de fase podem

causar degradação da qualidade do sinal (ALI; SCHOLAR, 2018). Tal parâmetro está diretamente relacionado ao ACR (*Attenuation to Crosstalk Ratio*) ou SNR (*Signal-to-noise ratio*) pois quanto menor o nível de ruído menores serão as distorções e conseqüentemente menores serão as proporções de bits com erro (OLIVIERO; WOODWARD, 2014). Em relação ao MOS (*Mean Opinion Score*) é o método mais usado para avaliar o desempenho da qualidade de voz, é definido pela recomendação ITU-T P.800. Este método se baseia na análise subjetiva de se medir o MOS com avaliadores independentes atribuindo uma pontuação de 5 (excelente) a 1 (péssima) à qualidade da fala (P.800, 1998). Os modelos objetivos são baseados em equações matemáticas com o intuito de procurar estimar o MOS o mais próximo possível das avaliações subjetivas.

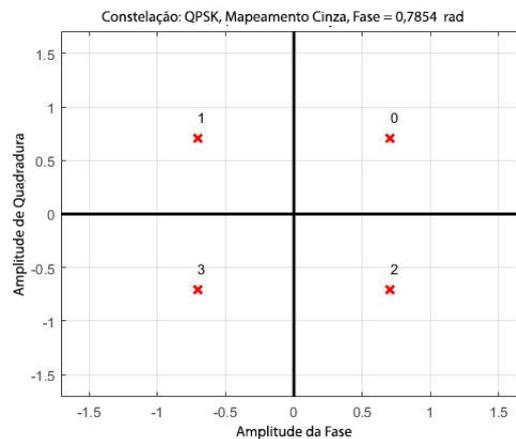
## 2.5 Esquema de modulação

A modulação é o processo através do qual uma ou mais características de uma onda portadora é modificada de acordo com a informação do sinal de mensagem. Diante disso, o sinal da informação (a mensagem) é denominado sinal modulante e a saída do processo de modulação é conhecida como sinal modulado (HAYKIN; MOHER, 2009). A modulação é importante para adequar o sinal ao meio de transmissão. Para isso, ela realiza uma translação do sinal em banda básica para a faixa de frequência da portadora do canal que será usado.

### 2.5.1 Modulação QPSK

Nesta modulação a frequência do sinal é constante, mudando somente o deslocamento da fase do sinal em quadratura, isto veio aperfeiçoar as técnicas anteriores através do aumento da eficiência espectral, utilizando para isso a variação de fase a cada  $90^\circ$  (graus) podendo assim representar dois *bits* (dobro do BPSK) e aumentando a velocidade de transmissão dos dados (duplicando a eficiência de banda) (ÖZCAN; ZIZKA; NAGAMALAI, 2011). A Figura 2.1 ilustra em um diagrama de fase e quadratura (IQ) os 4 possíveis símbolos gerados pela modulação QPSK usando 2 bits/símbolo.

Figura 2.1 – Representação da modulação QPSK.

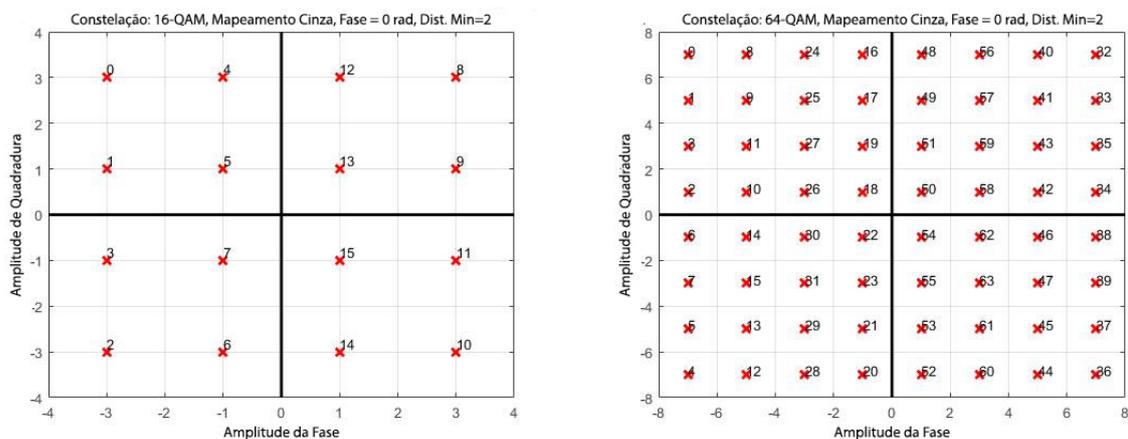


Fonte: do Autor

## 2.5.2 Modulação QAM

A modulação QAM (*Quadrature Amplitude Modulation*) modifica simultaneamente duas características da onda da portadora que são a amplitude e fase, que variam de acordo com a informação a ser transmitida permitindo transmissões de dados com alta taxa de transferência. A modulação 64-QAM alcança uma taxa de transmissão maior do que a 16-QAM, uma vez que, cada símbolo transporta um número maior de bits. Contudo a modulação 16-QAM, cuja distância euclidiana entre os símbolos é maior do que no caso 64-QAM, possibilita uma melhor qualidade de serviço (*QoS*), pois se tem uma maior distância entre os símbolos, dificultando assim erros de interpretação no receptor no momento que detecta um símbolo (HAYKIN, 2008). Ambas podem ser visualizadas na Figura 2.2

Figura 2.2 – Representações das modulações 16-QAM e 64-QAM.



Fonte: do Autor

## **2.6 Qualidade de Serviço e Qualidade de Experiência**

### **2.6.1 Qualidade de Serviço (*QoS*)**

Qualidade de serviço identifica a capacidade da rede em oferecer serviços diferenciados para um tráfego selecionado fluindo através de diversas tecnologias, seu principal objetivo é priorizar o tráfego interativo sensível a retardo, em detrimento ao tráfego referente à transferência de arquivos, que não seja sensível a retardo (ROSS, 2007).

### **2.6.2 Qualidade de Experiência (*QoE*)**

A QoE mede o grau de satisfação do usuário com um determinado fornecedor, seja este de conteúdo, internet, serviço, equipamentos, entre outros. Para o ITU-T (grupo de estudos que define padronizações para o setor de telecomunicações), a métrica QoE é definida como: Grau de satisfação ou aborrecimento do usuário de um aplicativo ou serviço. Esse conceito resulta do cumprimento de suas expectativas referentes à utilidade e/ou satisfação do aplicativo ou serviço em relação à personalidade do usuário e do estado atual (ITU, 2017).

## **2.7 Método de avaliação de qualidade de voz**

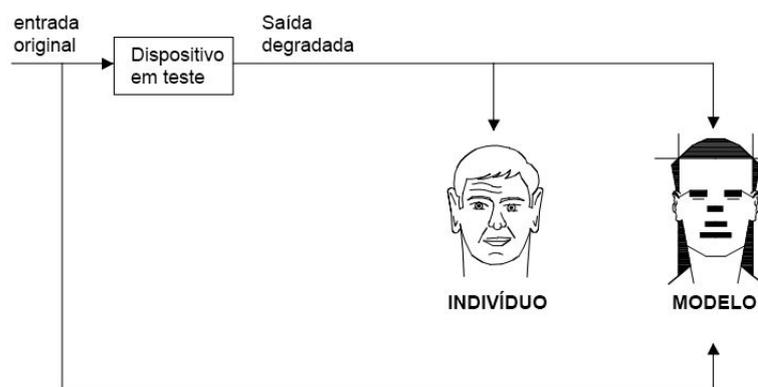
Para saber como o sinal da voz está chegando ao usuário final, ou seja, a sua percepção, é necessário que se tenha métodos que avaliem a qualidade, podendo ser objetivos e subjetivos (SOUSA, 2007). Em relação ao primeiro, ele se fundamenta em modelos matemáticos e podem ser intrusivos que são aqueles métodos que necessitam de uma amostra da fala no ponto de origem, onde aconteceu a comunicação para poder comparar com a amostra do ponto de destino, dando um resultado de avaliação da qualidade. E os não-intrusivos onde não é necessária uma amostra do sinal original da comunicação, a avaliação só é determinada pelo sinal no ponto que é analisado (RODRÍGUEZ, 2009). Já os subjetivos baseiam-se na avaliação de pessoas através da audição (SUN, 2004), em outras palavras, nele existe somente dependência das opiniões de usuários em pontuar a qualidade da voz percebida por esses mesmos usuários (AFFONSO; RODRÍGUEZ, 2016). Somente a métrica ITU-T P.862 (P.862, 2007), foi utilizada para fazer a medição da qualidade neste trabalho.

### 2.7.1 Recomendação ITU-T P.862

Este método objetivo é usado para prever a qualidade subjetiva de codificadores de áudios. É conhecido como PESQ (*Perceptual Evaluation of Speech Quality*), não sendo aplicado só a *codecs* de fala, mas também a medições de extremo a extremo. O PESQ conta com um novo algoritmo para distorções médias ao longo do tempo, incluindo uma série de experimentos que testam especificamente seu desempenho em combinações de fatores como filtragem, atraso variável, distorções de codificação e erros de canal. É recomendado que o PESQ seja utilizado para a avaliação da qualidade da fala de 3.1 kHz para telefonia celular e *codecs* de voz, ambos de banda estreita. Contudo, a emenda 2 da recomendação ITU-T P.862, permite avaliar a qualidade de voz de codificadores de banda larga (P.862, 2007). O algoritmo PESQ mede os efeitos de ruído e a distorção de voz unidirecional sobre a qualidade da voz, não estabelecendo uma avaliação exaustiva da qualidade da transmissão. Esse algoritmo compara o sinal original (entrada) com o sinal degradado (saída) do dispositivo sob teste usando um modelo perceptivo.

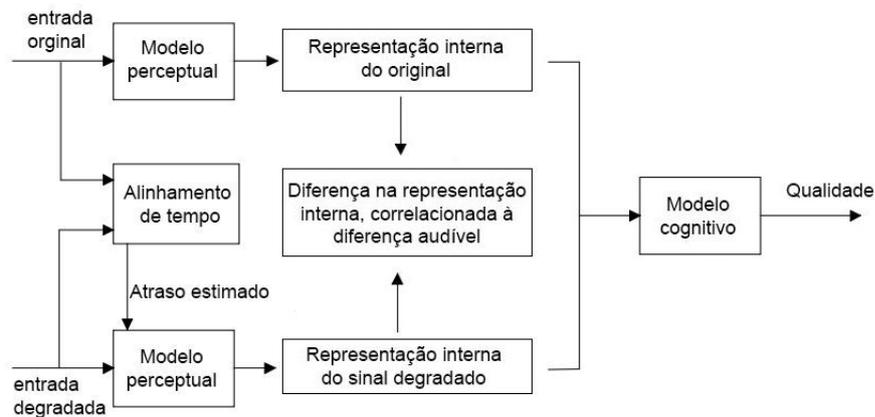
A saída PESQ é uma forma de predição do som ao passar por uma série de parâmetros que degradam o sinal original demonstrando a qualidade que um ouvinte perceberia ao fim do teste subjetivo de audição. Na Figura 2.3 mostra a ideia básica usada no PESQ. O modelo computacional compara a entrada e a saída do dispositivo em teste. Já na Figura 2.4 tanto o sinal original quanto o degradado são comparados por meio do modelo perceptual, baseando-se no conjunto de atrasos encontrados. O processo demonstrado pelas figuras é transformar ambos os sinais em uma forma de representação que seja similar ao sistema auditivo das pessoas, levando em consideração a frequência perceptual e a intensidade.

Figura 2.3 – Visão geral da filosofia básica usada no PESQ.



Fonte: Adaptado de (P.862, 2007)

Figura 2.4 – Representação da recomendação ITU-T P.862.



Fonte: Adaptado de (P.862, 2007)

## 2.8 Teste de Friedman-Nemenyi

O teste de Friedman é um teste estatístico não paramétrico desenvolvido pelo economista americano Milton Friedman. Semelhante ao ANOVA (*Analysis of Variance*) de medidas repetidas paramétricas, é usado para detectar diferenças nos tratamentos em várias tentativas de teste. O procedimento envolve a classificação de cada linha (ou bloco), considerando os valores de classificações por colunas. Aplicável a projetos completos de blocos, é, portanto, um caso especial do teste de Durbin. O teste de Friedman é usado para análise de variância de medidas repetidas bidirecionais por *rankings*. Em seu uso de classificações, é semelhante à análise de variância unidirecional de variância de Kruskal-Wallis por *ranks* (Nemenyi). Quando o número de blocos ou tratamentos é grande, a distribuição de probabilidade pode ser aproximada por qui-quadrado ou distribuição.

No teste de Friedman (FRIEDMAN, 1940) dado  $L$  conjuntos de dados e  $C$  algoritmos, com seus rankings médios  $B_j$ , onde  $j = 1, 2, 3, \dots, C$ ; assume-se a hipótese nula  $H_0$  de que os algoritmos são equivalentes. Neste caso, a estatística é distribuída de acordo com a distribuição  $F$  como demonstrado na Equação 2.1 com  $(C - 1)$  e  $(C - 1)(L - 1)$  graus de liberdade.

$$F_{(F)} = \frac{(L - 1) * X_F^2}{L * (C - 1) - X_F^2} \quad (2.1)$$

onde:  $X_F^2 = \frac{12L}{C(C+1)} (\sum_j B_j^2 - \frac{C(C+1)^2}{4})$

Se por acaso a hipótese nula  $H_0$  for rejeitada pelo teste de Friedman, os testes *post-hoc* de Nemenyi podem ser aplicados para apontar significância na comparação dos algoritmos, um-contrum e um-contratodos respectivamente.

Em relação ao teste de Nemenyi (NEMENYI, 1963) é utilizado quando se deseja realizar a comparação entre os algoritmos para se verificar quais são equivalentes entre si. A performance entre eles é significativamente diferente se os *rankings* médios forem diferentes entre si como demonstrado na Equação 2.2 onde o valor crítico  $q_\alpha$  é função do número de algoritmos e CD é a distância crítica.

$$CD = q_\alpha \sqrt{\frac{C(C+1)}{6L}} \quad (2.2)$$

## 2.9 Waikato Environment for Knowledge Analysis (Weka)

Weka é um software desenvolvido em Java de código aberto emitido sob a Licença Pública Geral GNU, criado pela universidade da Nova Zelândia (*The University of Waikato*). Se tratando de uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Contendo ferramentas para preparação de dados, classificação, regressão, agrupamento, mineração de regras de associação e visualização (HALL et al., 2009).

De acordo com (WITTEN; FRANK, 2005) Weka fornece implementações de algoritmos de aprendizagem que pode ser aplicado facilmente em um conjunto de dados. O ambiente de trabalho inclui métodos para todos os problemas de mineração de dados padrão: regressão, classificação, agrupamento, mineração de regras de associação e seleção de atributos. Esse *software* fornece visualização e ferramentas de pré-processamento de dados. Uma maneira de usar Weka é aplicar um método de aprendizado a um conjunto de dados e analisar sua saída para aprender mais sobre os dados. Outra é usar modelos aprendidos para gerar previsões sobre novas instâncias. Um terceiro é aplicar vários classificadores diferentes e comparar seu desempenho para escolher um para predição.

O Weka tem como objetivo principal reunir determinados algoritmos provenientes de diferentes abordagens e/ou paradigmas destinado ao estudo de aprendizagem de máquina. Onde por meio dessa ótica pretende-se desenvolver algoritmos e técnicas que permitam que um computador possa aprender, no sentido de obter novo conhecimento, quer indutiva quer dedutivamente.

### 2.9.1 Dataset

Em (HALL et al., 2009), um conjunto de itens de dados, é aproximadamente equivalente a uma planilha bidimensional ou tabela de banco de dados. No WEKA, ele é implementado pela

classe `weka.core.Instances`. Cada instância consiste em um número de atributos, qualquer um dos quais pode ser nominal, igual a uma lista pré-definida de valores, numérico igual a um número real ou inteiro, ou ainda uma cadeia que corresponde a uma lista longa e arbitrária de caracteres, entre aspas duplas. A representação externa de uma classe *Instances* é um arquivo ARFF, que consiste de um cabeçalho descrevendo os tipos de atributos e os dados como uma lista separada por vírgulas.

### 2.9.2 Formato ARFF

De acordo com (WITTEN et al., 2016) o formato ARFF simplesmente fornece um conjunto de dados; não especifica quais dos atributos serão previstos. Isso significa que o mesmo arquivo pode ser usado para investigar quão bem cada atributo pode ser previsto dos outros, para localizar regras de associação ou para *clustering*. Um arquivo ARFF consiste em uma lista de instâncias, e os valores do atributo para cada instância são separados por vírgulas. A maioria dos programas de planilhas e banco de dados permitem exportar dados para um arquivo em formato de valor separado por vírgula CSV (*Comma-Separated Values*) como uma lista de registros com vírgulas. Na formação do arquivo basta adicionar o nome do conjunto de dados usando a tag *@relation* que é o nome interno do arquivo, as informações de atributo usando a tag *@attribute* e a tag *@data* para os dados separados por vírgula.

### 2.9.3 Classifier

De acordo com (HALL et al., 2009) no Weka os algoritmos de classificação e regressão recebem a denominação de *Classifier* e qualquer algoritmo de aprendizado é derivado da classe *abstract* `weka.classifiers.AbstractClassifier`. Isso, por sua vez, implementa o `weka.classifiers.Classifier`. Mesmo um classificador básico precisa de uma rotina que gere um modelo para um conjunto de dados de treinamento e outra rotina que avalie o modelo gerado nesse conjunto de dados ou gere uma distribuição de probabilidade para todas as classes. Um modelo de classificador é um mapeamento complexo arbitrário de um conjunto de dados chamado de *all-but-one dataset* que nada mais é do que atributos para o atributo de classe. A forma específica e a criação desse mapeamento, ou modelo, que difere de classificador para classificador. Nesse trabalho foram usados 12 algoritmos de aprendizado que estão contidos nas classes do Weka onde os 4 primeiros foram utilizados tanto para classificação como para regressão. Do quinto ao oitavo somente para regressão e do nono em diante para classificar.

1. **weka.classifiers.trees.RandomForest:** O algoritmo *Random Forest* é uma coleção de árvores, todas ligeiramente diferentes que pertence à aprendizagem supervisionada. Esse algoritmo pode ser usado para vários tipos de problemas de classificação e regressão. De acordo com (BREIMAN, 2001) esse algoritmo é um classificador de conjunto que consiste em muitas árvores de decisão e gera a classe que é o modo das classes produzidas por árvores individuais. Esse algoritmo de aprendizado constrói um conjunto de classificadores e, em seguida, classifica novos pontos de dados tomando um voto (ponderado) de suas previsões.
2. **weka.classifiers.trees.RandomTree:** Classe para construção de uma árvore que considera K atributos escolhidos aleatoriamente em cada nó, não realiza poda. Também possui uma opção para permitir a estimativa de probabilidades de classe (ou média no caso de regressão) com base em um conjunto de *hold-out (backfitting)*. RandomTree foi introduzida por Leo Breiman e Adele Cutler, pode-se usar para problemas de classificação e regressão.
3. **weka.classifiers.lazy.IBk:** Classificador de vizinhos mais próximos. Pode-se selecionar o valor apropriado de K com base na validação cruzada. Também pode fazer ponderação de distância (AHA; KIBLER; ALBERT, 1991). Usado para regressão e classificação.
4. **weka.classifiers.functions.MLPClassifier:** No MLP todos os atributos são padronizados, incluindo o destino. Existem vários parâmetros. O parâmetro de *ridge* é usado para determinar a penalidade no tamanho dos pesos. O número de unidades ocultas também pode ser especificado. Nota-se que números grandes produzem longos tempos de treinamento. Para melhorar a velocidade, uma versão aproximada da função logística é usada como a função de ativação padrão para a camada oculta, mas outras funções de ativação podem ser especificadas. Na camada de saída, a função sigmoide é usada para classificação. Se o sigmoide aproximado for especificado para as camadas ocultas, ele também será usado para a camada de saída. Para regressão, a função identidade é usada na função de ativação na camada de saída. Além disso, se os valores delta na etapa de retro propagação estiverem dentro da tolerância especificada pelo usuário, o gradiente não será atualizado para essa instância específica, o que economiza tempo adicional. O cálculo paralelo da função de perda e gradiente é possível quando vários núcleos de CPU estão presentes. Os dados são divididos em lotes e processados em segmentos se-

parados nesse caso. Observe que isso só melhora o tempo de execução para conjuntos de dados maiores. Os atributos nominais são processados usando o filtro *NominalToBinary* não supervisionado e os valores ausentes são substituídos globalmente usando *ReplaceMissingValues*.

5. **weka.classifiers.trees.M5P:** De acordo com (QUINLAN, 1992) e (WANG; WITTEN, 1997) esse algoritmo implementa rotinas base para gerar árvores e regras do Modelo M5. Árvores de Modelo são estruturas hierárquicas usadas para predição de variáveis dependentes contínuas. Essas árvores Modelo são caracterizadas por guardar planos de regressão linear em suas folhas. A estrutura divide o espaço dos atributos em subespaços, e em cada um desses subespaços são aproximados por uma função linear. O critério de escolha destes atributos é usualmente implementado em razão de minimizar a variação intra conjunto nos valores da classe abaixo de cada ramo. No algoritmo M5P este processo é realizado por meio do cálculo da redução do desvio padrão sob uma porção  $D$  de instancias do conjunto de treino como pode ser visualizado na Equação 2.3.

$$SDR = sd(D) - \sum_{j=1}^N \frac{D_j}{D} sd(D_j) \quad (2.3)$$

Onde  $sd(.)$  é o desvio padrão,  $D$  é a porção do conjunto de treino que atingi o nó que é testado e  $D_j$  é a parte do treino que tem como resultado o particionamento do nó. Sendo que o processo de particionamento só termina quando os valores das classes das instâncias que atingiram o nó variam muito rápido, tal que, quando seu desvio padrão é somente uma fração pequena (geralmente menos que 5%) do desvio padrão do conjunto original da instância.

6. **weka.classifiers.functions.SMOreg:** De acordo com (SHEVADE et al., 2000) e (SMOLA; SCHÖLKOPF, 2004) o algoritmo SMOreg implementa a máquina de vetores de suporte para regressão. De acordo com (CHEN et al., 2012), SMOreg é uma implementação do algoritmo de Otimização Sequencial Mínima (Sequential Minimal Optimization - SMO) para treinamento de um modelo de regressão de vetor de suporte. O algoritmo SMOreg consiste em uma implementação Java e está disponível no pacote de software WEKA.
7. **weka.classifiers.meta.AdditiveRegression:** Classificador que aprimora o desempenho de um classificador de base de regressão. Cada iteração ajusta um modelo aos resíduos

deixados pelo classificador na iteração anterior. A previsão é realizada adicionando as previsões de cada classificador. A redução do parâmetro de encolhimento (taxa de aprendizado) ajuda a evitar o *overfitting* e tem um efeito de suavização, mas aumenta o tempo de aprendizado (FRIEDMAN, 2002).

8. **weka.classifiers.functions.SimpleLinearRegression:** Aprendizagem de um modelo de regressão linear simples, rápido de treinar e pode ter um ótimo desempenho se a variável de saída para seus dados for uma combinação linear de suas entradas. Ele funciona estimando coeficientes para uma linha ou hiperplano que melhor se adapta aos dados de treinamento. Só consegue lidar com atributos numéricos. O desempenho da regressão linear pode ser reduzido se os dados de treinamento tiverem atributos de entrada altamente correlacionados. O Weka pode detectar e remover automaticamente atributos de entrada altamente correlacionados, além disso, os atributos não relacionados à variável de saída podem também afetar negativamente o desempenho, porém o Weka pode executar automaticamente a seleção de recursos para selecionar apenas os atributos relevantes, isso já é ativado por padrão e pode ser desativado. Finalmente, a implementação do Weka usa uma técnica de regularização de cumeeira *Ridge Regularization* para reduzir a complexidade do modelo aprendido. Isso é feito minimizando o quadrado da soma absoluta dos coeficientes aprendidos, o que impedirá que qualquer coeficiente específico se torne muito grande, um sinal de complexidade nos modelos de regressão.
9. **weka.classifiers.trees.J48:** É um algoritmo desenvolvido por Ross Quinlan (SALZBERG, 1994) usado para gerar uma árvore de decisão. Ele é uma implementação de código aberto em Java do algoritmo C4.5 no aplicativo de mineração de dados Weka. Por sua vez o C4.5 constrói árvores de decisão a partir de um conjunto de dados de treinamento da mesma forma que o algoritmo ID3, utilizando o conceito de Entropia. O ID3 ou (Árvore de Decisão Indutiva) se fundamenta em árvore de decisão que é um método de aprendizado simbólico. Onde a tarefa de indução é gerar toda e possível decisão pela árvore, classificando corretamente o conjunto de treinamento e selecionar o mais simples. Esse algoritmo usa o fato de que cada atributo dos dados pode ser usado para tomar uma decisão, dividindo os dados em subconjuntos menores, sendo criadas sub-árvores entre a raiz e as folhas. As árvores de decisão geradas pelo J48 podem ser usadas para classificação e, por esse motivo é frequentemente chamado de classificador estatístico. O critério é o ganho de informação obtida na escolha do atributo para subdivisão. De

acordo com (SALZBERG, 1994) o J48 examina o ganho de informação normalizada, ou sejam a diferença de entropia, que vem da escolha de um atributo para dividir os dados. Para tomar a decisão, o atributo com maior ganho de informação é utilizado, desta forma o algoritmo repete nos subconjuntos menores. O processo de divisão é interrompido se todas as ocorrências em um subconjunto pertencem à mesma classe. Em seguida, um nó folha com a classe selecionada é criado na árvore de decisão.

10. **weka.classifiers.rules.OneR:** OneR, abreviação de *One Rule*, é um algoritmo de classificação simples, mas preciso, que gera uma regra para cada preditor nos dados e, em seguida, seleciona a regra com o menor erro total como "regra única" (HOLTE, 1993).
11. **weka.classifiers.rules.JRip:** É um classificador que implementa um aprendizado de regra proposicional, RIPPER (*Repeated Incremental Pruning to Produce Error Reduction*), que foi proposta por William W. Cohen (COHEN, 1995) como uma versão otimizada do IREP (*Incremental Reduced Error Pruning*). O método IREP utiliza árvores de decisão e as simplifica pela redução do erro, com um algoritmo trabalhando a técnica de dividir para conquistar. Quando uma regra é encontrada, todos os exemplos que são cobertos por ela são deletados. Um caminho para melhorar a abordagem incremental do IREP é adiar o processo de produção de regras deste método, assim esse método se aproxima do método de poda pelo erro, forma de otimização pelo qual o algoritmo JRip é conhecido.
12. **weka.classifiers.bayes.NaiveBayes:** É uma classe para o classificador *Naive Bayes* usando classes estimadoras. Os valores de precisão do estimador numérico são escolhidos com base na análise dos dados de treinamento. Por esse motivo, o classificador não é um *UpdateableClassifier* que no uso típico é inicializado com zero instâncias de treinamento. O classificador *NaiveBayesUpdateable* usará uma precisão padrão de 0,1 para atributos numéricos quando o *buildClassifier* for chamado com zero instâncias de treinamento (JOHN; LANGLEY, 1995).

## 2.10 Validação Cruzada (Cross-Validation)

A validação cruzada é um procedimento de reamostragem usado para avaliar modelos de aprendizado de máquina em uma amostra de dados limitada e é uma técnica importante de estatística. Esse procedimento tem um único parâmetro chamado *k* que se refere ao número de

grupos para os quais uma determinada amostra de dados deve ser dividida. Ele é frequentemente chamado de validação cruzada k-folds. Quando um valor específico para k é escolhido, ele pode ser usado no lugar de k na referência ao modelo, como  $k = 10$ , tornando-se uma validação cruzada de 10 vezes. A forma padrão de prever a taxa de erro de uma técnica de aprendizagem dada uma única amostra fixa de dados é usar a validação cruzada estratificada de dez vezes. Em (HALL et al., 2009) menciona que os dados são divididos aleatoriamente em 10 partes nas quais a classe é representada em aproximadamente com as mesmas proporções do conjunto de dados completo. Cada parte é retida por sua vez e o esquema de aprendizagem treina nos nove décimos restantes, então sua taxa de erro é calculada no conjunto de validação. Assim, o procedimento de aprendizagem é executado em um total de 10 vezes em diferentes conjuntos de treinamento. Finalmente, as 10 estimativas de erro são calculadas para gerar uma estimativa global de erros.

A validação cruzada é usada principalmente no aprendizado de máquina aplicado para estimar a habilidade de um modelo de aprendizado de máquina em dados não vistos. Ou seja, usar uma amostra limitada para estimar o desempenho esperado do modelo em geral quando usado para fazer previsões sobre dados não utilizados durante o treinamento do modelo. De acordo com (WITTEN et al., 2016) os testes extensivos em vários conjuntos de dados diferentes, com diferentes técnicas de aprendizado, mostraram que 10 é o número certo de partições dos dados para obter a melhor estimativa de erro, e há também algumas evidências teóricas que apoia essa afirmação. Embora esses argumentos não sejam conclusivos, a validação cruzada de 10 vezes tornou-se o método padrão em termos práticos. Testes também mostraram que o uso de estratificação melhora ligeiramente os resultados. Assim, a técnica de avaliação padrão em situações em que apenas dados limitados estão disponíveis é estratificada com a validação cruzada de 10 vezes. Nem a estratificação nem a divisão em 10 partições têm que ser exatas, basta dividir os dados em aproximadamente 10 conjuntos iguais em que os vários valores de classe são representados aproximadamente na proporção certa como demonstrado na Figura 2.5. Nessa figura cada bloco numerado de 1 até 272 correspondem ao conjunto de dados extraídos de cada imagem com seu respectivo valor do MOS da base de dados or105. No Weka os blocos são randomizados, isso é, dificilmente eles terão à disposição mostrada na figura 2.5, mas para efeito didático tomar-se-á isso como uma das disposições aceitáveis, onde  $k_1, k_2, \dots, k_{10}$  representam os k grupos, com o  $k_{10}$  tendo 29 instâncias e os demais apenas 27 cada.

Figura 2.5 – Representação do K-FOLD com K=10.

10-FOLDS

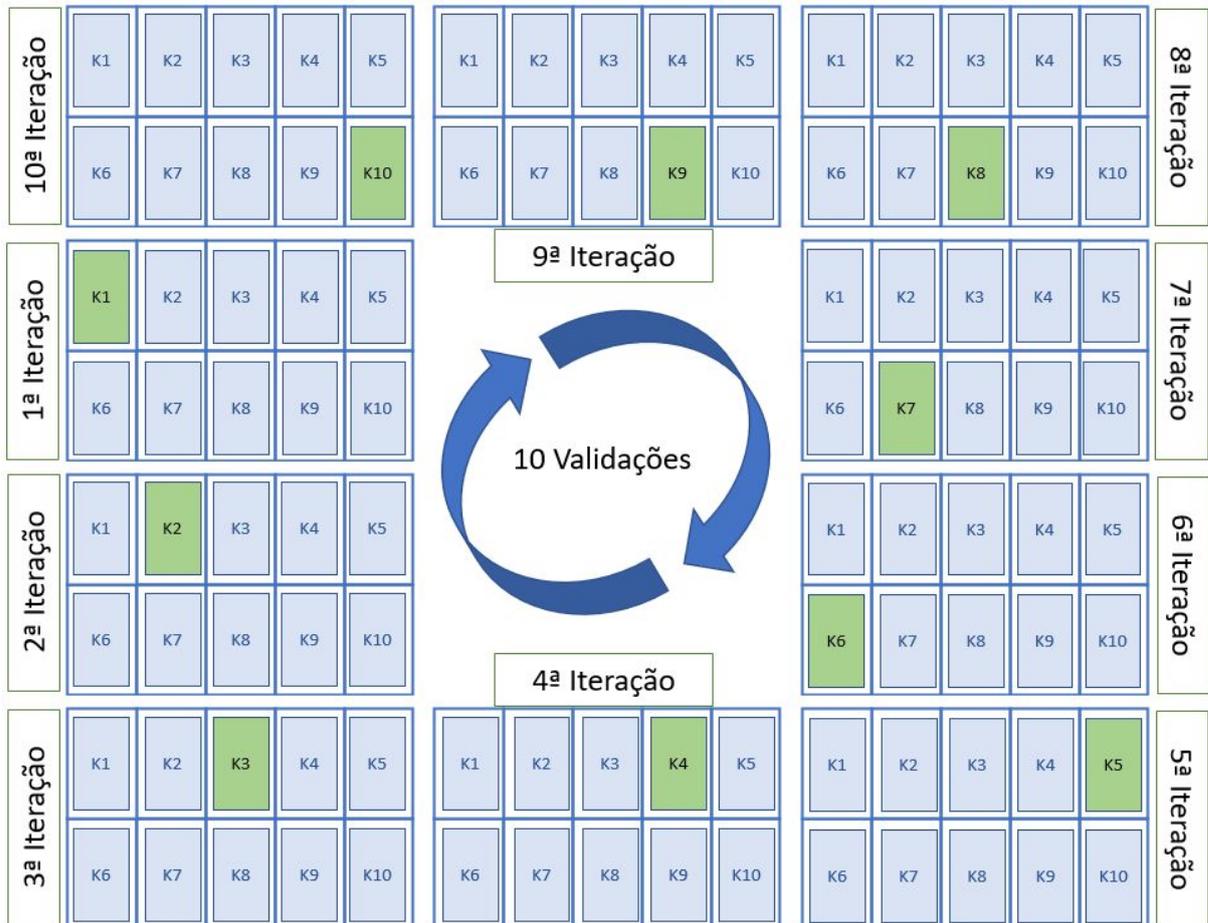
<table border="1"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>4</td><td>5</td><td>6</td></tr> <tr><td>7</td><td>8</td><td>9</td></tr> <tr><td>10</td><td>11</td><td>12</td></tr> <tr><td>13</td><td>14</td><td>15</td></tr> <tr><td>16</td><td>17</td><td>18</td></tr> <tr><td>19</td><td>20</td><td>21</td></tr> <tr><td>22</td><td>23</td><td>24</td></tr> <tr><td>25</td><td>26</td><td>27</td></tr> </table> <p>k1</p>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	<table border="1"> <tr><td>55</td><td>56</td><td>57</td></tr> <tr><td>58</td><td>59</td><td>60</td></tr> <tr><td>61</td><td>62</td><td>63</td></tr> <tr><td>64</td><td>65</td><td>66</td></tr> <tr><td>67</td><td>68</td><td>69</td></tr> <tr><td>70</td><td>71</td><td>72</td></tr> <tr><td>73</td><td>74</td><td>75</td></tr> <tr><td>76</td><td>77</td><td>78</td></tr> <tr><td>79</td><td>80</td><td>81</td></tr> </table> <p>k2</p>	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	<table border="1"> <tr><td>109</td><td>110</td><td>111</td></tr> <tr><td>112</td><td>113</td><td>114</td></tr> <tr><td>115</td><td>116</td><td>117</td></tr> <tr><td>118</td><td>119</td><td>120</td></tr> <tr><td>121</td><td>122</td><td>123</td></tr> <tr><td>124</td><td>125</td><td>126</td></tr> <tr><td>127</td><td>128</td><td>129</td></tr> <tr><td>130</td><td>131</td><td>132</td></tr> <tr><td>133</td><td>134</td><td>135</td></tr> </table> <p>k3</p>	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	<table border="1"> <tr><td>163</td><td>164</td><td>165</td></tr> <tr><td>166</td><td>167</td><td>168</td></tr> <tr><td>169</td><td>170</td><td>171</td></tr> <tr><td>172</td><td>173</td><td>174</td></tr> <tr><td>175</td><td>176</td><td>177</td></tr> <tr><td>178</td><td>179</td><td>180</td></tr> <tr><td>181</td><td>182</td><td>183</td></tr> <tr><td>184</td><td>185</td><td>186</td></tr> <tr><td>187</td><td>188</td><td>189</td></tr> </table> <p>k4</p>	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	<table border="1"> <tr><td>217</td><td>218</td><td>219</td></tr> <tr><td>220</td><td>221</td><td>222</td></tr> <tr><td>223</td><td>224</td><td>225</td></tr> <tr><td>226</td><td>227</td><td>228</td></tr> <tr><td>229</td><td>230</td><td>231</td></tr> <tr><td>232</td><td>233</td><td>234</td></tr> <tr><td>235</td><td>236</td><td>237</td></tr> <tr><td>238</td><td>239</td><td>240</td></tr> <tr><td>241</td><td>242</td><td>243</td></tr> </table> <p>k5</p>	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243			
1	2	3																																																																																																																																												
4	5	6																																																																																																																																												
7	8	9																																																																																																																																												
10	11	12																																																																																																																																												
13	14	15																																																																																																																																												
16	17	18																																																																																																																																												
19	20	21																																																																																																																																												
22	23	24																																																																																																																																												
25	26	27																																																																																																																																												
55	56	57																																																																																																																																												
58	59	60																																																																																																																																												
61	62	63																																																																																																																																												
64	65	66																																																																																																																																												
67	68	69																																																																																																																																												
70	71	72																																																																																																																																												
73	74	75																																																																																																																																												
76	77	78																																																																																																																																												
79	80	81																																																																																																																																												
109	110	111																																																																																																																																												
112	113	114																																																																																																																																												
115	116	117																																																																																																																																												
118	119	120																																																																																																																																												
121	122	123																																																																																																																																												
124	125	126																																																																																																																																												
127	128	129																																																																																																																																												
130	131	132																																																																																																																																												
133	134	135																																																																																																																																												
163	164	165																																																																																																																																												
166	167	168																																																																																																																																												
169	170	171																																																																																																																																												
172	173	174																																																																																																																																												
175	176	177																																																																																																																																												
178	179	180																																																																																																																																												
181	182	183																																																																																																																																												
184	185	186																																																																																																																																												
187	188	189																																																																																																																																												
217	218	219																																																																																																																																												
220	221	222																																																																																																																																												
223	224	225																																																																																																																																												
226	227	228																																																																																																																																												
229	230	231																																																																																																																																												
232	233	234																																																																																																																																												
235	236	237																																																																																																																																												
238	239	240																																																																																																																																												
241	242	243																																																																																																																																												
<table border="1"> <tr><td>28</td><td>29</td><td>30</td></tr> <tr><td>31</td><td>32</td><td>33</td></tr> <tr><td>34</td><td>35</td><td>36</td></tr> <tr><td>37</td><td>38</td><td>39</td></tr> <tr><td>40</td><td>41</td><td>42</td></tr> <tr><td>43</td><td>44</td><td>45</td></tr> <tr><td>46</td><td>47</td><td>48</td></tr> <tr><td>49</td><td>50</td><td>51</td></tr> <tr><td>52</td><td>53</td><td>54</td></tr> </table> <p>k6</p>	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	<table border="1"> <tr><td>82</td><td>83</td><td>84</td></tr> <tr><td>85</td><td>86</td><td>87</td></tr> <tr><td>88</td><td>89</td><td>90</td></tr> <tr><td>91</td><td>92</td><td>93</td></tr> <tr><td>94</td><td>95</td><td>96</td></tr> <tr><td>97</td><td>98</td><td>99</td></tr> <tr><td>100</td><td>101</td><td>102</td></tr> <tr><td>103</td><td>104</td><td>105</td></tr> <tr><td>106</td><td>107</td><td>108</td></tr> </table> <p>k7</p>	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	<table border="1"> <tr><td>136</td><td>137</td><td>138</td></tr> <tr><td>139</td><td>140</td><td>141</td></tr> <tr><td>142</td><td>143</td><td>144</td></tr> <tr><td>145</td><td>146</td><td>147</td></tr> <tr><td>148</td><td>149</td><td>150</td></tr> <tr><td>151</td><td>152</td><td>153</td></tr> <tr><td>154</td><td>155</td><td>156</td></tr> <tr><td>157</td><td>158</td><td>159</td></tr> <tr><td>160</td><td>161</td><td>162</td></tr> </table> <p>k8</p>	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	<table border="1"> <tr><td>190</td><td>191</td><td>192</td></tr> <tr><td>193</td><td>194</td><td>195</td></tr> <tr><td>196</td><td>197</td><td>198</td></tr> <tr><td>199</td><td>200</td><td>201</td></tr> <tr><td>202</td><td>203</td><td>204</td></tr> <tr><td>205</td><td>206</td><td>207</td></tr> <tr><td>208</td><td>209</td><td>210</td></tr> <tr><td>211</td><td>212</td><td>213</td></tr> <tr><td>214</td><td>215</td><td>216</td></tr> </table> <p>k9</p>	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	<table border="1"> <tr><td>244</td><td>245</td><td>246</td></tr> <tr><td>247</td><td>248</td><td>249</td></tr> <tr><td>250</td><td>251</td><td>252</td></tr> <tr><td>253</td><td>254</td><td>255</td></tr> <tr><td>256</td><td>257</td><td>258</td></tr> <tr><td>259</td><td>260</td><td>261</td></tr> <tr><td>262</td><td>263</td><td>264</td></tr> <tr><td>265</td><td>266</td><td>267</td></tr> <tr><td>268</td><td>269</td><td>270</td></tr> <tr><td>271</td><td>272</td><td></td></tr> </table> <p>k10</p>	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	
28	29	30																																																																																																																																												
31	32	33																																																																																																																																												
34	35	36																																																																																																																																												
37	38	39																																																																																																																																												
40	41	42																																																																																																																																												
43	44	45																																																																																																																																												
46	47	48																																																																																																																																												
49	50	51																																																																																																																																												
52	53	54																																																																																																																																												
82	83	84																																																																																																																																												
85	86	87																																																																																																																																												
88	89	90																																																																																																																																												
91	92	93																																																																																																																																												
94	95	96																																																																																																																																												
97	98	99																																																																																																																																												
100	101	102																																																																																																																																												
103	104	105																																																																																																																																												
106	107	108																																																																																																																																												
136	137	138																																																																																																																																												
139	140	141																																																																																																																																												
142	143	144																																																																																																																																												
145	146	147																																																																																																																																												
148	149	150																																																																																																																																												
151	152	153																																																																																																																																												
154	155	156																																																																																																																																												
157	158	159																																																																																																																																												
160	161	162																																																																																																																																												
190	191	192																																																																																																																																												
193	194	195																																																																																																																																												
196	197	198																																																																																																																																												
199	200	201																																																																																																																																												
202	203	204																																																																																																																																												
205	206	207																																																																																																																																												
208	209	210																																																																																																																																												
211	212	213																																																																																																																																												
214	215	216																																																																																																																																												
244	245	246																																																																																																																																												
247	248	249																																																																																																																																												
250	251	252																																																																																																																																												
253	254	255																																																																																																																																												
256	257	258																																																																																																																																												
259	260	261																																																																																																																																												
262	263	264																																																																																																																																												
265	266	267																																																																																																																																												
268	269	270																																																																																																																																												
271	272																																																																																																																																													

Fonte: do Autor

A Figura 2.6 mostra uma única validação cruzada de 10 iterações com os dados da Figura 2.5, onde cada retângulo verde representa o teste feito naquela iteração, uma única validação pode não ser suficiente para obter uma estimativa confiável do erro se os dados forem limitados. De acordo com (WITTEN et al., 2016) menciona que diferentes experimentos de validação cruzada de 10 em um mesmo esquema de aprendizagem e conjunto de dados, geralmente, produzem resultados diferentes, por causa do efeito da variação aleatória na escolha das próprias partições. A estratificação reduz à variação, mas certamente não a elimina totalmente. Ao procurar uma estimativa de erro precisa com dados limitados, é aceitável que se repita o procedimento padrão da validação cruzada por no mínimo 10 vezes, isto é, embaralhar aleatoriamente o conjunto de dados, dividir em k grupos, para cada iteração escolher um grupo que servirá para testar o conjunto de dados, usar os grupos restantes para treinamento, guardar a pontuação de avaliação, ao final tem-se o resultado da correlação. É importante ressaltar que cada observação na amostra de dados é atribuída a um grupo individual e permanece nesse grupo durante a duração das 10 validações. Isto significa que cada amostra tem a oportunidade de ser usada em cada iteração para treinar o modelo k-1 vezes, obtendo, assim, uma boa medida de desempenho. Ao final tem-se 90 treinamentos e 10 testes. Para uma melhor acurácia usou-se 50 embaralhamentos aleatórios, sobre um *K-fold* igual a 10, de 10 validações, ou seja, 50

randomizações x 10 validações x 10 Folds equivalendo-se a rodar o algoritmo de aprendizado 5000 iterações, onde teve-se 4500 treinamentos e 500 testes sobre 50 randomizações distintas.

Figura 2.6 – Representação de 10 Validações.



Fonte: do Autor

## 2.11 Medidas de desempenho

De acordo com (WITTEN; FRANK, 2005) são todas as medidas de avaliação que pertencem a situações de classificação em vez de situações de previsão numérica. Os princípios básicos usando um conjunto de testes independente e não o conjunto de treinamento para a avaliação do método de validação e validação cruzada, aplicam-se igualmente bem à previsão numérica. Mas a medida básica de qualidade oferecida pela taxa de erro não é a mais apropriada: os erros não estão simplesmente presentes ou ausentes, eles são diferentes. Os valores previstos nas instância de testes são  $p_1, p_2, \dots, p_n$  e os valores reais são  $a_1, a_2, \dots, a_n$ .

O erro quadrático médio (EQM), equação 2.4, é definido como sendo a média da diferença entre o valor do previsto e do real ao quadrado e pode ser usado como uma medida do

erro de previsão, onde o EQM é determinado somando-se os erros de previsão ao quadrado e dividindo pelo número total de erros usados no cálculo.

$$EQM = \frac{\sum_{i=1}^n (p_i - a_i)^2}{n} \quad (2.4)$$

O erro quadrático médio, tende a exagerar o efeito de *outliers*, isto é, instâncias cujo erro de previsão é maior que os outros, mas o erro absoluto médio (EAM), equação 2.5 não tem esse efeito, todos os tamanhos de erro são tratados uniformemente de acordo com sua magnitude. Às vezes, são os valores de erro relativos em vez de absolutos que são importantes. Por exemplo, se um erro de 10% é igualmente importante, seja um erro de 50 em uma previsão de 500 ou um erro de 0,2 em uma previsão de 2, então as médias de erro absoluto será sem sentido, nesses casos os erros relativos são apropriados. Este efeito seria levado em conta usando os erros relativos no cálculo do erro quadrático médio ou o cálculo do erro absoluto médio.

$$EAM = \frac{\sum_{i=1}^n |p_i - a_i|}{n} \quad (2.5)$$

O erro relativo quadrático (ERQ), equação 2.6, refere-se a algo bem diferente. O erro é feito em relação ao que teria sido se um simples *classifier* tivesse sido usado. A equação é apenas a média dos valores reais dos dados de treinamento, assim, o erro quadrado relativo leva em consideração o total do erro ao quadrado e normaliza dividindo pelo total do erro do preditor.

$$ERQ = \frac{\sum_{i=1}^n (p_i - a_i)^2}{\sum_{i=1}^n (a_i - \bar{a})^2}, \quad \text{onde } \bar{a} = \frac{1}{n} \sum_{i=1}^n a_i \quad (2.6)$$

O erro absoluto relativo (EAR), equação 2.7 é apenas o erro absoluto total, com o mesmo tipo de normalização. Nessas três medidas de erros relativos, os erros são normalizados pelo erro do preditor simples que prevê valores médios.

$$EAR = \frac{\sum_{i=1}^n |p_i - a_i|}{\sum_{i=1}^n |a_i - \bar{a}|} \quad \text{onde } \bar{a} = \frac{1}{n} \sum_{i=1}^n a_i \quad (2.7)$$

Coefficiente de correlação (CC) mede a correlação estatística entre os  $a$  e os  $p$  da equação 2.8. O coeficiente de correlação varia de 1 para resultados perfeitamente correlacionados, até 0 quando não há relação, para -1 quando os resultados são perfeitamente correlacionados negativamente, isto é, se um aumenta, o outro sempre diminui. A correlação é ligeiramente diferente

das outras medidas porque é independente da escala em que se toma um conjunto particular de previsões, o erro é inalterado se todas as previsões são multiplicadas por um fator constante e os valores reais são deixados inalterados. Este fator aparece em todos os termos do  $cov_{(P,A)}$  no numerador e em cada termo de  $var_{(P)}$  e  $var_{(A)}$  no denominador, cancelando assim. Porém, isto não é verdade para os números relativos ao erro, apesar da normalização, se multiplicar todas as previsões por uma constante grande, então a diferença entre o predito e os valores reais mudarão drasticamente, assim como os erros de porcentagem. Diferente em que o bom desempenho leva a um grande valor do coeficiente de correlação, enquanto que, como os outros métodos medem erro, o bom desempenho é indicado por valores pequenos.

$$CC = \frac{cov_{(P,A)}}{\sqrt{var_{(P)} * var_{(A)}}} = \frac{\frac{\sum_{i=1}^n (p_i - \bar{p})(a_i - \bar{a})}{n-1}}{\sqrt{\frac{\sum_{i=1}^n (p_i - \bar{p})^2}{n-1} * \frac{\sum_{i=1}^n (a_i - \bar{a})^2}{n-1}}} = \frac{\sum_{i=1}^n (p_i - \bar{p})(a_i - \bar{a})}{\sqrt{\sum_{i=1}^n (p_i - \bar{p})^2 * \sum_{i=1}^n (a_i - \bar{a})^2}} \quad (2.8)$$

onde:  $\bar{a} = \frac{1}{n} \sum_{i=1}^n a_i$        $\bar{p} = \frac{1}{n} \sum_{i=1}^n p_i$

Interpretando os valores de CC:

- \* 0,9 a 1,0 positivo ou negativo indica uma correlação muito forte.
- \* 0,7 a 0,9 positivo ou negativo indica uma correlação forte.
- \* 0,5 a 0,7 positivo ou negativo indica uma correlação moderada.
- \* 0,3 a 0,5 positivo ou negativo indica uma correlação fraca.
- \* 0,0 a 0,3 positivo ou negativo indica uma correlação desprezível

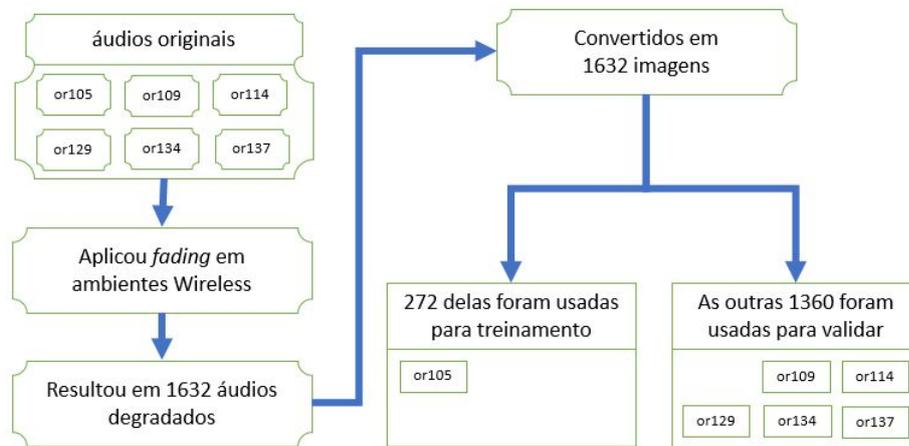
De acordo com (WITTEN; FRANK, 2005) a estatística Kappa foi introduzida por Cohen (1960) e é uma estatística que mede a concordância entre avaliadores para itens qualitativos. Geralmente, acredita-se que seja uma medida mais robusta do que o simples cálculo de percentual de concordância, já que Kappa leva em consideração a possibilidade de a concordância ocorrer por acaso. De acordo com (HALL et al., 2009) é uma medida de concordância usada em escalas nominais que fornece uma ideia do quanto as observações se afastam daquelas esperadas, fruto do acaso, indicando assim o quão legítimas as interpretações são onde o valor numérico 1,0 significa acordo completo. A magnitude da Estatística Kappa é uma medida de concordância bastante mais significativa do que a sua própria significância estatística. As diretrizes para a interpretação de Kappa são dadas a seguir.

- \* 0 indica concordância pobre.
- \* 0 a 0,2 indica concordância ligeira.
- \* 0,21 a 0,4 indica concordância considerável.
- \* 0,41 a 0,6 indica concordância moderada.
- \* 0,61 a 0,8 indica concordância substancial.
- \* 0,81 a 1,0 indica concordância excelente.

### 3 METODOLOGIA

Neste capítulo será apresentada a metodologia utilizada para a realização desta pesquisa, que foi dividida em duas etapas, dois passos na primeira e um na segunda, totalizando 3. A Figura 3.1 mostra o esquema adotado, onde por meio da implementação de um algoritmo usando o *Matlab*, denominado de cenário geral cujo objetivo foi degradar seis arquivos de áudio diferentes que geraram 1632 arquivos de áudios desvanecidos, os testes foram executados de modo a simular uma comunicação semelhante a um ambiente real.

Figura 3.1 – Esquema adotado.



Fonte: do Autor

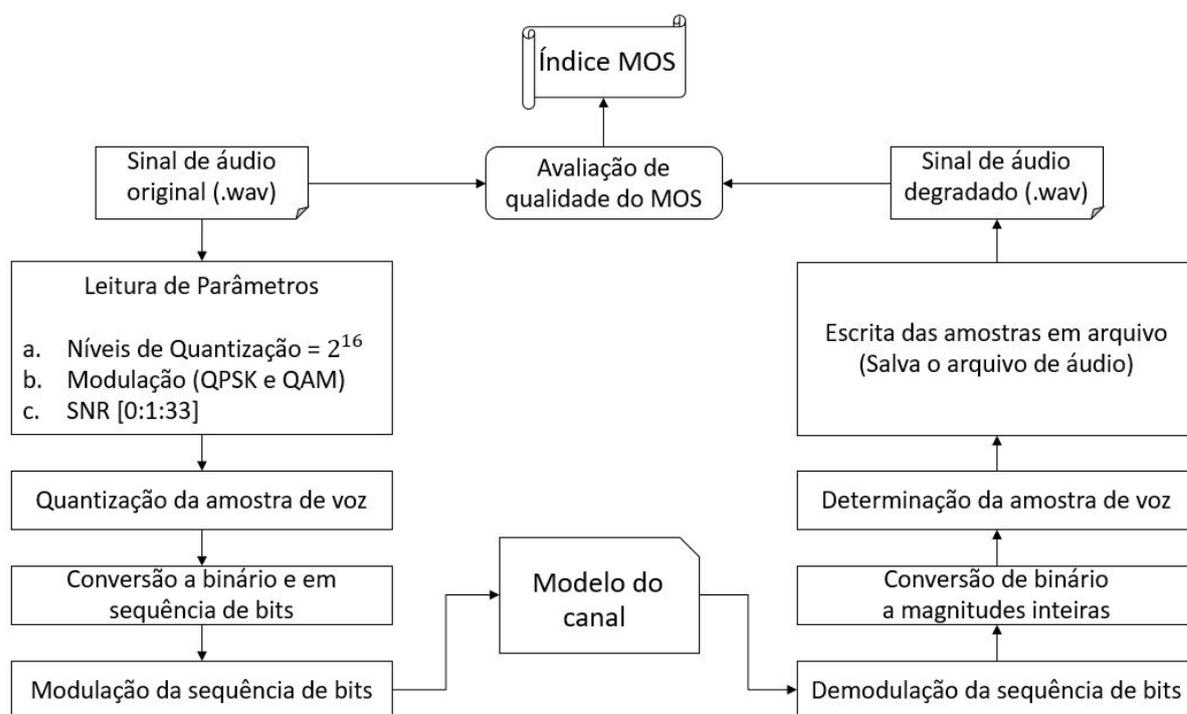
No primeiro passo da etapa 1, denominada de "Passo 1 - Formação do Banco de Dados", escolheu-se aleatoriamente seis arquivos de áudios originais (or105, or109, or114, or129, or134 e or137) todos pertencentes à base de dados da recomendação P.862 (ANNEX, 2005) no idioma inglês. Tais áudios foram usados para fazer aplicações de degradações em ambientes que simulam redes *Wireless* chamado de cenário geral, no qual dois parâmetros foram alterados, sendo eles: Relação Sinal-Ruído (SNR) e modulações QPSK e QAM. O nível de quantização teve um valor fixo de  $2^{16}$  para todos. As alterações desses parâmetros modificaram o sinal de fala, para avaliar a qualidade da voz sobre redes IP foi utilizado a recomendação ITU-T P.862 (P.862, 2007). Para realizar estas degradações, algoritmos implementados em *MATLAB* foram utilizados para simular o efeito do *fading* em ambientes *Wireless*. Por meio dessa implementação foi possível gerar diferentes sinais de voz degradada, deste modo, obtendo uma saída coerente com fenômenos do mundo real em comunicações *Wireless* (MATLAB, 2018).

O Passo 2 se refere à extração de características de gráficos no domínio do tempo referente a somente as 272 degradações do arquivo original or105, o tratamento dos dados, o teste

de hipótese de Friedman-Nemenyi e a construção dos modelos no Weka. E por fim, o Passo 3 se refere à implementação do *software* feito em Java denominado de PredictorFX, cuja função foi usar os modelos construídos no passo anterior para fazer previsão dos gráficos que representam seus respectivos arquivos de áudios degradados vindos dos outros cinco arquivos originais sendo eles o or109, or114, or129, or134 e or137.

### 3.1 Cenário geral

Figura 3.2 – Cenário geral que simula uma comunicação semelhante a um ambiente real.



Fonte: do Autor

A Figura 3.2 mostra o cenário geral com os passos seguidos pelo código na obtenção do índice MOS. Esse cenário dispõe de dois parâmetros, sendo eles a modulação que alternou entre QPSK, 2-QAM, 4-QAM, 16-QAM, 32-QAM, 64-QAM, 128-QAM, 256-QAM e relação sinal-ruído que alterou de 0 dB a 33 dB, variando de um em um. Esses três parâmetros simularam diferentes tipos de desvanecimentos como os que ocorrem em uma transmissão *Wireless*. O algoritmo do cenário geral carregou cada arquivo de áudio original, sem qualquer tipo de degradação, fazendo a quantização linear, em seguida, houve a conversão para binário e o sequenciamento desses bits. Dependendo da configuração, os bits sequenciados sofrem modulações QPSK ou QAM, desse modo, são gerados distintos tipos de desvanecimento. Após esse desvanecimento existe a sequência para agrupamento e posteriormente demodulação onde há

a conversão de binário para arquivo de som com a degradação, gerando desse modo diferentes sinais de áudio degradado, além de ter uma saída coerente com fenômenos do mundo real em comunicações *Wireless* (MATLAB, 2018). O esquema detalhado pode ser visto na Figura 3.3, a Tabela 3.1 contém as informações dos sinais de áudio originais (sem degradações) e a Figura 3.4 mostra os arquivos plotados no domínio do tempo.

Figura 3.3 – Esquema detalhado.



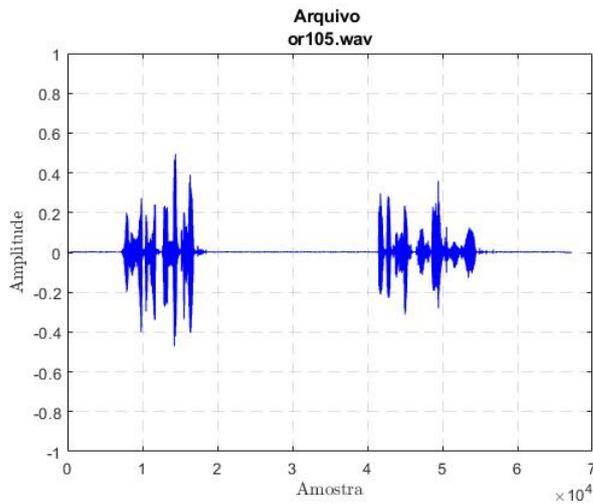
Fonte: do Autor

Tabela 3.1 – Informações dos áudios

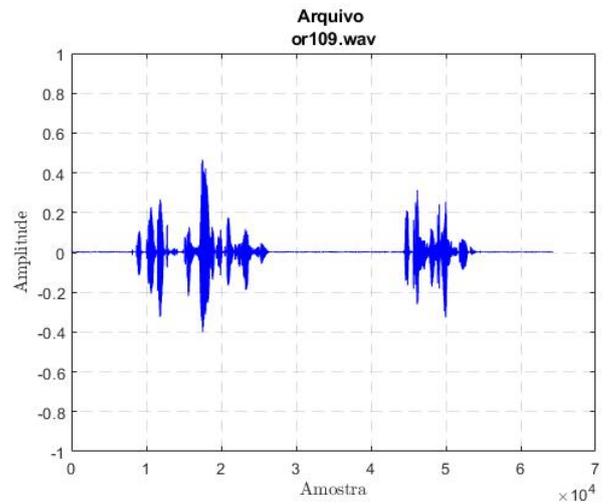
	<b>or105</b>	<b>or109</b>	<b>or114</b>	<b>or129</b>	<b>or134</b>	<b>or137</b>
Áudio Descompactado	Sim	Sim	Sim	Sim	Sim	Sim
Número de Canais	1	1	1	1	1	1
Taxa de Amostragem	8 kHz					
Total de amostras	67220	64314	68734	57849	64605	56474
Duração do áudio	8s 402ms	8s 390ms	8s 591ms	7s 231ms	8s 750ms	7s 593ms
Bit/amostra	16 bits					
Taxa de bits	128 Kbps					
Formato	Wave	Wave	Wave	Wave	Wave	Wave
Tamanho do arquivo	131 KB	126 KB	134 KB	113 KB	126 KB	110 KB
Sinal Sonoro	3,82 %	3,30 %	3,43 %	2,76 %	2,81 %	2,81 %
Ausência de Sinal Sonoro	96,18 %	96,70 %	96,57 %	97,24 %	97,19 %	97,19 %

Fonte: do autor

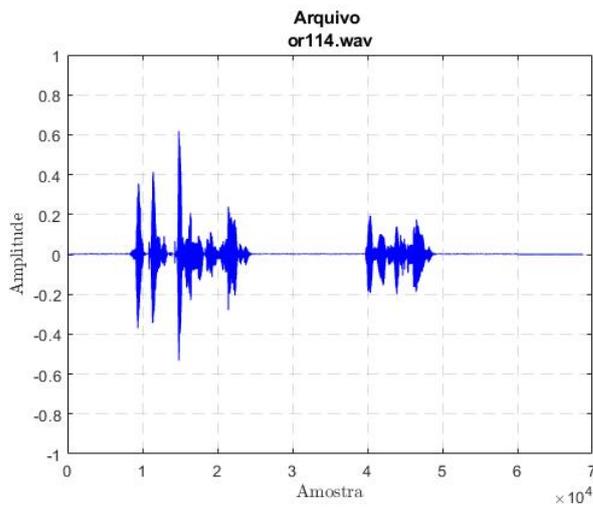
Figura 3.4 – Imagens dos áudios originais or105, or109, or114, or129, or134 e or137.



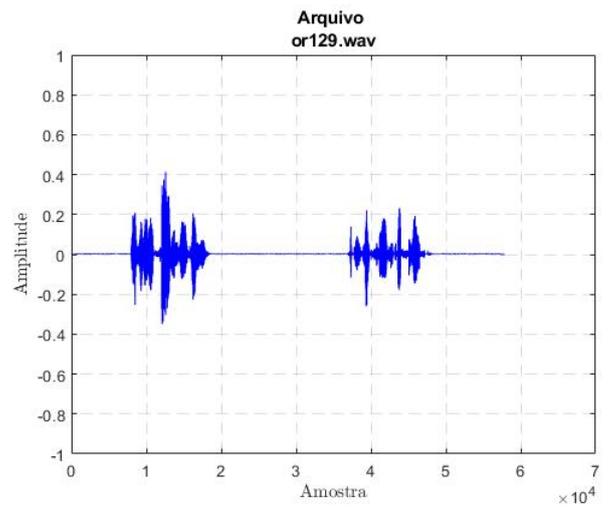
[a] Arquivo de áudio original or105



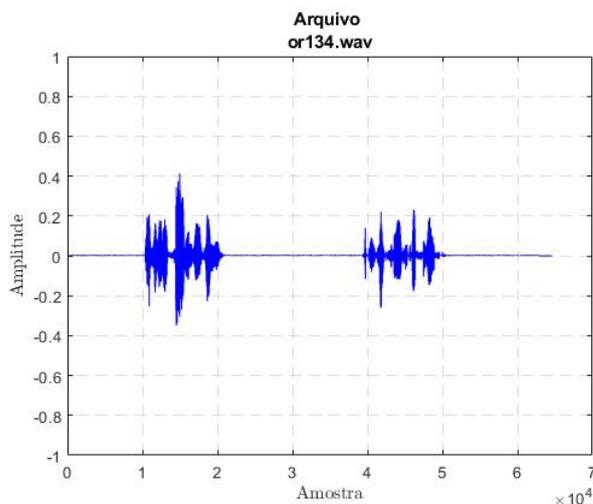
[b] Arquivo de áudio original or109



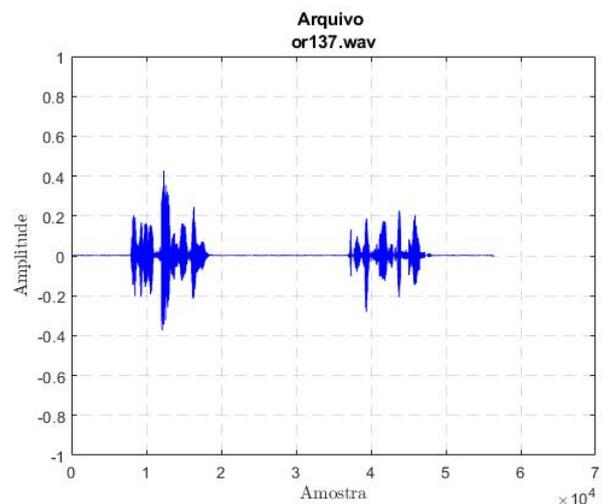
[c] Arquivo de áudio original or114



[d] Arquivo de áudio original or129



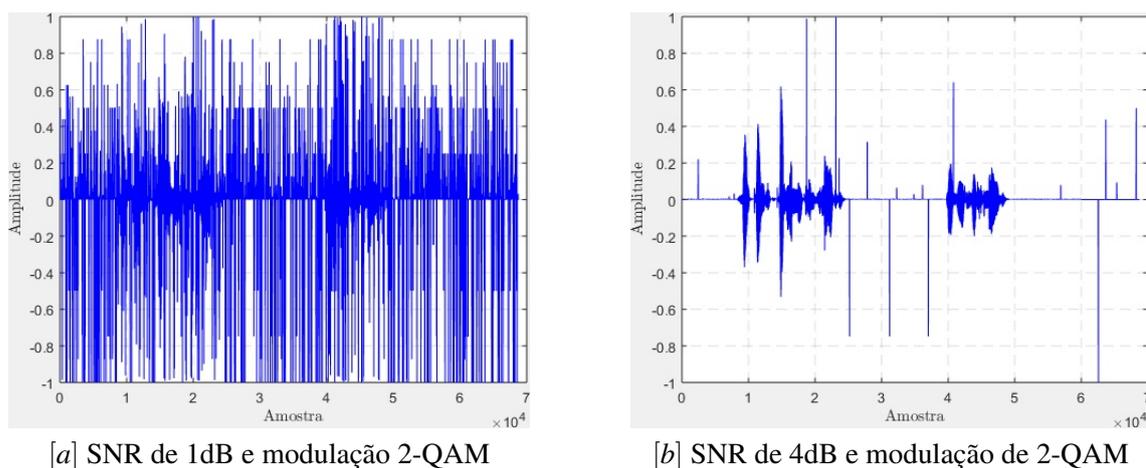
[e] Arquivo de áudio original or134



[f] Arquivo de áudio original or137

Após obter os 1632 áudios desvanecidos foi preciso fazer a plotagem desses arquivos, para isso foi implementado em *Matlab* o código denominado de P1F1 (Passo 1 Ferramenta 1). Esse código facilita o processo de aquisição dos *plots* dos arquivos de áudio desvanecidos no domínio do tempo e os salvam em arquivos com extensão *jpg*. As Figuras 3.5 [a] e 3.5 [b] mostram duas configurações para o nível de quantização igual a  $2^{16}$ , modulação 2-QAM para o áudio or114. Sendo que em [a] o valor do SNR é igual a 1dB (degradação p01), onde mostra o sinal de voz bastante degradado com um MOS igual a 1 e em [b] com um SNR de 4dB, demonstra um áudio um pouco melhor com MOS de 2,1. Os áudios degradados conservam as mesmas informações da Tabela 3.1 a única diferença é a qualidade do áudio onde as figuras demonstram ruídos, diferentemente ao que é mostrado na Figura 3.4 [c].

Figura 3.5 – Imagens do áudio desvanecido or114.



Fonte: do Autor

### 3.2 Mineração de Dados

A Tabela 3.2 refere aos MOS obtidos dos arquivos de desvanecimento, optou-se por seleccionar apenas o conjunto de imagens do áudio original or105 com 272 arquivos, poderia ter sido qualquer um dos seis.

Tabela 3.2 – Valores do MOS para os áudios desvanecidos do arquivo original or105

Repetição	QPSK	2-QAM	4-QAM	16-QAM	32-QAM	64-QAM	128-QAM	256-QAM
<i>p</i> <sub>00</sub>	1,1	1,2	1,1	1,1	1,2	1,0	1,1	1,2
<i>p</i> <sub>01</sub>	1,1	1,3	1,1	1,1	1,1	0,9	1,1	1,2
<i>p</i> <sub>02</sub>	1,1	1,2	1,0	1,1	1,1	1,0	1,0	1,1
<i>p</i> <sub>03</sub>	1,1	0,8	1,1	1,1	1,0	1,0	1,1	1,1
<i>p</i> <sub>04</sub>	1,1	1,9	1,1	1,0	1,1	1,0	1,0	1,2
<i>p</i> <sub>05</sub>	1,2	3,4	1,2	1,0	1,0	1,2	1,1	1,1
<i>p</i> <sub>06</sub>	1,4	4,5	1,3	1,0	1,0	1,0	1,0	1,0
<i>p</i> <sub>07</sub>	1,0	4,5	1,0	1,0	1,0	1,0	1,1	1,0
<i>p</i> <sub>08</sub>	1,0	4,5	1,0	1,0	1,0	1,2	1,0	1,1
<i>p</i> <sub>09</sub>	2,3	4,5	2,2	1,1	1,0	1,0	1,0	1,1
<i>p</i> <sub>10</sub>	4,5	4,5	3,9	1,1	1,0	1,1	1,0	1,0
<i>p</i> <sub>11</sub>	4,5	4,5	4,4	1,2	1,1	1,1	1,0	1,0
<i>p</i> <sub>12</sub>	4,5	4,5	4,5	1,3	1,0	1,0	1,2	1,0
<i>p</i> <sub>13</sub>	4,5	4,5	4,5	1,0	1,1	1,1	1,0	1,0
<i>p</i> <sub>14</sub>	4,5	4,5	4,5	0,7	1,1	1,1	1,0	1,1
<i>p</i> <sub>15</sub>	4,5	4,5	4,5	1,5	1,2	1,0	1,0	1,1
<i>p</i> <sub>16</sub>	4,5	4,5	4,5	3,1	1,3	1,1	1,1	1,1
<i>p</i> <sub>17</sub>	4,5	4,5	4,5	4,2	1,4	1,2	1,1	1,1
<i>p</i> <sub>18</sub>	4,5	4,5	4,5	4,5	1,4	1,3	1,1	1,2
<i>p</i> <sub>19</sub>	4,5	4,5	4,5	4,5	1,0	0,8	1,2	1,2
<i>p</i> <sub>20</sub>	4,5	4,5	4,5	4,5	1,0	0,6	1,2	1,3
<i>p</i> <sub>21</sub>	4,5	4,5	4,5	4,5	1,9	1,5	1,3	1,4
<i>p</i> <sub>22</sub>	4,5	4,5	4,5	4,5	3,2	3,0	1,3	1,4
<i>p</i> <sub>23</sub>	4,5	4,5	4,5	4,5	4,1	4,2	1,4	1,4
<i>p</i> <sub>24</sub>	4,5	4,5	4,5	4,5	4,5	4,5	1,4	1,0
<i>p</i> <sub>25</sub>	4,5	4,5	4,5	4,5	4,5	4,5	1,0	0,8
<i>p</i> <sub>26</sub>	4,5	4,5	4,5	4,5	4,5	4,5	1,1	1,2
<i>p</i> <sub>27</sub>	4,5	4,5	4,5	4,5	4,5	4,5	2,1	2,6
<i>p</i> <sub>28</sub>	4,5	4,5	4,5	4,5	4,5	4,5	3,3	3,8
<i>p</i> <sub>29</sub>	4,5	4,5	4,5	4,5	4,5	4,5	4,2	4,4
<i>p</i> <sub>30</sub>	4,5	4,5	4,5	4,5	4,5	4,5	4,5	4,5
<i>p</i> <sub>31</sub>	4,5	4,5	4,5	4,5	4,5	4,5	4,5	4,5
<i>p</i> <sub>32</sub>	4,5	4,5	4,5	4,5	4,5	4,5	4,5	4,5
<i>p</i> <sub>33</sub>	4,5	4,5	4,5	4,5	4,5	4,5	4,5	4,5

Fonte: do autor

Para que pudesse usar os algoritmos classificadores do Weka foi necessário fazer intervalos como demonstrado na Tabela 3.3, de modo a classificar uma imagem fornecendo como resposta o MOS literal. Tal medida foi necessária, uma vez que, o valor de saída do cenário geral (Passo 1) não é um número natural, mas sim real. Por causa disso, houve a necessidade de retirar uma subdivisão da escala de opinião mencionada na recomendação ITU-T P.800 (P.800, 1998) como demonstrado na Figura 3.6.

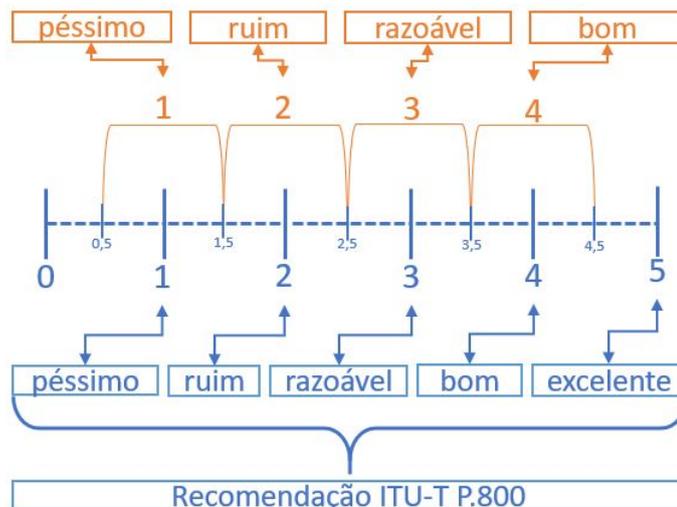
Tabela 3.3 – Intervalos do  $MOS_{Numérico}$  e  $MOS_{Literal}$ 

	Início			Fim		$MOS_{Numérico}$	$MOS_{Literal}$	
$Intervalo_{(1)}$	0,5	<	x	≤	1,5	≈	1	péssimo
$Intervalo_{(2)}$	1,5	<	x	≤	2,5	≈	2	ruim
$Intervalo_{(3)}$	2,5	<	x	≤	3,5	≈	3	razoável
$Intervalo_{(4)}$	3,5	<	x	≤	4,5	≈	4	bom

Fonte: do autor

Cada valor do MOS que adveio do Passo 1 que estava contido nos intervalos demonstrados na Tabela 3.2 recebeu uma escala de qualidade objetiva. No  $Intervalo_{(1)}$ , por exemplo, o  $MOS_{Numérico}$  igual a 1 está contido no intervalo dos valores maiores do que 0,5 e menores ou iguais a 1,5. Todos os valores dos MOS que estiverem nesse intervalo são arredondados para o valor 1, recebendo assim a denominação de péssimo. No  $Intervalo_{(2)}$  valores acima de 1,5 e menores ou iguais a 2,5 são arredondados para 2, recebendo assim a classificação de ruim. O  $Intervalo_{(3)}$  cujos valores estão acima de 2,5 e menores ou iguais a 3,5 são arredondados para 3, recebendo a classificação de razoável. E por último, no  $Intervalo_{(4)}$  valores acima de 3,5 e menores ou iguais a 4,5 são arredondados para 4 e recebem a denominação de bom como pode ser visto na Figura 3.6.

Figura 3.6 – Escala de conversão do MOS.



Fonte: do Autor

A Tabela 3.4 se refere à quantidade de imagens por  $MOS_{Literal}$ , ou seja, cada coluna contém a quantidade para péssima, ruim, razoável e boa.

Tabela 3.4 – Quantidade de imagens por MOS<sub>Literal</sub>.

	QPSK	2-QAM	4-QAM	16-QAM	32-QAM	64-QAM	128-QAM	256-QAM	TOTAL
péssimo	9	4	9	15	21	21	27	27	133
ruim	1	1	1	1	1	1	1	0	7
razoável	0	1	0	1	1	1	1	1	6
bom	24	28	24	17	11	11	5	6	126
Total	34	34	34	34	34	34	34	34	272

Fonte: do autor

### 3.3 Extração de Características de uma Imagem

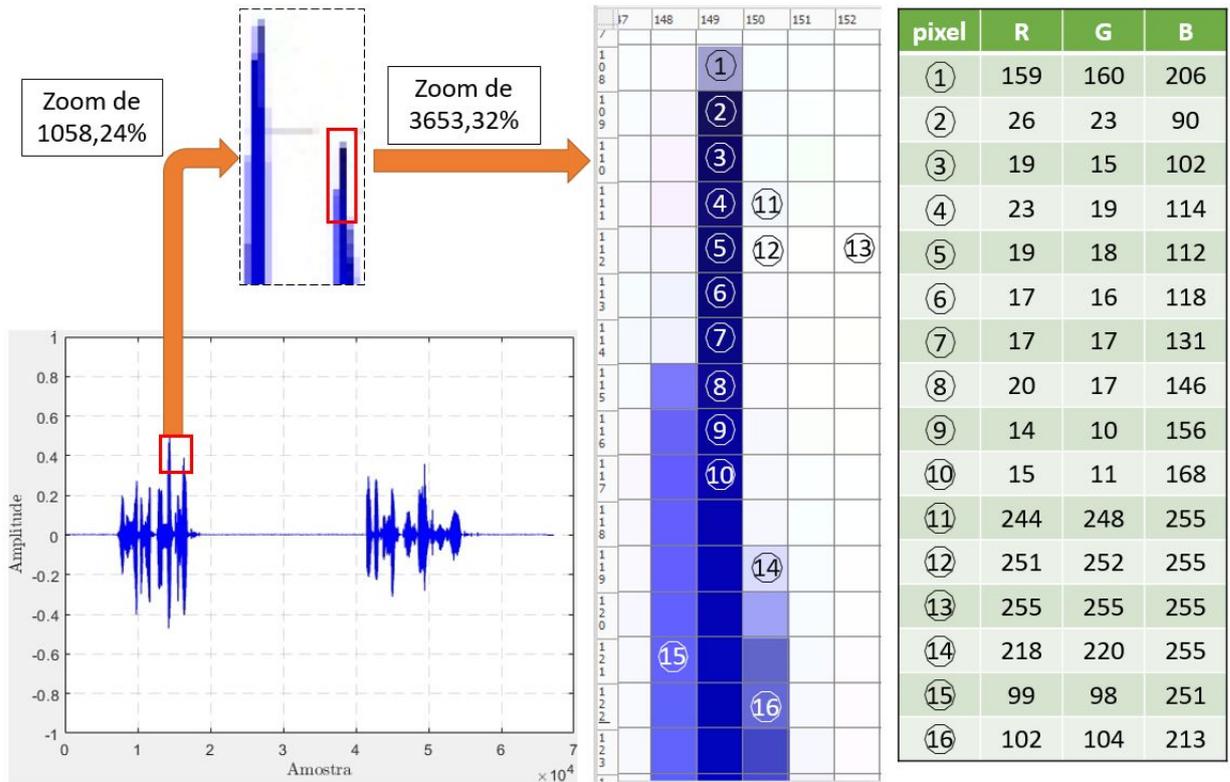
A Figura 3.7 [a] demonstra as diferentes cores que as imagens dos *plots* têm, com cada pixel contendo uma cor diferente, como demonstrado pelo seu RGB, onde R é a letra inicial da palavra inglesa *Red* (Vermelho), G para o *Green* (Verde) e B para *Blue* (Azul). A combinação das três gera a cor que é demonstrada em cada pixel marcado. O *plot* que aparece na Figura é a imagem `or105_deg_2-qam_SNR_p12` cuja modulação é 2-QAM, nível de quantização de  $2^{16}$  e SNR de 12dB produzindo um MOS de 4,5 com uma representação literal de áudio bom.

A escala para cor varia de 0 a 255, para cada componente (R, G, B), totalizando 256 que permutados dentro de cada variável do RGB produz uma quantidade grande de cores. Para se trabalhar com imagens é mais aconselhado que se trabalhe em uma escala de cinza contendo 256 tons de cinza, para fazer essa transformação usou-se a Equação 3.1.

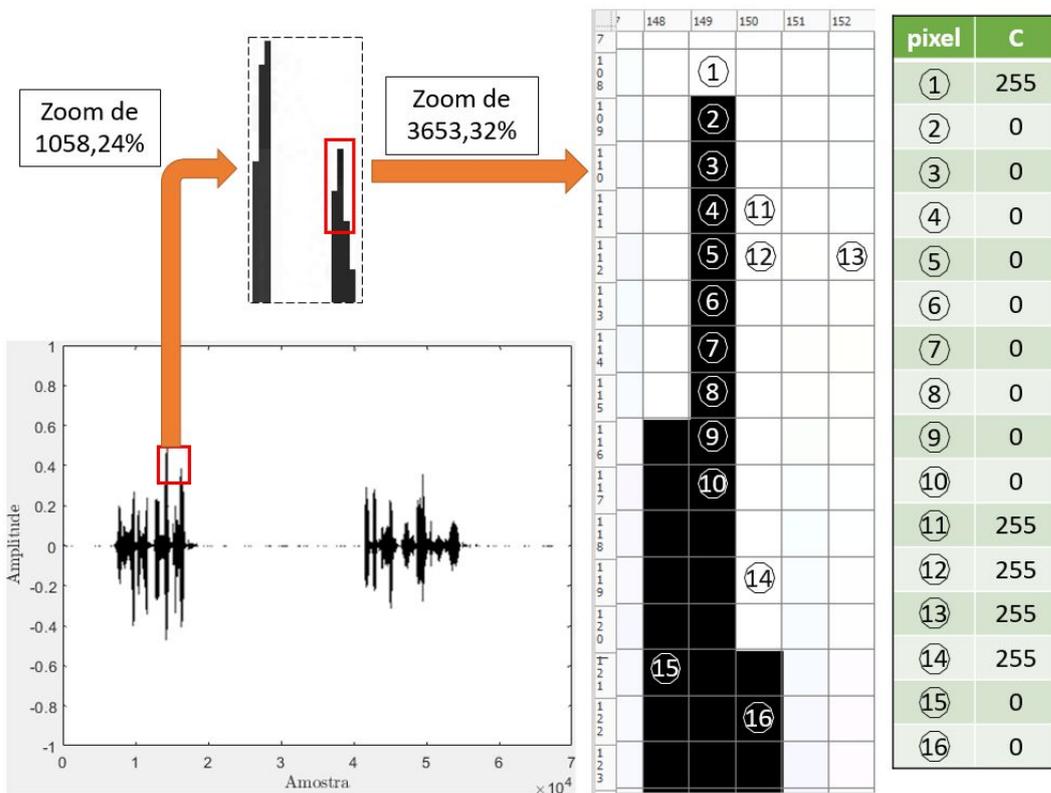
$$C = 0,2989 * R + 0,5870 * G + 0,1140 * B \quad (3.1)$$

Na Equação 3.1 cada valor que é atribuído às variáveis R, G e B obtém-se um valor para Cinza (C), que por sua vez, se esse valor for maior do que 128, ou seja, a metade de 256, ele é setado como pixel-branco e caso contrário pixel-preto como demonstrado na Figura 3.7 [b].

Figura 3.7 – Classificação dos pixels das imagens que representam o sinal de áudio.



[a] Pixels com tonalidade de azul



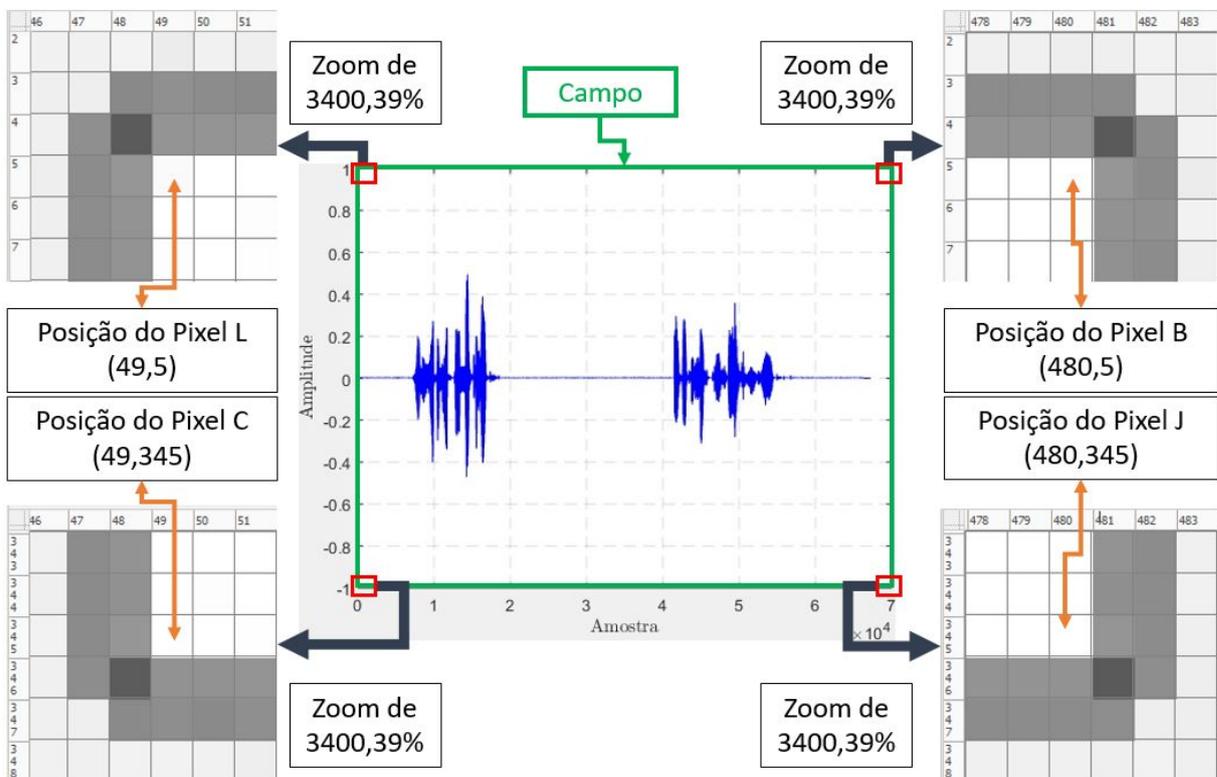
[b] Pixels preto e branco

Fonte: do Autor

### 3.3.1 Escaneando Imagens

Uma imagem que foi salva em um arquivo jpg nada mais é do que uma matriz com as dimensões ( $n \times p$ ), em que  $n$  representa o número de linhas e  $p$  o número de colunas, as imagens plotadas neste trabalho contém 485 pixels de largura por 390 pixels de altura. O processo de escaneamento de imagens acontece a partir da posição (49,5) que corresponde ao Pixel L que pertence à coluna  $p_{(49)}$  como demonstrado na Figura 3.8. O escaneamento acontece no Pixel L pois é a partir dele que começa a área onde todos os sinais do domínio do tempo são plotados, essa área delimitada pelos pixels L, C, B e J foi denominada de campo. Dado o início do processo de escaneamento, cada pixel da coluna  $p_{(49)}$  é verificado até atingir a posição do Pixel C cuja coordenada é (49,345). Esse processo nada mais é do que fixar a coluna, começar o escaneamento na linha 5 e percorrer as outras linhas até atingir a linha número 345, quando isso acontece há uma troca de coluna passando agora para a coluna  $p_{(50)}$ , repetindo assim, o processo de escaneamento, linha por linha. Isso só termina quando o último pixel é atingido na posição (480,345).

Figura 3.8 – Processo de escaneamento de imagens.

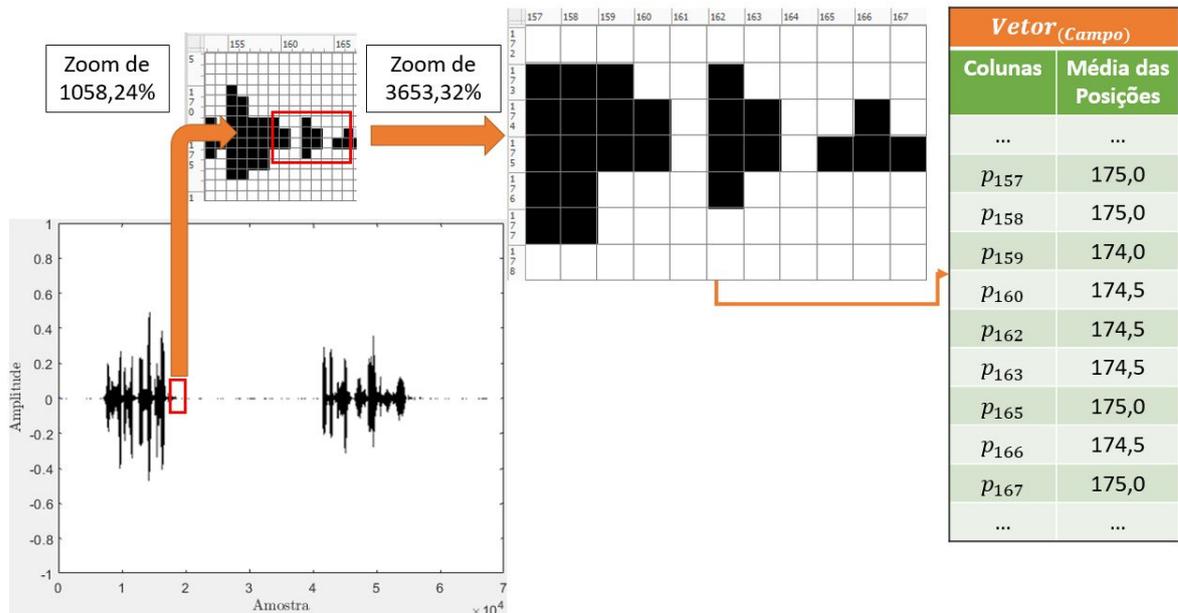


Fonte: do Autor

Para se fazer o escaneamento das 272 imagens degradadas advindas do arquivo original or105 foi implementado em Java o algoritmo P2F1 (Passo 2 Ferramenta 1) que carrega uma

imagem por vez, o algoritmo percorre cada coluna e usando a Equação 3.1 aplica o *threshold*. Para facilitar o entendimento do algoritmo a Figura 3.9 mostra o preenchimento do  $\text{Vetor}_{(Campo)}$ .

Figura 3.9 – Processo de escaneamento e composição do  $\text{Vetor}_{(Campo)}$



Fonte: do Autor

Quando o ponto (157,345) é atingido o código simplesmente já armazenou a posição de cada pixel-preto encontrado, referente à coluna  $p_{(157)}$ , antes de passar para a próxima,  $p_{(158)}$ , ele faz a média das posições armazenando no  $\text{Vetor}_{(Campo)}$  o valor 175,0, ou seja, na referida imagem a coluna  $p_{(157)}$  só tem 5 pixel-pretos, então esse estágio realiza o cálculo como demonstrado na Equação 3.2 e armazena o resultando 175,0 no  $\text{Vetor}_{(Campo)}$ . Esse processo continua até que o Pixel J localizado na posição (480,345) possa ser alcançado, caso não exista pixel-preto como ocorrem nas colunas  $p_{(161)}$  e  $p_{(164)}$ , simplesmente não há armazenamento, ou seja, não existe sinal sonoro naquelas colunas. Isso faz com que o tamanho do  $\text{Vetor}_{(Campo)}$  seja dinâmico pois é dependente da figura. Quando esse processo termina tem-se um vetor dinâmico podendo chegar ao tamanho máximo de 431 valores, ou seja, 480 - 49, referentes às médias das posições em cada coluna caso exista pixel-preto.

$$\text{Média}_{p(157)} = \frac{173 + 174 + 175 + 176 + 177}{5} = 175,0 \quad (3.2)$$

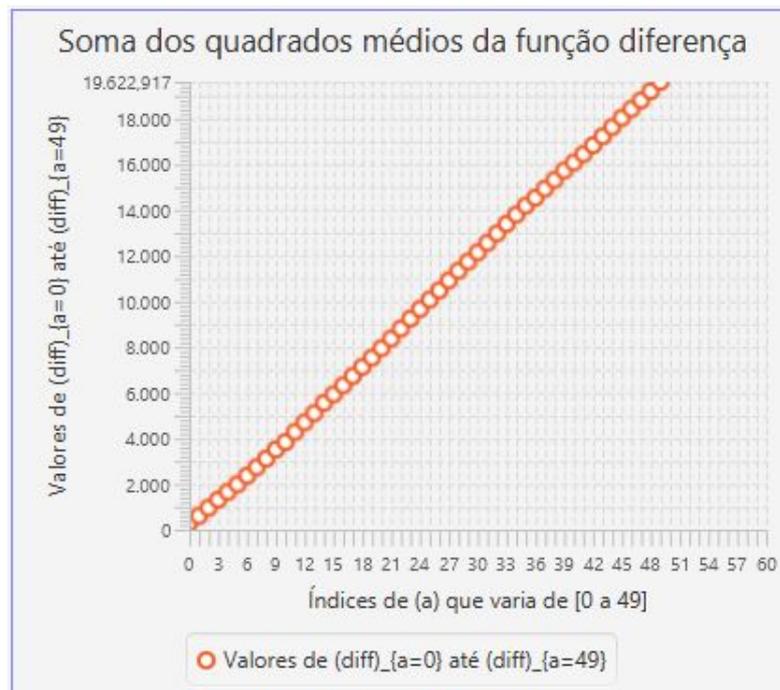
No próximo estágio, ainda dentro desse código, é usar a função diferença (*diff*), muito comum em *softwares* matemáticos, porém no Java não teve essa facilidade, foi preciso então, criar uma fórmula matemática específica para esse tipo de problema e que pode ser verificada na

Equação 3.3, sendo similar à função `diff` do *Software R*, medida essa necessária para visualizar o comportamento da função referente a cada imagem carregada e fazer os processos de regressão e classificação no *software Weka*.

$$(diff)_{a=0}^{49} = \left( \frac{\sum_{m=1}^{Tamanho\_do[Vetor(Campo)]-50} (PosVetor(m+a) - PosVetor(m-1))^2}{Tamanho\_do[Vetor(Campo)]} \right) \quad (3.3)$$

Os valores de  $(diff)_{a=0}$ ,  $(diff)_{a=1}, \dots, (diff)_{a=49}$  da Equação 3.3 representam pontos na Figura 3.10 que juntos formam uma função discreta, que por sua vez, representa a imagem carregada e escaneada. São esses pontos que serão usados no processo de treinamento nos algoritmos juntamente com o valor do MOS.

Figura 3.10 – Gráfico contendo 50 pontos da função diferença.



Fonte: do Autor

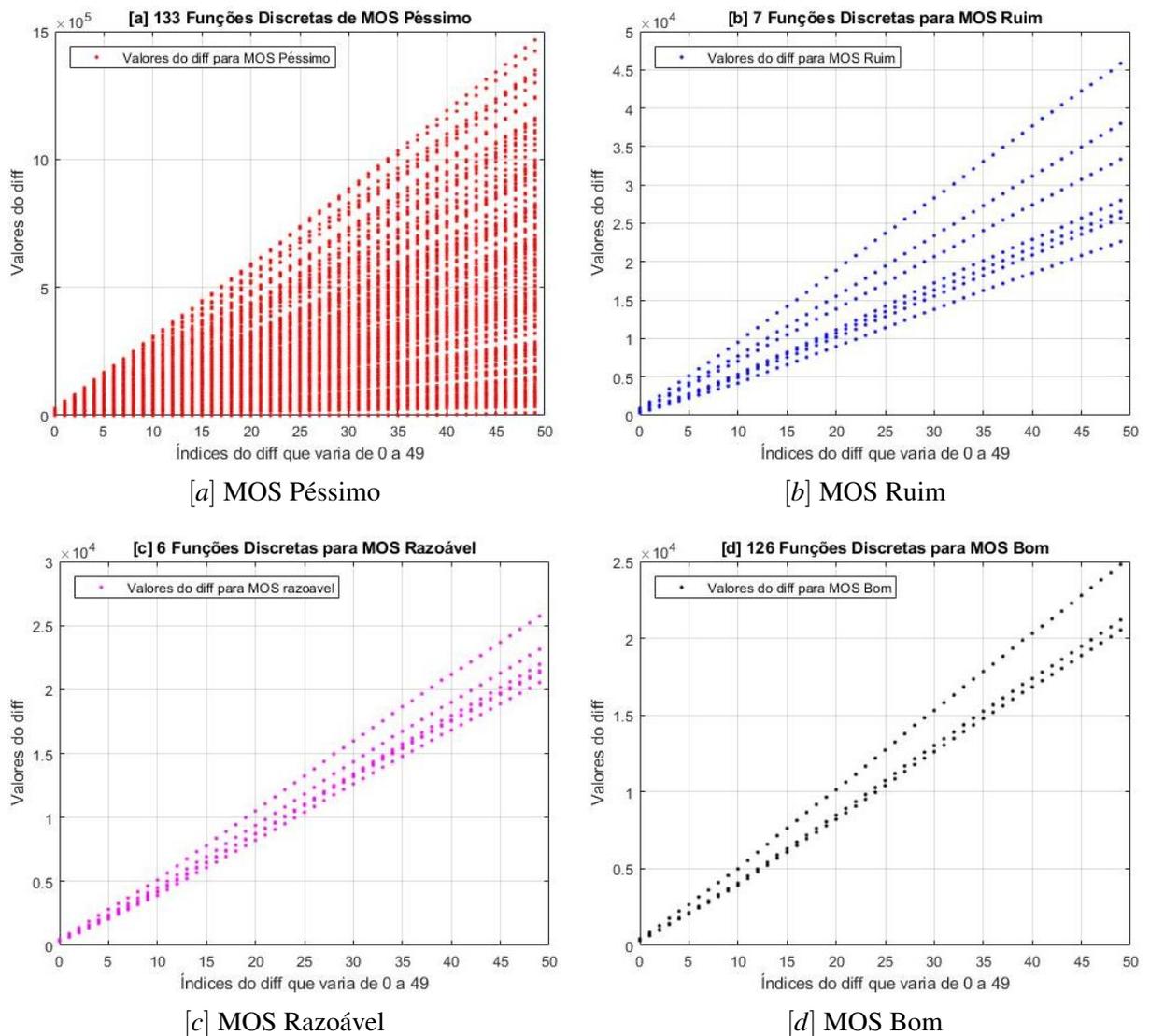
Usando o  $Vetor(Campo)$  e por meio da Equação 3.3 formou-se um outro vetor contendo apenas 50 valores, denominado de  $Vetor50$ , uma vez que, 50 números já sejam suficientes para verificar o comportamento da curva que melhor representa aquela imagem carregada, poderia ser mais ou menos, optou-se aqui, por 50. Onde  $a$  é uma variável que recebe um valor por vez, que começa em 0 e vai até 49,  $m$  é outra variável que começa em 1 e vai até  $Tamanho\_do[Vetor(Campo)] - 50$ , uma vez que, o vetor campo é dinâmico e depende das colu-

nas que existam pixel-pretos para se calcular a média, caso não exista pixel-preto a coluna não é contabilizada, passando para a seguinte.  $PosVetor$  é a posição do vetor referente à parcela  $(m + a)$ , que caracteriza o valor atual do vetor naquela posição e  $(m - 1)$  o valor anterior. Nessa equação deve se atentar para a parcela  $(m + a)$ , cujo valor não pode exceder 431, ou seja, a largura do campo, quando  $a$  for igual a 49 tem que garantir que  $m$  tenha um valor máximo de  $Tamanho\_do[Vetor_{(Campo)}] - 50$ , ou seja, 50 valores a menos, que são os números que vão de 0 a 49. A fórmula matemática consiste em subtrair o valor atual do vetor pelo anterior, elevando essa diferença ao quadrado, somando com as outras diferenças já calculadas, caso existam e depois tirar a média aritmética, repetindo o procedimento no passo seguinte para  $(diff)_{a=1}$ , assim sucessivamente até  $a=49$ , totalizando 50 valores referentes às médias das somas quadráticas. Então, desse modo, a Equação 3.4 representa os valores crescentes de uma curva que pode ser uma reta ou algo bem próximo representando assim a imagem carregada do áudio.

$$Vetor50 = \left( (diff)_{a=0} \quad (diff)_{a=1} \quad (diff)_{a=2} \quad \cdots \quad (diff)_{a=48} \quad (diff)_{a=49} \right) \quad (3.4)$$

Ao final, o código salva um único arquivo com a extensão padrão do Weka arff contendo 272 instâncias que correspondem às 272 imagens com 51 atributos cada, ou seja, 50 valores vindo do Vetor50 e 1 atributo vindo da classe MosNumeric que é o valor do respectivo MOS de cada imagem que está representada na Tabela 3.2, denominado de or105\_qt272\_r2. Onde or105 é o nome do arquivo original que foi degradado, qt272 é a quantidade de instâncias que corresponde à quantidade de imagens escaneadas e r2 representa que esse arquivo foi o segundo a ser analisado pelos algoritmos de regressão. Outro arquivo no mesmo molde também é salvo só que com a classe MosAlpha que é o  $MOS_{(literal)}$  devido à Tabela 3.3, denominado de or105\_qt272\_c1 onde c1 é referente à classificação, ambos os arquivos salvos foram usados nos algoritmos de treinamento. A Figura 3.11 [a] mostra 133 Funções Discretas (FD) para o MOS péssimo, a Figura [b] mostra 7 FD para MOS Ruim, a Figura [c] mostra 6 com MOS Razoável e a Figura [d] mostra 126 FD para MOS Ruim totalizando 272 inclinações referentes as imagens.

Figura 3.11 – Funções Discretas.

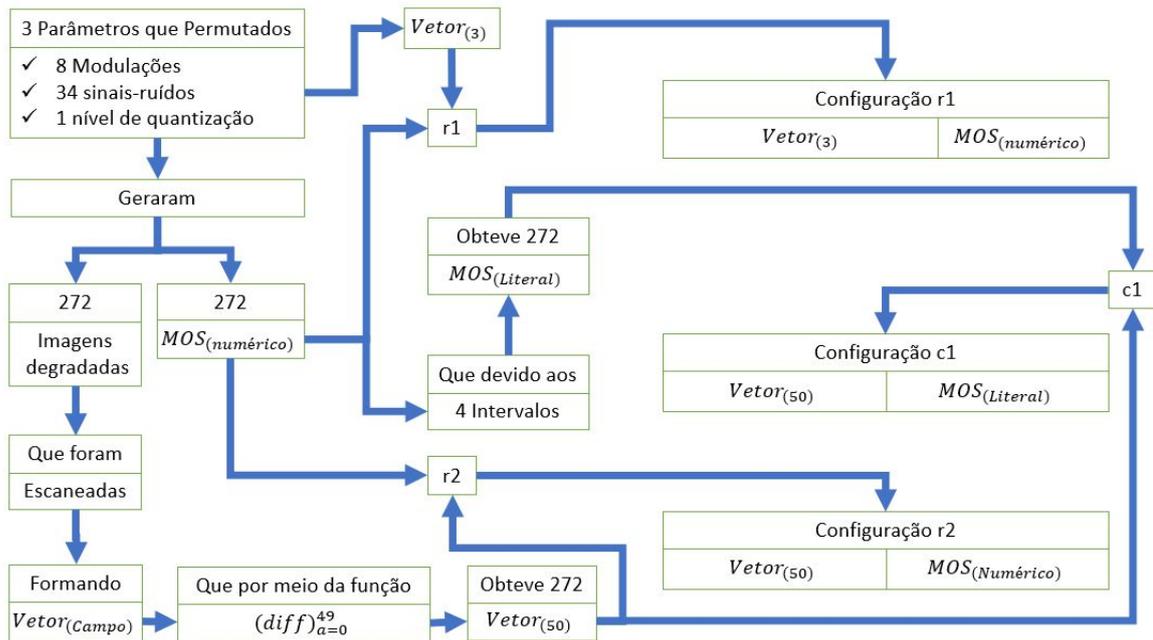


Fonte: do Autor

### 3.3.2 Execução dos algoritmos de treinamento do Weka

Construiu-se mais um arquivo de regressão do *software* Weka contendo 272 instâncias com somente 4 atributos, ou seja, os 3 parâmetros iniciais sendo eles o SNR que variou de 0 a 33, a modulação QPSK e QAM; e o nível de quantização que foi fixado com o valor de  $2^{16} \rightarrow 65536$ , os três parâmetros geraram o  $MOS_{(Numérico)}$ . Esse arquivo de regressão foi denominado de *or105\_qt272\_r1*, onde *r1* significa que é o primeiro arquivo de regressão que foi analisado. A Figura 3.12 mostra o esquema na construção das configurações para *r1*, *r2* e *c1* que foram usados no *software* Weka.

Figura 3.12 – Configuração r1, r2 e c1.

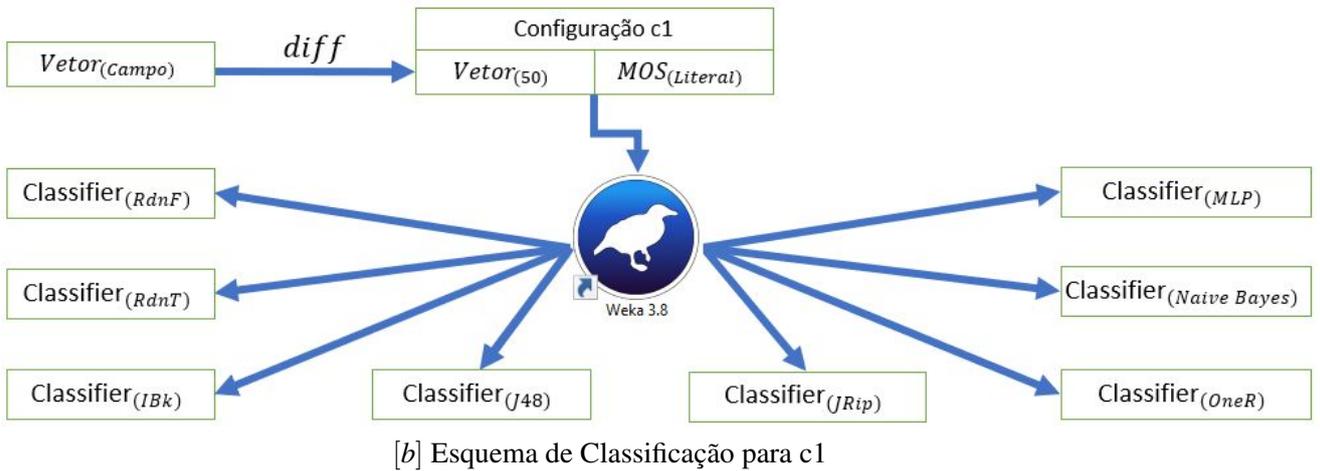
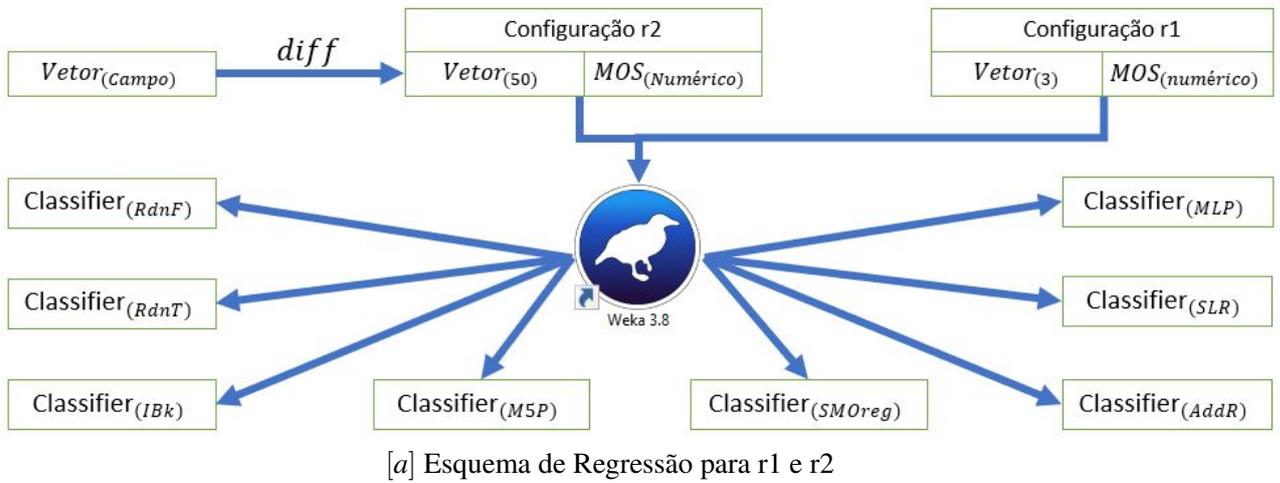


Fonte: do Autor

De posse desses dois arquivos numéricos r1 e r2, implementou-se o código denominado de P2F2 (Passo 2 Ferramenta 2) que sobre um  $K$ -fold com k igual a 10 executou 50 vezes, equivalendo-se a executar o algoritmo de aprendizado em um total de 500 validações ou 5000 iterações para cada um dos oito algoritmos de regressão usados, sendo eles o MLP-*Multilayer Perceptron*, SLR-*Simple Linear Regression*, SMOreg, AddR-*Additive Regression*, MSP, IBk, RdnF-*RandomForest* e RdnT-*RandomTree* como pode ser visto na Figura 3.13 [a]. Tal medida teve o intuito de não só tornar o processo mais otimizado, como também assertivo.

Para o arquivo c1 um outro código foi implementado no molde de P2F2, denominado de P2F3 (Passo 2 Ferramenta 3) que consiste em executar cada um dos algoritmos de classificação também 50 vezes sobre um  $K$ -fold com k igual a 10, sendo eles MLP-*Multilayer Perceptron*, J48, Naive Bayes, OneR, JRip, IBk, RdnF-*Random Forest* e RdnT-*RandomTree* como pode ser visualizado na Figura 3.13 [b].

Figura 3.13 – Esquema usado nos algoritmos de aprendizado de regressão e classificação.

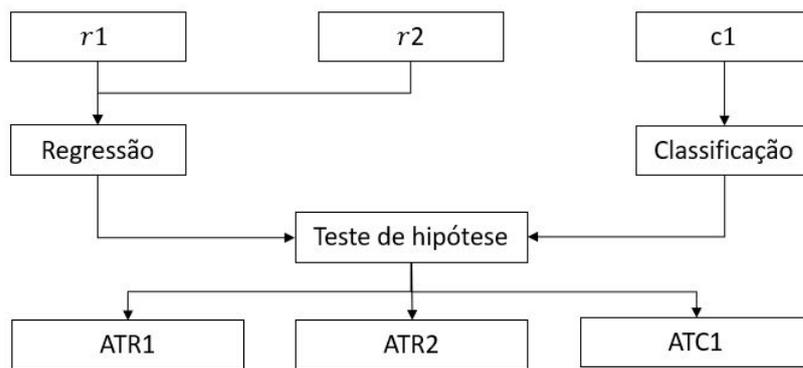


Fonte: do Autor

## 4 RESULTADOS E DISCUSSÕES

Nesse capítulo será discutido os resultados encontrados tanto para regressão quanto para a classificação. Os algoritmos treinados foram construídos usando as imagens degradadas do arquivo original or105 que totalizou 272 imagens de 1632, o restante referente as demais representações dos arquivos degradados foram usadas para validar o modelo, no total de 1360 imagens. Afim de verificar qual *classifier* era estatisticamente superior em relação aos demais, sendo esse passo necessário antes de construir o(s) modelo(s), foi feito o teste de hipótese usando o *package TStools* instalado no *Software RStudio* com nível de confiança de 95% para todos os testes. A Figura 4.1 mostra como os 3 arquivos r1, r2 e c1 foram agrupados, para os numéricos usou-se *classifiers* de regressão e para c1, classificadores.

Figura 4.1 – Esquema da formulação dos algoritmos treinados, ATR1 (Regressão 1), ATR2 (Regressão 2) e ATC1 (Classificação 1).



Fonte: do Autor

### 4.1 Resultados da Regressão

Nesta seção estão contidos os resultados dos algoritmos treinados ATR1 e ATR2.

#### 4.1.1 Resultado da Regressão para ATR1

Os dados usados em ATR1 são os mesmos utilizados no cenário geral para degradação dos arquivos originais e se referem ao nível de quantização fixado em  $2^{16}$ , as oito modulações sendo elas: QPSK, 2-QAM, 4-QAM, 16-QAM, 32-QAM, 64-QAM, 128-QAM e 256-QAM, além do SNR variando de 0 a 33, totalizando 272 instâncias que correspondem as características extraídas de cada imagem com seus respectivos valores de MOS para o áudio original or105. Cada *classifier* repetiu-se 50 vezes sobre um *K-fold* do *Cross-Validation* igual a 10, totalizando 500 validações ou 5000 iterações, sendo que 4500 foram usadas para treinamentos e 500 para

testes, nas 272 instâncias, a Tabela 4.1 mostra os resultados das médias dos coeficientes de correlação, onde o algoritmo que teve uma média superior foi o IBk com 0,9823. No entanto, como há outro *classifier* com valor próximo a este, foi preciso saber se esse *classifier* é estatisticamente superior aos demais, para isso foi necessário fazer o *Ranking* das médias como pode ser verificado na Tabela 4.2. Esse *Ranking* nada mais é do que fazer o levantamento de quantas vezes determinado algoritmo de treinamento teve melhor resultado, em relação aos demais, referente à mesma repetição e ao final fazer a média de todas as repetições e realizar o arredondamento.

Tabela 4.1 – Médias dos coeficientes de correlação para ATR1.

	IBk	RdnT	RdnF	M5P	SMOreg	MLP	AddR	SLR
Média	0,9823	0,9759	0,9242	0,9046	0,7774	0,7623	0,7706	0,6795

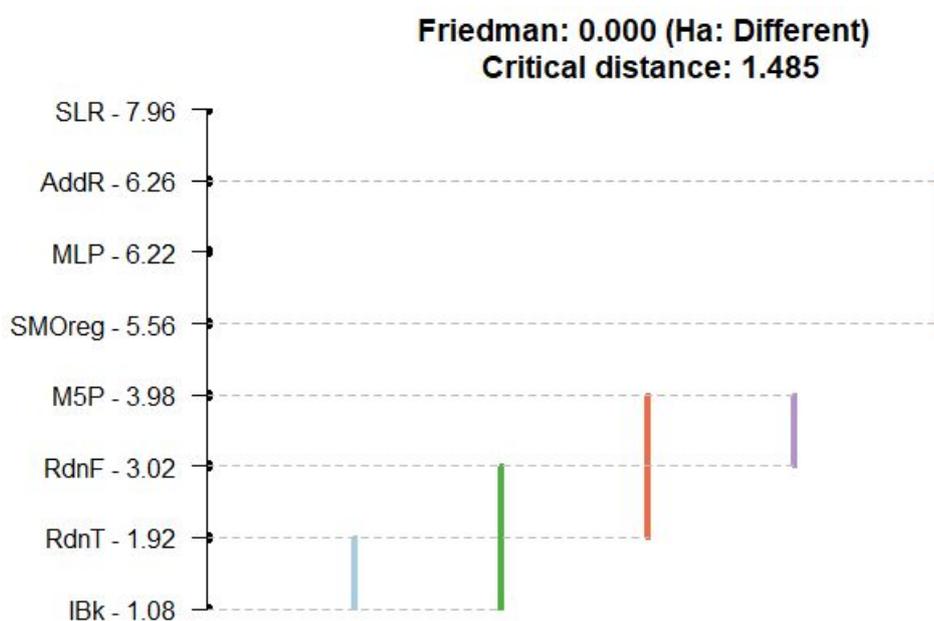
Fonte: do autor

Tabela 4.2 – *Ranking* para ATR1.

	IBk	RdnT	RdnF	M5P	SMOreg	MLP	AddR	SLR
Média	1,08	1,92	3,02	3,98	5,56	6,22	6,26	7,96
<i>Ranking</i>	1 <sup>o</sup>	2 <sup>o</sup>	3 <sup>o</sup>	4 <sup>o</sup>	5 <sup>o</sup>	6 <sup>o</sup>	7 <sup>o</sup>	8 <sup>o</sup>

Fonte: do autor

Figura 4.2 – Teste de hipótese para ATR1



Fonte: do Autor

A Figura 4.2 mostra o resultado do teste de hipótese e traz a informação de que quanto menor o número no eixo vertical, que nada mais é do que a Tabela 4.2 em ordem crescente, melhor ranqueado o *classifier* está, neste caso o algoritmo RdnF ficou em primeiro no *ranking* sendo o pior algoritmo o SLR. A Hipótese alternativa ( $H_a$ ) mostra que todos os *classifiers* têm *ranking* distintos e que a diferença estatística significativa que é a distância crítica (CD) entre um e outro é de 1,485. Isso significa dizer que, se a diferença entre o *Ranking* médio de um *classifier* em relação a outro for maior do que a distância crítica, eles são estatisticamente diferentes. Caso contrário, se for menor ou igual, não há como provar que exista diferença estatística significativa entre eles. Em outras palavras, tanto faz usar um ou outro(s) para a construção do(s) modelo(s). No teste pode-se ver dois *classifiers* se destacando sendo eles IBk e *RandomTree* onde não tem como provar que exista alguma diferença estatística significativa entre eles. Esse resultado demonstra que o modelo pode ser construído usando apenas um deles ou ambos. Por exemplo, do *RandomTree* para o IBk a distância é de  $d = 1,92 - 1,08 \rightarrow d = 0,84$  essa distância é menor que a distância crítica, ou seja, não se pode afirmar que há diferença estatística significativa em relação aos dois *classifiers* e que um tenha ficado a frente do outro por mero acaso. A segunda linha vertical mostra os três melhores algoritmos ranqueados, em relação aos outros *classifiers* que estão fora dessa segunda linha vertical, são menos precisos, pois, quanto mais distante eles estiverem, pior serão.

Tabela 4.3 – Sumário dos Algoritmos Treinados para or105\_r1.

	IBk	RdnT	RdnF	M5P	SMOreg	MLP	AddR	SLR
CC	1	0,9999	0,9979	0,9461	0,7819	0,9007	0,8248	0,6838
EAM	0	0,0165	0,0561	0,3726	0,8224	0,4384	0,8170	0,9441
EQM	0	0,0287	0,1129	0,5439	1,0368	0,7227	0,9415	1,2112
EAR	0%	1,0060%	3,4280%	22,7501%	50,2185%	26,7675%	49,8883%	57,6484%
EQR	0%	1,7289%	6,7993%	32,7667%	62,4577%	43,5327%	56,7153%	72,9641%
NTI	272	272	272	272	272	272	272	272

Fonte: do autor

A Tabela 4.3 mostra os sumários de todos os algoritmos treinados usando a totalidade da base de dados or105 para treinamento e teste, ou seja, usou-se os 10 grupos para fazer o treinamento e o teste. O motivo é simples, o modelo criado tem que ser o mais fiel possível à base de dados e não usar apenas 9 décimos da base de dados para construir o modelo como é feito no *Cross-validation*, por isso que os coeficientes de correlação são maiores. Nessa tabela a sigla CC é o Coeficiente de Correlação, EAM é o Erro Absoluto Médio que é a diferença

dos valores da Tabela 3.2 e dos valores previstos. EQM é o Erro Quadrático Médio que é muito útil na comparação dos algoritmos, uma vez que, mostra que o algoritmo mais eficaz é simplesmente aquele com a menor variância. EAR é o Erro Absoluto Relativo que é a razão entre o erro absoluto e o valor previsto de um número. EQR é o Erro Quadrático Relativo e NTI é o Número Total de Instâncias. Pela tabela 4.3 nota-se que o algoritmo IBk acerta todas as instâncias com um coeficiente de correlação igual a 1 e todos os erros iguais a zero na comparação com os outros algoritmos, isso demonstra que ele é o mais eficaz, por ter a menor variância. Mesmo o teste de Friedman-Nemenyi trazendo a informação de que não há diferenças entre um e outro, o algoritmo IBk saía melhor do que o RdnT, o segundo melhor ranqueado que quase acerta também a totalidade das instancias. O RdnF tem o EAM cerca de 3,5 vezes maior e o EQM cerca de 4 vezes maior do que o RdnT. A partir do M5P já não são tão eficazes, nota-se pelo EQM cada vez maior. Nota-se também que os coeficientes de correlação dos três melhores ranqueados estão bem próximos.

A Tabela 4.4 contém os resultados da validação do modelo pelos algoritmos treinados usando a base de dados or109, cuja base em nenhum momento foi usada para fazer qualquer tipo de treinamento dos modelos. Percebe-se que o IBk, RdnT e RdnF obtiveram os maiores coeficiente de correlação, o IBk tem um EAM cerca de 0,45% e um EQM cerca de 2,35% maiores do que o RdnT. O EAM do RdnF é cerca de 1,4 vezes e um EQAM cerca de 1,36 vezes maiores do que o IBk.

Tabela 4.4 – Validação do modelo para os arquivos degradados or109\_r1

	IBk	RdnT	RdnF	M5P	SMOreg	MLP	AddR	SLR
CC	0,9988	0,9989	0,9981	0,9474	0,7837	0,9035	0,8247	0,6824
EAM	0,0669	0,0666	0,0932	0,3982	0,8559	0,4798	0,8458	0,9807
EQM	0,1063	0,1038	0,1445	0,5632	1,0597	0,7401	0,9668	1,2484
NTI	272	272	272	272	272	272	272	272

Fonte: do autor

A Tabela 4.5 contém os resultados da validação do modelo pelos algoritmos treinados usando a base de dados or114, cuja base em nenhum momento foi usada para fazer qualquer tipo de treinamento dos modelos. Percebe-se que o IBk, RdnT e o RdnF obtiveram os maiores coeficientes de correlação se comparado com os demais, porém o IBk teve um menor EAM e EQM cerca de 0,19% maior do que o RdnT.

Tabela 4.5 – Validação do modelo para os arquivos degradados or114\_r1

	IBk	RdnT	RdnF	M5P	SMOreg	MLP	AddR	SLR
CC	0,9969	0,9969	0,9979	0,9529	0,7956	0,9069	0,8346	0,6934
EAM	0,1887	0,1896	0,2110	0,4739	0,9244	0,6018	0,8772	1,0831
EQM	0,2709	0,2704	0,2879	0,6303	1,1190	0,8172	1,0230	1,3333
NTI	272	272	272	272	272	272	272	272

Fonte: do autor

A Tabela 4.6 contém os resultados da validação do modelo pelos algoritmos treinados usando a base de dados or129, cuja base em nenhum momento foi usada para fazer qualquer tipo de treinamento dos modelos. Percebe-se que o IBk e RdnT obtiveram os maiores CC, também os menores EQM e EAM na comparação com os demais. O EAM do IBk é cerca de 0,44% menor do que o RdnT, porém o seu EQM é cerca de 0,34% maior do o RdnT.

Tabela 4.6 – Validação do modelo para os arquivos degradados or129\_r1

	IBk	RdnT	RdnF	M5P	SMOreg	MLP	AddR	SLR
CC	0,9961	0,9961	0,9917	0,9312	0,7622	0,8864	0,8056	0,6603
EAM	0,1814	0,1822	0,2042	0,4126	0,7918	0,4633	0,7796	0,8840
EQM	0,2645	0,2636	0,2797	0,5780	1,0136	0,7203	0,9340	1,1490
NTI	272	272	272	272	272	272	272	272

Fonte: do autor

A Tabela 4.7 contém os resultados da validação do modelo pelos algoritmos treinados usando a base de dados or134, cuja base em nenhum momento foi usada para fazer qualquer tipo de treinamento dos modelos. Percebe-se que o RdnT obteve maior CC, seguido por IBk e RdnF, porém o IBk teve menor EAM e um EQM cerca de 0,36% maior do que o RdnT.

Tabela 4.7 – Validação do modelo para os arquivos degradados or134\_r1

	IBk	RdnT	RdnF	M5P	SMOreg	MLP	AddR	SLR
CC	0,9940	0,9941	0,9886	0,9256	0,7563	0,8824	0,7999	0,6566
EAM	0,2085	0,2091	0,2346	0,4363	0,7971	0,4833	0,7877	0,8818
EQM	0,3021	0,3010	0,3205	0,6012	1,0213	0,7315	0,9447	1,1466
NTI	272	272	272	272	272	272	272	272

Fonte: do autor

A Tabela 4.8 contém os resultados da validação do modelo pelos algoritmos treinados usando a base de dados or137, cuja base em nenhum momento foi usada para fazer qualquer

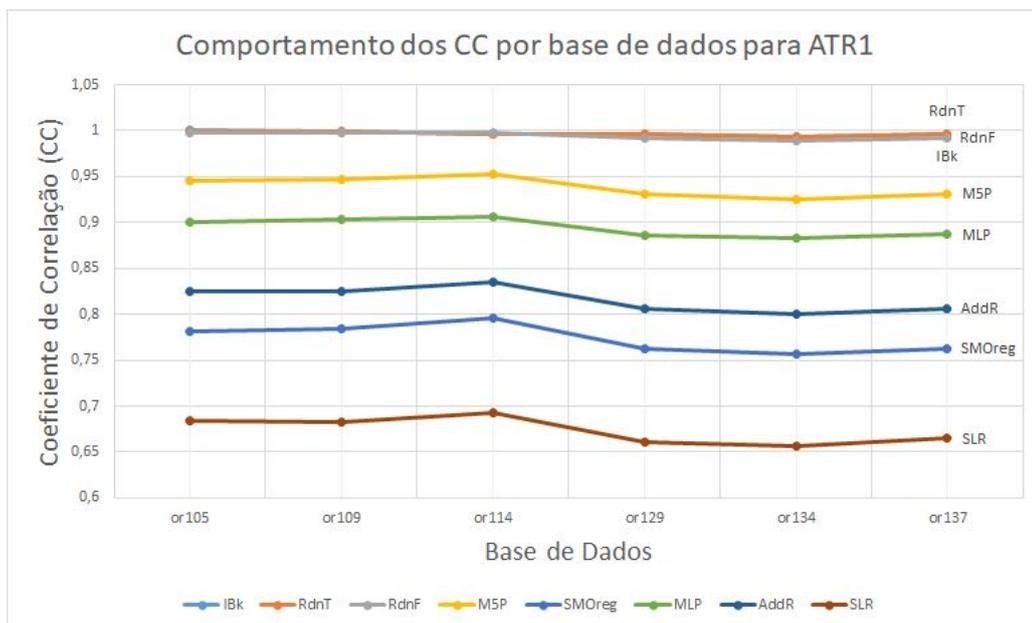
tipo de treinamento dos modelos. Percebe-se que o IBk e o RdnT obtiveram o mesmo valor para CC sendo os maiores na comparação dos demais, o IBk teve um EAM cerca de 0,74% menor do que o RdnT, porém o seu EQM é cerca de 0,34% maior. Nessa validação o algoritmo treinado que teve o melhor resultado foi o RdnT.

Tabela 4.8 – Validação do modelo para os arquivos degradados or137\_r1

	IBk	RdnT	RdnF	M5P	SMOreg	MLP	AddR	SLR
CC	0,9965	0,9965	0,9919	0,9308	0,7628	0,8867	0,8063	0,6646
EAM	0,1614	0,1626	0,1897	0,4074	0,7937	0,4507	0,7871	0,8834
EQM	0,2372	0,2364	0,2605	0,5789	1,0188	0,7223	0,9373	1,1539
NTI	272	272	272	272	272	272	272	272

Fonte: do autor

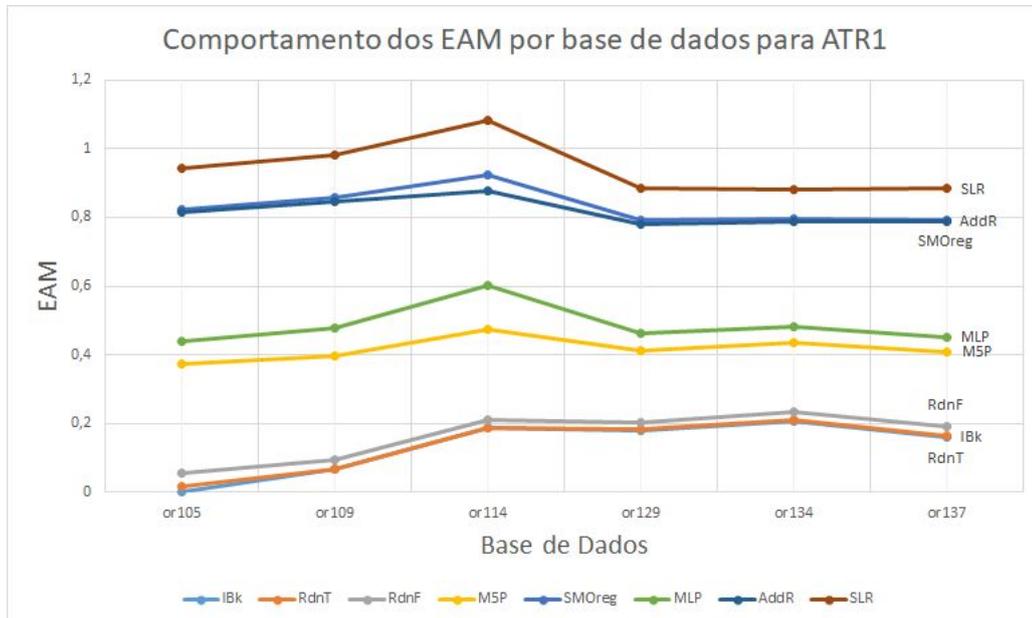
Figura 4.3 – Comportamento dos CC para ATR1



Fonte: do Autor

O gráfico da Figura 4.3 mostra o comportamento dos algoritmos treinados nas diferentes base de dados, onde cinco deles tiveram seus CC acima de 0,88; mas somente três deles ficaram próximos de 1 que foram o IBk, RdnT e o RdnF, sendo eles os melhores algoritmos ranqueados pelo teste de hipótese tendo quase uma sobreposição de valores, indicando assim grande aceitação deles para esse tipo de problema.

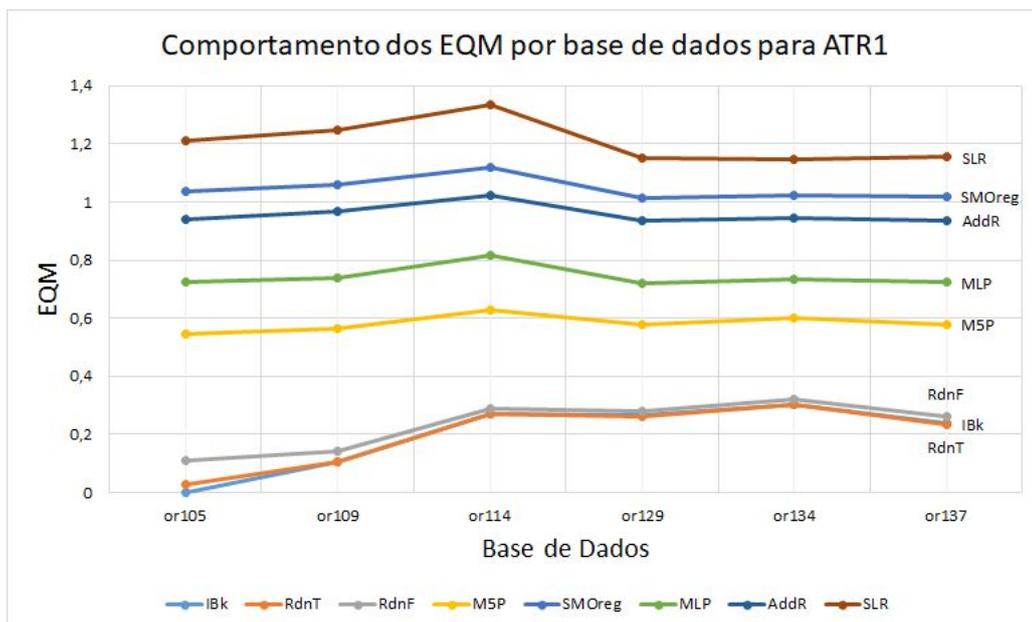
Figura 4.4 – Comportamento dos EAM para ATR1



Fonte: do Autor

A Figura 4.4 mostra o comportamento do erro absoluto médio para cada algoritmo ao longo das diferentes bases de dados e percebe-se que os algoritmos treinados que tiveram menores erros foram IBk, RdnT e o RdnF onde há quase uma sobreposição de valores do IBk e RdnT.

Figura 4.5 – Comportamento dos EQM para ATR1

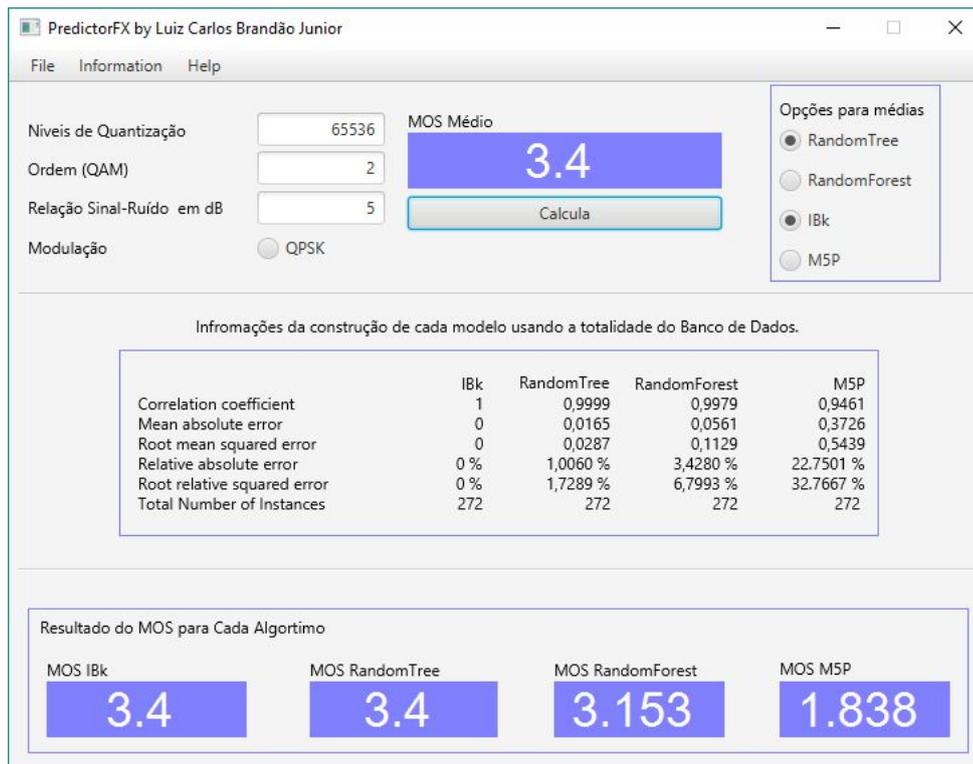


Fonte: do Autor

A Figura 4.5 mostra o comportamento do erro quadrático médio para cada algoritmo ao longo das diferentes bases de dados e percebe-se que os algoritmos que tiveram menores erros foram os IBk, RdnT e RdnF, com quase uma sobreposição de valores para o IBk e o RdnT.

A construção dos algoritmos treinados foi baseada nos quatro primeiros algoritmos com o intuito de mostrar a diferença de valores, sendo eles os *classifiers* IBk, *RandomTree*, *RandomForest* e M5P. O processo para alcançar o ATR1 foi denominado de Modelo KazaR. Para ajudar na predição implementou-se uma calculadora contida no software denominado de PredictorFX, que tem a função de prever os valores do MOS mediante a entrada de três parâmetros, sendo eles nível de quantização que foi fixado em  $2^{16} \rightarrow 65536$ , tipo de modulação e relação sinal-ruído. Sendo esses os mesmos três parâmetros usados no Passo 1, para calcular o MOS pelo código do cenário geral.

Figura 4.6 – Software PredictorFX para ATR1



Fonte: do Autor

A Figura 4.6 se refere a calculadora implementada e demonstra que mediante a entrada do nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação de 2-QAM e relação sinal-ruído igual a 5dB que foram as entradas que geraram um MOS de 3,4 para o desvanecimento do arquivo original or105 no Passo 1, que é a degradação  $p_{(05)}$  da Tabela 3.2. Os quatro modelos geraram cada um, seu respectivo MOS e percebe-se que o resultado do modelo do algoritmo IBk e RdnT

forneceram o mesmo valor em  $p_{(05)}$  da Tabela 3.2, como o teste de Friedman e Nemenyi mencionou que não há diferença estatística significativa em relação aos dois melhores *classifiers*. O RdnF com o valor do MOS de 3,153 foi um pouco pior do que os melhores ranqueados, porém bem melhor do que o MOS do M5P com o valor de 1,833. Na imagem da calculadora contém o MOS médio que corresponde à média de acordo com a escolha no menu Opções para médias.

#### 4.1.2 Resultado da Regressão para ATR2

Os dados contidos em ATR2 são os valores crescentes da Equação 3.4, ou seja, são os valores de 272 funções discretas que conjuntamente com seus respectivos MOS advindos da base de dados or105 foram usados para treinar os 8 algoritmos de regressão. Todo o processo para se construir os algoritmos treinados ATR2 foi denominado de Modelo RazaR. A Tabela 4.9 contém as médias dos coeficientes de correlação correspondente as 50 validações com o algoritmo IBk com maior coeficiente de correlação, seguido por RdnT, RdnF e AddR.

Tabela 4.9 – Médias dos coeficientes de correlação para ATR2.

	IBk	RdnT	RdnF	AddR	M5P	MLP	SMOreg	SLR
Média	0,9918	0,9897	0,9889	0,9812	0,9311	0,7175	0,6632	0,6623

Fonte: do autor

Pela Tabela 4.10 percebe-se que o *classifier* IBk desponta em primeiro como o algoritmo que mais vezes teve o maior coeficiente de correlação por validação, seguido do RdnT, RdnF e por fim o AddR.

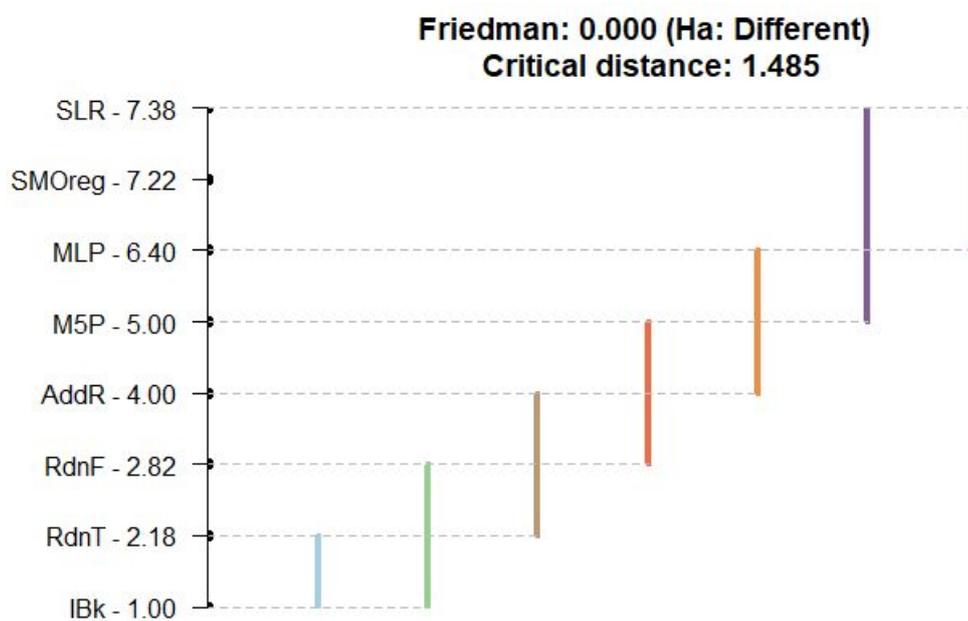
Tabela 4.10 – Ranqueamento para ATR2.

	IBk	RdnT	RdnF	AddR	M5P	MLP	SMOreg	SLR
Média	1,00	2,18	2,82	4,00	5,00	6,40	7,22	7,38
Ranking	1º	2º	3º	4º	5º	6º	7º	8º

Fonte: do autor

Pelo teste de hipótese demonstrado na Figura 4.7, a Hipótese alternativa ( $H_a$ ) diz que os *Ranking* médios são distintos entre si com uma distância crítica de 1,485; com apenas três *classifiers* melhores classificados e dois em que não se pode afirmar que exista alguma diferença estatística significativa entre eles, com o IBk melhor ranqueado.

Figura 4.7 – Teste de Friedman e Nemenyi para ATR2



Fonte: do Autor

A Tabela 4.11 mostra os sumários de todos os algoritmos treinados usando a totalidade da base de dados or105 para treinamento e teste.

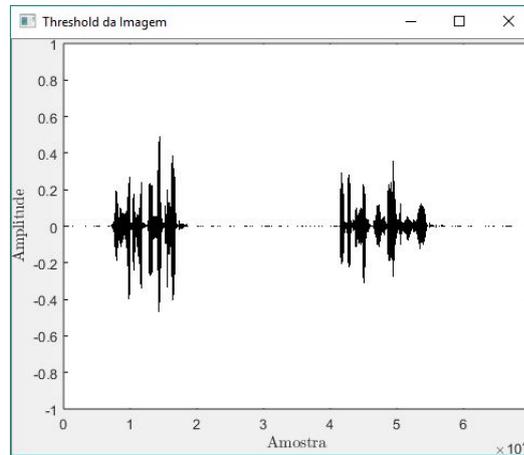
Tabela 4.11 – Sumário dos Algoritmos Treinados para or105\_r2.

	IBk	RdnT	RdnF	AddR	M5P	MLP	SMOreg	SLR
CC	0,9997	0,9995	0,9985	0,9876	0,9349	0,9123	0,6678	0,6685
EAM	0,0104	0,0286	0,0504	0,1369	0,3538	0,4195	0,9443	1,0945
EQM	0,0397	0,0499	0,0898	0,2608	0,5934	0,6832	1,4896	1,2346
EAR	0,6372%	1,7478%	3,0800%	8,3569%	21,6027%	25,6187%	57,6643%	66,8322%
EQR	2,3922%	3,0038%	5,4089%	15,7124%	35,7444%	41,1524%	89,7321%	74,3709%
NTI	272	272	272	272	272	272	272	272

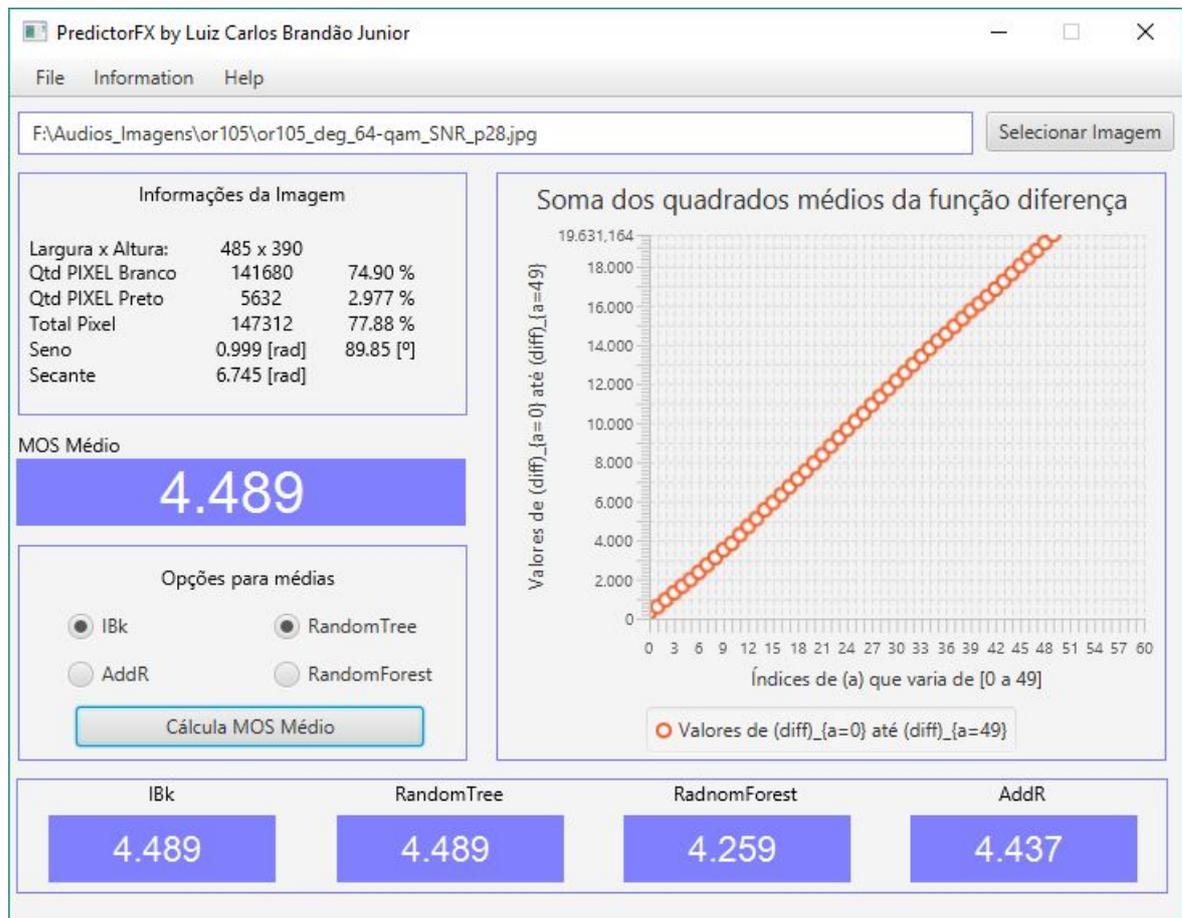
Fonte: do autor

Ao analisar a tabela percebe-se que tanto o EAM como o EQM do algoritmo IBk são menores se comparado aos demais, indicando uma boa aceitação desse *classifier* para esse tipo de problema. Nota-se também que os coeficientes de correlação estão bem próximos para os três melhores ranqueados, embora o RdnF tenha um EQM cerca de 2,26 vezes maior do que o melhor ranqueado.

Figura 4.8 – Predição da imagem or105 com SNR de 28dB e modulação 64-QAM para ATR2



[a] *Threshold* da imagem or105 com SNR de 28dB e modulação 64-QAM

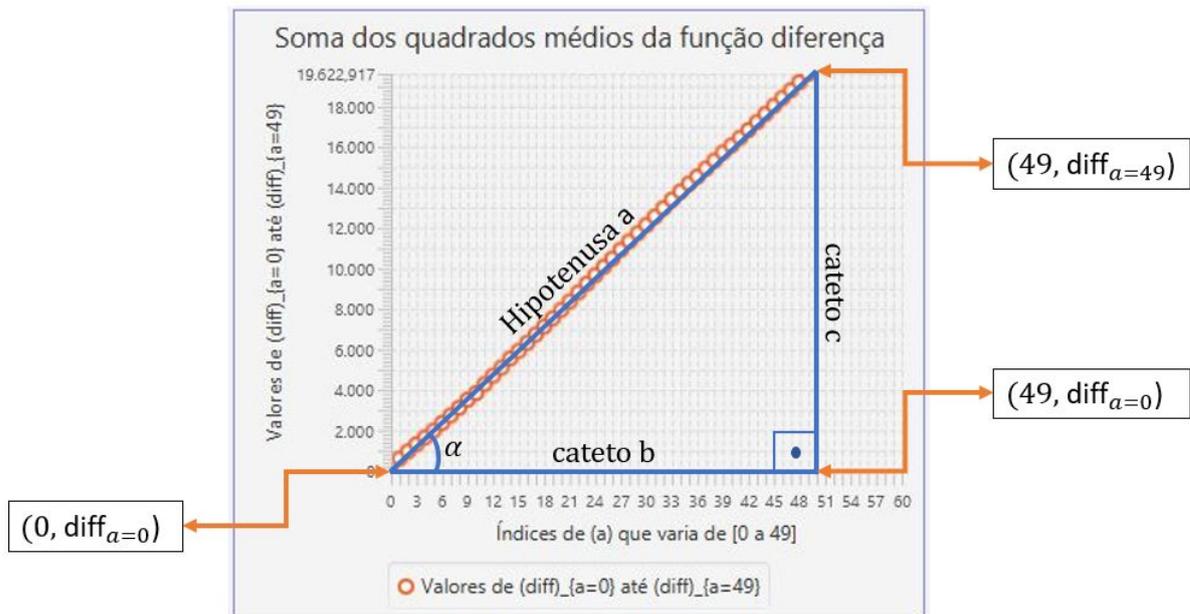


[b] *Software PredictorFX*

Quanto mais distante o algoritmo estiver do melhor ranqueado menos assertivo ele é, então preferiu-se levar em consideração o resultado somente dos 4 primeiros algoritmos para se construir os modelos no *software* PredictorFX. A Figura 4.8 [a] é uma das representações

do gráfico no domínio do tempo referente à degradação do arquivo original or105, cuja imagem pertence ao conjunto de imagens degradadas que foram usadas para fazer o treinamento dos algoritmos no Passo 1 sendo a imagem or105\_deg\_64-qam\_SNR\_p28. A Figura 4.8 [b] demonstra a parte do *software* PredictorFX feito para o ATR2 contendo os modelos dos quatro primeiros algoritmos treinados. A Imagem carregada do áudio representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 64-QAM e a relação sinal-ruído de 28dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de 4,5. Pela figura nota-se que os valores dos modelos dos algoritmos IBk e RdnT foram idênticos com valor de 4,489. O RdnF foi de 4,259 e o Addr foi de 4,437. Em Informações da imagem mostra as dimensões da respectiva figura carregada; a quantidade de pixel-brancos e pretos, suas porcentagens; o total de pixel, que percebe-se não é 100%, uma vez que, o escaneamento inicia no Pixel L, na posição (49,5) e termina no Pixel J, na posição (480,345) correspondendo a 77,88% de toda a imagem.

Figura 4.9 – Cálculo da secante no triângulo retângulo



Fonte: do Autor

O cálculo da secante que simplesmente é uma medida que foi usada, caso o valor do ângulo alfa desse valores muito próximos para figuras distintas, onde sua medida é transformada em radianos. Que nada mais é do que a hipotenusa da Figura 4.9 sobre o valor do eixo das abscissas (cateto b) que corresponde ao valor de 50 e traz a informação de que quanto menor este valor para um mesmo conjunto de imagens, melhor será o MOS da figura, nesta imagem ela tem o valor de 6,745 radianos; e o seno do ângulo que vai estar sempre próximo de 90° graus, uma vez que, o eixo x é muito pequeno se comparado ao eixo y, algumas vezes, na casa

dos milhões, nesta referida imagem o valor do ângulo alfa em radianos é 0,999996573 e em graus é de 89,85°. Cada círculo vermelho da Figura 4.9 corresponde aos valores do Vetor50.

A Tabela 4.12 mostra o resultado da avaliação do banco de dados or109 que em nenhuma vez foi usado para fazer qualquer tipo de treinamento, percebe-se que o algoritmo treinado RdnT se mostrou melhor por ter o EAM e EQM menores do que os demais. Embora o RdnF tenha o segundo maior coeficiente de correlação, o seu EAM é cerca de 2,79 vezes maior e o EQM é cerca de 1,67 vezes maior do que o RdnT. O EAM e o EQM do algoritmo IBk foram os segundos menores erros na comparação com os demais.

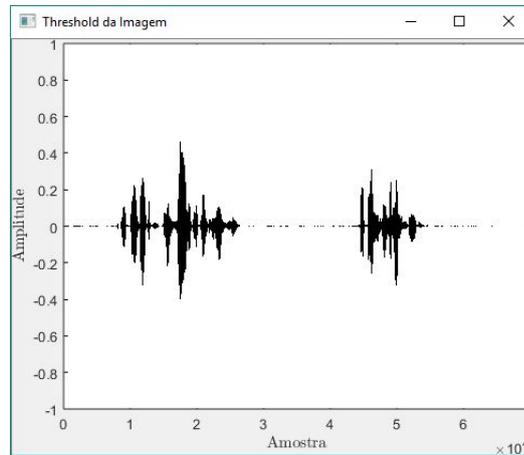
Tabela 4.12 – Validação do modelo para os arquivos degradados or109\_r2

	IBk	RdnT	RdnF	AddR	M5P	MLP	SMOreg	SLR
CC	0,9792	0,9888	0,9875	0,0251	0,9630	0,9179	0,6667	0,6732
EAM	0,1499	0,1375	0,3833	1,6734	0,3282	0,4131	1,0103	1,1216
EQM	0,3634	0,2751	0,4617	2,2733	0,4841	0,6825	1,5859	1,2719
NTI	272	272	272	272	272	272	272	272

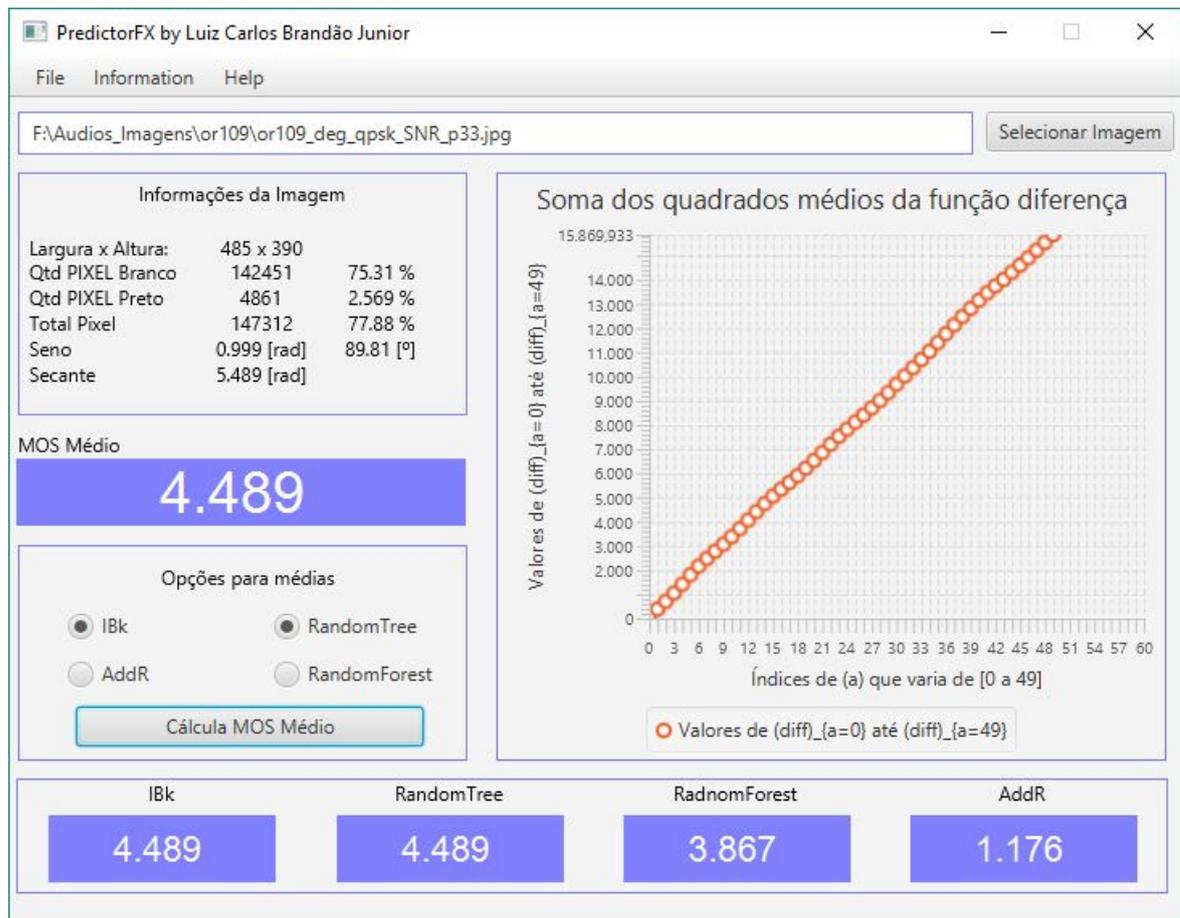
Fonte: do autor

A imagem carregada da Figura 4.10 [a] é uma das representações do gráfico no domínio do tempo referente à degradação do arquivo original or109 que foi adquirido no Passo 1, sendo a imagem or109\_deg\_qpsk\_SNR\_p33, que representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação QPSK e a relação sinal-ruído de 33dB, cujos parâmetros no Passo 1 gerou também um MOS com o valor de 4,5. Pela Figura 4.10 [b] nota-se que os modelos dos algoritmos IBk e RdnT ficaram bem próximos, com os valores de 4,489 que foram iguais aos da Figura 4.8. O MOS do RdnF foi de 3,867 e AddR de 1,176. O valor do ângulo alfa foi de 0,9999945 radianos e em graus de 89,81°, o valor da secante foi de valor de 5,489 radianos.

Figura 4.10 – Predição da imagem or109 com SNR de 33dB e modulação QPSK para ATR2



[a] *Threshold* da imagem or109 com SNR de 33dB e modulação QPSK



[b] *Software PredictorFX*

A Tabela 4.13 mostra o resultado da avaliação do banco de dados or114 que em nenhuma vez foi usado para fazer qualquer tipo de treinamento, percebe-se que o algoritmo treinado RdnT teve o maior CC, os menores EAM e EQM na comparação com os demais. O segundo maior

CC foi o do algoritmo IBk que teve também os segundos menores EAM e EQM na comparação com os outros. O RdnF teve o terceiro melhor CC, porém seu EAM foi cerca de 1,93 vezes maior e o seu EQM foi 1,34 vezes maior do que o RdnT.

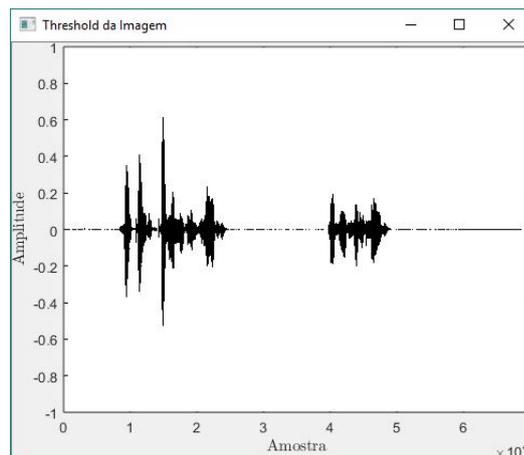
Tabela 4.13 – Validação do modelo para os arquivos degradados or114\_r2

	IBk	RdnT	RdnF	AddR	M5P	MLP	SMOreg	SLR
CC	0,9889	0,9896	0,9853	0,1050	0,9494	0,9384	0,7109	0,7102
EAM	0,2395	0,2304	0,4440	1,7472	0,4724	0,5068	0,9578	1,1256
EQM	0,3693	0,3594	0,4814	2,2596	0,6449	0,6976	1,5343	1,2967
NTI	272	272	272	272	272	272	272	272

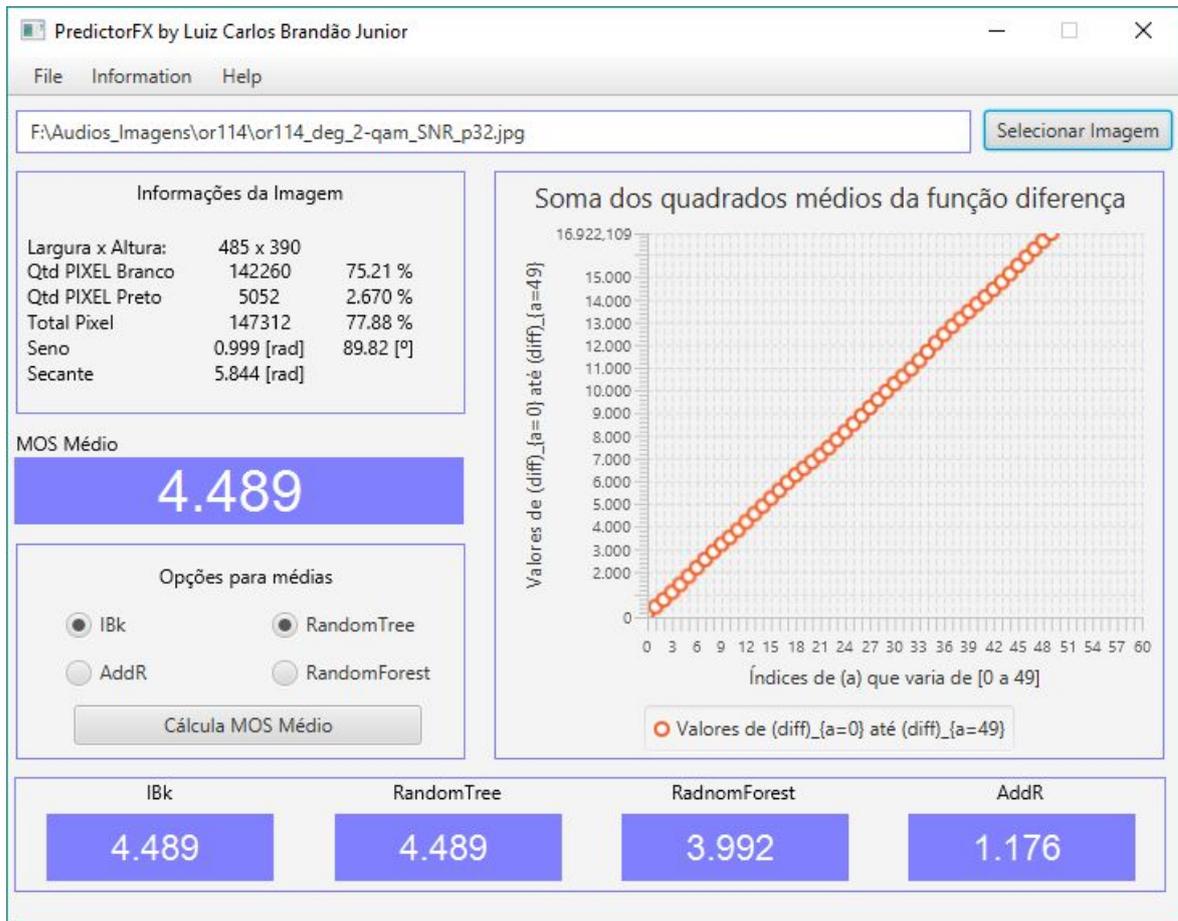
Fonte: do autor

A Figura 4.11 mostra o conjunto de dados da imagem or114\_deg\_2-qam\_SNR\_p32 que representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 2-QAM e a relação sinal-ruído de 32dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de 4,5. Pela figura nota-se que os valores do MOS para os modelos dos algoritmos IBk e RdnT foram idênticos aos valores das Figuras 4.8 [b] e 4.10 [b], o AddR teve o mesmo valor ao da Figura 4.10 [b], o único que mudou foi o MOS do RdnF com o valor de 3,992. O valor do ângulo alfa foi de  $89,82^\circ$  graus e a secante de 5,844 radianos.

Figura 4.11 – Predição da imagem or114 com SNR de 32dB e modulação 2-QAM para ATR2



[a] *Threshold* da imagem or114 com SNR de 32dB e modulação 2-QAM



[b] Software PredictorFX

A Tabela 4.14 mostra o resultado da avaliação do banco de dados or129 que em nenhuma vez foi usado para fazer qualquer tipo de treinamento, percebe-se que o algoritmo treinado IBk teve o maior coeficiente de correlação e menor EAM na comparação com os demais, porém tem um EQM cerca de 2,16% maior do que o segundo melhor ranqueado.

Tabela 4.14 – Validação do modelo para os arquivos degradados or129\_r2

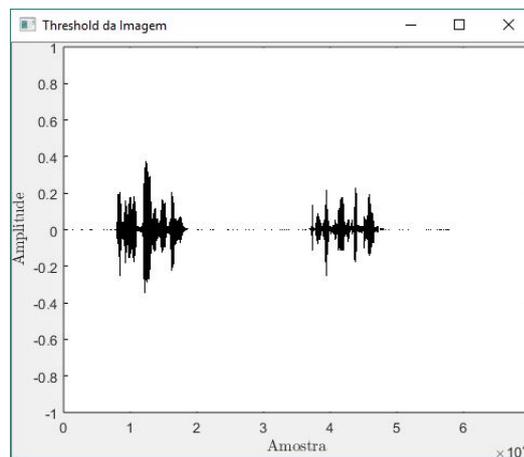
	IBk	RdnT	RdnF	AddR	M5P	MLP	SMOreg	SLR
CC	0,9568	0,9595	0,9252	0,0493	0,9356	0,8756	0,6400	0,6454
EAM	0,2832	0,2845	0,7144	1,8561	0,4229	0,5386	0,9231	1,0417
EQM	0,5002	0,4896	0,7987	2,3496	0,5670	0,7579	1,4611	1,1863
NTI	272	272	272	272	272	272	272	272

Fonte: do autor

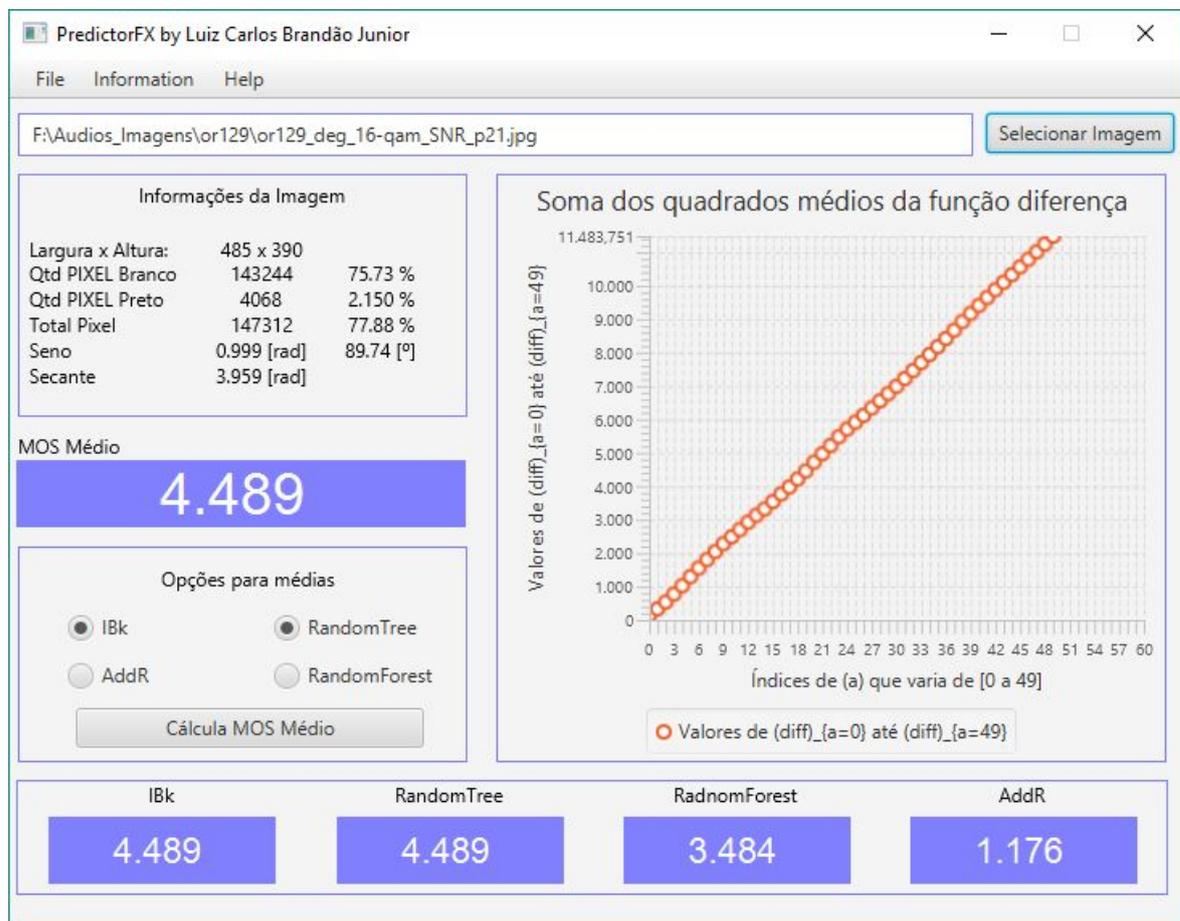
A Figura 4.12 mostra o conjunto de dados da imagem or129\_deg\_16-qam\_SNR\_p21 que representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 16-QAM e relação sinal-ruído de 21dB, cujos parâmetros no Passo 1 gerou um MOS com

o valor de 4,5. Pela figura nota-se que o valor do MOS para o IBk foi de 4,489, idêntico ao RdnT, o AddR foi de 1,76 e do RdnF de 3,484. O ângulo alfa foi de  $89,74^\circ$  graus e a secante de 3,959 radianos

Figura 4.12 – Predição da imagem or129 com SNR de 21dB e modulação 16-QAM para ATR2



[a] *Threshold* da imagem or129 com SNR de 21dB e modulação 16-QAM



[b] *Software PredictorFX*

A Tabela 4.15 mostra o resultado da avaliação do banco de dados or134 que em nenhuma vez foi usado para fazer qualquer tipo de treinamento, percebe-se que o algoritmo treinado IBk foi o que teve maior CC em comparação com os demais, teve menor EAM e EQM. O segundo melhor foi o RdnT seguido do RdnF. O AddR teve coeficiente negativo, isso demonstra sua ineficiência de predição para esse tipo de problema.

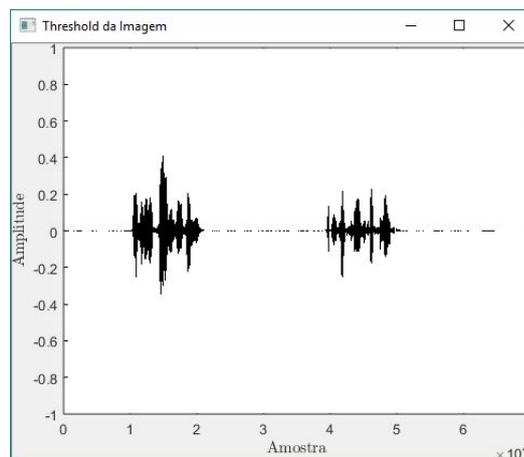
Tabela 4.15 – Validação do modelo para os arquivos degradados or134\_r2

	IBk	RdnT	RdnF	AddR	M5P	MLP	SMOreg	SLR
CC	0,9513	0,9493	0,9158	-0,0608	0,9129	0,8638	0,6047	0,6108
EAM	0,3213	0,3214	0,7419	1,8354	0,4650	0,5617	0,9451	1,0496
EQM	0,5344	0,5453	0,8295	2,3218	0,6379	0,7821	1,4711	1,1872
NTI	272	272	272	272	272	272	272	272

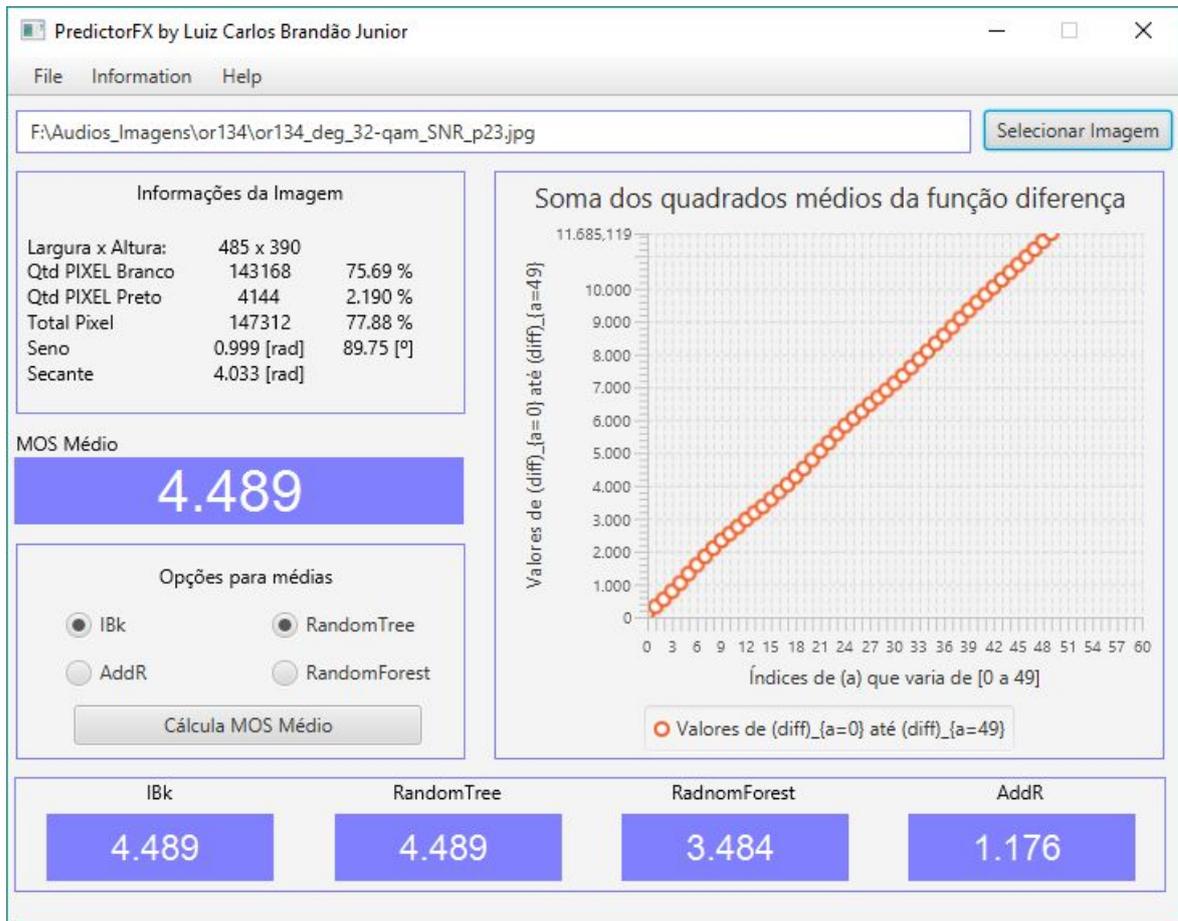
Fonte: do autor

A Figura 4.13 mostra o conjunto de dados da imagem or134\_deg\_32-qam\_SNR\_p23. Essa figura representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 32-QAM e relação sinal-ruído de 23dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de 4,5. Pela figura nota-se que os valores do modelo dos algoritmos foram os mesmos encontrados na Figura 4.12 [b]. O valor do ângulo alfa foi de  $89,75^\circ$  graus e a secante foi de 4,033 radianos.

Figura 4.13 – Predição da imagem or134 com SNR de 23dB e modulação 32-QAM para ATR2



[a] Threshold da imagem or134 com SNR de 23dB e modulação 32-QAM



[b] Software PredictorFX

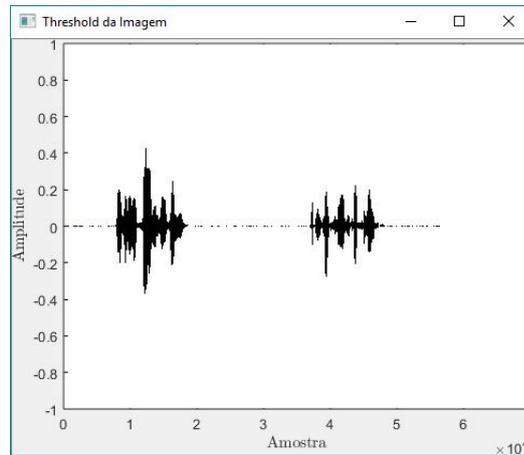
A Tabela 4.16 mostra o resultado da avaliação do banco de dados or137 que em nenhuma vez foi usado para fazer qualquer tipo de treinamento, percebe-se que o algoritmo treinado RdnT teve maior CC, menores EAM e EQM na comparação com os demais. O M5P teve os segundo melhores valores para CC, EAM e EQM, seguido por RdnF com o terceiro maior CC, menores EAM e EQM. O IBk e o AddR tiveram seus CC negativos, isso demonstra que nessa base de dados as grandezas são inversamente proporcionais.

Tabela 4.16 – Validação do modelo para os arquivos degradados or137\_r2

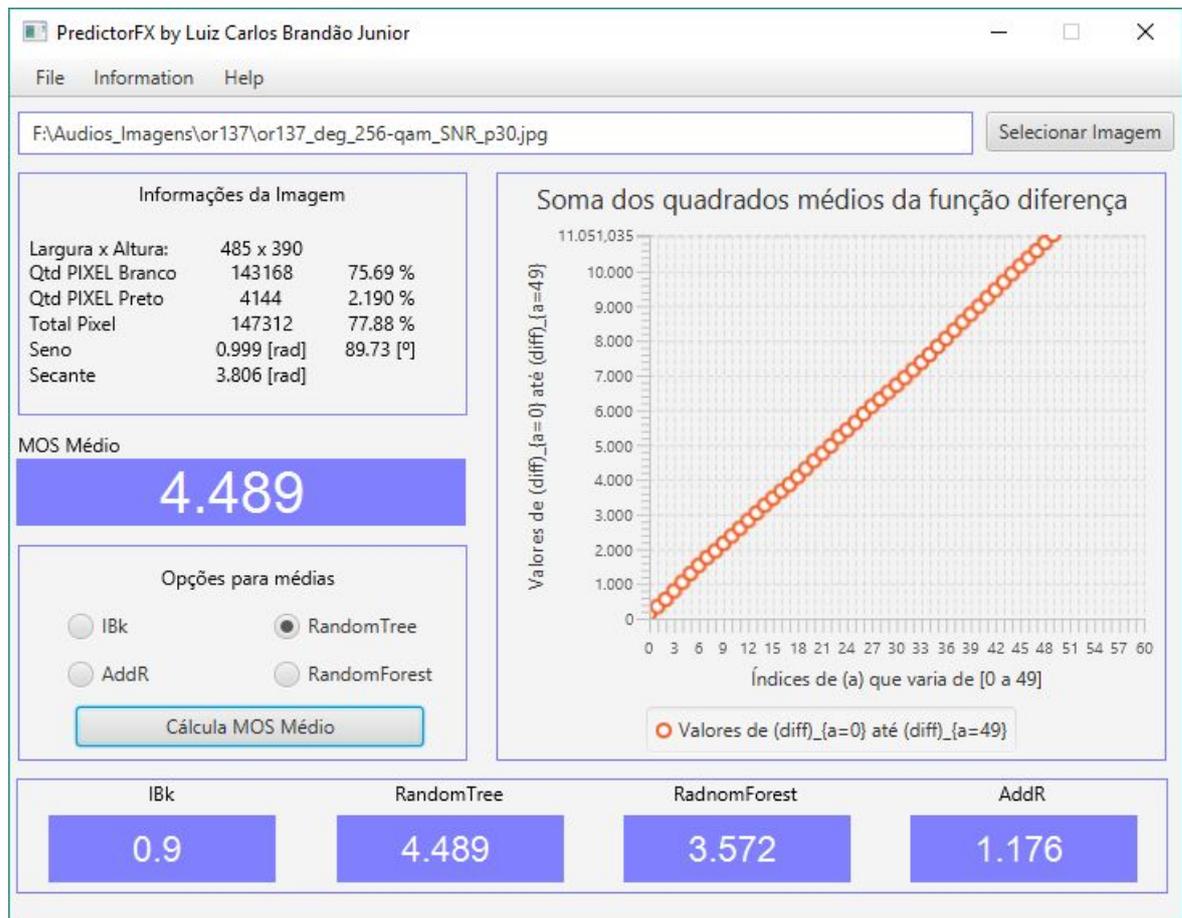
	IBk	RdnT	RdnF	AddR	M5P	MLP	SMOreg	SLR
CC	-0,2034	0,9421	0,9075	-0,0737	0,9272	0,8906	0,6212	0,6281
EAM	1,9346	0,2943	0,6825	1,8086	0,4143	0,5039	0,9720	1,0704
EQM	2,4930	0,5657	0,7949	2,3265	0,5942	0,7174	1,5113	1,2122
NTI	272	272	272	272	272	272	272	272

Fonte: do autor

Figura 4.14 – Predição da imagem or137 com SNR de 30dB e modulação 256-QAM para ATR2



[a] Threshold da imagem or137 com SNR de 30dB e modulação 256-QAM



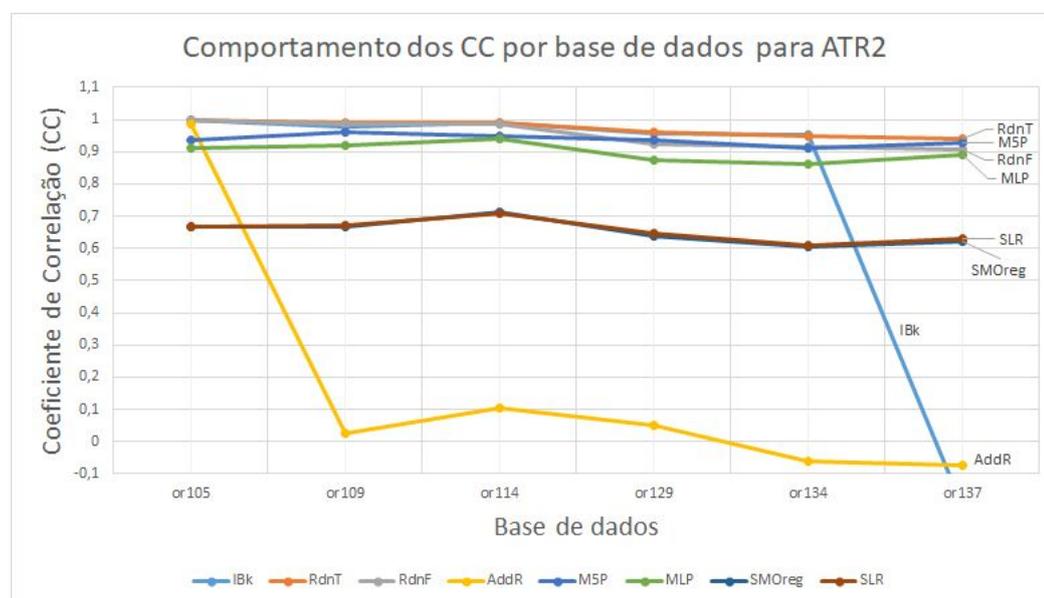
[b] Software PredictorFX

A Figura 4.14 [a] mostra a imagem or137\_deg\_256-qam\_SNR\_p30 que representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 256-QAM e relação sinal-ruído de 30dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de

4,5. Pela figura nota-se que o valor do MOS do modelo AddR foi de 1,176, o MOS do IBk foi de 0,9; RdnF foi de 3,572 e o RdnT foi de 4,489 esse último valor sendo igual a todos os testes anteriores realizados referentes aos arquivos degradados das bases de dados or105, or109, or114, or129 e or134. O valor do ângulo alfa foi de  $89,73^\circ$  graus e a secante de 3,806 radianos.

O gráfico da Figura 4.15 mostra o comportamento dos coeficientes de correlação para cada base de dados, percebe-se que na base de dados or105 que contém os dados dos arquivos degradados que foram usados para treinar os algoritmos, somente o SMOreg e o SLR que ficaram abaixo de 0,7, todos os outros ficaram acima de 0,9123. A partir da base de dados dos arquivos degradados or109 começa o processo de validação do modelo pelos algoritmos treinados, denominado aqui de Modelo RazaR, percebe-se que os algoritmos treinados RdnT, M5P, RdnF e MLP conseguem prever, nas diferentes bases de dados, os MOS com valores do coeficiente de correlação variando entre 0,8638 de mínima e 0,9997 de máxima. Isso indica que os algoritmos treinados usando a base de dados dos arquivos degradados or105 conseguem prever o MOS em outras bases com outros dados, valores esses que em nenhuma vez foram usados para treinar os algoritmos.

Figura 4.15 – Comportamento dos CC para ATR2.



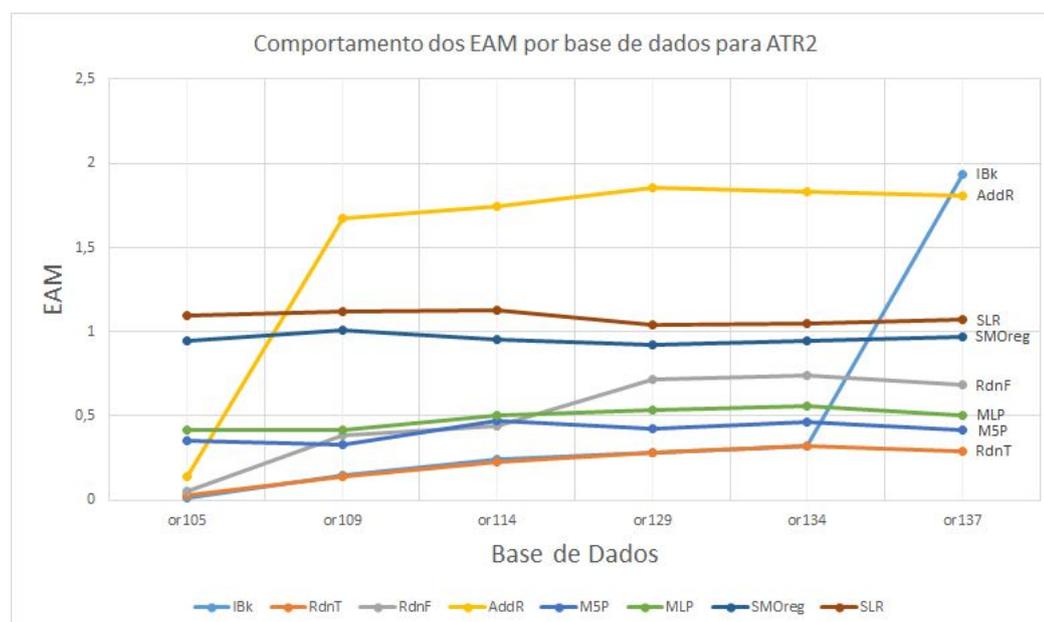
Fonte: do Autor

Ao atingir a base de dados or129 nota-se que o AddR já não foi capaz de prever corretamente os valores, mantendo esse comportamento até o final indicando assim sua ineficiência para esse tipo de problema. O algoritmo IBk não foi capaz de prever corretamente os dados da base or137 indicando sua ineficiência nessa base. Pelo comportamento tanto do AddR como

do IBk ao ter valores abaixo de zero percebe-se que o modelo construído com a base de dados or105 tem grandezas inversamente proporcionais a base de dados or134 e or137, determinar quais grandezas são essas requer um pouco mais de investigação. Percebe-se ainda que o M5P e o MLP embora ficaram fora dos 4 primeiros melhores ranqueados pelo teste de hipótese eles mantiveram seus valores de CC constantes nas diferentes bases de dados. Os algoritmos SLR e SMOreg também conseguiram manter constante seus valores de CC, só que com valores mais baixos, percebe-se que houve quase uma sobreposição de valores. O melhor algoritmo treinado foi o RdnT sendo capaz de prever os dados em bases distintas com coeficiente de correlação altíssimo.

A Figura 4.16 mostra o comportamento do erro absoluto médio para cada algoritmo ao longo das diferentes bases de dados e percebe-se que os algoritmos que tiveram menores erros foram o RdnT e IBk, quase tendo uma sobreposição de valores. Na última base de dados o EAM do IBk teve um aumento súbito terminando como o último algoritmo com o pior erro. O RdnF começou como terceiro melhor algoritmo e o seu erro foi aumentando terminando como o quarto maior erro. O MLP e o M5P tiveram erros baixos do início ao fim, se comparado com SLR e SMOreg.

Figura 4.16 – Comportamento dos EAM para ATR2.

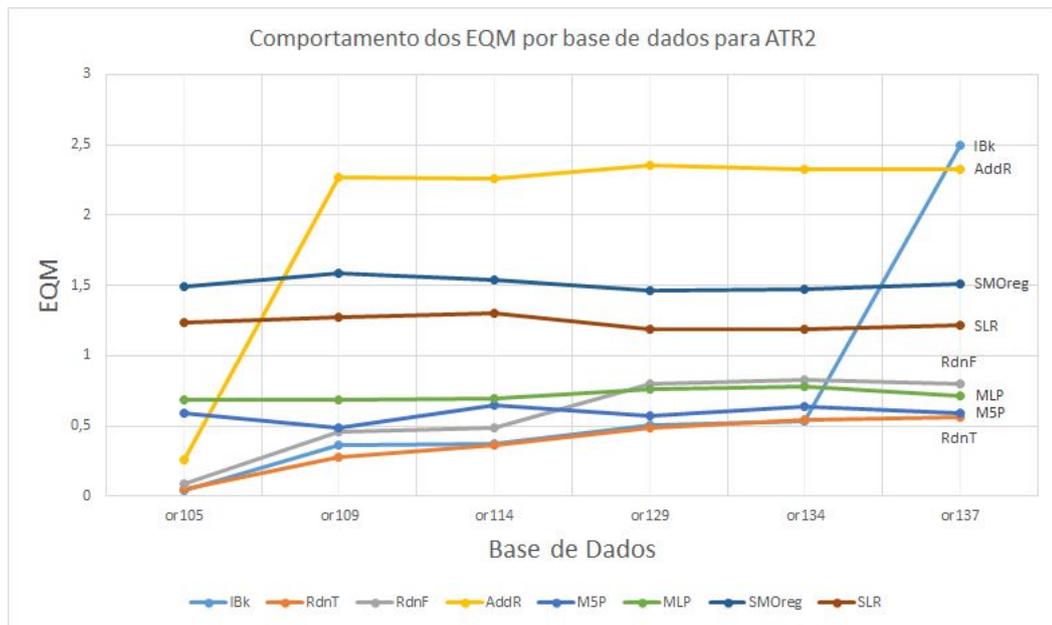


Fonte: do Autor

A Figura 4.17 mostra o comportamento do erro quadrático médio para cada algoritmo ao longo das diferentes bases de dados e percebe-se que o RdnT ao longo das bases teve o menor erro, o IBk que ficou próximo dele, na última base apresentou o maior erro. Embora o

MLP apresentou um erro maior nas diferentes bases de dados, percebe-se que esse erro é quase constante, não tendo grandes variações como ocorreu no M5P e RdnF. O AddR teve o segundo maior erro desde a base de dados or109, os erros do SMOreg e SLR apesar de estarem altos na comparação com o RdnT, percebe-se que também ficaram quase constantes ao longo das diferentes bases de dados.

Figura 4.17 – Comportamento dos EQM para ATR2.



Fonte: do Autor

## 4.2 Resultado da Classificação para ATC1

Cada resultado de cada *classifier* corresponde a uma média de 50 validações das instâncias corretamente classificadas com a opção *K-fold* do *Cross-Validation* igual a 10. A Tabela 4.17 mostra o resultado dessas médias, onde o algoritmo que teve uma média superior foi o J48 com 96,01 %. Nota-se que só o MLP que não teve o coeficiente de correlação maior que 90%.

Tabela 4.17 – Médias dos coeficientes de correlação para ATC1.

	J48	RdnF	IBk	RdnT	NaiveBayes	JRip	OneR	MLP
Média	96,01%	95,99%	95,74%	95,58%	94,98%	94,32%	93,21%	89,45%

Fonte: do autor

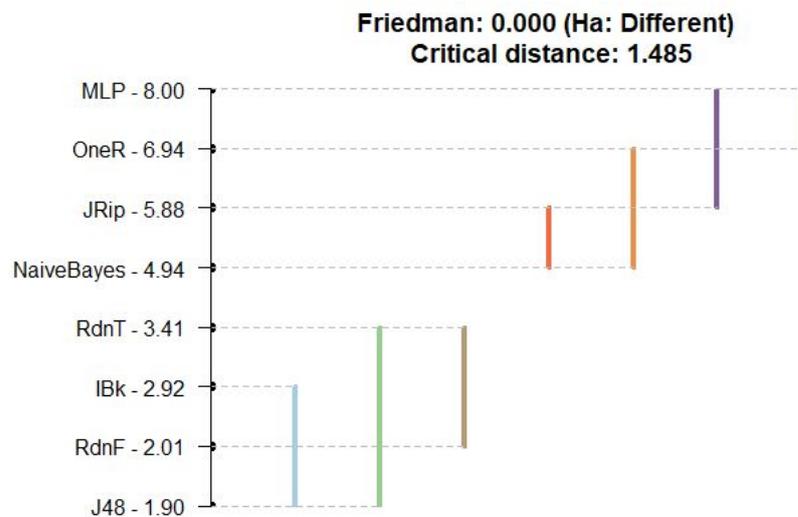
Tabela 4.18 – Ranqueamento para ATC1.

	J48	RdnF	IBk	RdnT	NaiveBayes	JRip	OneR	MLP
Média	1,90	2,01	2,92	3,41	4,94	5,88	6,94	8,00
Ranking	1º	2º	3º	4º	5º	6º	7º	8º

Fonte: do autor

Pela Tabela 4.18 percebe-se que o *classifier* J48 fica em primeiro, como o algoritmo melhor ranqueado, seguido por RdnF e IBk. Pelo teste de hipótese demonstrado na Figura 4.18, a Hipótese alternativa ( $H_a$ ) diz que os ranques médios são distintos entre si com uma distância crítica de 1,485; tendo quatros *classifiers* melhores classificados onde três deles não se pode afirmar que exista alguma diferença estatística significativa entre eles, com o J48 melhor ranqueado.

Figura 4.18 – Teste de Friedman e Nemenyi para ATC1.



Fonte: do Autor

A Tabela 4.19 mostra os sumários de todos os algoritmos treinados usando a totalidade da base de dados or105 para treinamento e teste. Sendo o processo denominado de Modelo KazaM, onde ICC é a sigla para as Instâncias Corretamente Classificadas, IIC para Instâncias Incorretamente Classificadas, EKP para Estatística Kappa que é uma medida de concordância usada em escalas nominais que fornece uma ideia do quanto as observações se afastam daquelas esperadas, fruto do acaso, indicando assim o quão legítimas as interpretações foram. EAM para Erro Absoluto Médio que é a diferença entre o valor da Tabela 3.2 e o valor previsto, trata-se de uma maneira de considerar o erro ao medir a precisão de valores. EQM para Erro Quadrático Médio é definido como sendo a média da diferença entre o valor do estimador e do parâmetro ao

quadrado indicando que quanto menor a variância melhor é o algoritmo treinado, EAR para Erro Absoluto Relativo, EQR para Erro Quadrático Relativo, NTI para Número Total de Instâncias e NvBy para NaiveBayes.

Tabela 4.19 – Sumário dos Algoritmos Treinados para or105\_c1.

	ICC	IIC	EKP	EAM	EQM	EAR	EQR	NTI
J48	266 (97,7941%)	6 (2,2059%)	0,96	0,0155	0,0882	5,6723%	23,8805%	272
RdnF	272 (100%)	0 (0%)	1	0,0087	0,0484	3,1792%	13,1006%	272
IBk	272 (100%)	0 (0%)	1	0,0030	0,0046	1,1011%	1,2585%	272
RdnT	272 (100%)	0 (0%)	1	0	0	0 %	0 %	272
NvBy	259 (95,2206%)	13 (4,7794%)	0,9124	0,0239	0,1546	8,7194%	41,8714%	272
JRip	266 (97,7941%)	6 (2,2059%)	0,96	0,0155	0,0882	5,6723%	23,8805%	272
OneR	258 (94,8529%)	14 (5,1471%)	0,9054	0,0257	0,1604	9,3900%	43,4521%	272
MLP	252 (92,6471%)	20 (7,3529%)	0,8599	0,0802	0,1869	29,2499%	50,6280%	272

Fonte: do autor

Embora o algoritmo JRip tenha dado os mesmos resultados que o J48 na Tabela 4.19, pelo teste de hipótese ele foi o antepenúltimo algoritmo pior classificado. Três algoritmos acertaram todas as instâncias, sendo eles o RdnF, IBk e RdnT o melhor foi o RdnT com erros zeros e Estatística Kappa excelente.

A base de dados dos arquivos degradados or105 tem 133 imagens de MOS péssimo, 7 de MOS ruim, 6 razoável e 126 com MOS bom, a Tabela 4.20 mostra as matrizes de confusão para os algoritmos, onde RdnF, IBk e RndT conseguiram acertar corretamente todas as instâncias. O J48 e o JRip tiveram os mesmos acertos e erros. NaiveBayes e MLP não conseguiram acertar nenhuma como razoável, OneR também não acertou nenhuma como ruim.

Tabela 4.20 – Matrizes de Confusão dos algoritmos treinados para ATC1.

J48					RdnF				
a	b	c	d	<- classificou como	a	b	c	d	<- classificou como
131	2	0	0	a = péssimo	133	0	0	0	a = péssimo
0	5	2	0	b = ruim	0	7	0	0	b = ruim
0	0	6	0	c = razoável	0	0	6	0	c = razoável
0	0	2	124	d = bom	0	0	0	126	d = bom

[a] Matriz de Confusão para J48

[b] Matriz de Confusão para RdnF

IBk				
a	b	c	d	<- classificou como
133	0	0	0	a = péssimo
0	7	0	0	b = ruim
0	0	6	0	c = razoável
0	0	0	126	d = bom

[c] Matriz de Confusão para IBk

RdnT				
a	b	c	d	<- classificou como
133	0	0	0	a = péssimo
0	7	0	0	b = ruim
0	0	6	0	c = razoável
0	0	0	126	d = bom

[d] Matriz de Confusão para RdnT

NaiveBayes				
a	b	c	d	<- classificou como
128	5	0	0	a = péssimo
0	6	0	1	b = ruim
0	1	0	5	c = razoável
0	1	0	125	d = bom

[e] Matriz de Confusão para NaiveBayes

JRip				
a	b	c	d	<- classificou como
131	2	0	0	a = péssimo
0	5	2	0	b = ruim
0	0	6	0	c = razoável
0	0	2	124	d = bom

[f] Matriz de Confusão para JRip

OneR				
a	b	c	d	<- classificou como
128	0	0	5	a = péssimo
3	0	4	0	b = ruim
0	0	6	0	c = razoável
0	0	2	124	d = bom

[g] Matriz de Confusão para OneR

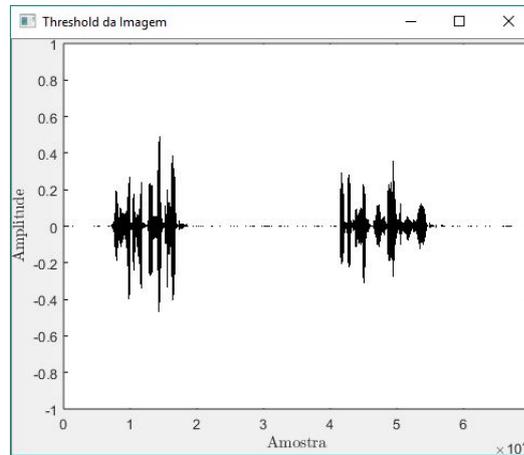
MLP				
a	b	c	d	<- classificou como
126	0	0	7	a = péssimo
0	0	0	7	b = ruim
0	0	0	6	c = razoável
0	0	0	126	d = bom

[h] Matriz de Confusão para MLP

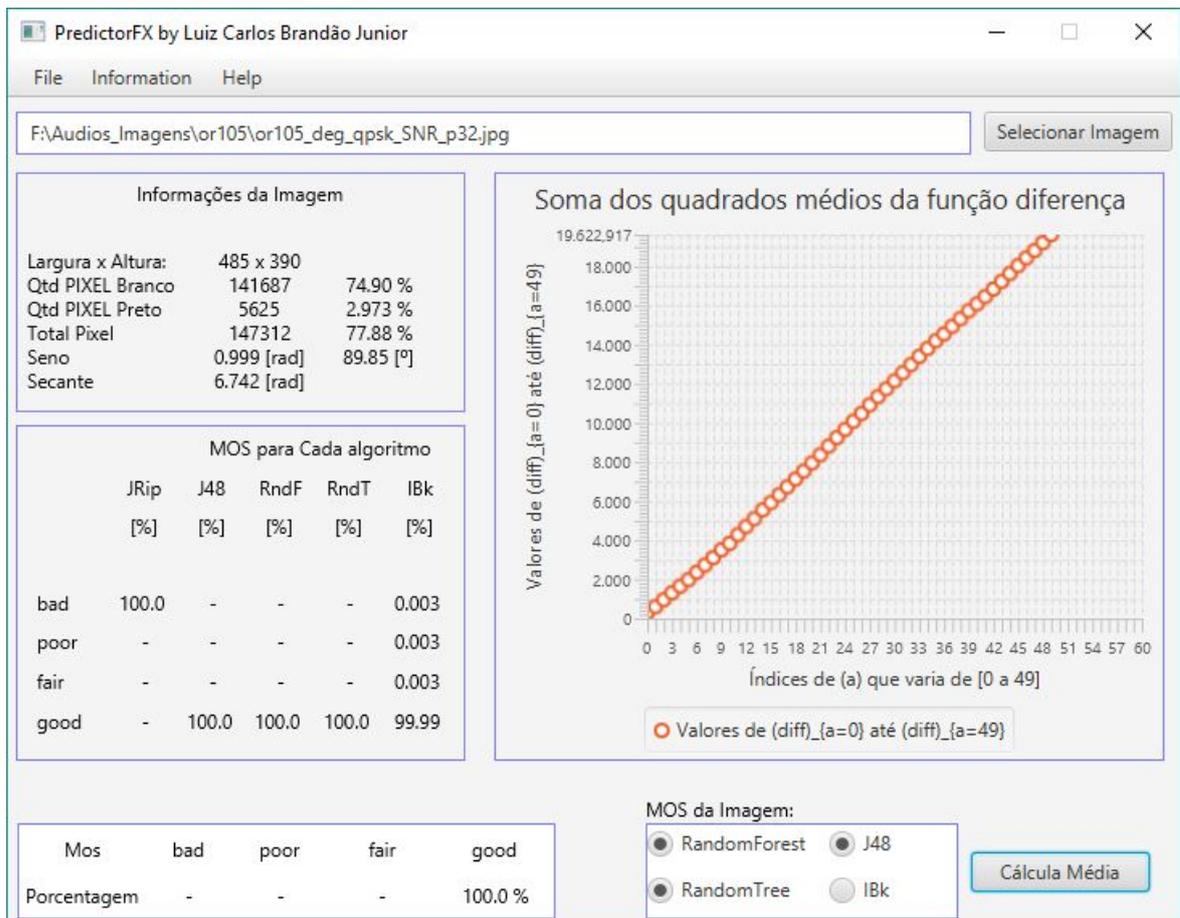
Fonte: do autor

A Figura 4.19 mostra a imagem *or105\_deg\_qpsk\_SNR\_p32* carregada no *software* PredictorFX para a opção ATC1. Essa imagem representa o arquivo desvanecido com o nível de quantização igual a 65536, modulação QPSK e a relação sinal-ruído de 32dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de 4,5 e devido à Tabela 3.3 recebeu a classificação de um MOS bom. A imagem carregada foi usada no Passo 1 para fazer o treinamento dos algoritmos, percebe-se que o JRip mesmo tendo os valores iguais ao do J48, na Tabela 4.19 ele não é capaz de prever corretamente a imagem que foi carregada, sendo essa uma das duas que ele errou na matriz de confusão [f], classificando-a como áudio cujo MOS é péssimo. Os modelos dos algoritmos IBk, RdnF e RdnT têm 100% de certeza que a referida imagem carregada pertence a um arquivo de áudio classificado como MOS bom. O IBk tem 99,99% de certeza que também é bom. O ângulo alfa foi de 89,85° graus e a secante de 6,742 radianos.

Figura 4.19 – Predição da imagem or105 com SNR de 32dB e modulação QPSK para ATC1.



[a] *Threshold* da imagem or105 com SNR de 32dB e modulação QPSK



[b] *Software PredictorFX*

A Tabela 4.21 mostra o resultado da validação dos algoritmos treinados para as degradações do arquivo original or109 que em nenhum momento essa base de dados foi usada para

treinamento. Percebe-se que o modelo RdnT teve menores EQM e EAM e maior ICC. Seguindo pelo algoritmo RdnF por ter o segundo menor EQM e maior ICC.

Tabela 4.21 – Validação do modelo para os arquivos degradados or109\_c1

	ICC	IIC	EKP	EAM	EQM	NTI
J48	257 (94,4853%)	15 (5,5147%)	0,8989	0,0262	0,1454	272
RdnF	262 (96,3235%)	10 (3,6765%)	0,9322	0,0379	0,1199	272
IBk	259 (95,2206%)	13 (4,7794%)	0,9113	0,0265	0,1539	272
RdnT	265 (97,4265%)	7 (2,5735%)	0,9528	0,0129	0,1134	272
NaiveBayes	253 (93,0147%)	19 (6,9853%)	0,8735	0,0393	0,1869	272
JRip	126 (46,3235%)	146 (53,6765%)	-0,0266	0,2647	0,5119	272
OneR	256 (94,1176%)	16 (5,8824%)	0,8892	0,0294	0,1715	272
MLP	249 (91,5441%)	23 (8,4559%)	0,8390	0,0770	0,1854	272

Fonte: do autor

O banco de dados referente às degradações do arquivo original or109 tem 133 imagens com MOS péssimo, 8 com MOS ruim, 5 com MOS razoável e 126 com MOS bom. A Tabela 4.22 mostra as matrizes de confusão, percebe-se que o algoritmo treinado J48 conseguiu prever corretamente 126 imagens com MOS ruim e a mesma quantidade para bom. Todos os algoritmos com exceção do JRip acertaram em sua totalidade as imagens com MOS bom. O RdnT foi o único que conseguiu acertar todas as 8 imagens com MOS ruim e foi o único também que conseguiu acertar uma imagem como razoável, os demais não conseguiram.

Tabela 4.22 – Matrizes de Confusão da validação da base de dados or109.

J48					RdnF				
a	b	c	d	<- classificou como	a	b	c	d	<- classificou como
126	7	0	0	a = péssimo	130	3	0	0	a = péssimo
0	5	1	2	b = ruim	0	6	2	0	b = ruim
0	0	0	5	c = razoável	0	0	0	5	c = razoável
0	0	0	126	d = bom	0	0	0	126	d = bom

[a] Matriz de Confusão para J48

[b] Matriz de Confusão para RdnF

IBk				
a	b	c	d	<- classificou como
128	3	0	2	a = péssimo
1	5	1	1	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[c] Matriz de Confusão para IBk

RdnT				
a	b	c	d	<- classificou como
130	3	0	0	a = péssimo
0	8	0	0	b = ruim
0	1	1	3	c = razoável
0	0	0	126	d = bom

[d] Matriz de Confusão para RdnT

NaiveBayes				
a	b	c	d	<- classificou como
122	11	0	0	a = péssimo
0	5	1	2	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[e] Matriz de Confusão para NaiveBayes

JRip				
a	b	c	d	<- classificou como
126	7	0	0	a = péssimo
8	0	0	0	b = ruim
5	0	0	0	c = razoável
126	0	0	0	d = bom

[f] Matriz de Confusão para JRip

OneR				
a	b	c	d	<- classificou como
130	0	0	3	a = péssimo
0	0	4	4	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[g] Matriz de Confusão para OneR

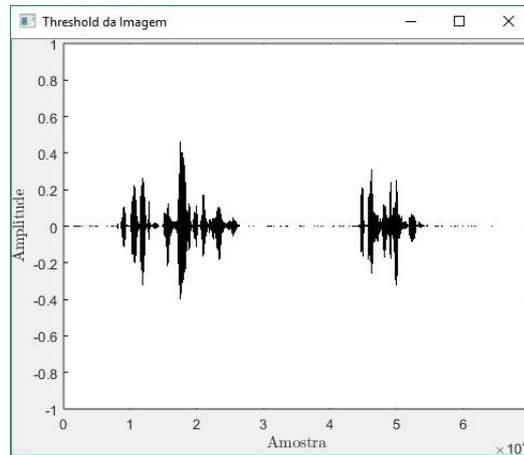
MLP				
a	b	c	d	<- classificou como
123	0	0	10	a = péssimo
0	0	0	8	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[h] Matriz de Confusão para MLP

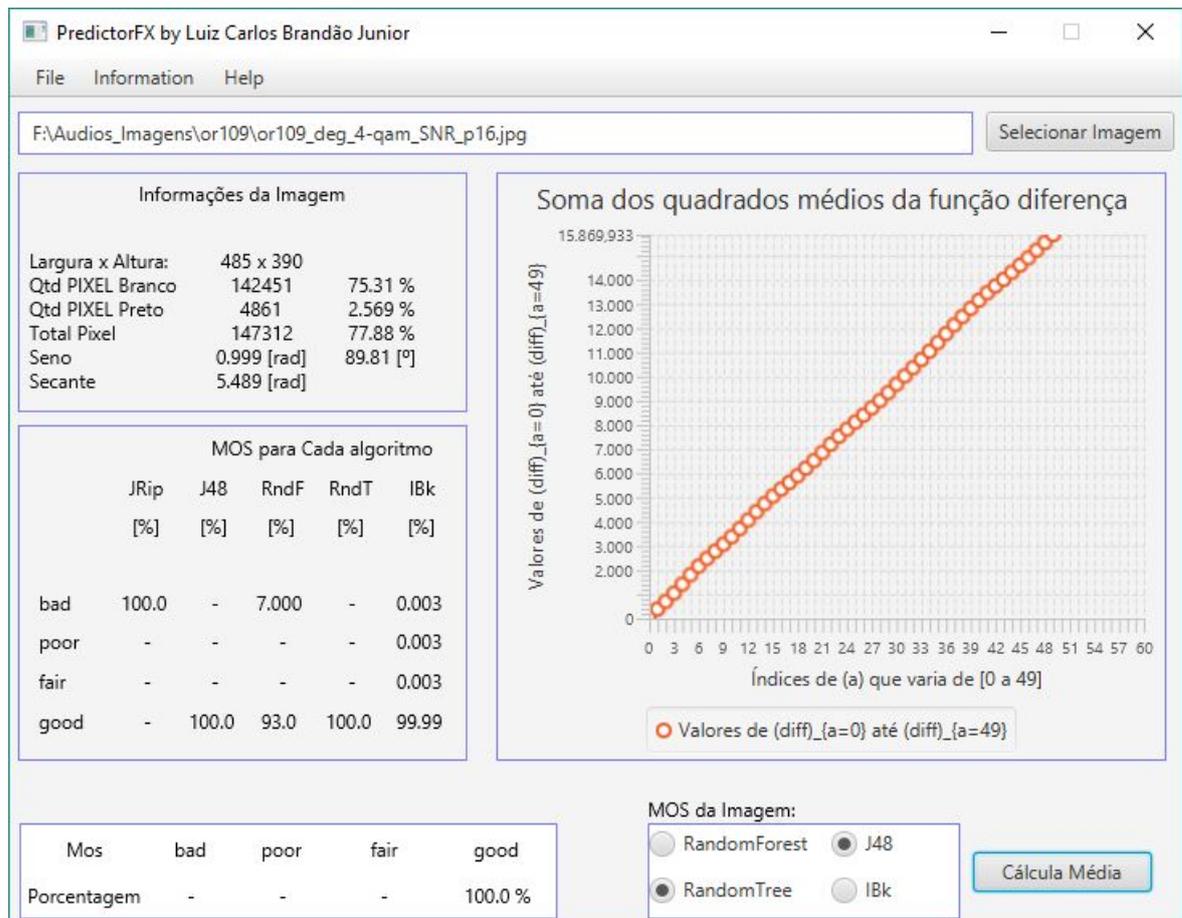
Fonte: do autor

A Figura 4.20 mostra a imagem or109\_deg\_4-qam\_SNR\_p16 carregada e representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 4-QAM e a relação sinal-ruído de 16dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de 4,5 e devido à Tabela 3.3 recebeu a classificação de áudio com MOS bom. Os modelos construídos pelos algoritmos RdnT e J48 têm 100% de certeza de que a imagem tem um MOS caracterizado como bom, RdnF têm 93% de certeza e o IBk tem 99,99%. O ângulo alfa foi de 89,81° graus e a secante de 5,489 radianos.

Figura 4.20 – Predição da imagem or109 com SNR de 16dB e modulação 4-QAM para ATC1.



[a] Threshold da imagem or109 com SNR de 16dB e modulação 4-QAM



[b] Software PredictorFX

A Tabela 4.23 mostra o resultado da validação do modelo para as degradações do arquivo original or114. Percebe-se que o modelo RdnT teve um maior acerto de instâncias e maior

estatística Kappa em relação aos demais. Além de ter um EAM menor com um EQM cerca de 1,05% maior do que o RdnF.

Tabela 4.23 – Validação do modelo para os arquivos degradados or114\_c1

	ICC	IIC	EKP	EAM	EQM	NTI
J48	258 (94,8529%)	14 (5,1471%)	0,9054	0,0236	0,1352	272
RdnF	259 (95,2206%)	13 (4,7794%)	0,9116	0,0324	0,1159	272
IBk	260 (95,5882%)	12 (4,4118%)	0,9186	0,0247	0,1480	272
RdnT	264 (97,0588%)	8 (2,9412%)	0,9459	0,0147	0,1213	272
NaiveBayes	256 (94,1176%)	16 (5,8824%)	0,8922	0,0296	0,1713	272
JRip	127 (46,6912%)	145 (53,3088%)	-0,0195	0,2630	0,5102	272
OneR	259 (95,2206%)	13 (4,7794%)	0,9099	0,0239	0,1546	272
MLP	252 (92,6471%)	20 (7,3529%)	0,8594	0,0743	0,1760	272

Fonte: do autor

A base de dados do arquivo original or114 contém 133 imagens degradadas que representa um MOS péssimo, 8 com MOS ruim, 4 com MOS razoável e 127 com MOS bom. A Tabela 4.24 mostra as matrizes de confusão e percebe-se que OneR foi o algoritmo que mais acertou imagens com MOS péssimo, e acertou todas com MOS bom, contudo não conseguiu acertar nenhuma como ruim e razoável. O RdnT errou somente um com MOS bom, os demais acertaram em suas totalidades, porém ele foi o que mais acertou com MOS ruim, errando somente 2. E também acertou 2 como razoável, nenhum outro algoritmo conseguiu prever corretamente imagens como razoável.

Tabela 4.24 – Matrizes de Confusão da validação da base de dados or114.

J48					RdnF				
a	b	c	d	<- classificou como	a	b	c	d	<- classificou como
127	6	0	0	a = péssimo	128	5	0	0	a = péssimo
0	4	3	1	b = ruim	0	4	2	2	b = ruim
0	0	0	4	c = razoável	0	0	0	4	c = razoável
0	0	0	127	d = bom	0	0	0	127	d = bom

[a] Matriz de Confusão para J48

[b] Matriz de Confusão para RdnF

IBk				
a	b	c	d	<- classificou como
128	5	0	0	a = péssimo
0	5	2	1	b = ruim
0	0	0	4	c = razoável
0	0	0	127	d = bom

[c] Matriz de Confusão para IBk

RdnT				
a	b	c	d	<- classificou como
130	3	0	0	a = péssimo
0	6	1	1	b = ruim
0	0	2	2	c = razoável
0	0	1	126	d = bom

[d] Matriz de Confusão para RdnT

NaiveBayes				
a	b	c	d	<- classificou como
124	9	0	0	a = péssimo
0	5	0	3	b = ruim
0	0	0	4	c = razoável
0	0	0	127	d = bom

[e] Matriz de Confusão para NaiveBayes

JRip				
a	b	c	d	<- classificou como
127	6	0	0	a = péssimo
7	0	1	0	b = ruim
4	0	0	0	c = razoável
127	0	0	0	d = bom

[f] Matriz de Confusão para JRip

OneR				
a	b	c	d	<- classificou como
132	0	0	1	a = péssimo
0	0	5	3	b = ruim
0	0	0	4	c = razoável
0	0	0	127	d = bom

[g] Matriz de Confusão para OneR

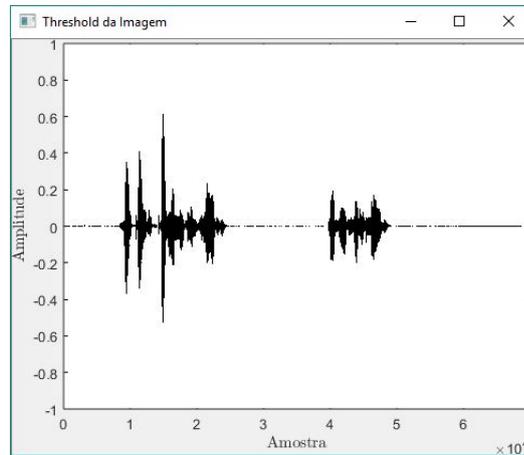
MLP				
a	b	c	d	<- classificou como
125	0	0	8	a = péssimo
0	0	0	8	b = ruim
0	0	0	4	c = razoável
0	0	0	127	d = bom

[h] Matriz de Confusão para MLP

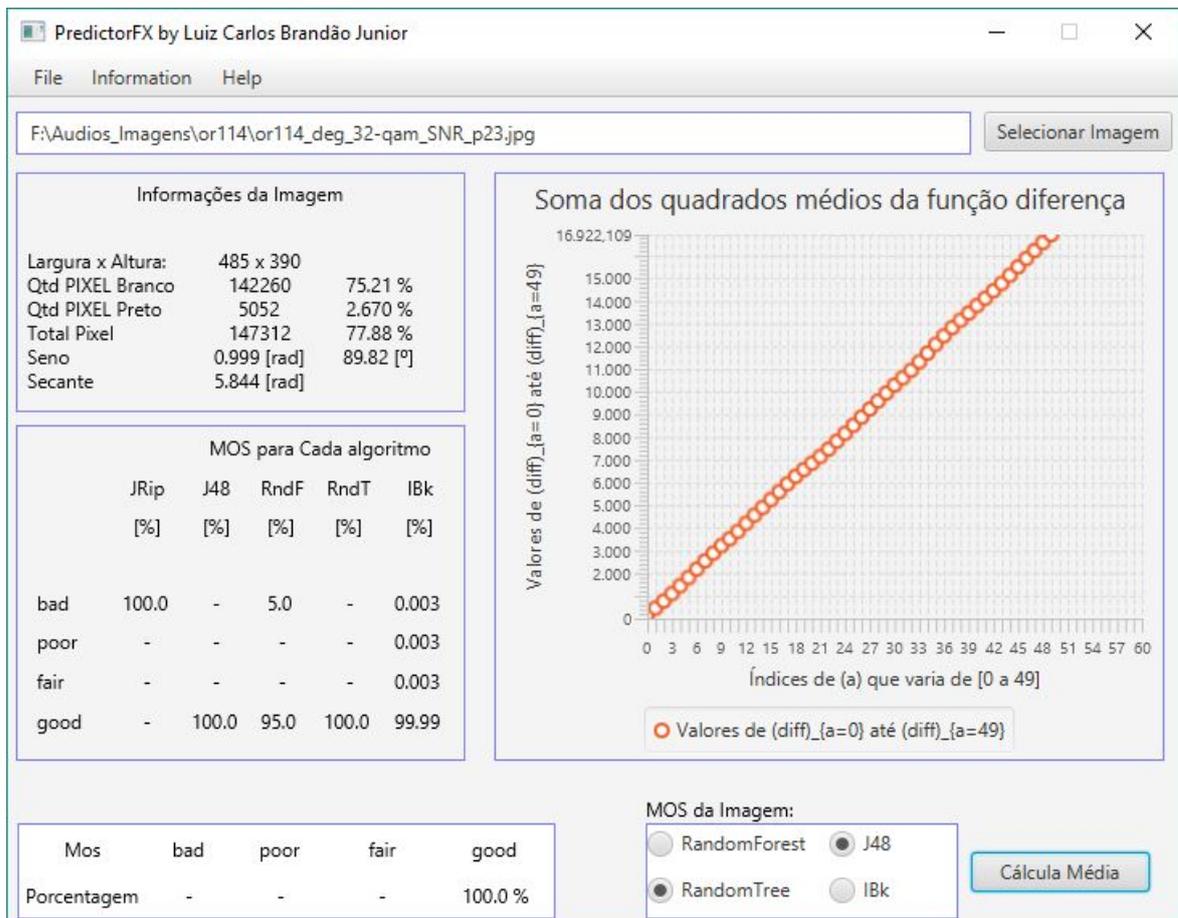
Fonte: do autor

A Figura 4.21 mostra a imagem or114\_deg\_32-qam\_SNR\_p23 carregada que representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 32-QAM e a relação sinal-ruído de 23dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de 4,1 e devido à Tabela 3.3 recebeu a classificação de MOS bom. Os algoritmos treinados RdnT e J48 têm 100% de certeza de que a imagem tem um MOS caracterizado como bom, RdnF tem somente 95% de certeza e o IBk tem 99,99%. O ângulo alfa foi de  $89,82^\circ$  graus e a secante 5,844 radianos.

Figura 4.21 – Predição da imagem or114 com SNR de 23dB e modulação 32-QAM para ATC1.



[a] *Threshold* da imagem or114 com SNR de 23dB e modulação 32-QAM



[b] *Software PredictorFX*

A Tabela 4.25 mostra o resultado da validação do modelo para as degradações do arquivo original or129, base de dados que em nenhum momento foi usado para fazer qualquer tipo de treinamento. Percebe-se que o algoritmo treinado que teve maior número de instâncias

corretamente classificadas foi o J48 com 79,04% também foi o que teve menor EQM e menor EAM. O RdnF conseguiu classificar corretamente 213 instâncias, teve o segundo maior Kappa, um EAM cerca de 46,48% e um EQM cerca de 4,91% maiores do que o J48. O terceiro melhor foi o IBk seguido por OneR.

Tabela 4.25 – Validação do modelo para os arquivos degradados or129\_c1

	ICC	IIC	EKP	EAM	EQM	NTI
J48	215 (79,0441%)	57 (20,9559%)	0,6599	0,1037	0,3113	272
RdnF	213 (78,3088%)	59 (21,6912%)	0,6454	0,1519	0,3266	272
IBk	212 (77,9412%)	60 (22,0588%)	0,6396	0,1117	0,3301	272
RdnT	89 (32,7206%)	183 (67,2794%)	0,0004	0,3364	0,5800	272
NaiveBayes	87 (31,9853%)	185 (68,0147%)	0,1736	0,3401	0,5832	272
JRip	86 (31,6176%)	186 (68,3824%)	-0,0107	0,3396	0,5809	272
OneR	211 (77,5735%)	61 (22,4265%)	0,6315	0,1121	0,3349	272
MLP	208 (76,4706%)	64 (23,5294%)	0,6061	0,1514	0,3318	272

Fonte: do autor

O banco de dados com as imagens degradadas do arquivo original or129 contém 87 imagens que representam o MOS péssimo, 54 com MOS ruim, 5 com MOS razoável e 126 com MOS bom. A Tabela 4.26 mostra as matrizes de confusão e percebe-se que o JRip foi o que mais acertou imagens com MOS péssimo, porém não conseguiu acertar nenhuma com MOS bom, assim como o NaiveBayes, o RdnT acertou somente 2 com MOS bom, os demais acertaram em suas totalidades. Os únicos que acertaram imagens com MOS ruim foram o J48 com 7, RdnF com 3, IBk com 4, RdnT com 2 e NaiveBayes com 9, os demais não conseguiram acertar. Só o NaiveBayes conseguiu prever corretamente 3 imagens com MOS razoável os demais não conseguiram.

Tabela 4.26 – Matrizes de Confusão da validação da base de dados or129.

J48					RdnF				
a	b	c	d	<- classificou como	a	b	c	d	<- classificou como
82	7	0	0	a = péssimo	84	4	0	1	a = péssimo
39	7	3	3	b = ruim	42	3	3	4	b = ruim
0	0	0	5	c = razoável	0	0	0	5	c = razoável
0	0	0	126	d = bom	0	0	0	126	d = bom

[a] Matriz de Confusão para J48

[b] Matriz de Confusão para RdnF

IBk				
a	b	c	d	<- classificou como
82	7	0	0	a = péssimo
40	4	2	6	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[c] Matriz de Confusão para IBk

RdnT				
a	b	c	d	<- classificou como
85	4	0	0	a = péssimo
45	2	2	3	b = ruim
0	0	0	5	c = razoável
124	0	0	2	d = bom

[d] Matriz de Confusão para RdnT

NaiveBayes				
a	b	c	d	<- classificou como
82	7	0	0	a = péssimo
44	9	2	4	b = ruim
0	0	3	2	c = razoável
0	0	126	0	d = bom

[e] Matriz de Confusão para NaiveBayes

JRip				
a	b	c	d	<- classificou como
86	3	0	0	a = péssimo
50	0	2	0	b = ruim
5	0	0	0	c = razoável
126	0	0	0	d = bom

[f] Matriz de Confusão para JRip

OneR				
a	b	c	d	<- classificou como
85	0	1	3	a = péssimo
39	0	5	8	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[g] Matriz de Confusão para OneR

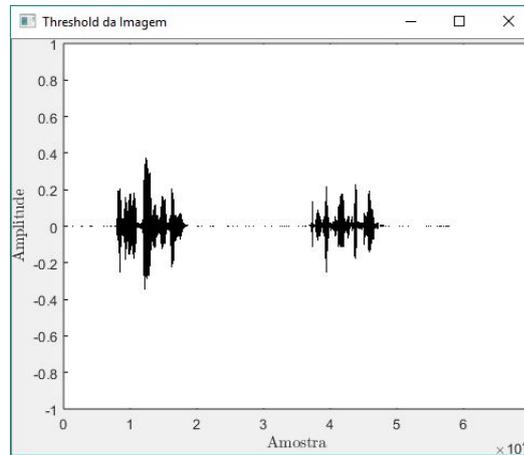
MLP				
a	b	c	d	<- classificou como
82	0	0	7	a = péssimo
39	0	0	13	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[h] Matriz de Confusão para MLP

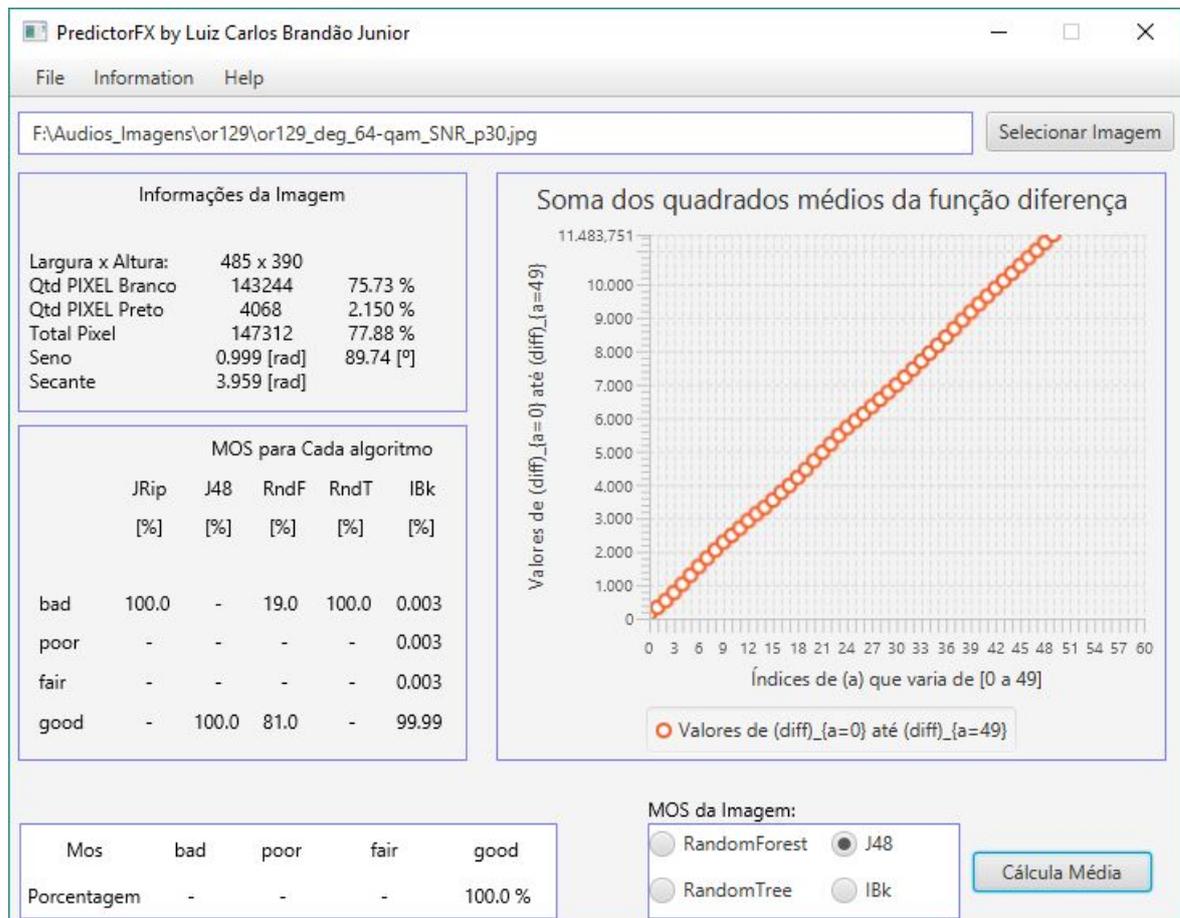
Fonte: do autor

A Figura 4.22 mostra a imagem or129\_deg\_64-qam\_SNR\_p30. Essa imagem representa o arquivo desvanecido da base de dados or129 com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 64-QAM e a relação sinal-ruído de 30dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de 4,5 e devido à Tabela 3.3 recebeu a classificação de um MOS bom. O algoritmo treinado J48 tem 100% de certeza de que a imagem tem um MOS caracterizado como bom, RdnF tem 81% de certeza e o IBk 99,99% de ser bom. O RdnT classificou a imagem como ruim. O ângulo alfa foi de  $89,74^\circ$  graus e a secante de 3,959 radianos.

Figura 4.22 – Predição da imagem or129 com SNR de 30dB e modulação 64-QAM para ATC1



[a] *Threshold* da imagem or129 com SNR de 30dB e modulação 64-QAM



[b] *Software PredictorFX*

A Tabela 4.27 mostra o resultado da validação do modelo para as degradações do arquivo original or134. Percebe-se que os algoritmos treinados RndT e JRip tiveram estatística Kappa menor que zero indicando insignificância desses algoritmos para esse tipo de problema. Os

algoritmos J48 e RdnF foram os que mais classificaram corretamente as instâncias. O OneR teve o menor EAM e o J48 o segundo, por outro lado o J48 teve o menor EQM na comparação com os demais.

Tabela 4.27 – Validação do modelo para os arquivos degradados or134\_c1

	ICC	IIC	EKP	EAM	EQM	NTI
J48	164 (60,2941%)	108 (39,7059%)	0,4231	0,1969	0,4384	272
RdnF	164 (60,2941%)	108 (39,7059%)	0,4226	0,2405	0,4452	272
IBk	163 (59,9265%)	109 (40,0735%)	0,4200	0,2005	0,4447	272
RdnT	39 (14,3382%)	233 (85,6618%)	-0,0260	0,4283	0,6545	272
NaiveBayes	44 (16,1765%)	228 (83,8235%)	0,0610	0,4191	0,6468	272
JRip	39 (14,3382%)	233 (85,6618%)	-0,0136	0,4270	0,6503	272
OneR	165 (60,6618%)	107 (39,3382%)	0,4279	0,1967	0,4435	272
MLP	160 (58,8235%)	112 (41,1765%)	0,3896	0,2361	0,4392	272

Fonte: do autor

O banco de dados com as imagens degradadas do arquivo original or134 contém 41 imagens que representam o MOS péssimo, 100 com MOS ruim, 5 com MOS razoável e 126 com MOS bom. A Tabela 4.28 mostra as matrizes de confusão e percebe-se que o algoritmo OneR foi o que mais conseguiu predizer corretamente as imagens com MOS péssimo, o menor foi NaiveBayes com 33. Imagens com MOS ruim apenas OneR e o MLP não conseguiram predizer nenhuma imagem, e o NaiveBayes foi o que mais acertou, com 7. Só o NaiveBayes conseguiu predizer 4 imagens com MOS razoável. E com MOS bom nem o NaiveBayes nem o JRip conseguiram predizer, o RdnT conseguiu apenas uma, e os demais a sua totalidade.

Tabela 4.28 – Matrizes de Confusão da validação da base de dados or134.

J48					RdnF				
a	b	c	d	<- classificou como	a	b	c	d	<- classificou como
35	4	1	1	a = péssimo	37	2	0	2	a = péssimo
89	3	3	5	b = ruim	92	1	1	6	b = ruim
0	0	0	5	c = razoável	0	0	0	5	c = razoável
0	0	0	126	d = bom	0	0	0	126	d = bom

[a] Matriz de Confusão para J48

[b] Matriz de Confusão para RdnF

IBk				
a	b	c	d	<- classificou como
36	4	1	0	a = péssimo
91	1	3	5	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[c] Matriz de Confusão para IBk

RdnT				
a	b	c	d	<- classificou como
37	3	0	1	a = péssimo
92	1	3	4	b = ruim
0	0	0	5	c = razoável
125	0	0	1	d = bom

[d] Matriz de Confusão para RdnT

NaiveBayes				
a	b	c	d	<- classificou como
33	7	0	1	a = péssimo
86	7	1	6	b = ruim
0	0	4	1	c = razoável
0	0	126	0	d = bom

[e] Matriz de Confusão para NaiveBayes

JRip				
a	b	c	d	<- classificou como
36	4	1	0	a = péssimo
95	3	2	0	b = ruim
5	0	0	0	c = razoável
126	0	0	0	d = bom

[f] Matriz de Confusão para JRip

OneR				
a	b	c	d	<- classificou como
39	0	1	1	a = péssimo
88	0	2	10	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[g] Matriz de Confusão para OneR

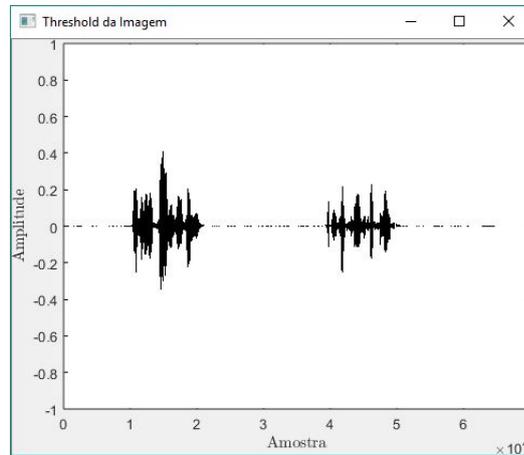
MLP				
a	b	c	d	<- classificou como
34	0	0	7	a = péssimo
86	0	0	14	b = ruim
0	0	0	5	c = razoável
0	0	0	126	d = bom

[h] Matriz de Confusão para MLP

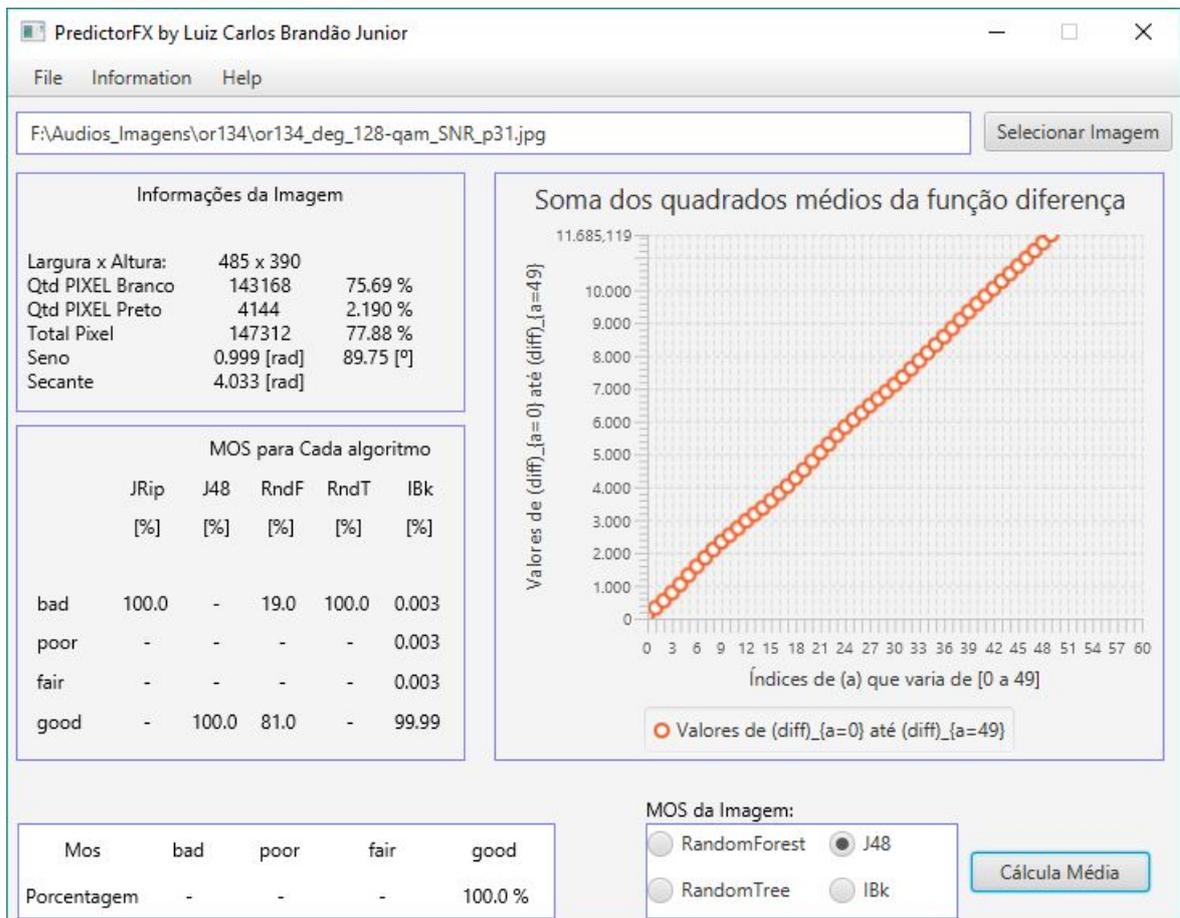
Fonte: do autor

A Figura 4.23 mostra a imagem or134\_deg\_128-qam\_SNR\_p31. Essa figura representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 128-QAM e a relação sinal-ruído de 31dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de 4,5 e devido à Tabela 3.3 recebeu a classificação de MOS bom. O algoritmo treinado J48 tem 100% de certeza de que a imagem tem um MOS caracterizado como bom, RdnF tem 81% de certeza e o IBk 99,99%. O RdnT tornou a classificar a imagem com um MOS péssimo. O ângulo alfa foi de  $89,75^\circ$  graus e a secante de 4,033 radianos.

Figura 4.23 – Predição da imagem or134 com SNR de 31dB e modulação 128-QAM para ATC1.



[a] Threshold da imagem or134 com SNR de 31dB e modulação 128-QAM.



[b] Software PredictorFX.

A Tabela 4.29 mostra o resultado da validação do modelo para as degradações do arquivo original or137. Percebe-se que o algoritmo OneR teve o menor EAM e o maior acerto para as instâncias corretamente classificadas com cerca de 84,19%, um EKP maior que os demais e

apesar de ter um EQM cerca de 3,27% maior seu EAM é 2,11% menor do que o J48. Os algoritmos IBk, RdnT e JRip tiveram um EKP menor que zero indicando que os dados previstos tiveram discordância com os valores contidos nesse banco de dados. Apesar do algoritmo J48 ter menor instâncias corretamente classificadas em relação ao RdnF, ele teve os EAM e EQM menores se comparado com o RdnF.

Tabela 4.29 – Validação do modelo para os arquivos degradados or137\_c1

	ICC	IIC	EKP	EAM	EQM	NTI
J48	226 (83,0882%)	46 (16,9118%)	0,7120	0,0807	0,2722	272
RdnF	227 (83,4559%)	45 (16,5441%)	0,7141	0,1155	0,2787	272
IBk	104 (38,2353%)	168 (61,7647%)	-0,0101	0,3098	0,5520	272
RdnT	103 (37,8676%)	169 (62,1324%)	-0,0163	0,3107	0,5574	272
NaiveBayes	104 (38,2353%)	168 (61,7647%)	0,2222	0,3088	0,5557	272
JRip	100 (36,7647%)	172 (63,2353%)	-0,0228	0,3136	0,5560	272
OneR	229 (84,1912%)	43 (15,8088%)	0,7239	0,0790	0,2811	272
MLP	225 (82,7206%)	47 (17,2794%)	0,6950	0,1197	0,2795	272

Fonte: do autor

O banco de dados com as imagens degradadas do arquivo original or137 contém 107 imagens que representam o MOS péssimo, 34 com MOS ruim, 4 com MOS razoável e 127 com MOS bom. A Tabela 4.30 mostra as matrizes de confusão e percebe-se que o OneR foi o que teve maior acerto nas imagens com MOS ruim com 102 delas classificadas corretamente. IBk e RdnT classificaram 101 delas como péssimo e RdnF somente 100 delas. o J48 classificou somente 3 como ruim, IBk somente 1, JRip 2 e o restante nenhuma. Somente o NaiveBayes classificou corretamente a totalidade como razoável o restante não conseguiu nenhuma classificação correta. Somente os algoritmos OneR, MLP, RdnF e J48 classificaram corretamente todas as imagens com MOS bom, o IBk somente duas, assim como RdnT, NaiveBayes somente 1 e JRip nenhuma.

Tabela 4.30 – Matrizes de Confusão da validação da base de dados or137.

J48					RdnF				
a	b	c	d	<- classificou como	a	b	c	d	<- classificou como
96	9	2	0	a = péssimo	100	5	2	0	a = péssimo
24	3	2	5	b = ruim	27	0	1	6	b = ruim
0	0	0	4	c = razoável	0	0	0	4	c = razoável
0	0	0	127	d = bom	0	0	0	127	d = bom

[a] Matriz de Confusão para J48

[b] Matriz de Confusão para RdnF

IBk				
a	b	c	d	<- classificou como
101	3	3	0	a = péssimo
26	1	0	7	b = ruim
1	0	0	3	c = razoável
125	0	0	2	d = bom

[c] Matriz de Confusão para IBk

RdnT				
a	b	c	d	<- classificou como
101	5	1	0	a = péssimo
27	0	1	6	b = ruim
1	0	0	3	c = razoável
125	0	0	2	d = bom

[d] Matriz de Confusão para RdnT

NaiveBayes				
a	b	c	d	<- classificou como
99	6	0	2	a = péssimo
27	0	2	5	b = ruim
0	0	4	0	c = razoável
0	0	126	1	d = bom

[e] Matriz de Confusão para NaiveBayes

JRip				
a	b	c	d	<- classificou como
98	7	2	0	a = péssimo
32	2	0	0	b = ruim
4	0	0	0	c = razoável
127	0	0	0	d = bom

[f] Matriz de Confusão para JRip

OneR				
a	b	c	d	<- classificou como
102	0	3	2	a = péssimo
26	0	0	8	b = ruim
0	0	0	4	c = razoável
0	0	0	127	d = bom

[g] Matriz de Confusão para OneR

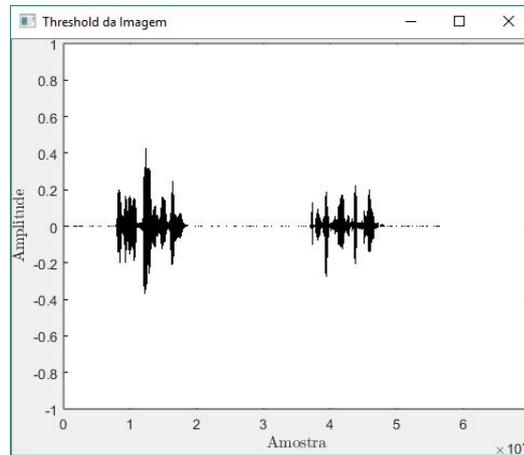
MLP				
a	b	c	d	<- classificou como
98	0	0	9	a = péssimo
26	0	0	8	b = ruim
0	0	0	4	c = razoável
0	0	0	127	d = bom

[h] Matriz de Confusão para MLP

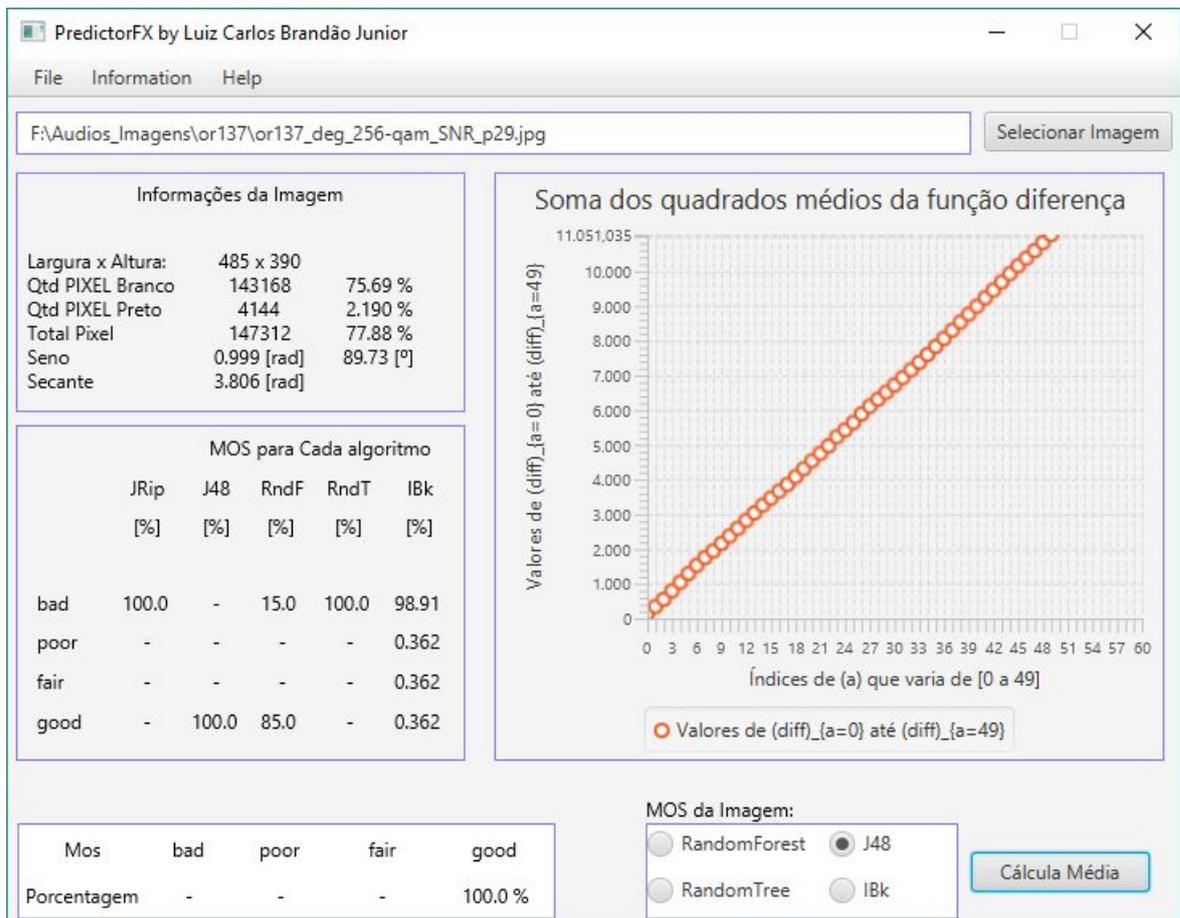
Fonte: do autor

A Figura 4.24 mostra à imagem `or137_deg_256-qam_SNR_p29` que representa o arquivo desvanecido com o nível de quantização igual a  $2^{16} \rightarrow 65536$ , modulação 256-QAM e a relação sinal-ruído de 29dB, cujos parâmetros no Passo 1 gerou um MOS com o valor de 4,4 e devido à Tabela 3.3 recebeu a classificação de bom. O algoritmo treinado J48 tem 100% de certeza de que a imagem tem um MOS caracterizado como bom, RdnF tem 85% de certeza e o IBk 98,91% de ser péssimo. O RdnT tornou a classificar à imagem carregada como péssimo. O ângulo alfa foi de 89,73° graus e a secante foi de 3,806 radianos.

Figura 4.24 – Predição da imagem or137 com SNR de 29dB e modulação 256-QAM para ATC1.



[a] *Threshold* da imagem or137 com SNR de 29dB e modulação 256-QAM.

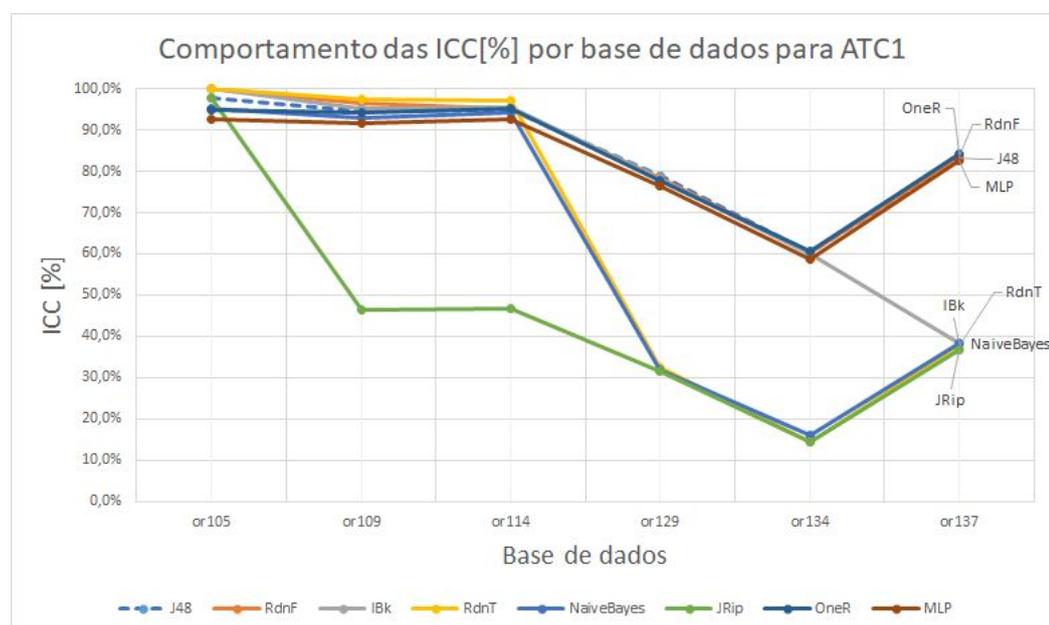


[b] *Software PredictorFX.*

O gráfico da Figura 4.25 mostra o comportamento das Instâncias Corretamente Classificadas de toda a base de dados, ou seja, das 1632 imagens. Nota-se que os algoritmos treinados que tiveram melhor rendimento foram J48, OneR, RdnF e MLP. Esses algoritmos se manti-

veram quase iguais nas bases or105, or109 e or114, na base or129 ficaram em torno de 80% das instâncias corretamente classificadas, na base or134 ficaram em torno de 60% e na última base, a or137 ficaram acima de 80% das ICC, isso indica que tais algoritmos conseguiram prever corretamente as instâncias classificadas no processo de validação ficando entre 58,82% e 96,32%. Os demais algoritmos foram ineficientes para esse tipo de problema, ficando abaixo dos 46,69% das instâncias corretamente classificadas.

Figura 4.25 – Comportamento das ICC para ATC1.



Fonte: do Autor

A Tabela 3.3 gera a Tabela 4.31 que representa a quantidade de imagens por intervalo, percebe-se ao analisar a queda das ICC dos algoritmos OneR, RdnF, J48 e MLP, para a base de dados or129 e or134, e a ligeira alta na base or137, que essa alternância de valores pode estar relacionada a quantidade de imagens que divergiram ao do modelo or105, para MOS Péssimo e Ruim. Gerando assim, essa queda brusca na base de dados or129 e or134, com uma ligeira alta na base or137. As Funções Discretas podem ser visualizadas a partir do APÊNDICE A.

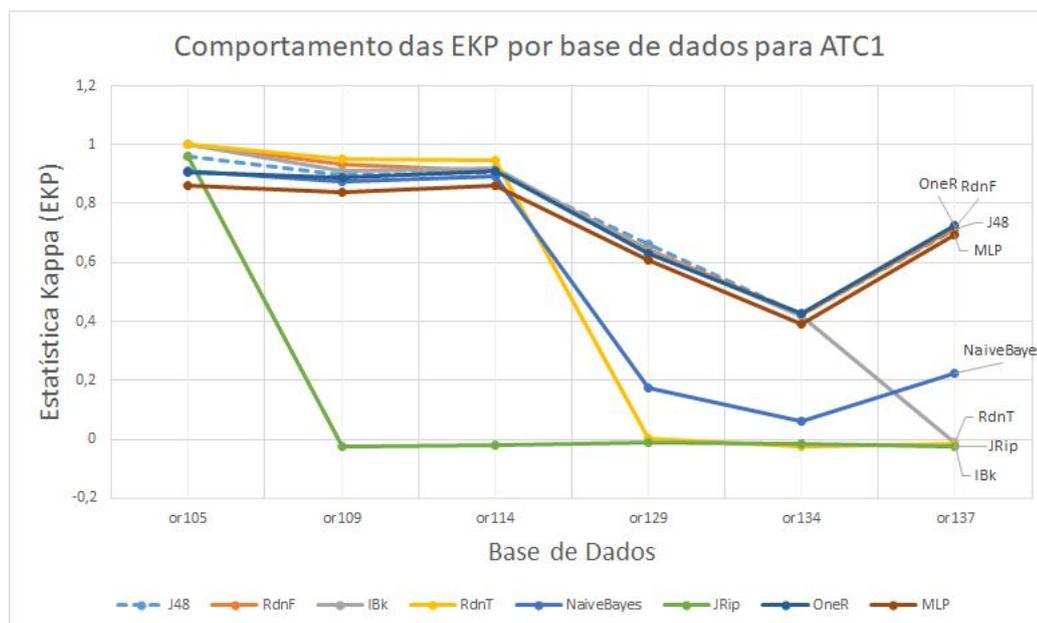
Tabela 4.31 – Quantidade de imagens por intervalo.

	or105	or109	or114	or129	or134	or137
Péssimo	133	133	133	89	41	107
Ruim	7	8	8	52	100	34
Razoável	6	5	4	5	5	4
Bom	126	126	127	126	126	127

Fonte: do autor

O gráfico da Figura 4.26 demonstra o comportamento dos coeficientes Kappa para as diversas base de dados e percebe-se que os algoritmos OneR, RdnF, J48 e MLP foram os que se mantiveram sempre com valores mais altos para a validação variando entre 0,9322 de máxima a 0,3896 de mínima, em alguns pontos há quase uma sobreposição de valores do OneR com J48.

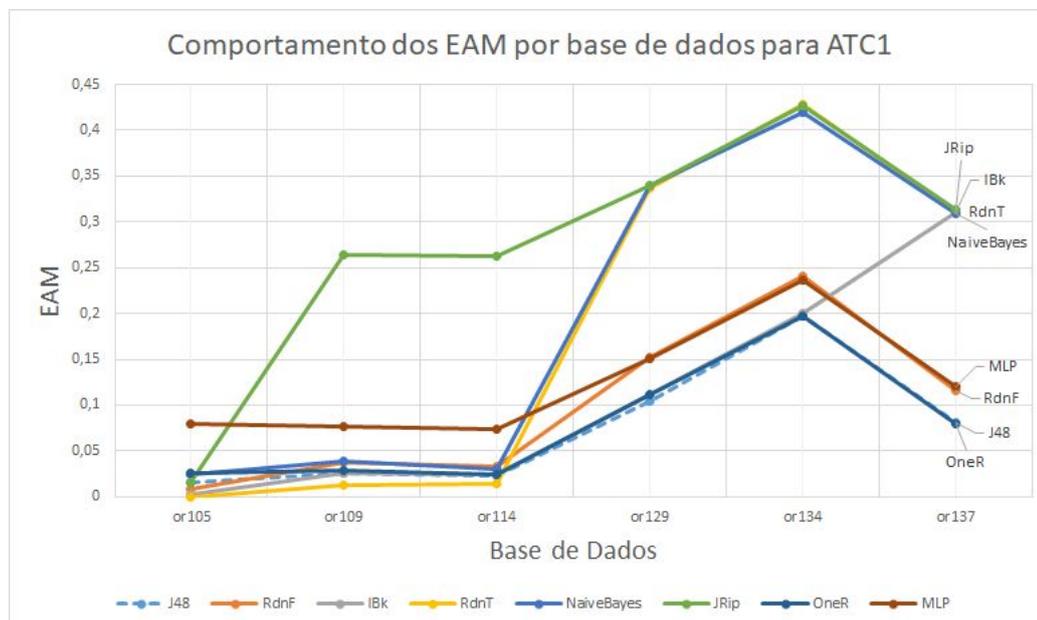
Figura 4.26 – Comportamento das EKP para ATC1.



Fonte: do Autor

O gráfico da Figura 4.27 demonstra o comportamento dos EAM ao longo das bases de dados e percebe-se que dois algoritmos possuem seus erros mais baixos a partir da base de dados or109, sendo eles o J48 e OneR com EAM variando entre 0,1969 de máxima e 0,0236 de mínima.

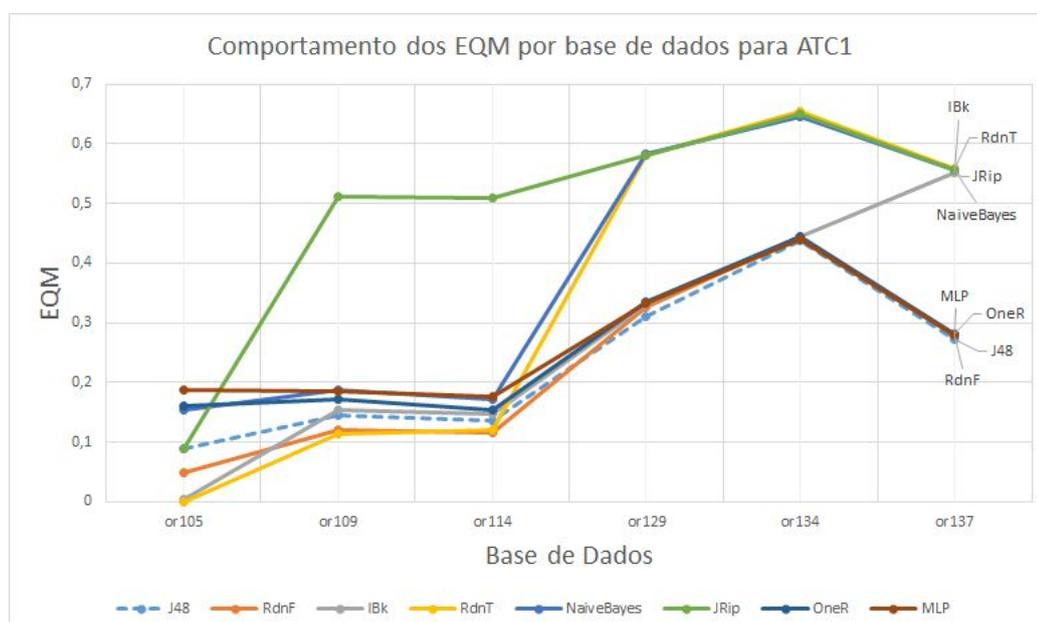
Figura 4.27 – Comportamento dos EAM para ATC1.



Fonte: do Autor

O gráfico da Figura 4.28 demonstra o comportamento dos EQM e a partir da base de dados or109 onde inicia o processo de validação dos algoritmos treinados, percebe-se que os algoritmos MLP, OneR, J48 e RdnF mantêm seus valores sempre entre 0,4452 de máxima e 0,1159 de mínima. Nota-se, nos demais que sempre em uma base há um escape atingindo erros maiores.

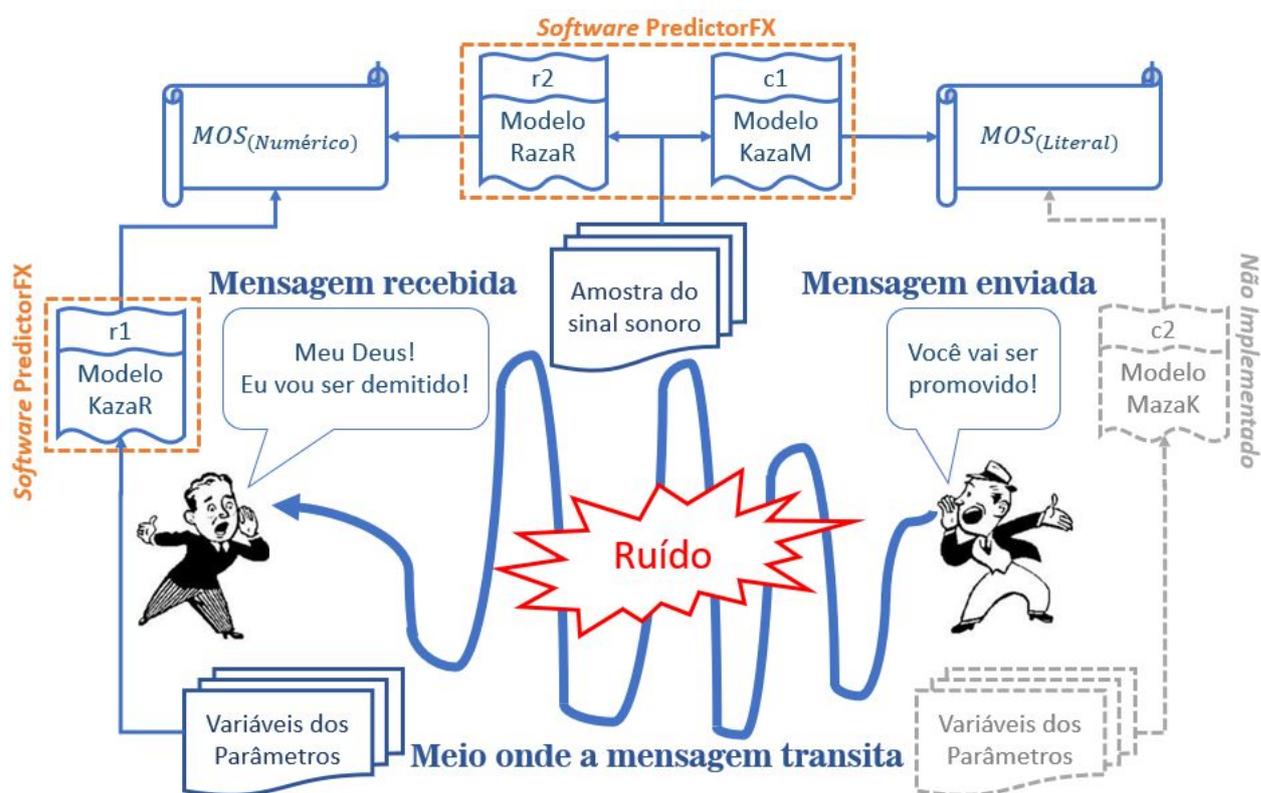
Figura 4.28 – Comportamento dos EQM para ATC1.



Fonte: do Autor

A Figura 4.29 mostra um apanhado de como os modelos construídos são usados para medir o  $MOS_{(Numérico)}$  usando o Modelo RazaR ou o  $MOS_{(Literal)}$  usando o Modelo KazaM, mediante uma amostra do sinal sonoro no domínio do tempo. E por meio da variação dos parâmetros tendo como resultado o  $MOS_{(Numérico)}$  com o Modelo KazaR e para trabalhos futuros o Modelo MazaK usando lógica *Fuzzy*.

Figura 4.29 – Modelos KazaR, RazaR, KazaM e MazaK.



Fonte: do Autor

## 5 CONCLUSÃO

No presente trabalho se demonstrou que é possível determinar a qualidade do sinal de voz utilizando as características no domínio do tempo do mesmo sinal e técnicas de aprendizado de máquina. Diferentes níveis de degradação foram inseridos em canal de transmissão *wireless* para obter diferentes áudios degradados, em um total de 272. Tais áudios foram avaliados com a métrica não intrusiva ITU-T P.862 gerando assim um valor de MOS para cada arquivo de áudio. Os áudios degradados foram plotados e salvos em arquivos com extensão jpg, posteriormente cada imagem foi escaneada determinando assim uma função discreta que melhor representasse a imagem, essa função juntamente com o valor do MOS respectivo de cada imagem foram analisadas por algoritmos de aprendizagem de máquina, os resultados demonstraram que o desempenho da métrica proposta teve resultados satisfatórios.

Foi possível medir a qualidade do sinal de maneira não intrusiva em representações de áudios no domínio do tempo usando para isso um modelo fundamentado na recomendação ITU-T P.862. Esse modelo proposto visou determinar a qualidade do sinal de voz nas 1360 imagens que em nenhuma das vezes foram usadas para qualquer tipo de treinamento e o modelo se mostrou satisfatório, uma vez que, usando apenas imagens no domínio do tempo e técnicas de aprendizado de máquina já são suficientes para fazer o teste da qualidade. Ou seja, treina-se somente uma única vez os algoritmos em um banco de dados e aplica esse algoritmo treinado nas demais bases de dados que se queira, além do processo ser rápido para fazer o teste nas demais imagens ele é confiável, uma vez que, antes do treinamento há a extração das características de cada imagem que representa o áudio degradado e a construção da função discreta que por sua vez representa aquela imagem, só depois disso que há o treinamento. Se o intuito é verificar a qualidade do sinal, nos demais arquivos sujeitos às mesmas condições, então os algoritmos suprem essa necessidade pois uma vez treinados eles serão capazes de encontrar o mesmo comportamento em áudios distintos, bastando treinar os algoritmos uma única vez e podendo usar infinitas vezes de maneira rápida e confiável.

Percebe-se tanto em ATR2 quanto no ATC1 onde houve o escaneamento das imagens de áudio no domínio do tempo que a acurácia se manteve alta, ou seja, as características extraídas na base de dados or105 devido às permutações dos parâmetros foram reconhecidas em outros arquivos que mesmo sendo de bases diferentes os algoritmos foram capazes de reconhecê-los independentemente das bases que estavam. Gerando assim dois modelos, o Modelo RazaR para identificar o  $MOS_{(Numérico)}$  e o Modelo KazaM para identificar o  $MOS_{(Literal)}$  das amostras ca-

pazes de reconhecer o comportamento igual ou similar das funções discretas após o escaneamento de cada amostra.

Após passar pelas várias etapas para tomar conhecimento de quais *classifiers* seriam melhores para compor o *software* PredictorFX, o resultado se mostrou satisfatório para ATR1. Ao invés de se trabalhar com apenas um deles pelo teste de hipótese provou que não há diferença estatística significativa entre o melhor ranqueado e o segundo ou até o alcance da distância crítica, podendo se usar uma combinação deles para melhor resultado. Para ATR1 os algoritmos que melhor se adaptaram foram sem dúvida o IBk, *RandomForest* e *RandomTree* com coeficientes de correlação altíssimos, erros absolutos médios pequenos em relação aos outros e também por terem os menores erros quadráticos médios indicando que tais algoritmos treinados tiveram uma boa aceitação para esse tipo de problema.

Para ATR2 dois dos melhores algoritmos ranqueados pelo teste de hipótese tiveram resultados satisfatórios, sendo eles RdnT e RdnF. Outros dois também se destacaram sendo eles o M5P e MLP. Os quatro tiveram coeficientes de correlação altos variando entre 0,8638 a 0,9896 a partir da base de dados or109 até or137. Seus erros absolutos médios ficaram baixos na comparação com os demais e o que teve menor EAM foi o RdnT, foi esse algoritmo também que teve menor erro quadrático médio seguido do M5P, MLP e RdnF. Os valores dos Coeficientes de Correlação dos algoritmos treinados IBk e AddR foram negativos para a base de dados or137 indicando que os áudios têm grandezas que são inversamente proporcionais, descobrir o porquê disso teria que ter um pouco mais de investigação, uma vez que, de acordo com a Tabela 3.1 uma das poucas informações que varia é o número total de amostras que é a menor que as demais.

Para ATC1 os algoritmos OneR, RdnF, J48 e MLP foram os que tiveram maiores acertos ao prever os valores das diferentes bases de dados, cujo comportamento ao longo das bases se perpetuou para os coeficientes da estatística Kappa, em alguns casos quase tendo uma sobreposição de valores. Quando se analisou o EAM percebeu-se que o OneR e o J48 tiveram erros menores ocorrendo em alguns pontos quase uma sobreposição de valores, em relação ao erro quadrático médio há de novo uma quase sobreposição de valores sendo os quatro algoritmos com erros menores do que os demais.

O *software* PredictorFX foi capaz de carregar e escanear as imagens que representam os vários arquivos degradados e por meio dos modelos gerados pelos algoritmos de treinamento foi capaz de prever o valor do MOS para aquelas imagens.

## 6 TRABALHOS FUTUROS

Analisando a Tabela 6.1 percebe-se que somente os algoritmos RdnT e MLP que ajudam na comparação dos modelos KazaR, RazaR e KazaM, onde I01 é a informação do Escaneamento com 100 iterações e aprendizado básico, I02 é o Tamanho da árvore dado em quantidade de nó, I03 é a quantidade de vizinhos mais próximos. I04 é a Quantidade de Nó Sigmoid, I05 é a Número de regras. I06 é o Número 37128 que corresponde a quantidade de avaliações do Kernel dado em porcentagem (*cached*), I07 é a Quantidade de modelos, I08 é a Equação para r1, I09 é a Equação para r2. I10 é a Quantidade de folhas e I11 é a informação de que o algoritmo fornece uma porcentagem para cada intervalo.

Tabela 6.1 – Comparação das informações dos algoritmos para or105\_r1.

	r1	r2	c1
RndF	I01: 100	I01: 100	I01: 100
RndT	I02: 141	I02: 101	I02: 31
IBk	I03: 1	I03: 1	I03: 1
MLP	I04: 2	I04: 25	I04: 30
M5P	I05: 18	I05: 2	-
SMOreg	I06: 94,226%	I06: 99,456%	-
AddR	I07: 9	I07: 9	-
SLR	I08	I09	-
J48	-	-	I10: 5 I02: 9
OneR	-	-	I05: 3
JRip	-	-	I05: 4
NvBy	-	-	I11

Fonte: do autor

No algoritmo RdnT o tamanho da árvore (I02) para KazaM é menor se comparado aos modelos KazaR e RazaR, uma sugestão seria, caso se utilize os algoritmos de classificação, talvez aumentar a quantidade de Intervalos da Tabela 3.3 para c1, deste modo aumentando o tamanho da árvore como ocorre em r1 e r2.

Uma outra sugestão seria ao invés de aumentar a quantidade de intervalos, seria diminuí-lo, uma vez que, teve somente 7 imagens com MOS ruim e 6 com MOS razoável, podendo juntar em uma única classificação. E em vez de usar os algoritmos de classificação, poderia usar somente os de regressão para avaliar a imagem carregada fornecendo o  $MOS_{(Numérico)}$  e só depois disso fornecer o  $MOS_{(Literal)}$  sem usar qualquer algoritmo de classificação.

Neste trabalho foram usadas imagens cujas dimensões foram de 485 pixel de largura por 390 pixel altura, com área real de plotagem (ARP) do sinal sonoro ficando limitado como de-

mostrado na Equação 6.1 onde os limites na plotagem das figuras no *Matlab* ficaram limitadas a  $[-1 \ 1]$  para o eixo y e de  $[0 \ 7e4]$  no eixo x, uma vez que, pela Tabela 3.1 a quantidade de amostra total é dado pelo áudio or114 com 68734 amostras. E como nenhum áudio excedia a altura de -1 e 1, mas ficava condicionada a variação da altura máxima de cada áudio, convencionou-se usar limites para os eixos, deste modo garantindo uma área de mesma dimensão para todos os áudios. Caso precisa-se mudar utilizando outros áudios, basta observar os limites do novo áudio que se queira usar, delimitar os eixos, plotar os gráficos, salvar e antes de escanear a imagem pelo código (P2F1) mudar os valores do Pixel L e J, de modo que comece o escaneamento dentro do campo de plotagem dos sinais sonoros (ARP), uma vez que, as equações são dinâmicas, não precisa ser feito qualquer tipo de modificação nelas porque estão condicionadas ao tamanho do  $\text{Vetor}_{(Campo)}$  que por sua vez está limitado as dimensões da ARP, que começa no Pixel L e vai até o Pixel J. Em outras palavras, o  $\text{Vetor}_{(Campo)}$  é preenchido independentemente do tamanho da imagem.

$$ARP = (480 - 49 \ x \ 345 - 5) \rightarrow (431 \ x \ 340) \quad (6.1)$$

Um outro trabalho futuro é a implementação do Modelo MazaK usando lógica *Fuzzy* onde por meio das variáveis dos parâmetros ser capaz de fornecer o  $\text{MOS}_{(Literal)}$ . Ou até usar uma amostra do sinal sonoro para adquirir o MOS.

## REFERÊNCIAS

- AFFONSO, E. T. F.; RODRÍGUEZ, D. Z. Impacto do desvanecimento do sinal na qualidade de uma comunicação voip. **XXXIV SIMPÓSIO BRASILEIRO DE TELECOMUNICAÇÕES, SBrT2016, 30 de Agosto a 2 de Setembro, Santarém-PA**, p. 578–582, AGOSTO 2016. Acessado em: 19/6/2018. Disponível em: <<http://www.sbrt.org.br/sbrt2016/anais/ST19/1570276870.pdf>>.
- AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. **Machine Learning**, v. 6, n. 1, p. 37–66, Jan 1991. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1007/BF00153759>>.
- ALI, I.; SCHOLAR, M. T. **Bit-Error-Rate (BER) Simulation Using MATLAB**. 2018. Acessado em: 31/12/2018. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.415.4700>>.
- ANATEL, A. N. de T. **Qualidade - Telefonia Fixa**. [S.l.], 2018. Acessado em: 25/06/2019. Disponível em: <<http://www.anatel.gov.br/dados/controle-de-qualidade/telefonia-fixa-controle>>.
- ANATEL, A. N. de T. **Radiofrequência**. [S.l.], 2018. Acessado em: 01/08/2018. Disponível em: <<http://www.anatel.gov.br/setorregulado/coordenacao-nacional>>.
- ANNEX, T. D. S. I.-T. P. R. **SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS. Methods for objective and subjective assessment of quality**. [S.l.], 2005. Acessado em: 31/1/2018. Disponível em: <<https://www.itu.int/rec/T-REC-P.862-200511-I!Amd2>>.
- BREIMAN, L. Random forests. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 45, n. 1, p. 5–32, out. 2001. ISSN 0885-6125. Disponível em: <<https://doi.org/10.1023/A:1010933404324>>.
- CHEN, C. et al. An imaging system for monitoring the in-and-out activity of honey bees. **Computers and Electronics in Agriculture**, v. 89, p. 100 – 109, 2012. ISSN 0168-1699. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0168169912002074>>.
- CISCO, W. P. **Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2017–2022**. [S.l.], 2019. Acessado em: 25/06/2019. Disponível em: <<https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-738429.pdf>>.
- COHEN, W. W. Fast effective rule induction. In: PRIEDITIS, A.; RUSSELL, S. (Ed.). **Machine Learning Proceedings 1995**. San Francisco (CA): Morgan Kaufmann, 1995. p. 115 – 123. ISBN 978-1-55860-377-6. Disponível em: <<http://www.sciencedirect.com/science/article/pii/B9781558603776500232>>.
- COMER, D. **Redes de Computadores e Internet - 6.ed**. Bookman Editora, 2016. Acessado em: 20/6/2019. Disponível em: <<https://books.google.com.br/books?id=1nwdDAAAQBAJ>>.
- DIGERATI, E. **Guia técnico de redes de computadores**. Universo dos Livros Editora, 2009. Acessado em: 19/6/2019. Disponível em: <<https://books.google.com.br/books?id=O1jslYzWZq8C>>.

ENTELCO. **SNR – O que é ? Como funciona ?** [S.l.], 2015. Acesso em: 4/4/2019.

FRENZEL, L. **Fundamentos de Comunicação Eletrônica - Volume 2**. McGraw Hill Brasil, 2009. Acessado em: 31/01/2018. Disponível em: <<https://books.google.com.br/books?id=jUZSpEkHIE8C>>.

FRIEDMAN, J. H. Stochastic gradient boosting. **Computational Statistics & Data Analysis**, v. 38, n. 4, p. 367 – 378, 2002. ISSN 0167-9473. Nonlinear Methods and Data Mining. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167947301000652>>.

FRIEDMAN, M. A comparison of alternative tests of significance for the problem of  $m$  rankings. **Ann. Math. Statist.**, The Institute of Mathematical Statistics, v. 11, n. 1, p. 86–92, mar 1940. Disponível em: <<https://doi.org/10.1214/aoms/1177731944>>.

HALL, M. et al. The weka data mining software: an update. **ACM SIGKDD explorations newsletter**, ACM, v. 11, n. 1, p. 10–18, 2009.

HAYKIN, S. **Communication Systems, 3Rd Ed**. [S.l.]: Wiley India Pvt. Limited, 2008. ISBN 9788126513666.

HAYKIN, S.; MOHER, M. **Sistemas modernos de comunicações wireless**. Bookman, 2009. Acessado em: 1 de maio de 2019. Disponível em: <<https://books.google.com.br/books?id=3v4hZYqOiJUC>>.

HAYKIN, S.; MOHER, M. **Sistemas de Comunicação - 5.ed**. [S.l.]: Bookman Editora, 2011.

HOLTE, R. C. Very simple classification rules perform well on most commonly used datasets. **Machine Learning**, v. 11, n. 1, p. 63–90, Apr 1993. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1023/A:1022631118932>>.

HU, Y.; LOUIZOU, P. C. Evaluation of objective quality measures for speech enhancement. **IEEE Transactions on Audio, Speech, and Language Processing**, v. 16, n. 1, p. 229–238, Jan 2008. ISSN 1558-7916.

ITU. **Quality of Service Regulation Manual**. ITU Telecom Publications, 2017. 162 p. Acessado em: 1/1/2019. Disponível em: <<https://www.itu-ilibrary.org/content/publication/pub-8108e11f-en>>.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. In: **Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (UAI'95), p. 338–345. ISBN 1-55860-385-9. Disponível em: <<http://dl.acm.org/citation.cfm?id=2074158.2074196>>.

MATLAB. **Fading Channels**. 2018. Disponível em: <<http://mathworks.com/help/comm/ug/fadingchannels.html>>. Acesso em: 13 novembro 2018.

MIYOSHI, E.; SANCHEZ, C. **Projetos de sistemas rádio**. [S.l.]: Érica, 2002. ISBN 9788571948686.

NEMENYI, P. **Distribution-free Multiple Comparisons**. [S.l.]: Princeton University, 1963.

OLIVIERO, A.; WOODWARD, B. **Cabling: The Complete Guide to Copper and Fiber-Optic Networking**. Wiley, 2014. Acessado em: 20/6/2019. Disponível em: <<https://books.google.com.br/books?id=BjYKAwAAQBAJ>>.

ÖZCAN, A.; ZIZKA, J.; NAGAMALAI, D. **Recent Trends in Wireless and Mobile Networks: Third International Conferences, WiMo 2011 and CoNeCo 2011, Ankara, Turkey, June 26-28, 2011. Proceedings.** [S.l.]: Springer Berlin Heidelberg, 2011. (Communications in Computer and Information Science).

P.563, I.-T. **Single-ended method for objective speech quality assessment in narrow-band telephony applications.** [S.l.], 2004. Acessado em: 4/6/2019. Disponível em: <<http://handle.itu.int/11.1002/1000/7305>>.

P.800, I.-T. **SERIES P: TELEPHONE TRANSMISSION QUALITY. Methods for objective and subjective assessment of quality.** [S.l.], 1998. Acessado em: 14/6/2018. Disponível em: <<https://www.itu.int/rec/T-REC-P.800-199608-I/en>>.

P.862, T. D. S. I.-T. **SERIES P: TELEPHONE TRANSMISSION QUALITY, TELEPHONE INSTALLATIONS, LOCAL LINE NETWORKS. Methods for objective and subjective assessment of quality.** [S.l.], 2007. Acessado em: 10/6/2019. Disponível em: <<https://www.itu.int/rec/T-REC-P.862/en>>.

PINHEIRO, J. **Guia Completo de Cabeamento de Redes.** Elsevier Brasil, 2017. Acessado em: 26/5/2019. Disponível em: <<https://books.google.com.br/books?id=2JbpCgAAQBAJ>>.

QUINLAN, R. J. Learning with continuous classes. In: **5th Australian Joint Conference on Artificial Intelligence.** Singapore: World Scientific, 1992. p. 343–348.

RAPPAPORT, T. S. **Comunicações sem fio: princípios e práticas.** [S.l.]: PRENTICE HALL BRASIL, 2009.

RODRÍGUEZ, D. Z. **Algoritmo para determinação da taxa de transmissão em uma rede IP.** Tese (Doutorado) — Universidade de São Paulo, 2009.

RODRÍGUEZ, D. Z. et al. Quality metric to assess video streaming service over tcp considering temporal location of pauses. **IEEE Transactions on Consumer Electronics**, IEEE, v. 58, n. 3, p. 985–992, 2012.

RODRÍGUEZ, D. Z.; ROSA, R. L.; BRESSAN, G. Improving a video quality metric with the video content type parameter. **IEEE Latin America Transactions**, IEEE, v. 12, n. 4, p. 740–745, 2014.

ROSS, J. **VoIP – Voz sobre IP.** Antenna Edições Técnicas, 2007. Acessado em: 14/5/2018. Disponível em: <<https://books.google.com.br/books?id=p-oc0YxjDQUC>>.

RUFINO, N. de O. **Segurança em Redes sem Fio – 4ª edição: Aprenda a proteger suas informações em ambientes Wi-Fi e Bluetooth.** NOVATEC, 2005. Acessado em: 3/6/2019. ISBN 9788575224137. Disponível em: <<https://books.google.com.br/books?id=cpmcfUKxp3kC>>.

SALZBERG, S. L. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. **Machine Learning**, v. 16, n. 3, p. 235–240, Sep 1994. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1007/BF00993309>>.

SHEVADE, S. K. et al. Improvements to the smo algorithm for svm regression. **IEEE Transactions on Neural Networks**, v. 11, n. 5, p. 1188–1193, Sep. 2000.

SILVA, V. A. d. **Modelagem computacional de canais de comunicação móvel**. Tese (Doutorado) — Universidade de São Paulo, 2004. Acessado em: 3/5/2018. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/3/3142/tde-19112004-172238/publico/vanderleiaparecido.pdf>>.

SMOLA, A. J.; SCHÖLKOPF, B. A tutorial on support vector regression. **Statistics and Computing**, Kluwer Academic Publishers, Hingham, MA, USA, v. 14, n. 3, p. 199–222, ago. 2004. Disponível em: <<https://doi.org/10.1023/B:STCO.0000035301.49549.88>>.

SOUSA, L. R. N. de. **Avaliação da Qualidade de Voz do Serviço VoIP em Sistemas HSDPA**. Dissertação (Mestrado) — Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza, 2007. Acessado em: 1/1/2019. Disponível em: <[http://repositorio.ufc.br/bitstream/riufc/16076/1/2007\\_dis\\_lrnsousa.pdf](http://repositorio.ufc.br/bitstream/riufc/16076/1/2007_dis_lrnsousa.pdf)>.

SUN, L. **Speech quality prediction for voice over internet protocol networks**. Tese (Doutorado), 2004. Acessado em: 1/1/2019. Disponível em: <<http://hdl.handle.net/10026.1/870>>.

WANG, L.; CHEN, S.; PEDRAM, M. Context-driven power management in cache-enabled base stations using a bayesian neural network. In: **2017 Eighth International Green and Sustainable Computing Conference (IGSC)**. [S.l.: s.n.], 2017. p. 1–8.

WANG, Y.; WITTEN, I. H. Inducing model trees for continuous classes. In: **In Proc. of the 9th European Conf. on Machine Learning Poster Papers**. [S.l.: s.n.], 1997. p. 128–137.

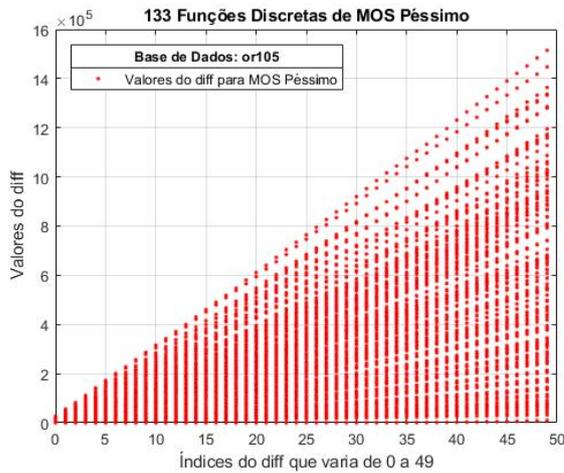
WITTEN, I.; FRANK, E. **Data Mining: Practical Machine Learning Tools and Techniques, Second Edition**. Elsevier Science, 2005. (The Morgan Kaufmann Series in Data Management Systems). Acessado em: 24/12/2018. Disponível em: <<https://books.google.com.br/books?id=QTnOcZJzIUoC>>.

WITTEN, I. et al. **Data Mining: Practical Machine Learning Tools and Techniques**. Elsevier Science, 2016. (The Morgan Kaufmann Series in Data Management Systems). Acessado em: 24/12/2018. Disponível em: <<https://books.google.com.br/books?id=1SylCgAAQBAJ>>.

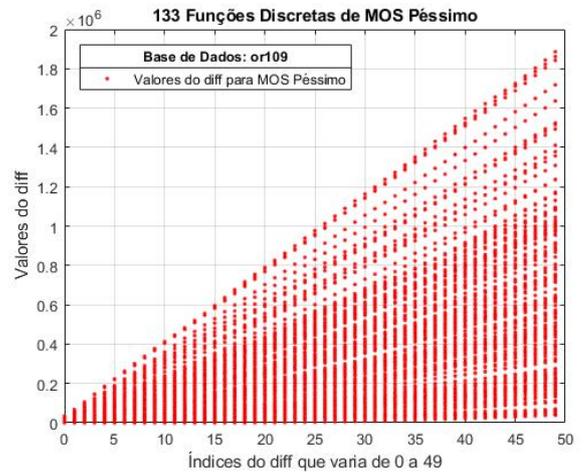
ZANATELI, E. **Engenharia De Sistemas Radiovisibilidade**. [S.l.]: Clube dos Autores (managed), 2007.

## APÊNDICE A – Funções Discretas para o MOS Péssimo

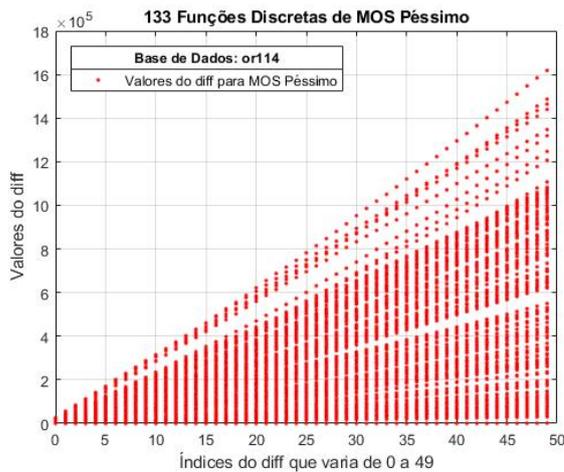
Figura 1 – Funções Discretas(FD) para o MOS Péssimo.



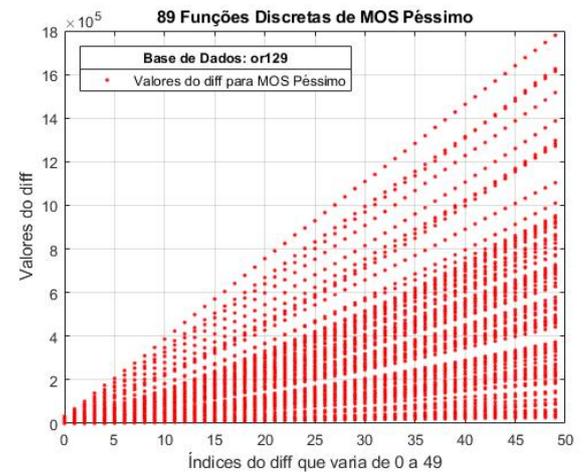
[a] 133 FD para or105 de MOS Péssimo



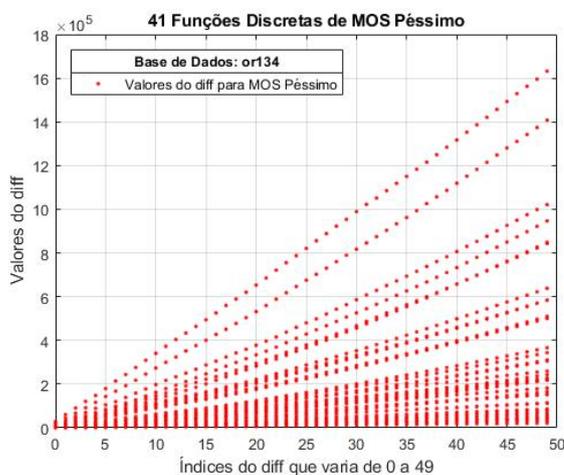
[b] 133 FD para or109 de MOS Péssimo



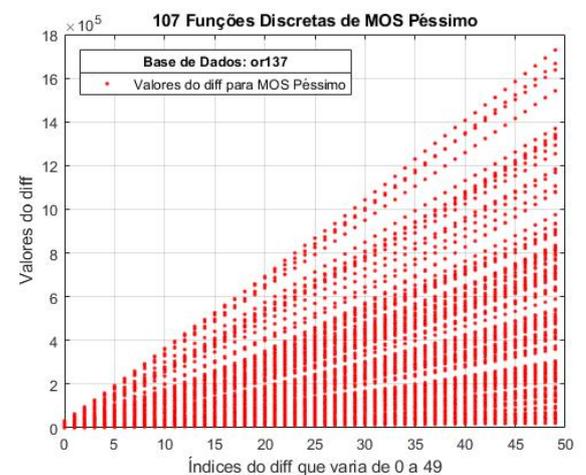
[c] 133 FD para or114 de MOS Péssimo



[d] 89 FD para or129 de MOS Péssimo



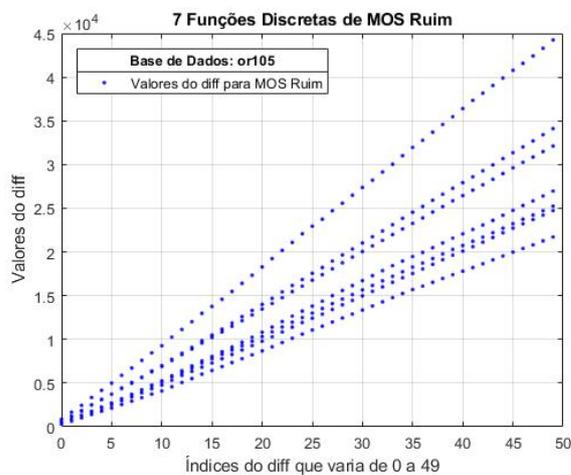
[e] 41 FD para or134 de MOS Péssimo



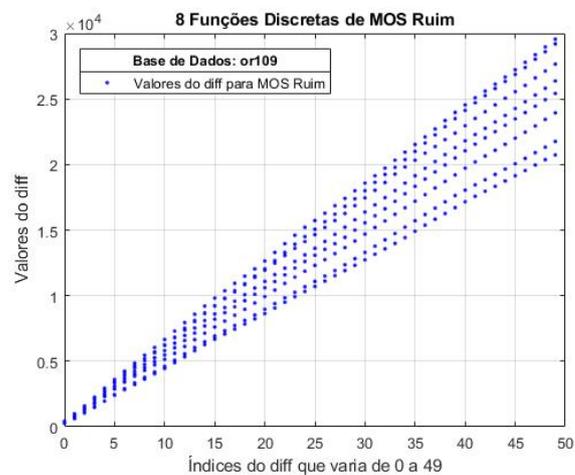
[f] 107 FD para or137 de MOS Péssimo

## APÊNDICE B – Funções Discretas para o MOS Ruim

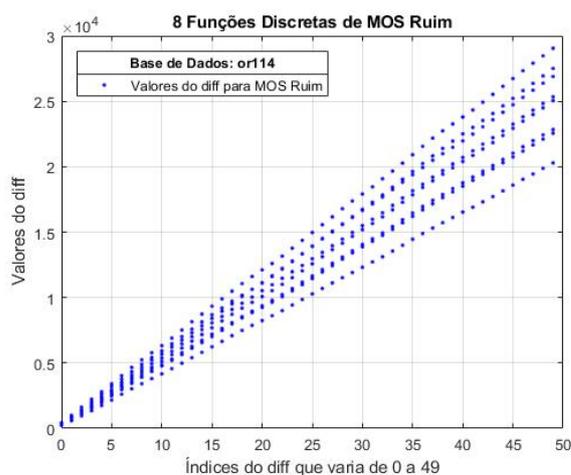
Figura 2 – Funções Discretas(FD) para o MOS Ruim.



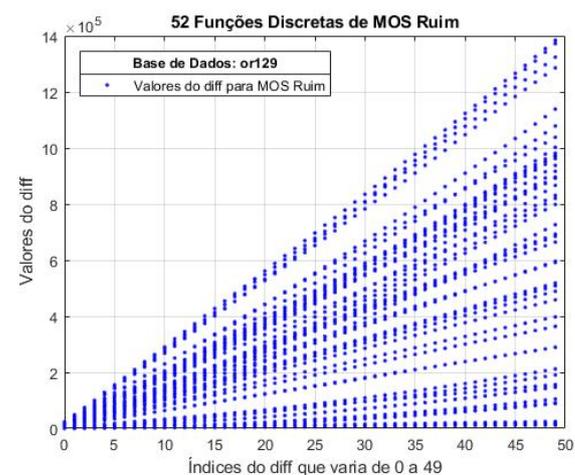
[a] 7 FD para or105 de MOS Ruim



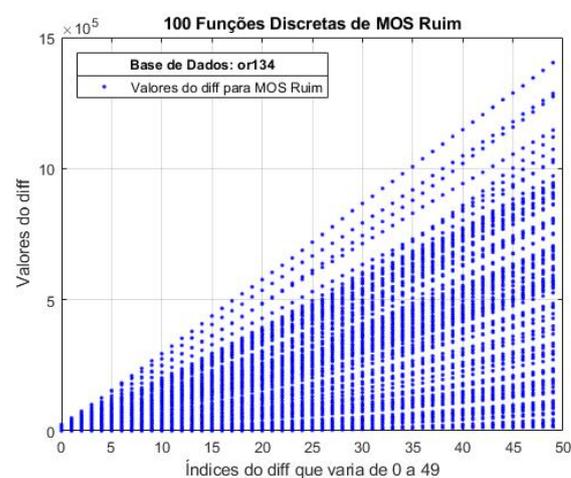
[b] 8 FD para or109 de MOS Ruim



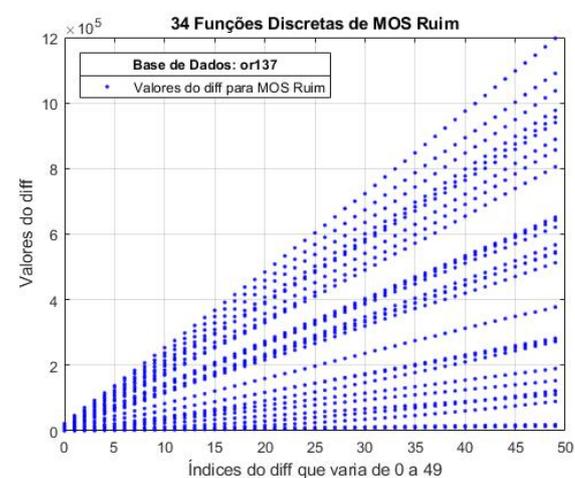
[c] 8 FD para or114 de MOS Ruim



[d] 52 FD para or129 de MOS Ruim



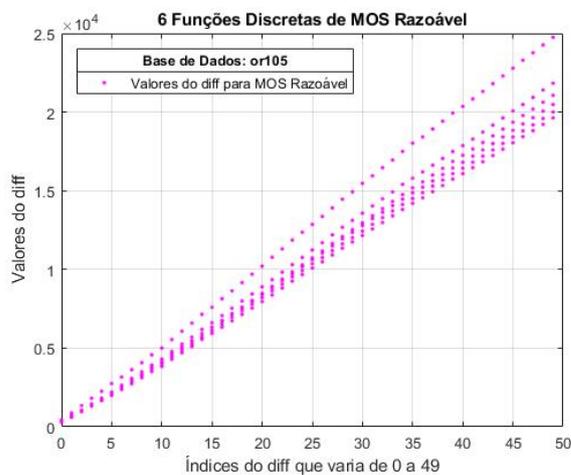
[e] 100 FD para or134 de MOS Ruim



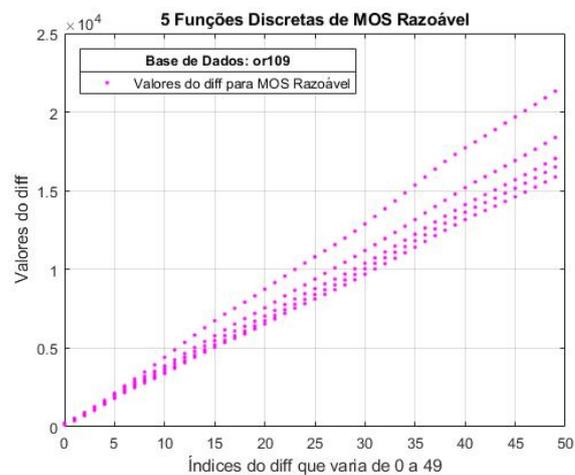
[f] 34 FD para or137 de MOS Ruim

## APÊNDICE C – Funções Discretas para o MOS Razoável

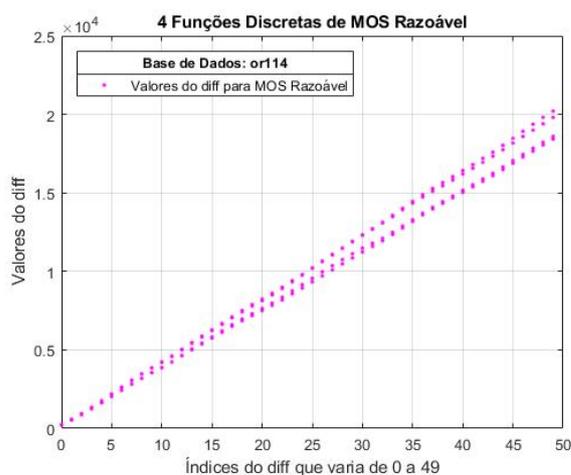
Figura 3 – Funções Discretas(FD) para o MOS Razoável.



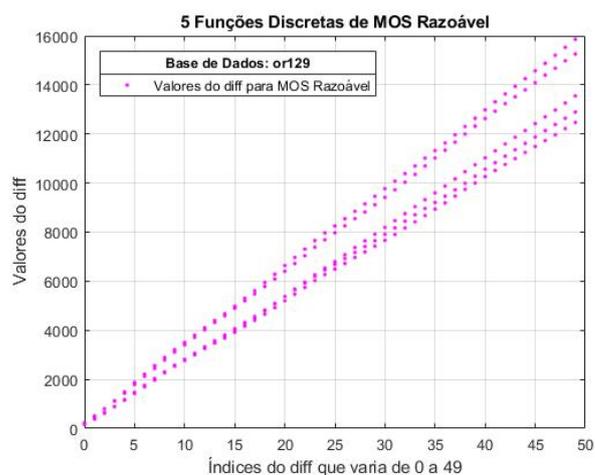
[a] 6 FD para or105 de MOS Razoável



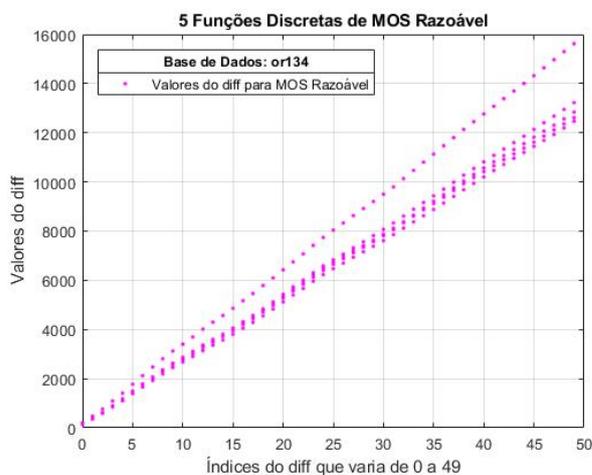
[b] 5 FD para or109 de MOS Razoável



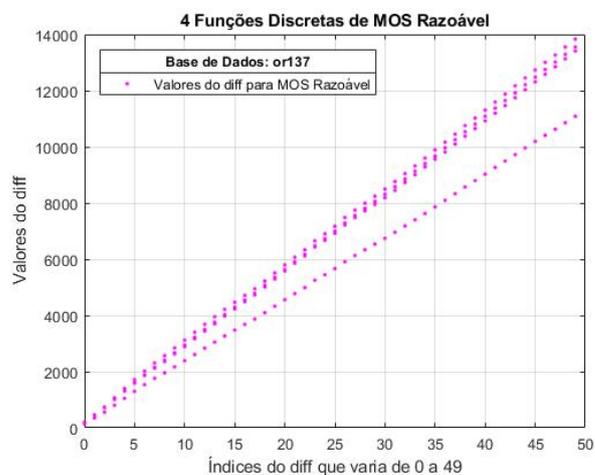
[c] 4 FD para or114 de MOS Razoável



[d] 5 FD para or129 de MOS Razoável



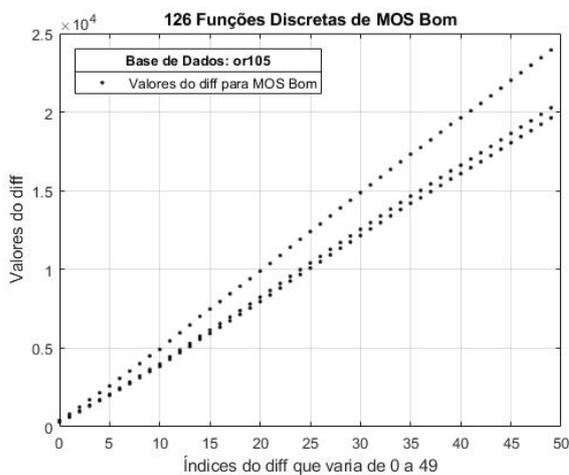
[e] 5 FD para or134 de MOS Razoável



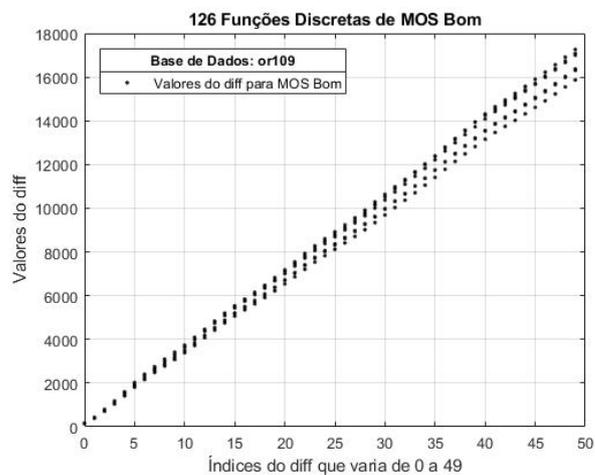
[f] 4 FD para or137 de MOS Razoável

## APÊNDICE D – Funções Discretas para o MOS Bom

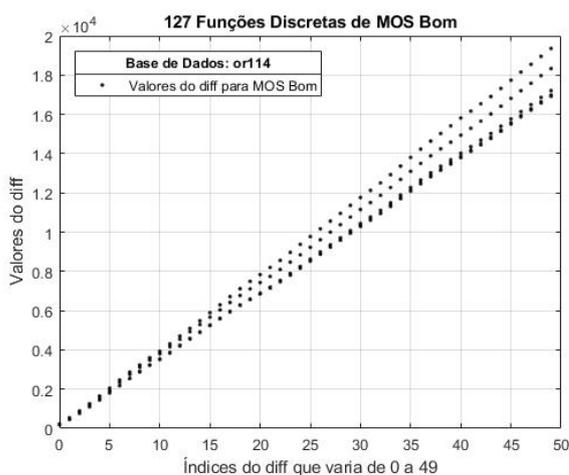
Figura 4 – Funções Discretas(FD) para o MOS Bom.



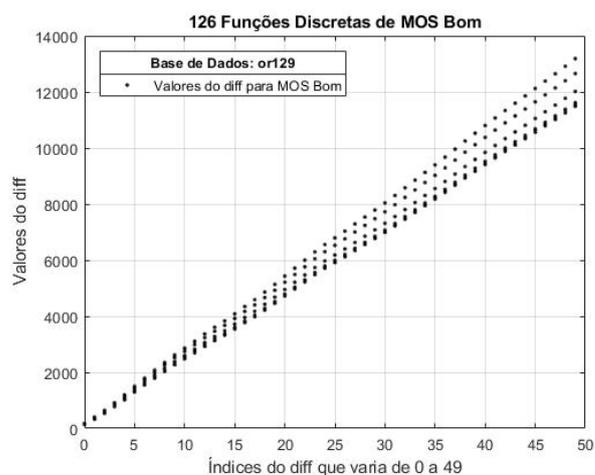
[a] 126 FD para or105 de MOS Bom



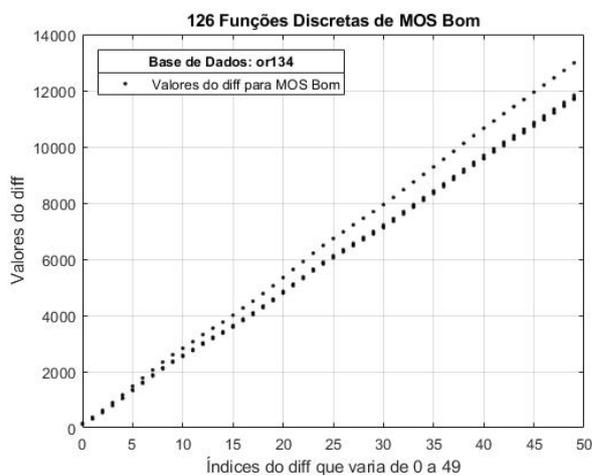
[b] 126 FD para or109 de MOS Bom



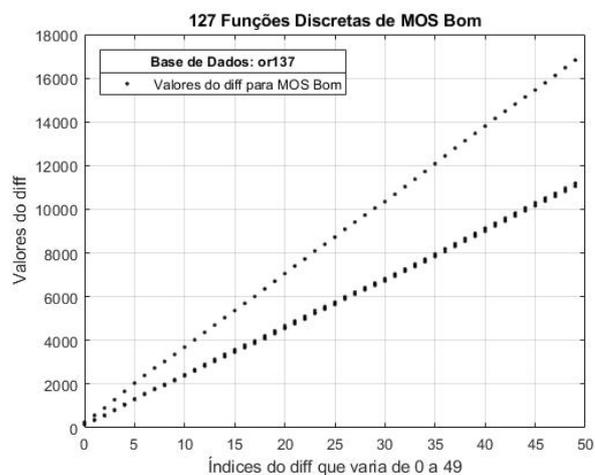
[c] 127 FD para or114 de MOS Bom



[d] 126 FD para or129 de MOS Bom



[e] 126 FD para or134 de MOS Bom



[f] 127 FD para or137 de MOS Bom