UNIVERSIDADE FEDERAL DO PARANÁ

DANIEL AWADA ELARRAT CANTO

IDENTIFICAÇÃO DE DEFEITOS EM MÁQUINAS ROTATIVAS UTILIZANDO MODELO COMPOSTO DE LSTM E FLORESTA ALEATÓRIA

CURITIBA

2024

DANIEL AWADA ELARRAT CANTO

IDENTIFICAÇÃO DE DEFEITOS EM MÁQUINAS ROTATIVAS UTILIZANDO MODELO COMPOSTO DE LSTM E FLORESTA ALEATÓRIA

Trabalho apresentado como requisito parcial para a obtenção do título de Mestre em Engenharia Mecânica pelo Programa de Pós Graduação em Engenharia Mecânica do Setor de Tecnologia da Universidade Federal do Paraná.

Orientadora: Profª Giuliana Sardi Venter

CURITIBA

2024

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Canto, Daniel Awada Elarrat
Identificação de defeitos em máquinas rotativas utilizando modelo composto de LSTM e floresta aleatória / Daniel Awada Elarrat Canto. – Curitiba, 2024.
1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Mecânica.

Orientador: Giuliana Sardi Venter

1. Máquinas - Manutenção e reparos. 2. Aprendizado de máquinas. 3. Floresta aleatória. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Engenharia Mecânica. III. Venter, Giuliana Sardi. IV. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA MECÂNICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **DANIEL AWADA ELARRAT CANTO** intitulada: **IDENTIFICAÇÃO DE DEFEITOS EM MÁQUINAS ROTATIVAS UTILIZANDO MODELO COMPOSTO DE LSTM E FLORESTA ALEATÓRIA**, sob orientação da Profa. Dra. GIULIANA SARDI VENTER, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 28 de Junho de 2024.

Documento assinado digitalmente
gov.br
GIULIANA SARDI VENTER
Data: 28/06/2024 16:47:50-0300
Verifique em https://validar.iti.gov.br

GIULIANA SARDI VENTER
Presidente da Banca Examinadora

Documento assinado digitalmente
gov.br
RODRIGO NICOLETTI
Data: 28/06/2024 16:15:55-0300
Verifique em https://validar.iti.gov.br

RODRIGO NICOLETTI
Avaliador Externo (UNIVERSIDADE DE SÃO PAULO - USP)

Documento assinado digitalmente
GOV.BR
EDUARDO LUIZ ORTIZ BATISTA
Data: 28/06/2024 16:35:18-0300
CPF: ***.521.889-**
Verifique as assinaturas em https://v.ufsc.br

EDUARDO LUIZ ORTIZ BATISTA
Avaliador Externo (UNIVERSIDADE FEDERAL DE SANTA CATARINA)

AGRADECIMENTOS

Aos meu pais, Jorge Elarrat e Samia Awada.

À minha orientadora, Giuliana Venter.

Aos coordenadores do Grupo de Vibrações e Som, Eduardo Lopes e Carlos Bavastri, pela ajuda imprescindível para a qualidade desse trabalho.

Aos meus colegas que me auxiliaram na criação desse projeto, entre eles, Leonardo Kaucz.

Aos gigantes em que me apoiei durante esse trabalho, como Maurizio Barghouthi, criador do Analisador de Sinais em LabView aqui utilizado.

RESUMO

O aprendizado de máquinas está revolucionando os processos industriais e vem incentivando a otimização em seus mais diversos setores, como o setor de fabricação com o processo de manutenção preditiva. Com o aprendizado de máquina, é possível prever defeitos e falhas em máquinas rotativas antes de elas causarem danos graves às máquinas e, assim, reduzir os custos de operação, aumentar a segurança, a eficiência e prolongar a vida útil dos equipamentos. Este trabalho aplicou técnicas de aprendizado de máquina para prever o comportamento futuro de máquinas rotativas e detectar defeitos de desbalanceamento e desalinhamento com antecedência utilizando modelos de *Long-Short Term Memory* (LSTM) e floresta aleatória. Os dados de treino foram coletados a partir de uma bateria de experimentos utilizando um aparato experimental composto por um acelerômetro triaxial, um módulo de aquisição de sinais, pesos para simulação de desbalanceamento e sistema de máquina rotativa. Um dos diferenciais desse projeto foi o método de *data augmentation* a partir da geração de dados sintéticos, o que possibilita a previsão do comportamento da máquina para estados intermediários de falhas. Esses dados foram gerados a partir da interpolação dos dados obtidos em laboratório, combinando linearmente no tempo 2 ou mais dados de diferentes estados de máquinas, para simular a transição entre 2 estados da máquina. Foram obtidos um total de 2750 experimentos diferentes (891 laboratoriais e 1.849 gerados sinteticamente), capazes de abranger os estados mais comuns de funcionamento de uma máquina rotativa, além de conter as informações de velocidade de rotação, estado de falha e posição do sensor. Foram utilizados janelamentos e extração de parâmetros para gerar os dados finais de treinamento da rede neural de LSTM utilizada por ser capaz de prever o comportamento futuro da máquina, retornando um valor numérico para cada série de parâmetros, que foram processados por duas redes, uma de classificação e outra de regressão, para a análise de presença de falhas. Para a escolha dos modelos finais de classificação e regressão, foram testados diversos métodos diferentes, são eles K-*nearest neighbors* (KNN), floresta aleatória, *support vector machine* (SVC) e *artificial neural network* (ANN). O modelo mais eficiente foi combinado com o modelo LSTM para formar o modelo final. Ele foi capaz de identificar corretamente o estado da máquina no futuro em 86,92% das vezes, e com erros que podem ser considerados desprezíveis: 0,202° para o ângulo de desalinhamento e 0,207% do peso da máquina em desbalanceamento.

Palavras-chaves: Manutenção preditiva, Aprendizado de máquinas, LSTM, Otimização de processos, Máquinas rotativas, Previsão de falhas, Floresta aleatória

ABSTRACT

Machine learning is revolutionizing industrial processes and driving optimization in various sectors, such as the manufacturing sector with predictive maintenance processes. With machine learning, it is possible to predict faults and failures in rotating machinery before they cause significant damage, thus reducing operating costs, increasing safety, efficiency, and extending the equipment's lifespan. This work applied machine learning techniques to predict the future behavior of rotating machinery and detect imbalance and misalignment defects in advance using Long-Short Term Memory (LSTM) models and Random Forests. Training data was collected from a series of experiments using an experimental setup composed of a triaxial accelerometer and a rotating machine system. One of the differentiating aspects of this project was the generation of synthetic data, which enables the prediction of machine behavior for intermediate fault states. These data were generated by interpolating laboratory data, linearly combining in time two or more data points from different machine states to simulate the transition between them. A total of 2750 different experiments (891 laboratory-based and 1,849 synthetically generated) were obtained, covering the most common operating states of a rotating machine, and including information on rotation speed, failure state, and sensor position. Windowing and parameter extraction were used to generate the final training data for the LSTM neural network. The LSTM network was employed for its ability to predict the machine's future behavior, returning a numerical value for each series of parameters. These were processed by two networks, one for classification and another for regression, to analyze the presence of faults. For the selection of the final classification and regression models, various methods were tested, including K-nearest neighbors (KNN), random forest, support vector machine (SVC), and artificial neural network (ANN). The most efficient model was combined with the LSTM model to form the final model. It was able to correctly identify the machine's future state 86.92% of the time, with negligible errors: 0.202° for the misalignment angle and 0.207% of the machine's weight in imbalance.

Key-words: Predictive maintenance, Machine learning, LSTM, Process optimization.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

As máquinas rotativas são essenciais para a indústria moderna, sendo utilizadas em uma variedade de aplicações, desde tornos industriais até eletrodomésticos. Seu principal objetivo é converter energia elétrica em mecânica de forma otimizada, com o mínimo de desperdício (APICELLA et al., 2021). No entanto, para garantir seu funcionamento ideal, é necessário realizar manutenções regulares para evitar falhas, o que exige a parada da máquina, gerando mais custos no processo, especialmente quando se analisam máquinas industriais (GHORBANI et al., 2022).

Os defeitos mais comuns em máquinas rotativas são o desbalanceamento, desalinhamento, desgaste dos rolamentos e flambagem do eixo. O desbalanceamento ocorre quando a distribuição de massa da máquina não está uniforme. Já o desalinhamento pode ser causado por problemas na instalação ou ajuste inadequado da ferramenta. O desgaste dos rolamentos é resultado do tempo de uso excessivo da máquina e da falta de lubrificação adequada, enquanto a flambagem do eixo ocorre quando a carga na máquina ultrapassa o limite de suporte do eixo (BIAO et al., 2022). Cada uma dessas falhas gera vibrações indesejadas que comprometem a qualidade da produção, a vida da máquina e aumentam o risco de acidentes. Por isso é fundamental identificar e corrigir essas falhas o mais cedo possível a partir de uma rotina saudável de manutenções (BIAO et al., 2022).

Os três tipos de manutenção que existem atualmente são a manutenção preventiva, a corretiva e a preditiva. Define-se a manutenção preventiva como aquela feita de forma planejada e com os objetivos de evitar as falhas e garantir a continuidade do processo. Já a manutenção corretiva é aquela realizada após a ocorrência de uma falha e seu objetivo é corrigir o problema para restabelecer o funcionamento da máquina (SCHWENDEMANN et al., 2021). Por fim, a manutenção preditiva é um método que utiliza tecnologia de monitoramento e análise de vibrações em tempo real para prever possíveis falhas em máquinas e equipamentos, permitindo que uma manutenção seja realizada apenas quando necessária (PAUL et al., 2022).

A manutenção preditiva é o método considerado mais eficiente, porque é o que permite uma melhor gestão dos recursos de manutenção, reduzindo custos e aumentando a eficiência da produção. Além disso, ela possibilita a identificação de problemas antes que eles se tornem críticos, evitando a necessidade de manutenção corretiva e minimizando os riscos de acidentes e perda de qualidade dos produtos (NASRFARD et al., 2023).

Embora a manutenção preditiva seja a mais eficiente, a sua implementação

ainda pode apresentar algumas dificuldades para os setores de produção das empresas, devido ao investimento inicial em tecnologias avançadas de monitoramento e análise de dados, além de profissionais capacitados para operar essas ferramentas (KAWAI et al., 2020). No entanto, os benefícios da manutenção preditiva, como a redução de custos e a melhoria da eficiência da produção, podem compensar esses desafios e garantir um retorno positivo sobre o investimento a longo prazo.

Com a evolução do aprendizado de máquina, é possível desenvolver metodologias inteligentes o suficiente para serem usadas no diagnóstico de integridade da máquina, assim facilitando a implementação da manutenção preditiva nas empresas e automatizando a análise para tomada de decisão (NUNES et al., 2023). O uso de algoritmos de aprendizado de máquina permite identificar padrões de comportamento das máquinas e prever falhas com grande precisão e velocidade, reduzindo a necessidade de intervenção humana.

O trabalho apresentado nesta dissertação propõe-se a trazer uma nova abordagem ao problema da manutenção preditiva assistida por aprendizado de máquina, com uma nova metodologia desde a coleta de dados em experimentos laboratoriais com máquinas rotativas reais - utilizando geração de dados sintéticos com variação temporal - até a sua aplicação em um modelo composto, formado por uma rede de regressão, de classificação e recorrente.

Os modelos de classificação e regressão são técnicas de aprendizado de máquina que permitem prever a classe ou os valores de uma resposta de um sistema a partir de seus parâmetros, sem a necessidade de intervenção humana. Entre os modelos mais comuns estão a árvore de decisão, floresta aleatória, KNN, SVM e redes neurais. Cada um deles serão avaliados separadamente para que seja identificado o mais eficiente para o problema proposto.

As redes recorrentes são técnicas de rede neural que possuem a capacidade de utilizar as informações anteriores de uma série temporal para prever quais serão seus valores no futuro. Um dos modelos mais utilizados é a rede *Long Short-Term Memory* (LSTM), que é capaz de lidar com longas sequências de dados temporais e interpretar não só os dados em si, mas a relevância da sua posição na série.

Assim, será criado um método prático de previsão de falhas que combina esses diferentes modelos de aprendizado de máquina. Esse método será capaz de analisar o comportamento de máquinas e de realizar tomadas de decisão focadas na qualidade e na segurança do processo, gerando eficiência tanto para a máquina quanto para a empresa que o utilizar.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Obter um modelo composto de aprendizado de máquinas formado por redes LSTM, classificador e regressor, e aplicá-lo à identificação do tipo e intensidade de falhas em máquinas rotativas apenas a partir dos seus dados de vibração no tempo, em tempo real.

1.1.2 Objetivos específicos

- Desenvolver uma metodologia de experimentos prática e reproduzível para máquinas rotativas em diferentes tipos de falhas;
- Gerar a base de dados, incluindo dados sintéticos, possibilitando valores de resposta contínuos;
- Desenvolver uma metodologia de pré-processamento específico para sistemas físicos;
- Extrair os valores dos parâmetros e das respostas a partir dos dados e pré-processar a tabela resultante, incluindo normalização dos dados, remoção de *outliers* e seleção de parâmetros, visando aumentar a qualidade e abrangência dos dados de treino;
- Identificar o melhor modelo de classificação e de regressão para os dados do sistema;
- Treinar ambos os modelos junto com uma rede LSTM, otimizando seus respectivos parâmetros para obtenção dos resultados finais.

1.2 ESTRUTURA DA DISSERTAÇÃO

As próximas seções estão divididas em fundamentação teórica, revisão bibliográfica, metodologia, resultados e conclusão.

No segundo capítulo será apresentada a revisão bibliográfica, em que serão explicados os conceitos fundamentais necessários para a compreensão desse trabalho. Nele será apresentado conhecimento básico de máquinas rotativas, os seus defeitos e os métodos de aprendizado de máquinas aqui utilizados. No terceiro capítulo, serão apresentados trabalhos já realizados nessa área, uma breve descrição da metodologia utilizada e os resultados obtidos em cada um deles, com o objetivo de enfatizar as diferenças entre este projeto e anteriores. No quarto capítulo será apresentada a metodologia, que está dividida em duas grandes partes: detalhamento dos equipamentos

e detalhamento dos processos. Nele, estão descritas as características de hardware e software de cada material utilizado nos experimentos, assim como os métodos utilizados para trabalhar com os dados que foram gerados. No quinto capítulo, serão apresentados os resultados finais, seguidos, no sexto capítulo, pelas considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão explorados brevemente a teoria de análise de falha para cada estado de uma máquina rotativa, e os modelos de aprendizado de máquinas.

2.1 ESTADOS DA MÁQUINA ROTATIVA

Uma máquina rotativa ideal é aquela capaz de trabalhar no seu estado normal por toda a sua vida útil. Porém, em condições reais de ambiente e de trabalho, as máquinas reais apresentam alterações no seu estado de funcionamento, com vibrações indesejadas, o que compromete o seu funcionamento. Entre os estados mais comuns, além do estado normal, estão o desbalanceamento e o desalinhamento (BRAUN et al., 2002).

Outras falhas que podem ocorrer são referentes aos elementos que compõem a máquina, como rolamentos, engrenagens e outros componentes críticos. Elas são causadas pelo desgaste dos rolamentos, pelo aumento do atrito entre rotor e estator ou pelo aumento da folga entre rotor e estator, causando esforços desnecessários e dissipação da energia do sistema para outras partes (WANG et al., 2023).

É de grande importância saber os tipos de estado que uma máquina pode assumir para poder identificar as necessidades que ela possui, de forma que seja possível trabalhar de forma assertiva no diagnóstico e corrigir possíveis problemas, a fim garantir a operação eficiente dessas máquinas.

2.1.1 Estado Normal

A máquina está em seu estado normal quando não há a presença de defeitos ou os defeitos presentes são negligenciáveis. Esse comportamento demonstra que há uma boa manutenção na máquina e na execução do seu processo, em que nenhuma interferência é necessária e sua eficiência é máxima. Qualquer outro estado diferente do normal é indesejado, pois pode causar efeitos graves no seu desempenho e na sua vida útil, como vibrações excessivas, estresse e desgaste prematuro em rolamentos e acoplamentos, o que leva à falhas prematuras de componentes.

2.1.2 Estado Desalinhado

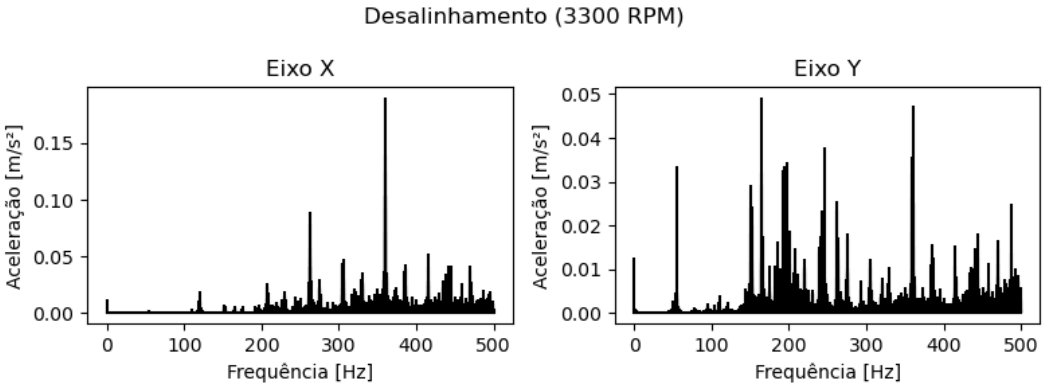
Uma das causas para uma máquina rotativa sair do seu estado normal é quando o seu eixo se desalinha em relação ao seu eixo de rotação. Isso pode ocorrer de duas formas:

1. Pelo desgaste do uso da máquina, como expansão térmica dos suportes, estresse vindo de tubulações, deformações nas estruturas e das mudanças na geometria do rotor, com a variação de temperatura;

2. Pelos problemas durante a instalação, a exemplo da manutenção inadequada, do desgaste de componentes e colisões da ferramenta.

No espectro da máquina, esse defeito ser identificado a partir dos picos de aceleração nas harmônicas do sistema. Na Figura 1 está a visualização de um espectro de uma máquina na frequência com o defeito de desalinhamento horizontal.

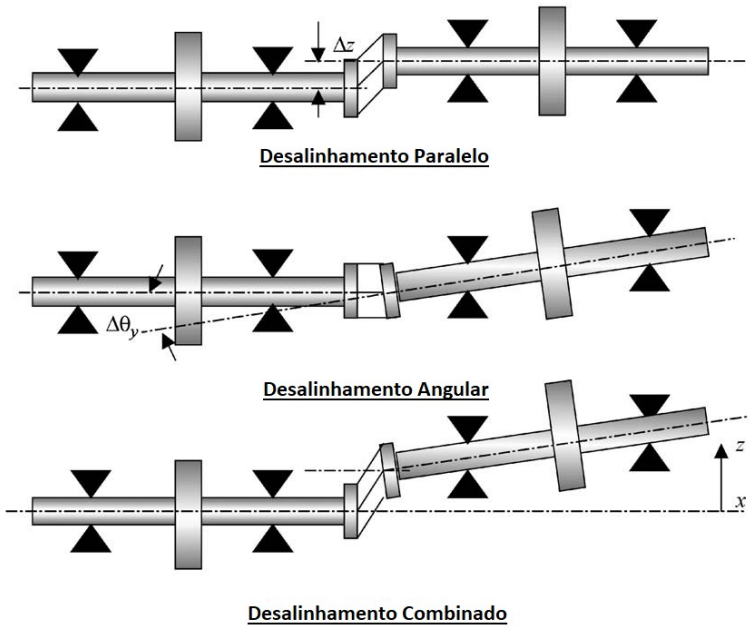
FIGURA 1 – Comportamento de uma máquina rotativa com desalinhamento na frequência.



Fonte: O AUTOR

Os desalinhamentos podem ocorrer de duas formas diferentes, paralelo ou angular. O desalinhamento paralelo é definido quando o eixo de rotação está em paralelo com o eixo da máquina, mas não estão concêntricos, gerando uma diferença de desvio. Já o desalinhamento angular, por sua vez, é quando esses eixos formam um ângulo entre si, deixando de ser paralelos. Durante o processo, a forma mais comum encontrada de desalinhamento é aquela que combina ambas as formas. Os tipos de desalinhamentos que podem ocorrer na máquina podem ser vistos na Figura 2.

FIGURA 2 – Tipos de desalinhamento que podem ocorrer durante o funcionamento da máquina.



Fonte: (SINHA et al., 2004) (adaptado)

As forças resultantes desse tipo de falha são proporcionais à magnitude e direção do desalinhamento. Logo, as forças resultantes em cada eixo são as seguintes:

$$\begin{cases} F_x = k\delta_x \\ F_y = k\delta_y \\ F_z = k\delta_z \end{cases} \tag{2.1}$$

sendo:

- $F_{x,y,z}$ a força resultante no eixo x, y, ou z, em Newtons (N);
- k a constante de rigidez do eixo, em Newtons por metro (N/m);
- $\delta_{x,y,z}$ o desalinhamento no eixo x, y, ou z, em metros (m).

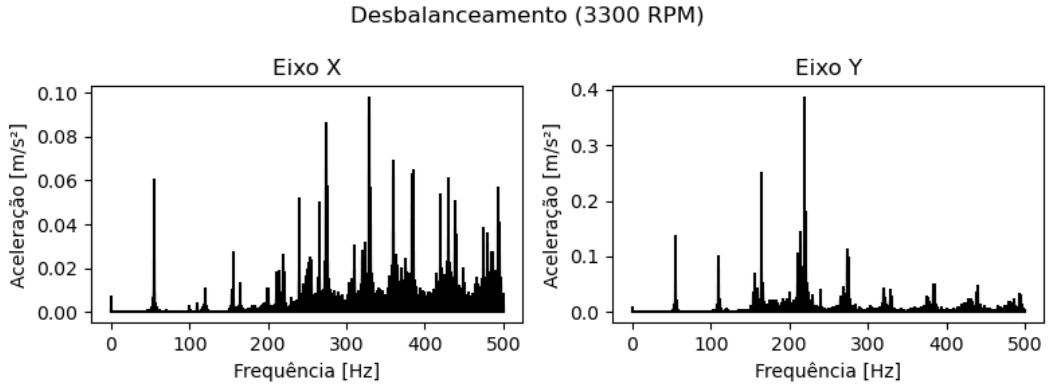
Esse tipo de defeito exige atenção imediata, pois a força gerada é prejudicial à máquina, gerando perda de energia durante o processo e podendo causar dano à mesma. As soluções podem ser a correção de peças de fixação soltas ou deterioradas, inspeção e manutenção do acoplamento e a execução de um alinhamento de precisão (BRAUN et al., 2002).

2.1.3 Estado Desbalanceado

Outro tipo de falha que pode tirar a máquina do seu estado normal é o desbalanceamento, que ocorre quando o centro de massa do sistema não está alinhado com o seu eixo de rotação, gerando uma força resultante que interfere na vibração da máquina (BIAO et al., 2022). O motivo desse desbalanceamento pode ser o desgaste da máquina, como acúmulos de sujidades, tinta, ferrugem ou outros materiais estranhos, ou a má configuração da máquina, como a má fixação ou completa ausência de fixadores (BRAUN et al., 2002).

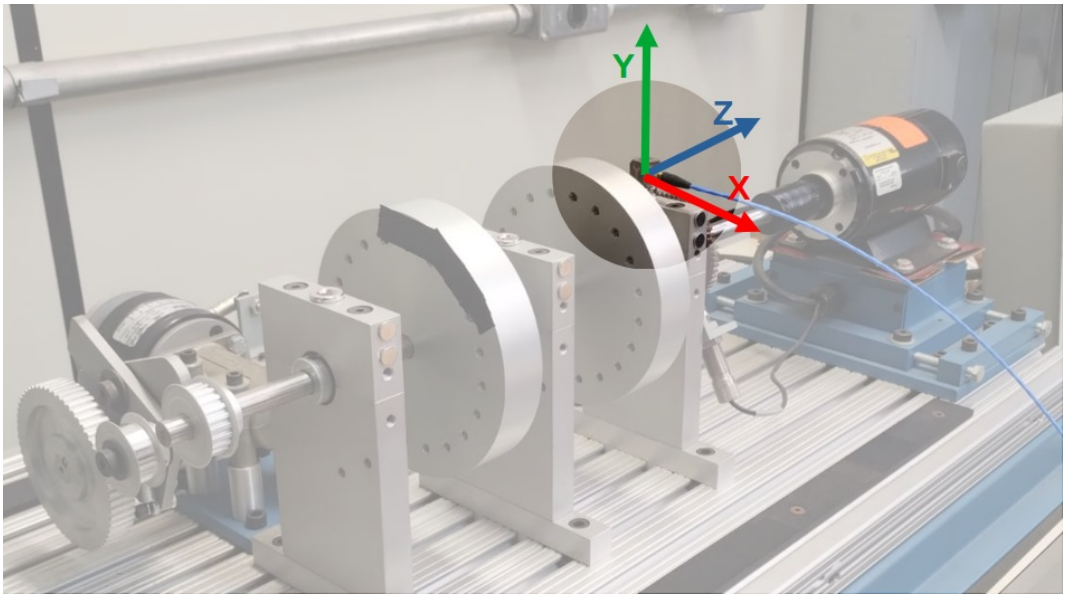
Como pode ser visto na Figura 3, essa falha causa um aumento da amplitude de vibrações em frequências indesejadas, diferente do comportamento normal onde a energia é distribuída de forma mais homogênea ao longo das frequências. Essa falha converte parte da energia da rotação para fins inadequados, como ruídos e estresse entre os componentes da máquina.

FIGURA 3 – Comportamento na frequência de uma máquina rotativa com desbalanceamento.



Fonte: O AUTOR

FIGURA 4 – Destaque para os eixos de referência da máquina rotativa



Fonte: O AUTOR

A força resultante desse desbalanceamento gera picos de aceleração na frequência de rotação da máquina, podendo também apresentar valores elevados nas suas frequências múltiplas. Dependendo da intensidade desses picos, sua operação continuada pode causar perdas de eficiência ou, dependendo da intensidade, até falhas graves na máquina. A intensidade dessa força pode ser calculada pela equação 2.2, usando o referencial indicado na Figura 4.

$$\begin{cases} F_x = m \cdot r \cdot \omega^2 \cdot \cos(\alpha) \\ F_y = m \cdot r \cdot \omega^2 \cdot \sin(\alpha) \\ F_z = 0 \end{cases} \tag{2.2}$$

Em que a definição de cada variável dessa equação é:

- $F_{x,y,z}$: Força resultante no eixo x, y, ou z, em Newtons (N);
- m: Massa desbalanceada, em quilos (Kg);
- r: Distância do centro de rotação até o centro de massa, em metros (m);
- ω: Velocidade angular, em radianos por segundo (rad/s)
- α: Ângulo do desbalanceamento, em radianos (rad)

2.2 MODELOS DE APRENDIZADO DE MÁQUINAS

O aprendizado de máquina é uma área da inteligência artificial que tem como objetivo desenvolver algoritmos capazes de identificar padrões de comportamentos baseados nas informações de entrada e saída de um sistema e automatizar tarefas, como a classificação ou a regressão de uma variável desconhecida desse sistema. Essa tecnologia tem se mostrado cada vez mais relevante na era da informação, sendo aplicada em diversas áreas, como análise de dados, previsões de séries temporais, entre outras.

A regressão no aprendizado de máquina é utilizada quando se faz necessário prever o valor numérico de uma ou mais características desconhecidas de um sistema a partir, apenas, dos valores de suas características conhecidas. Já a classificação é utilizada para atribuir uma classe a uma característica desconhecida do sistema (ZHANG et al., 2021). Ou seja, a regressão busca prever uma ou mais características numéricas contínuas, enquanto a classificação busca prever uma ou mais características binárias do sistema.

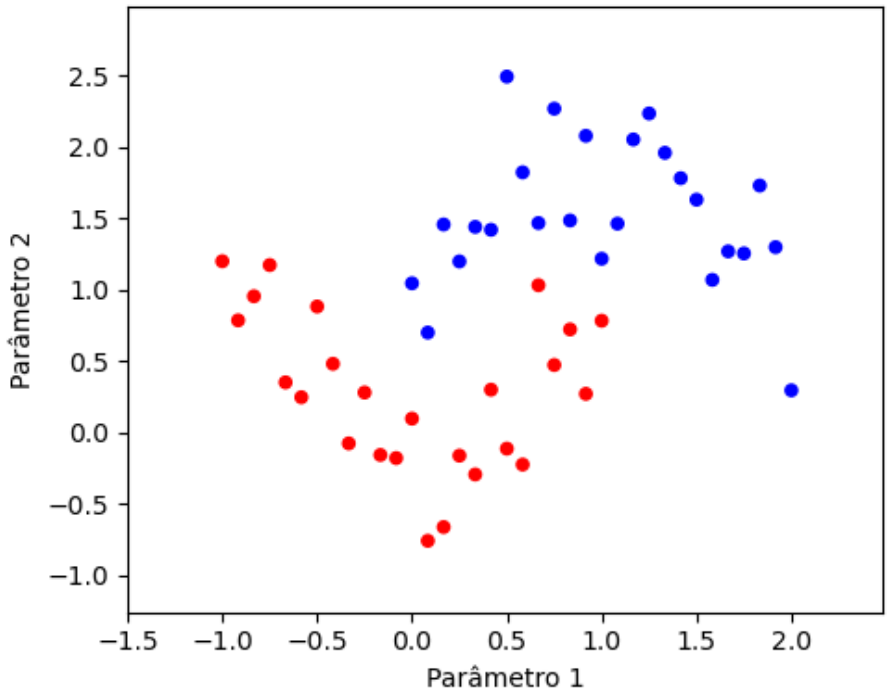
A forma como o modelo de inteligência artificial é capaz de fazer a regressão ou a classificação de um sistema é a partir da extrapolação e/ou interpolação dos seus dados de entrada e saídas conhecidos. A forma como as extrapolações e as interpolações são feitas depende do método de treinamento escolhido para o modelo. Os métodos que serão apresentados a seguir são K-vizinhos mais próximos (do inglês *K-Nearest Neighbors* - KNN), árvore de decisão, floresta aleatória, máquinas de vetores de suporte (do inglês *Support Vector Machine* - SVM) e rede neural. Cada modelo também possui alguns hiper-parâmetros que podem ser otimizados para um melhor resultado final na fase de testes do seu modelo.

Para exemplificação nesse trabalho, foi criado um pequeno banco de dados didático, apresentado na Figura 5, que apresenta duas classes distintas. Ele será utilizado nas próximas subseções para exemplificar a aplicação de diferentes métodos de classificação e permitir uma comparação direta entre eles dentro de um mesmo cenário. Os pontos de cada classe estão distribuídos em formato U e não são linearmente separáveis, o que exige uma solução não trivial dos modelos, exibindo com mais clareza as diferenças entre cada um dos modelos.

2.2.1 K-vizinhos mais próximos (KNN)

O primeiro modelo de aprendizado de máquinas tratado nesse trabalho é o KNN, que pode ser usado tanto para regressão quanto para classificação. Quando utilizado para regressão, esse modelo busca prever o valor numérico de uma resposta a partir da média dos valores de respostas dos K vizinhos mais próximos. Para a

FIGURA 5 – Dados de exemplo com duas classes (antes da classificação)



Fonte: O AUTOR

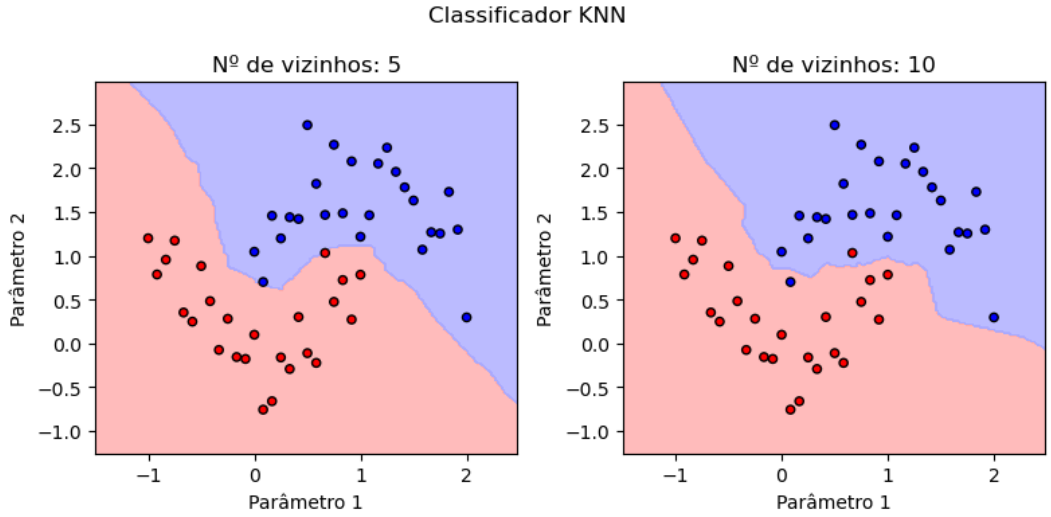
classificação, ele atribui uma nova classe para uma observação desconhecida a partir da classe votada pela maioria entre os K vizinhos mais próximos (ZHANG et al., 2021).

Dois dos parâmetros que podem ser ajustados na regressão por KNN, por exemplo, são o número K de vizinhos e a função de média. Para o número de vizinhos, quanto menor, mais sensível o modelo se torna a *outliers* e outras características locais, enquanto que, quanto maior, mais global o modelo irá tender a ser. A forma como a média dos K vizinhos é calculada pode ser outra além da média simples, como a média ponderada (em função da distância até o ponto), a mediana (CHATTERJEE; MOHANTY, 2020), utilizando a distância Euclidiana ou a de Manhattan (TAUNK et al., 2019).

Na Figura 6 pode-se visualizar esse modelo para classificação no conjunto de dados de exemplo. Nele, o número K de vizinhos observados está sendo variado, sendo igual a 5 para o gráfico da esquerda e 10 para o gráfico da direita. O gráfico mostra que, com menos vizinhos, o modelo se torna mais sensível aos dados locais, principalmente *outliers*, enquanto que, com um maior número de vizinhos, o modelo é menos sensível às variações locais e tende a uma média global.

O modelo KNN se destaca pela sua simplicidade e facilidade de implementação. Ele é um método não paramétrico, ou seja, ele não assume nenhuma distribuição específica dos dados e pode se ajustar a diferentes tipos de dados. Além disso, o KNN

FIGURA 6 – Exemplo de classificação por KNN, variando o hiper-parâmetro de número de vizinhos considerados



Fonte: O AUTOR

não requer qualquer treinamento prévio, o que o torna rápido e eficiente. (TAUNK et al., 2019)

Por outro lado, esse método pode ser sensível a dados ausentes ou inconsistentes, e pode ser afetado por pontos que são fora da curva. Além disso, ele não é indicado para problemas de alta dimensionalidade, por ser computacionalmente intensivo quando há grandes conjuntos de dados.

Apesar de simples, esse método é uma ferramenta poderosa para análise de dados e tomada de decisões em diversas áreas. Ele vem sendo amplamente utilizado em diversas aplicações práticas que vão desde o reconhecimento de padrões até a análise de dados financeiros (ZHANG et al., 2021).

2.2.2 Árvore de decisão

Outro método simples de aprendizado de máquina é a árvore de decisão. Esse modelo é capaz de identificar o valor numérico ou a classe de uma variável desconhecida a partir da construção de uma estrutura em forma de árvore que divide o espaço de um banco de dados em subespaços com base em um conjunto de regras (JIJO; ABDULAZEEZ, 2021).

A árvore de decisão para regressão visa minimizar o erro na previsão dos valores numéricos das respostas a partir da divisão do espaço de dados usando os valores dos parâmetros dos dados de treino. Em cada nó da árvore, o parâmetro que melhor divide o espaço é escolhido de forma a maximizar o ganho de informação e reduzir a variância. O ganho de informação é definido pela métrica escolhida para calcular o erro, sendo as mais comuns o erro quadrático médio e a soma dos resíduos

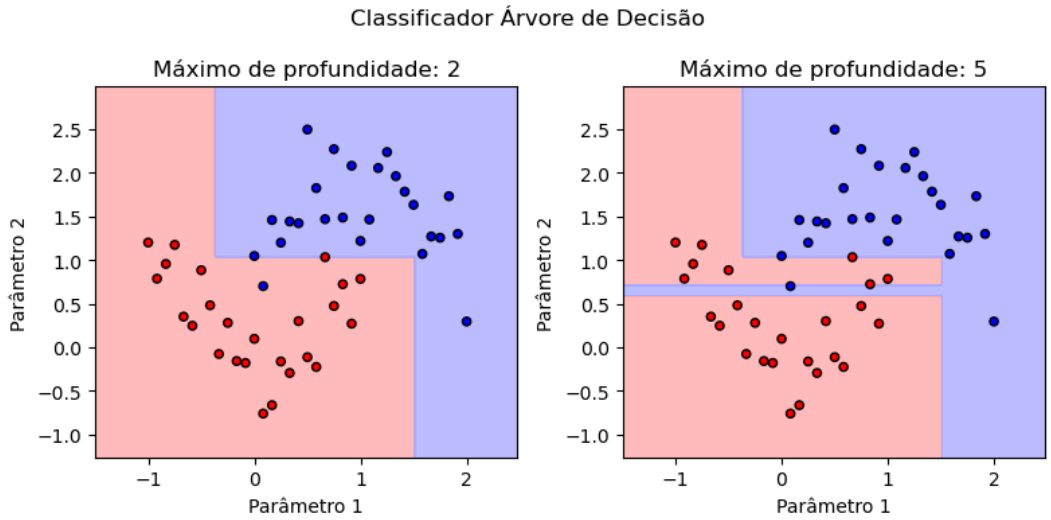
quadráticos (SINGH et al., 2020).

Já no caso do modelo de classificação, a árvore de decisão divide o espaço com base em um dos parâmetros, de forma a obter a melhor separação entre as classes das respostas. Em cada nó da árvore, o parâmetro que melhor separa as classes é escolhido a partir da métrica de entropia, que busca maximizar o ganho de informação e reduzir a incerteza na classificação dos dados.

Tanto para classificação quanto para regressão, cada decisão divide o espaço em dois subespaços em função dos valores de seus parâmetros. Cada folha da árvore possui um único valor de resposta que se aproxima de todos os pontos que contém.

A Figura 7 apresenta uma visualização da classificação por árvore de decisão. Nos gráficos, o hiper-parâmetro que está sendo variado é o número máximo de sub-divisões do espaço. No gráfico da esquerda, apenas 2 níveis de profundidade foram permitidos, enquanto que, à direita, até 5 níveis de profundidade foram permitidos, e mais subdivisões foram feitas no espaço. Se houver poucas divisões, o modelo se torna pouco complexo e, dependendo da base de dados utilizada, pode incorrer no erro de subajuste, não sendo capaz que abstrair com acurácia os dados, enquanto que, se houver muitas subdivisões, o modelo se torna muito complexo e pode incorrer em erro de sobreajuste, que também não será capaz de abstrair seus dados além dos dados de teste.

FIGURA 7 – Exemplo de classificação por árvore de decisão, variando o hiper-parâmetro de profundidade dos ganhos



Fonte: O AUTOR

A árvore de decisão possui as vantagens de ser um método simples e de fácil interpretação, além de ser capaz de lidar com variáveis tanto categóricas quanto numéricas. Além disso, ela é pouco influenciada por valores extremos ou ruídos nos

dados, além de ser eficiente mesmo com dados faltantes.

Apesar disso, os modelos de árvore de decisão possuem a tendência de causar sobreajuste, resultando em baixa generalização para novos dados, especialmente quando a árvore é muito profunda e complexa. Além disso, a árvore de decisão é sensível à escolha da métrica de avaliação, e uma escolha inadequada pode levar a um modelo sub-ótimo.

Por ser um modelo versátil e capaz de lidar com grande volume de dados, ele possui diversas aplicações, como na previsão de preços de imóveis, no diagnóstico de doenças e na previsão de comportamentos de clientes para uma empresa (JIJO; ABDULAZEEZ, 2021).

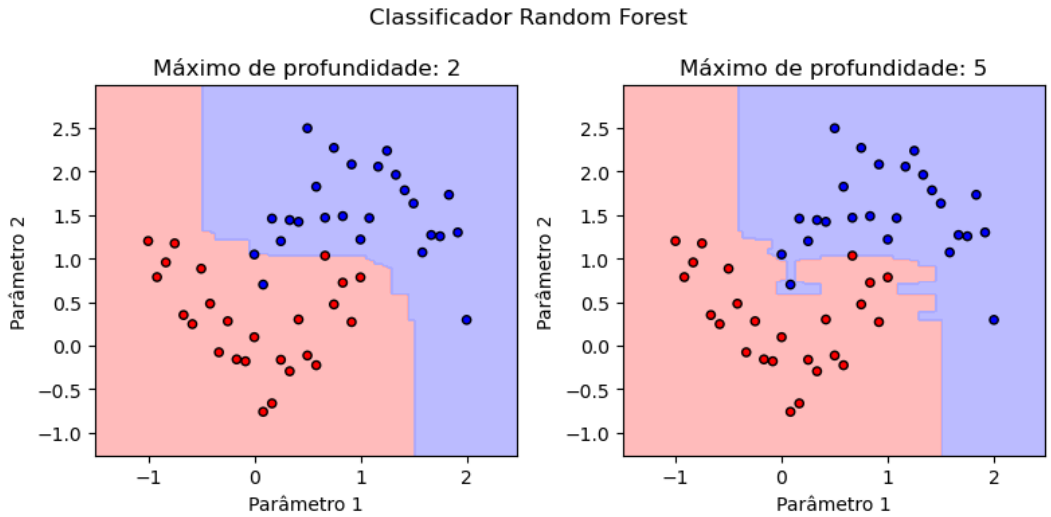
2.2.3 Floresta aleatória

A floresta aleatória é um conjunto de modelos menores de aprendizado de máquina no formato *bagging*, usando múltiplas árvores de decisão como base. Ou seja, ele combina múltiplas árvores de decisão independentes, cada uma treinada em um subconjunto diferente de dados e/ou com diferentes subconjuntos de parâmetros, para que o conjunto final tenha um aumento na precisão e uma redução no sobreajuste (RAZA et al., 2019).

Como esse modelo é composto da média de várias árvores de decisão, ele costuma ter resultados melhores do que uma árvore de decisão individual, pelo seu alto poder de generalização. A previsão é feita pela agregação dos resultados individuais de cada uma delas e, para a classificação, essa agregação é feita a partir de uma votação onde a classe mais frequente se sobressai, enquanto que, para a regressão, a agregação é uma média dos valores de respostas. (LIU et al., 2019)

A Figura 8 mostra dois modelos de classificação utilizando o modelo de exemplo descrito anteriormente, cada um com um número máximo de profundidade diferente. À esquerda está apresentada uma floresta aleatória formada apenas com árvores de até 2 níveis de profundidade, enquanto que, à direita, foi formada com árvores de até 5 níveis de profundidade.

FIGURA 8 – Exemplo de classificação por floresta aleatória, variando o parâmetro de profundidade máxima dos galhos



Fonte: O AUTOR

As principais vantagens do modelo de floresta aleatória, tanto na classificação quanto na regressão, são a redução do sobreajuste quando comparada com uma única árvore de decisão, a baixa sensibilidade à *outliers* e dados ausentes e a possibilidade de avaliar a relevância de cada parâmetro.

No entanto, quando comparada às árvores de decisão, a floresta aleatória possui uma maior dificuldade de interpretação do modelo. Além disso, o treinamento do modelo pode ser computacionalmente intensivo, especialmente quando o número de árvores ou o número de parâmetros é grande.

As aplicações da floresta aleatória são parecidas com as de uma árvore de decisão, sendo comumente usada em áreas como finanças, biologia, ciência dos materiais e marketing (LI et al., 2021).

2.2.4 Maquinas de vetores de suporte (SVM)

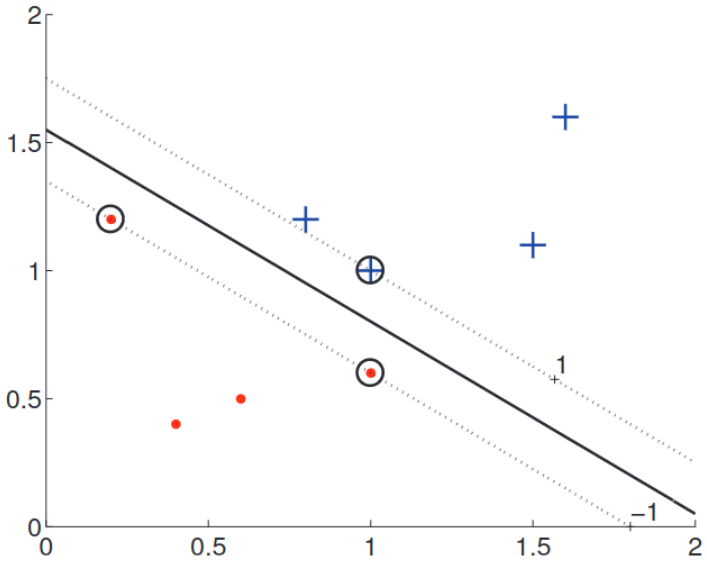
As maquinas de vetores de suporte (SVMs) são algoritmos de aprendizado de máquina baseados no conceito de encontrar os hiperplanos que melhor representam o comportamento de um conjunto de dados. Esse modelo pode ser usado tanto para classificação, chamado classificador por vetores de suporte (SVC), quanto para regressão, chamado de regressor por vetores de suporte (SVR) (LI et al., 2020).

O SVR, usado para regressão, funciona encontrando um hiperplano que minimiza a diferença entre as observações e o próprio hiperplano. Ele tem como objetivo prever os valores numéricos das respostas das observações, como mostrado na Figura 9, representado pela reta preta. Na figura, as cruzes azuis e pontos vermelhos são

pontos de duas classes distintas sendo divididas por essa reta.

Esse hiperplano é encontrado minimizando a diferença entre as observações e o hiperplano, usando a menor margem capaz de englobar o maior número possível de observações. As observações que ficam fora da margem são consideradas outliers e o peso de cada uma delas em relação ao tamanho da margem pode ser modificado para um ajuste fino do modelo, sacrificando a precisão do hiperplano para englobar mais observações ou vice-versa. Todas as observações do conjunto de dados são consideradas como dados de suporte e são usados para definir o hiperplano (LU et al., 2020).

FIGURA 9 – Vetores de suporte dividindo um espaço, classificando cada subespaço correspondente.



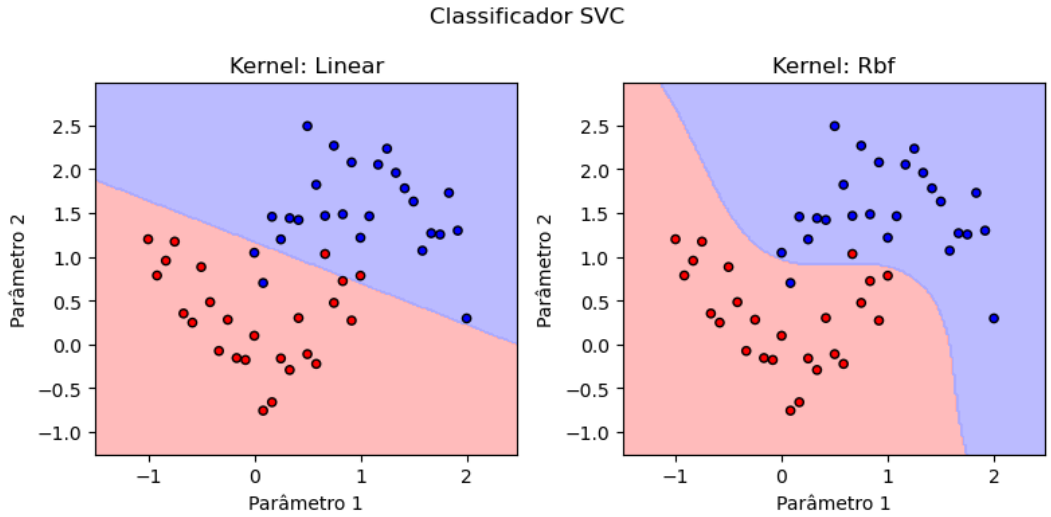
Fonte: (ALPAYDIN, 2020)

Para ser possível introduzir não-linearidade na função de previsão, são utilizadas as chamadas Funções de Kernel (ALPAYDIN, 2020). Cada função possibilita o desenho da função de previsão de formas distintas, a depender do problema que se deseja resolver. Alguns exemplos de funções de kernel incluem a função linear, a função polinomial e a função RBF (*Radial Basis Function*), essa última sendo considerada a mais versátil e que melhor se adapta a diversos conjuntos de dados.

Já o SVC para classificação, funciona por encontrar os melhores hiperplanos que separam as classes. O melhor hiperplano é aquele que maximiza a distância entre os hiperplanos e as observações mais próximas de cada classe. Os pontos que ficam mais próximos do hiperplano são chamados de vetores de suporte e são usados para definir o hiperplano. Os SVCs podem lidar com dados de alta dimensão e são eficazes em lidar com dados não linearmente separáveis.

A Figura 10 mostra uma visualização da classificação por SVR com diferentes funções de kernel para o mesmo exemplo utilizado nas subseções anteriores. O gráfico à esquerda mostra uma função linear sendo aplicada, que traça uma fronteira de decisão linear, e o gráfico à direita mostra uma função RBF, que permite uma fronteira de decisão não linear.

FIGURA 10 – Exemplo de classificação por SVM, variando o hiper-parâmetro de função do kernel



Fonte: O AUTOR

As vantagens dos SVMs incluem a capacidade de lidar com dados de alta dimensionalidade e não-lineares, de serem robustos contra sobreajuste e poderem lidar com dados desbalanceados, ou seja, onde as classes possuem tamanhos diferentes. No entanto, SVMs possuem algumas desvantagens, como a de serem computacionalmente intensivos e serem muito sensíveis aos seus hiperparâmetros.

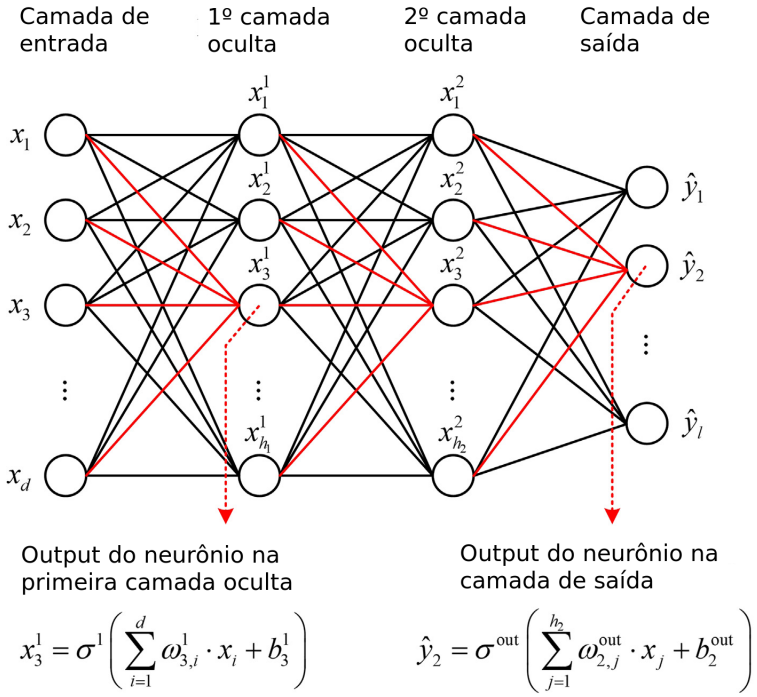
O SVM vem sendo aplicado em áreas como a classificação de texto, detecção de spam, detecção de fraude em cartões de crédito, diagnóstico médico e previsão de preços de ações, entre outras (LI et al., 2020).

2.2.5 Rede Neural Artificial (ANN)

Por fim, o modelo de regressão de aprendizado de máquinas com maior capacidade de representação é a rede neural artificial (do inglês Artificial Neural Network - ANN) que, assim como a rede neural humana, é composta por camadas de neurônios que processam informações e aprendem a partir de dados de entrada. Essas informações são processadas e transformadas a partir de pesos, vieses e funções de ativação. As funções de ativação são funções matemáticas que são aplicadas nos neurônios para modificar seu valor numérico e produzir um comportamento não-linear (APICELLA et al., 2021). Uma rede neural densa, ou seja, uma rede onde todos os

neurônios de uma camada se conectam com todos os neurônios da próxima camada, pode ser vista na Figura 11. Nas equações, σ representa a função de ativação para cada camada, $\omega_{i,j}$ representa o peso de um nó e $b_{i,j}$ o seu viés.

FIGURA 11 – Representação de uma rede neural densa



Fonte: (LEI et al., 2020)

Um neurônio é a unidade mais básica de uma rede neural artificial, representada como um nó da rede neural. Ele recebe os sinais de entrada da camada anterior e retorna um sinal de saída, esse sinal é dado pelo somatório de cada uma das entradas, ponderados por pesos sinápticos e somados com um viés final e, dependendo da soma desses sinais ponderados, produz uma saída que pode ser ativada ou não. Os pesos sinápticos ω são valores numéricos atribuídos às conexões entre os nós de diferentes camadas e seu papel é indicar a importância relativa de cada entrada no processo de decisão do neurônio. Já o viés b é o valor numérico que é adicionado à soma ponderada dos sinais de entrada, oferecendo ainda mais flexibilidade no modelo para representar diferentes classes de dados. E a função de ativação é uma função não-linear aplicada às camadas para permitir o comportamento não linear dos valores de resposta em relação aos valores de entrada. Com isso, os pesos, vieses e funções de ativação são os parâmetros que uma rede neural utiliza para ajustar sua resposta a diferentes entradas e, assim, realizar tarefas de classificação ou regressão.

O fluxo de dados é dado pelos seguintes passos:

1. Os nós da camada de entrada recebem os valores de entrada;

2. Cada nó da próxima camada é uma combinação linear dos valores da camada anterior, compostas pelos pesos e vieses dessa camada;
3. O resultado da soma é então passado por uma função de ativação, que é a mesma para todos os elementos de uma coluna. A função de ativação pode variar de coluna pra coluna;
4. Voltar ao passo 2 até que se chegue à última coluna
5. Os valores nos nós de saída serão os saídas da rede neural.

Na tabela 1, há alguns exemplos de funções de ativações que uma camada da rede neural pode ter.

TABELA 1 – EXEMPLOS DE FUNÇÕES DE ATIVAÇÃO

Função de ativação	Fórmula	Domínio da entrada	Domínio da saída
Linear	$f(x) = x$	$(-\infty, \infty)$	$(-\infty, \infty)$
Relu	$f(x) = max(0, x)$	$(-\infty, \infty)$	$(0, \infty)$
Tanh	$f(x) = tanh(x)$	$(-\infty, \infty)$	(-1, 1)
Sigmoidal	$f(x) = 1/(1 + e^{-x})$	$(-\infty, \infty)$	(0, 1)

FONTE: (APICELLA et al., 2021)

Para classificação, as redes neurais transformam os dados de entradas através das camadas, a partir da combinação dos valores numéricos de cada nó com os pesos e vieses de cada ligação, além das funções de ativação, até chegar à saída, onde o resultado é a probabilidade de aqueles dados de entrada pertencerem a cada classe. Nesse caso de classificação binária, é comum que a última camada tenha uma função de ativação sigmoidal, que transforma qualquer valor real em um valor de 0 a 1 (SINGH et al., 2020). No caso de classificação não-binária, a função de ativação deve ser a *softmax*, que garante que a soma das probabilidades de cada classe seja sempre 100%.

Já para regressão, as redes neurais procuram ajustar seus parâmetros para que a entrada seja transformada de tal modo que a sua saída tenha o mínimo de erro possível. Para terem melhores resultados, as redes neurais podem ter diferentes arquiteturas aplicadas, como *feedforward*, convolucionais ou recorrentes, e todas elas se utilizam do gradiente descendente como o algoritmo para otimização de seus resultados (SINGH et al., 2020).

As vantagens das redes neurais são várias: capacidade de lidar com dados complexos e não lineares, excelente capacidade de generalização, ou seja, a capacidade de aplicar o aprendizado em dados novos, e poder aprender características relevantes automaticamente. No entanto, para que tenham uma boa assertividade, é

necessário haver uma grande quantidade de dados para treino com pouco ruído, por ser muito sensível à dados ruidosos.

Hoje, as redes neurais são aplicadas nas mais variáveis áreas, como o reconhecimento de voz, diagnóstico médico, detecção de fraudes em transações financeiras, reconhecimento de imagens, entre várias outras, demonstrando sua alta capacidade de modelar os mais variados tipos de sistemas. (APICELLA et al., 2021).

2.3 GRADIENTE DESCENDENTE

O treinamento dos modelos de aprendizado de máquinas tem como objetivo minimizar uma função de perda para que esta tenha o menor erro possível dentro de um problema dado. Por isso, utilizar métodos de otimização robustos é crucial para que o modelo final tenha sucesso em sua tarefa.

A taxa de variação que a função de perda possui em relação a cada parâmetro da rede é o chamado gradiente da rede neural. Ele é calculado a partir da derivada parcial da função de perda em relação a cada parâmetro da rede. Controlar a variação dos parâmetros de uma rede neural de forma correta a cada iteração do seu treinamento é de grande importância, pois ela influência no resultado do modelo final. Um gradiente mal otimizado pode causar modelos com sobreajuste ou subajuste (FJELLSTRÖM; NYSTRÖM, 2022).

Já o gradiente descendente é um algoritmo de otimização usado para controlar o gradiente de uma rede neural enquanto minimiza a sua função de perda. Ele funciona encontrando a direção de descida mais acentuada na função de perda e ajustando os parâmetros da rede nessa direção, repetindo esse processo até que a função de perda seja a menor possível.

Ele é um método que pode ser amplamente utilizado em modelos de aprendizado de máquinas quando se deseja treinar com eficiência longas séries temporais e ajustar os parâmetros de uma rede de forma que seu gradiente não tenha valores inadequados.

2.4 REDES NEURAIS RECORRENTES

As redes neurais recorrentes (RNN) possuem o diferencial de permitir que o fluxo de informação retorne à camadas anteriores da sua rede, enquanto que as redes neurais tradicionais (*forward*) permitem que seus dados sigam em apenas uma direção. Isso permite, para as redes recorrentes, aplicar uma mesma transformação múltiplas vezes a um sinal, entretanto, se houver uma má calibração dos parâmetros da rede, é comum ocorrer o problema de desvanecimento ou de explosão do gradiente durante o treinamento da rede. Isso acontece pois, quando realizamos uma mesma transformação

linear múltiplas vezes, o seu valor passa a ser sempre muito alto (quando os pesos são maiores que 1) ou muito baixo (quando os pesos são menores que 1) (ZARZYCKI; ŁAWRYŃCZUK, 2022). Isso é chamado de Problema do Gradiente Descendente. Novas arquiteturas de redes neurais foram criadas, em parte, para resolver este problema.

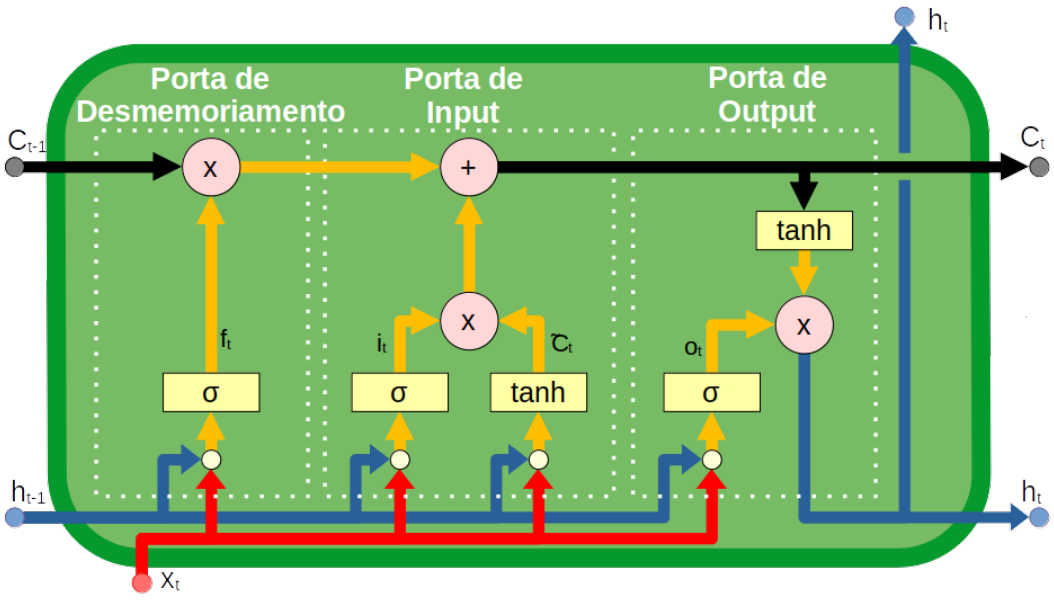
2.4.1 LSTM

O LSTM é um tipo especial de rede neural recorrente que permite o processamento do gradiente no treinamento da rede sem que haja o problema de desvanecimento ou de explosão. Ao contrário de outras redes recorrentes, o LSTM possui uma memória que lhes permite lembrar do valor de saída da última iteração e utilizá-la como entrada para encontrar o valor numérico resultante da próxima iteração (MA et al., 2021).

O diferencial das redes LSTM é a presença de dois novos vetores: o vetor de memória longa (C_t) e o vetor de memória curta (h_t). Esses vetores funcionam para modificar os pesos do próprio nó e evitar que o valor fique muito defasado após múltiplas iterações. O nome vem da abreviação de *Long-Short Term Memory*, que significa os dois tipos de memórias que essa rede possui: uma memória de curto prazo para eventos recentes e uma de longo prago para eventos importantes mais antigos.

A Figura 12 mostra os detalhes da arquitetura de um nó dessa rede LSTM e a Tabela 2 mostra as equações que regem as partes mais importantes dessa célula.

FIGURA 12 – Representação de um nó da rede LSTM



Fonte: O autor

Na imagem, as setas em preto representam o fluxo do valor da memória longa (C_{t-1} e C_t), em azul está o fluxo da memória curta (h_{t-1} e h_t), em vermelho o fluxo dos

valores de entrada e em amarelo o fluxo de valores intermediários entre a entrada e a saída do sistema. As áreas demarcadas pelas linhas brancas pontilhadas encasulam as 3 partes de um nó do LSTM: a porta de desmemoriamento, a porta de entrada e a porta de saída. Cada porta aplica uma transformação específica nos vetores e memória com o objetivo de prever os valores esperados. A tabela 2 mostra os detalhes de todas as transformações que ocorrem dentro de cada nó.

TABELA 2 – VARIÁVEIS DE UMA REDE LSTM

Variável	Símbolo	Definição
Vetor de entrada	x_t	x_t
Vetor entrada da Memória curta	h_{t-1}	h_{t-1}
Memória	C_{t-1}	C_{t-1}
Vetor de pesos do entrada	U	U
Vetor de pesos da Memória curta	W	W
Função de ativação Sigmoide	σ	$\sigma(x) = \frac{1}{1+e^{-x}}$
Função de ativação Tanh	$tanh$	$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
Porta de desmemoriamento	f_t	$f_t = \sigma(x_t U^f + h_{t-1} W^f)$
Porta de entrada	i_t	$i_t = \sigma(x_t U^i + h_{t-1} W^i)$
Porta de saída	o_t	$o_t = \sigma(x_t U^o + h_{t-1} W^o)$
Candidato à memória	$\tilde{C}_t$	$\tilde{C}_t = tanh(x_t U^g + ht - 1W^g)$
Vetor de saída	C_t	$C_t = \sigma(f_t * C_{t-1} + + i_t * \tilde{C}_t)$
Vetor saída da memórica curta	h_t	$h_t = tanh(C_t * o_t)$

FONTE: O AUTOR
LEGENDA: Operações que cada variável passa ao ser processada por uma rede LSTM

As funções de ativação utilizadas em uma rede LSTM são a função sigmoid e a função tanh (tangente hiperbólica). A primeira função, a sigmoid, é uma função que transforma qualquer valor real em um número entre 0 e 1, que na rede LSTM é utilizada para transformar um número real qualquer em um peso ou em uma porcentagem de 0% a 100% (APICELLA et al., 2021). Sua definição é dada pela equação 2.3.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

Ao serem transformados por essa função de ativação, valores positivos se aproximam de 1, valores negativos se aproximam de 0 e valores perto de zero se aproximam de 0,5. Ela é usada em suas 3 portas: Desmemoriamento, entrada e saída. Seu valor de 0 a 1 funciona como uma porcentagem que define o quanto que um valor irá passar de uma porta para outra.

Já a função tangente hiperbólica é uma função que transforma qualquer valor real em um número entre -1 e 1 (APICELLA et al., 2021). Na rede LSTM ela é usada como ponto de partida para se calcular a intensidade com que um nó será ajustado. Sua definição é dada pela equação 2.4.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.4}$$

Ao ser transformado por essa função de ativação, valores positivos se aproximam de 1, valores negativos se aproximam de -1 e valores perto de zero se aproximam de 0.

Nas redes LSTM essa função de ativação é usada em 2 de suas 3 portas: Entrada e saída. Seu valor de 0 a 1 funciona como uma porcentagem que define o quanto que um valor irá passar de uma porta para outra.

Essa rede passa por várias iterações durante seu aprendizado, atualizando os valores de seus parâmetros para que cada vez mais suas saídas estimadas se aproximem das saídas reais, e para que os parâmetros da memória passem a ser cada vez mais eficientes em identificar quais dados são mais importantes ou menos importantes de serem guardados e reutilizados no modelo.

2.4.2 Porta de Desmemoriamento

A porta de desmemoriamento tem a função de servir para calcular o peso do vetor de memória longa. Dada as seguintes entradas:

- Entradas:

  C_{t-1}: Memória longa

  h_{t-1}: Memória curta

  x_t: Entrada

  W_1: Peso 1 da memória curta

  U_1: Peso 1 da entrada

  b_1: Viés 1 do nó

  O passo a passo para calcular esse peso é:

1. Multiplicar a memória curta por seu peso;
2. Multiplicar a entrada por seu peso;
3. Somar o resultado das duas operações acima e adicionar também o viés;
4. Passar o resultado da operação acima por uma função de ativação sigmoide (f_t);

   O valor resultante será sempre um valor entre 0 e 1.
5. Multiplicar o resultado pelo valor da Memória Longa.

- Saídas:

  C'_{t-1}: Parte da memória longa que foi passada adiante

  A operação toda também pode ser definida pela equação 2.5

$$C'_{t-1} = C_{t-1} \cdot \sigma(W_1 h_{t-1} + U_1 x_t + b_1) \tag{2.5}$$

Essa etapa da transformação altera apenas o valor da memória longa e serve para escalá-la em função de quanto de seu valor deve ser passado adiante.

2.4.3 Porta de entrada

A porta de entrada tem a função de servir para calcular o viés do vetor de memória longa. Dada as seguintes entradas:

- Entradas:

  C'_{t-1}: Estado atual da memória longa

  h_{t-1}: Memória curta

  x_t: Entrada

  W_2: Peso 2 da memória curta

  U_2: Peso 2 da entrada

  b_2: Viés 2

  W_3: Peso 3 da memória curta

  U_3: Peso 3 da entrada

  b_2: Viés 3

  O passo a passo para calcular esse viés é:

1. Calcular i_t

   Multiplicar a memória curta por seu peso 2;

   Multiplicar a entrada por seu peso 2;

   Somar o resultado das duas operações acima e adicionar também o viés 2;

   Passar o resultado da operação acima por uma função de ativação sigmoide;

   O valor resultante será sempre um valor entre 0 e 1.

2. Calcular $\tilde{C}_t$

   Multiplicar a memória curta por seu peso 3;

   Multiplicar a entrada por seu peso 3;

   Somar o resultado das duas operações acima e adicionar também o viés 3;

   Passar o resultado da operação acima por uma função de ativação tangencial hiperbólica;

   O valor resultante será sempre um valor entre -1 e 1.

3. Multiplicar i_t por $\tilde{C}_t$;

4. Somar o resultado com o estado atual da memória longa C'_{t-1} e o viés 2 b_2

- Saídas:

  C_t: Novo estado da memória longa

Essa porta define a parte da memória curta que será adicionada à memória longa.

A operação toda também pode ser definida pela equação 2.6

$$C_t = C'_{t-1} + \big(\sigma(W_2 h_{t-1} + U_2 x_t + b_2) \cdot \tanh(W_3 h_{t-1} + U_3 x_t + b_3)\big) \tag{2.6}$$

2.4.4 Porta de saída

A porta de saída tem a função que calcula o novo estado da memória curta. Para os dados de entrada:

- Entrada:

  C_{t-1}: Novo estado da memória longa

  h_{t-1}: Memória curta

  x_t: Entrada

  W_2: Peso 4 da memória curta

  U_2: Peso 4 da entrada

  b_2: Viés 4

O passo a passo para calculá-lo é:

1. Calcular o_t

   Multiplicar a Memória curta por seu peso 4;

   Multiplicar a entrada por seu peso 4;

   Somar o resultado das duas operações acima e adicionar também o viés 4;

   Passar o resultado da operação acima por uma função de ativação sigmoide;

   O valor resultante será sempre um valor entre 0 e 1.

2. Passar o valor do novo estado da memória longa C_t pela função de ativação tangencial hiperbólica;

3. Multiplicar o resultado da função de ativação por o_t.

- Saídas:

  h_t: Novo estado da memória curta

  h_t: Saída

Essa porta define o estado final no nó para a memória curta. A saída é igual ao novo estado da memória curta.

A operação toda também pode ser definida pela equação 2.7.

$$h_t = \tanh(C_{t-1}) \cdot \big(\sigma(W_4 h_{t-1} + U_4 x_t + b_4)\big) \tag{2.7}$$

Os modelos de redes recorrentes LSTM são amplamente utilizados em problemas de previsão de séries temporais. Essas redes são capazes de lidar com essas séries temporais graças à sua habilidade de aprender e de lembrar informações relevantes de longo prazo, ao mesmo tempo em que descartam informações irrelevantes.

Além disso, as redes LSTM também têm sido aplicadas em problemas de classificação de sequência, processamento de linguagem natural, reconhecimento de fala, entre outras. Isso torna a LSTM uma ferramenta poderosa para solucionar problemas complexos em várias áreas de pesquisa (APICELLA et al., 2021).

3 REVISÃO DA LITERATURA

À medida que o cenário de manutenção de máquinas rotativas evolui, cresce também a demanda por métodos mais avançados de análise de defeitos. Este capítulo tem o objetivo de apresentar uma revisão crítica de abordagens já existentes para a detecção e previsão de defeitos em máquinas rotativas, e como essas técnicas estão evoluindo e ficando cada vez mais eficientes em suprir com as necessidades existentes do mercado.

3.1 DETECÇÃO DE DEFEITOS SEM O USO DE APRENDIZADO DE MÁQUINAS

Um dos métodos comuns de detecção de defeitos sem o uso de aprendizado de máquinas é a identificação de vibrações excessivas, que é realizada em 3 etapas: aquisição de dados, extração dos parâmetros e detecção de defeitos (TANDON; CHOUDHURY, 1999). A aquisição de dados é feita com a ajuda de sensores, como acelerômetros, transdutores de velocidade, sensores de proximidade, entre outros. Esses dados coletados são frequentemente contaminados por sinais de ruídos, por isso eles devem ser pré-processados e assim destacar as características principais do comportamento da máquina para a extração dos parâmetros.

Os parâmetros extraídos podem ser divididos em 3 grupos: no domínio do tempo, no domínio da frequência e no domínio do tempo-frequência. No domínio do tempo, os parâmetros mais comuns são a média, a raiz da média quadrática (RMS) e o fator de crista (TANDON; CHOUDHURY, 1999). No domínio da frequência, obtido a partir da transformada de Fourier, é possível observar o sistema a partir de um novo ponto de vista, e seus parâmetros mais comuns são as frequências de pico do seu espectro de potência. (YANG et al., 2002). Os métodos que utilizam o domínio de tempo-frequência utilizam parâmetros de ambos os domínios.

Após a extração dos parâmetros é feita uma regressão desses valores para detectar se há ou não a presença de defeitos. Os pesquisadores McFadden, Smith (MCFADDEN; SMITH, 1985) e Kim (KIM et al., 1995) combinaram alguns dos métodos já estabelecidos, como análise espectral não-paramétrica, análise de componentes principais (PCA), análise conjunta de tempo-frequência e transformada wavelet discreta para essa regressão. Lebold e McClintc (LEBOLD et al., 2000) utilizaram métodos estatísticos para diagnóstico de falhas em caixas de engrenagens. Tendon e Chaudharyn (TANDON; CHOUDHURY, 1999) utilizaram técnicas de medição acústica e de vibração para análise de falhas em rolamentos de máquinas rotativas. Tashin Doguer e Jens Strackeljan (CHOW, 2000) utilizaram os próprios parâmetros no domínio do tempo em

si para identificar pequenos defeitos de rolamentos.

A detecção de defeitos da forma tradicional, com métodos clássicos de análise de sinais e processamento de imagens, tem grande relevância na sua área, e com bom rendimento, mas seu alto custo financeiro e a mão-de-obra especializada necessária se tornaram muito onerosos quando comparados com métodos recentes que utilizam aprendizado de máquinas para os mesmos fins, com eficiência igual ou até maior do que os métodos clássicos.

3.2 DETECÇÃO DE DEFEITOS COM O USO DE APRENDIZADO DE MÁQUINAS

É notável a grande dificuldade de fazer a regressão de sistemas complexos, caso não sejam facilmente modelados por sistemas lineares. O aprendizado de máquinas surge para cumprir esse papel de descobrir a forma mais eficiente de realizar essa regressão e assim obter os resultados mais eficientes possíveis, sem a necessidade de supervisão humana, tornando o processo mais acessível para um maior número de pessoas.

O procedimento utilizando aprendizado de máquinas possui, igualmente, 3 etapas: a aquisição dos dados, a extração dos parâmetros e a detecção de defeitos. A diferença está entre a segunda e terceira etapa em que, ao invés de os métodos a serem utilizados terem que ser selecionados manualmente, eles são criados pela própria máquina que foi treinada para o conjunto de dados específico dado a ela.

O aprendizado de máquinas pode ser otimizado até para cenários com menos dados, garantindo o aproveitamento dos dados disponíveis e permitindo que a quantidade dos dados na etapa de aquisição seja menor. Xianping Zhong e Heng Ban, em 2022, realizaram um estudo de caso onde analisaram a integridade das máquinas do interior de uma central nuclear, (ZHONG; BAN, 2022), fazendo uso desta capacidade do aprendizado de máquinas de generalizar com poucos dados.

Eles observaram que havia pouco volume de dados disponíveis dentro da central, e os que tinham possuíam muito ruído, por isso os métodos tradicionais não seriam eficientes nesse cenário. Com a utilização do aprendizado de máquina e desenvolvimento de modelos específicos para o caso, os pesquisadores foram capazes de conseguir resultados satisfatórios para o banco de dados disponível, demonstrando a versatilidade dessa nova tecnologia.

Em 2023, os pesquisadores Lele Li, Jiawang Xu e Juguang Li, em seu artigo (LI et al., 2023), desenvolveram uma aplicação do aprendizado de máquinas específico para a previsão do tempo de vida útil restante (RUL) de uma máquina rotativa para a manutenção preditiva. O RUL é a capacidade de estimar por quanto tempo uma máquina rotativa ainda será capaz de realizar a sua operação de forma eficiente, antes

que ocorram falhas críticas ou outras avarias. Essa métrica tem grande importância na manutenção preditiva, pois permite a antecipação de potenciais problemas e a otimização dos processos de manutenção.

Nesse estudo, eles consideraram duas abordagens diferentes de fazer a previsão: uma baseada em modelos físicos e outra baseada em modelo de dados. As abordagens baseadas em física foram construídas com base em simulações das leis e princípios físicos, utilizando equações matemáticas que permitem prever o comportamento do sistema. Já os modelos baseados em dados foram gerados a partir da análise de grandes conjuntos de dados para identificar padrões e relações entre as entradas e saídas de um sistema.

Embora os métodos baseados em simulações físicas tenham sido eficazes, eles apresentaram dificuldade em prever falhas em cenários muito complexos. Por outro lado, os métodos baseados em dados usaram tecnologias avançadas de sensoriamento e processamento de sinais, a fim de conseguirem prever qualquer tipo de cenário, simples ou complexo. Entretanto, enfrentaram dificuldade em relação ao volume de dados, à extração de parâmetros e ao processamento de sinais. (LI et al., 2023)

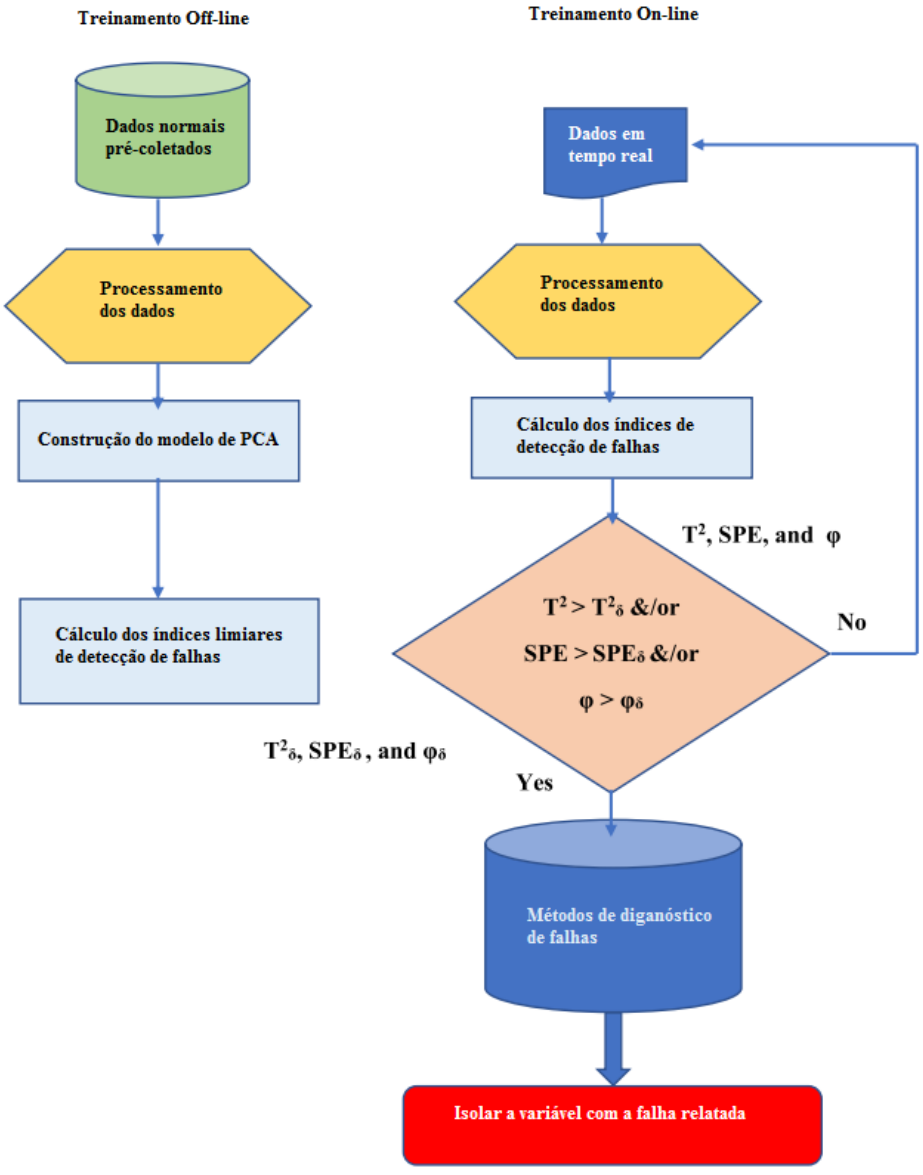
O estudo utilizou métodos de pré-processamento para o aprendizado profundo e redes recorrentes de longa duração para melhorar a precisão na previsão do RUL, são elas: dizimação, análise de componentes principais (PCA), seleção de parâmetros e combinação de redes convolucionais separáveis e de redes recorrentes, todos com o objetivo de reduzir ruídos, e de destacar as informações principais para, assim, melhorar a precisão da previsão de RUL.

No contexto desse trabalho, a dizimação é a prática de reduzir a quantidade de dados em uma amostra para simplificar o processamento e assim ter maior eficiência do modelo. A Análise de Componentes Principais (PCA) é a técnica que aplica transformações lineares em um banco de dados para desacoplar seus componentes, deixando-os independentes entre si, com o objetivo de simplificar a complexidade dos dados. A seleção de parâmetros é uma estratégia para escolher as variáveis mais relevantes, eliminando redundâncias e aprimorando a eficiência do modelo. As Redes Convolucionais Separáveis buscam reduzir a quantidade de parâmetros e dos custos computacionais, quando comparadas com CNNs tradicionais. Por fim, as Redes Recorrentes tem a função de lidar com sequências temporais de dados, mantendo uma memória interna, sendo particularmente eficazes em tarefas que envolvem padrões temporais.

O treinamento foi realizado de forma offline, ou seja, os dados foram obtidos anteriormente e não em tempo real, consistindo na coleta e pré-processamento dos dados para a geração das respostas e parâmetros. Já a implementação foi realizada de modo online, ou seja, com dados em tempo real, para comparar com os valores

encontrados no treino e, a partir dos valores de seus parâmetros, identificar a presença ou ausência de defeitos. O passo a passo da metodologia para cada uma dessas etapas está mostrada na figura 13.

FIGURA 13 – Fluxograma do processo de identificação de defeitos utilizando redes Neurais não supervisionadas, de (LI et al., 2023)



Fonte: Adaptado de (LI et al., 2023)

A decisão de fazer o treinamento offline e a implementação online foi utilizada para otimizar a eficácia no diagnóstico de saúde de máquinas rotativas. A coleta e pré-processamento offline dos dados durante o treinamento permitem uma análise aprofundada, facilitando a identificação dos padrões de defeitos. Já a implementação online oferece a vantagem de poder avaliar o desempenho do modelo em situações dinâmicas, comparando os resultados obtidos em tempo real com os valores previstos

durante o treinamento.

Esta abordagem é necessária quando a detecção em tempo real for crucial para a intervenção preventiva. Com o modelo sendo executado em tempo real, é possível realizar rapidamente o diagnóstico de saúde da máquina, permitindo ações imediatas para evitar possíveis danos ou paradas não programadas. No entanto, essa estratégia ainda possui seus desafios, como por exemplo, a necessidade de processamento em tempo real exige uma grande demanda de poder computacional. Além disso, a validação contínua do modelo em cenários dinâmicos é fundamental para garantir a confiabilidade e eficácia contínua do sistema de prognóstico.

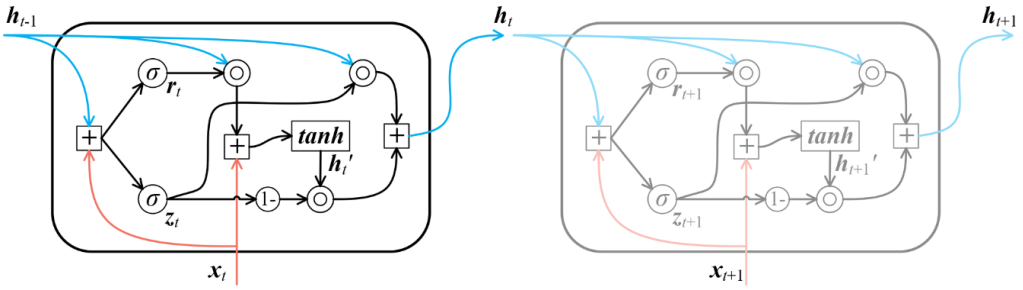
Uma outra estratégia inovadora feita pela mesma equipe, no mesmo ano, analisou a progressão da degradação em máquinas rotativas e a utilizou como o seu parâmetro de previsão. Eles abordaram a análise e o gerenciamento de saúde dessas máquinas como peça chave para os avanços nos processos de fabricação e a manutenção sustentável (LI et al., 2023).

O estudo propõe uma metodologia multitécnica que integra vários modelos de aprendizado de máquinas e aprendizado profundo, como *Relevance Vector Machine* (RVM), *Self-Organizing Convolutional Network* (SCN) e *Gated Recurrent Unit* (GRU). Ele utiliza os dados brutos de múltiplos sensores e de suas dependências temporais para diferentes estados de operação, e é realizado em duas fases: detecção de anomalias e previsão do tempo de vida útil restante (RUL).

O modelo RVM é um modelo de aprendizado de máquina baseado no SVM, mas com uma abordagem probabilística, especializado em lidar com problemas de regressão e classificação. Ele trabalha identificando os vetores de suporte que são essenciais para a decisão, tornando-o eficiente em situações onde há dados esparsos ou ruidosos. O modelo SCN é uma rede neural convolucional que se auto-organiza durante o treinamento, adaptando-se automaticamente aos padrões dos dados. O modelo GRU é uma unidade recorrente com mecanismos de portas lógicas capazes de encontrar e interpretar dependências temporais de longo prazo em sequências de dados. Essas técnicas, integradas de maneira multitécnica, foram utilizadas para aprimorar a eficácia na detecção de anomalias e a previsão do RUL em máquinas rotativas (LI et al., 2023).

A primeira fase foi implementada usando a técnica *PCA-Pooling* e o classificador RVM, enquanto a segunda fase utiliza um estimador de regressão baseado em aprendizado profundo. Este método visa superar as limitações dos modelos de análises existentes, que dependem de projetos de recursos manuais e não consideram da maneira adequada a fase da falha de um ciclo de vida completo. A figura 14 mostra a representação da rede neural recorrente utilizada.

FIGURA 14 – Representação da rede recorrente utilizada por (LI et al., 2023)



Fonte: (LI et al., 2023)

Os resultados experimentais desse estudo validaram a eficácia do método de análise de RUL para a manutenção preditiva. No entanto, o estudo também reconhece as limitações do modelo, como a necessidade de geração manual de parâmetros e a capacidade restrita de considerar múltiplos parâmetros simultaneamente para prever a resposta.

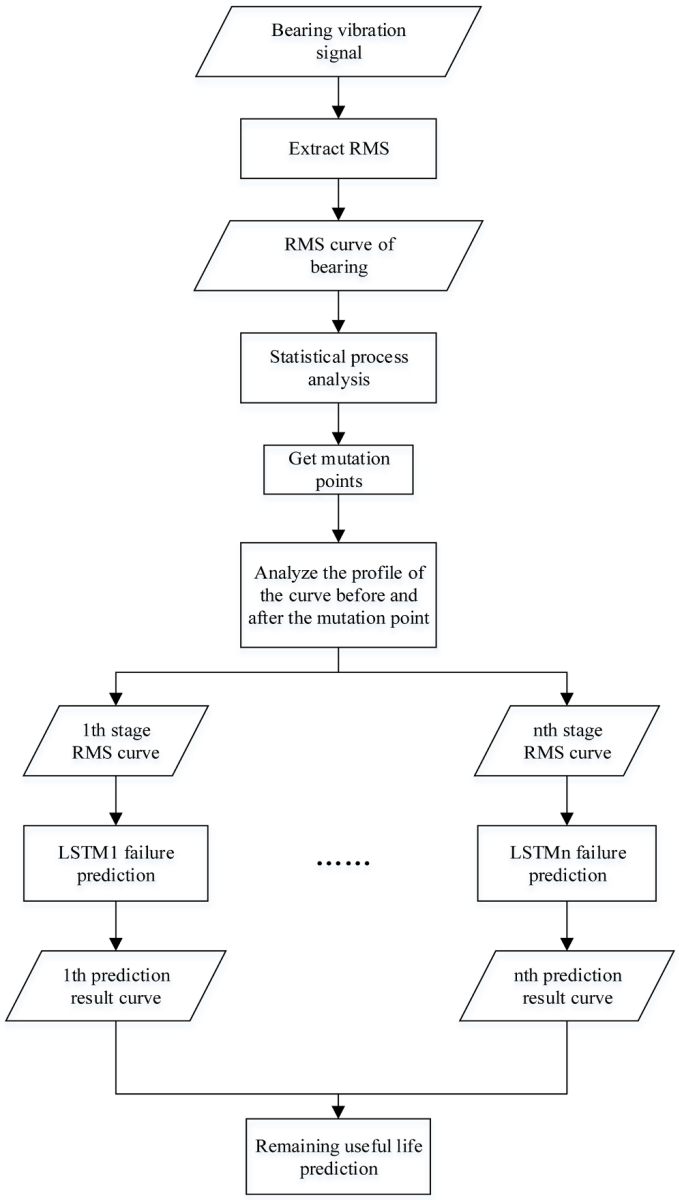
3.3 FORECASTING PARA DETECÇÃO DE DEFEITOS

O último estudo dessa revisão foi feito por Junqiang Liu, et al, em 2021, e traz grandes contribuições para o campo do aprendizado de máquinas na previsão de defeitos em rolamentos de motores de aviação. Esse tipo de defeito é responsável por mais de 80% dos problemas mecânicos dessa área, e por isso há muita demanda por novos métodos de monitoramento. Esse estudo propõe métodos de aprendizado profundo baseado na rede neural LSTM (Long Short-Term Memory) combinada com análise estatística de processo para fazer as previsões. Essa nova abordagem, chamada de LSS (LSTM e análise estatística de processo), foi testada em um conjunto de dados de rolamentos fornecido pela NASA e pelo Instituto FEMTO-ST. Ela demonstrou capacidade de reduzir a incerteza em melhorar a precisão da previsão de defeitos nos rolamentos de motores de aviação, o que pode melhorar a confiabilidade e a segurança desses sistemas. (LIU et al., 2021)

A Figura 15 apresenta de forma detalhada o passo a passo do algoritmo proposto, desde a entrada dos sinais até a previsão de RUL como saída.

Esse modelo se mostrou uma solução inovadora para prever defeitos de rolamentos em motores de aviação. O artigo apresentou experimentos que demonstram que o modelo melhora a precisão de previsão do RUL, em comparação com outros modelos. O artigo destaca também que futuros estudos poderão propor um modelo de diagnóstico de falhas interativo, para melhorar a confiabilidade dos rolamentos e do sistema de motores de aviação, garantindo ainda mais a segurança de voo.

FIGURA 15 – Fluxograma do algoritmo proposto por (LIU et al., 2021)



Fonte: (LIU et al., 2021)

3.4 CONSIDERAÇÕES DA REVISÃO DA LITERATURA

O modelo de aprendizado de máquina, que será apresentado nos próximos capítulos dessa dissertação, apresentará diversas vantagens que o tornam altamente eficaz na análise de dados em máquinas rotativas. Ele será capaz de extrair automaticamente os parâmetros necessários para a análise, tornando o modelo mais robusto e menos dependente de intervenções humanas. Ele será composto por um modelo de classificação e regressão acompanhados por uma rede recorrente, permitindo uma melhor abstração dos dados e a transferência de aprendizado para outras máquinas. Isso permite a identificação de padrões de comportamento, aumentando significativamente a precisão na previsão de defeitos em máquinas rotativas. Com essa abordagem, a ma-

nutenção preventiva pode ser otimizada, minimizando o tempo de parada e reduzindo os custos operacionais.

4 MATERIAIS E MÉTODOS

A primeira seção deste capítulo traz uma lista dos materiais utilizados, detalhando tanto as ferramentas e máquinas quanto os softwares. A segunda seção apresentará a metodologia aplicada, com o intuito de permitir a replicação desse experimento por terceiros e, assim, poder comparar com outros trabalhos semelhante a este.

4.1 MATERIAIS UTILIZADOS

Os materiais utilizados para a coleta de dados podem ser classificados em 5 categorias, são elas:

- Máquina;
- Sensores;
- Medidor de sinais;
- Desbalanceadores;
- Softwares.

O primeiro item é uma máquina rotativa de bancada DAC VAD # 203. Nela é possível forçar falhas de operação e medir seu comportamento nesses estados. As especificações da máquina podem ser vistas na Tabela 3.

TABELA 3 – ESPECIFICAÇÕES DA MÁQUINA ROTATIVA DAC VAD #203

PARÂMETROS	VALOR	UNIDADE
Modelo	AP7402	-
Potência do Motor	0.25	HP
Corrente	2.5	A
Máximo RPM	3450	rpm

FONTE: O AUTOR

O segundo e terceiro itens são os acelerômetros e os conversores, que transformam esses dados analógicos em digitais e permitem que eles sejam interpretados por um computador. O acelerômetro utilizado é triaxial, da PCB Piezotrônics. Alguns detalhes deles estão mostrados nas tabelas 4 e 5.

TABELA 4 – ESPECIFICAÇÕES DO ACELERÔMETRO

PARÂMETROS	VALOR	UNIDADE
Fabricante	LarsonDavis	-
Modelo	SEN021F	-
Sensibilidade de tensão	10	mV/g
Alcance de medição	500	$\pm$g·pk
Alcance de frequência	0.5-2500 (Y, Z) 0,5-3000 (X)	Hz
Frequência de ressonância de montagem	> 25	kHz
Amplitude linear	0,0005	g·rms
Amplitude da sensibilidade	< $\pm$1	%

FONTE: O AUTOR

TABELA 5 – ESPECIFICAÇÕES DO ANALISADOR DE SINAIS

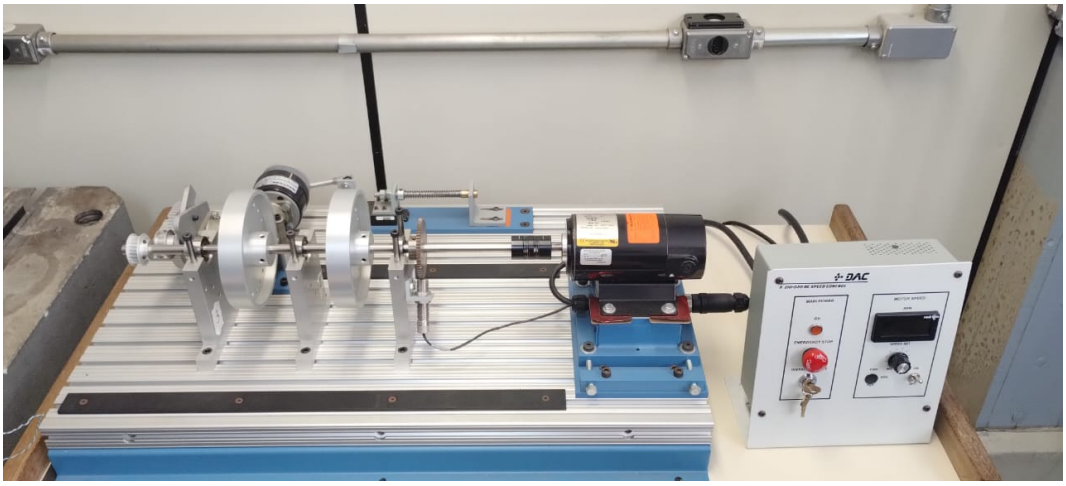
PARÂMETROS	VALOR	UNIDADE
Modelo	NI-4331	
Saídas analógicas 24-bit	1	unid.
Entradas analógicas 24-bit	4	unids.
Frequência de Aquisição	1000	Hz
Tensão de saída	±3.5	V
Tensão de entrada	±10	V
Alcance dinâmico	100	dB

FONTE: O AUTOR

Para forçar a falha de desbalanceamento são utilizados pequenos pesos de 6,80g os quais, ao serem adicionados aos discos, deslocam o cento de massa do sistema. Cada disco possui 22 espaços reservados para a adição desses desbalanceadores. Nos experimentos propostos, foram distribuídos uma quantidade de 1 a 3 pesos, para que essa variação ocorra da forma mais gradual possível ao longo do experimento. Os detalhes da montagem da máquina podem ser vistos na Figura 16.

A falha de desalinhamento, por sua vez, é feita a partir do deslocamento da base onde o motor está apoiado. Isso é feito rotacionando parafusos que puxam e empurram essa base. A distância deslocada é medida em frações de volta de cada parafuso e convertidos para milímetros. Como o passo do parafuso é conhecido e igual a 1,5mm, foi decidido que o experimento de desalinhamento será feito com 1/3 de volta, representando 0,5mm. Os detalhes da base de apoio do motor e dos parafusos fixadores estão apresentados nas Figuras 17 e 18.

FIGURA 16 – Montagem do experimento

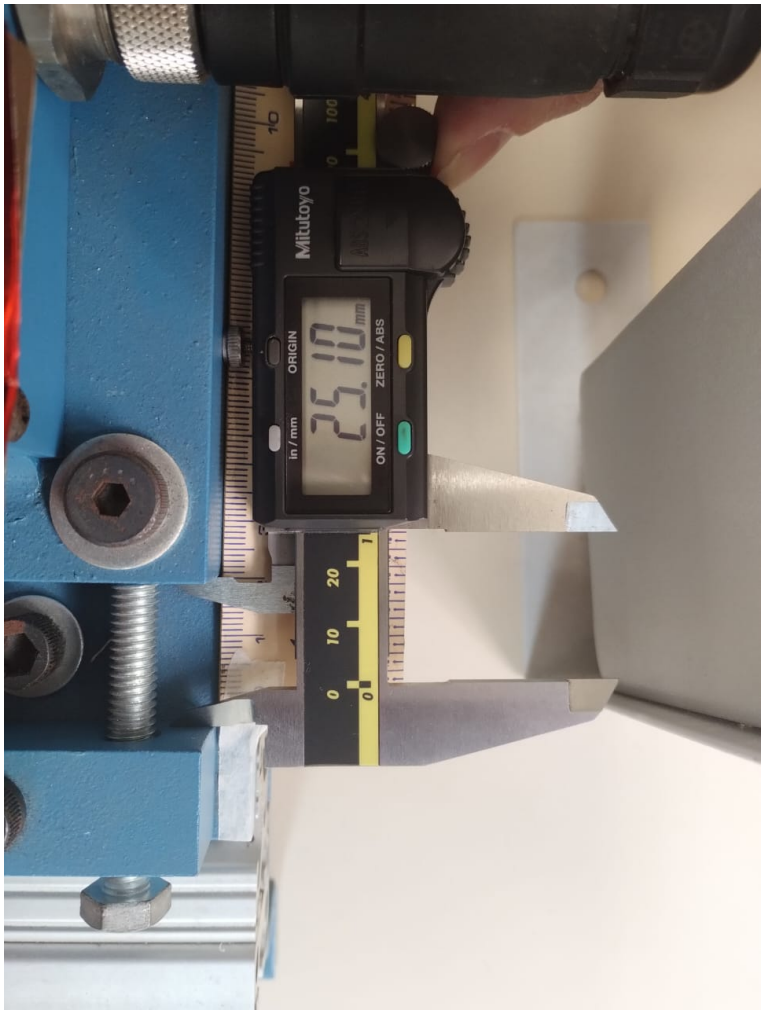


Fonte: O autor

FIGURA 17 – Detalhes da base do motor e parafusos onde será forçada a falha de desalinhamento

Fonte: O autor

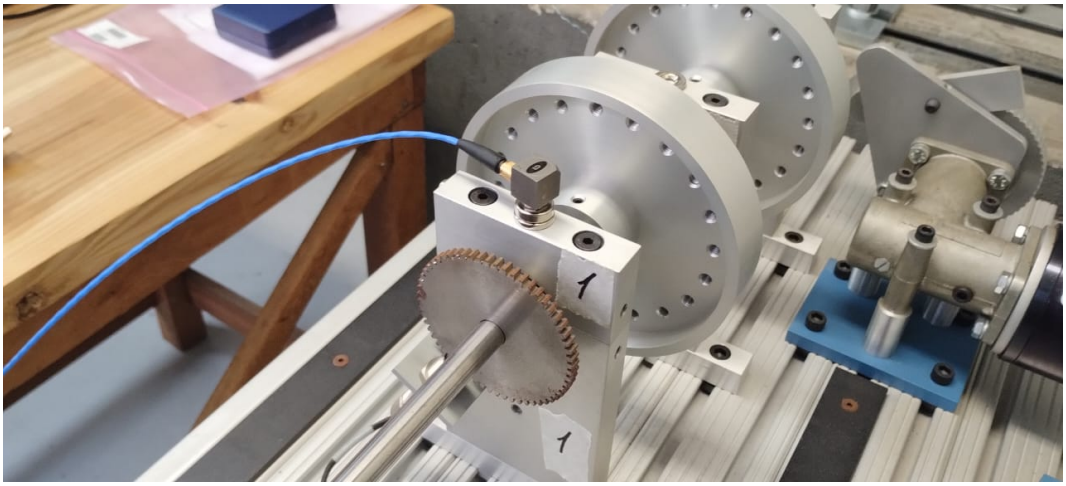
FIGURA 18 – Detalhe da base do motor onde será forçada a falha de desalinhamento, com ênfase na medição do seu ponto zero



Fonte: O autor

As imagens 19 e 20 mostram a montagem do sistema, com ênfase no sensor e conversor utilizado.

FIGURA 19 – Detalhes do sensor



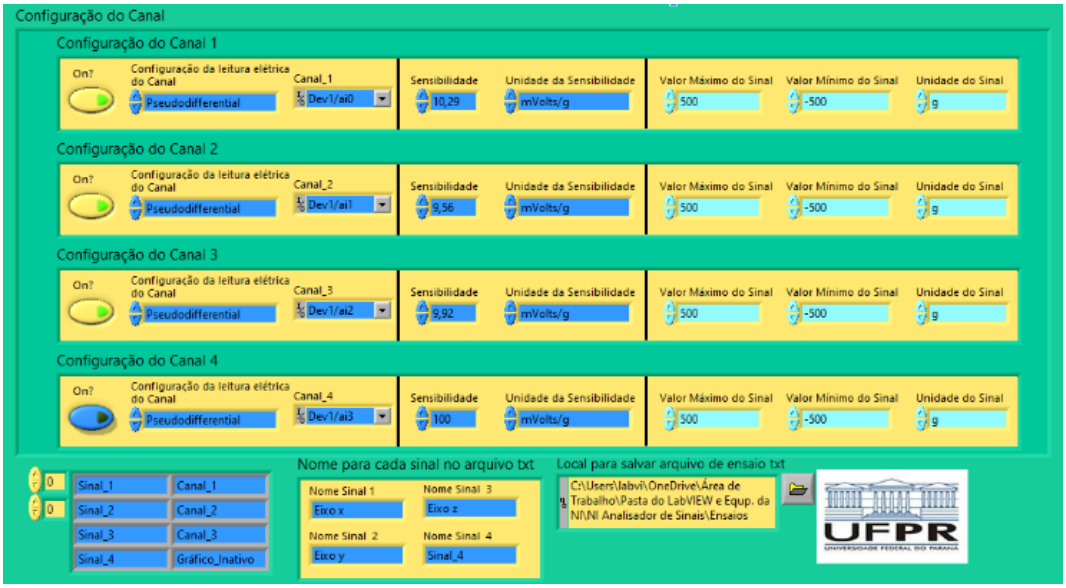
Fonte: O autor

FIGURA 20 – Detalhes do conversor analógico-digital

Fonte: O autor

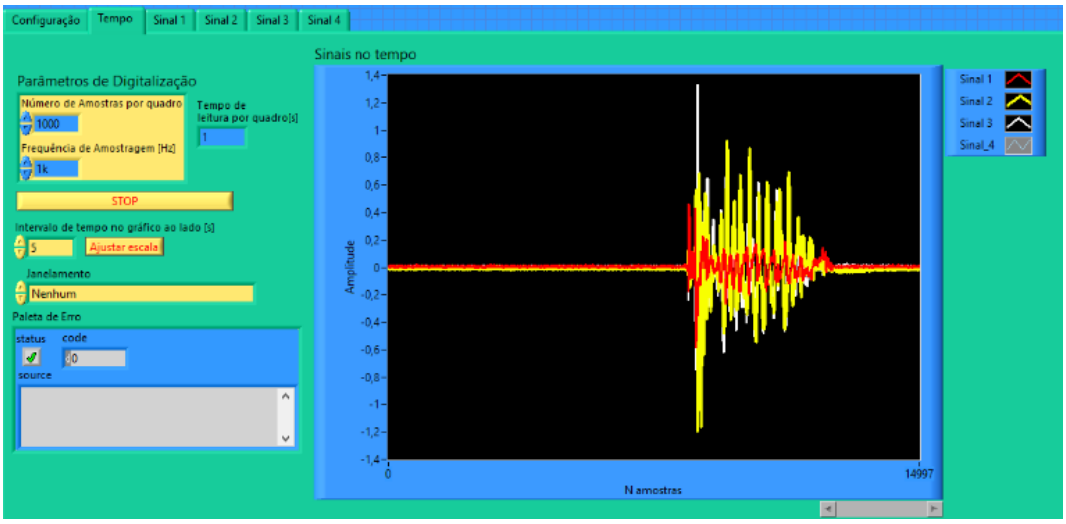
Para a aquisição dos dados, foi utilizado o *LabView* e o Software desenvolvido por Maurizio Radloff Barghouthi. As Figuras 21 e 22 mostram como o software realiza essa operação e exemplifica alguns dos dados coletados.

FIGURA 21 – Tela de configurações do software em LabView



Fonte: O autor

FIGURA 22 – Tela de visualização dos sinais do software em LabView



Fonte: O autor

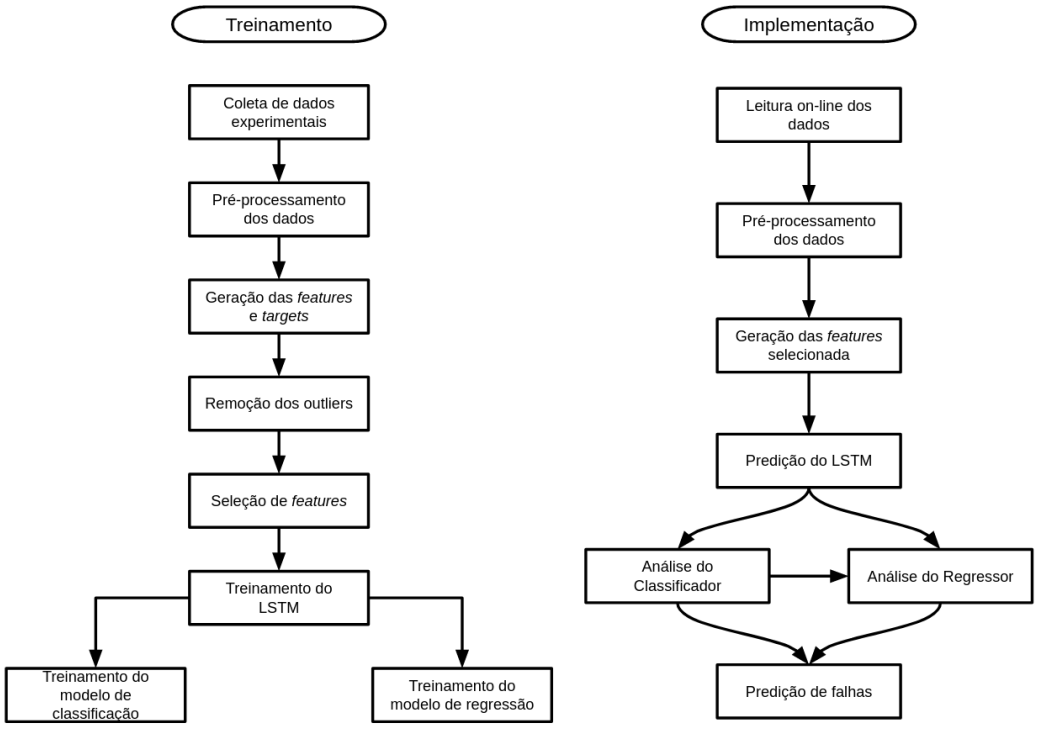
4.2 METODOLOGIA

A metodologia encontra-se dividida em duas partes: a de treinamento e a de implementação. Durante o treinamento dos modelos, após a coleta de dados da máquina rotativa, ocorrem as seguintes etapas: dizimação dos dados, geração de dados sintéticos, pré-processamento dos dados originais e sintéticos, treinamento do modelo de previsão de dados futuros com LSTM, criação e seleção dos parâmetros e o treinamento dos modelos restantes, sendo um de classificação e dois de regressão.

O modelo de previsão tem como objetivo antecipar o comportamento futuro da máquina com base em seus dados históricos. Já o modelo de classificação visa identificar o tipo de falha projetada no futuro, independentemente de sua intensidade. Por fim, o modelo de regressão utiliza o tipo de falha previsto como um parâmetro para, finalmente, prever sua intensidade.

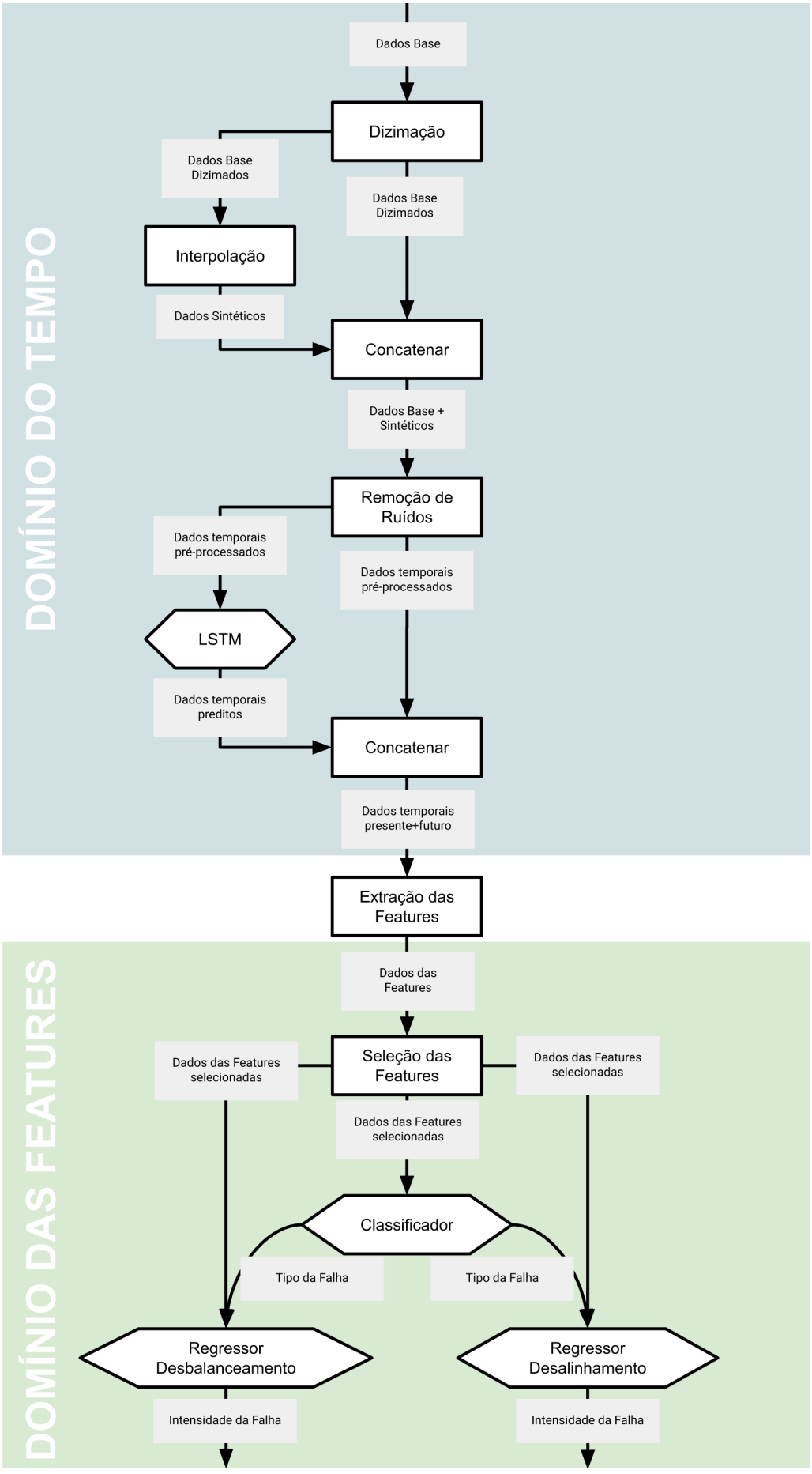
Para a etapa de implementação, serão coletadas amostras dos dados de aceleração de uma máquina em tempo real a partir de sensores acelerômetros, onde os parâmetros selecionados serão extraídos e passados para a rede LSTM treinada, a fim de prever o comportamento da máquina no futuro. Essa previsão é, então, utilizada pelo modelo de classificação para identificar a presença de falhas e pelo modelo de regressão para prever a sua intensidade. Ou seja, a saída do modelo de LSTM será usada como entrada para o modelo de classificação, e a saída será usada como entrada do modelo de regressão. A visualização de ambos esses processos está dada na Figura 24 e seus detalhes estão nas seções a seguir.

FIGURA 23 – Fluxograma de treinamento (à esquerda) e de implementação (à direita)



Fonte: O autor

FIGURA 24 – Fluxograma do processo de treinamento



Fonte: O autor

4.2.1 Aquisição dos dados

O experimento consiste de 891 amostras iniciais do funcionamento de uma máquina rotativa para diferentes falhas e intensidades, incluindo o modo normal, com ausência de falha. Os parâmetros de cada amostra foram coletados a partir de um acelerômetro triaxial posicionado em 3 posições diferentes, em cima de cada mancal disponível na máquina rotativa de testes.

A tabela 6 traz as variáveis do sistema que foram alteradas em cada experimento e os valores assumidos para cada uma delas.

TABELA 6 – DEFINIÇÃO DAS *VARIÁVEIS DO EXPERIMENTO*

Variável	Valores	Número de variações
Posição do sensor	1, 2 ou 3	3
RPM	300, 600, ..., 3300	11
Variação de rotação	Subida, constante, descida	3
Falhas	Normal, 6,80g, 13,60g, 20,4g, 0,5mm	9

FONTE: O Autor

Assim, tem-se um total de $3 * 11 * 3 * 9 = 891$ experimentos diferentes. O motivo do experimento sem falhas ser repetido mais vezes é para ter um banco de dados mais balanceado no final do processo. Para cada um dos experimentos, os dados que serão coletados são:

- Tempo;
- Rotações por minuto (RPM);
- Aceleração no eixo X (série no domínio do tempo);
- Aceleração no eixo Y (série no domínio do tempo);
- Aceleração no eixo Z (série no domínio do tempo).

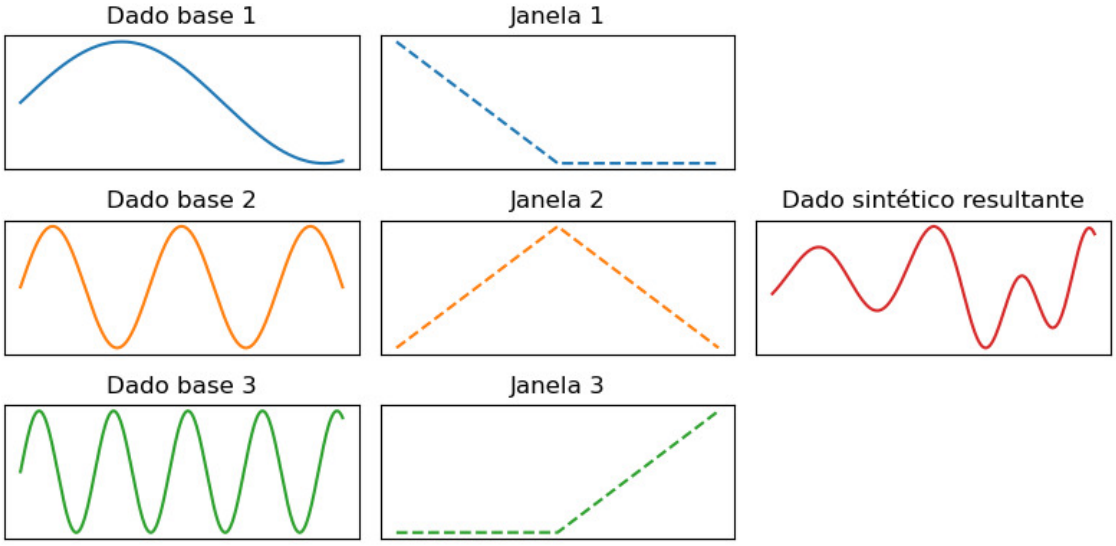
4.2.2 Adição de dados sintéticos

Cada experimento foi realizado com a falha e a intensidade constante ao longo do tempo, porém, para que a rede LSTM tenha uma maior eficiência na sua previsão, é necessário que esses valores mudem gradualmente com o tempo, haja vista ser esse o comportamento real esperado com tempo de uso da máquina rotativa. Além disso, a obtenção de padrões da máquina para saídas que mudam gradualmente também é essencial para a previsão da intensidade contínua do defeito, sem que caiamos em classificação apenas. Para se obter esses dados, foi necessário gerá-los de forma sintética com *data augmentation*, mesclando os sinais normais com sinais de diferentes

tipos de falhas, assim simulando como seria o comportamento da máquina para o crescimento gradual da intensidade da falha.

Esses dados sintéticos são gerados multiplicando os dados dos experimentos base por uma janela linear que é dependente do número de dados que estão sendo interpolados. A seguir, soma-se os sinais janelados de todos no final, resultando em um novo sinal original com uma variação linear do estado da máquina. Esse processo pode ser visualizado na Figura 25, que mostra um exemplo utilizando dados base senoidais de uma frequência. A coluna da esquerda representa os dados originais no tempo, a coluna do meio representa as janelas de cada dado respectivo e a coluna da direita mostra o resultado da soma dos produtos entre os dados bases e suas janeladas.

FIGURA 25 – Visualização dos dados utilizados durante a geração dos dados sintéticos



Fonte: O autor

Esse processo pode ser expandido para 4 ou mais sinais, onde uma falha passa por um crescimento cada vez mais gradual, otimizando assim a previsão de falhas da rede LSTM.

4.2.3 Geração dos parâmetros

Depois de coletados os dados de todos os experimentos usando o sistema de máquina-sensor-conversor, os dados no tempo foram então convertidos em parâmetros.

Alguns dados tiveram seus valores convertidos diretamente, enquanto outros foram transformados seguindo alguma função pré-definida. A tabela 7 mostra as funções pré-definidas escolhidas, onde x_i representa cada o i-ésimo ponto no domínio do tempo, X_i o seu equivalente do domínio da frequência e N o número total de pontos.

TABELA 7 – DEFINIÇÃO DOS PARÂMETROS

Parâmetro	Símbolo	Definição
Média	μ_x	$\mu_x = \frac{1}{N}\sum_{i=0}^{N} x_i$
Desvio Padrão	σ_x	$\sigma_x^2 = \frac{1}{N}\sum_{i=0}^{N}(x_i - \mu_x)^2$
Curtose	κ_x	$\kappa_x = \frac{1}{N}\sum_{i=0}^{N}\left(\frac{x_i-\mu_x}{\sigma_x}\right)^4$
Distorção	γ_x	$\gamma_x = \frac{1}{N}\sum_{i=0}^{N}\left(\frac{x_i-\mu_x}{\sigma_x}\right)^3$
Amplitude Pico a Pico	x_{ppv}	$x_{ppv} = max(x_i) - min(x_i)$
Valor Quadrático Médio	x_{rms}	$x_{rms} = \left(\frac{1}{N}\sum_{i=0}^{N} x_i^2\right)^{1/2}$
Raiz Quadrada da Amplitude	x_{sra}	$x_{sra} = \left(\frac{1}{N}\sum_{i=0}^{N}\sqrt{\lvert x_i\rvert}\right)^2$
Fator de Crista	x_{cf}	$x_{cf} = \frac{max(\lvert x_i\rvert)}{x_{rms}}$
Fator de Impulso	x_{if}	$x_{if} = \frac{max(\lvert x_i\rvert)}{\frac{1}{N}\sum_{i=0}^{N}\lvert x_i\rvert}$
Fator de Margem	x_{mf}	$x_{mf} = \frac{max(\lvert x_i\rvert)}{x_{sra}}$
Fator de Curtose	x_{kf}	$x_{kf} = \frac{\kappa_x}{x_{rms}^4}$
Média (FFT)	μ_x	$\mu_X = \frac{1}{N}\sum_{i=0}^{N} X_i$
Desvio Padrão (FFT)	σ_x	$\sigma_X^2 = \frac{1}{N}\sum_{i=0}^{N}(X_i - \mu_X)^2$
Valor Quadrático Médio (FFT)	x_{rms}	$x_{rms} = \left(\frac{1}{N}\sum_{i=0}^{N} X_i^2\right)^{1/2}$
Valor de Pico (FFT)	X_{max}	$X_{max} = max(X)$
Frequência do Pico (FFT)	f_{pico}	$f_{pico} = f(X_{max})$

FONTE: (MOREIRA; RIBEIRO, 2018)

Cada parâmetro foi escolhido por ser capaz de agregar características física do sinal como um todo em um único número, como a sua média, amplitude, energia, entre outras. Cada função representa o comportamento do sistema em uma dimensão diferente, e cada uma delas influencia o parâmetro de uma maneira diferente. Ao utilizá-las como os parâmetros, se está encontrando formas aplicáveis de ajustar uma máquina para podermos atingir a resposta desejada. Com todas elas definidas, o conjunto de dados final é feito transformando-se cada amostra em linhas e cada parâmetro em colunas.

4.2.4 Pré-processamento dos dados

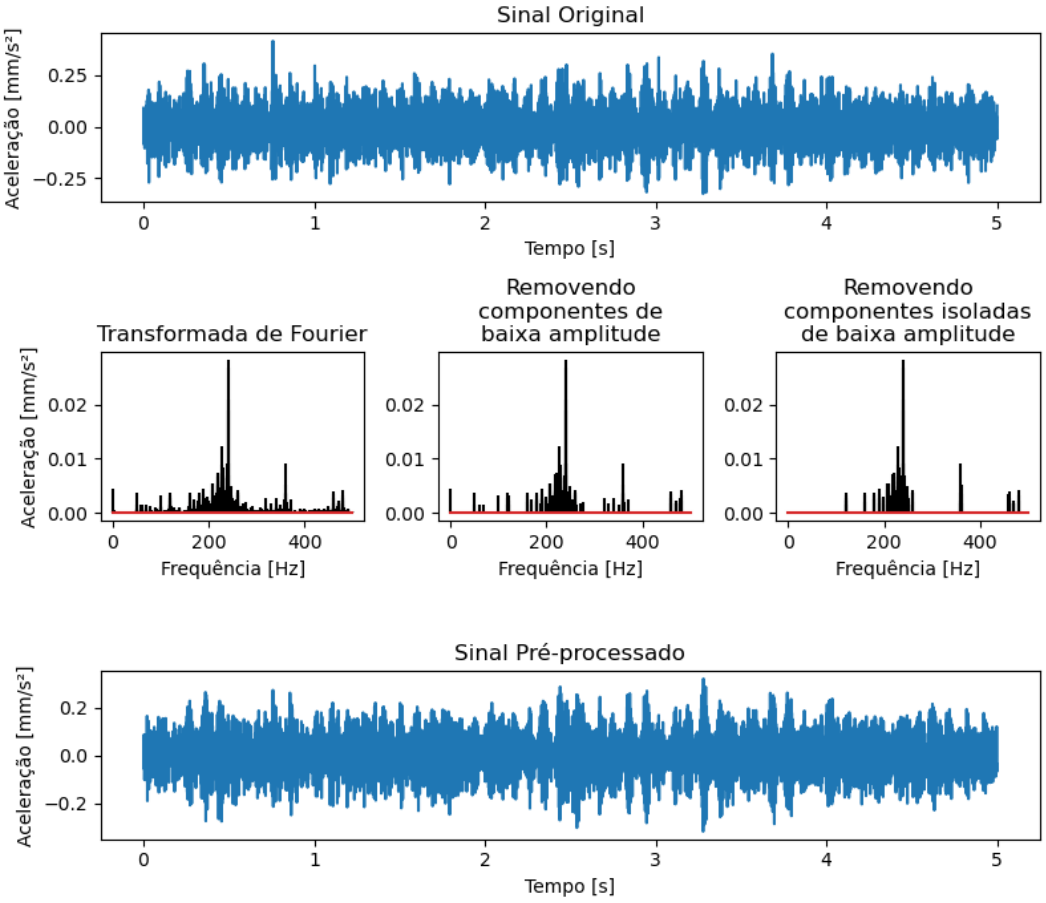
Antes de executar o algoritmo de aprendizado de máquina é necessário fazer um pré-processamento nos dados para melhorar a sua eficiência. Isso serve para reduzir ruídos e enfatizar os conjuntos de dados importantes a serem utilizados para o treinamento da máquina. Os passos utilizados no pré-processamento são:

1. Dizimação dos dados: Se a frequência de amostragem dos dados é muito alta, a diferença entre um dado e outro é muito baixa, o que pode resultar em uma demora no aprendizado. Como a velocidade máxima de rotação aplicada foi de

3300 rpm (55 Hz) e que a frequência de aquisição dos dados utilizada foi de 1KHz, foi escolhida uma dizimação de fator 9, aumentando 9x o período entre dados coletados.

2. Transformada de Fourier: Os dados no domínio do tempo restantes são convertidos para o domínio da frequência;
3. Filtro passa-baixa: Todo sinal acima de 500Hz na frequência foi considerado como nulo;
4. Remoção de baixas amplitudes: Todo sinal com amplitude menor que 5% da frequência de pico foi considerado ruído e zerado;
5. Remoção de baixas amplitudes isoladas: Todo sinal com amplitude menor que 10% da frequência de pico e com frequências vizinhas nulas foram consideradas ruído e zeradas;
6. Transformada Inversa de Fourier: Os dados no domínio da frequência são convertidos de volta para o domínio do tempo.

FIGURA 26 – Visualização do passo a passo para a remoção de ruídos



Fonte: O autor

Com a remoção dos ruídos dos dados para todos os tipos de falhas, outros dois tipos de limpeza nos dados ainda são necessários: A remoção de experimentos indesejados e a de parâmetros indesejados, a quais correspondem às linhas e às colunas do banco de dados, respectivamente.

Os experimentos indesejados são aqueles que possuem valores muito fora do padrão, como os *outliers*. Eles foram identificados a partir de um método de remoção de *outliers*, que consiste em encontrar todos os registros cuja diferença até a média seja pelo menos 3 vezes maior que a diferença do 0.99 percentil do *dataset* até essa mesma média. Esse processo é repetido de forma iterativa até que nenhum dado seja considerado um *outlier*. Por isso o número de pontos que podem ser excluídos por esse método não pode ser previsto facilmente até sua execução.

4.2.5 Seleção de parâmetros

Para a seleção dos parâmetros, foram realizados 2 filtros diferentes para selecionar os mais convenientes. O primeiro deles é utilizando o Índice de Fisher e excluindo os parâmetros que estiverem abaixo de um limiar pré-determinado. Para esse trabalho, o limiar escolhido foi de 1,5. Esse índice avalia o quão bem separável um parâmetro pode ser considerado, ou seja, se o parâmetro possui dados com baixa variação dentro de cada classe e dados com muita variação entre as classes. Esse índice avalia se os parâmetros possuem valores semelhantes para *targets* parecidos e valores distintos para *targets* diferentes. O índice de Fisher é calculado a partir da equação 4.1

$$f(X_i) = \frac{\sum_{j=1}^{C} N_j(\mu_{ij} - \mu_i)^2}{\sum_{j=1}^{C} N_j \sigma_{ij}^2} \tag{4.1}$$

O segundo filtro remove os parâmetros que possuem uma alta correlação com outros parâmetros. Nota-se que há vários parâmetros com um alto índice de correlação entre si, e isso deve ser evitado na hora de executar o aprendizado de máquina. Isso é resolvido executando um algoritmo que irá identificar todas as correlações dentro de um limiar (positivo e negativo) e excluir um dos dois parâmetros da correlação. Assim, restará apenas os parâmetros que possuem valores baixos de correlação entre si.

Importante lembrar que haverá 4 modelos ao final, um de previsão, um de classificação e dois de regressão, sendo um para o desbalanceamento e outro para o desalinhamento. Essa etapa de pré-processamento será realizada uma vez para cada um dos modelos de forma independente, o que significa de cada classificador terá parâmetros de entrada diferentes e, consequentemente, bancos de dados de treinamento diferentes e otimizados para cada aplicação, ao final do processo. Para o

treinamento, em todos os modelos, foram separados 20% do dataset pertinente para teste, enquanto 80% foram usados para treino.

4.2.6 Modelo LSTM

A rede LSTM é a primeira camada da rede neural final. Ela tem como objetivo interpretar os dados de uma série temporal e prever quais serão os valores futuros do comportamento da máquina em função dos seus dados históricos, sua posição na série e os seus valores anteriores. Isso será feito para cada um dos parâmetros em paralelo e a saída de cada uma delas será a entrada da rede neural convencional, assim servindo para o propósito de identificar a falha o mais rápido possível, devido à sua aplicação no tempo.

4.2.7 Modelo de Classificação

O melhor modelo será aquele com a melhor assertividade, ou seja, aquele que teve o maior número de verdadeiros positivos e verdadeiros negativos, além de serem avaliadas as acurácias e curva ROC-AUC. Os modelos a serem testados são:

- KNN;
- Floresta aleatória;
- SVC;
- Rede neural densa e direta.

4.2.8 Modelo de Regressão

A seleção do melhor método será feita a partir do valor da Média de Erro Absoluto (MAE). O MAE é uma métrica de medição do desempenho de modelos de regressão que calcula a média das diferenças absolutas entre as previsões e os valores reais e pode ser utilizada para medir a precisão do modelo na previsão de valores numéricos. Para se decidir qual o método mais eficiente de regressão, comparou-se os valores de MAE de cada um deles, e aquele com o menor valor será o modelo escolhido. Para os métodos de regressão, devem ser testados os seguintes modelos:

- KNN;
- Floresta aleatória;
- SVC;
- Rede Neural densa e direta.

4.2.9 Modelo composto

O modelo composto recebe os dados de vibração da máquina como entrada e retorna o tipo de falha e a intensidade da falha como saída. Dentro do modelo, para cada um dos parâmetros, uma rede LSTM irá converter cada série temporal em um único valor numérico e esses valores serão usados como dados de entrada de um modelo de classificador, o qual irá prever o tipo de falha que está ocorrendo na máquina. Em seguida, os dados de saída do LSTM e do classificador são usados como entrada em um modelo de regressão, que retorna a intensidade da falha para aquele dado tipo de falhas e para aquele dado comportamento de máquina. Ambas as saídas do classificador e regressor são a saída do modelo composto.

4.2.10 Exportação dos modelos treinados

Após cada modelo ser otimizado e o melhor de cada etapa ser escolhido, ele é salvo para que seja realizada a etapa de transferência de aprendizado. Nela, todos os modelos escolhidos são unidos e considerados um único sistema, e esse será o modelo final.

As etapas intermediárias entre um modelo e outro serão camadas intermediárias e apenas os resultados finais serão retornados pelo modelo final, simplificando os processos de implementação e de utilização para o usuário final.

5 RESULTADOS

Neste capítulo serão apresentados os resultados preliminares e finais utilizando a metodologia proposta.

5.1 RESULTADOS PRELIMINARES

De forma a testar a metodologia inicial de pré-processamento dos dados e a separabilidade das classes de defeito, antes de ser possível a obtenção do banco de dados original proposto neste trabalho, foi utilizado o banco de dados nomeado Mafaulda, desenvolvido pela UFRJ (MOREIRA; RIBEIRO, 2018). Ele tem o objetivo de estudar o comportamento de uma máquina rotativa para 10 conjuntos de dados de falhas diferentes, incluindo a ausência de falhas, em diferentes intensidades. São elas:

- Normal;
- Desalinhamento horizontal;
- Desalinhamento vertical;
- Desbalanceamento;
- Rolamento inferior - Falha na gaiola;
- Rolamento inferior - Falha no anel externo;
- Rolamento inferior - Falha na esfera;
- Rolamento superior - Falha na gaiola;
- Rolamento superior - Falha no anel externo;
- Rolamento superior - Falha na esfera.

O Mafaulda é um banco de dados composto por 1951 arquivos, em que cada um representa os dados de saída dos sensores de um único experimento. Cada arquivo é formado por 9 colunas e 250 mil linhas. Cada linha é uma amostra coletada na frequência de 50kHz durante 5 segundos, enquanto as colunas representam o vetor do tempo e mais 8 sensores de falhas em uma máquina rotativa feitos pela UFRJ. As informações coletadas em cada experimento são:

- Tacômetro;

- Acelerômetro de rolamento Axial suspenso;
- Acelerômetro de rolamento Tangencial suspenso;
- Acelerômetro de rolamento Radial suspenso;
- Acelerômetro de rolamento Axial de balanço;
- Acelerômetro de rolamento Tangencial de balanço;
- Acelerômetro de rolamento Axial de balanço;
- Microfone.

No caso, o primeiro e o último são descartados e apenas os acelerômetros serão considerados, totalizando 6 colunas.

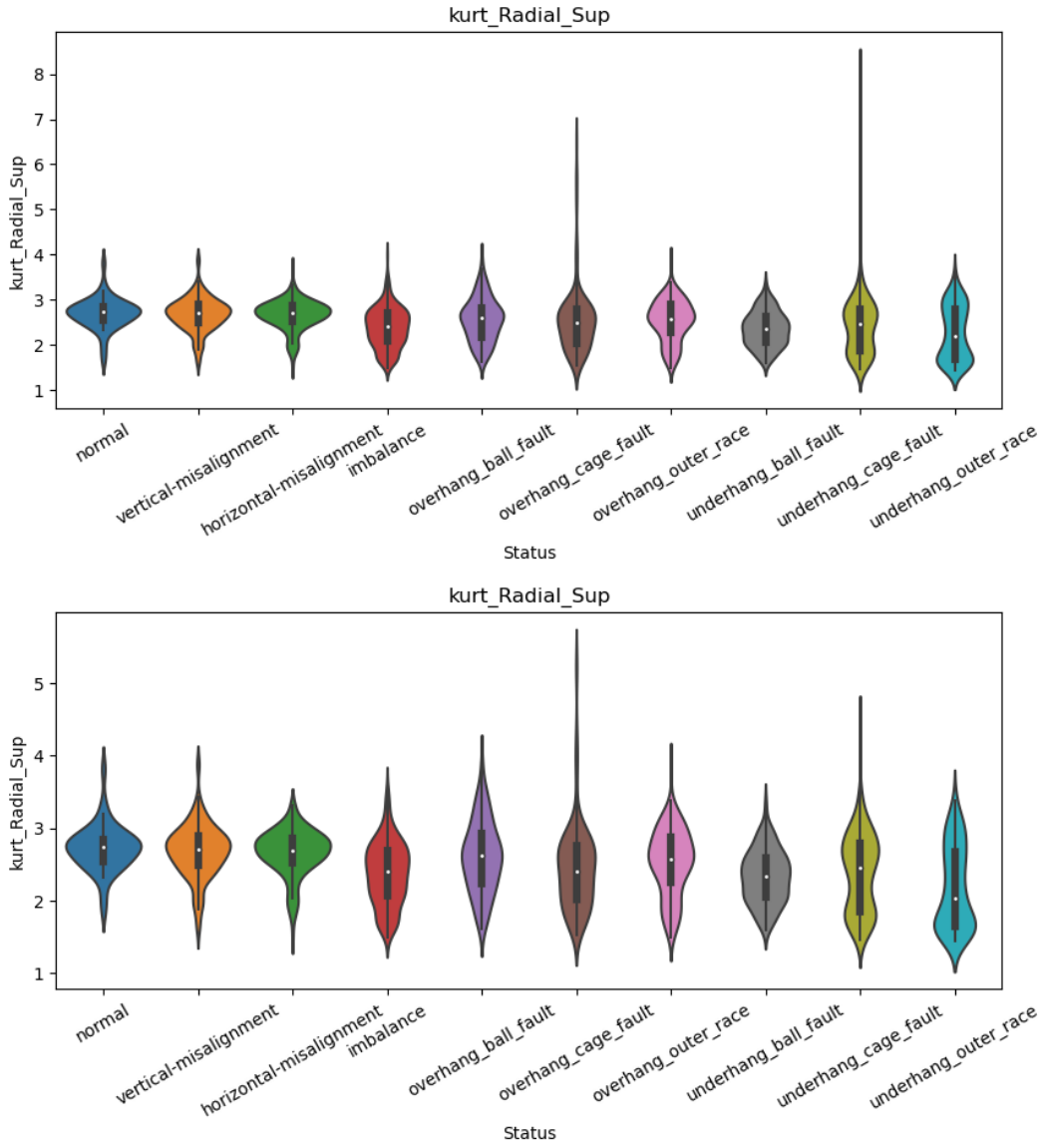
5.1.1 Teste da metodologia de pré-processamento

Para melhorar a qualidade dos dados e melhorar a eficiência do aprendizado de máquinas, foi realizado o pré-processamento dos dados. Em seguida foram extraídos 16 parâmetros numéricos (11 no domínio do tempo e 5 no domínio da frequência) para cada um dos 6 dados de acelerações, além de 1 *target* de categorização.

Como cada um desses parâmetros são aplicados em cada um dos 6 dados de acelerômetros, o total é de $16 * 6 = 96$ parâmetros iniciais. Após a extração dos parâmetros foi realizado um segundo pré-processamento, agora nos parâmetros, fazendo a seleção dos melhores parâmetros, a fim de melhorar mais ainda a sua qualidade para o treino. As duas etapas do pré-processamento dos parâmetros são a remoção dos parâmetros correlacionadas e a remoção dos parâmetros pouco informativos.

Para os dados do Mafaulda, durante a remoção dos *outliers*, foram encontrados 139 registros, os quais poderiam ser descartados, assim reduzindo-os de 1951 registros para 1812 (redução de 7,1%). A diferença causada pela remoção dos *outliers* pode ser visualizada na Figura 27. Destaque para a escala Y, que é menor após essa etapa de pós-processamento, mostrando que os pontos não possuem mais valores extremos.

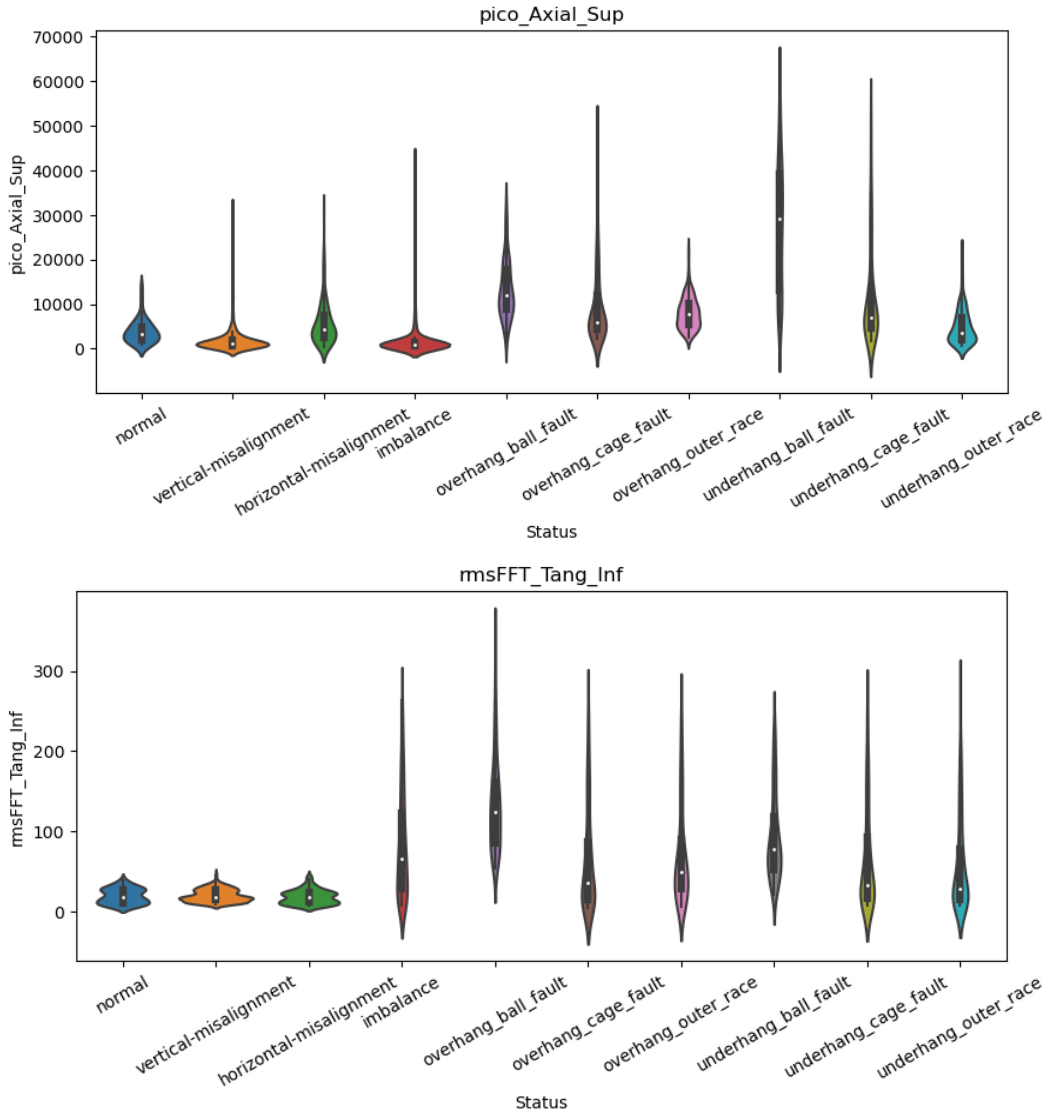
FIGURA 27 – Valor da curtose no acelerômetro radial superior antes (acima) e depois (abaixo) da remoção dos *outliers*



Fonte: O autor

Para a remoção dos parâmetros pouco informativos, foi utilizado um limiar de 1.5, ou seja, para um parâmetro ser considerado informativo, o desvio padrão entre uma classe e outra deve ser no mínimo 1.5 vezes maior do que o desvio padrão dentro de cada classe individualmente. Um exemplo de um parâmetro com muitas informações e um parâmetro com poucas informações pode ser visto na Figura 28. Destaque novamente para a escala Y, que também menor após essa etapa do pós-processamento.

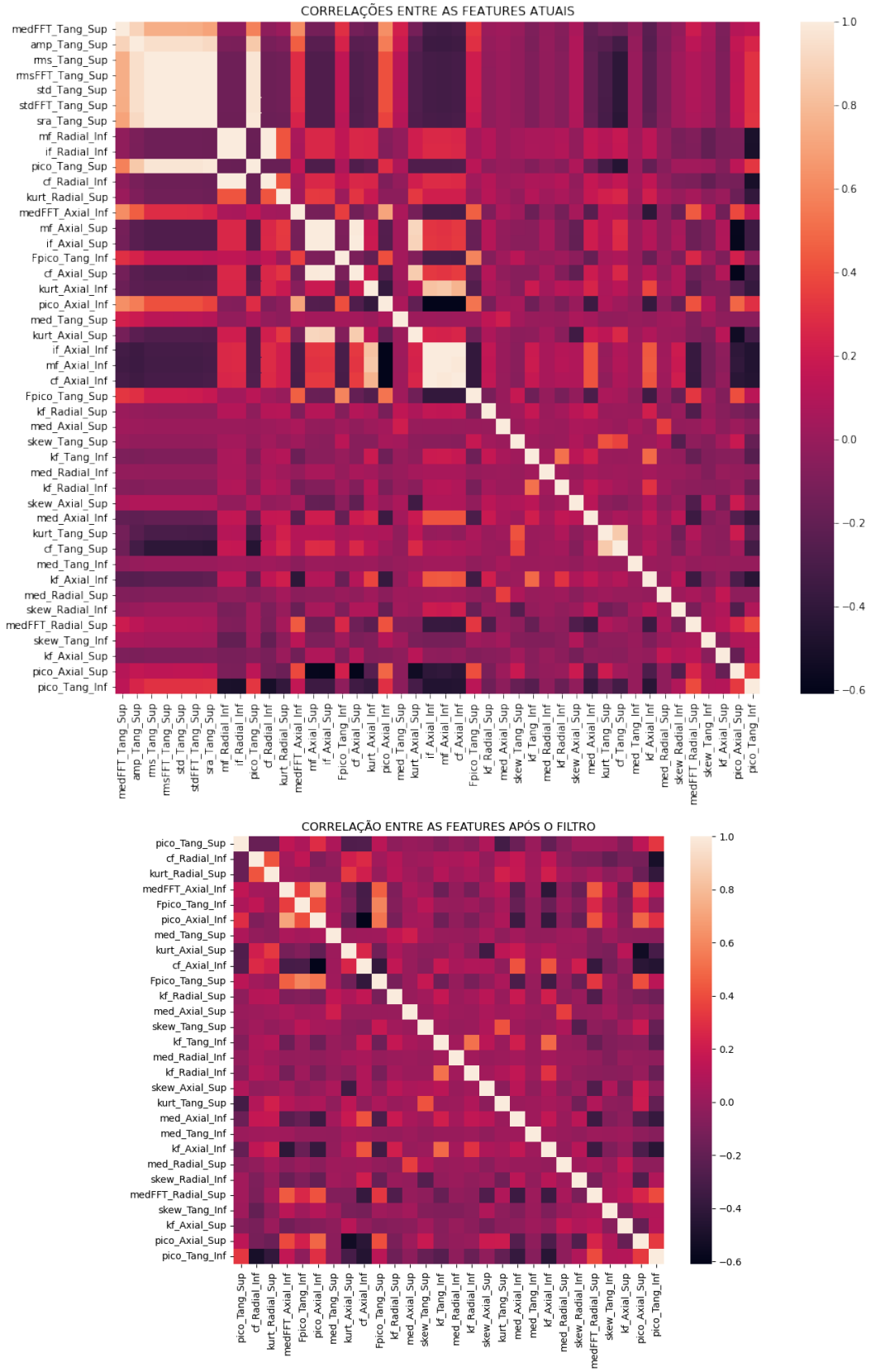
FIGURA 28 – Valor eficaz no domínio da frequência no acelerômetro tangencial inferior antes (acima) e depois (abaixo) da remoção dos *outliers*



Fonte: O autor

Para a remoção dos parâmetros correlacionadas, foi analisada qual a correlação entre os parâmetros que passaram no filtro anterior e, após, foi escolhido um limiar de 0.8, ou seja, qualquer parâmetro que tenha uma correlação acima de 0.8 ou abaixo de -0.8 com outro será descartado. Realizada a remoção, a correlação dos parâmetros restantes é mostrada na figura 29.

FIGURA 29 – Antes e depois da remoção dos parâmetros com alta correlação



Fonte: O autor

5.1.2 Teste do treinamento do classificador

Após finalizado o pré-processamento dos dados, foram treinados diferentes modelos de aprendizado de máquinas com vários métodos diferentes, visando encontrar aquele que fornecesse a maior acurácia nos dados de teste. Para cada modelo testado foi realizado um *Hyper Parameter Tuning* e em seguida obtida sua acurácia.

Para o método da Árvore de Decisão, após o *Hyper Parameter Tuning*, os parâmetros são mostrados na tabela 8.

TABELA 8 – HYPER-PARAMETER TUNING PARA A ÁRVORE DE DECISÃO

Parâmetro	**Valor**
Critério	Entropia
Profundidade máxima	10

FONTE: O AUTOR

A acurácia desse modelo foi de 84,57%

Para o método de floresta aleatória, após o *Hyper Parameter Tuning*, os parâmetros utilizados são mostrados na tabela 9.

TABELA 9 – HYPER-PARAMETER TUNING PARA A FLORESTA ALEATÓRIA

Parâmetro	**Valor**
Bootstrap	Falso
Critério	Entropia
Profundidade máxima	20

FONTE: O AUTOR

A acurácia desse modelo foi de 90.08%

Para o método SVM, após o *Hyper Parameter Tuning*, os parâmetros utilizados são mostrados na tabela 10.

TABELA 10 – HYPER-PARAMETER TUNING PARA O SVM

Parâmetro	**Valor**
C	7
Forma da função de decisão	Um contra um
Grau	2

FONTE: O AUTOR

A acurácia desse modelo foi de 92.56%, sendo a melhor entre os modelos testados.

Para o método KNN, após o *Hyper Parameter Tuning*, os parâmetros utilizados são mostrados na tabela 11.

TABELA 11 – HYPER-PARAMETER TUNING PARA O KNN

Parâmetro	**Valor**
Número de vizinhos	3
Grau da função de distância	1

FONTE: O AUTOR

A acurácia desse modelo foi de 86.77%

Para as redes neurais, após o *Hyper Parameter Tuning*, os parâmetros utilizados são mostrados na tabela 12.

TABELA 12 – HYPER-PARAMETER TUNING PARA A REDE NEURAL

Parâmetro	**Valor**
Taxa de aprendizado	0,025
Paciência	10

FONTE: O AUTOR

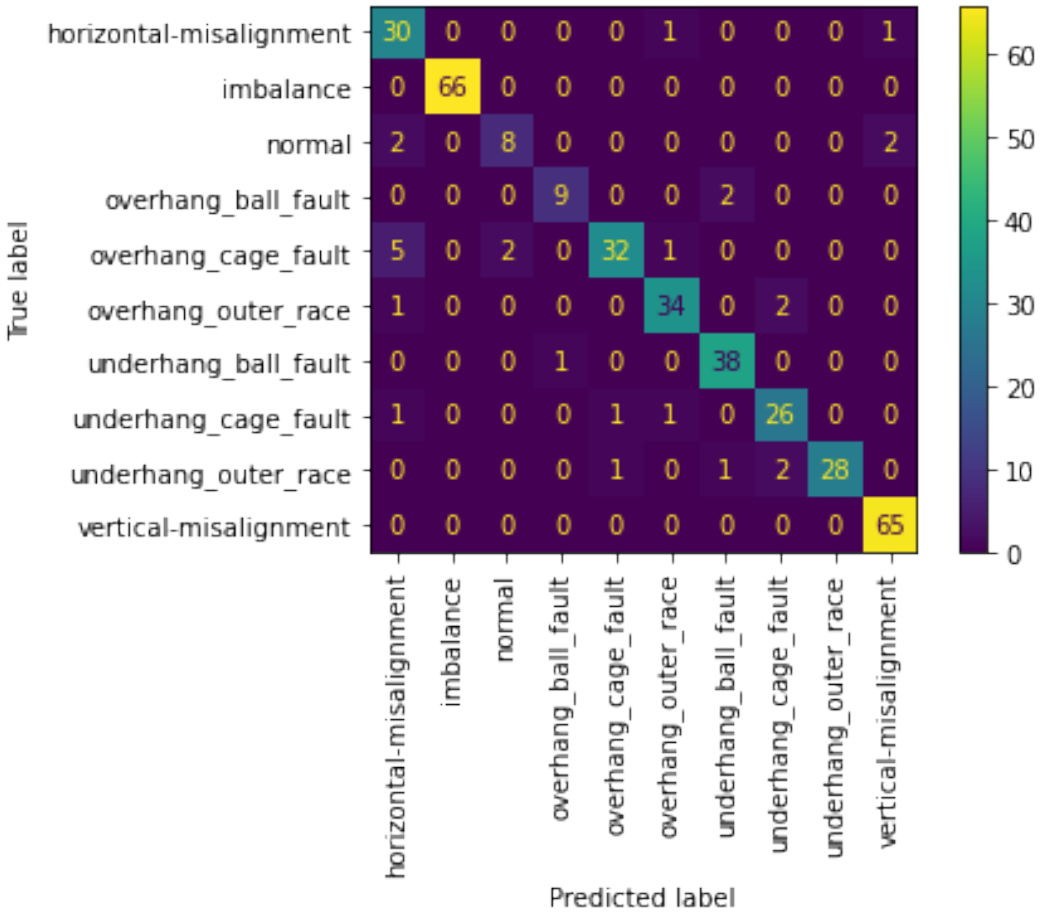
Outros exemplos de hiper parâmetros de uma rede neural são o número de neurônios por camada, a taxa de aprendizado, a razão de *dropout*, a arquitetura da rede e o número de épocas de treinamento. O número de neurônios por camada determina a flexibilidade de uma rede para aprender padrões complexos. A taxa de aprendizado determina o passo dos ajustes de peso durante o treinamento. A razão de *dropout* ajuda a evitar sobreajuste, removendo aleatoriamente alguns neurônios durante o treinamento e forçando a rede a ser capaz de resolver problemas mesmo com informações indisponíveis. A arquitetura da rede se refere ao número de camadas e de conexão entre elas, que pode ser densa, *feedforward*, etc, e afeta a eficiência e a capacidade da rede. O número de épocas de treinamento determina o tempo total de treinamento da rede. O ajuste desses hiper parâmetros tem o objetivo de garantir que a rede tenha um bom desempenho nos dados de teste e que não sofra de sobreajuste nos dados de treinamento. A acurácia desse modelo foi de 92.01%.

Após encontrados os melhores hiper parâmetros para cada um dos modelos e realizado o teste de classificação, o SVM foi o método que obteve a melhor acurácia, de 92.65%, e por isso foi o modelo final escolhido para o projeto. A partir daí, ele está pronto para ser implementado em campo para a classificação automática de falhas em máquinas rotativas.

A matriz de confusão na Figura 30 apresenta a comparação entre os dados reais e os estimados para o modelo escolhido de SVM. A matriz demonstra a excelente

capacidade de separação das classes do modelo, evidenciada pelos valores concentrados na diagonal principal, indicando que a maioria das previsões foram corretas. Os valores fora dessa diagonal são pequenos, o que reforça a assertividade do SVM para estimar a categoria real de um experimento.

FIGURA 30 – Matriz de confusão para o conjunto de testes



Fonte: elaborado pelo próprio autor

5.2 RESULTADOS FINAIS

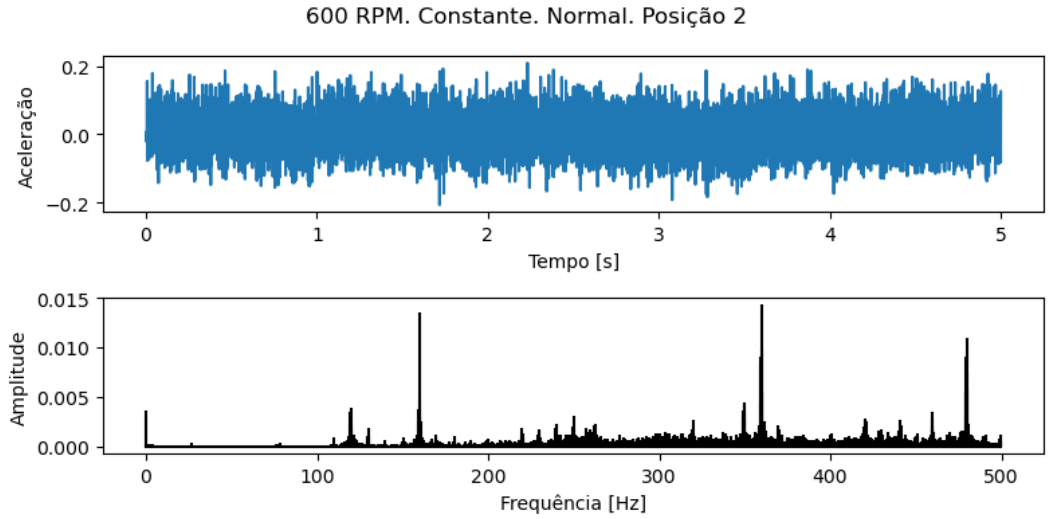
5.2.1 Coleta de dados

A Figura 31 é um exemplo de um dos experimentos coletados em laboratório. Ele mostra a aceleração no eixo X para um sensor na posição do mancal 2 e a máquina funcionando sem defeitos a 600RPM. A linha azul mostra a aceleração no domínio do tempo, enquanto as hastes pretas mostram-na no domínio da frequência.

A Figura 32, por sua vez, apresenta os mesmos dados após serem transformados pela metodologia de pré-processamento indicada na subseção 4.2.4. Apesar de alguns picos na frequência, devido à natureza rotativa da máquina. Esse experimento

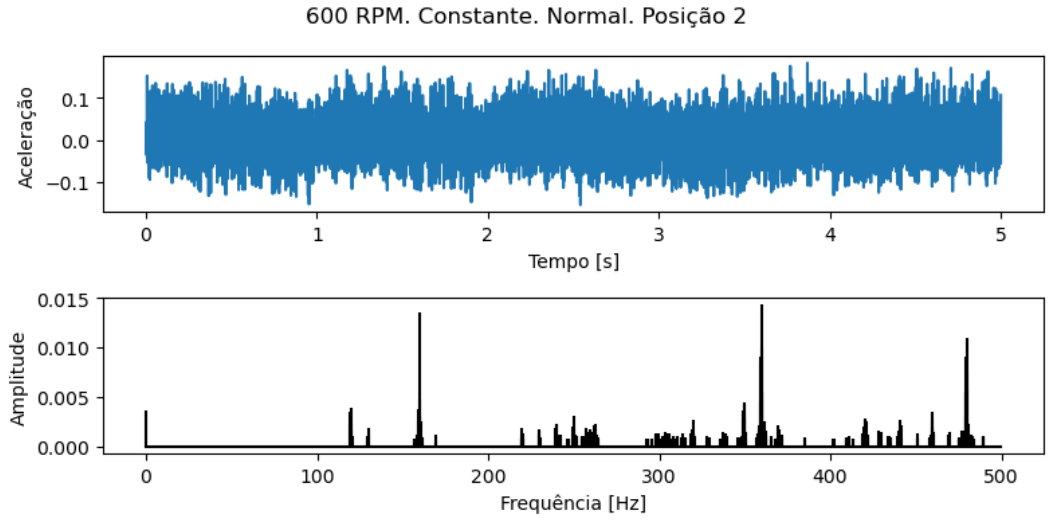
mostrou que um estado sem falhas exibe um comportamento homogêneo e constante. É esperado que em diferentes estados de falhas manifestem diferentes comportamentos, sendo possível catalogá-los e identificá-los de forma automática o mais rápido possível quando eles vierem a acontecer.

FIGURA 31 – Exemplo de dados experimentos coletados. Tanto no domínio do tempo (gráfico azul) quanto na frequência (gráfico preto)



Fonte: O autor

FIGURA 32 – Exemplo de dados pré-processados. No domínio do tempo (gráfico azul) e da frequência (gráfico preto)



Fonte: O autor

Esse experimento também exemplifica uma alteração importante que ocorre após o pré-processamento: a diminuição da amplitude do sinal. Nessas figuras, a escala da aceleração no tempo passou de $4 \cdot 10^{-1} m/s^2$ para $3 \cdot 10^{-5} m/s^2$, o que significa

pouca variação em seu comportamento ao longo do tempo, o que é esperado das máquinas em estado normal.

Cada um desses valores é adicionado em seu respectivo vetor e processado por uma rede LSTM. Os valores dos parâmetros que podem ser extraídos desse exemplo pré-processado podem ser vistos na tabela 13 e seus valores foram usados como entrada no modelo de classificação e no modelo de regressão para a análise de falhas na máquina.

TABELA 13 – PARÂMETROS EXTRAÍDOS

Média	7.361e-07	mm/s²
Desvio Padrão	6.065e-06	mm/s²
Curtose	2.646	*adim.*
Distorção	0.181	*adim.*
Amplitude Pico a Pico	6.110e-06	mm/s²
Valor Quadrático Médio	4.239e-06	mm/s²
Raiz Quadrada da Amplitude	3.722e-05	mm/s²
Fator de Crista	3.212	*adim.*
Fator de Impulso	3.965	*adim.*
Fator de Margem	4.630	*adim.*
Fator de Curtose	1.899e+21	*adim.*
Média (FFT)	0.0002497	mm/s²
Desvio Padrão (FFT)	0.0008239	mm/s²
Valor Quadrático Médio (FFT)	0.0008609	mm/s²
Valor de Pico (FFT)	0.01429	mm/s²
Frequência do Pico (FFT)	359.8	Hz

FONTE: O AUTOR

Ao final do pré-processamento descrito na metodologia e exemplificado na seção anterior, 2,18% dos dados foram considerados *outliers* e descartados. Dos 53 parâmetros iniciais, 9 foram considerados irrelevantes e descartados no primeiro filtro e, em seguida, mais 26 parâmetros apresentaram um alto índice de correlação, sendo descartadas no segundo filtro. Assim, sobraram apenas 19, resultando em uma redução final de 64,15% no número de parâmetros que serão usados para treinar o modelo.

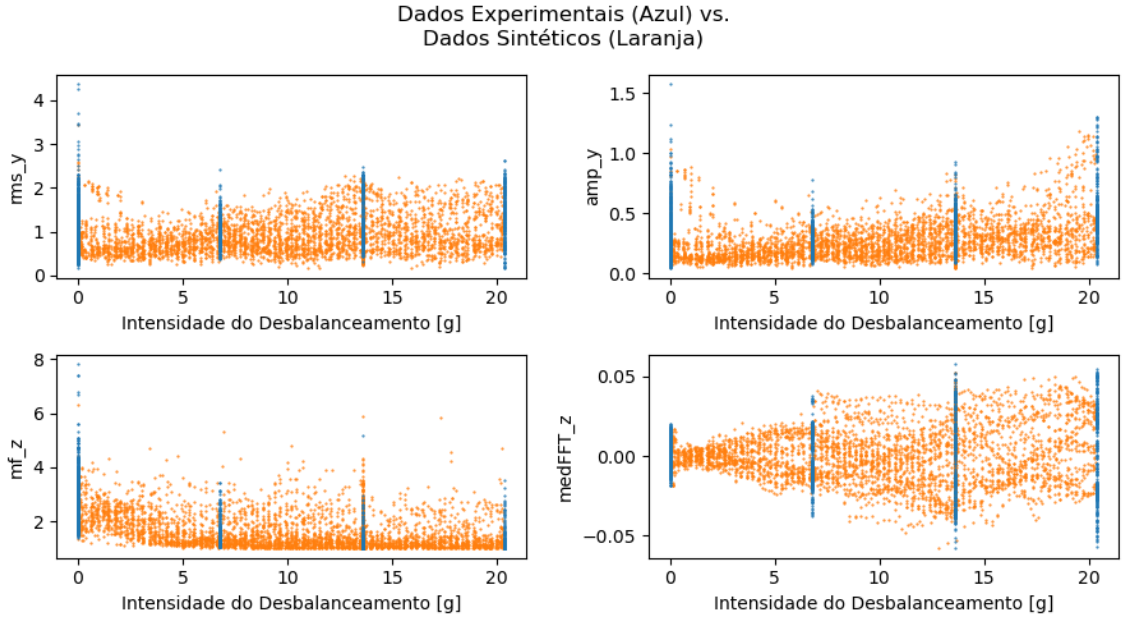
5.2.2 Geração de dados sintéticos

A partir de 891 experimentos originais, foram gerados mais 1.849 experimentos sintéticos, os quais representam todas as combinações possíveis entre estado normal com estado desbalanceado e estado normal com estado desalinhado, tanto crescente quanto decrescente.

Apesar dos dados terem sidos interpolados linearmente no tempo, isso não necessariamente significa que seus parâmetros também estão interpolados linearmente. A figura 33 mostra alguns exemplos do real comportamento dos dados sintéticos para 4 diferentes parâmetros, são eles:

- RMS no eixo Y (rms_y);
- Amplitude no eixo Y (amp_y);
- Fator de margem em Z (mf_z);
- Média no domínio da frequência de Z (medFFT_z).

FIGURA 33 – Visualização da interpolação dos dados sintéticos (laranja) entre os dados bases (azul) no domínio dos parâmetros



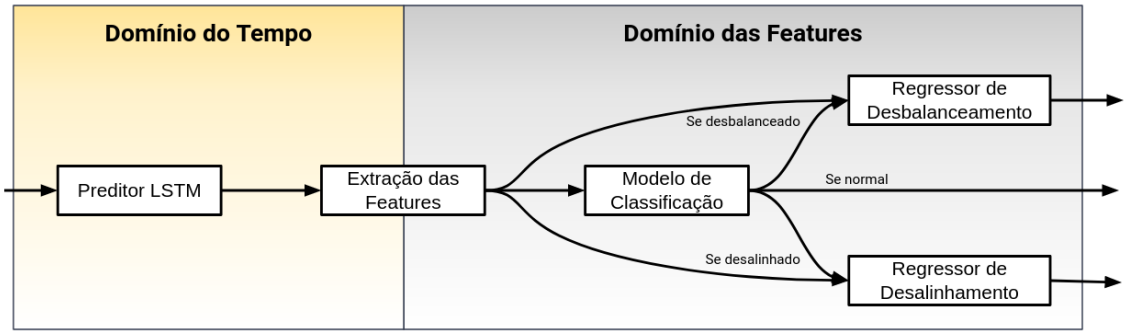
Fonte: O autor

Nele, é possível observar que os dados laboratoriais (azul) estão em passos discretos de intensidade de falha, enquanto os sintéticos compreendem todas as intensidades de falha como um valor contínuo, assim simulando com sucesso o comportamento da máquina com o crescimento gradual da falha e criando dados aptos para o treinamento do LSTM, o que possibilita a previsão do aumento ou diminuição das falhas ao longo do tempo.

5.2.3 Treinamento dos Modelos

Nesta seção, iniciam-se os treinamentos de cada um dos modelos descritos na Figura 24, para, no final, serem unidos como um modelo só. Os detalhes da arquitetura do modelo final podem ser observado na figura 34.

FIGURA 34 – Diagrama da arquitetura do modelo final



Fonte: O autor

5.2.3.1 Modelo de previsão

Para treinar o preditor LSTM, foram utilizados exclusivamente os dados sintéticos, pois o modelo de predição exige dados que apresentem uma variação gradual ao longo do tempo, que é exatamente o que o processo de geração de dados sintéticos simula. Como os dados de base contêm apenas informações sobre comportamentos constantes, o treinamento acabaria atribuindo um peso maior para alguns estados de falha do que outros, o que resulta em um viés durante o treinamento, prejudicando os resultados.

As entradas e saídas foram processadas no domínio do tempo, com uma previsão de 10%, ou seja, a rede LSTM é capaz de prever os dados no futuro em até 10% do tempo dos dados históricos disponíveis. Foram criados modelos distintos e independentes para cada eixo de vibração. Os resultados específicos para cada eixo estão apresentados na Tabela 14.

TABELA 14 – RESULTADOS DO MODELO LSTM

Feats	MSE	MAE
Eixo_x	0.054835 mm^2/s^4	0.146405 mm/s^2
Eixo_y	0.107580 mm^2/s^4	0.175518 mm/s^2
Eixo_z	0.038441 mm^2/s^4	0.110625 mm/s^2

FONTE: O AUTOR

A entrada desse módulo possui 5 segundos de informação, enquanto que a saída possui 0,5 segundo, totalizando 5,5 segundos de informações. Todas são usadas como entrada no extrator de parâmetros. Após os dados no tempo terem seus parâmetros extraídos, estes serão utilizados como entradas nos modelos de classificação e regressão.

5.2.3.2 Modelo de classificação

No treinamento do modelo de classificação, apenas os conjuntos de dados originais (base) sem variação no estado de máquina são utilizados, pois, por não terem os estados misturados, eles são os melhores representantes de cada estado do sistema.

Cada conjunto de dados poderia ser uma das 3 classes possíveis:

- Normal;
- Desbalanceado;
- Desalinhado.

Foram treinados 4 modelos diferentes, o nome de cada modelo e suas respectivas acurácias estão dados na tabela 15. O modelo de rede neural para classificação foi descartada durante os treinamentos devido a complexidade exigida pelo modelo para ter resultados satisfatórios.

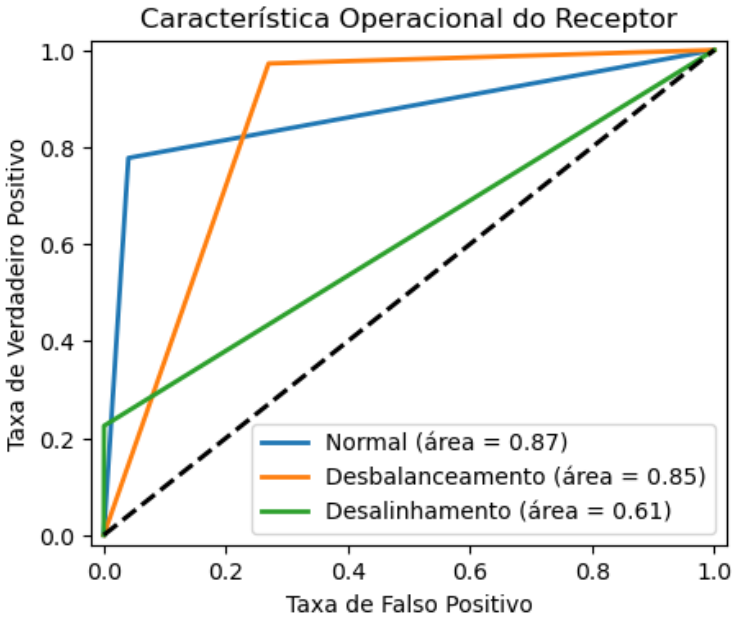
TABELA 15 – RESULTADOS DO CLASSIFICADOR

Modelo	Acurácia
Árvore de decisão	87.93%
Floresta aleatória	92.14%
SVR	90.88%
KNN	86.74%

FONTE: O AUTOR

O modelo com a melhor acurácia foi a floresta aleatória, com 92,14% de acerto entre todos os 3 tipos operacionais possíveis da máquina. A Figura 35 apresenta a curva da característica operacional do receptor (ROC) e a área abaixo da curva (AUC) para cada classe do modelo. Ambos são métricas de medição do desempenho de modelos de classificação em aprendizado de máquina. O ROC é um gráfico que representa a taxa de verdadeiros positivos (TPR) em função da taxa de falsos positivos (FPR) para determinado modelo. O AUC, por sua vez, é a área sob a curva ROC, que varia entre 0 e 1, sendo 1 o valor ideal que indica um modelo perfeito. O AUC permite avaliar o desempenho de forma independente dos pontos de corte escolhidos para a classificação, o que o torna uma métrica amplamente utilizada na escolha do melhor modelo.

FIGURA 35 – Característica operacional do receptor para cada classe do classificador

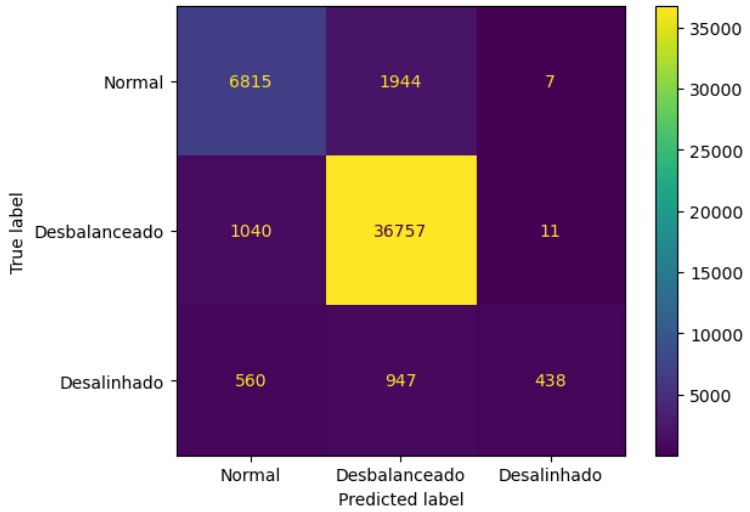


Fonte: O autor

Uma característica peculiar das curvas ROCs apresentadas é que cada uma aparece formada por apenas duas retas. Isso se deve ao fato de o modelo utilizado ter sido a floresta aleatória. Como visto na sessão 2.2.3, esse modelo não retorna valores graduais em seus resultados, mas sim valores descontínuos para diferentes regiões do seu espaço. Essas retas representam essas descontinuidades e diferentes regiões do modelo de floresta aleatória.

A matriz de confusão para o treinamento desse modelo está apresentada na Figura 36. A matriz de confusão é uma das principais ferramentas para avaliar a qualidade de um modelo de classificação. Suas linhas são compostas por suas classes previstas, suas colunas pelas classes reais e as células pelas contagens de pontos que pertencem à classe da linha correspondente e que foram classificados como a classe da coluna correspondente. Seu principal indicador de qualidade é a sua diagonal, que representa predições feitas corretamente, enquanto os valores fora da diagonal indicam erros de classificação.

FIGURA 36 – Matriz de confusão do classificador final



Fonte: O autor

Nesses gráficos, se evidencia que o ótimo resultado do model vem da sua capacidade de classificar com grande precisão os estados de desbalanceamento e normais, que aparecem com maior peso no seu banco de dados de treino e possuem áreas abaixo da curva ROC de 0,85 e 0,87, respectivamente. Enquanto que, mais uma vez, pode-se ver que o modelo teve mais dificuldades para a classificação de desalinhamento, que é o estado com menos participação dentro do banco de dados e possui uma área abaixo da curva ROC de 0,61. Isso evidencia que o classificador apresenta maior dificuldade em classificar falhas de desalinhamento, devido a quantidade menor de dados quando comparado aos outros tipos (dados de desalinhamentos representam apenas 4% do banco de dados completo). Mesmo assim, os resultados mostrados ainda são convenientes para o projeto.

5.2.3.3 Modelo de regressão

Para a previsão da intensidade da falha, foram utilizados dois modelos de regressão independentes, sendo um específico para falhas de desbalanceamento e outro para de desalinhamento. Para ambos os casos, foram utilizados tanto os dados sintéticos, com variação no estado da máquina, quanto os de base, sem variação no estado de máquina. Para cada um, foram testados diversos métodos diferentes de regressão. Os resultados de cada método estão nas tabelas 16 e 17, para as falhas de desalinhamento e desbalanceamento, respectivamente. Os valores da tabela significam o erro de cada modelo, ou seja, quanto menor o valor, mais assertivo ele é.

Para o desalinhamento, o modelo com melhor resultado foi a floresta aleatória, com um erro médio de apenas $4,67 \cdot 10^{-2}$ milímetros. Considerando que o comprimento

TABELA 16 – RESULTADOS DO REGRESSOR DE DESALINHAMENTO

Modelo	MSE	MAE
Árvore de decisão	0.016784 mm²	0.079223 mm
Floresta aleatória	0.007806 mm²	0.046780 mm
SVR	0.013083 mm²	0.090526 mm
KNN	0.012475 mm²	0.080337 mm
Rede Neural	0.167081	0.373396

FONTE: O AUTOR

do segmento de eixo que foi desalinhado é de 25 centímetros, isso representa um desalinhamento angular de apenas $atan\frac{0,047}{25} = 0,107$ graus. Isso significa uma variação desprezível, validando o regressor com um modelo preciso e apto para identificar níveis de desalinhamento em máquinas rotativas.

TABELA 17 – RESULTADOS DO REGRESSOR DE DESBALANCEAMENTO

Modelo	MSE	MAE
Árvore de decisão	17.803424 g²	3.269856 g
Floresta aleatória	8.757786 g²	3.108507 g
SVR	19.328000 g²	3.460872 g
KNN	10.362047 g²	3.219013 g
Rede Neural	155.733040 mm²	11.294327 mm

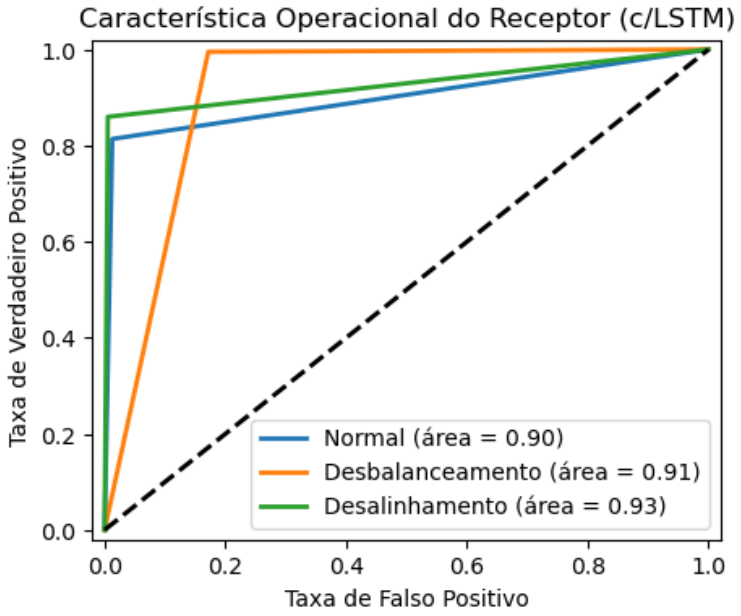
FONTE: O AUTOR

Para o desbalanceamento, o modelo mais eficiente também foi a floresta aleatória, com um erro absoluto médio de 3.1 gramas. Para um maquinário que possui 2 discos, cada um deles pesando 1 quilo, esse modelo foi capaz de identificar variações de desbalanceamento de apenas $0,155\%$ do peso do sistema, representando uma sensibilidade muito boa, sendo capaz de captar falhas até nas suas menores intensidades.

5.2.3.4 Resultados finais

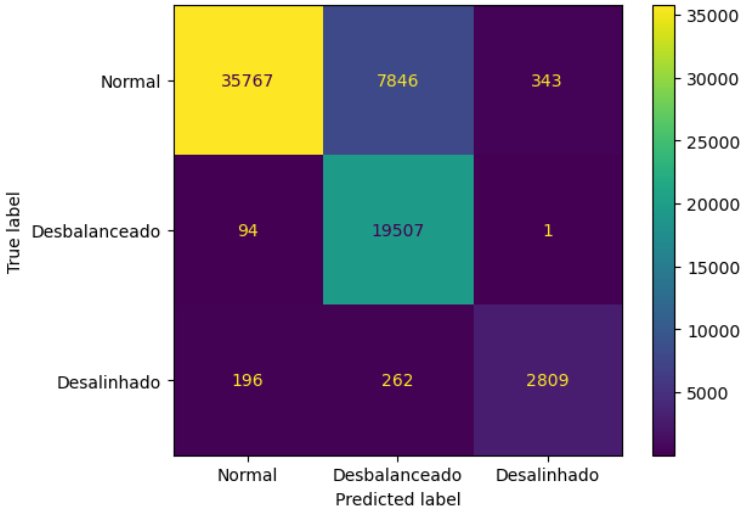
Ao unir os modelos conforme o fluxograma na Figura 34, consegue-se não só fazer o controle de defeitos em tempo real em uma máquina, mas também fazer esse controle com 10% de tempo previsto no futuro, graças à etapa de predição do LSTM. A nova curva ROC e matriz de confusão, considerando a predição do LSTM, podem ser observadas na Figuras 37 e Figura 38, respectivamente. Ao trabalhar com os dados de previsão do modelo de LSTM, espera-se que os resultados finais diminuam ligeiramente, isso se deve à existência de uma margem de erro no preditor, e os modelos de classificação e regressores estarão trabalhando com valores aproximados, ao invés de valores exatos.

FIGURA 37 – Característica operacional do receptor para cada classe do classificador, considerando o preditor LSTM



Fonte: O autor

FIGURA 38 – Matriz de confusão do classificador final, considerando o preditor LSTM



Fonte: O autor

Os dados apresentam um grande aumento na área embaixo da curva do ROC, o que significa que, com a presença dos dados de previsão do LSTM, o modelo final foi capaz de classificar ainda melhor os dados do que sem os dados previsos. Isso indica de que o LSTM obteve sucesso não só de identificar e extrapolar os dados de falha, mas também de acentuar as suas características, facilitando o trabalho para o modelo de classificação. Nos resultados finais, apresentados na tabela 18, as acurácias estão um pouco menores, conforme esperado e discutido anteriormente. Essa diminuição

era esperada, devido à margem de erro do LSTM, mas também pelo banco de dados de treinamento estar mais balanceado, dando menos ênfase ao desbalanceamento, que era o estado que o modelo tinha maior acurácia, e mais ênfase ao desalinhamento, que era o modelo com menor acurácia.

TABELA 18 – RESULTADOS DO MODELO FINAL

Modelo	Acurácia
Classificador floresta aleatória	86,92%
Regressor floresta aleatória (Desalinhamento)	0,202°
Regressor floresta aleatória (Desbalanceamento)	0,207%

FONTE: O AUTOR

Quando considerado a previsão do LSTM, a acurácia do regressor caiu de 92,14% para 86,92%, a margem de erro de desalinhamento aumentou de 0,107° para 0,202° e a de desbalanceamento aumentou de 0,155% da massa total do sistema para 0,207% desse total. Ainda assim, as margens de erro podem continuar sendo consideradas muito pequenas, permitindo ainda ótima sensibilidade para detectar falhas em seus estados iniciais e com bastante acurácia, assim concluindo com sucesso o objetivo de fazer o controle de saúde de máquina de forma online, automatizada e segura.

6 CONSIDERAÇÕES FINAIS

O banco de dados gerado para esse projeto cumpre com os quesitos de qualidade para o treinamento de um modelo de aprendizado de máquinas. Pode, inclusive, ser utilizado por outros estudos independentes que possuam o mesmo propósito. Possui 2.750 experimentos totais (891 dados laboratoriais e 1.849 dados sintéticos), compreendendo 3 diferentes tipos de comportamento de máquinas rotativas e com valores de intensidades contínuos.

O método de pré-processamento e engenharia de parâmetros mostrou ótimos resultados em aumentar a qualidade do treinamento de modelos voltados para sistemas físicos. Ele possui características abrangentes que podem ser utilizadas em aplicações semelhantes e é robusto o suficiente para ser reutilizado sem necessidade de alterações e sem perda de qualidade.

A partir dos dados utilizados, o projeto teve sucesso em desenvolver um modelo de aprendizado de máquinas capaz de prever o comportamento de uma máquina rotativa no futuro e identificar defeitos com antecedência a partir dessa previsão. O modelo de classificação teve sucesso em classificar o comportamento atual da máquina em 92,14% dos casos, representando um resultado excelente. Os regressores para cada um dos dois tipos de defeito também tiveram sucesso nos resultados, contendo erros desprezíveis: 0,107º para o ângulo de desalinhamento e 0,155% do peso da máquina em desbalanceamento.

A etapa de transferência de aprendizado, que uniu todos os modelos em um só, apresentou um modelo sólido e robusto para a identificação de falhas, mas com dados no futuro, extrapolados em 10%. Com esse fator de previsão, o modelo final foi capaz de identificar corretamente o estado da máquina em 86,92% das vezes no futuro, e com erros ainda desprezíveis: 0,202º para o ângulo de desalinhamento e 0,207% do peso da máquina em desbalanceamento.

Com as máquinas rotativas sendo amplamente utilizadas pela indústria e a busca pela sua manutenção preditiva cada vez maior, espera-se, com o desenvolvimento desse método, deixar essa tecnologia ainda mais acessível, com a intenção de causar um impacto positivo significativo às empresas, gerando processos mais eficientes, reduzindo os custos e aumentando a segurança de seus processos.

6.1 TRABALHOS FUTUROS

Estre projeto destinou-se tornar o aprendizado de máquinas mais acessível ao utilizar como linguagem de programação o Python, uma linguagem de fácil interpretação humana e amplamente difundida, e ao deixar os bancos de treino e teste do modelo abertos para o público, assim permitindo que outros pesquisadores gerem seus novos modelos sem a necessidade de iniciar do zero. Com a intenção de dar continuidade para esse projeto, há alguns pontos que podem ser melhorados com pesquisas futuras, quais sejam:

- Aumento no volume de dados de desalinhamento: Com uma baixa representação no banco de dados atual, os modelos têm dificuldade em encontrar os padrões exclusivos desse tipo de defeito. Dar continuidade à aquisição de dados, com ênfase nos experimentos com esse tipo de defeito, poderá melhorar consideravelmente a qualidade dos futuros modelos;
- Pesquisar novos métodos de pré-processamento: A etapa de pré-processamento desse trabalho obteve ótimos resultados, mas não é a única maneira que poderia ter sido feita. Diferentes abordagens, como diferentes parâmetros ou diferentes métodos de seleção de parâmetros, podem apresentar ainda melhores resultados. Por exemplo, abordagens aplicando PCA não foram exploradas durante esse trabalho;
- Otimizar o modelo de redes neurais: Este trabalho buscou comparar diferentes modelos de classificação e regressão, realizando a otimização dos seus hiper parâmetros. Apesar dessa otimização ter mostrado bons resultados para a maioria dos modelos, o modelo de rede neural possui uma complexidade muito maior, e sua otimização não foi explorada na mesma proporção. É possível que, concentrando-se na otimização apenas no modelo de rede neural, sejam encontrado modelos ainda mais eficientes que os modelos de floresta aleatória apresentado nesta dissertação.

Com esses pontos listados, futuros engenheiros podem se inspirar para continuarem explorando essa área de aplicação da tecnologia de aprendizado de máquinas e assim, com seus conhecimentos, colaborar no avanço da engenharia como um todo.

REFERÊNCIAS

ALPAYDIN, E. **Introduction to machine learning**. [S.l.]: MIT press, 2020. Citado 1 vez na página 27.

APICELLA, A.; DONNARUMMA, F.; ISGRÒ, F.; PREVETE, R. **A survey on modern trainable activation functions**. v. 138. [S.l.]: Elsevier Ltd, jun. 2021. P. 14–32. DOI: 10.1016/j.neunet.2021.01.026. Citado 6 vezes nas páginas 12, 28, 30, 31, 33, 37.

BIAO, H.; QIN, Y.; LUO, J.; WU, F.; XIAO, D. Rotating machine fault diagnosis by a novel fast sparsity-enabled feature-energy-ratio method. **ISA Transactions**, Elsevier BV, out. 2022. ISSN 00190578. DOI: 10.1016/j.isatra.2022.10.026. Citado 3 vezes nas páginas 12, 19.

BRAUN, S. G.; EWINS, D. J.; RAO, S. S. **Encyclopedia of Vibration: FP**. [S.l.]: Academic press, 2002. v. 2. Citado 3 vezes nas páginas 16, 18, 19.

CHATTERJEE, J.; MOHANTY, S. A comparative study of machine learning algorithms for classification of neurological disorders. **Journal of Medical Systems**, Springer, v. 44, n. 12, p. 209, 2020. DOI: 10.1007/s10916-020-01656-6. Citado 1 vez na página 22.

CHOW, M.-Y. Guest editorial special section on motor fault detection and diagnosis. **IEEE Transactions on Industrial Electronics**, IEEE, v. 47, n. 5, p. 982–983, 2000. Citado 1 vez na página 38.

FJELLSTRÖM, C.; NYSTRÖM, K. Deep learning, stochastic gradient descent and diffusion maps. **Journal of Computational Mathematics and Data Science**, Elsevier BV, v. 4, p. 100054, ago. 2022. ISSN 27724158. DOI: 10.1016/j.jcmds.2022.100054. Citado 1 vez na página 31.

GHORBANI, M.; NOURELFATH, M.; GENDREAU, M. A two-stage stochastic programming model for selective maintenance optimization. **Reliability Engineering and System Safety**, Elsevier Ltd, v. 223, jul. 2022. ISSN 09518320. DOI: 10.1016/j.ress.2022.108480. Citado 1 vez na página 12.

JIJO, B. T.; ABDULAZEEZ, A. M. Classification based on decision tree algorithm for machine learning. **evaluation**, v. 6, n. 7, 2021. Citado 2 vezes nas páginas 23, 25.

KAWAI, K.; IRINO, N.; IIYAMA, K.; MATSUBARA, A.; MIZUYAMA, H.; MORI, K.; FUJISHIMA, M. A prediction model for high efficiency machining conditions based on machine learning. In: v. 101, p. 54–57. ISBN 9781713835431. DOI: 10.1016/j.procir.2020.09.188. Citado 1 vez na página 13.

KIM, Y. et al. Analysis and processing of shaft angular velocity signals in rotating machinery for diagnostic applications. in Acoustics, Speech, and Signal Processing, 1995. Citado 1 vez na página 38.

LEBOLD, M.; MCCLINTIC, K.; CAMPBELL, R.; BYINGTON, C.; MAYNARD, K. Review of vibration analysis methods for gearbox diagnostics and prognostics. In: CITESEER. PROCEEDINGS of the 54th meeting of the society for machinery failure prevention technology. [S.l.: s.n.], 2000. v. 634, p. 16. Citado 1 vez na página 38.

LEI, Y.; YANG, B.; JIANG, X.; JIA, F.; LI, N.; NANDI, A. K. **Applications of machine learning to machine fault diagnosis: A review and roadmap**. v. 138. [S.l.]: Academic Press, abr. 2020. DOI: 10.1016/j.ymssp.2019.106587. Citado 0 vez na página 29.

LI, L.; XU, J.; LI, J. Estimating remaining useful life of rotating machinery using relevance vector machine and deep learning network. **Engineering Failure Analysis**, Elsevier Ltd, v. 146, abr. 2023. ISSN 13506307. DOI: 10.1016/j.engfailanal.2023.107125. Citado 4 vezes nas páginas 39–43.

LI, Q.; DING, L.; LI, Y.; LIANG, Y. A Novel Hybrid Feature Selection Method Based on PCA and Random Forest. **IEEE Access**, IEEE, v. 9, p. 41398–41411, 2021. DOI: 10.1109/ACCESS.2021.3062451. Citado 1 vez na página 26.

LI, Y.; WANG, W.; ZHANG, J.; LU, X.; LI, T.; CHEN, F. A comparative study of machine learning algorithms for real-time prediction of engine fuel consumption and emissions. **Fuel**, Elsevier, v. 265, p. 116944, 2020. DOI: 10.1016/j.fuel.2019.116944. Citado 2 vezes nas páginas 26, 28.

LIU, J.; PAN, C.; LEI, F.; HU, D.; ZUO, H. Fault prediction of bearings based on LSTM and statistical process analysis. **Reliability Engineering System Safety**, v. 214, p. 107646, 2021. ISSN 0951-8320. DOI: https://doi.org/10.1016/j.ress.2021.107646. Disponível em: https://www.sciencedirect.com/science/article/pii/S0951832021001873. Citado 1 vezes nas páginas 43, 44.

LIU, X.; JIANG, H.; LI, R. A comparative study of machine learning algorithms for credit scoring. **Journal of Intelligent & Fuzzy Systems**, IOS Press, v. 36, n. 3, p. 2763–2771, 2019. DOI: 10.3233/JIFS-181233. Citado 1 vez na página 25.

LU, X.; GAO, H.; ZHAO, X.; GAO, H.; YU, Q. A comparative study of machine learning algorithms for building energy consumption prediction. **Applied Energy**, Elsevier, v. 260, p. 114292, 2020. DOI: 10.1016/j.apenergy.2019.114292. Citado 1 vez na página 27.

MA, X.; XIE, X.; LI, X.; LI, X.; LI, J. Research on the classification algorithm of elevator faults based on machine learning. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 11, n. 7, p. 3073, 2021. DOI: 10.3390/app11073073. Citado 1 vez na página 32.

MCFADDEN, P.; SMITH, J. The vibration produced by multiple point defects in a rolling element bearing. **Journal of Sound and Vibration**, v. 98, n. 2, p. 263–273, 1985. ISSN 0022-460X. DOI: https://doi.org/10.1016/0022-460X(85)90390-6. Disponível em: https://www.sciencedirect.com/science/article/pii/0022460X85903906. Citado 1 vez na página 38.

MOREIRA, F.; RIBEIRO, L. **SIMILARITY-BASED METHODS FOR MACHINE DIAGNOSIS**. [S.l.], 2018. Citado 1 vezes nas páginas 56, 61.

NASRFARD, F.; MOHAMMADI, M.; KARIMI, M. A Petri net model for optimization of inspection and preventive maintenance rates. **Electric Power Systems Research**, Elsevier Ltd, v. 216, mar. 2023. ISSN 03787796. DOI: 10.1016/j.epsr.2022.109003. Citado 1 vez na página 12.

NUNES, P.; ROCHA, E.; SANTOS, J.; ANTUNES, R. Predictive maintenance on injection molds by generalized fault trees and anomaly detection. **Procedia Computer Science**, v. 217, p. 1038–1047, 2023. ISSN 18770509. DOI: 10.1016/j.procs.2022.12.302. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S1877050922023808. Citado 1 vez na página 13.

PAUL, A. R.; BISWAS, S.; MUKHERJEE, M. Conceptualisation of a novel technique to incorporate artificial intelligence in preventive and predictive maintenance in tandem. **Materials Today: Proceedings**, Elsevier Ltd, v. 66, p. 3814–3821, jan. 2022. ISSN 22147853. DOI: 10.1016/j.matpr.2022.06.250. Citado 1 vez na página 12.

RAZA, S. Z.; KHAN, A. Y.; MAHMOOD, A.; KHAN, F. A.; KHAN, Z. A. A comparative study of machine learning regression algorithms for prediction of oil production. **Journal of Petroleum Science and Engineering**, Elsevier, v. 174, p. 11–22, 2019. DOI: 10.1016/j.petrol.2018.10.012. Citado 1 vez na página 25.

SCHWENDEMANN, S.; AMJAD, Z.; SIKORA, A. **A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines**. v. 125. [S.l.]: Elsevier B.V., fev. 2021. DOI: 10.1016/j.compind.2020.103380. Citado 1 vez na página 12.

SINGH, P. K.; KUMAR, M.; SINGH, A. K.; SRIVASTAVA, M. A review on regression models with application in engineering field. **Materials Today: Proceedings**, Elsevier, v. 26, p. 4014–4019, 2020. DOI: 10.1016/j.matpr.2020.04.611. Citado 3 vezes nas páginas 24, 30.

SINHA, J.; LEES, A.; FRISWELL, M. Estimating unbalance and misalignment of a flexible rotating machine from a single run-down. **Journal of Sound and Vibration**, v. 272, n. 3, p. 967–989, 2004. ISSN 0022-460X. DOI: https://doi.org/10.1016/j.jsv.2003.03.006. Disponível em: https://www.sciencedirect.com/science/article/pii/S0022460X03008502. Citado 0 vez na página 18.

TANDON, N.; CHOUDHURY, A. A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. **Tribology International**, v. 32, n. 8, p. 469–480, 1999. ISSN 0301-679X. DOI: https://doi.org/10.1016/S0301-679X(99)00077-8. Disponível em: https://www.sciencedirect.com/science/article/pii/S0301679X99000778. Citado 3 vez na página 38.

TAUNK, K.; DE, S.; VERMA, S.; SWETAPADMA, A. A brief review of nearest neighbor algorithm for learning and classification. In: IEEE. 2019 International Conference on Intelligent Computing and Control Systems (ICCS). [S.l.: s.n.], 2019. P. 1255–1260. Citado 2 vezes nas páginas 22, 23.

WANG, Z.; XIA, H.; YIN, W.; YANG, B. An improved generative adversarial network for fault diagnosis of rotating machine in nuclear power plant. **Annals of Nuclear Energy**, Elsevier Ltd, v. 180, jan. 2023. ISSN 18732100. DOI: 10.1016/j.anucene.2022.109434. Citado 1 vez na página 16.

YANG, D.-M.; STRONACH, A.; MACCONNELL, P.; PENMAN, J. THIRD-ORDER SPECTRAL TECHNIQUES FOR THE DIAGNOSIS OF MOTOR BEARING CONDITION USING ARTIFICIAL NEURAL NETWORKS. **Mechanical Systems and Signal Processing**, v. 16, n. 2, p. 391–411, 2002. ISSN 0888-3270. DOI: https://doi.org/10.1006/mssp.2001.1469. Disponível em: https://www.sciencedirect.com/science/article/pii/S0888327001914694. Citado 1 vez na página 38.

ZARZYCKI, K.; ŁAWRYŃCZUK, M. Advanced predictive control for GRU and LSTM networks. **Information Sciences**, Elsevier Inc., v. 616, p. 229–254, nov. 2022. ISSN 00200255. DOI: 10.1016/j.ins.2022.10.078. Citado 1 vez na página 32.

ZHANG, H.; FAN, S.; ZHANG, X.; YANG, X.; WANG, Z. Comparison of machine learning algorithms for predicting compressive strength of steel fiber-reinforced concrete. **Construction and Building Materials**, Elsevier, v. 271, p. 121472, 2021. DOI: 10.1016/j.conbuildmat.2020.121472. Citado 3 vezes nas páginas 21–23.

ZHONG, X.; BAN, H. Crack fault diagnosis of rotating machine in nuclear power plant based on ensemble learning. **Annals of Nuclear Energy**, Elsevier Ltd, v. 168, abr. 2022. ISSN 18732100. DOI: 10.1016/j.anucene.2021.108909. Citado 1 vez na página 39.

7 APÊNDICE 1

Neste apêndice 1 encontra-se o artigo do COBEM publicado em 2023, referente a este trabalho, em conjunto com a orientadora.

COBEM 2023 27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-1198
Fault detection in rotating machines using LSTM and time series analysis

Daniel Awada Elarrat Canto
Maurizio Radloff Barghouthi
Leonardo Kaucz
Eduardo Márcio de Oliveira Lopes
Giuliana Sardi Venter
Federal University of Paraná
daniel.awada@ufpr.br
giuliana.venter@ufpr.br

Abstract. *The need to reduce the downtime in machines has led to increased interest in process optimization in various fields, including mechanical engineering. One area of focus is predictive machine health analysis, which utilizes machine learning to predict when a serious failure may occur in a machine and minimize downtime. While predicting machine failures is a complex task, recent advances in machine learning have made this method more accessible and attracted growing interest from industries and researchers seeking to improve maintenance practices. This study focuses on the detection of faults in rotating machines caused by unbalance using machine learning. Experimental data was collected on a rotating machine test bench DAC VAD 203 under different conditions, with rotating speeds varying from 300rpm to 3300rpm and a forced unbalance faults ranging from 6.80 g to 20.4 g that, combined, resulted in 266 different experiments. Additional 242 experiments were synthetically generated by interpolating data with the same rpm and different forced unbalance, creating a new set of data that simulates the behavior of a machine when its unbalance intensity is changing. The data was collected using a triaxial accelerometer positioned on top of a selected ball bearing of a rotating system and then processed into time series by windowing the raw sensor data into equal time intervals of 0.5 seconds and extracting the time and frequency-domain features for each time segment, allowing the visualization of fault occurrences and analysis over time. A composed model consisted of a layer of a recurrent neural network and a regressor model was chosen and trained with the dataset in order o automate the task of fault and intensity detection. The recurrent model used was the Long-Short Term Memory (LSTM), as it has a good performance to make prediction based on time series. The model was tuned using an optimizer based on the Gradient Descent algorithm in Python, which iteratively adjusts the weights and biases of the model to minimize the loss function and improve the accuracy of its predictions. An margin of error lower than 6.8 g in unbalance is expected, with a short processing time, based on preliminary results. Comparing with other strategies, it is expected that this new method can significantly outperform previous approaches, highlighting the importance of tailored optimization strategies for complex machine learning models.*

Keywords: *Machine Learning, Rotating Dynamics, Synthetic Data, Process Monitoring, Recurrent Neural Networks*

1. INTRODUCTION

Rotating machines play a fundamental role in various industrial sectors, driving production processes and ensuring the proper functioning of equipment and operational systems. However, like any mechanical equipment, they are also susceptible to failures, which can compromise their efficiency and cause significant damage. Among the most common failures is the unbalance, characterized by the non-uniform distribution of mass along the rotor. The early detection of these failures is extremely important to prevent unplanned downtime, production losses and safety risks (**?**).

In this context, predictive maintenance has proven to be an effective approach in the management and maintenance of rotating machines. Unlike preventive maintenance, which is based on fixed time intervals, and corrective maintenance, which occurs only after failure, predictive maintenance uses data and advanced techniques for continuous monitoring of the machine's health, allowing for prediction and proactive action regarding potential failures (Schwendemann *et al.* (2021)). However, its implementation is still challenging due to its complexity and the high initial investments required for monitoring technologies and data analysis (Kawai *et al.* (2020)). In this context, the use of machine learning to monitor condition of production systems in general (Żabiński *et al.* (2019)) and of rotors especifically (Zhu *et al.* (2023)) emerges as a promising approach to tackle the complexity of the problem and provide efficient solutions.

Although the use of machine learning to predict faults in rotating machinery has been extensively researched, with applications ranging from using simple models such as Random Forests (Hu *et al.* (2020)) to very deep neural networks (Xu *et al.* (2022)), the analysis of the system in time series has been scarce. One complication of the analysis in time series resides in the dataset used to train the models, as it is necessary that the dataset change over time, which is both time consuming and difficult to obtain.

This article presents a detailed study on the application of machine learning for the detection of failures in rotating machines, with a focus on unbalance, in the time series. The proposed model is based on the combination of a Long Short-Term Memory (LSTM) networks, widely used in time series analysis, with a regression model, that aims to predict the machine unbalance based on the machine behavior. The use of time series data allows for capturing dynamic information about the machine behavior over time and making predictions about its future behavior, providing the regression model accurate estimates of unbalance in the future time. In order to do so, a time series synthetic data generation method is used.

Throughout the article, the details of the proposed model will be presented, as well as the methods used for data collection and preprocessing. After that, the results obtained through laboratory experiments will be discussed, comparing the performance of different regression models, such as Decision Tree, Random Forest, SVR, KNN, and the LSTM neural network. The analysis of the results will evaluate the effectiveness of the proposed model in identifying and analyzing failures in rotating machines.

The results obtained so far are promising, demonstrating the effectiveness of this method in accurately and reliably identifying faults in rotating machines. With this advancement, it is expected that the implementation of predictive maintenance in the industry will be more widespread and efficient, contributing to process optimization and ensuring operational safety.

2. METHODS

2.1 Data gathering

To obtain the data necessary for the training of machine learning models, a series of experiments was conducted in a controlled laboratory. The experiment involved using a rotating machine, a dynamic signal acquisition module, a triaxial accelerometer, weights to induce unbalance in the machine and software for data visualization.

The variables parameters in the experiments were the rotations per minute of the machine and the level of the induced unbalance. The rotations per minute was adjusted in the range of 300 to 3300, with increments of 300. As for the unbalance, four levels were used: 0.00 g (normal state), 6.80 g, 13.60 g, and 20.4 g of forced unbalance was induced by adding weights to each of the two discs of the system. The experiments were conducted with different combinations of weights, ranging from 0 to 3 weights on each disc.

Each experiment had a duration of 5 seconds, with the exception of the experiments with 0 g of unbalance, that have 1 minute each to balance the amount of data in normal state and in fault state. For each experiment the acceleration data were collected on the three axes (X, Y, and Z) of the triaxial accelerometer, along with the corresponding timestamp. The sampling frequency used was 1000 Hz, ensuring a high temporal resolution in data capture.

The use of this laboratory method enabled the acquisition of a diverse and controlled dataset, ranging different combinations of RPM and unbalance levels. This data is essential for the training and validation of the machine learning models, enabling precise and efficient analysis in detecting faults in rotating machines.

2.2 Time series synthetic data generation

In order to predict the fault in time series, it is necessary to create a dataset that changes considerably in time. Although it would be possible to create such a dataset by collecting data for a long period of time, it would be cost prohibitive. Therefore, the data on the machine's behavior during the transition from its normal state to an arbitrary unbalanced state - or with variations on the intensity of this unbalance - is generated trough a time series synthetic data generation. To achieve this, additional data was added to the dataset by interpolating data with different unbalance intensities and simulating the machine's behavior in these scenarios.

The data interpolation was performed linearly over time between two or more time series. For each quantity of interpolated series, the intensity of each one and the visualization of the used windowing can be seen in Figure 1. The sum of all windows is always constant and equal to 1 due to the proportionality in the decrease of one window in relation to the increase of the others, ensuring conservation of the signal's amplitude and energy during the interpolation (Sestito *et al.* (2022)).

The interpolations were only performed between time series with the same rotations per minute and with unbalance intensity ordered in ascending order. Thus, considering all possible combinations of different unbalances and rotations, 242 new synthetic experiments were generated in the database, improving the quality of the data and the training of the machine learning models.

2.3 Pre-processing

To emphasize the essential properties of the data and improve the quality of the model training, a pre-processing module was performed in two steps. The first step involved removing frequencies with low amplitude, where any frequency

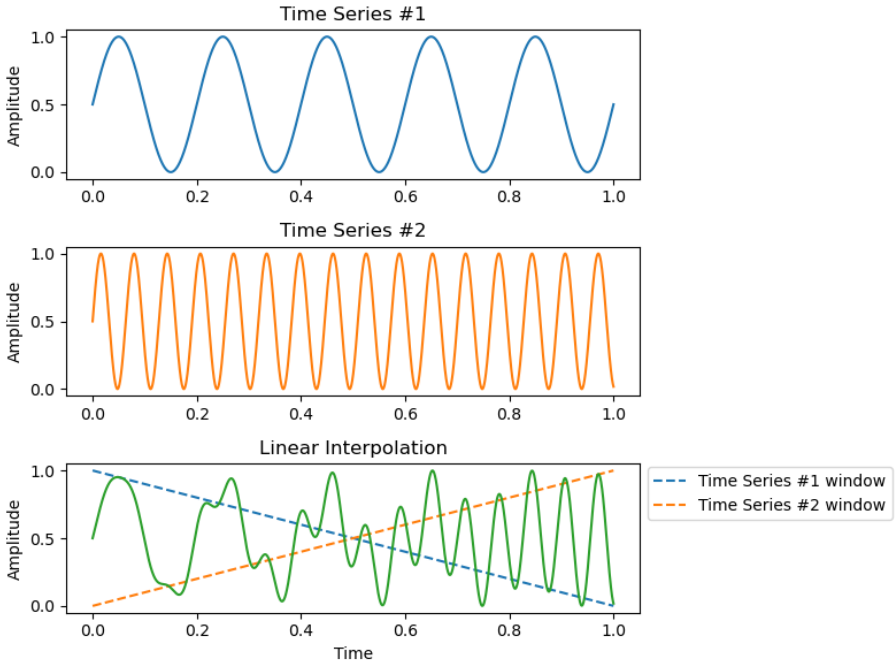


Figure 1. Example of a linear interpolation between 2 time series

with an amplitude less than 5% of the peak was considered noise and its amplitude was reduced to 0. In the second step, isolated frequency with an amplitude equal or less than 20% of the peak is identified and have its amplitude also reduced to 0. Where 'isolated frequency' means a frequency with the value of its adjacent frequencies equals to 0.

Figure 2 illustrates an example signal at each stage of the pre-processing, from the transformation to the frequency domain using the Fast Fourier Transform to the final signal reconstruction. The ultimate goal of this pre-processing is to remove a significant portion of the noise present in the signals and emphasize relevant vibration frequencies.

2.4 Feature Extraction

After completing the processing, it is necessary to extract the features that will be used in the prediction model. For this purpose, the time series are synthesized into unique values that represent different properties of the machine's behavior, both in the time domain and in the frequency domain. The chosen properties were defined independently for each of the X, Y, and Z axes, resulting in a total set of 48 features.

The definitions and calculations of the properties can be found in Table 1, where the symbol N represents the number of data points in the series, x_i represents a given value at a time instant, and X_i is a the absolute value of an amplitude in the frequency domain. These features are calculated using aggregation functions applied to the original time series, allowing the capture of relevant information for fault detection.

These features were chosen for its ability of summarizing the main characteristics of a signal as a whole. They represent the system's behavior in various different aspects, where each one of them may influence the target in a different way. By using these features as inputs, it is found suitable ways to adjust a machine in order to achieve the desired target Ribeiro (2018).

2.5 Outliers Removal

To ensure the quality of the data used for training the machine learning model, this study includes a custom outlier removal method. The outliers are identified calculating the distances between the maximum and minimum points with respect to the series mean and them compared with the 99th and 1st percentiles. If the ratio between the distances is equal to or greater than 2, the point is considered an outlier and is excluded from the database. The impact of outlier removal is visualized in the Figure 3, showing the series mean as the black line, the 99th and 1st percentiles as the orange lines, and the maximum and minimum as the green lines. Initially the dataset with all the features extracted both by the experimental data and the synthetic data was consisted by 20,900 samples with 823 of them being considered outliers and removed from the dataset, a reduction of 3.94%. This outlier removal process is crucial in preparing the data for regression model training, improving the quality of the training set and enhancing the model's performance (Theissler *et al.* (2021)).

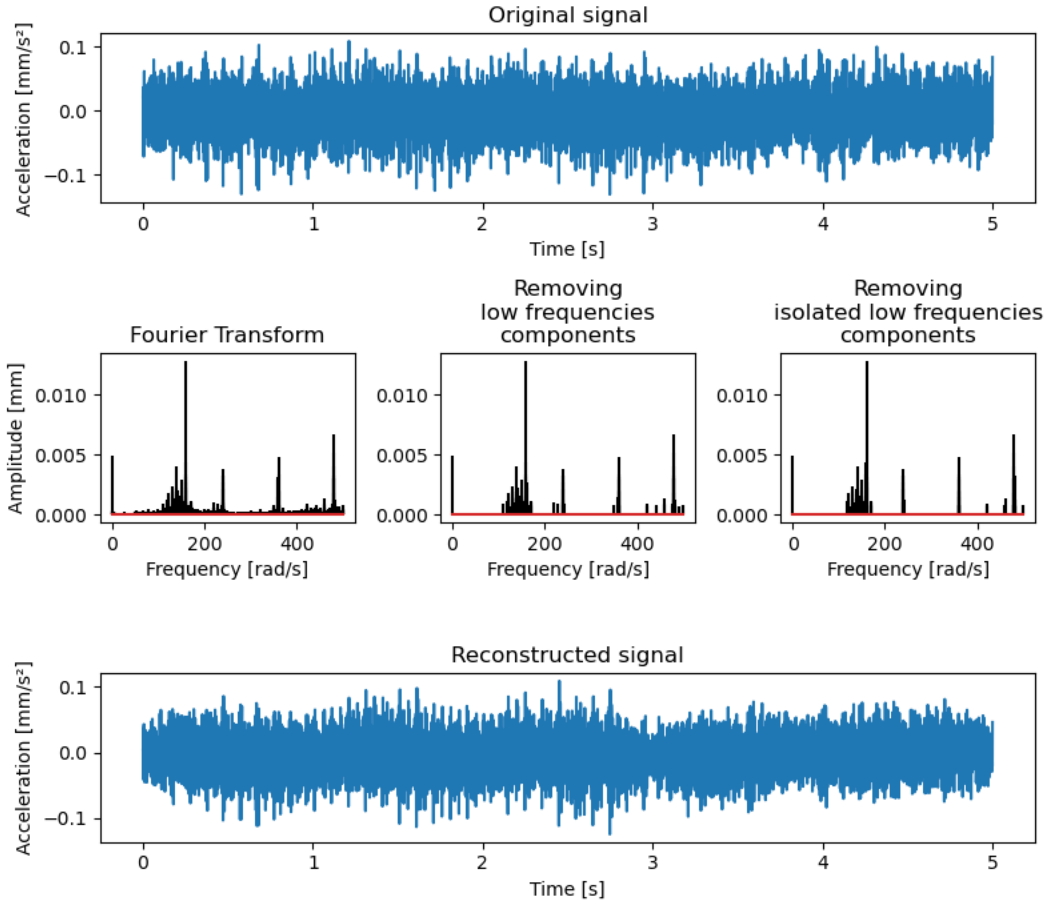


Figure 2. Visualization of every step of the pre-processing

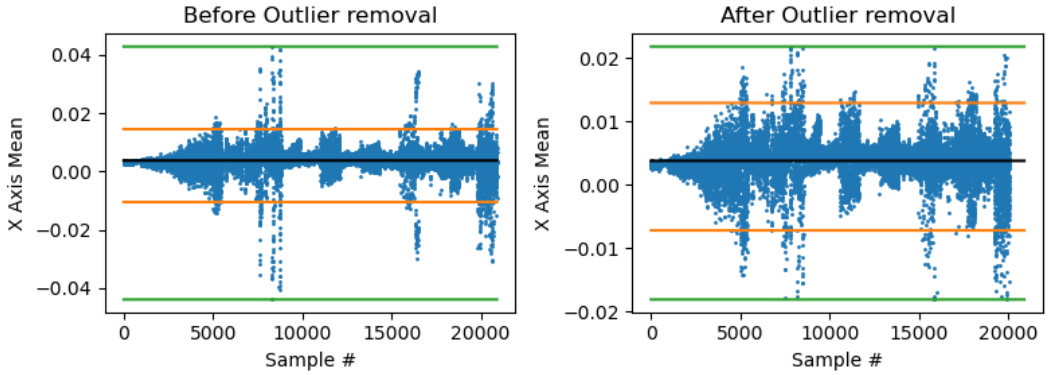


Figure 3. Before and after the Outlier Removal step for one of the features

2.6 Feature Selection

To reduce the number of features and speed up the method for both training and implementation of the final model, two filters will be applied: the information quality filter and the correlation filter. The first filter, the informational quality of a feature, measures the ratio between the standard deviation within a class and the standard deviation between classes. If the data within a class for a feature are all similar but different from the other classes, that feature will be considered highly informative; otherwise, it will be considered less informative.

Furthermore, having features with high correlation hinders training because it gives more weight to certain properties than others. Therefore, the next step in the feature selection is to remove those with high correlation. The chosen threshold value is 0.8, meaning that if two features have a correlation equal or greater than this value, one of them will be disregarded. This filter will return 20 final features that will be considered in the final training of the models, which are presented in Table 2.

Table 1. Features definitions

Feature	**Symbol**	**Definition**
Mean	μ_x	$\mu_x = \frac{1}{N}\sum_{i=0}^{N} x_i$
Standard Deviation	σ_x	$\sigma_x^2 = \frac{1}{N}\sum_{i=0}^{N}(x_i - \mu_x)^2$
Kurtosis	κ_x	$\kappa_x = \frac{1}{N}\sum_{i=0}^{N}\left(\frac{x_i-\mu_x}{\sigma_x}\right)^4$
Skewness	γ_x	$\gamma_x = \frac{1}{N}\sum_{i=0}^{N}\left(\frac{x_i-\mu_x}{\sigma_x}\right)^3$
Peak-to-Peak Amplitude	x_{ppv}	$x_{ppv} = max(x_i) - min(x_i)$
Root Mean Square	x_{rms}	$x_{rms} = \left(\frac{1}{N}\sum_{i=0}^{N} x_i^2\right)^{1/2}$
Square Root Amplitude	x_{sra}	$x_{sra} = \left(\frac{1}{N}\sum_{i=0}^{N}\sqrt{\lvert x_i\rvert}\right)^2$
Crest Factor	x_{cf}	$x_{cf} = \frac{max(\lvert x_i\rvert)}{x_{rms}}$
Impulse Factor	x_{if}	$x_{if} = \frac{max(\lvert x_i\rvert)}{\frac{1}{N}\sum_{i=0}^{N}\lvert x_i\rvert}$
Margin Factor	x_{mf}	$x_{mf} = \frac{max(\lvert x_i\rvert)}{x_{sra}}$
Kurtosis Factor	x_{kf}	$x_{kf} = \frac{\kappa_x}{x_{rms}^4}$
Mean (FFT)	μ_x	$\mu_X = \frac{1}{N}\sum_{i=0}^{N} X_i$
Standard Deviation (FFT)	σ_x	$\sigma_X^2 = \frac{1}{N}\sum_{i=0}^{N}(X_i - \mu_X)^2$
Root Mean Square (FFT)	x_{rms}	$x_{rms} = \left(\frac{1}{N}\sum_{i=0}^{N} X_i^2\right)^{1/2}$
Peak Value (FFT)	X_{max}	$X_{max} = max(X)$
Peak frequency (FFT)	f_{pico}	$f_{pico} = f(X_{max})$

Table 2. Selected features

No.	**Axis**	**Feature**
1	X Axis	Mean
2	X Axis	Peak-to-Peak Amplitude
3	X Axis	Kurtosis
4	X Axis	Skewness
5	X Axis	Kurtosis Factor
6	X Axis	Mean (FFT)
7	Y Axis	Mean
8	Y Axis	Kurtosis
9	Y Axis	Skewness
10	Y Axis	Kurtosis Factor
11	Y Axis	Impulse Factor
12	Y Axis	Peak Value (FFT)
13	Y Axis	Mean (FFT)
14	Z Axis	Mean
15	Z Axis	Kurtosis
16	Z Axis	Skewness
17	Z Axis	Kurtosis Factor
18	Z Axis	Margin Factor
19	Z Axis	Mean (FFT)
20	Z Axis	Peak Value (FFT)

An additional feature also considered is the rpm, due to its importance for unbalance prediction, totaling 21 resources in total.

2.7 LSTM training

The LSTM model has the objective to predict the features values in the future based on its historic values (Ma *et al.* (2021)). Before being used in the LSTM model, the signal is normalized relative to the test data, ensuring that all features are in the range of values from 0 to 1. A single LSTM model is built to predict all features at once, with one recurrent layer containing 16 units and an output dense layer with 21 units corresponding to the 21 input features. The model uses the Mean Squared Error as its loss function, is trained with 36 samples for 100 epochs and a batch size of 16. It is important to remember that these data points form a time series, as shown in the Figure 4. Given the features for each chunk of a signal, the LSTM model should be capable to predict the feature of the next chuck in the future

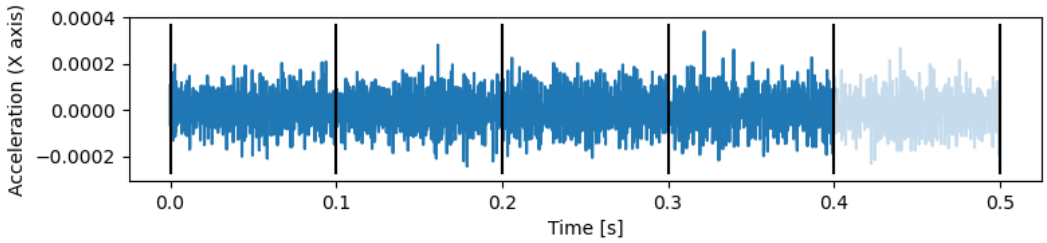


Figure 4. Time series associated to the LSTM model

This LSTM model is capable of making predictions for a time horizon of 0.1 seconds. This prediction can be performed iteratively to project further into the future, but compromising the accuracy of the predictions. The metric used to evaluate the performance of the LSTM model is Mean Squared Error (MSE), which measures the average squared error between the model's predictions and the actual values.

2.8 Regressor model training

To find the regression model that accurately estimates the unbalance of a rotating machine, five different model options were explored: Decision Tree, Random Forest, Support Vector Regression (SVR), K-Nearest Neighbors (KNN), and Artificial Neural Network (ANN). An hyperparameter tuning was performed for each model with the intention of finding the best possible model for the problem.

The MSE will serve as the key metric for selecting the best regressor model to be combined with the LSTM layer, resulting in the final model. A lower MSE value indicates superior regression performance, making it the primary criterion for model selection. Detailed explanations of the MSE and other relevant metrics can be found in the subsequent section.

2.9 Model validation

To evaluate the errors in regression tasks, three metrics were used: the coefficient of determination R-Squared (R^2) which represents how well the model represents the dataset, given by the Eq. 1, the Mean Absolute Error (MAE), which represents the margin of error of the predictor, given by the Eq. 2, and the Mean Squared Error (MSE), that represents how well the predicted data fits the real data, given by the Eq. 3.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(Z_{s,i} - \hat{Z}_{s,i})^2}{\sum_{i=1}^{n}(Z_{s,i} - \bar{Z}_s)^2}, \tag{1}$$

$$MAE = \frac{\sum_{i=1}^{n} \mid \hat{Z}_{s,i} - Z_{s,i} \mid}{n}, \tag{2}$$

$$MSE = \frac{\sum_{i=1}^{n}(\hat{Z}_{s,i} - Z_{s,i})^2}{n}, \tag{3}$$

in which n is the number of datapoints on the dataset, $Z_{s,i}$ is the actual target value, $\hat{Z}_{s,i}$ is the predicted target value, and $\bar{Z}_s$ is the average of the target values.

All the models were trained and tested with a 0.2 train test split, in which 20% of the dataset is randomly separated to be used as a validation dataset.

3. RESULTS

3.1 LSTM model

Performance evaluation of the LSTM model was made based on three metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Coefficient of Determination (R2). The values for each metric are presented in the Table 3.

Table 3. LSTM Model errors metrics

Metric	Value
MSE	3.1868e-3
MAE	3.3125e-2
R2	0.634221

These results highlight the effectiveness of the LSTM model for this proposed method, providing accurate and reliable behavior predictions based on the input features. The low error rates, both in terms of MSE and MAE, indicate a good generalization ability of the model for unseen data. Additionally, the significant R2 value demonstrates the model's capacity to explain a considerable portion of the variability in the test data.

The obtained results reinforce the relevance and applicability of the LSTM model in the project's context, providing valuable insights for understanding and predicting the desired behavior. These findings contribute to the advancement of the machine learning field and emphasize the feasibility and effectiveness of the approach adopted in this study.

3.2 Regressor model

The training results for each of the five models are presented in Table 4 and the model with the lowest Mean Squared Error (MSE) score was considered the best one. The table shows that the Random Forest model obtained the lowest MSE value (9.13) compared to the others, indicating it is the most efficient model for estimating the unbalance of a rotating machine.

The Random Forest also had the lowest Mean Absolute Error (MAE) values and the highest coefficient of determination (R2), demonstrating its ability to make more accurate predictions on the unbalance of a rotating machine and explain a greater percentage of the data variability.

Table 4. Regression Models comparison

Metric	MSE	MAE	R2
Decision Tree	13.286442	2.383776	0.722186
Random Forest	9.126615	2.118423	0.809166
SVR	19.371223	3.156162	0.594955
KNN	11.956285	2.223472	0.749999
Nural Network	17.728253	2.984338	0.629309

It is important to note that the unbalance ranged from 0 to 20.4 grams during the experiments, with increments of 6.8 grams. Therefore, the Random Forest was able to predict the unbalance with an average precision of 2.12 grams, which represents an absolute variation of only 31.2% compared to the smallest unit of unbalance. This indicates that the chosen model is effective in estimating the unbalance in rotating machines, aiding in the identification and prevention of failures and contributing to predictive maintenance and the improvement of operational efficiency.

3.3 Final model

After training the individual LSTM Predictor and regression models, a pipeline was created to perform the analysis and prediction of unbalance in rotating machines. The pipeline is presented in Figure 5.

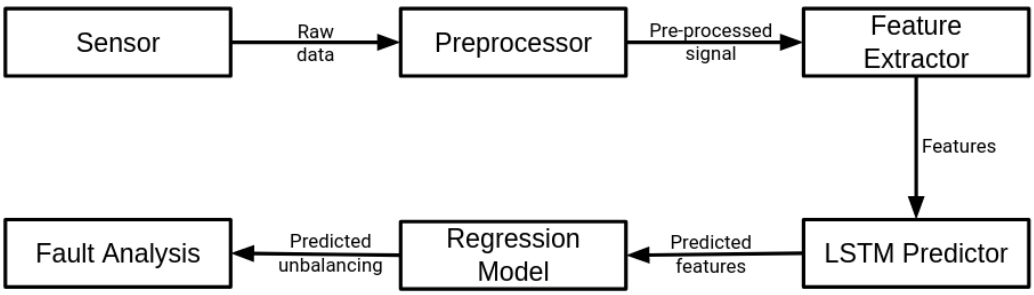


Figure 5. Methodology flowchart

In this model, the LSTM function is to predict future values of features based on historical data and then use these predicted features as input to the regression model, which estimates the unbalance. Finally, the predicted result is compared

with acceptable unbalance values in the fault analysis step, generating reports and issuing alerts in case of significant failures. The results obtained with the complete model can be seen in Table 5.

Table 5. Final Model errors metrics

Metric	Value
MSE	25.8794
MAE	3.9852,
R2	0.3781

This final machine learning model, made by a combination of a LSTM Predictor and a regressor, has shown the ability to make accurate predictions and estimates of unbalance in rotating machines. Considering that the unbalance ranged from 0 to 20.4 grams during the experiments, an MAE of 3.99 indicates that the model has an absolute variation smaller than the smallest increment of unbalance, which is considered an satisfactory margin of error for efficient and secure implementation.

4. CONCLUSIONS

In conclusion, this study successfully applied machine learning techniques, combining a LSTM model with an regression model, for the detection of unbalance in rotary machines. By utilizing time series data, applying the necessary preprocessor and extracting relevant features, the proposed model demonstrated accurate predictions and estimations of machine unbalance. The results showed low errors in terms of MSE and MAE, indicating the model's ability to generalize well to unseen data. Additionally, the high coefficient of determination (R2) highlighted the model's capability to explain a considerable portion of the variability in the test data. The combination of LSTM and regression proved to be effective in forecasting and estimating machine unbalance, enabling proactive identification and analysis of faults. This approach contributes to the advancement of machine learning in the field of predictive maintenance and enhances operational efficiency by preventing unplanned downtime and minimizing potential damages. The findings from this study emphasize the relevance and applicability of the proposed model, providing valuable insights for understanding and predicting machine behavior accurately.

5. ACKNOWLEDGEMENTS

Eduardo M. O. Lopes acknowledges the support of CNPq - National Council for Scientific and Technological Development for his scholarship, as Maurizio R. Barghouthi and Daniel A. E. Canto acknowledge the support of CAPES - Coordination for the Improvement of Higher Education Personnel for their scholarship.

6. REFERENCES

Hu, Q., Si, X.S., Zhang, Q.H. and Qin, A.S., 2020. "A rotating machinery fault diagnosis method based on multi-scale dimensionless indicators and random forests". *Mechanical Systems and Signal Processing*, Vol. 139, p. 106609. ISSN 0888-3270. doi:https://doi.org/10.1016/j.ymssp.2019.106609. URL `https://www.sciencedirect.com/science/article/pii/S0888327019308301`.

Kawai, K., Irino, N., Iiyama, K., Matsubara, A., Mizuyama, H., Mori, K. and Fujishima, M., 2020. "A prediction model for high efficiency machining conditions based on machine learning". Elsevier B.V., Vol. 101, pp. 54–57. ISBN 9781713835431. ISSN 22128271. doi:10.1016/j.procir.2020.09.188.

Ma, X., Xie, X., Li, X., Li, X. and Li, J., 2021. "Research on the classification algorithm of elevator faults based on machine learning". *Applied Sciences*, Vol. 11, No. 7, p. 3073. doi:10.3390/app11073073.

Ribeiro, F.M.L., 2018. "Similarity-based methods for machine diagnosis".

Schwendemann, S., Amjad, Z. and Sikora, A., 2021. "A survey of machine-learning techniques for condition monitoring and predictive maintenance of bearings in grinding machines". doi:10.1016/j.compind.2020.103380.

Sestito, G.S., Venter, G.S., Ribeiro, K.S.B., Rodrigues, A.R. and da Silva, M.M., 2022. "In-process chatter detection in micro-milling using acoustic emission via machine learning classifiers". *International Journal of Advanced Manufacturing Technology*, Vol. 120, pp. 7293–7303. ISSN 14333015. doi:10.1007/s00170-022-09209-w.

Theissler, A., Pérez-Velázquez, J., Kettelgerdes, M. and Elger, G., 2021. "Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry". *Reliability Engineering and System Safety*, Vol. 215. ISSN 09518320. doi:10.1016/j.ress.2021.107864.

Xu, H., Liu, J., Peng, X., Wang, J. and He, C., 2022. "Deep dynamic adaptation network: a deep transfer learning framework for rolling bearing fault diagnosis under variable working conditions". *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 45, No. 1, p. 41. ISSN 1806-3691. doi:10.1007/s40430-022-03950-9. URL `https://doi.org/10.1007/s40430-022-03950-9`.

Zhu, Z., Lei, Y., Qi, G., Chai, Y., Mazur, N., An, Y. and Huang, X., 2023. "A review of the application of deep learning in intelligent fault diagnosis of rotating machinery". *Measurement*, Vol. 206, p. 112346. ISSN 0263-2241. doi:https://doi.org/10.1016/j.measurement.2022.112346. URL https://www.sciencedirect.com/science/article/pii/S0263224122015421.

Żabiński, T., Mączka, T., Kluska, J., Madera, M. and Sęp, J., 2019. "Condition monitoring in industry 4.0 production systems - the idea of computational intelligence methods application". *Procedia CIRP*, Vol. 79, pp. 63–67. ISSN 2212-8271. doi:https://doi.org/10.1016/j.procir.2019.02.012. URL https://www.sciencedirect.com/science/article/pii/S221282711930126X. 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy.

7. RESPONSABILITY NOTICE

The authors are the only responsible for the printed material included in this paper.