

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA - CURSO DE ENGENHARIA MECÂNICA
TRABALHO DE CONCLUSÃO DE CURSO

**INVESTIGAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA
DETECÇÃO DE FALHAS EM EQUIPAMENTOS INDUSTRIAIS ATRAVÉS DE
ANÁLISE SONORA**

por

Alexandre Stefano Menegazzo

Monografia apresentada ao Departamento de Engenharia Mecânica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para obtenção do diploma de Engenheiro Mecânico.

Porto Alegre, maio de 2021.

CIP - Catalogação na Publicação

Menegazzo, Alexandre Stefano

INVESTIGAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE FALHAS EM EQUIPAMENTOS INDUSTRIAIS ATRAVÉS DE ANÁLISE SONORA. / Alexandre Stefano Menegazzo. -- 2021. 30 f.

Orientador: Herbert Martins Gomes.

Trabalho de conclusão de curso (Graduação) -- Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Curso de Engenharia Mecânica, Porto Alegre, BR-RS, 2021.

1. Aprendizado de Máquina. 2. Redes Neurais. 3. Floresta Aleatória. 4. Análise sonora. I. Gomes, Herbert Martins, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

Alexandre Stefano Menegazzo

**INVESTIGAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA
DETECÇÃO DE FALHAS EM EQUIPAMENTOS INDUSTRIAIS ATRAVÉS DE
ANÁLISE SONORA**

ESTA MONOGRAFIA FOI JULGADA ADEQUADA COMO PARTE DOS
REQUISITOS PARA A OBTENÇÃO DO TÍTULO DE
ENGENHEIRO MECÂNICO
APROVADA EM SUA FORMA FINAL PELA BANCA EXAMINADORA DO
CURSO DE ENGENHARIA MECÂNICA

Prof. Mário Roland Sobczyk Sobrinho
Coordenador do Curso de Engenharia Mecânica

Área de Concentração: Mecânica dos Sólidos

Orientador: Prof. Dr. Herbert Martins Gomes

Comissão de Avaliação:

Prof. Dr. Jakson Manfredini Vassoler

Prof. Dr. Walter Jesus Paucar Casas

Prof. Dr. Herbert Martins Gomes

Porto Alegre, maio de 2021.

DEDICATÓRIA

Dedico este trabalho a toda a minha família e amigos, pelo apoio, ajuda e companhia durante os diferentes trajetos da minha vida, antes e durante a graduação.

AGRADECIMENTOS

Agradecimento especial a todos os meus colegas de trabalho da STIHL Ferramentas Motorizadas pelo apoio e incentivo à busca por conhecimento e aprimoração, apoio que foi essencial para o desenvolvimento dos conhecimentos aplicados no presente trabalho.

Em essência, todos os modelos estão errados, mas alguns são úteis.

George E. P. Box

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA - CURSO DE ENGENHARIA MECÂNICA
TRABALHO DE CONCLUSÃO DE CURSO – 2021

INVESTIGAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA
DETECÇÃO DE FALHAS EM EQUIPAMENTOS INDUSTRIAIS ATRAVÉS DE
ANÁLISE SONORA

Menegazzo, Alexandre Stefano
alexandre.menegazzo@gmail.com

Resumo. A detecção de futuras falhas em equipamentos mecânicos é uma atividade essencial dentro da área da Manutenção Preditiva, contribuindo para a resolução de problemas potencialmente críticos envolvendo produtos ou equipamentos fabris, antes mesmo que tais eventos aconteçam. O crescente aumento no volume de dados que podem ser adquiridos durante um processo de fabricação ou uso de um equipamento estimula o surgimento de novas tecnologias e metodologias capazes de extrair informações sobre o fenômeno estudado, sendo uma dessas a detecção de falhas através do som emitido pelos equipamentos. O presente trabalho tem como objetivo investigar e avaliar a aplicação de algoritmos de inteligência artificial para a detecção de anomalias sonoras em um componente industrial através de gravações sonoras obtidas de um banco de dados público. Um modelo de classificação de anomalias por Floresta Aleatória e uma Rede Neural Percéptron Multicamada foram construídos e treinados a partir de parâmetros acústicos extraídos dos sons, e métricas de avaliação como acurácia, precisão e recall foram utilizadas para a comparação de performance. Após o treinamento, ambos os modelos apresentaram uma taxa de acerto de sons anômalos superior a 95%, com o modelo de Rede Neural apresentando o melhor desempenho.

Palavras-chave: aprendizado de máquina, redes neurais, floresta aleatória, análise sonora

Abstract. The detection of future failures in mechanical equipment is an essential activity to the field of Predictive Maintenance, contributing to the solution of potentially critical problems involving products or industrial machinery before those problems can actually happen. The current increase in the volume of data that can be acquired during an industrial process or use of equipment promotes the development of new technology and methods capable of extracting information from the studied phenomena, one of those methods being the prediction of failures through the sound emitted by a machine. The objective of the present work is to investigate and evaluate the application of Machine Learning algorithms to the task of acoustic anomaly detection in an industrial equipment using sound recordings from a public database. A Random Forest anomaly classifier and a Multilayer Perceptron Neural Network were built and trained with acoustic features extracted from the sounds, and evaluation metrics such as accuracy, precision and recall were used for performance comparison. After the training, both models presented an accuracy rate above 95%, with the Neural Network showing the best results.

Keywords: machine learning, neural networks, random forest, sound analysis

LISTA DE FIGURAS

Figura 1 – Exemplo esquemático do algoritmo de uma árvore de decisão para classificação binária de amostras, com valores arbitrários α , β e γ a serem selecionados pelo modelo.	3
Figura 2 – Exemplo esquemático de uma rede neural Percéptron multicamada, sendo X o dado de entrada, h as computações intermediárias e Y o dado de saída.	4
Figura 3 – Exemplo de Curvas ROC para um modelo genérico de classificação.	7
Figura 4 – Comparação das ondas sonoras de uma bomba hidráulica em operação normal e em operação com falhas.	9
Figura 5 – Comparação dos espectrogramas de uma bomba hidráulica em operação normal e em operação com falhas.	9
Figura 6 – Matriz de confusão resultante da classificação dos dados desconhecidos para o modelo de Floresta Aleatória.	12
Figura 7 – Evolução da acurácia das classificações e função erro durante o treinamento do modelo.	13
Figura 8 – Matriz de confusão resultante da classificação dos dados desconhecidos para o modelo de Rede Neural.	14
Figura 9 – Comparação das curvas ROC e parâmetro AUC dos modelos treinados.	15
Figura A.1 – Exemplo esquemático do algoritmo de extração de coeficientes MFCC de acordo com os passos apresentados na seção 2.2.	17
Figura B.1 – Fluxograma dos algoritmos de ML desenvolvidos.	17
Figura C.1 – Comparação da variação dos 13 coeficientes MFCC para uma bomba hidráulica em operação normal e em operação com falhas.	18
Figura C.2 – Comparação da variação das derivadas dos 13 coeficientes MFCC para uma bomba hidráulica em operação normal e em operação com falhas.	18
Figura C.3 – Comparação da variação das derivadas de segunda ordem dos 13 coeficientes MFCC para uma bomba hidráulica em operação normal e em operação com falhas.	18

LISTA DE TABELAS

Tabela 1 – Pseudocódigo do algoritmo de Floresta Aleatória.....	3
Tabela 2 – Matriz de confusão para duas classes, considerando-se valores pertencentes à classe 2 como positivos.	6
Tabela 3 – Banco de dados total obtido a partir da segmentação dos arquivos em frames e extração dos coeficientes MFCC.	10
Tabela 4 – Banco de dados de treinamento e teste para o modelo de Floresta Aleatória.....	11
Tabela 5 – Estrutura da rede neural Percéptron Multicamada utilizada.....	11
Tabela 6 - Banco de dados de treinamento, validação e teste para o modelo de Rede Neural Percéptron Multicamada.	12
Tabela 7 – Resultados das métricas de avaliação e tempo de treinamento para o modelo de Floresta Aleatória.....	13
Tabela 8 – Resultados das métricas de avaliação e tempo de treinamento para o modelo de Rede Neural Percéptron.	14

NOMENCLATURA

Símbolos

f	Frequência da onda sonora	[Hz]
f_a	Função de ativação	[-]
H	Conjunto Vazio	[-]
h	Neurônio intermediário	[-]
k	Filtro de frequências	[-]
v, w	Pesos Multiplicadores	[-]
v^j_{MFCC}	j-ésimo coeficiente MFCC	[-]
X	Dados de entrada	[-]
$X[i]$	Atributo i dos dados de entrada	[-]
$ X(k, n) $	Magnitude espectral da onda sonora	[-]
Y	Dados de saída	[-]
$\Delta MFCC$	Derivada do coeficiente MFCC	[-]
$\Delta\Delta MFCC$	Derivada segunda do coeficiente MFCC	[-]

Símbolos gregos

α, β, γ	Valores arbitrários do exemplo esquemático
π	Número Pi

Abreviaturas e acrônimos

AUC	<i>Area Under the Curve</i>
DCT	<i>Discrete Cosine Transform</i>
FN	Falso Negativo
FP	Falso Positivo
MFCC	<i>Mel Frequency Cepstral Coefficients</i>
ML	<i>Machine Learning</i>

MSE	<i>Mean Squared Error</i>
ReLu	<i>Rectified Linear Units</i>
RF	<i>Random Forests</i>
RNA	Rede Neural Artificial
ROC	<i>Receiver Operating Characteristics</i>
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Motivação	1
1.2	Revisão Bibliográfica	1
1.3	Objetivo	1
2	FUNDAMENTAÇÃO	2
2.1	Aprendizado de Máquina.....	2
2.1.1	Aprendizado por Floresta Aleatória.....	2
2.1.2	Aprendizado por Redes Neurais Percéptron Multicamada.....	3
2.2	MFCC - Coeficientes Cepstrais de Frequência Mel.....	5
2.3	Métricas de Avaliação de Performance	6
2.3.1	Matriz de Confusão.....	6
2.3.2	Acurácia, precisão e recall.....	6
2.3.3	Curva ROC e parâmetro AUC.....	7
2.4	Generalização e <i>Overfitting</i>	8
3	METODOLOGIA	8
3.1	Seleção do banco de dados	8
3.2	Pré-processamento dos dados e extração dos atributos acústicos	10
3.3	Seleção e construção dos modelos de aprendizado	10
3.3.1	Modelo de Florestas Aleatórias	10
3.3.2	Modelo de Rede Neural Percéptron Multicamada.....	11
3.4	Custo e complexidade computacional	12
4	RESULTADOS	12
4.1	Modelo de Floresta Aleatória	12
4.2	Modelo de Rede Neural Percéptron.....	13
4.3	Comparação de performance e curvas ROC.....	14
5	CONCLUSÃO	15
	REFERÊNCIAS BIBLIOGRÁFICAS	16
	APÊNDICE A	17
	APÊNDICE B.....	17
	APÊNDICE C.....	18

1 INTRODUÇÃO

1.1 Motivação

A área da Manutenção preditiva desempenha um papel essencial para o controle da qualidade e desempenho de um equipamento mecânico em operação, podendo fornecer diagnósticos em tempo real e muitas vezes não invasivos sobre o estado atual do equipamento. A detecção de falhas futuras através da manutenção preditiva baseia-se em inspeções periódicas nas quais fenômenos como temperatura, vibração, ruídos excessivos etc. são observados através de instrumentos específicos (ALMEIDA, 2016). Tais inspeções requerem uma grande mão de obra especializada por longos períodos de tempo, logo, o desenvolvimento de metodologias de aprendizado de máquina capazes de ensinar sistemas computadorizados a prever falhas em componentes mecânicos tem recebido grande destaque em diferentes áreas da engenharia.

1.2 Revisão Bibliográfica

HOTHU; MITA, (2013) utilizaram algoritmos de Máquina de Suporte de Vetor para detectar a localização e severidade de falhas em vigas através de dados de vibração. Através da variação das frequências naturais entre corpos normais e corpos apresentando falhas, o modelo desenvolvido pelos pesquisadores identificou com sucesso problemas estruturais em corpos simulados e em corpos submetidos a testes.

A detecção de anomalias em rolamentos de máquinas rotativas é também uma área de elevado interesse dentro do campo de análise vibracional. (VARGAS-MACHUCA; GARCÍA; CORONADO, 2020) realizaram uma comparação detalhada entre o uso de modelos tradicionais e modelos de aprendizado de máquina na tarefa de classificação de falhas em rolamentos através de dados vibracionais, obtendo, com estes últimos, uma performance de detecção de falhas até 33% superior aos modelos tradicionais.

Com o surgimento de bancos de dados públicos de anomalias acústicas, diferentes métodos para a detecção e previsão de falhas através de seu som vêm sendo estudados. DUMAN; BAYRAM; İNCE, (2020) realizaram a detecção de falhas em diferentes processos industriais através da criação de um algoritmo do tipo *autoencoder* utilizando, como dados de entrada, espectrogramas gerados a partir dos sons.

Diferentes métodos de extração de parâmetros acústicos também são explorados, como no trabalho de DENG *et al.*, (2020), em que os pesquisadores utilizaram redes neurais convolucionais para a detecção de anomalias cardíacas a partir de coeficientes espectrais extraídos de amostras sonoras, obtendo até 98% de taxa de acerto nos modelos treinados.

1.3 Objetivo

De frente à grande gama de possibilidades de aplicações presentes na literatura, tanto de métodos de aprendizado de máquina quanto de métodos de entrada de dados, o presente trabalho tem como objetivo investigar a performance de um modelo de Florestas Aleatórias e um modelo de Rede Neural Artificial Percéptron Multicamadas quando aplicados para detecção de um tipo específico de anomalia acústica que pode ocorrer em um equipamento industrial. O aprendizado dos modelos será realizado através da extração de parâmetros acústicos conhecidos como Coeficientes Cepstrais de Frequência Mel, que vêm mostrando resultados excelentes na literatura existente e simulam a maneira que um ouvido humano interpreta e diferencia ondas sonoras.

2 FUNDAMENTAÇÃO

2.1 Aprendizado de Máquina

O aprendizado de máquina, ou *Machine Learning* (ML), é um nome genérico que se dá na Ciência da Computação a todas as técnicas e algoritmos ligados ao reconhecimento de padrões, inteligência artificial, classificação e previsão de dados. Estas técnicas permitem aos sistemas computacionais aprender de forma independente sem que sejam explicitamente programados para isto. Exemplos de sistemas de aprendizado de máquina são as Redes Neurais Artificiais (RNA) e suas inúmeras variantes, Máquinas de Suporte de Vetor, Aprendizado Profundo (*Deep Learning*), Árvores de Decisão, Florestas Aleatórias e até mesmo Superfícies de Resposta baseadas em regressões lineares e não lineares, entre outros.

As duas principais formas de aprendizado utilizadas são o aprendizado supervisionado e o não supervisionado. Nas técnicas de aprendizado supervisionado, o algoritmo aprende a extrair informações de dados previamente conhecidos e classificados, e posteriormente testa a eficácia de tal aprendizado em dados desconhecidos. O aprendizado não supervisionado, por outro lado, consiste na obtenção de informações diretamente a partir de dados desconhecidos, através da inferência de características e padrões contidos no banco de dados estudado (LUCIAN, 2020). O presente trabalho focará no estudo de algoritmos de aprendizado supervisionado, ou seja, a partir de um banco de dados de informações previamente classificadas.

2.1.1 Aprendizado por Floresta Aleatória

Florestas Aleatórias, ou *Random Forests* (RF), é um método de aprendizado de máquina registrado por (BREIMAN, 2011) como uma extensão do método de Árvores de Decisão, baseando-se na agregação dos resultados de um conjunto destes estimadores mais simples.

Uma árvore de decisão é um algoritmo de ML supervisionado que é amplamente usado em problemas de classificação e regressão. Enquanto outros modelos de ML usam uma fórmula preditiva global contendo todo o volume de dados, a árvore de decisão subdivide este volume em regiões menores e controláveis. Se utilizado como classificador, as regras seguidas pelo algoritmo para realizar tal subdivisão são determinadas pela seleção de atributos pertencentes às amostras que melhor as dividam entre as classes existentes.

No método de regressão com árvore de decisão, a função para medir a qualidade de uma divisão é o erro quadrático médio (MSE, *Mean Squared Error*), que equivale à redução de variância como um critério de seleção de características. Após esta subdivisão inicial, novas e sucessivas divisões são realizadas até que todas as amostras sejam classificadas, ou um critério de parada seja atingido. A Figura 1 apresenta um exemplo de árvore de decisão para a classificação binária de amostras X contendo 3 atributos, $X[1]$, $X[2]$ e $X[3]$ por amostra.

Como mencionado anteriormente, cada árvore de decisão em uma floresta aleatória é treinada no subconjunto de dados fornecidos e a saída de múltiplas árvores de decisão criadas aleatoriamente são combinadas para gerar a saída final. O algoritmo RF é não tendencioso e funciona bem mesmo quando os dados têm valores ausentes ou não foram bem escalonados (MÜLLER, A., & GUIDO, 2018). Além disso, ele melhora a acurácia preditiva e controla o *overfitting* durante o treinamento.

O pseudocódigo deste algoritmo é ilustrado na Tabela 1, com a coluna à direita representando a descrição de cada linha. Basicamente, um conjunto de árvores de decisão (floresta) é criado e treinado como regressor ou classificador para apenas um subconjunto das características (saídas) dos pontos de treinamento (amostras). Após o treinamento, uma amostra qualquer é avaliada pela floresta de árvores de decisão, resultando em diversos estimadores para aquela amostra. A saída da floresta aleatória é obtida pela média do conjunto de saídas de árvores escolhidas aleatoriamente desta floresta, como indicado por (BREIMAN, 2011).

Figura 1 – Exemplo esquemático do algoritmo de uma árvore de decisão para classificação binária de amostras, com valores arbitrários α , β e γ a serem selecionados pelo modelo.

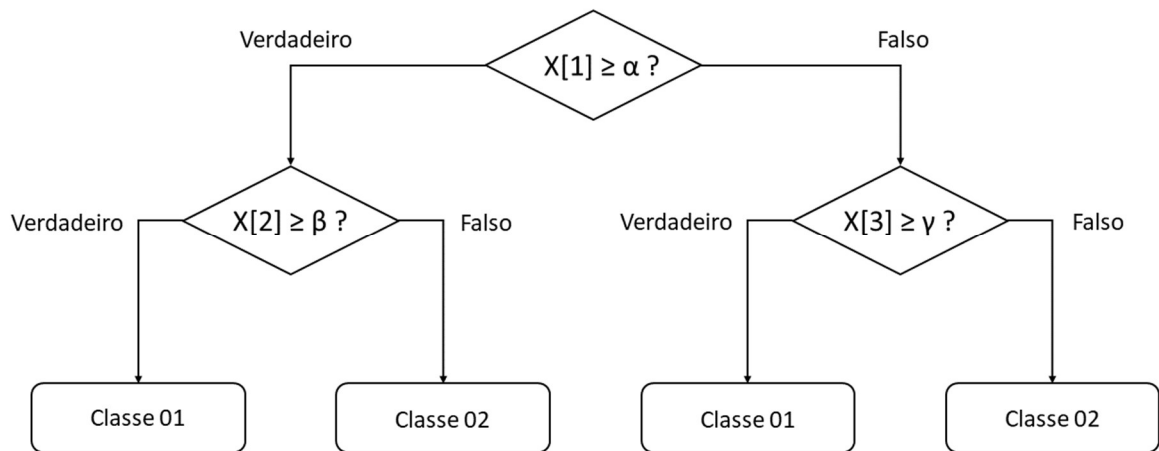


Tabela 1 – Pseudocódigo do algoritmo de Floresta Aleatória.

Algoritmo de Floresta Aleatória

1	Função Floresta Aleatória (X, X[])	%X é o conjunto de dados de treinamento e X[] o de atributos
2	H ← ∅	%Inicializa conjunto H como vazio
3	Laço i=1,...,N	%Laço para o conjunto de N árvores (floresta)
4	X ⁱ ← amostra retirada de X, com reposição	%Retira-se uma amostra com reposição de X
5	h ⁱ ← Árvore Aleatória(X ⁱ , X[])	%Busca-se com X ⁱ uma árvore aleatória para os atributos X[]
6	H ← H ∪ {h ⁱ }	%Aumenta-se o tamanho do conjunto H com a árvore treinada h ⁱ
7	Fim laço i	%Fim do laço de árvores
8	Fim Função Floresta Aleatória	%Fim da função de floresta aleatória
9		
10	Função Árvore Aleatória(X, X[])	%Função de árvore aleatória
11	Em cada nó da árvore	%Para cada nó da árvore
12	x[] ← subconjunto pequeno de X[]	%seleciona um subconjunto pequeno de atributos de X
13	Divida no melhor atributo em x[]	%Dividir no melhor atributo em x
14	Retorne a árvore de aprendida	%Retorna a árvore aprendida
15	Fim função Árvore Aleatória	%Fim da função Árvore Aleatória

Adaptado de BREIMAN, (2011)

2.1.2 Aprendizado por Redes Neurais Percéptron Multicamada

Redes Neurais Artificiais (RNA) são modelos computacionais cujo processamento de informação é inspirado nas vias neurais de um cérebro humano. Em uma RNA, um agrupamento de camadas de módulos (neurônios), conectados entre si, recebe um sinal de entrada e o propaga pela rede, com cada camada recebendo a soma ponderada da informação de saída da camada anterior combinada com pesos e funções não lineares. O processo de aprendizado, então, se dá pela busca da melhor combinação de pesos para cada neurônio que

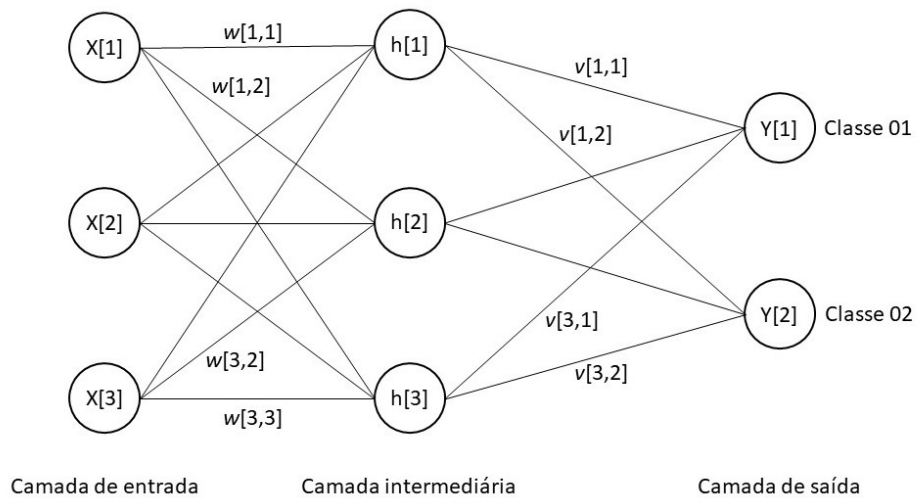
minimize o erro na tarefa de classificação ou regressão dos dados, de forma que tal tarefa possa ser realizada com elevada taxa de acerto para um conjunto desconhecido de sinais de entrada.

Como descrito por (LUCIAN, 2020), a arquitetura de uma RNA é definida pelo seu número de camadas, pelo número de neurônios de entrada, das camadas intermediárias e da camada de saída, bem como pelo grau de conectividade entre os neurônios e pela existência ou não de conexões de retroalimentação.

A RNA do tipo Percéptron Multicamada foi proposta por (ROSENBLATT, 1958) e é constituída de uma camada de entrada, uma ou mais camadas intermediárias, e uma camada de saída. A quantidade de neurônios de entrada equivale à quantidade de atributos pertencentes aos dados, enquanto que a quantidade de neurônios na camada de saída é unitária para um modelo de regressão, e equivalente à quantidade de classes para um modelo de classificação. Neste tipo de arquitetura as camadas são totalmente conectadas, não existindo conexões de retroalimentação, sendo portanto chamadas também de redes do tipo *feed-forward*.

A Figura 2 apresenta um exemplo esquemático de RNA do tipo Percéptron Multicamada para a classificação binária de amostras X contendo 3 atributos (dimensões) cada. Neste esquema, cada conexão entre neurônios possui um peso, ou parâmetro, w e v , e cada camada de neurônios possuirá uma função não linear, chamada de função de ativação. As Equações 1 a 3 apresentam o cálculo das funções intermediárias, h , enquanto as Equações 4 e 5 apresentam o cálculo da saída final da rede, Y , como descrito por (MÜLLER, A., & GUIDO, 2018).

Figura 2 – Exemplo esquemático de uma rede neural Percéptron multicamada, sendo X o dado de entrada, h as computações intermediárias e Y o dado de saída.



$$h[1] = f_a \cdot (w[1,1] \cdot X[1] + w[2,1] \cdot X[2] + w[3,1] \cdot X[3]) \quad (1)$$

$$h[2] = f_a \cdot (w[1,2] \cdot X[1] + w[2,2] \cdot X[2] + w[3,2] \cdot X[3]) \quad (2)$$

$$h[3] = f_a \cdot (w[1,3] \cdot X[1] + w[2,3] \cdot X[2] + w[3,3] \cdot X[3]) \quad (3)$$

$$Y[1] = v[1,1] \cdot h[1] + v[2,1] \cdot h[2] + v[3,1] \cdot h[3] \quad (4)$$

$$Y[2] = v[1,2] \cdot h[1] + v[2,2] \cdot h[2] + v[3,2] \cdot h[3] \quad (5)$$

Neste conjunto de equações, f_a representa a função de ativação a ser selecionada para acrescentar não-linearidades no sistema de equações e tornar o aprendizado mais poderoso e eficiente, de acordo com os mesmos autores.

Em um problema de aprendizado supervisionado, cada saída Y é comparada com sua classificação real através de uma função de erro, sendo as mais utilizadas funções de medida de variância, como o MSE, ou funções de medida da incerteza na previsão associada à distribuição dos dados, como a Entropia Cruzada (KLINE; BERARDI, 2005).

2.2 MFCC - Coeficientes Cepstrais de Frequência Mel

Dentre os diferentes métodos de extração de atributos acústicos de uma onda sonora, como espectrogramas ou transformadas de Fourier, um dos métodos mais conhecidos e utilizados é a extração de coeficientes cepstrais de frequência Mel, ou *Mel Frequency Cepstral Coefficients* (MFCC). Comumente utilizado para modelos de classificação de voz, os coeficientes extraídos podem ser vistos como uma descrição compacta da forma de um envelope espectral de um sinal de áudio (LERCH, 2012). Em outras palavras, um específico coeficiente representa numericamente a contribuição de um específico intervalo de frequências na energia espectral total de um segmento de áudio utilizado. Neste método, os intervalos de frequências utilizados são definidos a partir da escala Mel, de forma a simular a percepção humana de uma onda sonora.

O cálculo do j -ésimo coeficiente para cada segmento n de onda sonora, $v^j_{MFCC}(n)$, é descrito por LERCH, 2012, a partir da Eq.(6).

$$v^j_{MFCC}(n) = \sum_{k=1}^K \log(|X(k, n)|) \cdot \cos\left(j \cdot \left(k - \frac{1}{2}\right) \frac{\pi}{K}\right) \quad (6)$$

com $|X(k, n)|$ representando a magnitude espectral do sinal computado através de um banco de K filtros de frequências na escala Mel, definida a partir da Eq.(7).

$$mel(f) = 2595 \log\left(1 + \frac{f}{700}\right) \quad (7)$$

com f sendo a frequência do sinal (Hz). A metodologia para sua extração é apresentada a seguir, de acordo com (AJIBOLA ALIM; KHAIR ALANG RASHID, 2018):

1. É realizada a divisão do sinal de áudio em n segmentos, ou *frames*, tipicamente com duração de milissegundos;
2. É calculada a transformada de Fourier de cada segmento para a determinação das magnitudes espectrais, $|X(k, n)|$.
3. As magnitudes espectrais são processadas através de um banco de K filtros de frequência na escala Mel.
4. As magnitudes espectrais resultantes da etapa anterior são transformadas para um domínio logarítmico e uma transformada discreta de cosseno (*Discrete Cosine Transform*, DCT) é aplicada para a obtenção dos coeficientes.

Um exemplo esquemático da aplicação dos passos acima encontra-se no Apêndice A. De acordo com os autores, a quantidade de coeficientes resultantes depende do número escolhido de filtros do banco frequências, sendo normalmente utilizados os primeiros 13 coeficientes.

2.3 Métricas de Avaliação de Performance

2.3.1 Matriz de Confusão

A matriz de confusão é uma tabela que permite avaliar o desempenho de um algoritmo de classificação, como, por exemplo, um algoritmo treinado com aprendizado de máquina, frente aos seus acertos e erros na predição da classificação de amostras.

A matriz tem dimensão do número de classes e é quadrada, com as linhas representando as classes reais, e as colunas as classes previstas pelo sistema treinado. A diagonal principal da matriz indica os acertos verdadeiros, enquanto que os elementos fora da diagonal indicam classificações erradas. O sistema de classificação ideal é aquele que possui apenas elementos diagonais cuja soma resultam em 100%, indicando total acerto em qualquer classe avaliada. A Tabela 2 representa uma matriz de confusão para uma classificação binária considerando a classe 2 como classificações positivas, para fins de definição das métricas de classificação.

Tabela 2 – Matriz de confusão para duas classes, considerando-se valores pertencentes à classe 2 como positivos.

		Classe Prevista	
		Classe 1	Classe 2
Classe Real	Classe 1	Verd. Negativo (VN)	Falso Positivo (FP)
	Classe 2	Falso Negativo (FN)	Verd. Positivo (VP)

2.3.2 Acurácia, precisão e recall

Embora a matriz de confusão ofereça uma análise qualitativa da performance geral do modelo, a partir dela é possível se definir diferentes métricas capazes de avaliar quantitativamente um modelo, como realizado por (MÜLLER, A., & GUIDO, 2018).

De posse dos valores de Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP) e Falsos Negativos (FN) de uma determinada classificação, a Acurácia de um modelo, ou seja, a fração de previsões corretas dentre o total de previsões realizadas, pode ser definida pela Equação 8:

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (8)$$

Em casos em que a falha na previsão correta pode ser prejudicial, também é relevante analisar a performance de classificação de um modelo para classes específicas. Isso pode ser realizado através de métricas como a Precisão e a Revocação (*Recall*), que representam, respectivamente, a fração de amostras verdadeiramente positivas (ou negativas) dentre o total

de previsões positivas (ou negativas) realizadas e a fração de amostras verdadeiramente positivas (ou negativas) dentre o total de amostras existentes, como definido a seguir:

$$Precisão = \frac{VP}{VP + FP} \quad (9)$$

$$Recall = \frac{VP}{VP + FN} \quad (10)$$

Um aprendizado eficiente de um modelo de aprendizado de máquina é aquele capaz de balancear uma alta precisão com um alto *recall*, ou seja, obter uma alta precisão nas classificações sem permitir a elevada ocorrência de falsos negativos, os quais podem ser prejudiciais em modelos de detecção de anomalias. Tal balanço entre precisão e *recall* pode ser medido através da métrica conhecida como *F₁ score*, como definida abaixo:

$$F_1score = 2 \cdot \left(\frac{Precisão * Recall}{Precisão + Recall} \right) \quad (11)$$

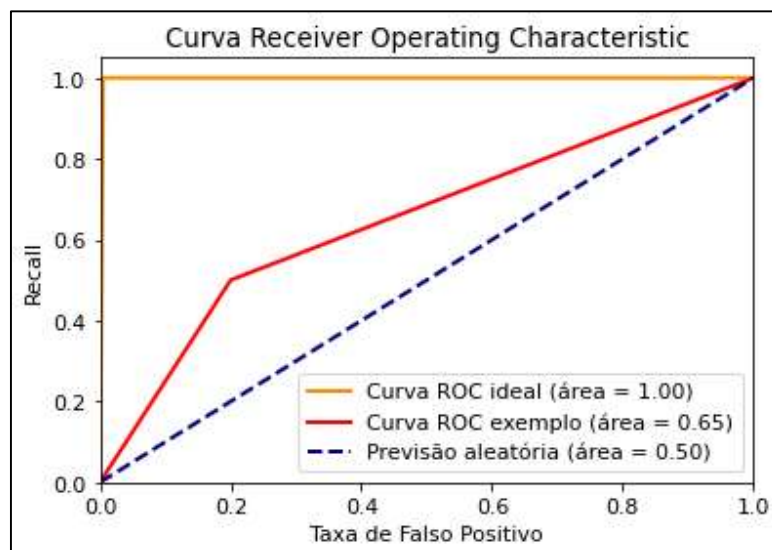
2.3.3 Curva ROC e parâmetro AUC

Uma outra ferramenta amplamente utilizada na literatura para a correta comparação de performance de diferentes classificadores é a *Receiver Operating Characteristics Curve*, ou curva ROC, definida por Prati et al (2008) como o gráfico entre a taxa de falsos positivos, *false positive rate* (FPR), e a taxa de verdadeiros positivos, *true positive rate* (TPR). A FPR pode ser definida pela Equação 12:

$$FPR = \frac{FP}{FP + VN} \quad (12)$$

enquanto que a TPR é definida como o próprio *recall*. A Figura 3 apresenta um exemplo de curva ROC para um modelo genérico de classificação. A linha pontilhada representa uma tentativa aleatória de classificação, ou seja, realizada por um modelo antes do treinamento.

Figura 3 – Exemplo de Curvas ROC para um modelo genérico de classificação.



Através da imagem é possível concluir que o modelo de maior performance será aquele que forneça o maior *recall* enquanto mantém uma baixa taxa de falsos positivos. Esta avaliação da curva ROC também pode ser fornecida através da área sob a curva, ou *area under the curve* (AUC), de forma que uma curva ideal possuirá um AUC unitário, enquanto que uma curva de previsão aleatória possuirá um AUC de 0,50, por exemplo.

2.4 Generalização e *Overfitting*

O objetivo de um modelo de aprendizado de máquina é aprender as características relevantes do banco de dados estudado de forma que seja capaz de realizar previsões corretas quando alimentado com dados desconhecidos, sendo tal capacidade chamada de generalização.

Caso um modelo seja demasiadamente complexo para o banco de dados avaliado ou a quantidade de dados disponíveis muito pequena, o mesmo irá aprender cada particularidade dos dados existentes, não sendo capaz de generalizar quando de frente a dados desconhecidos. Tal fenômeno é conhecido como *overfitting*, e pode ser evitado através de testes com dados desconhecidos e otimizações dos parâmetros do modelo utilizado, como a quantidade e profundidade máxima das árvores de decisão em um algoritmo de floresta aleatória e a quantidade de neurônios e camadas em uma rede neural (MÜLLER, A., & GUIDO, 2018).

3 METODOLOGIA

O Apêndice B apresenta o fluxograma dos algoritmos desenvolvidos no presente trabalho, partindo-se do pré-processamento dos áudios do banco de dados selecionado, da extração de atributos acústicos do sinal, treinamento e teste dos modelos construídos e resultados das classificações.

3.1 Seleção do banco de dados

O primeiro passo para a investigação de algoritmos de classificação de anomalias acústicas deve ser a criação ou seleção de um banco de dados de amostras sonoras, sendo procurado um banco de dados público formado por sons de maquinários normalmente utilizados na indústria.

O banco de dados selecionado foi o *Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection (MIMII dataset)*, disponibilizado por (PUROHIT et al., 2019) e contém gravações de som de 4 diferentes equipamentos mecânicos utilizados na indústria: ventiladores, válvulas, bombas hidráulicas e esteiras deslizantes. Os autores realizaram gravações de diferentes modelos de cada tipo de equipamento, com cada modelo possuindo entre 5 mil e 10 mil segundos de som normal e 1000 segundos de som anômalo. Foi utilizado um conjunto de microfones *System In Frontier Inc.* em uma distribuição circular ao redor do equipamento. As gravações foram feitas com uma taxa de amostragem de 16 kHz e resolução de 16 bits, com 8 canais de áudio.

Devido à extensão do banco de dados e à elevada complexidade de uma análise de diferentes classificadores para diferentes equipamentos e modelos de equipamentos, serão utilizados apenas sons de um dos modelos de bomba hidráulica, com o objetivo de avaliar a robustez dos classificadores para estas condições.

O modelo de bomba hidráulica selecionado foi gravado em operações de sucção e descarga de água em tanques, e o banco de dados formado por suas gravações compõe 1006 arquivos de áudio de operação normal e 146 segmentos de áudio com operação anômala, ambos com duração de 10 segundos cada segmento.

As figuras 4 e 5 apresentam uma comparação entre as ondas sonoras e espectrogramas de uma amostra de som normal e som anômalo para o equipamento selecionado. Ambas as figuras foram geradas com a biblioteca *Librosa* de processamento de áudio para Python (MCFEE et

al., 2020) e nelas é possível perceber as mudanças tanto em termos de amplitude quanto em termos de frequências dominantes entre uma onda sonora de bomba hidráulica em operação normal e uma em operação com falhas, evidenciando um som de “batimento” ocasionado por defeitos na bomba hidráulica em questão.

Figura 4 – Comparação das ondas sonoras de uma bomba hidráulica em operação normal e em operação com falhas.

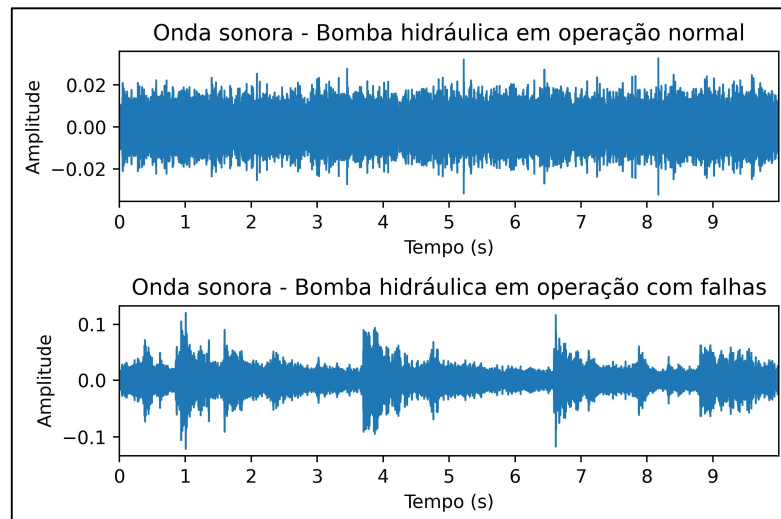
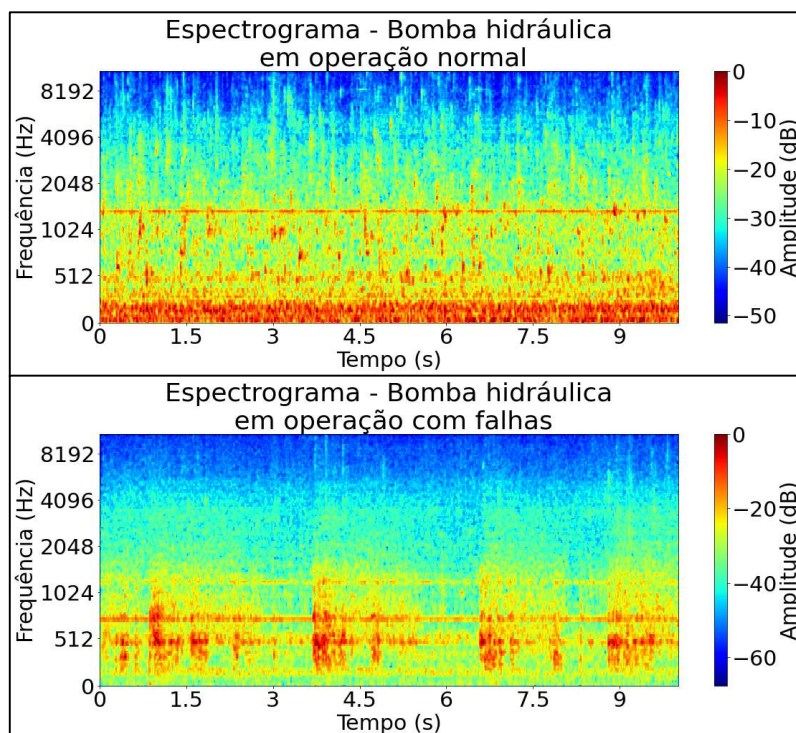


Figura 5 – Comparação dos espectrogramas de uma bomba hidráulica em operação normal e em operação com falhas.



3.2 Pré-processamento dos dados e extração dos atributos acústicos

Foi realizada a normalização das amplitudes das ondas sonoras, assim como a redução dos 8 canais de áudio iniciais para apenas um canal, composto da média das amplitudes sonoras obtidas pelos 8 microfones independentes, de forma a se garantir a homogeneidade do domínio dos dados de entrada.

Para a extração dos coeficientes MFCC, foi utilizada a função *feature.mfcc* da biblioteca *Librosa*. Cada arquivo de áudio de 10 segundos de duração foi segmentado em 313 frames de aproximadamente 30 milissegundos cada. Foi utilizada uma sobreposição padrão de 75% entre frames sucessivos e a quantidade padrão de 13 coeficientes por frame.

Segundo DENG *et al.*, (2020), os coeficientes obtidos anteriormente refletem apenas as características estacionárias do som, logo, para que as características dinâmicas dos sons gravados, como mudanças bruscas de rotação, possam ser levadas em consideração, as derivadas temporais de primeira e segunda ordem de cada coeficiente MFCC também foram extraídas, totalizando 39 atributos acústicos para cada frame dos arquivos de áudio. Diferentemente das ondas sonoras, a normalização dos coeficientes MFCC depende do modelo de aprendizado a ser usado (MÜLLER, A., & GUIDO, 2018), e será discutida durante a construção de cada modelo.

O Apêndice C apresenta a variação dos 13 coeficientes e suas derivadas para um arquivo de 10 segundos de áudio segmentado em 313 frames.

De forma a se obter um banco de dados balanceado entre a classe de sons normais e sons anômalos, uma quantidade igual de arquivos de áudio normais e anômalos foram selecionados para a extração dos coeficientes. A Tabela 3 apresenta a quantidade total de amostras obtidas após a segmentação dos arquivos em frames e extração dos coeficientes MFCCs.

Tabela 3 – Banco de dados total obtido a partir da segmentação dos arquivos em frames e extração dos coeficientes MFCC.

Banco de dados de entrada dos modelos		
	Sons Normais	Sons Anômalos
Número de arquivos	143	143
Duração de cada arquivo [s]	10	10
Frames por arquivo	313	313
Amostras (frames) totais	44759	44759
Atributos por amostra	39	39
Matriz resultante	(89518 x 39)	

3.3 Seleção e construção dos modelos de aprendizado

A partir da literatura existente e das características do banco de dados a ser analisado, em especial o grande número de atributos a serem aprendidos pelos modelos, foram selecionados algoritmos que apresentem robustez na tarefa de classificação e que possuam uma complexidade similar quanto à abordagem de aprendizado.

3.3.1 Modelo de Florestas Aleatórias

Conforme descrito anteriormente, o algoritmo de florestas aleatórias é construído a partir de um número previamente selecionado de modelos de árvore de decisão, ou classificadores. Para

o presente trabalho, foram utilizados 30 classificadores (árvores), cujos resultados individuais de classificação a partir dos atributos são combinados pelo algoritmo de floresta aleatória.

O modelo foi construído através da função *RandomForestClassifier* do módulo Scikit-Learn de aprendizado de máquina para Python (PEDREGOSA ET AL., 2011). Testes realizados com diferentes quantidades de classificadores mostraram que a partir de 30 árvores os resultados não apresentavam mudança, porém aumentavam em custo computacional. Os outros parâmetros da função foram mantidos como padrão.

Devido à robustez do algoritmo de florestas aleatórias quanto a dados não escalonados, como mencionado na seção 2.1.1 e descrito por (MÜLLER, A., & GUIDO, 2018), não foi realizada a normalização do banco de dados de coeficientes MFCCs. De forma a se utilizar a mesma proporção de dados de teste para ambos os algoritmos testados, aproximadamente 10% do banco de dados foi separado do treinamento para posterior comparação de performance com dados desconhecidos pelo modelo treinado. A Tabela 4 apresenta o resultado da divisão do banco de dados entre treinamento e teste.

Tabela 4 – Banco de dados de treinamento e teste para o modelo de Floresta Aleatória.

Divisão do banco de dados - Modelo Floresta Aleatória				
	Som Normal	Som Anômalo	Total [amostras]	Total [%]
Amostras (frames) totais	44759	44759	89518	100%
Amostras de treinamento	40283	40283	80566	90%
Amostras de teste	4476	4476	8952	10%

3.3.2 Modelo de Rede Neural Percéptron Multicamada

O modelo de RNA utilizado foi construído através da plataforma *TensorFlow* de *Machine Learning* para Python (ABADI et al., 2016) e sua estrutura pode ser visualizada na Tabela 5.

Tabela 5 – Estrutura da rede neural Percéptron Multicamada utilizada.

Camada	Número de atributos	Parâmetros
01 - Densa (entrada)	39	1560
Função de ativação: ReLu		
02 - Densa	64	2560
Função de ativação: ReLu		
DropOut	64	0
03 - Densa (saída)	2	130
Função de ativação: Softmax		

Como descrito anteriormente, a rede é formada por camadas de neurônios totalmente conectadas (camadas Densas), sem a presença de conexões de retroalimentação, possuindo um total de 4250 parâmetros treináveis, ou seja, 4250 pesos a serem aprendidos para a classificação dos dados com alta precisão. A função de ativação das camadas intermediárias utilizada foi a *Rectified Linear Units* (ReLu), uma função de cancelamento de pesos negativos para melhor propagação da informação. Foi utilizada a função *Softmax* para a saída dos dados, função que retorna as probabilidades de uma amostra pertencer a cada classe existente. Uma camada *DropOut* foi utilizada de forma a acrescentar ruído nos dados com o objetivo de aumentar a

capacidade de generalização do modelo e prevenir o fenômeno de *overfitting* (DENG et al., 2020).

Devido à sensibilidade das redes neurais Percéptron quanto ao escalonamento dos dados (MÜLLER, A., & GUIDO, 2018), foi realizada a normalização do banco de dados, de forma que os valores dos atributos existam no domínio [0, 1]. Previamente à extração dos atributos, os arquivos foram separados aleatoriamente entre banco de dados de treinamento, validação e teste dos modelos. A proporção selecionada foi de 80% para treinamento, 10% para validação e 10% para teste, como explicitado na Tabela 6.

Tabela 6 - Banco de dados de treinamento, validação e teste para o modelo de Rede Neural.

Divisão do banco de dados - Modelo Rede Neural Percéptron				
	Som Normal	Som Anômalo	Total [amostras]	Total [%]
Amostras (frames) totais	44759	44759	89518	100%
Amostras de treinamento	35807	35807	71614	80%
Amostras de validação	4476	4476	8952	10%
Amostras de teste	4476	4476	8952	10%

3.4 Custo e complexidade computacional

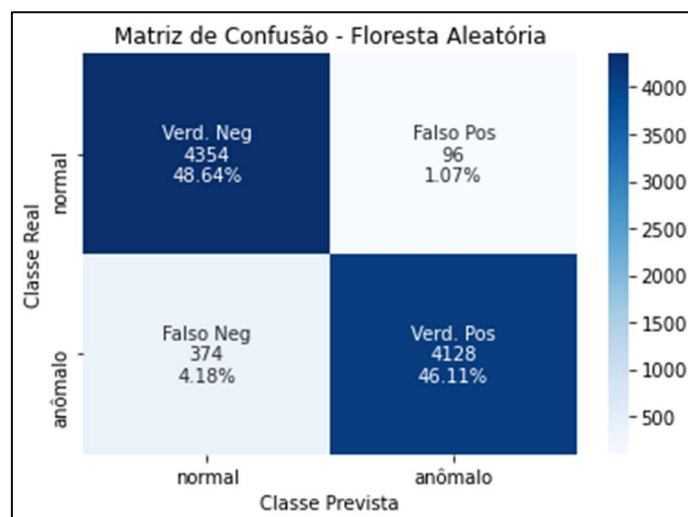
Os modelos foram implementados em uma CPU com Intel Core i7 de 2.6 GHz e 8 Gb de RAM através da distribuição Anaconda da linguagem de programação Python, versão 3.7.9. O espaço em disco total ocupado pelo banco de dados utilizado foi de aproximadamente 700 Mb.

4 RESULTADOS

4.1 Modelo de Floresta Aleatória

A Figura 6 apresenta a matriz de confusão resultante do teste do modelo de floresta aleatória em dados desconhecidos.

Figura 6 – Matriz de confusão resultante da classificação dos dados desconhecidos para o modelo de Floresta Aleatória.



A partir dos valores de classificação obtidos, fica evidente a elevada taxa de acerto atingida pelo modelo, assim como uma maior dificuldade na classificação de amostras anômalas. Tal dificuldade pode ser analisada através das métricas de avaliação definidas anteriormente, cujos valores são apresentados na Tabela 7.

Tabela 7 – Resultados das métricas de avaliação e tempo de treinamento para o modelo de Floresta Aleatória.

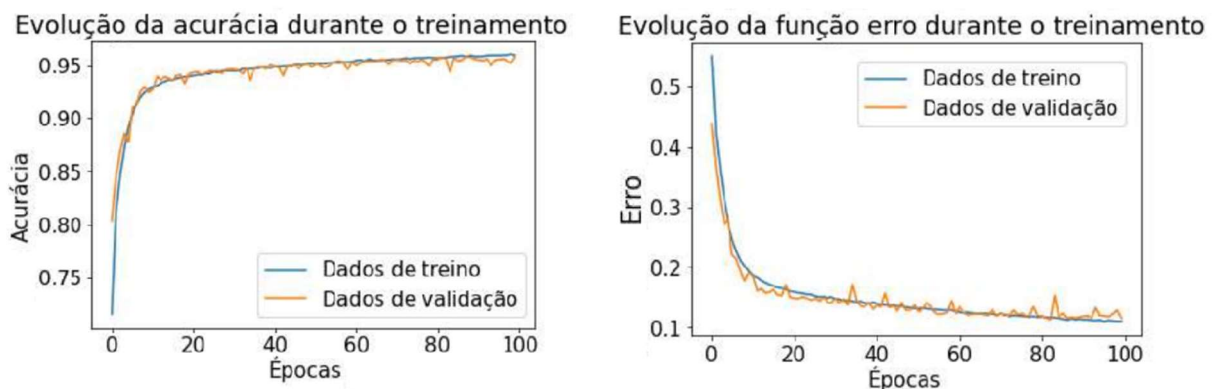
Métricas de avaliação e tempo de treinamento – Floresta Aleatória				
Acurácia	Precisão	Recall	F1-Score	Tempo de treinamento
95%	98%	92%	95%	17 s

A avaliação das métricas revela que, embora o modelo tenha sido treinado com sucesso para realizar classificações altamente precisas, como evidenciado pela precisão de 98%, a maior dificuldade na detecção de amostras anômalas prejudica sua acurácia, como apresentado pela maior taxa de falsos negativos na matriz de confusão.

4.2 Modelo de Rede Neural Percéptron

O modelo de rede neural construído foi testado sob diferentes configurações de parâmetros de forma a se encontrar aquela melhor adequada para os dados estudados, ou seja, aquela que apresente uma diminuição da função erro durante a evolução do treinamento e um aumento adequado da taxa de acerto. Com os testes, foi definida uma taxa de aprendizado, ou *learning rate*, de 1×10^{-4} , um *batch size* de 8 amostras e um número de épocas de 100. Tais parâmetros representam, respectivamente, a taxa de atualização dos pesos, a quantidade de amostras analisadas a cada iteração e o número de vezes em que o conjunto total de amostras do banco de dados será analisado. Na Figura 7 são apresentadas a evolução da taxa de acerto e da função erro durante o treinamento.

Figura 7 – Evolução da acurácia das classificações e função erro durante o treinamento do modelo.



Através da interface de aprendizado de máquina *Keras* para *TensorFlow*, é possível se controlar a evolução da taxa de acerto e da função erro, de forma que os pesos finais a serem utilizados pelo modelo treinado serão aqueles que apresentarem o menor erro durante o treinamento. Após a finalização do treinamento, as amostras de teste são fornecidas ao modelo para avaliar sua performance em dados desconhecidos. A Figura 8 e a Tabela 8 apresentam a

matriz de confusão da classificação dos dados desconhecidos e as métricas resultantes, respectivamente.

Figura 8 – Matriz de confusão resultante da classificação dos dados desconhecidos para o modelo de Rede Neural.

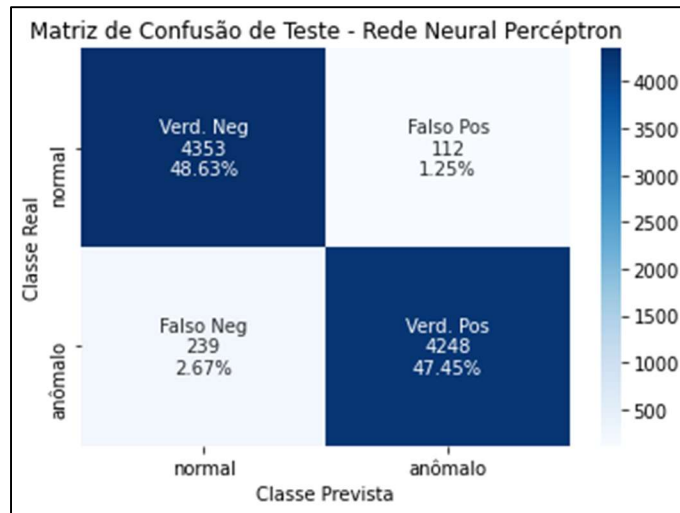


Tabela 8 – Resultados das métricas de avaliação e tempo de treinamento para o modelo de Rede Neural.

Métricas de avaliação e tempo de treinamento – Rede Neural Percéptron				
Acurácia	Precisão	Recall	F1-Score	Tempo de treinamento
96%	97%	95%	96%	620 s

Através das métricas de avaliação e da matriz de confusão percebe-se que o melhor desempenho da rede neural se deve ao aumento do *recall* através da menor classificação de falsos negativos, em relação aos resultados do modelo de floresta aleatória.

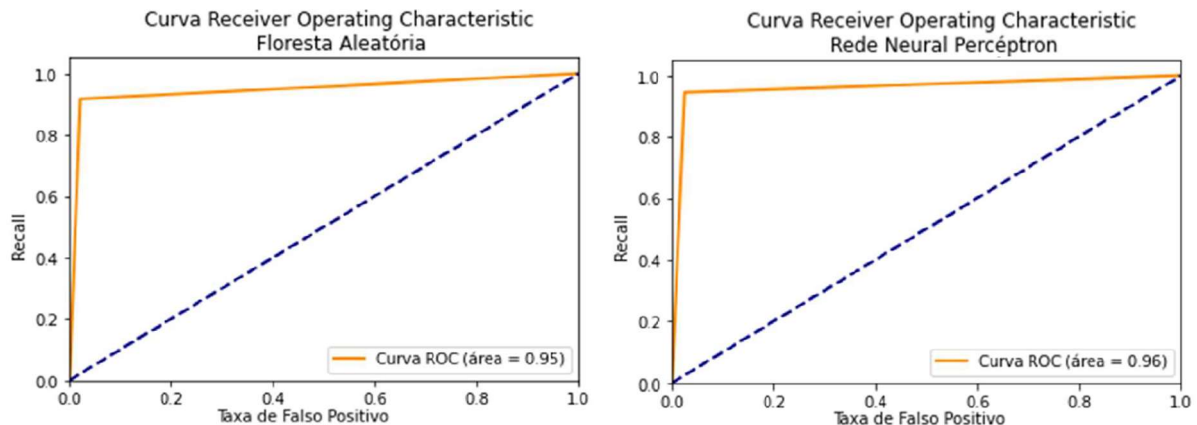
4.3 Comparação de performance e curvas ROC

Dos dados e métricas apresentados até o momento fica evidente que ambos os modelos treinados realizaram a tarefa de classificação de anomalias acústicas, dentro das devidas restrições impostas ao banco de dados, com elevado sucesso.

Com o intuito de selecionar um dos modelos para a aplicação na tarefa de detecção de anomalias acústicas, pode-se usar as curvas ROC de cada modelo para condensar de maneira eficiente as informações obtidas até então para cada modelo. A Figura 9 apresenta uma comparação dessas curvas para o modelo de floresta aleatória e de rede neural Percéptron.

Através da comparação das curvas é possível perceber que o melhor desempenho da rede neural Percéptron foi devido à maior prevenção de falsos negativos, aumentando assim seu *recall*. Este foi o principal fator responsável para que este modelo tenha apresentado o maior valor de AUC dentre os dois modelos construídos.

Figura 9 – Comparação das curvas ROC e parâmetro AUC dos modelos treinados.



5 CONCLUSÃO

O presente trabalho teve como objetivo investigar o desempenho de duas opções clássicas de algoritmos de aprendizado de máquina, aplicados com uma técnica específica de extração de parâmetros acústicos, na tarefa de detectar e classificar falhas em um equipamento industrial específico através de gravações do som deste equipamento, dentro das condições delimitadas pelo banco de dados selecionado. Os resultados de performance obtidos para ambos os métodos foram comparáveis a resultados considerados excelentes na literatura. Tais resultados comprovam o grande potencial que esta tecnologia possui para o desenvolvimento de técnicas não invasivas de controle e prevenção de falhas em equipamentos e processos industriais indispensáveis no mundo moderno. Porém, deve-se destacar que o uso limitado do banco de dados restringe os modelos treinados apenas a testes de comportamento, não bastando para um uso em campo para detecção de anomalias genéricas não presentes no banco de dados usado.

Comparados a outros métodos de extração de parâmetros acústicos, como o uso direto das amplitudes sonoras para o treinamento, ou o uso dos espectrogramas mostrados anteriormente para a realização de um treinamento a partir de imagens, o método de extração dos Coeficientes Cepstrais na Escala Mel e o uso conjunto de suas derivadas temporais possui diversas vantagens. A partir de um número relativamente pequeno de amostras sonoras, foi possível se obter um banco de dados de excelente volume e com dezenas de diferentes parâmetros presentes por amostra.

Deve-se ressaltar que, embora as métricas de performance e as curvas ROC obtidas pareçam sugerir que o algoritmo ideal para a detecção deste tipo de falha seja a rede neural, muitos fatores devem ser levados em consideração antes de tal escolha. Como apresentado, o tempo de treinamento da rede neural foi uma ordem de grandeza maior, logo, o poder computacional disponível, o tamanho do banco de dados a ser utilizado e o tempo de treinamento devem ser pesados frente à confiabilidade e acurácia almejada. Porém, é importante lembrar que o modelo de rede neural utilizado foi um dos mais simples existentes. Logo, diferentes técnicas e modificações podem ser realizadas para otimizar tanto a performance de classificação do modelo quanto sua performance computacional.

De forma a melhor se comprovar a eficácia da metodologia utilizada e aumentar a generalização dos modelos para diferentes equipamentos e situações, algumas propostas para futuros trabalhos podem ser levantadas. Uma análise mais detalhada dos resultados, com o uso de um banco de dados maior e mais abrangente pode ser realizada; o teste de arquiteturas mais modernas de rede neural, como redes recorrentes, pode ser testado; a implementação dos modelos para detecção em tempo real de anomalias sonoras pode ser feita, de forma a se investigar os ganhos reais que o uso desta tecnologia pode trazer a um processo, dentre muitos outros temas.

REFERÊNCIAS BIBLIOGRÁFICAS

ABADI, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. 14 mar. 2016.

AJIBOLA ALIM, S.; KHAIR ALANG RASHID, N. **Some Commonly Used Speech Feature Extraction Algorithms From Natural to Artificial Intelligence - Algorithms and Applications**, 2018.

BREIMAN, L. Random Forests. **Machine Learning**, v. 45, p. 5–32, 2011.

DENG, M. et al. Heart sound classification based on improved MFCC features and convolutional recurrent neural networks. **Neural Networks**, v. 130, p. 22–32, 2020.

DUMAN, T. B.; BAYRAM, B.; İNCE, G. **Acoustic Anomaly Detection Using Convolutional Autoencoders in Industrial Processes Advances in Intelligent Systems and Computing**, 2020.

HOTHU, H.; MITA, A. Damage Detection Method Using Support Vector Machine and First Three Natural Frequencies for Shear Structures. **Open Journal of Civil Engineering**, v. 03, n. 02, p. 104–112, 2013.

KLINE, D. M.; BERARDI, V. L. Revisiting squared-error and cross-entropy functions for training neural network classifiers. **Neural Computing and Applications**, v. 14, n. 4, p. 310–318, 2005.

LERCH, A. **An introduction to audio content analysis: Applications in signal processing and music informatics**. [s.l.] Wiley-IEEE Press, 2012.

LUCIAN, L. M. **Fundamentos de Aprendizagem de Máquina**. 1. ed. Porto Alegre: SAGAH, 2020.

MCFFEE, B. et al. librosa/librosa: 0.8.0. 22 jul. 2020.

MÜLLER, A., & GUIDO, S. **Introduction to Machine Learning with Python: A Guide for Data Scientists 1st Edition**. O'Reilly Media. [s.l.] O'Reilly Media, 2018. v. 53

PEDREGOSA ET AL. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

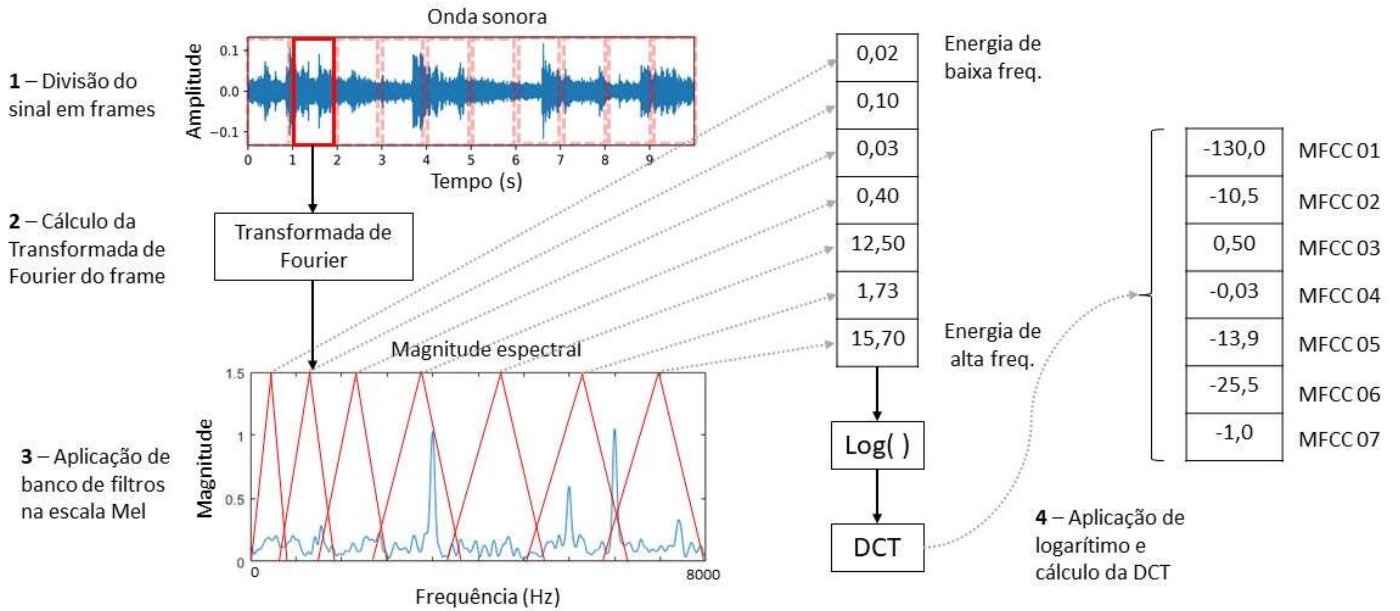
PUROHIT, H. et al. MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection. **arXiv**, 2019.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, 1958.

VARGAS-MACHUCA, J.; GARCÍA, F.; CORONADO, A. M. Detailed Comparison of Methods for Classifying Bearing Failures Using Noisy Measurements. **Journal of Failure Analysis and Prevention**, v. 20, n. 3, p. 744–754, 2020.

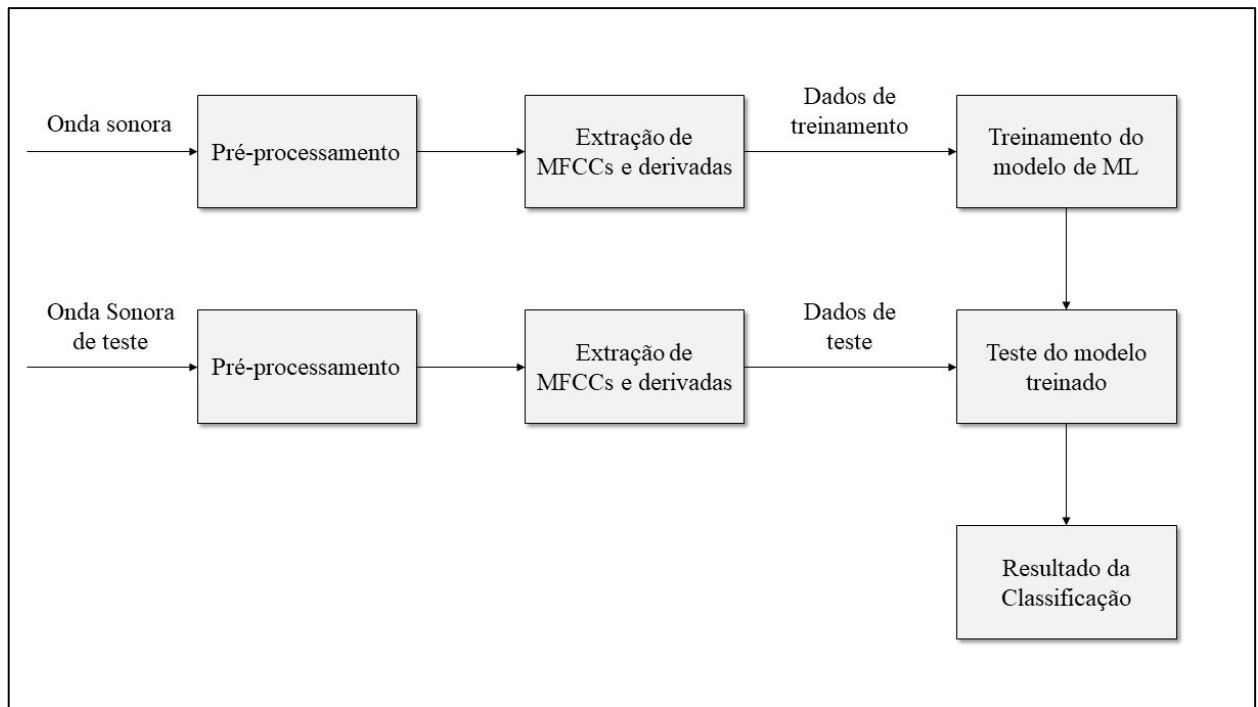
APÊNDICE A

Figura A.1 – Exemplo esquemático do algoritmo de extração de coeficientes MFCC de acordo com os passos apresentados na seção 2.2.



APÊNDICE B

Figura B.1 – Fluxograma dos algoritmos de ML desenvolvidos.



Adaptado de DENG et al., 2020

APÊNDICE C

Figura C.1 – Comparação da variação dos 13 coeficientes MFCC para uma bomba hidráulica em operação normal e em operação com falhas.

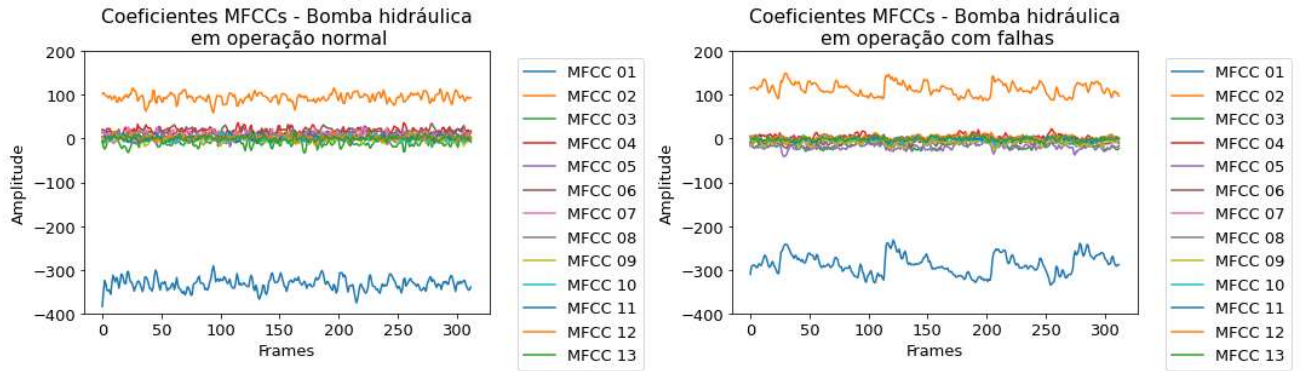


Figura C.2 – Comparação da variação das derivadas dos 13 coeficientes MFCC para uma bomba hidráulica em operação normal e em operação com falhas.

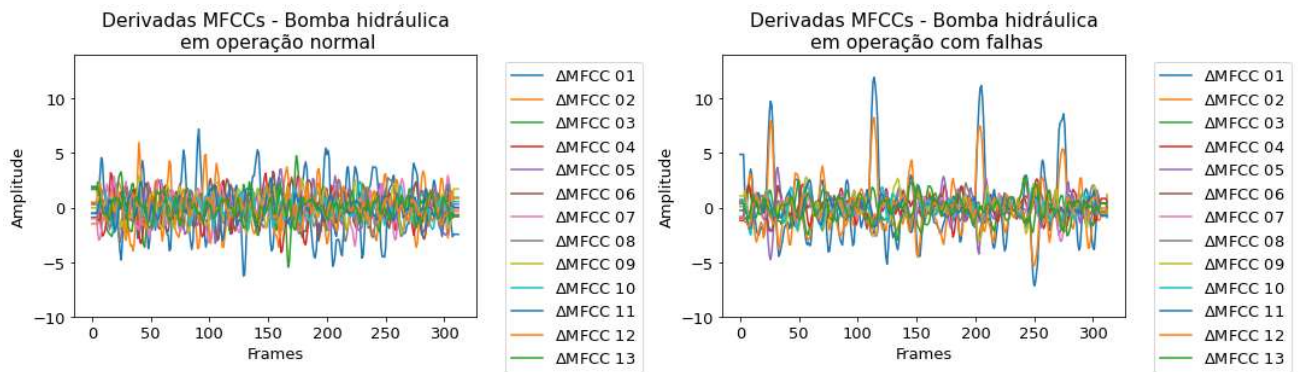


Figura C.3 – Comparação da variação das derivadas de segunda ordem dos 13 coeficientes MFCC para uma bomba hidráulica em operação normal e em operação com falhas.

