

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

APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE ANOMALIAS EM
MÁQUINAS ROTATIVAS POR SINAIS DE VIBRAÇÃO

por

Klaus Jürgen Folz

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, novembro de 2021.

CIP - Catalogação na Publicação

Folz, Klaus Jürgen

APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE ANOMALIAS
EM MÁQUINAS ROTATIVAS POR SINAIS DE VIBRAÇÃO. / Klaus
Jürgen Folz. -- 2021.
14 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. Monitoramento de
vibração. 3.Vibração. 4. detecção de falha. 5.
manutenção preditiva. I. Gomes, Herbert Martins,
orientador. III. Título.

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

Klaus Jürgen Folz

APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE ANOMALIAS EM MÁQUINAS
ROTATIVAS POR SINAIS DE VIBRAÇÃO

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. Dr. Cirilo Sperb Bresolin
Coordenador(a) do Curso de Engenharia Mecânica

Área de Concentração: Mecânica dos Sólidos

Orientador: Prof. Dr. Herbert Martins Gomes

Comissão de Avaliação:

Prof. Dr. Herbert Martins Gomes

Prof. Dr. Felipe Tempel Stumpf

Prof. Dr. Rodrigo Rossi

Porto Alegre, novembro de 2021.

AGRADECIMENTOS

Agradeço a todos os meus colegas da universidade que fizeram parte da minha jornada acadêmica e me apoiaram através de sua companhia, conversas e risadas. Expresso o meu profundo agradecimento aos meus colegas de trabalho da empresa TK Elevator, cujo profissionalismo, comprometimento e incentivo tornaram esse trabalho possível e alavancaram meu desenvolvimento profissional e acadêmico. Agradeço ao Departamento de Engenharia Mecânica e todos os seus colaboradores pelo privilégio que me foi dado de estudar e aprender ao longo dos últimos anos aqui. Agradeço à minha namorada, por sua companhia e paciência. Por fim, agradeço à minha família, especialmente à minha mãe, pai e avó, sem o apoio de vocês, a conclusão dessa graduação não teria sido possível.

“ Wenn Sie die Art und Weise ändern, wie Sie die Dinge betrachten, ändern sich die Dinge, die Sie betrachten.“

Max Planck

**APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE ANOMALIAS EM
MÁQUINAS ROTATIVAS POR SINAIS DE VIBRAÇÃO**

Klaus Jürgen Folz
klausjurgenfolz@gmail.com

Resumo. *A manutenção de máquinas requer o contínuo aperfeiçoamento de técnicas e equipamentos para o monitoramento de parâmetros de funcionamento dos equipamentos. Os ganhos financeiros em evitar problemas catastróficos e em cascata em plantas industriais devido às falhas supera, em muito, gastos com monitoramento investimentos em novas tecnologias. Em máquinas rotativas, a vibração é um dos parâmetros mais investigados, visto que a ele está geralmente associado desbalanceamentos, desalinhamentos, defeitos em mancais e rolamentos. O uso de técnicas de aprendizado de máquina vem se tornando uma área de intensa pesquisa, muito em parte do relativo sucesso destas metodologias na classificação de defeitos, prognósticos de tempo de vida útil, além da possibilidade de tratamento online e em tempo real do monitoramento de máquinas. Este trabalho propõe a análise de técnicas existentes de aprendizado de máquinas para a classificação de defeitos em máquinas baseando-se em medidas de sinais de vibração e sinal sonoro do motor tomadas em pontos estratégicos das máquinas monitoradas. Prevê-se a construção de ferramentas que poderão prever e classificar defeitos em máquinas com acurácia e eficiência*

Palavras-chave: *Aprendizado de máquina, monitoramento de vibração, detecção de falha, manutenção preditiva*

Machine learning for detection of anomalies in rotary machines by vibration signs

Abstract. *Machine maintenance requires continuous improvement of techniques and equipment for monitoring equipment operating parameters. The financial gains in avoiding catastrophic and cascading problems in industrial plants due to failures far outweigh the expenses with monitoring investments in new technologies. In rotating machines, vibration is one of the most investigated parameters, as it is generally associated with unbalances, misalignments, defects in bearings and bearings, etc. The use of machine learning techniques has become an area of intense research, largely due to the relative success of these methodologies in defect classification, lifetime predictions, in addition to the possibility of online and real-time processing of monitoring machines. This work proposes the analysis of existing machine learning techniques for classifying machine defects based on measurements of vibration signals and sound taken at strategic points of monitored machines. The construction of tools that will be able to predict and classify machine defects with relative accuracy and efficiency is foreseen.*

Keywords: *Machine learning, vibration monitoring, failure detection, predictive maintenance*

NOMENCLATURA

Símbolos

A	Constante do modelo de Johnson-Cook	[MPa]
f	Frequência da onda	[Hz]
f_{mel}	Frequência na escala de Mel	[m]
C	Parâmetro de Regularização	
K	Kernel	
$c(n)$	N-ésimo coeficiente cepstral de mel	
L	Função Lagrangiana	
$H_m(k)$	Peso do k-ésimo espectro de energia	
$X(k)$	Espectro de magnitude	
\vec{w}	Vetor normal ao hiperplano	
\vec{u}	Vetor amostral	
\vec{x}_+	Vetor de suporte positivo	
\vec{x}_-	Vetor de suporte negativo	
M	Número total de filtros triangulares ponderados de Mel	

Símbolos gregos

α_j, α_i	Multiplicadores de Lagrange
φ	Função de Kernel
γ	Parâmetro Gama

Abreviaturas e acrônimos

AUC	<i>Area Under the Curve</i>
DCT	<i>Discrete Cosine Transform</i>
MFCC	<i>Mel Frequency Cepstral Coefficients</i>
ML	<i>Machine Learning</i>
RF	<i>Random Forests</i>
ROC	<i>Receiver Operating Characteristics</i>
SVM	<i>Support Vector Machines</i>

SUMÁRIO

1. INTRODUÇÃO.....	1
1.1. Motivação	1
1.2. Objetivos.....	1
2. REVISÃO BIBLIOGRÁFICA.....	1
3. FUNDAMENTAÇÃO.....	2
3.1. Máquina de Vetores de Suporte.....	2
3.1.1. Algoritmo	2
3.1.2. Kernel	4
3.1.3. Parâmetro de Regularização (C).....	5
3.2. Florestas Aleatórias.....	5
3.3. Coeficientes Cepstrais das Frequências de Mel.....	6
3.4. Acurácia	7
3.5. Escore F1, Precisão e Recall.....	7
3.6. Curva Característica de Operação do Receptor	8
3.7. Matriz de Confusão.....	8
4. METODOLOGIA.....	8
4.1. Banco de dados	9
4.2. Processamento de dados	10
4.3. Seleção e Parametrização dos Modelos	11
4.3.1. Pesquisa em Grade para Otimização dos Parâmetros.....	11
4.4. Resultados.....	12
4.4.1. Máquina de Vetores de Suporte.....	12
4.4.2. Florestas Aleatórias	13
5. CONCLUSÕES	15
REFERÊNCIAS BIBLIOGRÁFICAS	16
APÊNDICE A	18
APÊNDICE B.....	19
APÊNDICE C.....	20
APÊNDICE D	21

LISTA DE FIGURAS

Figura 1 - Matriz de confusão do modelo SVM com MFCCs calculados a partir dos sinais de vibração dos acelerômetros (a) e sinal de áudio (b).	13
Figura 2 - Curva ROC para o modelo SVM com MFCCs calculados a partir dos sinais de vibração dos acelerômetros (a) e sinal de áudio (b).	13
Figura 3 - Matriz de confusão do modelo Florestas Aleatórias com MFCCs calculados a partir dos sinais de vibração dos acelerômetros (a) e sinal de áudio (b).	14
Figura 4 - Curva ROC para o modelo de Florestas Aleatórias com MFCCs calculados a partir dos sinais de vibração dos acelerômetros (a) e sinal de áudio (b).	15
Figura 5 - Hiperplano e margens geradas pelo algoritmo SVM para um problema de classificação binário	18
Figura 6 – Mapeamento do espaço dos atributos para o espaço das características de alta dimensão.....	18
Figura 7 – Margens e hiperplanos obtidos via modelo SVM em um problema de classificação binário para valores do parâmetro de regularização (C) iguais a 10 e 100	19
Figura 8 - Esquema do algoritmo de florestas aleatórias.....	19
Figura 9 - Exemplo de Curva ROC e parâmetro AUC.....	20
Figura 10 - modelo para interpretação dos resultados da curva ROC, para comparação da performance entre diferentes modelos.....	20
Figura 11 - Simulador de Falha de Maquinário (MFS) da empresa <i>SpectraQuest</i> , Alinhamento-Equilíbrio-Vibração ABVT (Fonte: <i>SpectraQuest, Inc</i>).....	21

LISTA DE TABELAS

Tabela 1 - Pseudocódigo do algoritmo de máquina de vetores de suporte. (Adaptado de PEDERSEN e SCHOEBERL, 2008).	4
Tabela 2 - Pseudocódigo do algoritmo de Florestas Aleatórias (Adaptado de SIRIKULVIRIYA e SINTHUPINYO, 2011).	6
Tabela 3 – Esquema de Matriz de Confusão para um modelo classificador binário.....	8
Tabela 4 - Especificações do Motor e Rolamentos utilizados (Adaptado de MAFUALDA, 2014).	9
Tabela 5 - Condições da totalidade dos testes realizados (Adaptado de MAFUALDA, 2014)	10
Tabela 6 – Segregação do conjunto original em subconjuntos de treino e teste para validação dos algoritmos.	11
Tabela 7 – Valores para cada parâmetro testados para o modelo SVM.....	12
Tabela 8 - Valores para cada parâmetro testados para o modelo de Florestas Aleatórias.....	12
Tabela 9 – Escore F1 para o modelo SVM utilizando MFCCs calculados a partir dos Sinais de Vibração (Acelerômetros) e sinal de áudio (Microfone).....	13
Tabela 10 - Escore F1 para o modelo de Florestas Aleatórias utilizando MFCCs calculados a partir dos Sinais de Vibração (Acelerômetros) e sinal de áudio (Microfone).	14

1. INTRODUÇÃO

As máquinas rotativas modernas estão presentes nas mais diversas atividades, nas quais se exige força e torque para a movimentação de turbinas, caldeiras, esteiras e outros componentes. Os problemas mais comuns em máquinas rotativas dizem respeito a fenômenos que envolvem seu intrínseco funcionamento quanto a eixos, partes e massas rotativas. Assim problemas como desalinhamento, desbalanceamento e problemas de acoplamento podem levar estes equipamentos a condições perigosas de uso ou mesmo à sua falha prematura. A exata identificação de anomalias de funcionamento destes equipamentos pode trazer a possibilidade de correção prematura e, portanto, extensão do tempo de vida em serviço, ou até mesmo evitar falhas mais catastróficas ou paradas frequentes para conserto.

1.1. Motivação

A eficiência e confiabilidade de máquinas rotativas são muito relevantes em diversas aplicações industriais. De modo que, a falha de um equipamento desse tipo pode ser um problema sério em termos de segurança ou custos. Para que resultados adequados sejam atingidos, aplicam-se métodos de manutenção preditiva, que por meio do monitoramento de condição constante, possibilitam a previsão e a detecção de falhas. Práticas de manutenção preditiva, quando comparadas a métodos tradicionais de manutenção, provocam o aumento do tempo médio entre cada revisão, redução do estoque de peças de reposição e a minimização de paradas para manutenção corretiva (ARALTO, 2004). Atualmente as técnicas de monitoramento de condição geram grandes quantidades de dados, exigindo a automatização dos procedimentos de diagnóstico, o que leva à necessidade de desenvolvimento de métodos de interpretações autônomas de dados e de monitoramento de condição de vibração. Para este fim, algoritmos de aprendizado de máquina são aplicados e aprimorados para processar os dados e classificar o funcionamento do equipamento em tempo real, sendo este anômalo ou não.

1.2. Objetivos

Utilizando dados de vibração disponibilizados pelo Laboratório de Sinais, Multimídia e Telecomunicações (SMT) da Universidade Federal do Rio de Janeiro (UFRJ) na *Machinery Fault Database* (MAFAULDA, 2014), o objetivo do presente trabalho é desenvolver um modelo de aprendizado de máquina em linguagem Python para classificação de falhas em máquinas rotativas, para este fim se faz necessário avaliar o desempenho dos algoritmos Máquinas de Vetores de Suporte (*Support Vector Machine*, SVM) e Floresta Aleatória (*Random Forest*, RF) na classificação e detecção dessas anomalias, utilizando os coeficientes cepstrais das frequências de Mel como parâmetros para treino. Os dados de vibração e som são fornecidos pela base de dados aberta *online Machinery Fault Database* (MAFAULDA). Os coeficientes são calculados a partir dos sinais de vibração de dois acelerômetros de 3 canais instalados em um rolamento do eixo de um motor de $\frac{1}{4}$ CV. Prevê-se avaliar os algoritmos propostos com base nas métricas de *Receiver Operating Characteristics Curve* (Curva ROC), Acurácia Média, Pontuação F1 e Matrizes de Confusão. Por fim, disponibilizar o modelo desenvolvido utilizando o algoritmo de maior performance para implementação na indústria.

2. REVISÃO BIBLIOGRÁFICA

Aprendizado de Máquina (*Machine Learning*, ML) é o termo genérico usualmente empregado para designar uma série de técnicas e algoritmos que possuem a capacidade de classificar, e adquirir informação a partir de dados e informações escassas. Especificamente na avaliação de vibrações, HANG et al. (2017) avaliaram o desempenho dos algoritmos: Máquina de Suporte de Vetor e Algoritmo de *clustering* rápido, combinados com o método decomposição de modo variacional e análise de componente principal, na manutenção preditiva

de máquinas rotativas desbalanceadas. Os autores concluíram que o modelo seria capaz de diagnosticar efetivamente as falhas causadas por desbalanceamento no maquinário rotativo.

Um modelo de predição para prevenção proativa das falhas em um acelerador de gravitacional foi proposto por LEE et al. (2021), utilizando um acelerômetro de 4 canais acoplado ao alojamento do mancal. O algoritmo foi treinando com um modelo de aprendizagem profunda (*Deep Learning*), convertendo o sinal de vibração em uma transformada de Fourier de curta duração (*Short Term Fourier Transform*, STFT) e em um espectrograma de coeficientes cepstrais de frequência de Mel (*Mel Frequency Cepstral Coefficients*, MFCC) convertendo o resultado em uma imagem bidimensional. Como resultado, foi possível quantificar a gravidade do desbalanceamento observando as áreas de defeito que não podem ser vistas com imagens unidimensionais.

PITTINO et al. (2020) conduziram uma investigação de diferentes métodos estatísticos de aprendizado de máquina e de gráficos de controle para a detecção automática de anomalias na operação de um rolamento rotativo em uma máquina comercial de fabricação de semicondutores, tanto em operação quanto em condições de desgaste. Os desenvolvedores mostraram que tanto os gráficos de controle quanto os métodos de classificação totalmente supervisionados apresentaram desempenho muito semelhante, condicionados diretamente à qualidade dos dados de treino.

Um modelo de aprendizagem de máquina não supervisionado para monitoramento de condições de máquinas rotativas usando uma abordagem de detecção de anomalias foi proposto por AHMAD et al. (2021). A eficácia foi avaliada comparando os resultados gerados automaticamente com um conjunto de dados manuais treinando um modelo de Floresta de Isolamento alcançando uma pontuação média de F1-score, uma métrica para avaliar a eficiência de um classificador e que combina o *Recall* e Precisão de que retratem em um único número, de 99,6%.

3. FUNDAMENTAÇÃO

3.1. Máquina de Vetores de Suporte

O SVM é um classificador de aprendizagem supervisionado, empregado tanto em classificação como para análise de regressão, *clustering* e outras aplicações de aprendizado de máquina utilizando um conceito geométrico denominado hiperplano, empregado como limite de decisão. Por se tratar de um algoritmo de ML supervisionado, se faz necessário o fornecimento de um conjunto de dados de treino rotulado, possibilitando o cálculo do hiperplano para classificação.

O objetivo do SVM é calcular um hiperplano em um espaço N-dimensional (onde N é o número de atributos) com o intuito de classificar (separar) distintamente os pontos de dados, um hiperplano genérico para um problema de classificação binário é exibido na Figura 5 no Anexo A. Contudo, para um determinado conjunto de dados, existem distintos planos que podem ser aplicados e separam com sucesso as duas classes. A definição de qual hiperplano será utilizado se origina na maximização da separação do hiperplano e de seus pontos (atributos) mais próximos, também chamado de margem. A robustez do algoritmo está diretamente ligada à necessidade de que a separação entre os pontos e o hiperplano seja a maior possível para todas as classes. Dessa forma, o objetivo se torna calcular o plano que tem a máxima distância entre os pontos de dados das classes.

3.1.1. Algoritmo

Se um vetor \vec{w} é sempre normal ao hiperplano, é possível calcular a distância de uma amostra ao limite de decisão utilizando a projeção do vetor da amostra \vec{u} sobre o vetor \vec{w} . Caso essa projeção for maior que um determinado valor c , o vetor da amostra \vec{u} assume um valor

positivo, como mostra a Equação 1, caso contrário, irá assumir um valor negativo. A projeção é proporcional ao produto interno do primeiro vetor pelo segundo. Na sequência, uma constante b é introduzida na Equação 1, onde $b = -c$, resultando na Equação 2. O Passo seguinte se resume simplesmente em forçar que a função de decisão retorne um valor maior ou igual a 1 para amostras positivas, conforme a Equação 3 e 4, e -1 para amostras negativas.

$$\vec{w} \vec{u} \geq c \quad (1)$$

$$\vec{w} \vec{u} + b \geq 0 \quad (2)$$

$$\vec{w} \vec{x} + b \geq 1 \quad (3)$$

$$\vec{w} \vec{x} + b \geq -1 \quad (4)$$

Com base nas equações anteriores fica evidente que amostras localizadas entre o limite de decisão e os vetores de suporte não serão consideradas, em outras palavras, caso o vetor \vec{w} seja igual a um vetor de suporte, o valor do lado esquerdo da equação será 0. Para consolidar as equações 3 e 4 é então adicionado uma variável y , resultando na equação 5, onde y é igual 1 para amostras positivas e -1 para amostras negativas, conforme a Equação 6.

$$y_i(\vec{w} \vec{x}_i + b) - 1 \geq 0 \quad (5)$$

$$y_i = \begin{cases} +1, & \text{amostras +} \\ -1, & \text{amostras -} \end{cases} \quad (6)$$

Visando maximizar a margem e determinar uma equação para sua largura, subtrai-se um vetor de suporte positivo \vec{x}_+ de um negativo \vec{x}_- e multiplicamos pelo vetor unitário \vec{w}_u , obtido pela divisão do vetor \vec{w} pela sua magnitude (Equação 7).

$$\text{largura} = \frac{(\vec{x}_+ - \vec{x}_-) \vec{w}}{\|\vec{w}\|} \quad (7)$$

Aplicando as equações 3 e 4 na equação 7 é possível encontrar as equações 8 e 9. Portanto, para selecionar o limite de decisão ideal, precisamos maximizar a Equação 9. A máxima largura é proporcional ao inverso do máximo da magnitude do vetor \vec{w} , sendo equivalente ao mínimo do mesmo, como ilustra a Equação 10. A Equação 10 também pode ser calculada utilizando a Equação 11.

$$\text{largura} = \frac{((1 - b) - (-1 - b))}{\|\vec{w}\|} \quad (8)$$

$$\text{largura} = \frac{2}{\|\vec{w}\|} \quad (9)$$

$$\max \frac{2}{\|\vec{w}\|} \propto \max \frac{1}{\|\vec{w}\|} \equiv \min \|\vec{w}\| \quad (10)$$

$$\min \|\vec{w}\| = \min \frac{1}{2} \|\vec{w}\|^2 \cdot \frac{d}{dx} \frac{1}{2} x^2 = x \quad (11)$$

A função objetivo é convexa e os pontos que satisfazem as restrições formam um conjunto convexo, portanto esse problema possui um único mínimo global. Logo, pode ser solucionado com a introdução de uma função Lagrangiana, Equação 12, que engloba as restrições da função objetivo, associadas a parâmetros denominados multiplicadores de Lagrange (α_i).

$$L = \frac{1}{2} \|\vec{w}\|^2 - \sum_i^n \alpha_i [y_i(\vec{w} \vec{x}_i + b) - 1] \quad (12)$$

Derivando em relação a \vec{w} e igualando a equação 12 a zero (Equação 13), conclui-se que \vec{w} é uma combinação linear dos vetores \vec{x}_i . O valor de α_i será diferente de 0 somente para os

pontos que se encontram sobre os hiperplanos, isto é, as amostras que se situam mais próximas ao hiperplano, logo os demais pontos não participam do cálculo de \vec{w} . Os pontos que possuem $\alpha_i > 0$ são denominados vetores de suporte e configuram as amostras mais informativas do conjunto de treinamento. Substituindo a Equação 13 em 12, chega-se à Equação 14 evidenciando que a regra de decisão depende apenas do produto interno dos vetores da amostra pelo vetor desconhecido. Utilizando o mesmo procedimento para a constante b é possível deduzir a Equação 15.

$$\vec{w} = \sum_i^n \alpha_i y_i \vec{x}_i \quad (13)$$

$$\vec{w} = \sum_i^n \alpha_i y_i \vec{x}_i \vec{u} + b \geq 0 \quad (14)$$

$$\sum_i^n \alpha_i y_i = 0 \quad (15)$$

A Equação 16 é encontrada ao substituir as Equações 13, 14 e 15 na Equação 12. A Equação 16 é muito relevante para o algoritmo SVM, uma vez que a otimização depende do produto interno de pares de amostras e a dependência total da matemática envolvida é de produtos internos, o que permite a utilização da técnica do truque de *kernel* onde usam-se transformações das variáveis cujos respectivos produtos internos podem ser calculados eficientemente. O “truque de *kernel*” também permite que o espaço original dos dados a serem separados seja mapeado em um espaço de produto escalar de alta-dimensão, chamado espaço de características, onde os dados podem então ser linearmente separáveis, como evidencia a Figura 6 no Apêndice A. A Tabela 1 mostra um pseudocódigo de treinamento do SVM

$$L = \sum_i^n \alpha_i - \frac{1}{2} \sum_i^n \sum_j^n \alpha_i \alpha_j y_i y_j \vec{x}_i \vec{x}_j \quad (16)$$

Tabela 1 - Pseudocódigo do algoritmo de máquina de vetores de suporte. (Adaptado de PEDERSEN e SCHOEBERL, 2008).

Algoritmo 1, treinando um SVM

Solicitar: x e y carregados com dados rotulados de treinamento, um $\alpha \leftarrow 0$ ou $\alpha \leftarrow$ parcialmente treinado SVM

1: $C \leftarrow$ algum valor (10 por exemplo)

2: **Repetir**

3: **Laço:** (todos) $\{x_i, y_i\}, \{x_j, y_j\}$ **faça**

4: Otimize α_i e α_j

5: **Fim do Laço**

6: **Até** que nenhuma mudança em um ou outro critério de restrição de recurso seja atendido

Assegure: Reter apenas os vetores de suporte ($\alpha_i > 0$)

3.1.2. Kernel

Alguns algoritmos de aprendizado de máquina assumem que os dados de entrada são separáveis linearmente, como o SVM, contudo muitos problemas reais não apresentam essa característica linear. Apesar disso, como a Equação 16 depende exclusivamente de produtos internos é possível utilizar uma função não trivial e arbitrária para projetar os dados de entrada

de dimensão N em um espaço de dimensão M , onde: $M > N$. Qualquer modelo linear pode ser transformado em um modelo não linear aplicando o truque do kernel ao modelo: substituindo seus atributos (preditores) por uma função de kernel (Equação 17).

$$K(x, x') = \langle \varphi(x), \varphi(x') \rangle, \quad (17)$$

onde x e x' são os dados de entrada e $\langle . \rangle$ representa o produto interno. Dentre as diversas funções existentes, são utilizadas no escopo desse trabalho a função linear de kernel denotada pela Equação 18. A Função polinomial exibida na Equação 19, onde d representa o grau do polinômio e r uma constante, ambos devem ser determinados na construção do modelo. A função de kernel de base radial (RBF) na Equação 20.

$$K(x, x') = \langle x, x' \rangle \quad (18)$$

$$K(x, x') = (\gamma \langle x, x' \rangle + r)^d \quad (19)$$

$$K(x, x') = \exp(\gamma \|x - x'\|^2) \quad (20)$$

3.1.3. Parâmetro de Regularização (C)

Sem perda de generalidade, o parâmetro de regularização (c) pode ser interpretado como a flexibilidade em relação a *outliers*. De outra maneira, ele permite o controle sobre a compensação entre a acurácia de classificação e o modelo de generalização. De maneira geral, esse parâmetro configura a otimização do SVM no que diz respeito à tolerância de erros ao criar o modelo de aprendizado de máquina. Para valores elevados de C , é adotado um hiperplano de margem inferior, caso ele obtenha resultados mais precisos com relação aos pontos de treinamento. De mesmo modo, para valores baixos de C , o otimizador calcula um hiperplano de separação de margem maior, com maiores erros, contudo com ganhos consideráveis em performance, a Figura 7 no Apêndice A ilustra os resultados para valores de C iguais a 10 e 100, respectivamente, para um problema de classificação binária.

3.2. Florestas Aleatórias

HO (1995) idealizou e desenvolveu o primeiro algoritmo de Florestas Aleatórias (RF) usando um método de subespaço aleatório. Posteriormente o modelo foi aprimorado por Leo Breiman e Adele Cutler (*apud* Breiman, 2001). O algoritmo de florestas aleatórias é definido por BREIMAN (2001) como uma combinação de preditores de árvore de decisão, de modo que cada árvore depende unicamente dos valores de um conjunto de dados de entrada independente e com a mesma distribuição para todas as árvores do conjunto (floresta), a Figura 8 no Apêndice B apresenta um esquema do algoritmo de florestas aleatórias, destacando a combinação dos preditores de árvore de decisão. Desse modo, um subconjunto aleatório de um determinado tamanho é produzido a partir do espaço de possíveis atributos (dados de entrada) de divisão. Logo, a melhor divisão é o recurso selecionado deterministicamente desse subconjunto.

Um pseudocódigo para o algoritmo de floresta aleatória é mostrado na Tabela 2. Para a tarefa de classificar, o algoritmo classifica a instância simplesmente combinando todos os resultados de cada uma das árvores na floresta. Em outras palavras, o método usado para combinar os resultados pode ser tão simples como prever a classe obtida a partir do maior número de árvores, como por exemplo em uma votação por maioria. HORNING (2010) destaca a facilidade de parametrização do modelo, de modo que diferentes valores para os parâmetros são pouco influentes na performance e acurácia. Contudo, valores diferentes para o número total de árvores utilizadas na floresta e a máxima profundidade de cada árvore são comumente testados para avaliações minuciosas da precisão do modelo.

Tabela 2 - Pseudocódigo do algoritmo de Florestas Aleatórias (Adaptado de SIRIKULVIRIYA e SINTHUPINYO, 2011).

Algoritmo 2, Pseudocódigo para o algoritmo da floresta aleatória

Para gerar classificadores c:

- 1: **Lço:** de $i = 1$ a c **faça**
- 2: Amostra aleatória dos dados de treinamento D , substituir para produzir D_i
- 3: Criar um nó raiz, N_i contendo D_i
- 4: Chamar *Árvore* (N_i)
- 5: **Fim do Lço**
- 6: **Árvore (N):**
- 7: **Se** N contém instâncias de apenas uma classe, então
- 8: Retorna
- 9: **Caso contrário**
- 10: Selecione aleatoriamente $x\%$ dos possíveis recursos de divisão em N
- 11: Selecione o recurso F com o maior ganho de informação para dividir
- 12: Crie f nós filhos de N, N_1, N_2, \dots, N_f , onde F tem f valores possíveis (F_1, \dots, F_f)
- 13: **Lço:** de $i = 1$ a f fazer
- 14: Defina o conteúdo de N_i para D_i , onde D_i são todas as instâncias em N que correspondem a F_i
- 15: Chamar *Árvore* (N_i)
- 16: **Fim do Lço**

3.3. Coeficientes Cepstrais das Frequências de Mel

Mel Frequency Cepstral Coefficients (MFCCs) são atributos extraídos de um sinal ondulatório, amplamente aplicados em algoritmos de reconhecimento de voz. LERCH (2012) interpreta os coeficientes como uma descrição compacta da forma de um envelope espectral de um sinal. A técnica de extração de recursos do MFCC, conforme RAO e MANJUNATH (2017), engloba basicamente a divisão do sinal em janelas, a aplicação da Transformada Discreta de Fourier (DFT), a obtenção do logaritmo da magnitude e a distorção das frequências em uma escala de Mel, seguido pela aplicação da Transformada discreta do cosseno (DCT) inversa. A descrição detalhada das várias etapas envolvidas na extração do recurso MFCC é explicada abaixo.

1. **Divisão do sinal em janelas:** A análise deve sempre ser realizada em segmentos curtos nos quais o sinal é considerado estacionário.
2. **Espectro de Frequências (DFT):** Cada segmento é convertido em espectro de magnitude aplicando a DFT.
3. **Espectro de Mel:** O Espectro é calculado passando o sinal transformado de Fourier por um conjunto de filtros de passagem de banda conhecido como banco de filtros de Mel. A aproximação de Mel da frequência física pode ser expressa conforme a Equação 21. Onde f denota a frequência em Hz e f_{mel} a frequência na escala de Mel.

$$f_{mel} = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \quad (21)$$

4. **Banco de Filtros:** As frequências centrais dos filtros são normalmente espaçadas uniformemente no eixo da frequência. O espectro de Mel calculado a partir do espectro de magnitude $X(k)$ na Equação 22 é calculado multiplicando o espectro de magnitude por cada um dos filtros ponderados triangulares de Mel conforme a Equação 23, onde M é o número total de filtros triangulares ponderados de Mel e $H_m(k)$ é o peso dado ao k -ésimo espectro de energia que contribui para a m -ésima banda de saída, expresso na Equação 24.

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-\frac{j2\pi nk}{N}} \quad (22)$$

$$s(m) = \sum_{k=0}^{N-1} [|X(k)|^2 |H_m(k)|] \quad (23)$$

$$H_m(k) = \begin{cases} 0, & k < f(m-1) \\ \frac{2(k-f(m-1))}{f(m+1)-f(m)}, & f(m-1) \leq k \leq f(m) \\ \frac{2(k-f(m-1))}{f(m+1)-f(m)}, & f(m) < k \leq f(m+1) \\ 0, & k > f(m+1) \end{cases} \quad (24)$$

5. **Transformada discreta do cosseno:** A DCT é aplicada aos coeficientes de frequência de Mel transformados e produz um conjunto de coeficientes cepstrais, os MFCCs, calculados conforme a Equação 25. Onde $c(n)$ são os coeficientes cepstrais e C é o número de MFCCs. Os sistemas MFCC tradicionais usam apenas 8 a 13 coeficientes cepstrais. O coeficiente zero é frequentemente excluído, pois representa a energia logarítmica média do sinal de entrada, que transporta apenas poucas informações específicas.

$$c(n) = \sum_{m=0}^{M-1} \log_{10}(s(m)) \cos\left(\frac{\pi n(m-0.5)}{M}\right) \quad (25)$$

3.4. Acurácia

A Acurácia de um algoritmo pode ser definida como a razão entre as previsões corretas e o número total de previsões realizadas. Para um problema com apenas duas classes a Equação 26 pode ser aplicada para cálculo da acurácia.

$$Acurácia = \frac{V_{positivos} + V_{negativos}}{(V_{positivos} + V_{negativos} + F_{positivos} + F_{negativos})} \quad (26)$$

Onde $V_{positivos}$ são as ocorrências positivas (pertencentes a classe) previstas corretamente, $V_{negativos}$ representa as previsões negativas corretas, $F_{negativos}$ denota as previsões negativas classificadas incorretamente pelo modelo, de mesmo modo $F_{positivos}$ representa as previsões positivas errôneas.

A Equação 26 possibilita o cálculo da acurácia para uma única classe. Como o presente trabalho propõe a classificação de diferentes classes de falhas em máquinas rotativas, é necessário generalizar a Equação 27 para n-classes. Para este fim, calcula-se simplesmente a acurácia para cada classe separadamente, seguido do cálculo da média aritmética para todas as categorias, retornando acurácia média como resultado.

$$Acurácia Média = \frac{1}{N} \sum_{i=1}^N Acurácia_i \quad (27)$$

3.5. Escore F1, Precisão e Recall

A precisão é calculada pela Equação 28 e representa a razão entre o número de resultados positivos verdadeiros e o número de todos os resultados positivos, incluindo os falsos positivos.

A Equação 29 representa o *recall*, sendo o número de resultados positivos verdadeiros dividido pelo número de verdadeiros positivos somados com o número de falsos negativos. A pontuação F1 na Equação 30 é simplesmente a média harmônica da precisão e recall.

$$\text{Precisão} = \frac{V_{\text{positivos}}}{(V_{\text{positivos}} + F_{\text{positivos}})} \quad (28)$$

$$\text{Recall} = \frac{V_{\text{positivos}}}{(V_{\text{positivos}} + F_{\text{negativos}})} \quad (29)$$

$$\text{Escore F1} = \frac{2}{(\text{Precisão}^{-1} + \text{Recall}^{-1})} \quad (30)$$

3.6. Curva Característica de Operação do Receptor

A Curva Característica de Operação do Receptor (*Receiver Operating Characteristic Curve*, ROC) representa graficamente a performance de um modelo classificador, cujos pontos são definidos pelo Recall (Equação 29) no eixo ordenado e pela taxa de falsos positivos (False Positive Rate, FPR) no eixo das abscissas, definida pela razão entre falsos positivos e número total de negativos, para vários valores do limiar de classificação. A Figura 9 no Apêndice C exibe um exemplo de curva ROC de um classificador binário, juntamente com o valor da área abaixo da curva, ou *area under the curve* (AUC). Classificadores de alto desempenho apresentam valores AUC e de Recall próximos (ou iguais) a 1. A Figura 10 no Apêndice C apresenta um modelo para interpretação dos resultados da curva ROC, para comparação da performance entre diferentes modelos.

3.7. Matriz de Confusão

Uma matriz de confusão é uma tabela com N linhas e N colunas, onde N é o número de classes atribuídos ao modelo, consequentemente uma matriz de confusão será sempre quadrada. A tabela exibe o número de falsos positivos, falsos negativos, verdadeiros positivos e verdadeiros negativos, permitindo uma análise detalhada do comportamento do modelo para todas as classes e tipos de erros. Na matriz as linhas representam as classes reais, enquanto as colunas representam as classificações realizadas pelo modelo. Portanto, os elementos da diagonal representam todas as classificações corretas, enquanto elementos fora da diagonal erros. Essa métrica pode ser empregada para avaliação de performance de modelos classificadores binários, assim como para problemas envolvendo diversas classes. A Tabela 3 apresenta um modelo de matriz de confusão para um problema de classificação binário, nesse exemplo o valor positivo está atribuído a classe 2.

Tabela 3 – Esquema de Matriz de Confusão para um modelo classificador binário.

	Classe 1	Classe 2
Classe 1	Verdadeiros Negativos	Falsos Positivos
Classe 1	Falsos Negativos	Verdadeiros Positivos

4. METODOLOGIA

Com o intuito de avaliar a performance dos algoritmos classificadores na detecção de diferentes falhas em máquinas rotativas, a classificação realizada com os sinais provenientes dos acelerômetros será comparada com os resultados obtidos utilizando apenas o sinal de áudio

para cálculo dos MFCCs utilizando o mesmo método, atributos e segregação treino/teste com estado aleatório (*random state*) fixo. O procedimento se resume em tratar os dados provenientes do sinal de vibração e áudio separadamente e calcular os MFCCs para um intervalo de tempo de 5 segundos. Em seguida, segregar os dados em duas categorias, treino (70% dos dados) e teste (30% dos dados), randomicamente (com estado randômico fixo para reprodutibilidade dos resultados), treinar cada um dos algoritmos com os dados de treino e por fim, avaliar a performance de cada algoritmo com base nas métricas de acurácia média, curva ROC e matriz de confusão.

4.1. Banco de dados

Os dados de vibração foram disponibilizados pelo Laboratório de Sinais, Multimídia e Telecomunicações (SMT) da Universidade Federal do Rio de Janeiro (UFRJ) na *Machinery Fault Database* (MAFAULDA, 2014). Este banco de dados é composto por 1951 arquivos contendo sinais de vibração multivariadas no domínio tempo, adquiridos por sensores em um Simulador de Falha de Maquinário (MFS) da empresa *SpectraQuest*, Alinhamento-Equilíbrio-Vibração (ABVT), exibida na Figura 11 no Apêndice D. Este equipamento permite a simulação dos principais tipos de falhas presentes em equipamentos rotativos. A Tabela 4 evidencia as principais características do motor utilizado no simulador, bem como as características dos rolamentos acoplados ao eixo dele.

Tabela 4 - Especificações do Motor e Rolamentos utilizados (Adaptado de MAFUALDA, 2014).

Especificação	Valor	Unidade
Potência do Motor	1/4	CV
Rotação	700-3600	rpm
Peso	22	kg
Diâmetro do Eixo	16	mm
Comprimento do Eixo	520	mm
Rotor	15.24	cm
Distância dos Rolamentos	390	mm
Número de Esferas	8	-
Diâmetro das Esferas	0.7145	cm
Diâmetro da Gaiola	28519	cm
<i>Fundamental Train Frequency</i> (FTF)	0.3750	CPM/rpm
Frequência de passagem da esfera externa (<i>Ball passage frequency Outer</i> , BPFO)	29980	CPM/rpm
Frequência de passagem da esfera interna (<i>Ball passage frequency Inner</i> , BPFI)	50020	CPM/rpm

A simulação compreende seis diferentes estados simulados: (1) função normal, (2) falha de Desbalanceamento, (3) falhas de desalinhamento horizontal e (4) vertical e (5) falhas de rolamento interno e (6) externo. Cada arquivo foi gerado a uma taxa de amostragem de 50 kHz durante 5 s, totalizando 250.000 amostras.

Para o maquinário em funcionamento normal foram realizadas 49 medições sem qualquer falha, cada uma com uma velocidade de rotação fixa na faixa de 737 rpm a 3686 rpm com passos de aproximadamente 60 rpm. Sinais para o conjunto desbalanceado, foram utilizados valores de carga na faixa de 6 a 35 gramas. Para cada valor de carga abaixo de 30 g, a frequência de rotação assumiu os mesmos 49 valores empregados no caso de operação normal.

Desalinhamento horizontal foi induzido no simulador deslocando o eixo do motor horizontalmente de 0,5 mm, 1,0 mm, 1,5 mm e 2,0 mm. Desalinhamento vertical foi forçado no ensaio deslocando o eixo do motor horizontalmente de 0,51 mm, 0,63 mm, 1,27 mm, 1,40

mm, 17,8 mm e 1,90 mm. Usando a mesma faixa de frequência de rotação como na operação normal para cada deslocamento vertical e horizontal.

Três rolamentos defeituosos foram utilizados no ensaio, cada um com um elemento defeituoso distinto (anel externo, esferas e gaiola), colocados um de cada vez em duas posições diferentes na bancada experimental: entre o rotor e o motor (posição de balanço), ou na posição externa, tendo o rotor entre o mancal e o motor (posição de balanço). As falhas nos rolamentos são praticamente imperceptíveis quando não há desequilíbrio. Assim, três massas de 6 g, 20 g e 35 g foram adicionadas para induzir um efeito detectável, com diferentes frequências de rotação como antes.

Tabela 5 - Condições da totalidade dos testes realizados (Adaptado de MAFUALDA, 2014)

Classe	Medições	Duração (s)
Normal	49	245
Desalinhamento Horizontal	197	985
Desalinhamento Vertical	301	1505
Desbalanceamento	333	1665
Falha na Gaiola	376	1880
Falha no Anel Externo	372	1860
Falha na Esfera	323	1615
Total	1951	9755

Para a aquisição de dados foram utilizados três sensores industriais IMI, acelerômetros modelos 601A01 nas direções radial, axial e tangencial com sensibilidade ($\pm 20\%$) 100 mV por g (10,2 mV por m/s^2), alcance de frequência (± 3 dB) 16-600000 CPM (0,27-10.000 Hz) e faixa de medição de ± 50 g (± 490 m/s^2). Um acelerômetro triaxial da IMI Sensors, Modelo 604B31, retornando dados nas direções radial, axial e tangencial de sensibilidade ($\pm 20\%$) 100 mV por g (10,2 mV por m/s^2), alcance de frequência (± 3 dB) 30-300000 CPM (0,5-5.000 Hz) e faixa de medição ± 50 g (± 490 m/s^2). Dois módulos de aquisição analógica de 4 canais NI 9234 da *National Instruments*, com taxa de amostragem de 51,2 kHz. Para aquisição dos dados acústicos foi utilizado o microfone *Shure SM81* com faixa de frequência de 20-20.000 Hz. Portanto, cada medição corresponde a 8 medidas individuais diferentes: Medida 1: sinal do tacômetro. Medidas 2 a 4: acelerômetro do rolamento inferior (direções axial, tangencial e radial direções). Medidas 5 a 7: acelerômetro de rolamento de balanço (direções axial, tangencial e radial ao eixo de rotação do motor) e a medida 8: Sinal de áudio do microfone.

4.2. Processamento de dados

Uma única medição representa 5 segundos para cada classe de falha, representando 250 mil pontos no tempo. Portanto, os 1951 arquivos com suas 487,75 milhões de linhas e 8 colunas, foram importados iterativamente para cálculo dos MFCCs. Para economia de memória RAM, os dados do sinal de vibração no domínio tempo são então descartados, apenas os MFCCs são armazenados em uma variável do tipo dicionário. A função *feature.mfcc* da biblioteca *Librosa* para Python foi utilizada para obtenção dos MFCCs, com número de coeficientes igual a 40, resolução da transformada rápida de Fourier igual a 2048, e cada conjunto de dados foi segmentado em 5 partes iguais, resultando em um subconjunto de 1 segundo por *frame* sobrepostos por 512 medições ou 0,01024 segundos.

O conjunto de MFCCs resultante é uma matriz (*numpy array* em Python) bidimensional de 40 linhas por 98 colunas para cada conjunto de dados de 5 segundos de uma única medição. A aplicação dos algoritmos SVM e Florestas Aleatórias exigem um conjunto de atributos unidimensional. Visando contornar essa limitação, a matriz resultante dos MFCCs passa por

uma justaposição de todas suas linhas, transformando a matriz original de 40 x 98 em um vetor unidimensional com 3920 elementos para cada medição utilizando o método *numpy.flatten* da biblioteca *Numpy* para Python. Por fim, para uma única medição de vibração, o resultado é uma matriz de 1.951 linhas e 3921 colunas, das quais uma contém a descrição da falha ou estado normal sem falhas e as outras 3920 os respectivos MFCCs justapostos para um único eixo (axial, tangencial ou radial) ou para o sinal de áudio.

Com o intuito de utilizar todos os eixos medidos e englobar ambos os acelerômetros (superior e inferior) e não correr o risco de perda de informações dos sinais originais, o procedimento de justaposição das linhas é repetido. Contudo, nessa etapa a justaposição é aplicada nos coeficientes cepstrais calculados para cada eixo (axial, tangencial e radial) para cada acelerômetro. O resultado é um vetor unidimensional com 23.520 colunas (atributos), representando todos os MFCCs calculados para os sinais de todos os eixos. O procedimento não é repetido para o sinal de áudio, uma vez que se trata de apenas uma medição.

Por fim, os dados são segregados aleatoriamente em dois subconjuntos: teste e treino. É importante destacar que o estado aleatório é fixo, permitindo reprodutibilidade dos testes e validações e comparação dos resultados para os diferentes algoritmos e diferentes iterações para ajuste de parâmetros para cada modelo. A Tabela 6 exhibe a política de divisão dos dados, informando que os dados de treino representam 70% do conjunto original, isto é, 1.366 amostras contendo 32.128.320 coeficientes calculados a partir dos sinais de vibração e 5.354.720 para o sinal de áudio. Os dados de teste contêm 585 amostras diferentes, 30% do conjunto original, com 13.759.200 MFCCs para sinais de vibração e 2.293.200 para a onda sonora.

Tabela 6 – Segregação do conjunto original em subconjuntos de treino e teste para validação dos algoritmos.

Método de Aquisição de dados	Subconjunto	Quantidade Relativa	Amostras	Quantidade MFCCs
Aceleração	Treino	70%	1.366	32.128.320
Aceleração	Teste	30%	585	13.759.200
Áudio	Treino	70%	1.366	5.354.720
Áudio	Teste	30%	585	2.293.200

4.3. Seleção e Parametrização dos Modelos

A seleção dos algoritmos de máquinas de vetores de suporte e florestas aleatórias foi influenciada pela eficácia em aplicações com conjunto de dados com elevado número de dimensões, facilidade de parametrização e performance computacional dos modelos. Florestas aleatórias não são muito sensíveis aos parâmetros usados e, de maneira geral, é fácil determinar quais parâmetros usar (BREIMAN, 2001). Além disso, HORNING (2010) destaca a precisão elevada, facilidade de parametrização e acurácia do modelo de florestas aleatórias. Segundo SUPRIYA e VALERIY (2001) o Algoritmo SVM combina a eficiência do treinamento e a simplicidade dos algoritmos lineares com a precisão das melhores técnicas não lineares. Em muitas aplicações práticas, o modelo pode tolerar dados de alta dimensão e/ou incompletos.

4.3.1. Pesquisa em Grade para Otimização dos Parâmetros

O processo de testar diferentes combinações dos parâmetros mais influentes para cada modelo é conhecido pesquisa em grade, do inglês *Grid Search*. Para este fim, diferentes valores para o parâmetro de regularização (C), Kernel, grau da função de Kernel polinomial (d) e constante de parametrização (r) descritos no capítulo 3.1 são testados, visando a obtenção da melhor performance possível. Para esse procedimento a métrica de comparação utilizada será

a acurácia média para cada iteração. Na Tabela 7 estão listados os valores de cada parâmetro, todas as combinações possíveis foram testadas, entretanto, variações distintas do grau (d) e constante (r) foram aplicados apenas para kernel polinomial, sendo mantidos fixos para as outras funções de kernel com os valores padrão, respectivamente 3 e 0.

Tabela 7 – Valores para cada parâmetro testados para o modelo SVM

Parâmetro	Valores Testados
Parâmetro de regularização (C)	[0.01, 1, 10, 100, 1000]
Kernel	[Polinomial, Linear, RBF]
Grau (d)	[3, 4, 5, 6]
Constante (r)	[2, 3, 6, 12, 18, 24]

Embora o algoritmo de Florestas Aleatórias apresente fácil parametrização e, quando comparado a modelos SVM, diferentes valores para os parâmetros acarretam pouca influência na acurácia do modelo, foram testadas diferentes combinações dos parâmetros no processo de pesquisa em grade: número de árvores na floresta e máxima profundidade, com o objetivo de avaliar a performance do classificador para cada combinação possível.

Tabela 8 - Valores para cada parâmetro testados para o modelo de Florestas Aleatórias

Parâmetro	Valores Testados
Número de Árvores na Floresta	[100, 1000, 10000]
Máxima profundidade	[10, 50, 100, Nenhum]

4.4. Resultados

4.4.1. Máquina de Vetores de Suporte

Os testes realizados com as diferentes configurações de parâmetros (Tabela 7) apresentaram os melhores resultados para função de Kernel polinomial, com grau (d) e constante (r) iguais a 3 e parâmetro de regularização (C) igual a 10. Resultando em uma classificação sem erros com acurácia de 100% para os MFCCs calculados a partir dos sinais de vibração provenientes dos acelerômetros. A Figura 1 (a) exibe a matriz de confusão para os resultados correspondentes ao sinal de vibração. A matriz de confusão não apresenta um único elemento fora da diagonal, o que é característico de um classificador perfeito. Igualmente, a Curva ROC com parâmetro AUC igual a 1, exibida na Figura 2 (a), é idêntica à curva de um classificador perfeito. O treino do modelo teve duração de 51 segundos para o conjunto dos MFCCs calculados a partir dos dados dos sinais vibracionais.

Para os coeficientes calculados a partir do sinal de áudio, a acurácia média foi de apenas 69,6 %. Os resultados do Escore F1 para cada classe estão exibidos na Tabela 9, com pontuações insatisfatórias para todas as classes de anomalias, exceto para a classe desalinhamento vertical, com 91,4%. O menor escore F1, pertencente a classe normal (24%), é decorrente do baixo número de amostras para esta classe, com apenas 49 medições realizadas. A matriz de confusão para o conjunto de MFCCs calculados a partir do sinal de áudio é exibida na Figura 1 (b), com um número elevado de erros de predição, representados por todos os elementos fora da diagonal. É importante destacar novamente a baixa acurácia do modelo para classe normal, das 18 classificações realizadas, apenas 3 estavam corretas. O treino do modelo teve duração de 33 segundos, aproximadamente 37% mais rápido do que modelo anterior. A maior velocidade na etapa de treino é influenciada pelo menor número de atributos para o modelo construído com o

signal do microfone, com 3920 coeficientes contra 23520 coeficientes do modelo construído a partir do sinal de vibração.

Tabela 9 – Escore F1 para o modelo SVM utilizando MFCCs calculados a partir dos Sinais de Vibração (Acelerômetros) e sinal de áudio (Microfone).

Classe	Escore F1 (Sinal de Vibração)	Escore F1 (Sinal de Áudio)
Desalinhamento Vertical	100,0%	91,4%
Normal	100,0%	24,0%
Falha no Anel Externo	100,0%	64,6%
Falha na Gaiola	100,0%	57,9%
Desbalanceamento	100,0%	73,3%
Falha na Esfera	100,0%	66,7%
Desalinhamento Horizontal	100,0%	71,4%

Figura 1 - Matriz de confusão do modelo SVM com MFCCs calculados a partir dos sinais de vibração dos acelerômetros (a) e sinal de áudio (b).

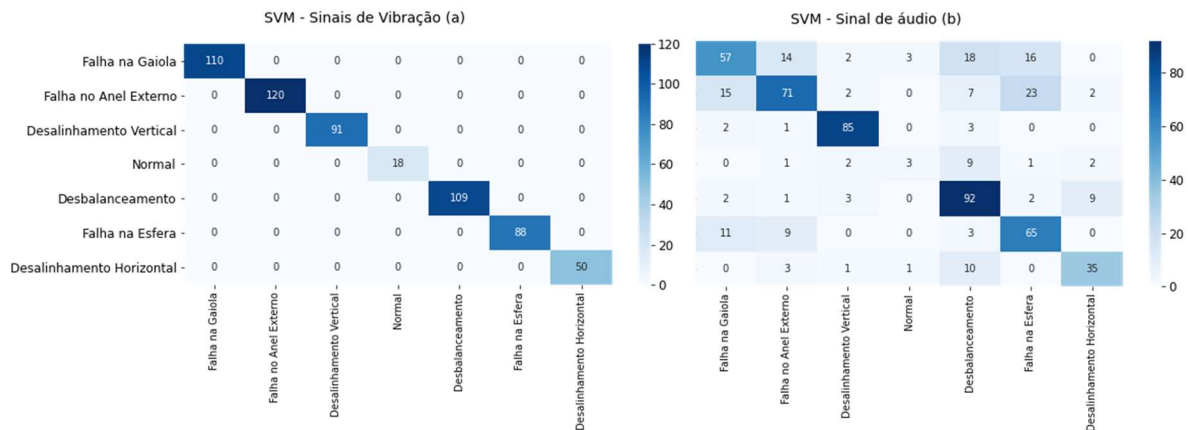
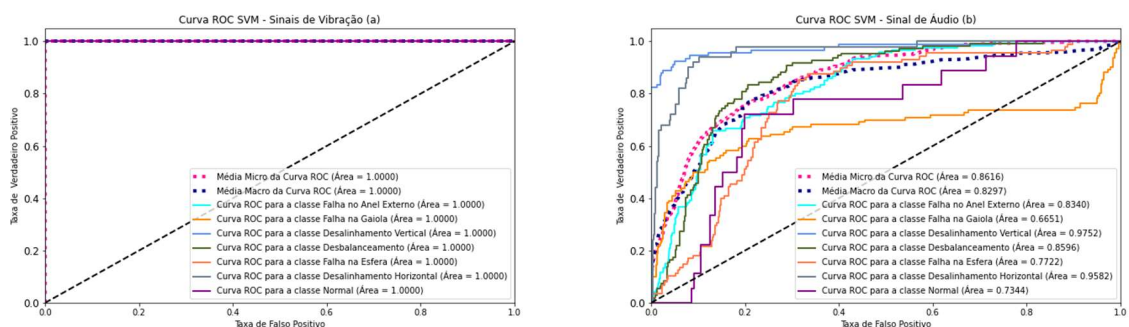


Figura 2 - Curva ROC para o modelo SVM com MFCCs calculados a partir dos sinais de vibração dos acelerômetros (a) e sinal de áudio (b).



4.4.2. Florestas Aleatórias

Com o número de árvores na floresta igual a 1000 e profundidade máxima igual a 10, foram obtidos os melhores resultados ao aplicar o algoritmo de florestas aleatórias. A duração total da etapa de treino foi 43 segundos para o sinal de áudio e 1 minuto e 17 segundos para coeficientes calculados a partir dos sinais de vibração, ambos valores superiores aos obtidos na etapa de treino do modelo SVM. A acurácia média para os MFCCs calculados a partir dos sinais de

vibração foi de 99,15%, evidenciando que, mesmo com a melhor performance do algoritmo SVM, o algoritmo florestas aleatórias é um excelente classificador para o problema proposto. A matriz de confusão na Figura 3 (a) mostra todas as classificações realizadas pelo modelo, com muitos elementos na diagonal, mostrando a alta performance do modelo. É perceptível na matriz 4 erros de classificação para a classe normal, com 3 classificações errôneas desbalanceamento e 1 falha na gaiola. Além disso, ocorreu um erro na classificação da anomalia desalinhamento horizontal, sendo classificada como falha na gaiola erroneamente. Todos os erros são representados pelos elementos fora da diagonal. De mesmo modo, a curva ROC do modelo se aproxima da curva de um classificador perfeito, sendo comprometida apenas pelos erros cometidos na classe normal, com AUC de 0,9996.

A acurácia média para os MFCCs calculados a partir dos sinais de áudio foi de 63,82 %. Os resultados do Escore F1 para cada classe estão exibidos na Tabela 10, com a classe normal sendo novamente responsável pelo menor escore (0%). Similarmente, A matriz de confusão com muitos elementos fora da diagonal e a Curva ROC com valores dos parâmetros AUC para as 6 classes evidenciam inferiores aos modelos construídos com os MFCCs calculados com os sinais de vibração evidenciam a baixa performance do modelo do sinal de áudio.

Tabela 10 - Escore F1 para o modelo de Florestas Aleatórias utilizando MFCCs calculados a partir dos Sinais de Vibração (Acelerômetros) e sinal de áudio (Microfone).

Classe	Escore F1 (Sinal de Vibração)	Escore F1 (Sinal de Áudio)
Desalinhamento Vertical	100,0%	88,3%
Falha no Anel Externo	100,0%	52,4%
Normal	87,5%	0,0%
Falha na Gaiola	99,1%	64,6%
Falha na Esfera	100,0%	54,8%
Desalinhamento Horizontal	99,0%	42,1%
Desbalanceamento	98,6%	75,8%

Figura 3 - Matriz de confusão do modelo Florestas Aleatórias com MFCCs calculados a partir dos sinais de vibração dos acelerômetros (a) e sinal de áudio (b).

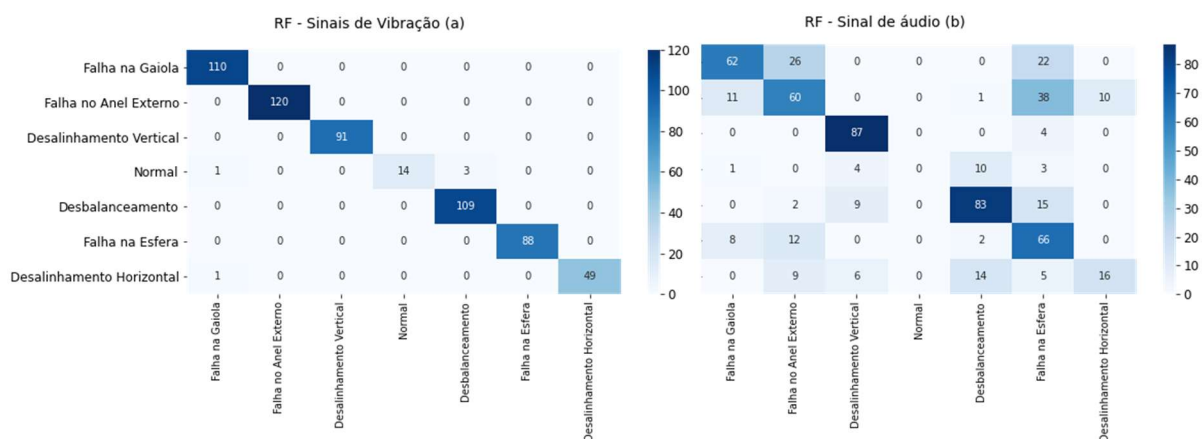
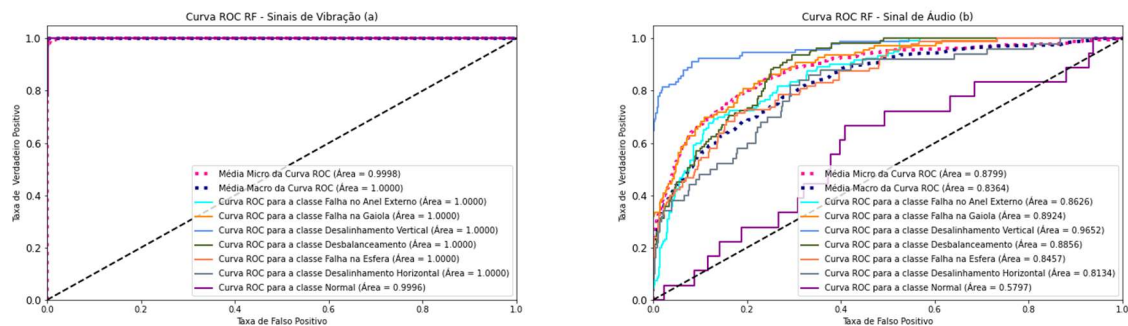


Figura 4 - Curva ROC para o modelo de Florestas Aleatórias com MFCCs calculados a partir dos sinais de vibração dos acelerômetros (a) e sinal de áudio (b).



A performance dos algoritmos para o conjunto de MFCCs gerados a partir dos sinais de vibração e sinal de áudio diferem substancialmente. Enquanto os dados de vibração e seus respectivos coeficientes cepstrais proporcionam a criação de modelos de aprendizado de máquina muito próximos de classificadores perfeitos, os algoritmos modelos com os coeficientes provenientes dos sinais de áudio apresentam resultados insatisfatórios, com comportamento similar a classificadores aleatórios. Outrossim, todas as métricas apresentadas neste trabalho apontam para o mesmo resultado, com o algoritmo SVM sendo o melhor classificador em termos de performance de classificação e tempo de processamento na etapa de treino do modelo.

5. CONCLUSÕES

O algoritmo máquina de vetores de suporte emerge como a melhor opção para a classificação de problemas e anomalias em máquinas rotativas utilizando como dados de treino os coeficientes cepstrais das frequências de Mel calculados a partir dos sinais de vibração do motor provenientes de dois acelerômetros. O Algoritmo apresentou acurácia média e escore F1 de 100% para todas as classes para os MFCCs obtidos via sinais de vibração e acurácia média de 69,6% para coeficientes calculados a partir do sinal sonoro. Além disso, o modelo SVM se mostrou mais rápido na fase de treinamento quando comparado ao modelo de Florestas Aleatórias, com uma duração de 51 segundos contra 1 minuto e 17 segundos do algoritmo concorrente. Esse resultado se deve ao elevado número de atributos, MFCCs, e a escolha correta dos parâmetros para obtenção dos melhores resultados.

Os coeficientes cepstrais das frequências de mel foram excelentes atributos para aplicações de detecção de falhas em máquinas rotativas por sinais de vibração utilizando os algoritmos de aprendizado de máquina selecionados. Contudo, utilizando os mesmos métodos e parâmetros, com os coeficientes calculados a partir do sinal sonoro do motor (aplicação muito similar para a qual os MFCCs foram inicialmente desenvolvidos) os resultados não são satisfatórios, com melhor acurácia média de 69,6% proveniente do modelo de Máquina de Vetores de Suporte.

Embora os resultados obtidos via sinal sonoro não tenham sido adequados, a aplicação de modelos desse tipo em situações reais na indústria seria menos complexa e de menor custo, quando comparada a aplicação utilizando os acelerômetros, devido ao custo desses equipamentos e complexidade da configuração deles em uma máquina. Isto posto, algumas propostas para futuros trabalhos podem ser levantadas: uma análise mais detalhada e direcionada para a aplicação dos mesmos modelos utilizando o sinal sonoro para obtenção dos MFCCS ou a adoção de outros algoritmos mais adequados para esses atributos.

REFERÊNCIAS BIBLIOGRÁFICAS

AHMADs S.; STYP-REKOWSKI, K.; NEDELKOSKI, S; O. KAO, Autoencoder-based Condition Monitoring and Anomaly Detection Method for Rotating Machines. **2020 IEEE International Conference on Big Data (Big Data)**, 2020, p. 4093-4102.

ANTONI, L.; ZIEMOWIT, D.; PIOTR, C. An anomaly detection method for rotating machinery monitoring based on the most representative data. **Journal of Vibroengineering** Vol. 23, Issue 4, 2021, p. 861-876.

ARALTO, A. **Manutenção Preditiva: Usando Análise de Vibrações**. São Paulo: Manole, 2004.

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

HO, TIN KAM (1995). Random Decision Forests. **Proceedings of the 3rd International Conference on Document Analysis and Recognition**. Montreal, QC, 14–16 August 1995. pp. 278–282.

K.S. RAO AND MANJUNATH K.E., Speech Recognition Using Articulatory and Excitation Source Features. **Springer Briefs in Speech Technology**, 2017.

LEE, S.; YU, H.; YANG, H.; SONG, I.; CHOI, J.; YANG, J.; LIM, G.; KIM, K.-S.; CHOI, B. Kwon, J. A Study on Deep Learning Application of Vibration Data and Visualization of Defects for Predictive Maintenance of Gravity Acceleration Equipment. **Appl. Sci.** vol. 11, p. 1564, 2021

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

MAFAULDA. COPPE/Poli/UFRJ. (2014). **Machinery Fault Database - MAFAULDA**. Disponível em: <http://www02.smt.ufrj.br/~offshore/mfs/page_01.html> Acesso em: 21 de setembro de 2021.

MCFEE, B. et al. *librosa/librosa*: 0.8.0. 22 Julho, 2020.

PEDERSEN, R.; SCHOEBERL, M. (2006). **An Embedded Support Vector Machine. In Proceedings of the Fourth Workshop on Intelligent Solutions in Embedded Systems**. WISES, 2006, pp. 79-89.

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

PITTINO, F., PUGGL M., MOLDASCHL, T., HIRSCHL, C. Automatic Anomaly Detection on In-Production Manufacturing Machines Using Statistical Learning Methods. **Sensors**. v. 20, p. 8: 2344, 2020.

PRATI, R. C.; BATISTA, G. E. A. P. A; MONARD, M. C., Evaluating Classifiers Using ROC Curves. **IEEE Latin America Transactions**, vol. 6, no. 2, pp. 215-222, Junho, 2008.

PYTHON. **Support Vector Machine Python Example. Towards data science**. Disponível em: <<https://towardsdatascience.com/support-vector-machine-python-example-d67d9b63f1c8>> Acesso em: 15 de julho de 2021.

SCIKIT. **Support Vector Machines – Scikit-Learn Documentation**. Disponível em: <<https://scikit-learn.org/stable/modules/svm.html#mathematical-formulation>> Acesso em: 11 de outubro de 2021.

SIRIKULVIRIYA, NAPHAPORN AND SUKREE SINTHUPINYO. Integration of Rules from a Random Forest. **International Conference on Information and Electronics Engineering IPCSIT** vol.6, 2011.

SpectraQuest. **SpectraQuest, Inc.** Disponível em: <<https://spectraquest.com/>> Acesso em: 21 de setembro de 2021

VAPNIK, V. N. **The Nature of Statistical Learning Theory**. Springer, NY, 1995.

ZHANG, X.; JIANG, D.; HAN, T.; WANG, N.; YANG, W.; YANG, Y. Rotating Machinery Fault Diagnosis for Imbalanced Data Based on Fast Clustering Algorithm and Support Vector Machine. **Journal of Sensors**, Beijing, China, v. 2017, p.0-15, 2017.

HORNING, N. **Random Forests: An algorithm for image classification and generation of continuous fields data sets**, 2010.

GAVRISHCHAKA, V.; GANGULI, S. Support vector machine as an efficient tool for high-dimensional data processing: Application to substorm forecasting. **Journal of Geophysical Research**. 106. 29911-29914, 2001.

SCIKIT. **Metrics and scoring: quantifying the quality of predictions**. Disponível em: <https://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics> Acesso em: 07 de novembro de 2021.

APÊNDICE A

Figura 5 - Hiperplano e margens geradas pelo algoritmo SVM para um problema de classificação binário

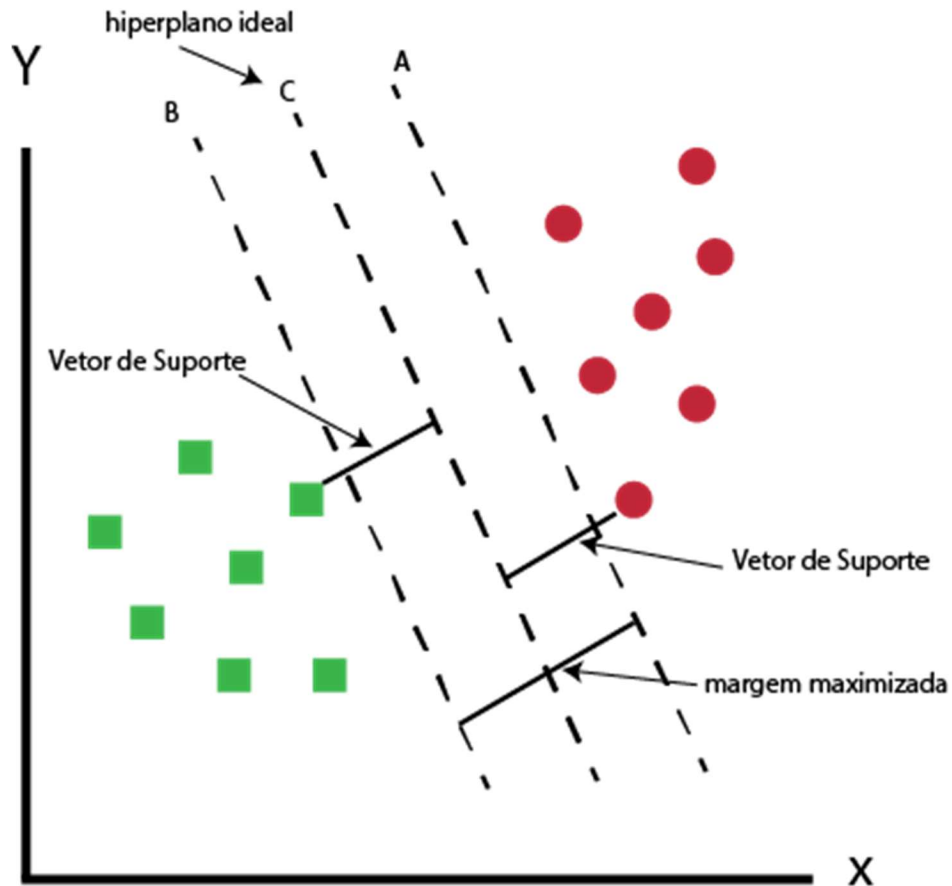


Figura 6 – Mapeamento do espaço dos atributos para o espaço das características de alta dimensão

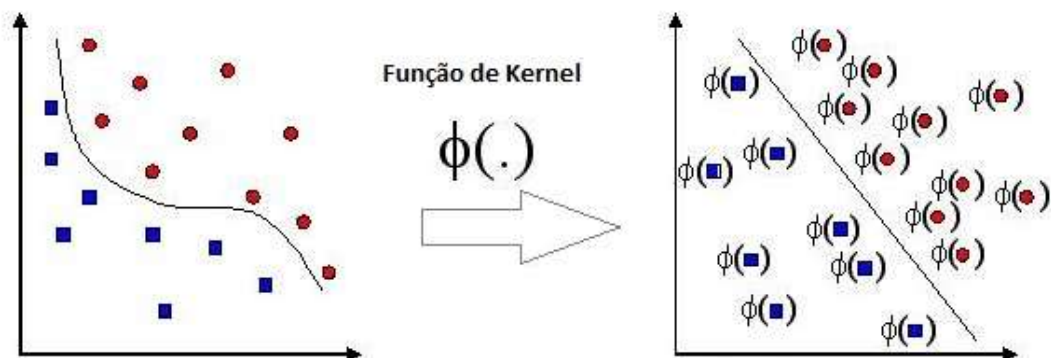
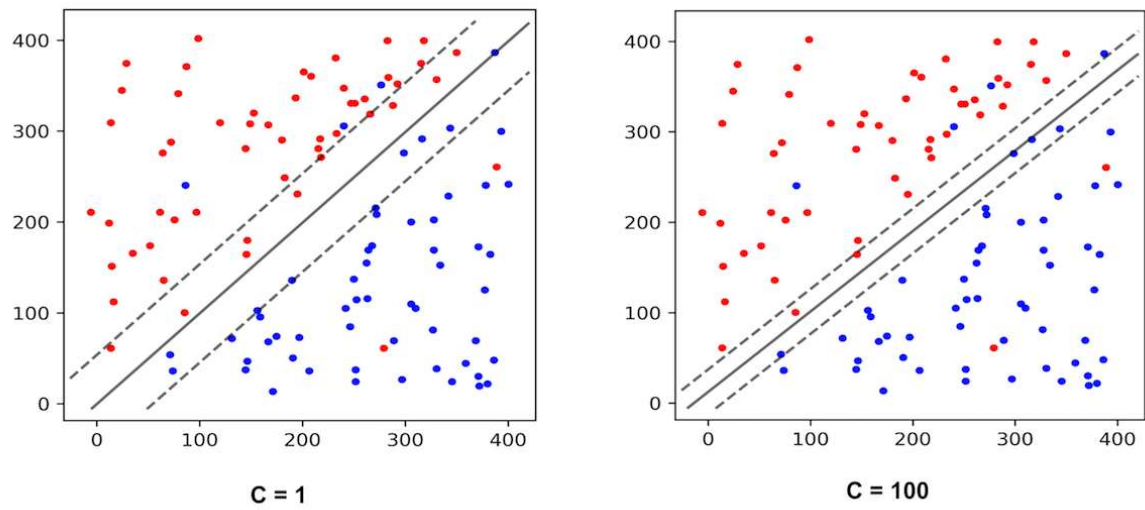
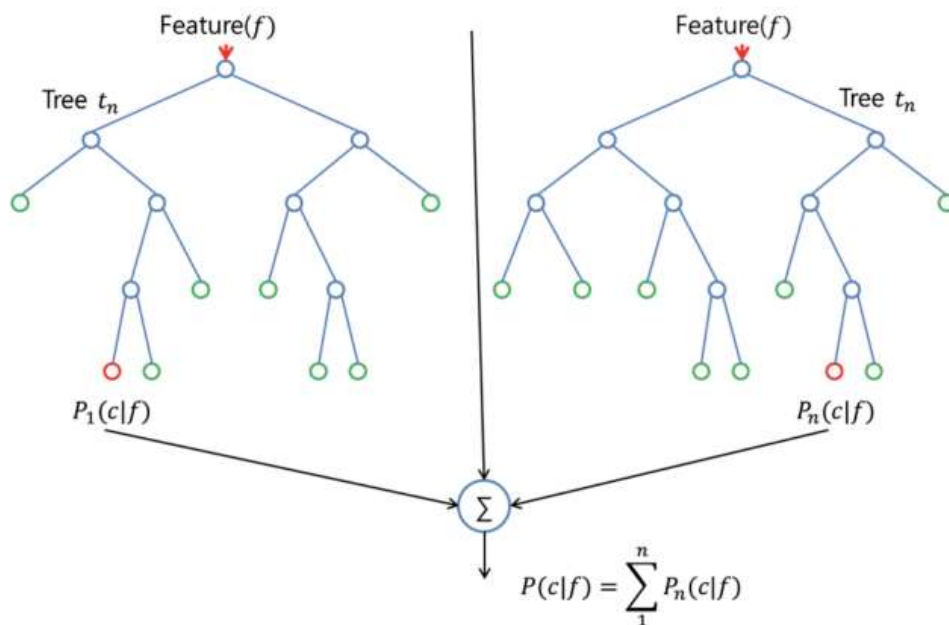


Figura 7 – Margens e hiperplanos obtidos via modelo SVM em um problema de classificação binário para valores do parâmetro de regularização (C) iguais a 10 e 100



APÊNDICE B

Figura 8 - Esquema do algoritmo de florestas aleatórias



APÊNDICE C

Figura 9 - Exemplo de Curva ROC e parâmetro AUC

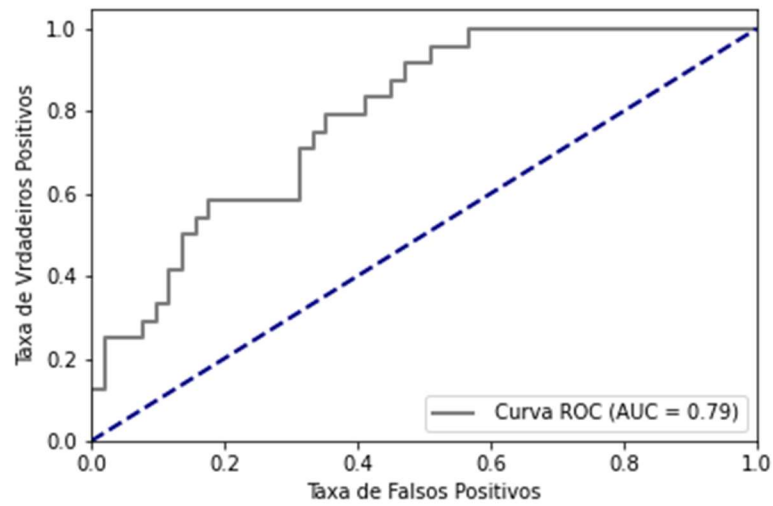
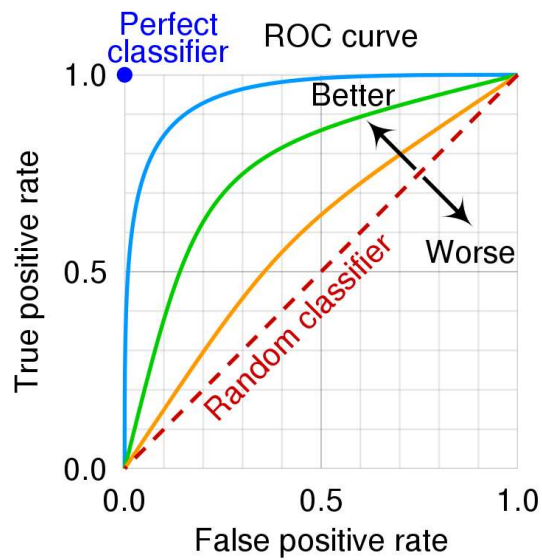


Figura 10 - Modelo para interpretação dos resultados da curva ROC, para comparação da performance entre diferentes modelos



APÊNDICE D

Figura 11 - Simulador de Falha de Maquinário (MFS) da empresa *SpectraQuest*, Alinhamento-Equilíbrio-Vibração ABVT (Fonte: *SpectraQuest, Inc.*).

